UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

Learning to Generate Things and Stuff:

Guided Generative Adversarial Networks for Generating Human Faces, Hands, Bodies, and Natural Scenes

Hao Tang

Advisor

Prof. Dr. Nicu Sebe

University of Trento

May 2021

Contents

Publications

This thesis consists of the following publications:

- Chapter 2:
  - **Hao Tang**, Wei Wang, Dan Xu, Yan Yan, Nicu Sebe. GestureGAN for Hand Gesture-to-Gesture Translation in the Wild. *ACM MM 2018*, Seoul, Korea. **Oral, Best Paper Nomination (4/757)**, **Code**
  - **Hao Tang**, Hong Liu, Nicu Sebe. Unified Generative Adversarial Networks for Controllable Image-to-Image Translation. *IEEE Transactions on Image Processing (TIP)*, 2020. **Code**
- Chapter 3:
  - **Hao Tang**, Dan Xu, Gaowen Liu, Wei Wang, Yan Yan, Nicu Sebe. Cycle In Cycle Generative Adversarial Networks for Keypoint-Guided Image Generation. ACM MM 2019, Nice, France. **Oral**, **Code**
  - **Hao Tang**, Nicu Sebe. Total Generate: Cycle In Cycle Generative Adversarial Networks for Generating Human Faces, Hands, Bodies and Natural Scenes. *Submitted to TMM.* **Code**
- Chapter 4:
  - **Hao Tang**, Song Bai, Li Zhang, Philip H.S. Torr, Nicu Sebe. XingGAN for Person Image Generation. *ECCV 2020*, Glasgow, UK. **Code**
  - **Hao Tang**, Song Bai, Philip H.S. Torr, Nicu Sebe. Bipartite

Graph Reasoning GANs for Person Image Generation. *BMVC 2020*, Manchester, UK. **Oral**, **Code**

  - **Hao Tang**, Bin Ren, Ling Shao, Philip H.S. Torr, Nicu Sebe. Cross-Attention Is What You Need for Person Image Generation and Virtual Try-On. *Submitted to TPAMI.* **Code**
  - **Hao Tang**, Ling Shao, Philip H.S. Torr, Nicu Sebe. Bipartite Graph Reasoning GANs for Person Pose and Facial Image Synthesis. *Submitted to TPAMI.* **Code**

- Chapter 5:
  - **Hao Tang**, Dan Xu, Nicu Sebe, Yanzhi Wang, Jason J. Corso, Yan Yan. Multi-Channel Attention Selection GAN with Cascaded Semantic Guidance for Cross-View Image Translation. *CVPR 2019*, Long Beach, USA. **Oral**, **Code**
  - **Hao Tang**, Dan Xu, Yan Yan, Jason J. Corso, Philip H.S. Torr, Nicu Sebe. Multi-Channel Attention Selection GANs for Guided Image-to-Image Translation. *arXiv preprint arXiv:2002.01048*, 2020. *Submitted to TPAMI.* **Code**

- Chapter 6:
  - **Hao Tang**, Dan Xu, Yan Yan, Philip H.S. Torr, Nicu Sebe. Local Class-Specific and Global Image-Level Generative Adversarial Networks for Semantic-Guided Scene Generation. *CVPR 2020*, Seattle, USA. **Code**
  - **Hao Tang**, Ling Shao, Philip H.S. Torr, Nicu Sebe. Local and Global GANs with Semantic-Aware Upsampling for Image Generation. *Submitted to TPAMI.* **Code**
  - **Hao Tang**, Xiaojuan Qi, Dan Xu, Philip H.S. Torr, Nicu Sebe. Edge Guided GANs with Semantic Preserving for Semantic Image Synthesis. *arXiv preprint arXiv:2003.13898*, 2020. *Submitted to ICCV 2021.* **Code**

- Chapter 7:
  - **Hao Tang**, Sergey Tulyakov, Bin Duan, Ling Shao, Nicu Sebe. Can't Stop Dancing: Music-Guided Dance Video Synthesis. *Submitted to ICCV 2021.* **Code**

The following papers were published during the course of the Ph.D but are not included in this thesis:

- **Hao Tang**, Song Bai, Nicu Sebe. Dual Attention GANs for Semantic Image Synthesis. *ACM MM 2020*, Seattle, USA. **Code**
- **Hao Tang**, Hong Liu, Wei Xiao, Nicu Sebe. When Dictionary Learning Meets Deep Learning: Deep Dictionary Learning and Coding Network for Image Recognition with Limited Data. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2020. **Code**
- **Hao Tang**, Wei Wang, Songsong Wu, Xinya Chen, Dan Xu, Nicu Sebe, Yan Yan. Expression Conditional GAN for Facial Expression-to-Expression Translation. *ICIP 2019*, Taipei, Taiwan. **Oral**
- **Hao Tang**, Dan Xu, Nicu Sebe, Yan Yan. Attention-Guided Generative Adversarial Networks for Unsupervised Image-to-Image Translation. *IJCNN 2019*, Budapest, Hungary. **Oral**, **Code**
- **Hao Tang**, Xinya Chen, Wei Wang, Dan Xu, Jason J. Corso, Nicu Sebe, Yan Yan. Attribute-Guided Sketch Generation. *FG 2019*, Lille, France. **Oral**
- **Hao Tang**, Heng Wei, Wei Xiao, Wei Wang, Dan Xu, Yan Yan, Nicu Sebe. Deep Micro-Dictionary Learning and Coding Network. *WACV 2019*, Hawaii, USA. **Oral**, **Code**
- **Hao Tang**, Dan Xu, Wei Wang, Yan Yan, Nicu Sebe. Dual Generator Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. *ACCV 2018*, Perth Western, Australia. **Oral**, **Code**
- **Hao Tang**, Hong Liu, Wei Xiao, Nicu Sebe. Fast and Robust Dynamic Hand Gesture Recognition via Key Frames Extraction and Fea-

ture Fusion. *Elsevier Neurocomputing*, 2018. **Code**

- Songsong Wu, Yan Yan, **Hao Tang**, Jianjun Qian, Jian Zhang, Xiaoyuan Jing, Yuning Dong. Structured Discriminative Tensor Dictionary Learning for Unsupervised Domain Adaptation. *Elsevier Neurocomputing*, 2021.
- Bin Duan, **Hao Tang**, Wei Wang, Ziliang Zong, Guowei Yang, Yan Yan. Audio-Visual Event Localization via Recursive Fusion by Joint Co-Attention. *WACV 2021*, Hawaii, USA.
- Lei Ding, **Hao Tang**, Lorenzo Bruzzone. Improving Semantic Segmentation of Aerial Images Using Patch-based Attention. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 2020.
- Xinya Chen, Yanrui Bin, Changxin Gao, Nong Sang, **Hao Tang**. Relevant Region Prediction for Crowd Counting. *Elsevier Neurocomputing*, 2020.
- Jichao Zhang, Jingjing Chen, **Hao Tang**, Wei Wang, Yan Yan, Enver Sanginetom, Nicu Sebe. Dual In-painting Model for Unsupervised Gaze Correction and Animation in the Wild. *ACM MM 2020*, Seattle, USA. **Code**
- Hao Ding, Songsong Wu, **Hao Tang**, Fei Wu, Guangwei Gao, Xiaoyuan Jing. Cross-View Image Synthesis with Deformable Convolution and Attention Mechanism. *PRCV 2020*, Nanjing, China.
- Bin Duan, Wei Wang, **Hao Tang**, Hugo Latapie, Yan Yan. Cascade Attention Guided Residue Learning GAN for Cross-Modal Translation. *ICPR 2020*, Milan, Italy. **Code**
- Gaowen Liu, **Hao Tang**, Hugo Latapie, Yan Yan. Exocentric to Egocentric Image Generation via Parallel Generative Adversarial Network. *ICASSP 2020*, Barcelona, Spain. **Oral**
- Songsong Wu, Zhiqiang Lu, **Hao Tang**, Yan Yan, Songhao Zhu, Xiaoyuan Jing, Zuoyong Li. Joint Learning of Self-Representation and

Indicator for Multi-View Image Clustering. *ICIP 2019*, Taipei, Taiwan.

- Dan Xu, Wei Wang, **Hao Tang**, Nicu Sebe, Elisa Ricci. Structured Attention Guided Convolutional Neural Fields for Monocular Depth Estimation. *CVPR 2018*, Salt Lake City, USA. **Spotlight**, **Code**

Chapter 1

Introduction

1.1 Background

In this thesis, we mainly focus on how to enable machines to generate a target image given an input image. This has many application scenarios such as human-computer interaction, entertainment, virtual reality, and data augmentation. However, this problem is challenging since it needs a high-level semantic understanding of the image mapping between the input domain and the output domain. Recently, Generative Adversarial Networks (GANs) [41] have shown the potential to solve this challenging task and can be utilized, for example, to translate a neutral face into a smiling face or to transfer a specific pose into different poses. GANs are generative models based on game theory, and consist of a generator and a discriminator where the goal of the generator is to produce photo-realistic images so that the discriminator cannot tell the generated images apart from real images.

GANs have achieved impressive performance in a wide range of applications such as high-quality image generation [63, 10, 156, 140, 163], image inpainting [30, 217], cross-modal retrieval/matching [198, 199], semantic segmentation [175, 180], object detection [80, 189], depth estimation [7], and image/action recognition [174, 117]. However, it is still hard for vanilla

GANs to generate images in a controlled setting. To generate images that meet user requirements, Conditional GAN (CGAN) [103] is employing the conditioned guidance information to guide the image generation process. A CGAN model always combines a vanilla GAN and an external source, such as discrete class labels or tags [113, 123, 25], text descriptions [208, 78], object keypoints [132, 98], human skeletons [163, 146], segmentation maps [133, 134, 119, 92], conditional images [55], object masks [105], and attention maps [166, 20, 100].

In this thesis, we mainly focus on image generation. However, one can still observe unsatisfying results produced by existing state-of-the-art methods. To address this limitation and further improve the quality of generated images, we propose a few novel models. The image generation task can be roughly divided into three subtasks, i.e., person image generation, scene image generation, and cross-modal translation (Figures 1.1 and 1.2). Person image generation can be further divided into three subtasks, namely, hand gesture generation, facial expression generation, and person pose generation. Meanwhile, scene image generation can be further divided into two subtasks, i.e., cross-view image translation and semantic image synthesis. For each task, we have proposed the corresponding solution. Specifically, for hand gesture generation, we have proposed the Gesture-GAN framework. For facial expression generation, we have proposed the Cycle-in-Cycle GAN (C2GAN) framework. For person pose generation, we have proposed the XingGAN and BiGraphGAN frameworks. For cross-view image translation, we have proposed the SelectionGAN framework. For semantic image synthesis, we have proposed the Local and Global GAN (LGGAN), EdgeGAN, and Dual Attention GAN (DAGAN) frameworks. Although each method was originally proposed for a certain task, we later discovered that each method is universal and can be used to solve different tasks. For instance, GestureGAN can be used to solve both hand gesture

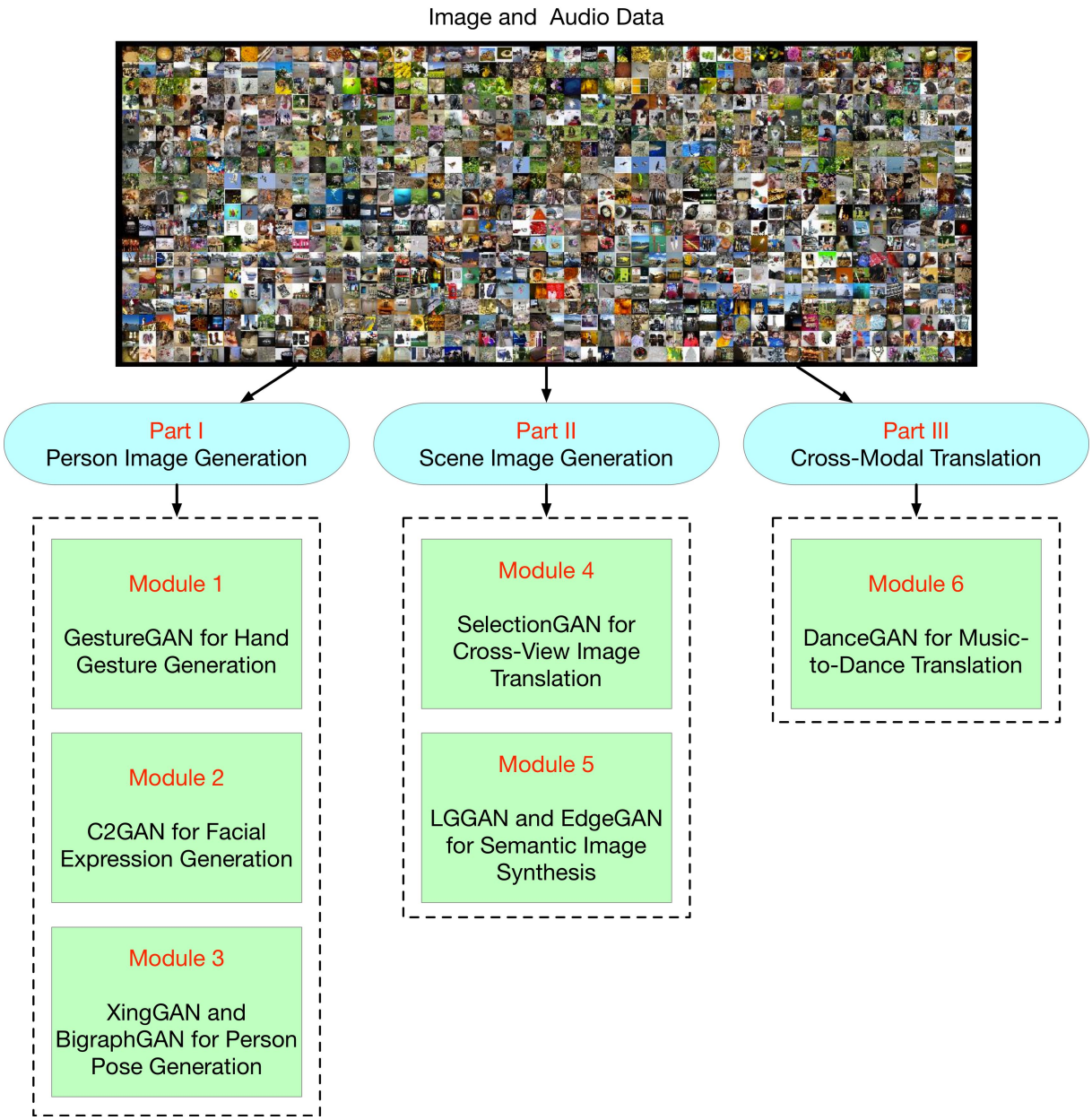


Figure 1.1: The illustration of our approachs for image generation.

generation and cross-view image translation tasks, as shown in Chapter 2. C2GAN can be used to solve facial expression generation, person pose generation, hand gesture generation, and cross-view image translation, as shown in Chapter 3. SelectionGAN can be used to solve cross-view image translation, facial expression generation, person pose generation, hand gesture generation, and semantic image synthesis, as shown in Chapter 5.

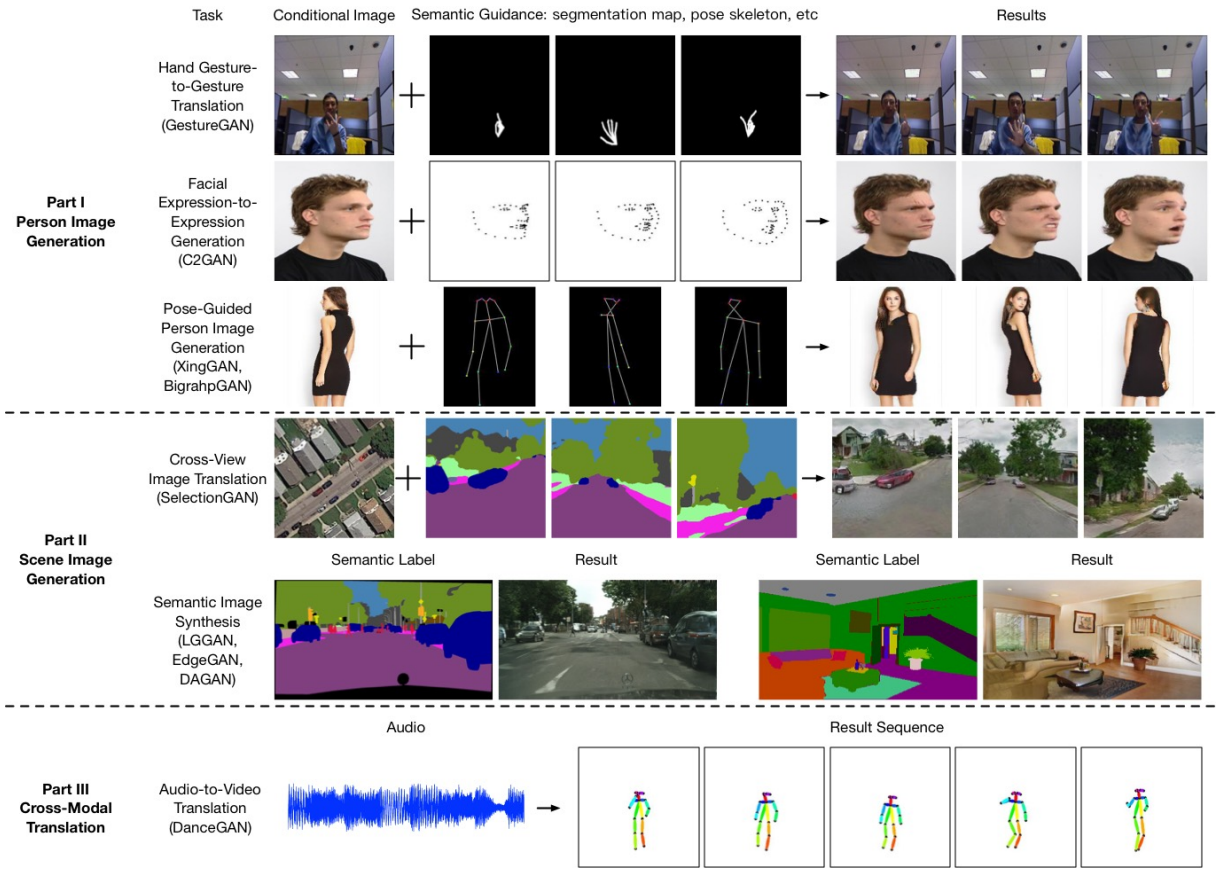


Figure 1.2: Example results of the proposed methods in this thesis.

Moreover, we explore cross-modal translation and propose a novel DanceGAN for audio-to-video translation.

1.2 Thesis Overview

In this thesis, we explore Generative Adversarial Network [41] to learn to generate images including human faces, hands, bodies, and natural scenes. See Figures 1.1 and 1.2 for better understanding. Specifically:

Part I. Person Image Generation. Chapter 2 describes the proposed GestureGAN [163] for hand gesture-to-gesture translation, which can generate target images with arbitrary poses, sizes, structures, and locations in the wild. We also propose three novel objective functions to better

optimize the proposed GAN model, i.e., color loss, controllable structure guided cycle consistency loss, and self-content preserving loss. These optimization functions and the proposed GAN framework are jointly trained in an end-to-end fashion to improve both the quality and fidelity of the generated images. Lastly, we introduce an efficient Fréchet ResNet Distance (FRD) metric to evaluate the similarity of the real and generated images, which is more consistent with human judgment.

Chapter 3 introduces the proposed C2GAN [164] for facial expression-to-expression translation, which organizes the guidance and the image data in an interactive manner, instead of using as input only the guidance information. The proposed cycle in cycle network structure is a new design that explores the effective use of cross-modal information for guided image-to-image translation tasks. The designed cycled sub-networks connect different modalities and implicitly constraint each other, leading to extra supervision signals for better image generation. We also investigate cross-modal discriminators and cycle losses for more robust network optimization.

Chapter 4 presents the proposed XingGAN [156] and BiGraphGAN [155] for pose-guided preson image generation, respectively. XingGAN explores cascaded guidance with two different generation branches, and aims at progressively producing a more detailed synthesis from both person shape and appearance embeddings. We then propose SA and AS blocks, which effectively transfer and update person shape and appearance features in a crossing way to mutually improve each other, and are able to significantly boost the quality of the final outputs.

Moreover, the proposed BiGraphGAN aims to progressively reason the pose-to-pose and pose-to-image relations via two novel proposed blocks. We also propose a novel Bipartite Graph Reasoning (BGR) block to effectively reason the crossing long-range relations between the source pose

and the target pose in a bipartite graph by using Graph Convolutional Networks (GCNs). Finally, we present a new Interaction-and-Aggregation (IA) block to interactively enhance both person's appearance and shape feature representations.

Part II. Scene Image Generation. Chapter 5 introduces the proposed SelectionGAN [165] for cross-view image translation, which explores cascaded semantic guidance with a coarse-to-fine inference, and aims at producing a more detailed synthesis from richer and more diverse multiple intermediate generations. We also propose a novel multi-scale spatial pooling & channel selection module, which is utilized to automatically enhance the multi-scale feature representation in both spatial and channel dimensions. We introduce a novel multi-channel attention selection module, which is utilized to attentively select interested intermediate generations and is able to significantly boost the quality of the final output. The multi-channel attention module also effectively learns uncertainty maps to guide the pixel loss for more robust optimization.

Chapter 6 presents the proposed LGGAN [169] and EdgeGAN [161] for semantic image synthesis, respectively. LGGAN explores image generation from the local context, which we believe is beneficial for generating richer details compared with the existing global image-level generation methods. A new local class-specific generative structure is designed for this purpose. It can effectively handle the generation of small objects and details, which are common difficulties encountered by the global-based generation. We also propose a novel global and local generative adversarial network design able to take into account both the global and local contexts. To stabilize the optimization of the proposed joint network structure, a fusion weight map generator and a dual-discriminator are introduced. Moreover, to learn discriminative class-specific feature representations, a novel classification module is proposed. We also introduce a novel semantic-aware upsampling

(SAU) to dynamically upsample a small subset of relevant pixels based on the semantic information. SAU is more efficient than deconvolution, pixel shuffle, and spatial attention, and can capture more complete semantic information than traditional upsampling methods such as nearest-neighbor interpolation.

At the same time, we propose a novel EdgeGAN for the challenging semantic image synthesis task. To the best of our knowledge, we are the first to explore the edge generation from semantic layouts and then utilize the generated edges to guide the generation of realistic images. We then propose an effective attention guided edge transfer module to selectively transfer useful edge structure information from the edge generation branch to the image generation branch. We also design a new semantic preserving module to highlight class-dependent feature maps based on the input semantic label map for generating semantically consistent results. Both ideas have not been investigated by any existing GAN-based generation works.

Part III. Cross-Modal Translation. Chapter 7 describes the proposed DanceGAN for cross-modal audio-to-video translation, which consists of two generation stages, i.e., music-to-skeleton translation, and skeleton-to-video translation. We also propose two graph attention networks to explicitly model dependencies across joints spatially and temporally at the same time. To the best of our knowledge, we are the first to explore the spatial and temporal graph attention networks to guide the generation of coordinated and coherent skeleton sequence from the input music. We also design a new self-supervised regularization network to enhance the video generation process from both forward and backward directions, which has not been investigated by any existing GAN-based generation works.

Conclusion. In Chapter 8, we summarize the contributions of this thesis and discuss several future directions in multi-modal synthesis including

image, text, audio, video, and 3D object.

Part I

Person Image Generation

Chapter 2

GestureGAN

We propose a unified Generative Adversarial Network (GAN) model for controllable image-to-image translation, i.e., transferring an image from a source to a target dn guided by controllable structures. In addition to conditioning on a reference image, we show how the model can generate images conditioned on controllable structures, e.g., class labels, object keypoints, human skeletons, and scene semantic maps. The proposed model consists of a single generator and a discriminator taking a conditional image and the target controllable structure as input. In this way, the conditional image can provide appearance information and the controllable structure can provide the structure information for generating the target result. Moreover, our model learns the image-to-image mapping through three novel losses, i.e., color loss, controllable structure guided cycle-consistency loss, and controllable structure guided self-content preserving loss. Also, we present the Fréchet ResNet Distance (FRD) to evaluate the quality of the generated images. Experiments on two challenging image translation tasks, i.e., hand gesture-to-gesture translation and cross-view image translation, show that our model generates convincing results, and significantly outperforms other state-of-the-art methods on both tasks. Meanwhile, the proposed framework is a unified solution, thus it can be applied to solv-

ing other controllable structure guided image translation tasks, such as landmark guided facial expression translation and keypoint guided person image generation. To the best of our knowledge, we are the first to make one GAN framework work on all such controllable structure guided image translation tasks. The source code and trained models are available at https://github.com/Ha0Tang/GestureGAN.

2.1 Introduction

At a high level, current image-to-image translation techniques usually fall into one of two types: supervised/paired [55, 184] and unsupervised/unpaired [231, 25]. However, existing image-to-image translation frameworks are inefficient for the multi-domain image-to-image translation task. For instance, given n different image domains, Pix2pix [55] and BicycleGAN [232] need to train $A_n^2{=}n(n{-}1){=}\Theta(n^2)$ models. CycleGAN [231], DiscoGAN [67] and DualGAN [206] need to train $C_n^2{=}\frac{n(n-1)}{2}{=}\Theta(n^2)$ models, or $n(n{-}1)$ generator/discriminator pairs since one model has two different generator/discriminator pairs for these methods. ComboGAN [5] requires $n{=}\Theta(n)$ models. G^{2}GAN [167] needs to train two generators, i.e., the generation generator and the reconstruction generator, while StarGAN [25] only needs one model. However, for some specific image-to-image translation applications such as hand gesture-to-gesture translation [163] and person image generation [98], n could be arbitrary large since hand gestures and human bodies in the wild can have arbitrary poses, sizes, appearances, locations, and self-occlusions.

To address these limitations, several works have been proposed to generate images based on controllable structures, e.g., object keypoints, human skeleton, and scene semantic maps. These works can be divided into three different categories: (1) Object keypoint guide methods. Reed et

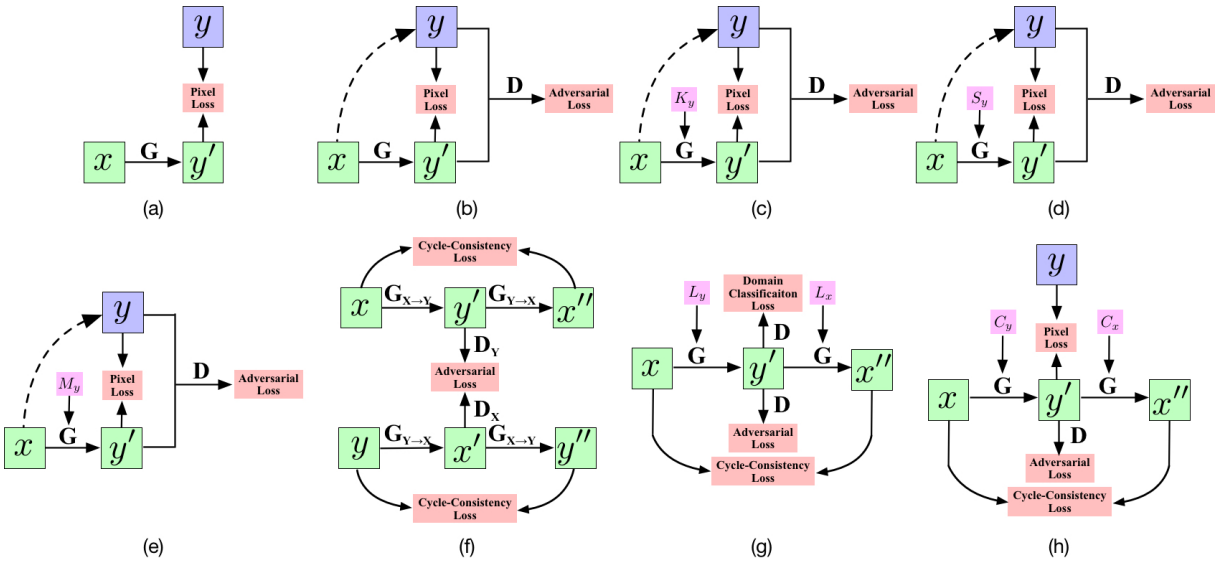


Figure 2.1: Comparison with state-of-the-art image-to-image translation methods. (a) Traditional deep learning methods, e.g., Context Encoder [121]. (b) Adversarial learning methods, e.g., Pix2pix [55] and BicycleGAN [232]. (c) Keypoint-guided image generation methods, e.g., PG^2 [98], G2GAN [149] and DPIG [99]. (d) Skeleton-guided image generation methods, e.g., SAMG [201] and PoseGAN [146]. (e) Semantic-guided image generation methods, e.g., SelectionGAN [165] and X-Fork [133]. (f) Adversarial unsupervised learning methods, e.g., CycleGAN [231], DiscoGAN [67] and DualGAN [206]. (g) Multi-domain image translation methods, e.g., ComboGAN [5], G^{2}GAN [167] and StarGAN [25]. (h) Proposed GAN model in this paper. Note that the proposed GAN model is a unified solution for controllable structure guided image-to-image translation problem, i.e., controllable structure C can be one of class label L, object keypoint K, human skeleton S or semantic map M. Notations: x and y are the real images; x' and y' are the generated images; x'' and y'' are the reconstructed images; K_y is the keypoint of y; S_y is the skeleton of y; M_y is the semantic map of y; L_x and L_y are the class labels of x and y, respectively; C_x and C_y are the controllable structures of x and y, respectively; G, $G_{X\mapsto Y}$ and $G_{Y\mapsto X}$ represent generators; D, D_Y and D_X denote discriminators.

al. [132] proposed GAWWN, which generates bird images conditioned on bird keypoints. Song et al. [149] propose G2GAN for facial expression synthesis based on facial landmarks. Ma et al. propose PG^2 [98] and a two-stage reconstruction pipeline [99] achieving person image translation using a conditional image and a target pose image. (2) Human skeleton guided methods. Siarohin et al. [146] introduce PoseGAN based on the human skeleton for human image generation. Tang et al. [163] propose a

novel GestureGAN conditioned hand skeleton for hand gesture-to-gesture image translation tasks. Yan et al. [201] propose a method to generate human motion sequences with simple backgrounds using CGANs and human skeleton information. (3) Scene semantic guide methods. Wang et al. [184] propose Pix2pixHD, which can be used for turning semantic label maps into photo-realistic images or synthesizing portraits from face label maps. Park et al. [119] propose the spatially-adaptive normalization, a simple but effective layer for synthesizing images given an input semantic layout. Regmi and Borji [133] propose X-Fork and X-Seq, which aim to generate images across two drastically different views by using the guidance of semantic maps.

The aforementioned methods have achieved impressive results on the corresponding tasks. However, each of them is tailored for a specific application limiting their capability to generalize. Our framework does not impose any application-specific constraint. This makes our setup considerably simpler than the other approaches (see Figure 2.1). To achieve this goal, we propose a unified solution for controllable image-to-image translation. It allows generating high-quality images with arbitrary poses, sizes, structures, and locations in the wild. Our GAN model only consists of one generator and one discriminator, taking a conditional image and the novel controllable structures as inputs. In this way, the conditional image can provide appearance information and the controllable structures can provide structure information for generating the target image. In addition, to better learn the mapping between inputs and outputs, we propose three novel losses, i.e., color loss, controllable structure guided cycle-consistency loss, and self-content preserving loss. The proposed color loss can handle the problem of 'channel pollution' that is frequently occurring in generative models such as PG^2 [98], leading the generated images to be sharper and having higher quality. The proposed controllable structure guided cycle-

consistency loss is more flexible than the one proposed in CycleGAN [231], reducing further the space of possible mappings between different domains. The proposed self-content preserving loss can preserve color composition, object identity, and global layout of generated images. These optimization loss functions and the proposed GAN framework are jointly trained in an end-to-end fashion to improve both fidelity and visual naturalness of the generated images. Furthermore, we propose the Fréchet ResNet Distance (FRD), which is a novel and better evaluation metric to evaluate the generated images of GANs. Extensive experiments on two challenging controllable image-to-image translation tasks with four different datasets, i.e., hand gesture-to-gesture translation and cross-view image translation, demonstrate that the proposed GAN model generates high-quality images with convincing details and achieves state-of-the-art performance on both tasks. Finally, the proposed GAN model is a general-purpose solution that can be applied to solve a wide variety of controllable structure guided image-to-image translation problems.

In summary, the contributions of this paper are as follows:

- We propose a unified GAN model for controllable image-to-image translation tasks, which can generate target images with arbitrary poses, sizes, structures, and locations in the wild.
- We propose three novel objective functions to better optimize the proposed GAN model, i.e., color loss, controllable structure guided cycle consistency loss, and self-content preserving loss. These optimization functions and the proposed GAN framework are jointly trained in an end-to-end fashion to improve both the quality and fidelity of the generated images.
- We propose an efficient Fréchet ResNet Distance (FRD) metric to evaluate the similarity of the real and generated images, which is more consistent with human judgment.

- Qualitative and quantitative results demonstrate the superiority of the proposed GAN model over the state-of-the-art methods on two challenging controllable image translation tasks with four datasets, i.e., hand gesture-to-gesture translation and cross-view image translation.

Part of this work has been published in [163]. We extend it in numerous ways: (1) We extend GestureGAN proposed in [163] to a unified GAN framework for handling all controllable image-to-image translation tasks. (2) We further tune our whole pipeline and improve its performance and generalizability for hand gesture-to-gesture translation and cross-view image translation by employing three additional losses, i.e., controllable structure guided self-content preserving loss, perceptual loss, and Total Variation loss. Moreover, we extend the one-cycle framework in [163] to a two-cycle framework and validate the effectiveness. (3) We extend the experimental evaluation provided in [163] in several directions. First, we conduct extensive experiments on two challenging generative tasks with four different datasets, demonstrating the wide application scope of our GAN framework. Second, we conduct exhaustive ablation studies to evaluate each component of the proposed method. Third, we investigate the influence of hyper-parameters on generation performance. Forth, we compare the model parameters of different methods. Lastly, we provide arbitrary image translation results on both tasks.

2.2 Related Work

Generative Adversarial Networks (GANs) are unsupervised learning methods and have been proposed in [41]. Recently, GANs have shown promising results in various applications, e.g., image generation [63, 10, 215]. Existing approaches employ the idea of GANs for conditional image generation, such as image-to-image translation [55], text-to-image transla-

tion [130, 171], audio-to-image [34], and sketch generation [157, 18]. The key success of GANs is the adversarial loss, which allows the model to generate images that are indistinguishable from real images, and this is exactly the goal that many tasks aim to optimize. In this paper, we mainly focus on image-to-image translation tasks.

Image-to-Image Translation is the problem of transferring an image from a source domain to a target domain, which uses input-output data to learn a parametric mapping between inputs and outputs, e.g., Isola et al. [55] propose Pix2pix, which uses a conditional GAN to learn a translation function from input to output image domains with paired training data. However, collecting large sets of image pairs is often prohibitively expensive or unfeasible. To solve this limitation, Zhu et al. [231] propose CycleGAN, which can learn to translate between domains without paired input-output examples by using the cycle consistency loss. Similar ideas have been proposed in several works [206, 25, 167]. For example, Choi et al. [25] introduce StarGAN, which can perform image-to-image translation for multiple domains.

However, existing image-to-image translation models are inefficient and ineffective. For example, with n image domains, CycleGAN [231], DiscoGAN [67], and DualGAN [206] need to train $2C_n^2{=}n(n{-}1){=}\Theta(n^2)$ generators and discriminators, while Pix2pix [55] and BicycleGAN [232] have to train $A_n^2{=}n(n{-}1){=}\Theta(n^2)$ generator/discriminator pairs. Recently, Anoosheh et al. propose ComboGAN [5], which only needs to train n generator and discriminator pairs for n different image domains, having a complexity of $\Theta(n)$. Tang et al. [167] propose G^{2}GAN, which can perform image-to-image translations for multiple domains using only two generators, i.e., the generation generator and the reconstruction generator. Additionally, Choi et al. [25] propose StarGAN, in which a single generator and a discriminator can perform unpaired image-to-image translations for multiple

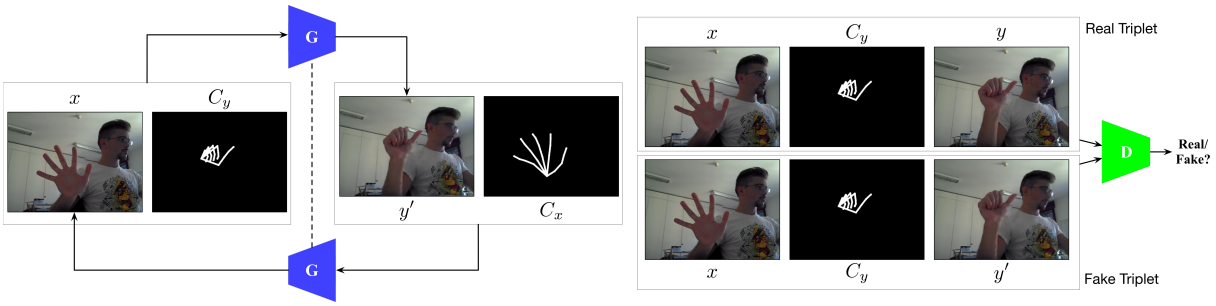


Figure 2.2: Pipeline of the proposed unified GAN model for controllable image-to-image translation tasks. The proposed GAN framework consists of a single generator G and an associated adversarial discriminator D, which takes a conditional image x and a controllable structure C_y as input to produce the target image y'. We have two cycles and here we only show one of them. Note that the controllable structure C_y can be a class label, object keypoints, human skeletons, semantic maps, etc.

domains. Although the computational complexity of StarGAN is $\Theta(1)$, this model has only been validated on the face attributes modification task with clear background and face cropping. More importantly, for some specific image-to-image translation tasks such as hand gesture-to-gesture translation [163] and person image generation [98, 155] tasks, the image domains could be arbitrarily large, e.g., hand gestures and human bodies in the wild can have arbitrary poses, sizes, appearances, structures, locations, and self-occlusions. The aforementioned approaches are not effective in solving these specific situations.

Controllable Image-to-Image Translation. To fix these limitations, several recent works have been proposed to generate persons, birds, faces and scene images based on controllable structures, i.e., object keypoints [132, 98, 99], human skeletons [201, 164, 146, 156] and semantic maps [184, 119, 133, 169]. In this way, controllable structures provide four types of information to guide the image generation process, i.e., category, scale, orientation, and location. Although significant efforts have been made to achieve controllable image-to-image translation in the area of computer vision, there has been very limited research on universal controllable image translation. That is, the typical problem with the aforementioned generative models is

that each of them is tailored for a specific application, which greatly limits the generalization ability of the proposed models. To handle this problem, we propose a novel and unified GAN model, which can be tailored for handling all kinds of problem settings of controllable structure guided image-to-image translation, including object keypoint guided generative tasks, human skeleton guided generative tasks and semantic map guided generative tasks, etc.

2.3 Model Description

In this section, we present the details of the proposed GAN model (Figure 2.2). We present a controllable structure guided generator, which utilizes the images from one domain and conditional controllable structures from another domain as inputs and produces images in the target domain. Moreover, we propose a novel discriminator which also takes the controllable structure into consideration. The proposed GAN model is trained in an end-to-end fashion mutually benefiting from the generator and the discriminator.

2.3.1 Controllable Structure Guided Generator

Controllable Structure Guided Generation. Image-to-image translation tasks, such as hand gesture-to-gesture translation [163], person image generation [98], facial expression-to-expression translation [164] and cross-view image translation [133] are very challenging. In these tasks, the source domain and the target domain may have large deformations. Moreover, these tasks can be treated as an infinite mapping problem leading to ambiguity issues in the translation process. For instance, in the hand gesture-to-gesture translation task, if you input a hand gesture image to the generator, it has no idea which gestures should output.

To fix this limitation, we employ controllable structures as conditional guidance to guide the image generation process. The controllable structures can be class labels, object keypoints, human skeletons or semantic maps, etc. Following [98, 146, 133] we generate the controllable structures using deep models pretrained from other large-scale datasets, e.g., we apply the pose estimator OpenPose [13] to obtain approximate human body poses and hand skeletons. Specifically, as shown in Figure 2.2, we concatenate the input conditional image x from the source domain and the controllable structure C_y from a target domain, and input them into the generator G and synthesize the target image $y'=G(x, C_y)$. In this way, the ground-truth controllable structure C_y provides stronger supervision and structure information to guide the image-to-image translation in the deep network, while the conditional image x provides the appearance information to produce the final result y'.

Controllable Structure Guided Cycle. Guided by the controllable structure C_y, our generator can produce the corresponding image y'. However, state-of-the-art controllable image-to-image translation methods such as [98, 146, 165, 133] only consider the image translation process, i.e., from the source domain to the target domain. Different from them, we consider both the image translation process and image reconstruction process, i.e., from the source domain to the target domain and from the target domain back to the source domain. The intuition behind this is that if we translate from one domain to the other and back again we should arrive at where we started. The proposed controllable structure guided cycle is different from the cycle proposed in CycleGAN [231], which uses a cycle consistency loss to preserve the content of its input images while changing only the domain-related part of the inputs. The main difference is that CycleGAN can only handle two different domains, while an image translation problem such as hand gesture-to-gesture translation task has arbitrary domains, e.g., hand

gestures in the wild can have arbitrary poses, sizes, appearances, structures, locations, and self-occlusions. Therefore, we need the controllable structure to guide the learning of the proposed cycle. The proposed controllable structure guided cycle is also different from the cycle proposed in StarGAN [25], which translates an original image into an image in the target domain and then reconstructs the original image from the translated image through feeding the target label. However, class labels can only provide the category information, while the controllable structure can provide four types of information for generation at the same time, i.e., category, location, scale, and orientation. More specifically, as shown in Figure 2.2, the generated image y' and the controllable structure C_x are concatenated to input into the generator G. The proposed controllable structure guided cycle can be formulated as follows,

$$x'' = G(y', C_x) = G(G(x, C_y), C_x) \approx x. \tag{2.1}$$

Note that we use a single generator twice, first to translate an original image into an image in the target domain and then to reconstruct the original image from the translated image. Image translation and image reconstruction are simultaneously considered in our framework, constructing a full mapping cycle. Similarly, we have another cycle,

$$y'' = G(x', C_y) = G(G(y, C_x), C_y) \approx y. \tag{2.2}$$

Controllable Structure Guided Cycle Consistency Loss. To better optimize the proposed cycle, we propose a novel controllable structure guided cycle consistency loss. It is worth noting that CycleGAN [231] is different from the Pix2pix model [55] as the training data in CycleGAN is unpaired. CycleGAN introduces the cycle consistency loss to enforce forward-backward consistency. In that case, the cycle consistency loss can

be regarded as 'pseudo' pairs of training data even though we do not have the corresponding data in the target domain which corresponds to the input data from the source domain. However, in this paper, we introduce the controllable structure guided cycle consistency loss for the paired image-to-image translation task. This loss ensures the consistency between source images and the reconstructed image, and it can be expressed as,

$$\begin{aligned}\mathcal{L}_{cyc}(G, C_x, C_y) =& \mathbb{E}_{x,C_x,C_y}\left[||x - G(G(x, Cy), C_x)||_1\right] \\ &+ \mathbb{E}_{y,C_x,C_y}\left[||y - G(G(y, Cx), C_y)||_1\right],\end{aligned} \tag{2.3}$$

where G is the generator; x and y are the input images; C_x and C_y are the controllable structures of image x and y, respectively. As mentioned before, we use the same generator G twice. Equipped with this loss, the proposed generator G further improves the image quality due to its implicit data augmentation effect from a multi-task learning setting.

2.3.2 Controllable Structure Guided Discriminator

Conditional GANs (CGANs) such as Pix2pix [55] learn the mapping $G(x) \mapsto y$, where x is the input conditional image. Generator G is trained to generate image y' that cannot be distinguished from 'real' image y by an adversarially trained discriminator D, while the discriminator D is trained as well as possible to detect the 'fake' images generated by the generator G. The objective function of CGANs is defined as follows,

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}\left[\log D(x, y)\right] + \mathbb{E}_x\left[\log(1 - D(x, G(x)))\right], \tag{2.4}$$

where generator G tries to minimize this objective function while the discriminator D tries to maximize it. Thus, the solution can be expressed as $G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D)$. In this paper, we try to learn two mappings through one generator, i.e., $G(x, C_y) \mapsto y$ and $G(y', C_x) \mapsto x$. As shown in

Figure 2.2, in order to learn both mappings, we employ the controllable structures explicitly. Thus, the adversarial losses of the two mappings are defined respectively, as follows:

$$\begin{aligned}\mathcal{L}_{adv}(G, D, C_y) =& \mathbb{E}_{[x,C_y],y}\left[\log D([x, C_y], y)\right] \\ &+\mathbb{E}_{[x,C_y]}\left[\log(1 - D([x, C_y], G(x, C_y)))\right],\end{aligned} \tag{2.5}$$

where C_y is the controllable structure of image y and $[\cdot, \cdot]$ represents the concatenation operation. This controllable structure guided input encourages D to capture the local-aware information and generate semantic-matched target images. Similarly, we have another adversarial loss,

$$\begin{aligned}\mathcal{L}_{adv}(G, D, C_x) =& \mathbb{E}_{[y,C_x],x}\left[\log D([y, C_x], x)\right] \\ &+\mathbb{E}_{[y,C_x]}\left[\log(1 - D([y, C_x], G(y, C_x)))\right].\end{aligned} \tag{2.6}$$

Thus, the final adversarial loss is the sum of Equations (2.5) and (2.6),

$$\mathcal{L}_{adv}(G, D) = \mathcal{L}_{adv}(G, D, C_x) + \mathcal{L}_{adv}(G, D, C_y). \tag{2.7}$$

2.3.3 Optimization Objective

Color Loss. Previous work indicates that mixing the adversarial loss with a traditional loss such as $L1$ loss [55] or $L2$ loss [121] between the generated images and the ground truth images improves the generation performance. The definitions of $L1$ and $L2$ losses are:

$$\begin{aligned}\mathcal{L}_{L\{1,2\}}(G) =& \mathbb{E}_{[x,C_y],y}\left[\|y - G([x, C_y])\|_{\{1,2\}}\right], \\ &+\mathbb{E}_{[y,C_x],x}\left[\|x - G([y, C_x])\|_{\{1,2\}}\right].\end{aligned} \tag{2.8}$$

However, we observe that the existing image-to-image translation models such as PG^2 [98] cannot retain the holistic color of the input images. An example is shown in Figure 2.3, where PG^2 is affected by the pollution issue and produces more unrealistic regions. Therefore, to address this lim-

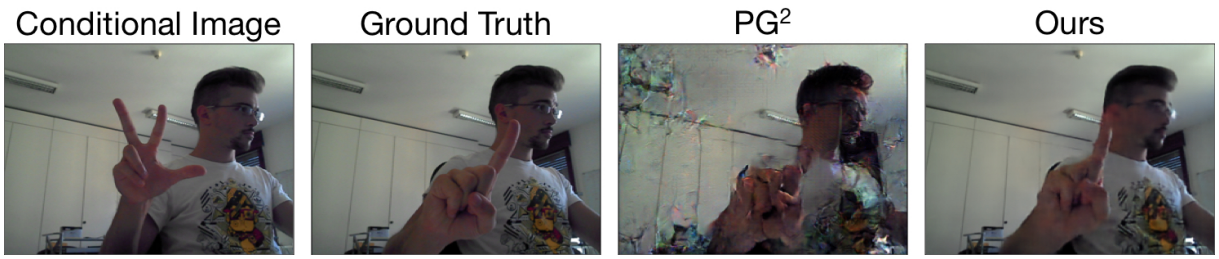


Figure 2.3: Illustration of the 'channel pollution' issue. From left to right: Conditional Image, Ground Truth, PG^2 [98], and Ours.

itation we introduce a novel channel-wise color loss. Traditional generative models convert an entire image into another, which leads to the 'channel pollution' problem. However, the color loss treats r, g, and b channels independently and generates only one channel each time, and then these three channels are combined to produce the final image. Intuitively, since the generation of a three-channel image space is much more complex than the generation of single-channel image space, leading to a higher possibility of artifacts, we independently generate each channel. The objective of r, g and b channel losses can be defined as follows,

$$\begin{aligned}\mathcal{L}_{color_c\{1,2\}}(G) =& \mathbb{E}_{[x_c,C_y],y_c}\left[\|y_c - G([x_c, C_y])\|_{\{1,2\}}\right] \\ &+\mathbb{E}_{[y_c,C_x],x_c}\left[\|x_c - G([y_c, C_x])\|_{\{1,2\}}\right].\end{aligned} \tag{2.9}$$

where $c \in \{r, g, b\}$; x_r, x_g and x_b denote the r, g and b channels of image x respectively similar to y_r, y_g and y_b; $\|\cdot\|_1$ and $\|\cdot\|_2$ represent $L1$ and $L2$ distance losses. Thus, the color $L1$ and $L2$ losses can be expressed as,

$$\mathcal{L}_{color\{1,2\}}(G) = \mathcal{L}_{Color_r\{1,2\}} + \mathcal{L}_{Color_g\{1,2\}} + \mathcal{L}_{Color_b\{1,2\}}. \tag{2.10}$$

In Equation (2.8), one channel is always influenced by the errors from other channels. On the contrary, if we compute the loss for each channel independently as shown in Equation (2.10), we can avoid such influence. In this way, the error in one channel will not influence other channels. We observe that this novel loss can improve the image quality in our experimental

section.

Controllable Structure Guided Self-Content Preserving Loss. To preserve the image content information (e.g., color composition, object identity, global layout) between the input and output, CycleGAN [231] proposes the identity preserving loss. However, different from [231], we propose the controllable structure guided self-content preserving loss, which can be expressed as follows,

$$\mathcal{L}_{con}(G) = \mathbb{E}_{x,C_x}\left[||x - G(x, C_x)||_1\right] + \mathbb{E}_{y,C_y}\left[||y - G(y, C_y)||_1\right]. \quad (2.11)$$

We aim to minimize the $L1$ difference between the real image x/y and the self-content preserving image $G(x, C_x)/G(y, C_y)$ for content information preservation. In this way, we regularize the generator to be near a self-content mapping when real images and self controllable structures are provided as the input to the generator.

Perceptual Loss measures the perceptual similarity in a high-level feature space. This loss has been shown to be useful for many tasks such as style transfer [60] and image translation [184]. The formulation of this loss is as follows:

$$\mathcal{L}_{vgg}(y') = \frac{1}{W_{i,j}H_{i,j}} \sum_{w=1}^{W_{i,j}} \sum_{h=1}^{H_{i,j}} ||\mathcal{F}^k(y) - \mathcal{F}^k(G(x, C_y))||_1, \quad (2.12)$$

where $\mathcal{F}^k$ indicates the feature map obtained by the k-th convolution within the VGG network [148], $W_{i,j}$ and $H_{i,j}$ are the dimensions of the respective feature maps within the VGG network. Similarly, we have another loss for the generated image x', which can be formulated as,

$$\mathcal{L}_{vgg}(x') = \frac{1}{W_{i,j}H_{i,j}} \sum_{w=1}^{W_{i,j}} \sum_{h=1}^{H_{i,j}} ||\mathcal{F}^k(x) - \mathcal{F}^k(G(y, C_x))||_1. \quad (2.13)$$

Thus, the final perceptual loss is the sum of both, $\mathcal{L}_{vgg}=\mathcal{L}_{vgg}(y')+\mathcal{L}_{vgg}(x')$.

Total Variation Loss. Usually, the images synthesized by GAN models have many unfavorable artifacts, which deteriorate the visualization and recognition performance. We impose the Total Variation (TV) loss [60] on the final synthesized image y' to alleviate this issue,

$$\mathcal{L}_{tv}(y') = \sum_{c=1}^{C} \sum_{w,h=1}^{W,H} |y'(w+1,h,c) - y'(w,h,c)| + |y'(w,h+1,c) - y'(w,h,c)|\,, \tag{2.14}$$

where W and H represent the width and height of the generated image y'. Similarly, we have another loss for the generated image x' and the final total variation loss is the sum of both.

Overall Loss. The total optimization loss is a weighted sum of the above losses. Generator G and discriminator D are trained in an end-to-end fashion to optimize the following min-max function,

$$\begin{aligned} G^* = \arg\min_G \max_D (&\mathcal{L}_{adv} + \lambda_{color}\mathcal{L}_{color} + \lambda_{cyc}\mathcal{L}_{cyc} + \\ &\lambda_{con}\mathcal{L}_{con} + \lambda_{vgg}\mathcal{L}_{vgg} + \lambda_{tv}\mathcal{L}_{tv}), \end{aligned} \tag{2.15}$$

where λ_{color}, λ_{cyc}, λ_{con}, λ_{vgg} and λ_{tv} are five hyper-parameters controlling the relative importance of these six losses. Solving this min-max problem enables our model to generate the target images guided by controllable structures in a photo-realistic manner.

2.3.4 Implementation Details

Network Architecture. We adopt our generator architecture G from [60], which has shown effective in many applications such as unsupervised image translation [231] and neural style transfer [60]. We use nine residual blocks for both 64×64 and 256×256 image resolutions. The last layer of the generator is the Tanh activation function. For the discriminator D,

we adopt 70×70 PatchGAN proposed in [55]. PatchGAN tries to decide if any 70×70 patch in an image is real or fake. The final layer of discriminators employs the Sigmoid activation function to produce a 1-dimensional output. Therefore, we are averaging all responses to provide the ultimate output of the discriminator.

Optimization Details. We observe that the proposed controllable structure guided discriminator achieves promising generation results. However, to further improve the image quality, we use the scheme of training a dual-discriminator instead of one discriminator as a more stable way to improve the capacity of discriminators similar to Nguyen et al. [110], which have demonstrated that they improve the ability of discriminator to generate more photo-realistic images. To be more specific, dual-discriminator architecture can better approximate optimal discriminator. If one of the discriminators is trained to be far superior over the generators, the generators can still receive instructive gradients from the other one. In addition to the proposed controllable structure guided discriminator, we use a traditional one, which takes the input image and the generated image as input. Both discriminators have the same network architecture structure.

We follow the standard optimization method in [41, 55] to optimize the proposed GAN model, i.e., one gradient descent step on discriminators and generator alternately. We first train generator G with discriminators fixed, and then train discriminators with generator G fixed. In addition, as suggested in [41], we train to maximize $\log D([x, C_y], y')$ rather than $\log(1-D([x, C_y], y'))$. Moreover, in order to slow down the rate of D relative to G we divide the objective function by 2 while optimizing D. The proposed GAN model is trained and optimized in an end-to-end fashion. We employ the Adam [68] optimizer with momentum terms β_1=0.5 and β_2=0.999 as our solver. The initial learning rate for Adam is 0.0002.

We follow [98] and exploit OpenPose [147] to detect the ground-truth

hand skeletons as training data for the hand gesture-to-gesture translation task. We then connect the 21 keypoints (hand joints) detected by OpenPose to obtain the hand skeleton. The hand skeleton image visually contains richer hand structure information than the hand keypoint image. In hand skeleton images, the hand joints are connected by the lines with a width of 4 and with white color. In addition, we follow [133] and use RefineNet [90] to generate the ground-truth semantic maps as training data for the cross-view image translation task.

2.3.5 Fréchet ResNet Distance

We also propose a novel evaluation metric to measure the image quality of the generated images by GAN models, i.e., Fréchet ResNet Distance (FRD). FRD provides an alternative method to quantify the quality of synthesis and is similar to the FID proposed by [48]. FID is a measure of similarity between two datasets of images. The authors have shown that the FID is more robust to noise than IS and correlates well with the human judgment of visual quality [48]. To calculate FID between two image domains y and y', they first embed both into a feature space F given by an Inception model. Then viewing the feature space as a continuous multivariate Gaussian as suggested in [48], Fréchet distance between the two Gaussians is used to quantify the quality of the data. The definition of FID is:

$$\mathrm{FID}(y, y') = \|\mu_y - \mu_{y'}\|_2^2 + \mathrm{Tr}(\textstyle\sum_y + \sum_{y'} - 2(\sum_y \sum_{y'})^{\frac{1}{2}}), \tag{2.16}$$

where $(\mu_y, \sum_y)$ and $(\mu_{y'}, \sum_{y'})$ are the mean and the co-variance of the data distribution and model distribution, respectively. Note that we regard the images in y' and y as two wholes respectively, and then calculate the Fréchet distance between y' and y for calculating FID.

Unlike FID, which calculates the distance between two distributions, the proposed FRD is inspired by the feature matching method [158], and separately calculates the Fréchet distance between each generated image and the corresponding real image from a semantic point of view. In this way, images from the two domains do not affect each other when computing the Fréchet distance. Moreover, for FID the number of samples should be greater than the dimension of the coding layer, while the proposed FRD does not have this limitation. We denote y_i and y'_i as images in the y and y' domains, respectively. For calculating FRD, we first embed both images y_i and y'_i into a feature space F with 1,000 dimensions given by a ResNet50 pretrained model. We then calculate the Fréchet distance between two feature maps $f(y_i)$ and $f(y'_i)$. The Fréchet distance $F(f(y_i), f(y'_i))$ is defined as the infimum over all reparameterizations α and β of $[0, 1]$ of the maximum over all $t \in [0, 1]$ of the distance in F between $f(y_i)(\alpha(t))$ and $f(y'_i)(\beta(t))$, where α and β are continuous, non-decreasing surjections of the range $[0, 1]$. The proposed FRD is a measure of similarity between the feature vector of the real image $f(y_i)$ and the feature vector of the generated image $f(y'_i)$ by calculating the Fréchet distance between them. The Fréchet distance is defined as the minimum cord-length sufficient to join a point traveling forward along $f(y'_i)$ and one traveling forward along $f(y_i)$, although the rate of travel for each point may not necessarily be uniform. Thus, the definition of FRD between two image domains y and y' is:

$$\mathrm{FRD}(y, y') = \frac{1}{N} \sum_{1}^{N} \inf_{\alpha,\beta} \max_{t\in[0,1]} \left\{ d\big(f(y_i)(\alpha(t)), f(y'_i)(\beta(t))\big) \right\}, \tag{2.17}$$

where d is the distance function of F and N is the total number of images in y and y' domains.

2.4 Experiments

To explore the generality of the proposed GAN model, we evaluate the proposed model on a variety of tasks and datasets, including hand gesture-to-gesture translation and cross-view image translation.

2.4.1 Hand Gesture-to-Gesture Translation

Datasets. We follow GestureGAN [163] and evaluate the proposed GAN model on two hand gesture datasets, i.e., NTU Hand Digit [137] and Creative Senz3D [101], which include different hand gestures. We use the hand gesture images and filter out failure cases in hand estimation for both training and testing sets. (1) NTU Hand Digit [137] contains 10 hand gestures (e.g., decimal digits from 0 to 9) color images and depth maps collected with a Kinect sensor under cluttered backgrounds. We randomly select 84,636 pairs, each of which is comprised of two images of the same person but different gestures. 9,600 pairs are randomly selected for the testing subset and the rest of 75,036 pairs as the training set. (2) Creative Senz3D [101] includes static hand gestures performed by 4 people, each performing 11 different gestures repeated 30 times each in the front of a Creative Senz3D camera. We randomly select 12,800 pairs and 135,504 pairs as the testing and training set, each pair is composed of two images of the same person but different gestures.

Parameter Settings. For both datasets, we do left-right flip and random crops for data augmentation. For optimization, models are trained with a batch size of 4 for 20 epochs on both datasets. At inference time, we follow the same settings of PG^2 [98] to randomly select the target keypoint or skeleton.

Evaluation Metrics. Following GestureGAN [163], we use Peak Signal-to-Noise Ratio (PSNR), Inception Score (IS), Fréchet Inception Distance

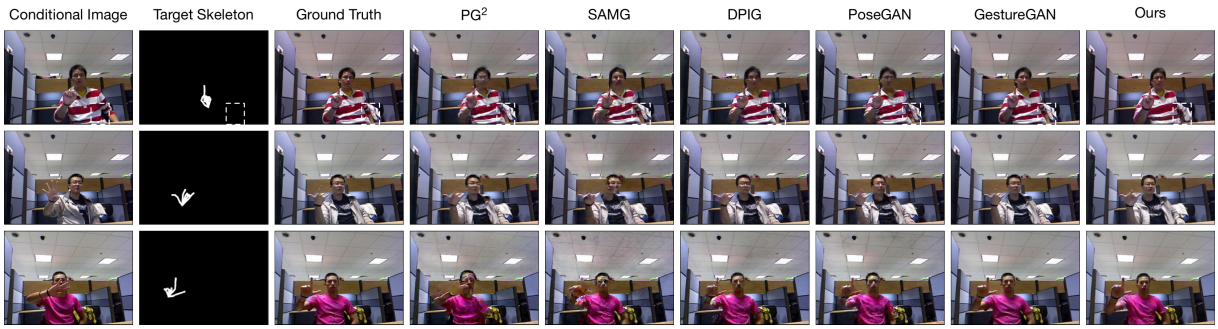


Figure 2.4: Different methods for hand gesture-to-gesture translation on NTU Hand Digit.

(FID), and the proposed FRD to evaluate the quality of generated images.

State-of-the-Art Comparisons. We compare the proposed model with the most related works, i.e., PG^2 [98], SAMG [201], PoseGAN [146], DPIG [99] and GestureGAN [163]. PG^2 and DPIG try to generate a person image with different poses based on conditional person images and target keypoints. SAMG and PoseGAN explicitly employ human skeleton information to generate person images. Note that SAMG adopts a CGAN to generate motion sequences based on appearance information and skeleton information by exploiting frame-level smoothness. We re-implemented this model to generate a single frame for a fair comparison. These methods are paired image-to-image models and comparison results are shown in Figures 2.4 and 2.5. As we can see in both figures, the proposed model consistently produces sharper images with convincing details compared with other baselines on both datasets. We also note that the proposed GAN model is more robust than existing methods as shown in the first row of Figure 2.4. Existing methods are easy to overfit since they generate the dropping arm as shown in the white dotted box while the proposed model failed to generate it. It is hard to generate the dropping arm since no guidance has been provided to generate it, while exiting methods just simply memorize the blocks from training images to generate new ones rather than to learn the representations between different images.

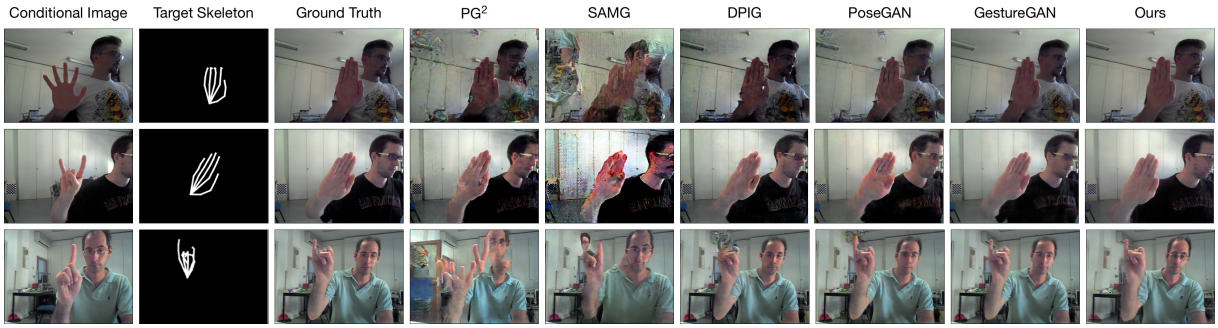


Figure 2.5: Different methods for hand gesture-to-gesture translation on Senz3D.

Method	NTU Hand Digit					Senz3D				
	PSNR ↑	IS ↑	AMT ↑	FID ↓	FRD ↓	PSNR ↑	IS ↑	AMT ↑	FID ↓	FRD ↓
PG² [98]	28.2403*	2.4152*	3.5%*	24.2093*	2.6319*	26.5138*	3.3699*	2.8%*	31.7333*	3.0933*
SAMG [201]	28.0185*	2.4919*	2.6%*	31.2841*	2.7453*	26.9545*	3.3285*	2.3%*	38.1758*	3.1006*
DPIG [99]	30.6487*	2.4547*	7.1%*	6.7661*	2.6184*	26.9451*	3.3874*	6.9%*	26.2713*	3.0846*
PoseGAN [146]	29.5471*	2.4017*	9.3%*	9.6725*	2.5846*	27.3014*	3.2147*	8.6%*	24.6712*	3.0467*
GestureGAN [163]	32.6091*	**2.5532***	26.1%*	7.5860*	2.5223*	27.9749*	**3.4107***	22.6%*	18.4595*	2.9836*
Ours	**32.6574**	2.3783	**29.3%**	**6.7493**	**1.7401**	**31.5420**	2.2159	**27.6%**	**12.4465**	**2.2104**

Table 2.1: Comparison results with state-of-the-art models for hand gesture-to-gesture translation on NTU Hand Digit and Senz3D. For all metrics except FID and FRD, higher is better. (∗) These results are reported in [163].

Moreover, we also provide quantitative results in Table 2.1, and we can see that the proposed GAN model produces more photo-realistic results that other baselines on all metrics expect IS. This phenomenon can also be observed in PG² [98], GestureGAN [163], and other super-resolution work such as [60], i.e., sharper results have a lower IS. Finally, we also show some results of the arbitrary hand gesture-to-gesture translation on NTU Hand Digit dataset in Figure 2.8. Given a single image and several hand skeletons, the proposed model can generate the corresponding hand gestures.

User Study. We also conducted a user study similar to [98, 231, 146]. We follow the same settings as in [55] to perform an Amazon Mechanical Turk (AMT) perceptual study and gather data from 50 participants per algorithm we tested. Specifically, participants were presented a sequence of pairs of images, a 'real' image and a 'fake' image (generated by our algo-

rithm or a baseline), and asked to click on the image they thought was real. The first 10 images of each session were practice and feedback was given. The remaining 40 images were used to assess the rate at which each algorithm fooled participants. Each session only tested a single algorithm, and participants were only allowed to complete a single session. The results on NTU Hand Digit and Senz3D datasets compared with the baseline models are shown in Table 2.1. We observe that the proposed model consistently achieves the best performance compared with these baselines.

FID v.s. FRD. We also compare the performance between FID and the proposed FRD metric. The results are shown in Table 2.1 and we can observe that FRD is more consistent with the human judgment, i.e., the AMT score, than the FID metric. Moreover, we observe that the difference in FRD between GestureGAN and the other methods is not as obvious as in the results from the user study, i.e., the AMT metric. The reason is that FRD calculates the Fréchet distance between the feature maps extracted from the real image and the generated image using CNNs which are trained with semantic labels. Thus, these feature maps are employed to reflect the semantic distance between the images. The semantic distance between the images is not very large considering they are all hands. On the contrary, the user study measures the generation quality from a perceptual level. The difference on the perceptual level is more obvious than on the semantic level, i.e., the generated images with small artifacts show minor differences on the feature level, while are being judged with a significant difference from the real images by humans.

2.4.2 Cross-View Image Translation

Datasets. We follow [133] and conduct the experiments on two public datasets: 1) For Dayton [179], following the same setting of [133], we select 76,048 images and create a train/test split of 55,000/21,048 pairs.

The images in the original dataset have 354×354 resolution. We resize them to 256×256. 2) CVUSA [192] consists of 35,532/8,884 image pairs in train/test split. Following [133], the aerial images are center-cropped to 224×224 and resized to 256×256. For the ground-level images and corresponding segmentation maps, we take the first quarter of both and resize them to 256×256.

Parameter Settings. We follow [133] and all images are scaled to 256×256, and we enabled random crops for data augmentation. The low-resolution experiments on Dayton are carried out for 100 epochs with a batch size of 16, whereas the high-resolution experiments for this dataset are trained for 35 epochs with a batch size of 4. For CVUSA, we follow the same setup as in [133] and train our network for 30 epochs with a batch size of 4.

Evaluation Metrics. We follow [133] and use Inception Score (IS), top-k prediction accuracy, KL score, Structural-Similarity (SSIM), PSNR, and Sharpness Difference (SD) for the quantitative analysis. Moreover, we employ LPIPS [220] to evaluate the quality of the generated images. LPIPS uses pretrained deep models to evaluate the similarity, which highly agrees well with humans' perception. Specifically, we use the default pretrained AlexNet provided by the authors [220] to calculate the LPIPS metric.

State-of-the-Art Comparison. We compare the proposed model with five recently proposed state-of-the-art methods on the cross-view image translation task, i.e., Pix2pix [55], Zhai et al. [211], X-Fork [133], X-Seq [133] and SelectionGAN [165]. The comparison results are shown in The qualitative results in higher resolution on Dayton and CVUSA datasets are shown in Figures 2.6 and 2.7. The proposed model generates better results against other baselines in terms of detail preservation and translation visual effects. In addition, it can be seen that our method generates more clear details on objects/scenes such as road, trees, clouds than SelectionGAN in the generated ground-level images (zoom-in for details in

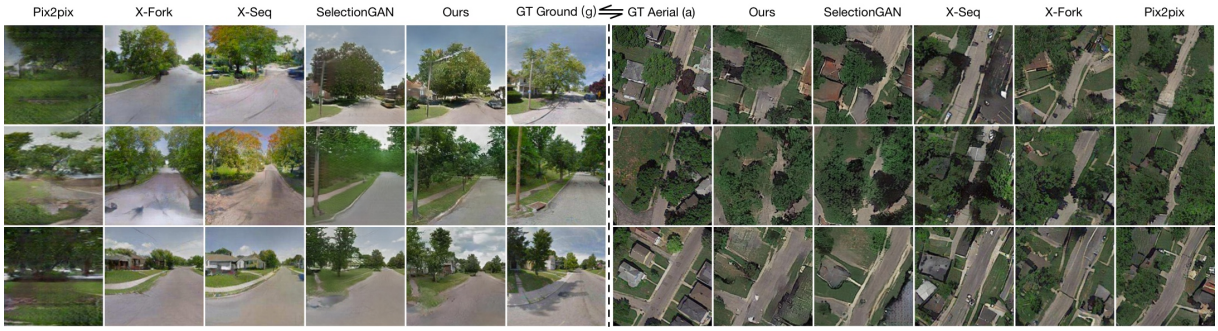


Figure 2.6: Different methods for cross-view image translation task in 256×256 resolution on Dayton.

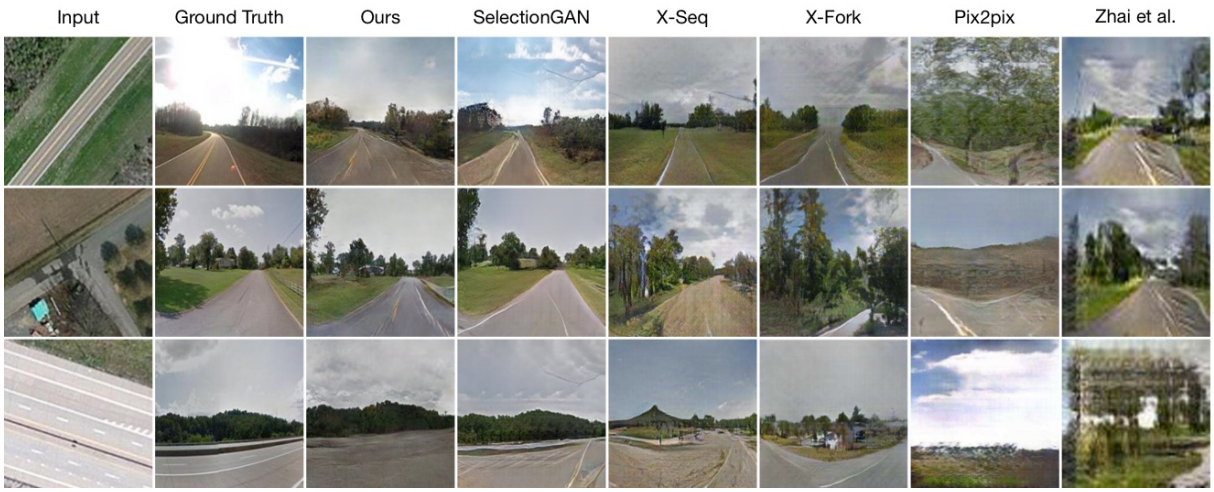


Figure 2.7: Different methods for cross-view image translation task in 256×256 resolution on CVUSA.

Figure 2.6). For the generated aerial images, we can observe that grass, trees, and house roofs are well-rendered compared to others. Moreover, the results generated by our method are closer to the ground truth in layout and structure (see the results in the a2g direction in Figures 2.6 and 2.7).

The quantitative comparison results are shown in Tables 2.2, 2.3, 2.4 and 2.5. We can observe the significant improvement of the proposed model in these tables. The proposed model consistently outperforms Pix2pix, Zhai et al., X-Fork, and X-Seq on all the metrics. Moreover, comparing against SelectionGAN, the proposed model still achieves competitive performance on all metrics excepting SSIM, PSNR, and SD. In most cases of

Dir. ⇆	Method	Accuracy (%) ↑				Inception Score ↑			SSIM ↑	PSNR ↑	SD ↑	KL ↓
		Top-1		Top-5		all	Top-1	Top-5				
a2g	Pix2pix [55]	7.90*	15.33*	27.61*	39.07*	1.8029*	1.5014*	1.9300*	0.4808*	19.4919*	16.4489*	6.29 ± 0.80*
	X-Fork [133]	16.63*	34.73*	46.35*	70.01*	1.9600*	1.5908*	2.0348*	0.4921*	19.6273*	16.4928*	3.42 ± 0.72*
	X-Seq [133]	4.83*	5.56*	19.55*	24.96*	1.8503*	1.4850*	1.9623*	0.5171*	20.1049*	16.6836*	6.22 ± 0.87*
	SelectionGAN [165]	45.37†	79.00†	83.48†	97.74†	**2.1606**†	1.7213†	**2.1323**†	**0.6865**†	**24.6143**†	**18.2374**†	1.70 ± 0.45†
	Ours	**49.86**	**84.41**	**86.14**	**99.61**	2.1059	**1.7342**	2.0737	0.6754	24.2814	18.1361	**1.54 ± 0.39**
	Real Data	-	-	-	-	2.3534†	1.8135†	2.3250†	-	-	-	-
g2a	Pix2pix [55]	1.65*	2.24*	7.49*	12.68*	1.7970*	1.3029*	1.6101*	0.3675*	20.5135*	14.7813*	6.39 ± 0.90*
	X-Fork [133]	4.00*	16.41*	15.42*	35.82*	1.8557*	1.3162*	1.6521*	0.3682*	20.6933*	14.7984*	4.45 ± 0.84*
	X-Seq [133]	1.55*	2.99*	6.27*	8.96*	1.7854*	1.3189*	1.6219*	0.3663*	20.4239*	14.7657*	7.20 ± 0.92*
	SelectionGAN [165]	14.12†	**51.81**†	39.45†	74.70†	**2.1571**†	**1.4441**†	**2.0828**†	**0.5118**†	23.2657†	16.2894†	2.25 ± 0.56†
	Ours	**16.65**	44.83	**44.03**	**77.01**	2.0802	1.4360	2.0628	0.5064	**23.3632**	**16.4788**	**2.16 ± 0.59**
	Real Data	-	-	-	-	2.3015†	1.5056†	2.2095†	-	-	-	-

Table 2.2: Quantitative evaluation of Dayton in 64×64 resolution. For all metrics except KL score, higher is better. (∗, †) These results are reported in [133] and [165], respectively.

Dir. ⇆	Method	Accuracy (%) ↑				Inception Score ↑			SSIM ↑	PSNR ↑	SD ↑	KL ↓
		Top-1		Top-5		all	Top-1	Top-5				
a2g	Pix2pix [55]	6.80*	9.15*	23.55*	27.00*	2.8515*	1.9342*	2.9083*	0.4180*	17.6291*	19.2821*	38.26 ± 1.88*
	X-Fork [133]	30.00*	48.68*	61.57*	78.84*	3.0720*	2.2402*	3.0932*	0.4963*	19.8928*	19.4533*	6.00 ± 1.28*
	X-Seq [133]	30.16*	49.85*	62.59*	80.70*	2.7384*	2.1304*	2.7674*	0.5031*	20.2803*	19.5258*	5.93 ± 1.32*
	SelectionGAN [165]	42.11†	68.12†	77.74†	92.89†	3.0613†	2.2707†	3.1336†	**0.5938**†	**23.8874**†	**20.0174**†	2.74 ± 0.86†
	Ours	**49.12**	**80.43**	**81.20**	**94.87**	**3.3210**	**2.3494**	**3.3522**	0.5633	23.3515	19.7692	**2.17 ± 0.77**
	Real Data	-	-	-	-	3.8319†	2.5753†	3.9222†	-	-	-	-
g2a	Pix2pix [55]	10.23*	16.02*	30.90*	40.49*	3.5676*	2.0325*	2.8141*	0.2693*	20.2177*	16.9477*	7.88 ± 1.24*
	X-Fork [133]	10.54*	15.29*	30.76*	37.32*	3.1342*	1.8656*	2.5599*	0.2763*	20.5978*	16.9962*	6.92 ± 1.15*
	X-Seq [133]	12.30*	19.62*	35.95*	45.94*	**3.5849***	2.0489*	2.8414*	0.2725*	20.2925*	16.9285*	7.07 ± 1.19*
	SelectionGAN [165]	**20.66**†	**33.70**†	**51.01**†	**63.03**†	3.2446†	**2.1331**†	**3.4091**†	0.3284†	21.8066†	17.3817†	**3.55 ± 0.87**†
	Ours	17.31	29.40	43.58	55.27	3.2131	2.0916	3.3637	**0.3357**	**22.0273**	**17.6542**	5.17 ± 1.23
	Real Data	-	-	-	-	3.7196†	2.3626†	3.8998†	-	-	-	-

Table 2.3: Quantitative evaluation of Dayton in 256×256 resolution. For all metrics except KL score, higher is better. (∗, †) These results are reported in [133] and [165], respectively.

the a2g direction in Tables 2.2 and 2.3 we achieve a slightly lower performance as compared with SelectionGAN. However, we consistently achieve better performance than SelectionGAN on the LPIPS metric as shown in Table 2.5, which agrees more with human judgments as indicated in [220]. We also report both FID and FRD results compared with the most related SelectionGAN in Tables 2.6 and 2.7. We can see that the proposed method achieves better results than SelectionGAN in most cases. Finally, we also note that SelectionGAN is carefully designed for the cross-view image translation task while the proposed model is a generic framework.

Arbitrary Cross-View Image Translation. Existing methods such as

Method	Accuracy (%) ↑				Inception Score ↑			SSIM ↑	PSNR ↑	SD ↑	KL ↓
	Top-1		Top-5		all	Top-1	Top-5				
Zhai et al. [211]	13.97*	14.03*	42.09*	52.29*	1.8434*	1.5171*	1.8666*	0.4147*	17.4886*	16.6184*	27.43 ± 1.63*
Pix2pix [55]	7.33*	9.25*	25.81*	32.67*	3.2771*	2.2219*	3.4312*	0.3923*	17.6578*	18.5239*	59.81 ± 2.12*
X-Fork [133]	20.58*	31.24*	50.51*	63.66*	3.4432*	2.5447*	3.5567*	0.4356*	19.0509*	18.6706*	11.71 ± 1.55*
X-Seq [133]	15.98*	24.14*	42.91*	54.41*	3.8151*	2.6738*	4.0077*	0.4231*	18.8067*	18.4378*	15.52 ± 1.73*
SelectionGAN [165]	41.52†	65.51†	74.32†	89.66†	3.8074†	2.7181†	3.9197†	0.5323†	**23.1466**†	19.6100†	2.96 ± 0.97†
Ours	**45.06**	**70.04**	**78.31**	**93.47**	**3.9469**	**2.8779**	**4.0383**	**0.5366**	22.8223	**19.8276**	**2.60 ± 0.97**
Real Data	-	-	-	-	4.8741†	3.2959†	4.9943†	-	-	-	-

Table 2.4: Quantitative evaluation of CVUSA in the a2g direction. For all metrics except KL score, higher is better. (∗, †) These results are reported in [133] and [165], respectively.

Dir.	Method	Dayton (64×64)	Dayton (256×256)	CVUSA
a2g	SelectionGAN [165]	0.1786	0.4996	0.4652
	Ours	**0.1712**	**0.3529**	**0.3817**
g2a	SelectionGAN [165]	0.2489	0.5264	-
	Ours	**0.2382**	**0.4527**	-

Table 2.5: LPIPS of [165] and the proposed method for cross-view image translation. For this metric, lower is better.

Zhai et al. [211], Pix2pix [55], X-Fork [133] and X-Seq [133] focus on the cross-view image translation task. However, this task is essentially an ill-posed problem and has limited scalability and robustness in handling more than two viewpoints. A recent work SelectionGAN [165] extends the cross-view image translation task to a more generic task of the problem, i.e., the arbitrary cross-view image translation, in which a single input view can be translated to different target views. For the arbitrary cross-view image translation, conditional labels are usually required since learning a one-to-many mapping is more challenging and extremely hard to optimize. Similarly to the arbitrary hand gesture-to-gesture translation in Figure 2.8, we show several results of arbitrary cross-view image translation on Dayton in Figure 2.9. We believe this task has many applications such as cross-view image geo-localization.

Network Parameter Comparisons. We compare the overall network parameter with Pix2pix [55], X-Fork [133], X-Seq [133] and SelectionGAN [165] on cross-view image translation task. Results are shown in

Dir.	Method	Dayton (64×64)	Dayton (256×256)	CVUSA
a2g	SelectionGAN [165]	28.4787	38.3498	**43.1102**
	Ours	**18.7225**	**35.9220**	47.3500
g2a	SelectionGAN [165]	60.7903	**85.4072**	-
	Ours	**60.1969**	88.8195	-

Table 2.6: FID of [165] and the proposed method for cross-view image translation. For this metric, lower is better.

Dir.	Method	Dayton (64×64)	Dayton (256×256)	CVUSA
a2g	SelectionGAN [165]	3.3066	3.5060	3.1641
	Ours	**3.1658**	**3.3694**	**3.1547**
g2a	SelectionGAN [165]	3.8033	**3.7646**	-
	Ours	**3.7078**	3.8943	-

Table 2.7: FRD of [165] and the proposed method for cross-view image translation. For this metric, lower is better.

Table 2.8. As we can see, the proposed model achieves superior model capacity and produces better generation performance comparing with existing models.

Model	Pix2pix [55]	X-Fork [133]	X-Seq [133]	SelectionGAN [165]	Ours
G	39.0820 M	39.2163 M	39.0820*2 M	55.4808 M	11.3876 M
D	2.7696 M	2.7696 M	2.7696*2 M	2.7687 M	2.7678+2.7709 M
Total	41.8516 M	41.9859 M	83.7032 M	58.2495 M	**16.9263 M**

Table 2.8: Comparison of the number of network parameters on cross-view image translation task.

2.4.3 Ablation Study

We perform an ablation study in the a2g (aerial-to-ground) direction on Dayton for cross-view image translation. Following [165], to reduce the training time, we randomly select 1/3 samples from the whole 55,000/21,048 samples, i.e., around 18,334 samples for training and 7,017 samples for testing.

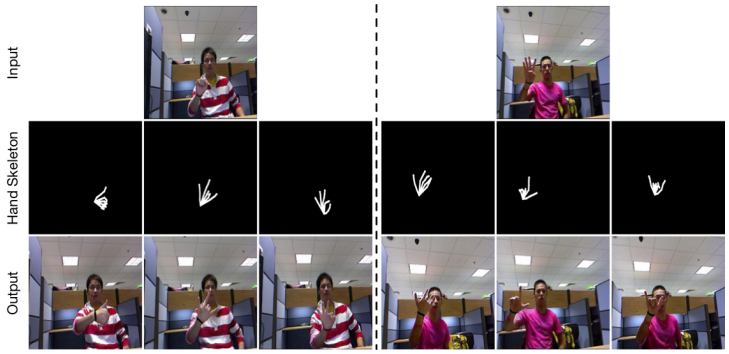


Figure 2.8: Arbitrary hand gesture-to-gesture translation of our model.

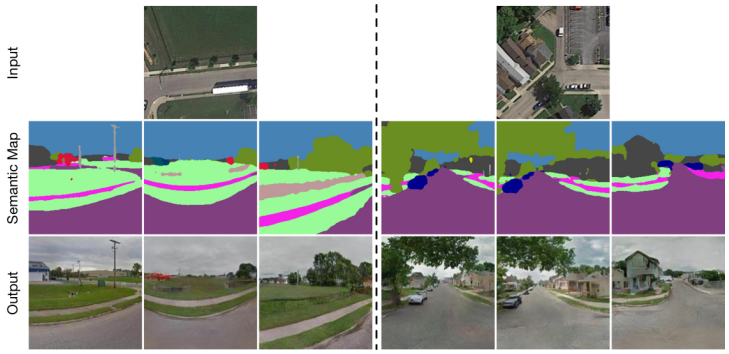


Figure 2.9: Arbitrary cross-view image translation of our model.

Baseline Models. The proposed GAN model has nine baselines (A, B, C, D, E1x, E2x, E3, E4x, F) as shown in Table 2.9. Baseline A uses a CycleGAN model [231] and generates y' using an unpaired image x. Baseline B uses a Pix2pix structure [55], and generates y' based on x using a supervised way. Baseline C also uses the Pix2pix structure and inputs the combination of a conditional image x and the controllable structure C_y to the proposed controllable structure guided generator G. Baseline D uses the proposed controllable structure guided cycle upon baseline C. Baseline E1x explores the proposed color loss in several different ways to avoid the 'channel pollution' issue. Baseline E2x employs the proposed controllable structure guided discriminator to stabilize the optimization process. Baseline E3 adds the proposed controllable structure guided self-content preserving loss to preserve content information. Baseline E4x adds the perceptual loss and the Total Variation loss on the generated result

Baseline	Experimental Setting	SSIM ↑	PSNR ↑	SD ↑	Accuracy ↑				KL ↓	LPIPS ↓
					Top-1		Top-5			
A	$x \xrightarrow{G} y'$ (Unsupervised Learning)	0.4110	17.9868	18.5195	27.28	47.47	52.47	71.63	8.69±1.36	0.5913
B	$x \xrightarrow{G} y'$ (Supervised Learning)	0.4555	19.6574	18.8870	27.46	46.84	58.20	77.17	6.25±1.30	0.5520
C	$[x, C_x] \xrightarrow{G} y'$ (Controllable Structure Guided Generation)	0.5374	22.8345	19.2075	39.76	68.44	72.22	89.85	3.32±1.10	0.4010
D	$[x, C_y] \xrightarrow{G} [y', C_x] \xrightarrow{G} x'$ (Controllable Structure Guided Cycle)	0.5547	23.1531	19.6032	42.43	70.82	75.40	91.16	2.89±1.05	0.3821
E11	D + Color L1 Loss on x'	0.5515	23.1345	19.6257	41.08	68.31	75.26	90.60	3.02±1.09	0.3968
E12	D + L1 Loss on y'	0.5541	23.1492	19.6423	41.73	68.99	75.13	89.48	2.89±1.02	0.3835
E13	D + L2 Loss on y'	0.5481	23.0939	19.5534	43.51	72.08	75.79	91.23	2.86±0.99	0.3913
E14	D + Color L1 Loss on y'	0.5600	23.3692	19.7018	44.38	73.21	75.93	91.69	2.73±0.98	0.3782
E15	D + Color L2 Loss on y' + L1 loss on y'	0.5568	23.3930	19.6273	43.19	72.58	75.63	91.67	2.77±1.10	0.3793
E16	D + Color L1 Loss on y' + L1 loss on y'	**0.5631**	**23.4600**	**19.7650**	44.97	73.65	76.28	92.32	2.70±1.08	0.3765
E21	D + Controllable Structure Guided Discriminator	0.5340	22.8176	19.4404	43.08	72.80	74.98	90.89	3.06±1.09	0.4003
E22	D + Dual Discriminator	0.5255	22.5405	19.4104	43.12	74.85	76.14	91.23	2.93±1.02	0.3937
E3	D + Controllable Structure Guided Self-Content Preserving Loss	0.5473	23.0475	19.5561	42.81	70.18	76.71	91.32	2.76±0.99	0.3877
E41	D + Perceptual Loss	0.5494	23.1075	19.5197	45.34	75.40	78.09	93.24	2.87±0.79	0.3545
E42	D + Perceptual Loss + Total Variation Loss	0.5577	23.0242	19.4943	44.76	73.96	77.81	93.69	2.84±0.79	**0.3543**
F	D + E16 + E22 + E3 + E42	0.5603	23.1626	19.7455	**46.43**	**76.94**	**79.54**	**94.33**	**2.35±0.84**	0.3571

Table 2.9: Ablation study of the proposed method on Dayton for cross-view image translation. For all evaluation metrics except KL and LPIPS, higher is better.

y'. Baseline F is our full model integrating baselines D, E16, E22, E3, and E42. All the baseline models are trained and tested on the same data using the same configuration.

Note that each baseline in E (i.e., E1x, E2x, E3, and E4x) focuses on improving each aspect of the performance of the generated images. More specifically, the proposed color loss aims to avoid the 'channel pollution' issue and thus improve the pixel-level similarity metrics, i.e., SSIM, PSNR, and SD. The proposed controllable structure guided discriminator tries to improve the structure accuracy since the controllable structure can provide strong supervision to the discriminator. The proposed controllable structure guided self-content preserving loss can push the generated data distribution close to the real data distribution. Finally, the perceptual loss and the Total Variation loss aim to improve image fidelity.

Ablation Analysis. The results of the ablation study are shown in Table 2.9. We observe that Baseline B is better than baseline A since the ground truth image y can provide strong supervised information to the generator G. Comparing Baseline B with Baseline C, the controllable structure guided generation improves the performance on all metrics by large margins, which confirms that the controllable structures can provide

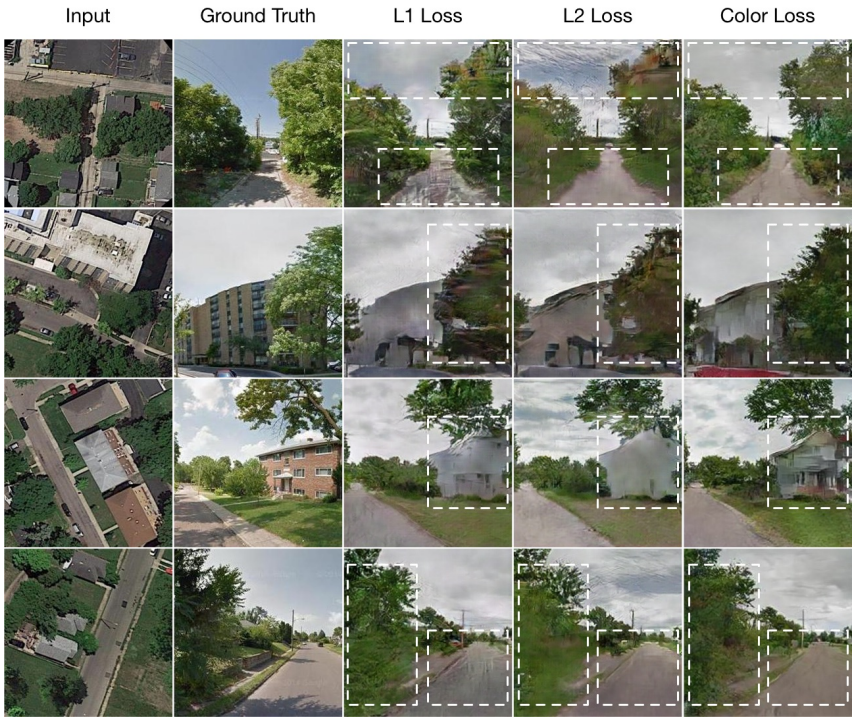


Figure 2.10: Comparison results of $L1$ Loss, $L2$ Loss and the proposed Color Loss for cross-view image translation.

more structural information to the generator G. By using the proposed controllable structure guided cycle, Baseline D further improves over baseline C, meaning that the cycle structure indeed utilizes the controllable structure information in a more effective way, confirming our design motivation. Baseline E14 outperforms baselines D, E12, and E13 on SSIM, PSNR, and SD metrics showing the importance of using the proposed color loss to avoid the 'channel pollution' issue. Visualization results of $L1$ loss, $L2$ loss and the proposed color $L1$ loss are shown in Figure 2.10. We can see that the proposed color $L1$ loss generates more clear and visually plausible details than both $L1$ and $L2$ losses, which validates the effectiveness of the proposed color loss. By further combining the color $L1$ loss and the $L1$ loss on the generated image y', we can further improve the performance as shown in baseline E16. However, replacing the color $L1$ loss with the color $L2$ loss will degrade the performance as shown by baseline E15 but

the results are still better than using baseline D. We also use the proposed color loss on the reconstructed image x' as presented in baseline E11, but it achieves the worst results. Comparing Baseline D with Baseline E21, the proposed controllable structure guide discriminator improves the top-1 accuracy by 0.65 and 1.98, which confirms the importance of our design motivation. By further combining the controllable structure guide discriminator with the traditional discriminator in baseline E22, both top-1 and top-5 accuracies are further boosted. Baseline E3 outperforms D with around 0.13 gains on the KL metric, clearly demonstrating the effectiveness of the proposed controllable structure guided self-content preserving loss. By adding the perceptual loss and the TV loss in baseline E4x, the overall performance is further improved on LPIPS metric [220], which uses pretrained deep models to evaluate the similarity and highly agrees with human perception. Finally, we demonstrate the advantage of the proposed full model in baseline F, which integrates baseline D, E14, E22, E3, and E42. It is obvious that baseline F achieves the best results on both accuracy and KL score metrics. However, we observe that baseline F achieves worse results on SSIM, PSNR, and SD compared with baseline E14, and at the same time, it achieves worse results on the LPIPS metric compared with baseline E42. This is also observed in LPIPS [220], i.e., the traditional metrics (i.e., SSIM, PSNR, SD, FSIM) disagree with metrics based on deep architectures such as VGG [148]. Thus, we try to balance both metrics to reasonable results without dropping significantly the performance, and we still observe that baseline F achieves better performance on all SSIM, PSNR, SD, and LPIPS metrics than baseline D.

Hyper-parameter Analysis. (1) For cross-view image translation tasks, we follow [55] and set λ_{color}=100 since $\mathcal{L}_{color}$ denotes a pixel-wise reconstruction loss. We then follow [165] and set λ_{tv}=1e−6. In addition to λ_{color} and λ_{tv}, we also introduce λ_{cyc}, λ_{con} and λ_{vgg}. Thus, we investigate the influence

λ_{cyc}	SSIM ↑	PSNR ↑	SD ↑	Inception Score ↑			Accuracy ↑				KL ↓	LPIPS ↓
				all	Top-1	Top-5	Top-1		Top-5			
100	0.5383	23.0283	19.5731	2.9278	1.9960	2.9823	39.22	67.86	69.55	88.03	3.96 ± 1.32	0.4082
10	0.5475	23.1264	19.5590	3.2344	2.2321	3.2983	42.30	67.99	74.98	89.54	**2.87 ± 1.01**	0.3832
1	0.5478	23.1153	19.5158	3.1918	2.2025	3.2362	42.11	**72.26**	75.37	**91.33**	2.88 ± 1.02	0.3869
0.1	**0.5547**	**23.1731**	**19.6032**	**3.2823**	**2.2401**	**3.3081**	**42.43**	70.82	**75.40**	91.16	2.89 ± 1.05	**0.3821**

Table 2.10: The influence of λ_{cyc} on Dayton for cross-view image translation.

λ_{con}	0.1	1	5	10	100
KL ↓	2.85±1.03	2.93±1.00	2.83±1.01	3.00±1.02	**2.76±0.99**

Table 2.11: The influence of λ_{con} on Dayton for cross-view image translation.

λ_{vgg}	1	10	20	50	100
LPIPS ↓	0.3812	0.3708	0.3628	0.3571	**0.3545**

Table 2.12: The influence of λ_{vgg} on Dayton for cross-view image translation.

λ_{cyc}	PSNR ↑	IS ↑	FID ↓	FRD ↓
0.001	28.5673	**2.4851**	23.9935	2.8468
0.01	28.5475	2.3719	23.8958	2.8991
0.1	28.4967	2.4755	**21.6280**	**2.7571**
1	28.5370	2.3436	23.5811	2.8467
10	28.5627	2.4815	22.5539	2.8401
100	**28.5854**	2.4191	23.5617	2.8080

Table 2.13: The influence of λ_{cyc} on NTU Hand Digit for hand gesture-to-gesture translation.

λ_{con}	PSNR ↑	IS ↑	FID ↓	FRD ↓
0.001	28.5638	2.4335	20.6123	2.7273
0.01	28.4607	2.3665	**19.9356**	**2.6960**
0.1	**28.6696**	2.3446	23.2919	2.8326
1	28.6478	2.3522	24.4331	2.9171
10	28.6642	2.3528	21.7138	2.8778
100	28.5207	**2.4881**	24.3938	2.9104

Table 2.14: The influence of λ_{con} on NTU Hand Digit for hand gesture-to-gesture translation.

of λ_{cyc}, λ_{con}, λ_{vgg} to the performance of our model. The results are shown in Tables 2.10, 2.11 and 2.12. In Table 2.10, when λ_{cyc} becomes smaller, we achieve better results on most metrics. This means that adjusting the

λ_{vgg}	PSNR ↑	IS ↑	FID ↓	FRD ↓
0.001	28.4537	2.3096	23.7465	2.6976
0.01	28.5580	**2.4825**	23.4135	2.6966
0.1	28.5741	2.4684	23.1802	2.6872
1	28.5625	2.3182	20.1516	2.6653
10	28.5486	2.2502	19.8930	2.6004
100	**28.9545**	2.0455	17.1370	2.4461
1000	28.8131	2.0965	**14.1617**	**2.2135**
10000	27.4805	2.4538	65.1080	3.2607

Table 2.15: The influence of λ_{vgg} on NTU Hand Digit for hand gesture-to-gesture translation.

λ_{color}	PSNR ↑	IS ↑	FID ↓	FRD ↓
100	28.8131	2.0965	14.1617	2.2135
200	29.2343	2.1537	**13.4811**	2.2421
500	29.9973	2.1332	13.4823	**2.2039**
800	30.4531	2.1898	13.9475	2.2176
1000	30.7087	**2.2138**	15.3634	2.2134
2000	31.4232	2.1991	17.1864	2.2872
5000	**32.3025**	2.1022	28.5587	2.3715

Table 2.16: The influence of λ_{color} on NTU Hand Digit for hand gesture-to-gesture translation.

ratio of weighting parameters of the cycle can obtain further performance improvement. This is different from CycleGAN [231], which uses the same weights for both forward and backward cycle-consistency losses since CycleGAN tries to learning two mappings, while in our model we only focus on generating photo-realistic y' and do not care about the quality of the reconstructed image x''. Thus, the forward part has a lager weight than the backward part. Moreover, we also investigate the influence of λ_{con} and λ_{vgg}. The results are listed in Tables 2.11 and 2.12. When both λ_{con} and λ_{vgg} become bigger, the generator with a larger error loss dominates the training, making the whole model generating better results. Therefore, we empirically set λ_{cyc}=0.1, λ_{con}=100, λ_{vgg}=100, λ_{color}=100 and λ_{tv}=1e−6 in Equation (2.15) for this task.

# Cycle	PSNR ↑	IS ↑	FID ↓	FRD ↓
One-Cycle	30.4531	**2.1898**	13.9475	2.2176
Two-Cycle	**31.4924**	2.1493	**11.2084**	**2.0774**

Table 2.17: The influence of the number of cycles on NTU Hand Digit for hand gesture-to-gesture translation.

(2) For hand gesture-to-gesture translation tasks, we first follow [165] and set λ_{tv}=1e−6. Next, we investigate the influence of λ_{cyc}, λ_{con}, λ_{vgg} and λ_{color} to the performance of our model. Results are shown in Tables 2.13, 2.14, 2.15 and 2.16. According to these tables, we empirically set λ_{cyc}=0.1, λ_{con}=0.01, λ_{vgg}=1000, λ_{color}=800 and λ_{tv}=1e−6 in Equation (2.15) for this task. Moreover, we also investigate the influence of the number of cycles on this task. Results are shown in Table 2.17 and we observe that the two-cycle framework achieves better results than one-cycle framework on most metrics.

2.5 Conclusion

In this paper, we focus on the challenging task of controllable image-to-image translation. To this end, we propose a unified GAN framework, which can generate target images with different poses, sizes, structures, and locations based on a conditional image and controllable structures. In this way, the conditional image can provide appearance information and the controllable structures can provide structure information for generating the final results. Moreover, we also propose three novel losses to learn the mapping from the source domain to the target domain, i.e., color loss, controllable structure guided cycle-consistency loss, and controllable structure guided self-content preserving loss. It is worth noting that the proposed color loss handles the 'channel pollution' problem when back-propagating the gradients, which frequently occurs in the existing generative models.

The controllable structure guided cycle-consistency loss can reduce the dismatch between the source domain and the target domain. The controllable structure guided self-content preserving loss aims to preserve the image content information of generated images. In addition, we present a novel Fréchet ResNet Distance (FRD) metric to evaluate the quality of generated images. Experimental results show that the proposed unified GAN framework achieves competitive performance compared with the state of the art using carefully designed frameworks on two challenging generative tasks, i.e., hand gesture-to-gesture translation and cross-view image translation. Note that the proposed GAN framework is not tuned to any specific controllable image-to-image translation tasks.

In next chapter, we will introduce C2GAN, a novel and unified cross-modal generative adversarial network for guided image-to-image translation tasks, which organizes the guidance and the image data in an interactive manner, instead of using as input only the guidance information.

Chapter 3

C2GAN

We propose a novel and unified Cycle In Cycle Generative Adversarial Network (C2GAN) for generating human faces, hands, bodies, and natural scenes. Our proposed C2GAN is a cross-modal model exploring a joint exploitation of the input image data and the guidance data in an interactive manner. C2GAN contains two different generators, i.e., an image-generation generator and a guidance-generation generator. Both generators are mutually connected and trained in an end-to-end fashion and explicitly form three cycled sub-nets, i.e., one image generation cycle and two guidance generation cycles. Each cycle aims at reconstructing the input domain, and simultaneously produces useful output involved in the generation of another cycle. In this way, the cycles constrain each other implicitly providing complementary information from both image and guidance modalities and bringing extra supervision gradient across cycles facilitating a more robust optimization of the whole model. Extensive experimental results on four guided image-to-image translation sub-tasks, i.e., person image generation, facial expression generation, hand gesture-to-gesture translation and cross-view image translation, demonstrate that the proposed C2GAN is effective in generating more realistic images compared with state-of-the-art models. The source code and trained models

are available at https://github.com/HaoTang/C2GAN.

3.1 Introduction

Recent works have developed powerful image-to-image translation systems, e.g., Pix2pix [55], Pix2pixHD [184] and GauGAN [119] in supervised settings, and CycleGAN [231] and DualGAN [206] in unsupervised settings. However, these methods are proposed and tailored to merely two domains at a time, and scaling them to more would require a quadratic number of models to be trained. For instance, with m different image domains, CycleGAN and Pix2pix need to train $m(m-1)/2$ and $m(m-1)$ models, respectively. To overcome this, Choi et al. propose StarGAN [25], in which a single generator/discriminator can perform image-to-image translation for multiple domains. However, StarGAN is not effective in handling some specific image-to-image translation tasks such as human pose generation [98, 146], hand gesture generation [163], and cross-view image translation [133], in which image generation could involve infinite image domains since human body, hand gesture, and natural scene in the wild can have arbitrary poses, sizes, appearances, locations, and viewpoints.

To address these limitations, many methods are proposed to generate images based on extra semantic guidance such as object keypoints [132, 129, 149, 98], human skeletons [146, 163], or segmentation maps [133, 134, 165, 119, 161]. For instance, Song et al. [149] propose a G2GAN framework for facial expression synthesis based on facial landmarks. Siarohin et al. [146] introduce a PoseGAN model for pose-based human image generation conditioned on human body skeletons. Regmi and Borji [133] propose both X-Fork and X-Seq for cross-view image translation conditioned on segmentation maps. However, current state-of-the-art guided image-to-image translation methods such as PG2 [98], PoseGAN [146], X-

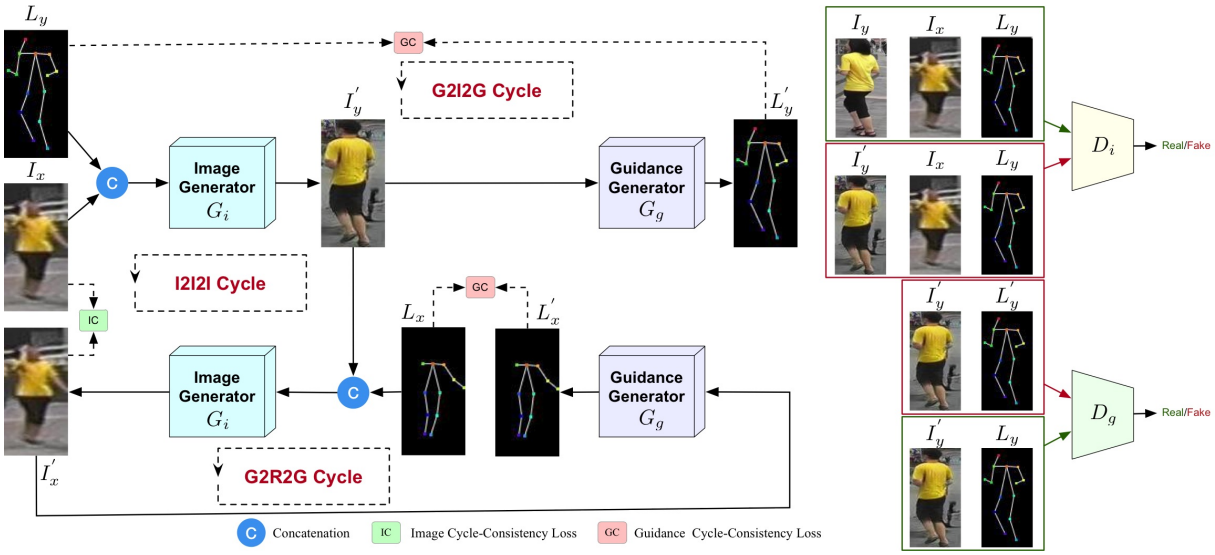


Figure 3.1: Framework overview of the proposed C2GAN, which consists of two types of generators, i.e., image generator G_i and guidance generator G_g. Parameter-sharing strategies can be used in between the image or the guidance generators to reduce the model capacity. During the training stage, two generators G_i and G_g are explicitly connected and trained by three cycles, i.e., the image cycle I2I2I: $[I_x, L_y] \overset{G_i}{\rightarrow} [I'_y, L_x] \overset{G_i}{\rightarrow} I'_x$ and two guidance cycles G2I2G: $[I_x, L_y] \overset{G_i}{\rightarrow} I'_y \overset{G_g}{\rightarrow} L'_y$, G2R2G: $[I'_y, L_x] \overset{G_i}{\rightarrow} I'_x \overset{G_g}{\rightarrow} L'_x$. The right side of the figure shows the cross-modal discriminators (i.e., D_i and D_g) for a better network optimization.

Fork [133], and X-Seq [133] have two main issues: (1) they directly transfer an image from the source domain to the target domain, without considering the mutual translation between each other, while the translation across different image and guidance modalities in a unified framework would bring rich cross-modal information; (2) they simply employ the guidance data as input to guide the generation process, without involving the generated guidance as supervisory signals to further improve the network optimization. Both issues lead to unsatisfactory results.

To fix both issues, we propose a novel and unified Cycle In Cycle Generative Adversarial Network (C2GAN), in which explicitly three cycled sub-nets are formed to learn both image and guidance modalities in a joint model. The framework of the proposed C2GAN is shown in Figure 3.1. Specifically, C2GAN contains a basic image cycle, i.e., I2I2I

($[I_x, L_y] \overset{G_i}{\rightarrow} [I_y^{'}, L_x] \overset{G_i}{\rightarrow} I_x^{'}$), which aims at reconstructing the input and further refines the generated images $I_y^{'}$. The guidance information (such as the human body skeleton) in C2GAN is not only utilized as input but also acts as output, meaning that the guidance is also a generative objective. The input and output of the guidance are connected by two guidance cycles, i.e., G2I2G ($[I_x, L_y] \overset{G_i}{\rightarrow} I_y^{'} \overset{G_g}{\rightarrow} L_y^{'}$) and G2R2G ($[I_y^{'}, L_x] \overset{G_i}{\rightarrow} I_x^{'} \overset{G_g}{\rightarrow} L_x^{'}$), where G_i and G_g denote an image and a guidance generator, respectively. In this way, guidance cycles can provide weak supervision to the generated images $I_y^{'}$. The intuition behind the guidance cycles is that if the generated guidance is very close to the real guidance, then the corresponding images should be similar (see Figure 3.2). In other words, a better guidance generation will boost the performance of image generation, and conversely the improved image generation will further facilitate the guidance generation. The proposed three cycles inherently constraint each other in an end-to-end training fashion. Moreover, for a better optimization of the proposed three cycles we further propose two novel cycle losses, i.e., Image Cycle-consistency loss (IC) and Guidance Cycle-consistency loss (GC). With both cycle losses, each cycle can benefit from each other in a joint learning way. We also propose two cross-modal discriminators corresponding to the generators.

Our contributions can be summarized as follows:

- We propose C2GAN, a novel and unified cross-modal generative adversarial network for guided image-to-image translation tasks, which organizes the guidance and the image data in an interactive manner, instead of using as input only the guidance information.
- The proposed cycle in cycle network structure is a new design which explores the effective use of cross-modal information for guided image-to-image translation tasks. The designed cycled sub-networks connect different modalities and implicitly constraint each other, leading to

extra supervision signals for better image generation. We also investigate cross-modal discriminators and cycle losses for a more robust network optimization.

- Extensive results on four challenging guided image-to-image translation tasks, i.e., person image generation, facial expression generation, hand gesture-to-gesture translation, and cross-view image translation demonstrate the effectiveness of the proposed C2GAN and show more photo-realistic images compared with state-of-the-art models.

3.2 Related Work

Image-to-Image Translation models use input-output data to learn a mapping between the source domain and the target domain. Isola et al. propose Pix2pix [55], which employs a CGAN to learn a image mapping from the input domain to the output domain. Moreover, unpaired image-to-image translation approaches [231, 152, 167, 206, 166, 67, 228, 5, 160] have been proposed to learn the mapping without paired training data. However, these existing image translation models are inefficient and ineffective as indicated in the introduction section. Most importantly, these aforementioned approaches cannot handle some specific guided image-to-image translation tasks such as person image generation [98, 146], and hand gesture-to-gesture translation [163].

Guided Image-to-Image Translation. To address these aforementioned limitations, several works have been proposed to generate images based on object keypoints [71, 28, 98, 186, 151], human hand/body skeleton [146, 201, 14], and scene segmentation map [134, 134, 165, 119, 169]. For instance, Wang et al. [186] propose a Conditional MultiMode Network (CMM-Net) for the facial landmark guided smile generation. Tang et al. [163] propose a novel GestureGAN to perform the hand gesture-to-gesture

translation task conditioned on an input image and several novel hand skeletons. Park et al. [119] propose a novel spatially-adaptive normalization for the semantic image synthesis task based on semantic labels. These methods only focus on a single image generation task.

However, we propose a multi-task framework aiming at handling two tasks using a single network, i.e., image generation and guidance generation. During the training stage, the two generation tasks are restricted mutually by the proposed three cycles and then benefit from each other. To the best of our knowledge, the proposed C2GAN is the first attempt to generate both the image and the guidance domain in an interactive generation strategy within a unified cycle in cycle GAN model, for guided image-to-image translation tasks.

3.3 Model Description

We first introduce the network structures of the three cycles and describe the details for the corresponding generators and cross-modal discriminators. Then we present the proposed objective functions and implementation details.

3.3.1 Model Overview

The proposed C2GAN learns two different generators in a single network, i.e., image generator and guidance generator. The two generators are mutually connected through three novel generative adversarial cycles, i.e., one image-oriented cycle and two guidance-oriented cycles. In the training stage, these three cycles are jointly optimized in an end-to-end way and each generator can benefit from the others due to the richer cross-modal information and the crossing cycle supervision. The core framework of our C2GAN is illustrated in Figure 3.1.

3.3.2 Image-Domain Generative Adversarial Cycle

I2I2I Cycle. The image cycle I2I2I aims to generate the image $I_y^{'}$ by using the combination of the input image I_x and the target guidance L_y, and then reconstruct the input image I_x by using the combination of the generated image $I_y^{'}$ and the guidance L_x of image I_x. Thus, the image cycle can be formulated as:

$$[I_x, L_y] \xrightarrow{G_i} [I_y^{'}, L_x] \xrightarrow{G_i} I_x^{'}, \tag{3.1}$$

where G_i is the image generator.

Different from the previous guided image-to-image translation methods such as PG2 [98], X-Fork [133], and PoseGAN [146] employing only one mapping $[I_x, L_y] \xrightarrow{G_i} I_y^{'}$, StarGAN [25] employs the target and the original domain labels l_y and l_x as extra guidance information to reconstruct the input image. However, StarGAN can only handle tasks which have a fixed number of the image categories. In order to solve this limitation, we replace the domain labels l_y and l_x in StarGAN by using the guidances L_y and L_x. The guidances can be object keypoints, human skeletons, or scene segmentation maps. Specifically, I_x and L_y are first fed into the image generator G_i to generate the desired image $I_y^{'}$. Next, the generated image $I_y^{'}$ and the guidance L_x are concatenated as the input of G_i to reconstruct the original image I_x. In this way, the forward and backward consistency can be guaranteed.

Image Generator. The U-Net architecture [138] is adopted for our image generator G_i. U-Net consists of an encoder and a decoder with skip connections between them. The generator G_i is used two times for generating image $I_y^{'}$ and reconstructing the original image $I_x^{'}$. To reduce the model capacity, the image generator G_i shares parameters between image generation and reconstruction. For image generation, the target of G_i is generating an

image $I_y^{'}$=$G_i(I_x, L_y)$ conditioned on the target guidance L_y which is similar to the real image I_y. For image reconstruction, the goal of G_i is recovering an image $I_x^{'}$=$G_i(I_y^{'}, L_x)$ that looks similar to the input image I_x. The image generator G_i learns a combined data distribution between the image generation and the image reconstruction by sharing parameters, meaning that G_i receives double data during the network optimization compared to those generators without using the parameter-sharing strategy.

Cross-Modal Image Discriminator. Different from previous works such as PG2 [98] employing a single-modal discriminator, we propose a novel cross-modal discriminator which receives both image and guidance data as input. The framework of the proposed cross-modal discriminator is shown in Figure 3.1. The image discriminator D_i receives two images and one guidance data as input. More specifically, D_i aims to distinguish between the generated triplet $[I_x, L_y, G_i(I_x, L_y)]$ and the real triplet $[I_x, L_y, I_y]$ during image generation stage.

We further propose an image adversarial loss $\mathcal{L}_{GAN}^{i}(G_i, D_i, I_x, I_y, L_y)$ based on the vanilla adversarial loss [41], which can be expressed as follows:

$$\begin{aligned}\mathcal{L}_{GAN}^{i}(G_i, D_i, I_x, I_y, L_y) =& \mathbb{E}_{I_x, L_y, I_y \sim p_{\text{data}}(I_x, L_y, I_y)} [\log D_i([I_x, L_y, I_y])] \\ &+ \mathbb{E}_{I_x, L_y \sim p_{\text{data}}(I_x, L_y)} [\log(1 - D_i([I_x, L_y, G_i(I_x, L_y)]))],\end{aligned} \tag{3.2}$$

where the image generator G_i tries to minimize the image adversarial loss $\mathcal{L}_{GAN}^{i}(G_i, D_i, I_x, I_y, L_y)$ while the image discriminator D_i tries to maximize it.

Another image adversarial loss for the image reconstruction mapping $G_i : [I_y^{'}, L_x] \rightarrow I_x^{'}$ is defined as:

$$\begin{aligned}\mathcal{L}_{GAN}^{i}(G_i, D_i, I_x, I_y, L_x) =& \mathbb{E}_{I_x, L_x, I_y \sim p_{\text{data}}(I_x, L_x, I_y)} [\log D_i([I_y, L_x, I_x])] \\ &+ \mathbb{E}_{I_y', L_x, I_y \sim p_{\text{data}}(I_y', L_x, I_y)} [\log(1 - D_i([I_y, L_x, G_i(I_y^{'}, L_x)]))],\end{aligned} \tag{3.3}$$

where the image discriminator D_i aims at distinguishing between the fake triplet $[I_y, L_x, G_i(I_y^{'}, L_x)]$ and the real triplet $[I_y, L_x, I_x]$.

Therefore, the overall image adversarial loss is the sum of Equations (3.2) and (3.3):

$$\begin{aligned}\mathcal{L}^i_{GAN}(G_i, D_i, I_x, I_y, L_x, L_y) =& \mathcal{L}^i_{GAN}(G_i, D_i, I_x, I_y, L_y) \\ &+\mathcal{L}^i_{GAN}(G_i, D_i, I_x, I_y, L_x).\end{aligned} \tag{3.4}$$

Image Cycle-Consistency (IC) Loss. We also propose the IC loss to better learn the image cycle I2I2I, which can be formulated as:

$$\mathcal{L}^i_{CYC}(G_i, I_x, L_x, L_y) = \mathbb{E}_{I_x, L_x, L_y \sim p_{\text{data}}(I_x, L_x, L_y)}[\|G_i(G_i(I_x, L_y), L_x) - I_x\|_1]. \tag{3.5}$$

The reconstructed images $I_x^{'}$=$G_i(G_i(I_x, L_y), L_x)$ should closely match with the input image I_x. Notably, the image generator G_i is used two times with the parameter-sharing strategy and the $L1$ distance is adopted in Equation (3.5) to compute a pixel-to-pixel difference between the recovered image $I_x^{'}$ and the real input image I_x.

3.3.3 Guidance-Domain Generative Adversarial Cycle

The motivation of the guidance cycle is that, if the generated guidance is similar to the real guidance then the corresponding two images should be very close (see Figure 3.2). The proposed C2GAN has two guidance cycles, i.e., G2I2G and G2R2G, as shown in Figure 3.1. Both cycles can provide extra supervision information for better optimizing the image cycle I2I2I.

G2I2G Cycle. For the G2I2G cycle, $[I_x, L_y]$ is first fed into the image generator G_i to produce the target image $I_y^{'}$. Then the guidance generator G_g tries to produce the guidance $L_y^{'}$ from the generated image $I_y^{'}$. The generated guidance $L_y^{'}$ should be very close to the real guidance L_y. The

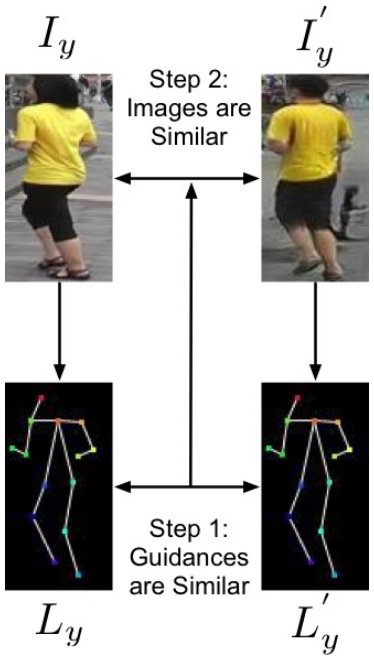


Figure 3.2: The motivation of the guidance cycle. If the generated guidance $L_y^{'}$ is close to the real guidance L_y, then the corresponding images (i.e., $I_y^{'}$ and I_y) should be similar.

formulation of the G2I2G cycle can be expressed as:

$$[I_x, L_y] \overset{G_i}{\rightarrow} I_y^{'} \overset{G_g}{\rightarrow} L_y^{'}. \tag{3.6}$$

G2R2G Cycle. For the G2R2G cycle, the generated image $I_y^{'}$ and guidance L_x are first concatenated, and then fed into G_i to produce the recovered image $I_x^{'}$. Next, the guidance generator G_g generates the guidance $L_x^{'}$ from the recovered image $I_x^{'}$. We assume that the generated guidance $L_x^{'}$ is very similar to the real guidance L_x. The G2R2G cycle can be formulated as:

$$[I_y^{'}, L_x] \overset{G_i}{\rightarrow} I_x^{'} \overset{G_g}{\rightarrow} L_x^{'}. \tag{3.7}$$

Both generated guidances $L_y^{'}$=$G_g(G_i(I_x, L_y))$ and $L_x^{'}$=$G_g(G_i(I_y^{'}, L_x))$ should have a close match to the real guidance L_y and L_x, respectively. Note that the guidance generator G_g could share parameters between these two cycles.

Guidance Generator. The U-Net structure [138] is employed for our guidance generator G_g. The input of G_g is an image and the output is a guidance. The guidance generator respectively produces $L_y^{'}$=$G_g(I_y^{'})$ and $L_x^{'}$=$G_g(I_x^{'})$ from the generated images $I_y^{'}$ and $I_x^{'}$, which further provide more supervision gradient to guide the image generator G_i to produce more realistic images.

Cross-Modal Guidance Discriminator. As shown in Figure 3.1, the proposed guidance discriminator D_g is a cross-modal discriminator receiving both image and guidance data as the inputs. Thus, the guidance adversarial loss for D_g can be defined as:

$$\begin{aligned}\mathcal{L}_{GAN}^{g}(G_g, D_g, I_y^{'}, L_y) =& \mathbb{E}_{I_y^{'}, L_y \sim p_{\text{data}}(I_y^{'}, L_y)} \Big[\log D_g([I_y^{'}, L_y])\Big] \\ &+ \mathbb{E}_{I_y^{'} \sim p_{\text{data}}(I_y^{'})}[\log(1 - D_g([I_y^{'}, G_g(I_y^{'})]))],\end{aligned} \tag{3.8}$$

where the guidance generator G_g aims to minimize the guidance adversarial loss $\mathcal{L}_{GAN}^{g}(G_g, D_g, I_y^{'}, L_y)$ while the guidance discriminator D_g tries to maximize it. The discriminator D_g aims to distinguish between the fake pair $[I_y^{'}, L_y^{'}]$ and the real pair $[I_y^{'}, L_y]$.

A similar guidance adversarial loss for the mapping function $G_g : I_x^{'} \rightarrow L_x^{'}$ is defined as:

$$\begin{aligned}\mathcal{L}_{GAN}^{g}(G_g, D_g, I_x^{'}, L_x) =& \mathbb{E}_{I_x^{'}, L_x \sim p_{\text{data}}(I_x^{'}, L_x)} \Big[\log D_g([I_x^{'}, L_x])\Big] \\ &+ \mathbb{E}_{I_x^{'} \sim p_{\text{data}}(I_x^{'})}[\log(1 - D_g([I_x^{'}, G_g(I_x^{'})]))],\end{aligned} \tag{3.9}$$

where the guidance discriminator D_g aims to distinguish between the fake pair $[I_x^{'}, L_x^{'}]$ and the real pair $[I_x^{'}, L_x]$.

Thus, the total guidance adversarial loss is the sum of Equations (3.8)

and (3.9):

$$\mathcal{L}^{g}_{GAN}(G_g, D_g, I^{'}_{x}, I^{'}_{y}, L_x, L_y) = \mathcal{L}^{g}_{GAN}(G_g, D_g, I^{'}_{y}, L_y) + \mathcal{L}^{g}_{GAN}(G_g, D_g, I^{'}_{x}, L_x). \tag{3.10}$$

Guidance Cycle-Consistency (GC) Loss. A novel GC loss is further proposed to better learn both the guidance cycles (i.e., G2I2G and G2R2G), which can be expressed as:

$$\begin{aligned}\mathcal{L}^{g}_{CYC}(G_g, G_i, I_x, I^{'}_{y}, L_x, L_y) =& \mathbb{E}_{I_x, L_y \sim p_{\text{data}}(I_x, L_y)}[\|G_g(G_i(I_x, L_y)) - L_y\|_1] \\ &+ \mathbb{E}_{I^{'}_{y}, L_x \sim p_{\text{data}}(I^{'}_{y}, L_x)}[\|G_g(G_i(I^{'}_{y}, L_x)) - L_x\|_1],\end{aligned} \tag{3.11}$$

where the $L1$ distance is used to compute the pixel-to-pixel difference between the generated guidance (i.e., $L^{'}_{x}$ and $L^{'}_{y}$) and the corresponding real guidance (i.e., L_x and L_y).

During the training stage, the proposed guidance cycle-consistency loss can back-propagate errors from the guidance generator G_g to the image generator G_i facilitating the optimization of the image generator and then boosting the image generation performance.

3.3.4 Optimization Objective

We follow existing methods [146, 98, 133, 165] and use the image pixel loss to reduce the changes between the generated image $I^{'}_{y}$=$G_i(I_x, L_y)$ and the corresponding real one I_y. This loss can be expressed as:

$$\mathcal{L}^{i}_{PIXEL}(G_i, I_x, L_y, I_y) = \mathbb{E}_{I_x, L_y, I_y \sim p_{\text{data}}(I_x, L_y, I_y)}[\|G_i(I_x, L_y) - I_y\|_1], \tag{3.12}$$

where the $L1$ distance is adopted as the loss measurement in the image pixel loss. By doing so, more constrains can be added on both the image generator G_i.

Consequently, the complete objective loss of the proposed C2GAN frame-

work is:

$$\begin{aligned}\mathcal{L}(G_i, G_g, D_i, D_g) =& \lambda^i_{gan} * \mathcal{L}^i_{GAN} + \lambda^i_{cyc} * \mathcal{L}^i_{CYC} \\ &+ \lambda^i_{pixel} * \mathcal{L}^i_{PIXEL} + \lambda^g_{gan} * \mathcal{L}^g_{GAN} + \lambda^g_{cyc} * \mathcal{L}^g_{CYC},\end{aligned} \tag{3.13}$$

where λ^i_{gan}, λ^i_{cyc}, λ^i_{pixel}, λ^g_{gan}, and λ^g_{cyc} are parameters controlling the relative relation of objectives terms.

3.3.5 Implementation Details

Network Architecture. We adopt the U-Net architecture [138] consisting of an encoder and a decoder for our generators G_i and G_g. Moreover, we employ the PatchGAN discriminator [55] for our discriminators D_i and D_g, which has shown the effectiveness in previous image-to-image translation works [55, 231]. The difference between a PatchGAN and a regular GAN discriminator is that the regular GAN maps from an image to a single scalar output, 'real' or 'fake', whereas the PatchGAN tries to classify if each $N \times N$ patch in an image is real or fake. By doing so, PatchGAN can alleviate the generation of visual artifacts and achieves better performance. More details about PatchGAN can be found in [55].

Training Strategy. We follow the standard optimization method from [41] to optimize the proposed C2GAN, i.e., we alternate between one gradient descent step on G_i, D_i, G_g, and D_g, respectively. The proposed C2GAN is trained end-to-end and can generate image and guidance simultaneously, then the generated guidances will benefit the quality of the generated images. Moreover, to slow down the rate of discriminators D_i and D_g relative to generators G_i and G_g, we divide the objectives by 2 while optimizing discriminators.

The public software OpenFace [4] is employed to extract facial landmark on the Radboud Faces dataset for the facial expression generation task. While OpenPose [13] is used to extract human hand and body skeleton

on the Creative Senz3D and Market-1501 datasets for hand gesture-to-gesture translation and person image generation task, respectively. Next, we follow [133] and employ RefineNet [90] to extract segmentation maps on the Dayton dataset for the cross-view image translation task.

Inference Strategy. During the inference stage, the proposed G2GAN receives an image I_x and a guidance L_y into the image generator G_i, and outputs a target image $I_y^{'}$. At the same time, the guidance generator G_g receives the image I_x as input and outputs the corresponding guidance $L_x^{'}$.

Parameter Setting. For a fair comparison, all competing models are trained for 200 epochs on the Radboud Faces dataset for the facial expression generation task. All models are trained around 90 epochs on the person image generation task. For the hand gesture-to-gesture translation task, we train the model for 20 epochs. For the cross-view image translation task, we train the model for 35 epochs. The Adam solver [68] with the momentum terms β_1=0.5 and β_2=0.999 is adopted as our optimizer. We also incorporate the mask loss proposed in PG2 [98] for the person image generation task.

The parameters λ_{gan}^{i}, λ_{gan}^{g}, λ_{cyc}^{i}, λ_{pixel}^{i} and λ_{cyc}^{g} in Equation (3.13) are set to 1, 1, 10, 10 and 10, respectively. The proposed C2GAN is implemented using public deep learning software PyTorch [120].

3.4 Experiments

Extensive experiments are conducted on four guided image-to-image translation tasks, i.e., person image generation [98], facial expression generation [164], hand gesture-to-gesture translation [163], and cross-view image translation [133], to evaluate the effectiveness of the proposed C2GAN framework.

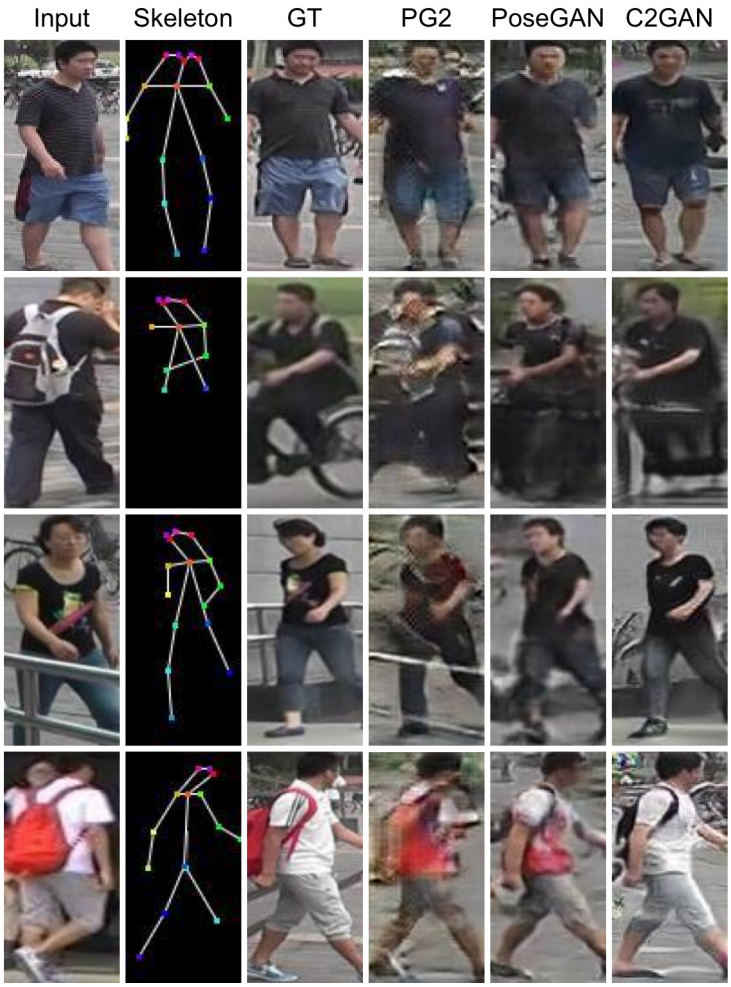


Figure 3.3: Qualitative comparison of person image generation on the Market-1501 dataset. From left to right: Input, Body Skeleton, Ground Truth (GT), PG2 [98], PoseGAN [146], and C2GAN (Ours).

3.4.1 Person Image Generation

Datasets. We follow [98] and employ the Market-1501 dataset [224] for the person image generation task. The Market-1501 dataset [224] is a challenging person re-id dataset containing 32,668 images of 1,501 persons collected from six surveillance cameras. We adopt the training and testing splits used in [146] and obtain 263,631 and 12,000 pairs for the training and testing subset, respectively.

Evaluation Metrics. We follow [146, 98] and adopt Inception Score

Model	AMT (R2G) ↑	AMT (G2R) ↑	SSIM ↑	IS ↑	Mask-SSIM ↑	Mask-IS ↑
PG2 [98]	11.2	5.5	0.253	3.460	0.792	3.435
DPIG [99]	-	-	0.099	**3.483**	0.614	3.491
PoseGAN [146]	22.7	**50.2**	**0.290**	3.185	0.805	3.502
Pix2pixSC [182]	18.6	41.5	0.275	3.141	0.790	3.468
CocosNet [219]	20.1	45.7	0.280	3.275	0.801	3.514
C2GAN (Ours)	**23.8**	47.3	0.285	3.362	**0.813**	**3.526**
Real Data	-	-	1.000	3.860	1.000	3.360

Table 3.1: Quantitative comparison of person image generation on the Market-1501 dataset. For all the metrics, higher is better.

(IS) [139], SSIM and their corresponding masked versions Mask-SSIM and Mask-IS as our evaluation metrics. Moreover, we adopt the AMT perceptual user study to evaluate the generated images by different models.

State-of-the-Art Comparisons. We compare the proposed C2GAN with several state-of-the-art person image generation methods, i.e., PG2 [98], DPIG [99], PoseGAN [146], Pix2pixSC [182], and CocosNet [219]. Different from these models which focus on the person image generation task, the proposed method is a general framework and learns image and guidance generation simultaneously in a joint network.

Quantitative results are shown in Table 3.1. We observe that the proposed C2GAN achieves better results than PG2, DPIG, Pix2pixSC, and CocosNet. Moreover, compared with PoseGAN [146], the proposed C2GAN also yields very competitive results. Specifically, the proposed C2GAN obtains better results in terms of most metrics, i.e., AMT (R2G), IS, mask-SSIM, and mask-IS. Qualitative comparison results compared with PG2 and PoseGAN are shown in Figure 3.3. We can see that the proposed C2GAN can generate more clear and visually plausible person images than both leading methods, validating the effectiveness of the proposed C2GAN. Moreover, we observe that the generated images by our method are more similar to the ground truths.

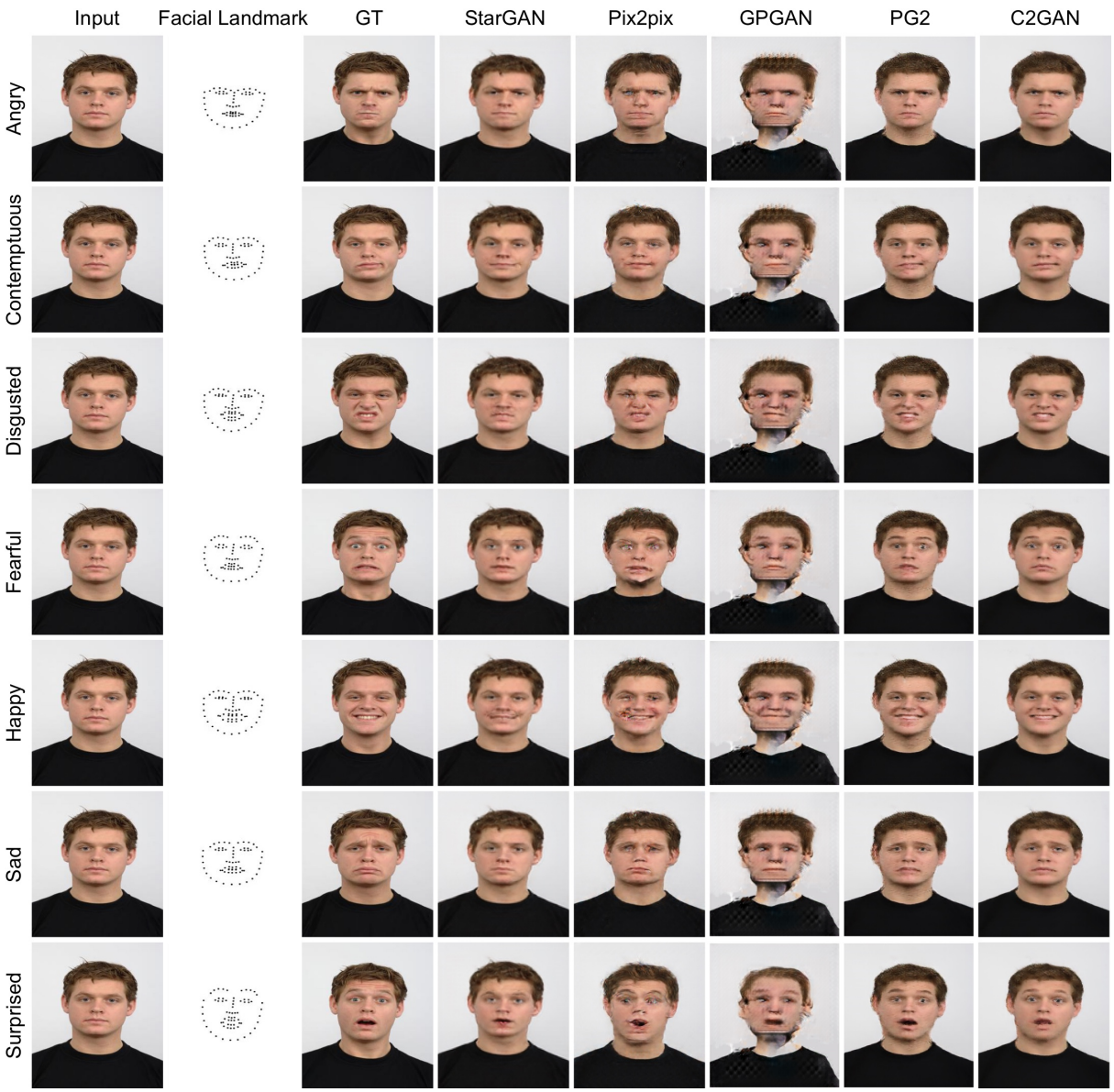


Figure 3.4: Qualitative comparison of facial expression generation on the Radboud Faces dataset. From left to right: Input, Facial Landmark, Ground Truth (GT), StarGAN [25], Pix2pix [55], GPGAN [28], PG2 [98], and C2GAN (Ours).

3.4.2 Facial Expression Generation

Datasets. We employ the Radboud Faces dataset [74] for the facial expression generation task. This dataset contains over 8,000 color face images with eight different facial expressions. We randomly select 67% of images for training and the rest 33% images for testing. We remove the images

Model	AMT ↑	SSIM ↑	PSNR ↑	LPIPS ↓
StarGAN [25]	24.7	0.8345	19.6451	-
Pix2pix [55]	13.4	0.8217	19.9971	0.1334
GPGAN [28]	0.3	0.8185	18.7211	0.2531
PG2 [98]	28.4	0.8462	20.1462	0.1130
Pix2pixSC [182]	30.8	0.8433	20.3584	0.1042
CocosNet [219]	31.3	0.8524	20.7915	0.0985
C2GAN (Ours)	**34.2**	**0.8618**	**21.9192**	**0.0934**

Table 3.2: Quantitative comparison of facial expression generation on the Radboud Faces dataset. For all the metrics except LPIPS, higher is better.

in which the face is not correctly detected by OpenFace [4], then combine two different facial expression images of the same person to form an image pair for training. Therefore, 5,628 and 1,407 image pairs are obtained for training and testing, respectively.

Evaluation Metrics. We first adopt the AMT user study to evaluate the generated images. Moreover, we employ Structural Similarity (SSIM) [191], Peak Signal-to-Noise Ratio (PSNR), and LPIPS [220] for quantitative measurements. SSIM and PSNR measure the image quality from a pixel level, while LPIPS evaluates the generated from a deep feature level.

State-of-the-Art Comparisons. The proposed method is compared with several facial image generation models, i.e., StarGAN [25], Pix2pix [55], GPGAN [28], PG2 [98], Pix2pixSC [182], and CocosNet [219]. Note that StarGAN is an unsupervised generation method, while the others are supervised generation models. The comparison with StarGAN is just to see how big the gap between supervised and unsupervised methods is for this task. The results are shown in Table 3.2. We observe that the proposed C2GAN achieves the best results on all four evaluation metrics, validating the effectiveness of our method.

Qualitative comparison results compared with StarGAN, Pix2pix, GPGAN, PG2 are shown in Figure 3.4. Clearly, GPGAN performs the worse among all the comparison models. Pix2pix can generate the correct expres-

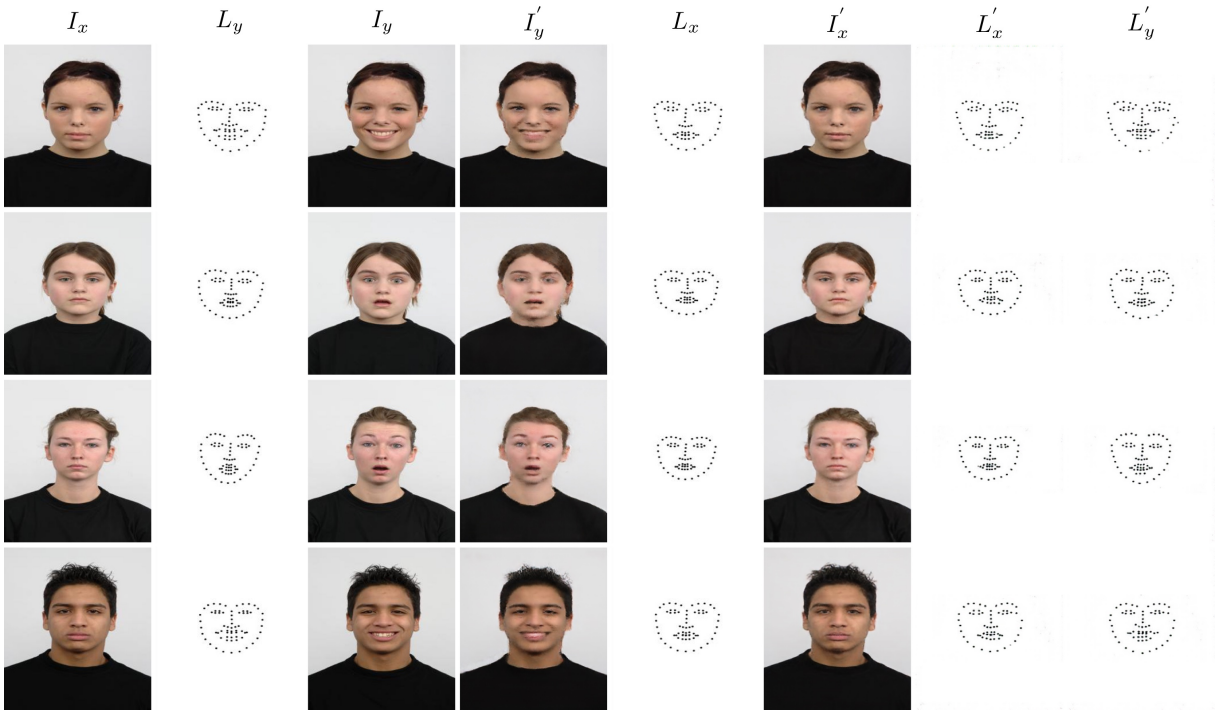


Figure 3.5: Visualization of facial landmark generation on the facial expression generation task.

sion but the faces are distorted. StarGAN can generate sharper faces, but the details of these generated faces are not convincing. For instance, the mouths in StarGAN are blurred or not correct. Moreover, we observe that the results of PG2 tend to be blurry. Compared with exiting leading methods, the results generated by the proposed C2GAN are smoother, sharper and contain more details. We also show some generated facial landmarks in Figure 3.5. We see that the proposed method not only produces realistic images but also generates reasonable facial landmarks. This is not provided by any existing facial expression generation works.

3.4.3 Hand Gesture-to-Gesture Translation

Datasets. We follow GestureGAN [163] and adopt the Creative Senz3D dataset [101] for the hand gesture-to-gesture translation task. This dataset contains 11 different hand gestures performed by four people, each perform-

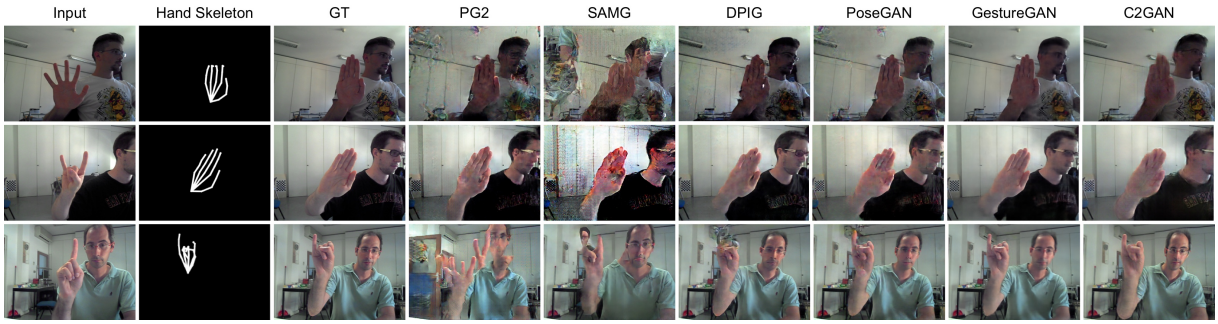


Figure 3.6: Qualitative comparison of hand gesture-to-gesture translation on the Senz3d dataset. From left to right: Input, Hand Skeleton, Ground Truth (GT), PG2 [98], SAMG [201], DPIG [99], PoseGAN [146], GestureGAN [163], and C2GAN (Ours).

Method	PSNR ↑	AMT ↑	FRD ↓
PG2 [98]	26.5138	2.8	3.0933
SAMG [201]	26.9545	2.3	3.1006
DPIG [99]	26.9451	6.9	3.0846
Pix2pixSC [182]	27.0569	7.2	3.0814
CocosNet [219]	27.1532	7.9	3.0741
PoseGAN [146]	27.3014	8.6	3.0467
GestureGAN [163]	**27.9749**	**22.6**	**2.9836**
C2GAN (Ours)	27.2531	12.7	3.0573

Table 3.3: Quantitative comparison of hand gesture-to-gesture translation on the Senz3D dataset. For all metrics except FRD, higher is better.

ing gesture is repeated 30 times, thus we have 4 subjects × 11 gestures × 30 times = 1320 images in total. We follow [163] and select 12,800 and 135,504 pairs as testing and training data, respectively.

Evaluation Metrics. We follow [163] and adopt Peak Signal-to-Noise Ratio (PSNR) and FRD [163] as evaluation metrics. PSNR measures the similarity between the real image and the generated image from a pixel level. FRD measures the distance between the real image and the fake image from a deep feature level. Moreover, we follow [163] and conduct a user study to evaluate the generated image by different models.

State-of-the-Art Comparisons. We adopt the most related several works, i.e., PG2 [98], DPIG [99], PoseGAN [146], GestureGAN [163], SAMG [201], Pix2pixSC [182], and CocosNet [219], as our baselines for

the facial expression generation task. Comparison results are shown in Table 3.3. We observe that the proposed method achieves very competitive results compared with the leading methods. Specifically, the proposed C2GAN achieves significantly better results than PG2, SAMG, DPIG, Pix2pixSC, and CocosNet on all metrics. Moreover, we see that PoseGAN obtains slightly better results than our C2GAN on both PSNR and FRD metrics, however, the proposed C2GAN achieves better AMT than PoseGAN. Also, the proposed C2GAN achieves visually better results than PoseGAN, as shown in Figure 3.6. Lastly, we observe that GestureGAN achieves better results than C2GAN on all metrics. The reason is that GestureGAN is carefully tailored and designed for the specific hand gesture-to-gesture translation task, meaning that GestureGAN is fine-turned on this task with the network structure, loss objective, and hyper-parameter selection. However, the proposed C2GAN is a novel and unified GAN model, which can be used to handle all kinds of settings of guided image-to-image translation without modifying the network structure, the loss objective, and hyper-parameters. Furthermore, our C2GAN can generate both images and guidances, which is not considered in GestureGAN.

Qualitative comparison results compared with PG2, DPIG, PoseGAN, GestureGAN, SAMG are shown in Figure 3.6. We observe that the proposed method generates much better images than PG2, DPIG, SAMG, and PoseGAN. Moreover, our results are very close to those generated by GestureGAN. Our C2GAN is a joint learning framework and it is not only able to generate the target images but is also able to produce the hand skeleton of the input image, which will benefit other computer vision tasks such as hand pose estimation. The results of the generated hand skeletons are shown in Figure 3.7. We see that the generated hand skeleton $L_x^{'}$ is very similar to the real hand skeleton L_x, which verifies the effectiveness of the guidance generator G_g and our joint learning strategy.

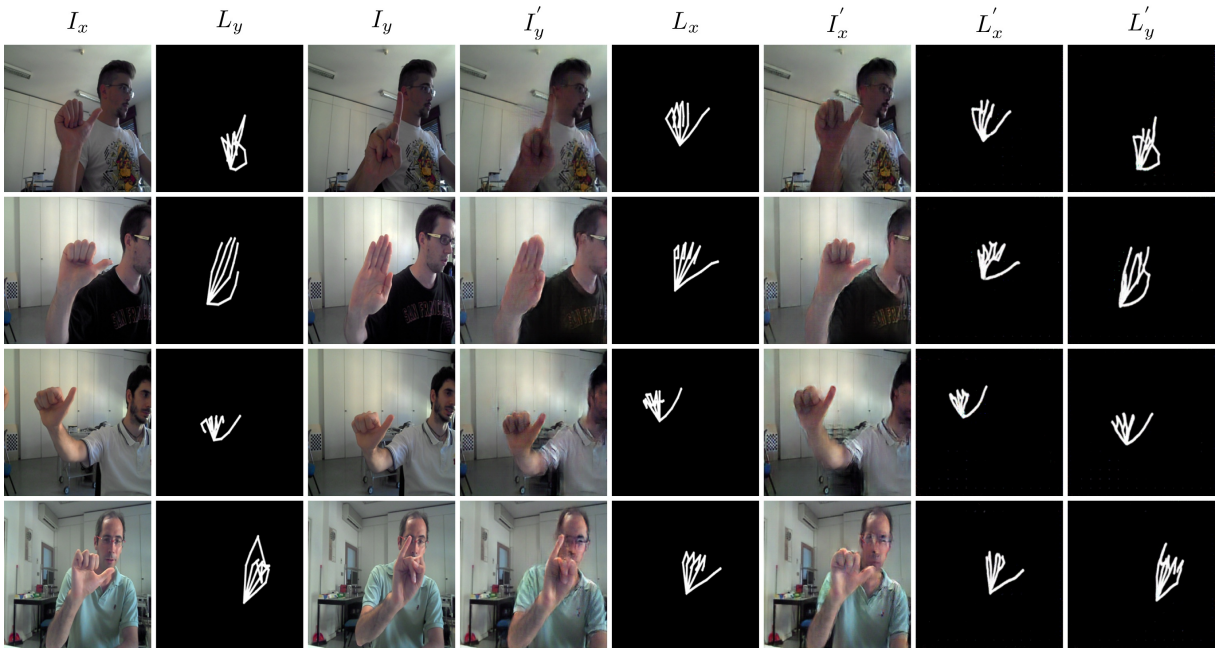


Figure 3.7: Visualization of hand skeleton generation on the hand gesture-to-gesture translation task.

3.4.4 Cross-View Image Translation

Datasets. We follow [133] and adopt the Dayton dataset [179] to evaluate the cross-view image translation task. This dataset contains 76,048 images and we create a training/testing split of 55,000/21,048. The original size of images is 354×354 resolution. We resize them to 256×256.

Evaluation Metrics. Following [133], we employ Inception Score (IS), top-k prediction accuracy, and KL score for the quantitative analysis. These three metrics evaluate the generated images from a high-level feature space.

State-of-the-Art Comparisons. We adopt several leading cross-view image translation methods as our baselines, i.e., Pix2pix [55], X-SO [134], X-Fork [133] and X-Seq [133]. These methods aim to generate images based on a given image. To further evaluate the proposed C2GAN, we introduce four strong baselines, i.e., Pix2pix++ [55], X-Fork++ [133], X-Seq++ [133] and SelectionGAN [165]. These four models aim to generate images based

Method	Accuracy (%)				Inception Score			KL ↓
	Top-1 ↑		Top-5 ↑		All ↑	Top-1 ↑	Top-5 ↑	
Pix2pix [55]	6.80	9.15	23.55	27.00	2.8515	1.9342	2.9083	38.26 ± 1.88
X-SO [134]	27.56	41.15	57.96	73.20	2.9459	2.0963	2.9980	7.20 ± 1.37
X-Fork [133]	30.00	48.68	61.57	78.84	3.0720	2.2402	3.0932	6.00 ± 1.28
X-Seq [133]	30.16	49.85	62.59	80.70	2.7384	2.1304	2.7674	5.93 ± 1.32
Pix2pix++ [55]	32.06	54.70	63.19	81.01	**3.1709**	2.1200	3.2001	5.49 ± 1.25
X-Fork++ [133]	34.67	59.14	66.37	84.70	3.0737	2.1508	3.0893	4.59 ± 1.16
X-Seq++ [133]	31.58	51.67	65.21	82.48	3.1703	2.2185	**3.2444**	4.94 ± 1.18
SelectionGAN [165]	42.11	68.12	**77.74**	**92.89**	3.0613	**2.2707**	3.1336	2.74 ± 0.86
C2GAN (Ours)	**45.80**	**75.28**	76.03	90.67	2.9603	2.1225	2.9435	**2.70 ± 1.02**

Table 3.4: Quantitative comparison of cross-view image translation on the Dayton dataset in a2g direction. For all metrics except KL, higher is better.

on a given image and several novel segmentation maps. Note that we implement Pix2pix++, X-Fork++ and X-Seq++ using their released public code.

Comparison results are shown in Table 3.4. We see that the proposed C2GAN achieves the best results on several metrics such as KL and Top-1 Accuracy. For other metrics, the proposed method still achieves very competitive results, which validates the effectiveness of the proposed C2GAN.

Several qualitative comparison results are also provided in Figure 3.8. We see that our C2GAN generates much better realistic images than other baselines. Moreover, we show the generated segmentation maps by our method in Figure 3.9, the proposed C2GAN can generate reasonable segmentation maps, which we believe our method can be used to improve the performance of semantic segmentation tasks.

3.4.5 Ablation Study

We conduct extensive experiments on the Radbound Faces dataset to verify the effectiveness of each component of the proposed C2GAN. All the experiments are trained with 50 epochs and Table 3.5 shows the quantitative comparison results.

Figure 3.8: Qualitative comparison of cross-view image translation on the Dayton dataset. From left to right: Input, Ground Truth (GT), Pix2pix [55], X-SO [134], X-Fork [133], X-Seq [133], Pix2pix++ [55], X-Fork++ [133], X-Seq++ [133], SelectionGAN [165], and C2GAN (Ours).

Influence of Individual Generation Cycle. To evaluate the influence of individual generation cycle, we test with four different combinations of the cycles, i.e., 'I2I2I', 'I2I2I+G2I2G', 'I2I2I+G2R2G', and 'I2I2I+G2I2G+G2R2G'. All four combinations use the same training strategies and hyper-parameters. Comparison results are shown in Table 3.5. We can see that 'I2I2I', 'G2I2G', and 'G2R2G' are all critical to the final result and the removal of one of them degrades the generation performance, demonstrating that by using cross-modal data in a joint framework and by making the cycles constraint on each other improve the final generation performance. Moreover, 'I2I2I+G2I2G+G2R2G' obtains the best performance among all the four combination settings. Meanwhile, 'I2I2I+G2I2G+G2R2G' achieves remarkably better results than I2I2I on all metrics, demonstrating the effectiveness of constraining both image and guidance cycles facilitating thus a more robust optimization of the whole model. Moreover, some visualization results are provided in Figure 3.10 to show the influence of each generation cycle. We can obtain the similar conclusion as the one from Table 3.5, further validating our network design.

Cross-Modal vs. Single-Modal Discriminator. We then evaluate the influence of the proposed cross-modal discriminator, i.e., 'C2GAN w/

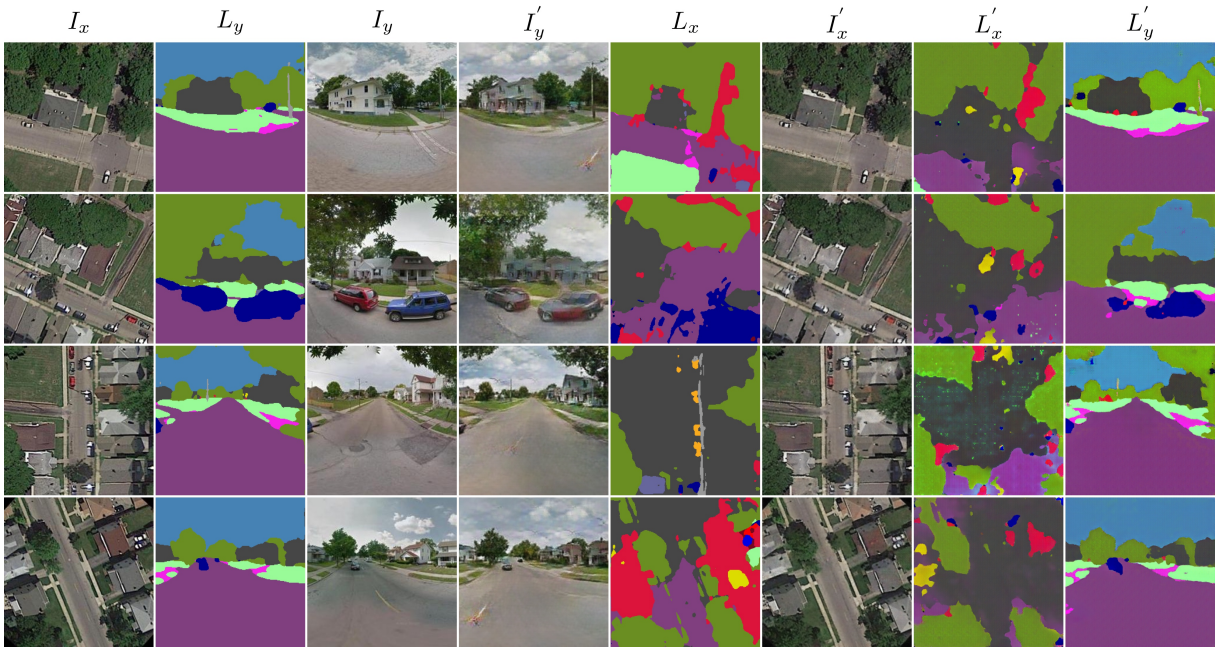


Figure 3.9: Visualization of segmentation map generation on the cross-view image translation task.

I2I2I+G2I2G+G2R2G'. Our baseline is the traditional single-modal discriminator ('C2GAN w/ Single-Modal D'). The results are listed in Table 3.5. We see that the proposed cross-modal discriminator achieves much better results than the single-modal discriminator on all metrics, meaning that the rich cross-modal information helps to learn a better discriminator and thus facilitates the optimization of the generator.

Parameter Sharing between Generators. The parameter sharing could remarkably reduce parameters of the whole network. We then evaluate how the parameter-sharing strategy would affect the generation results. Specifically, two different baselines are tested: one is 'C2GAN w/ I2I2I+G2I2G+G2R2G', which shares the network parameters between the two image generators, and between the two guidance generators, respectively. While 'C2GAN w/ Non-Sharing G' learns the four different generators, separately. We can see from Table 3.5 that the non-sharing one achieves slightly better performance than the sharing one. However, the

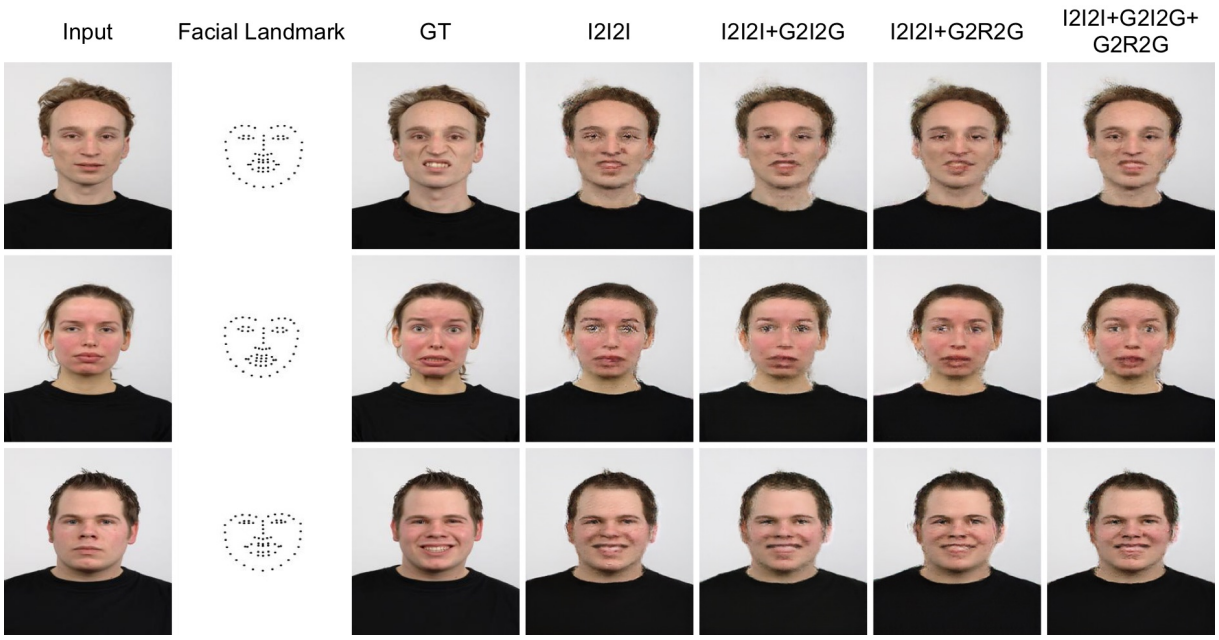


Figure 3.10: Influence of individual generation cycle on the Radbound Faces dataset.

Baseline	AMT ↑	PSNR ↑	SSIM ↑
C2GAN w/ I2I2I	25.3	21.2030	0.8449
C2GAN w/ I2I2I + G2I2G	28.2	20.8708	0.8419
C2GAN w/ I2I2I + G2R2G	28.7	21.0156	0.8437
C2GAN w/ I2I2I + G2I2G + G2R2G	30.8	21.6262	0.8540
C2GAN w/ Single-Modal D	26.4	21.2794	0.8426
C2GAN w/ Non-Sharing G	**32.9**	**21.6353**	**0.8611**

Table 3.5: Quantitative comparison of ablation study on the Radbound Faces dataset. For all metrics, higher is better.

number of parameters of non-sharing one is 217.6M, which is twice as much as that of the sharing one. This means that the parameter-sharing strategy is a good way to balance both image performance and network overhead.

3.5 Conclusion

We propose a novel and unified Cycle In Cycle Generative Adversarial Network (C2GAN) for guided image-to-image translation tasks. The proposed C2GAN contains two different types of generators, i.e., image-oriented gen-

erator and guidance-oriented generator. Both generators are connected in three generation cycles and can be optimized in an end-to-end fashion. Extensive qualitative and quantitative experimental results on four challenging generative tasks demonstrate that the proposed C2GAN is effective to generate photo-realistic images with convincing details.

In next chapter, we will introdcue XingGAN and BiGraphGAN for person image generation. XingGAN explores cascaded guidance with two different generation branches, and aims at progressively producing a more detailed synthesis from both person shape and appearance embeddings. BiGraphGAN aims to progressively reason the pose-to-pose and pose-to-image relations via the proposed Bipartite Graph Reasoning (BGR) block.

Chapter 4

XingGAN and BiGraphGAN

We propose a novel Generative Adversarial Network (XingGAN or CrossingGAN) for person image generation tasks, i.e., translating the pose of a given person to a desired one. The proposed Xing generator consists of two generation branches that model the person's appearance and shape information, respectively. Moreover, we propose two novel blocks to effectively transfer and update the person's shape and appearance embeddings in a crossing way to mutually improve each other, which has not been considered by any other existing GAN-based image generation work. Extensive experiments on two challenging datasets, i.e., Market-1501 and DeepFashion, demonstrate that the proposed XingGAN advances the state-of-the-art performance both in terms of objective quantitative scores and subjective visual realness. The source code and trained models are available at https://github.com/HaOTang/XingGAN.

We also present a novel Bipartite Graph Reasoning GAN (BiGraphGAN) for the challenging person image generation task. The proposed graph generator mainly consists of two novel blocks that aim to model the pose-to-pose and pose-to-image relations, respectively. Specifically, the proposed Bipartite Graph Reasoning (BGR) block aims to reason the crossing long-range relations between the source pose and the target pose in a bi-

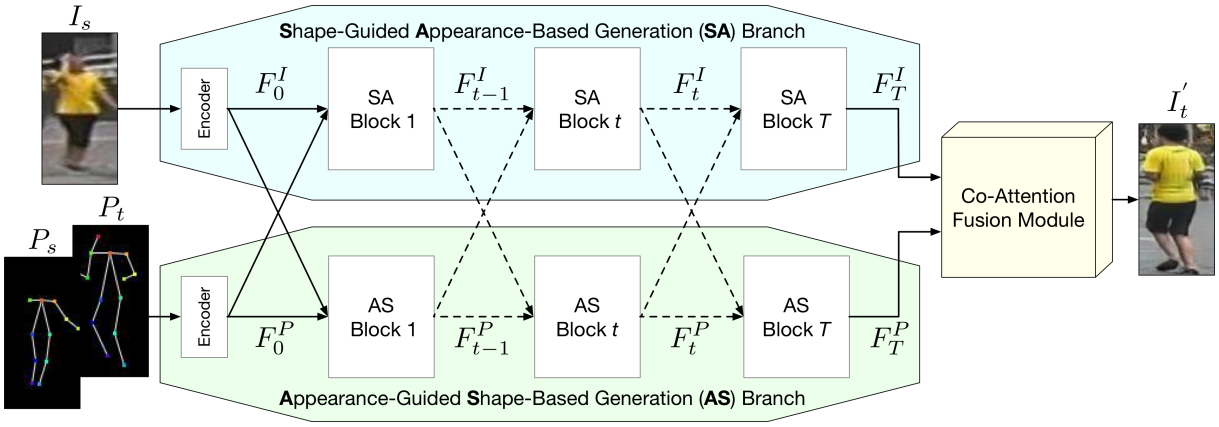


Figure 4.1: Overview of the proposed Xing generator. Both the Shape-guided Appearance-based generation (SA) and the Appearance-guided Shape-based generation (AS) branches consist of a sequence of SA and AS blocks in a crossing way. All these components are trained in an end-to-end fashion so that the SA branch and AS branch can benefit from each other to generate more shape-consistent and appearance-consistent person images.

partite graph, which mitigates some challenges caused by pose deformation. Moreover, we propose a new Interaction-and-Aggregation (IA) block to effectively update and enhance the feature representation capability of both person's shape and appearance in an interactive way. Experiments on two challenging and public datasets, i.e., Market-1501 and DeepFashion, show the effectiveness of the proposed BiGraphGAN in terms of objective quantitative scores and subjective visual realness. The source code and trained models are available at https://github.com/HaoTang/BiGraphGAN.

4.1 Introduction

The problem of person image generation aims to generate photo-realistic person images conditioned on an input person image and several desired poses. This task has a wide range of applications such as person image/video generation [202, 43, 8, 54, 94] and person re-identification [234, 128]. Exiting methods such as [98, 99, 146, 234, 164] have achieved promising performance on this challenging task. For example, Zhu et al. [234]

recently proposed a conditional GAN model that comprises a sequence of pose-attentional transfer blocks. Wherein, each block transfers certain regions it attends to and progressively generates the desired person image.

Although [234] performed an interesting exploration, we still observe unsatisfactory aspects and visual artifacts in the generated person images due to several reasons. First, [234] stacks several convolution layers to generate the attention maps of the shape features, then the generated attention maps are used to attentively highlight the appearance features. Since convolutional operations are building blocks that process one local neighborhood at a time, this means that they cannot capture the joint influence between the appearance and the shape features. Second, the attention maps in [234] are only produced by using one single modality, i.e., the pose, leading to insufficiently accurate correlations for both modalities (i.e., the pose and the image modality), and thus misguiding the image generation.

Based on these observations, we propose a novel Generative Adversarial Network (XingGAN or CrossingGAN), which consists of a Xing generator, a shape-guided discriminator, and an appearance-guided discriminator. The overall framework is shown in Figure 4.1. The Xing generator consists of three parts, i.e., a Shape-guided Appearance-based generation (SA) branch, an Appearance-guided Shape-based generation (AS) branch, and a co-attention fusion module. Specifically, the proposed SA branch contains a sequence of SA blocks, which aim to progressively update the appearance representation under the guidance of the shape representation, while the proposed AS branch contains a sequence of AS blocks, which aim to progressively update the shape representation under the guidance of the appearance representation. We also present a novel crossing operation in both SA and AS blocks to capture the joint influence between the image modality and the pose modality by creating attention maps jointly pro-

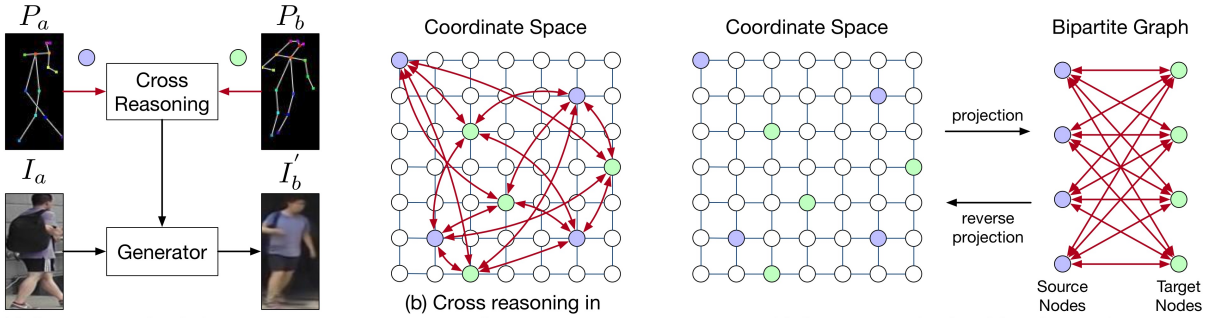
duced by both modalities. Moreover, we introduce a co-attention fusion model to better fuse the final appearance and shape features to generate the desired person images. We present an appearance-guided discriminator and a shape-guided discriminator to jointly judge how likely is that the generated image contains the same person in the input image and how well the generated image aligns with the targeted pose, respectively. The proposed XingGAN is trained in an end-to-end fashion so that the generation branches can enjoy the mutually improved benefits from each other.

We conduct extensive experiments on two challenging datasets, i.e., Market-1501 [224] and DeepFashion [97]. Qualitative and quantitative results demonstrate that XingGAN achieves better results than state-of-the-art methods, regarding both visual fidelity and alignment with targeted person poses.

To summarize, the contributions of our paper are three-fold:

- We propose a novel XingGAN (or CrossingGAN) for person image generation. It explores cascaded guidance with two different generation branches, and aims at progressively producing a more detailed synthesis from both person shape and appearance embeddings.
- We propose SA and AS blocks, which effectively transfer and update person shape and appearance features in a crossing way to mutually improve each other, and are able to significantly boost the quality of the final outputs.
- Extensive experiments clearly demonstrate the effectiveness of XingGAN, and show new state-of-the-art results on two challenging datasets, i.e., Market-1501 [224] and DeepFashion [97].

We also observe that existing person image generation methods such as [98, 99, 146, 164, 2, 36, 234, 14, 8, 210, 88, 94] always rely on building convolution layers. Due to the physical design of convolutional filters, convolution operations can only model local relations. To capture global



(a) Cross reasoning in image space. (b) Cross reasoning in coordinate space. (c) Cross reasoning in a bipartite graph.

Figure 4.2: Illustration of our motivation. We propose a novel BiGraphGAN (Fig. (c)) for capturing crossing long-range relations between the source pose P_a and the target pose P_b in a bipartite graph. The node features from both source and target poses in the coordinate space are projected into the nodes in a bipartite graph, thereby forming a fully-connected bipartite graph. After cross-reasoning the graph, the node features are projected back to the original coordinate space for further processing.

relations, existing methods such as [234, 164] inefficiently stack multiple convolution layers to enlarge the receptive fields to cover all the body joints from both the source pose and the target pose. However, none of the above-mentioned methods explicitly consider modeling the cross relations between the source pose and the target pose.

In this paper, we propose a novel Bipartite Graph Reasoning GAN (BiGraphGAN), which mainly consists of two novel blocks, i.e., Bipartite Graph Reasoning (BGR) block and Interaction-and-Aggregation (IA) block. The BGR block aims to efficiently capture the crossing long-range relations between the source pose and the target pose in a bipartite graph (see Figure 4.2). Specifically, the BGR block first projects both the source pose feature and the target pose feature in the original coordinate space onto a bipartite graph. Next, both source and target pose features are represented by a set of nodes to form a fully-connected bipartite graph, on which crossing long-range relation reasoning is performed by Graph Convolution Networks (GCNs). To the best of our knowledge, we are the first to explore GCNs to model the crossing long-range relations for solving the challenging person image generation task. After reasoning, we project the

node features back to the original coordinate space for further processing.

Also, the proposed IA block is proposed to effectively and interactively enhance person's shape and appearance features. We also introduce an Attention-based Image Fusion (AIF) module to selectively generate the final result using an attention network. Qualitative and quantitative experiments on two challenging datasets, i.e., Market-1501 [224] and DeepFashion [97], demonstrate that the proposed BiGraphGAN generates better person images than several state-of-the-art methods, i.e., PG2 [98], DPIG [99], Deform [146], C2GAN [164], BTF [2], VUnet [36] and PATN [234].

The contributions of this paper are summarized as follows,

- We propose a novel Bipartite Graph Reasoning GAN (BiGraphGAN) for person image generation. The proposed BiGraphGAN aims to progressively reason the pose-to-pose and pose-to-image relations via two novel proposed blocks.
- We propose a novel Bipartite Graph Reasoning (BGR) block to effectively reason the crossing long-range relations between the source pose and the target pose in a bipartite graph by using Graph Convolutional Networks (GCNs). Moreover, we present a new Interaction-and-Aggregation (IA) block to interactively enhance both person's appearance and shape feature representations.
- Extensive experiments on two challenging datasets, i.e., Market-1501 [224] and DeepFashion [97], demonstrate the effectiveness of the proposed BiGraphGAN and show significantly better performance compared with state-of-the-art approaches.

4.2 Related Work

Generative Adversarial Networks (GANs) [41] have shown the potential to generate realistic images [140, 63, 10]. For instance, Shaham

et al. propose an unconditional SinGAN [140] which can be learned from a single image. Moreover, to generate user-defined images, Conditional GAN (CGAN) [103] has been proposed recently. A CGAN always consists of a vanilla GAN and external guide information such as class labels [193, 25, 216], segmentation maps [165, 119, 169, 92], attention maps [66, 166, 100], and human skeleton [2, 8, 234, 163, 156]. In this work, we mainly focus on the challenging person image generation task, which aims to transfer a person image from one pose to another one.

Image-to-Image Translation aims to learning the translation mapping between target and input images. CGANs have achieved decent results in pixel-wise aligned image-to-image translation tasks [55, 163, 2]. For example, Isola et al. propose Pix2pix [55], which adopts CGANs to generate the target domain images based on the input domain images, such as photo-to-map, sketch-to-image, and night-to-day. However, pixel-wise alignment is not suitable for person image generation tasks due to the shape deformation between the input person image and target person image.

Person Image Generation. To remedy this, several works started to use poses to guide person image generation [98, 99, 146, 36, 164, 234]. For example, Ma et al. first present PG2 [98], which is a two-stage model to generate the target person images based on an input image and the target poses. Moreover, Siarohin et al. propose PoseGAN [146], which requires an extensive affine transformation computation to deal with the input-output misalignment caused by pose differences. Zhu et al. propose Pose-Transfer [234], which contains a sequence of pose-attentional transfer blocks to generate the target person image progressively. Besides the aforementioned supervised methods, several works focus on solving this task in an unsupervised setting [125, 150]. For instance, Pumarola et al. propose an unsupervised framework [125] to generate person images, which induces some geometric errors as revealed in their paper.

Note that the aforementioned methods adopt human keypoints or skeleton as pose guidance, which are usually extracted by using OpenPose [13]. In addition, several works adopt DensePose [109], 3D pose [87], and segmented pose [31] to generate person images because they contain more information about body depth and part segmentation, producing better results with more texture details. However, the keypoint-based pose representation is much cheaper and more flexible than the DensePose, 3D pose, segmented pose representations, and can be more easily applied to practical applications. Therefore, we favor keypoint-based pose representation in this paper.

Also, person image generation is a challenging task due to the pose deformation between the source image and the target image. Modeling the long-range relations between the source pose and the target pose is the key to solving this challenging task. However, existing methods such as [98, 99, 8, 146, 164, 2, 36, 234, 14, 210, 88, 94] built through the stacking of convolutional layers, which can only leverage the relations between the source pose and the target pose locally. For instance, Zhu et al. [234] propose a Pose-Attentional Transfer Block (PATB), in which the source and target poses are simply concatenated and then fed into an encoder to capture their dependencies.

Unlike existing methods for modeling the relations between the source and target poses in a localized manner, we show that the proposed Bipartite Graph Reasoning (BGR) block can bring considerable performance improvements in the global view.

Image-Guidance Conditioning Schemes. Recently, there were proposed many schemes to incorporate the extra guidance (e.g., human poses [98, 234], segmentation maps [119, 165, 169], facial landmarks [164, 209], etc) into an image-to-image translation model, which can be divided into four categories, i.e., input concatenation [164, 194, 221], feature concate-

nation [98, 99, 36, 87, 73, 86], one-way guidance-to-image interaction [146, 119, 52, 124], two-way guidance-and-image interaction [234, 2, 22].

The most straightforward way of conditioning the guidance is to concatenate the input image and the guidance along the channel dimension. For example, C2GAN [164] takes the input person image and the targeted poses as input to output the corresponding targeted person images. Instead of concatenating the guidance and the image at the input, several works [98, 99, 36] concatenate their feature representations at a certain layer. For instance, PG2 [98] concatenates the embedded pose feature with the embedded image feature at the bottleneck fully connected layer. Another more general scheme is to use the guidance to guide the generation of the image. For example, Siarohin et al. [146] first learn an affine transformation between the input and the target pose, then they use it to 'move' the feature maps between the input image and the targeted image. Unlike existing one-way guidance-to-image interaction schemes that allow information flow only from the guidance to the input image, a recent scheme, i.e., two-way guidance-and-image interaction, also considers the information flow from the input image back to the guidance [234, 2]. For example, Zhu et al. [234] propose an attention-based GAN model to simultaneously update the person's appearance and shape features under the guidance of each other, and show that the proposed two-way guidance-and-image interaction strategy leads to better performance on person image generation tasks.

Contrary to the existing two-way guidance-and-image interaction schemes [234, 2] that allow both the image and guidance to guide and update each other in a local way, we show that the proposed cross-conditioning strategy can further improve the performance of person image generation tasks.

Graph-Based Reasoning. Graph-based approaches have shown to be an efficient way to reason relation in many computer vision tasks such as semi-

supervised classification [69], video recognition [188], crowd counting [19], action recognition [200, 122] and semantic segmentation [21, 218].

Compared to these graph-based reasoning methods which model the long-range relations within the same feature map to incorporate global information, we focus on developing a novel BiGraphGAN framework that reasons and models the crossing long-range relations between different features of the source pose and target pose in a bipartite graph. Then the crossing relations are further used to guide the image generation process (see Figure 4.2). This idea has not been investigated in existing GAN-based image translation methods.

4.3 Model Description

We start by presenting the details of the proposed XingGAN (Figure 4.1) consisting of three parts, i.e., a Shape-guided Appearance-based generation (SA) branch modeling the person shape representation, an Appearance-guided Shape-based generation (AS) branch modeling the person appearance representation, and a Co-Attention Fusion (CAF) module for fusing these two branches. In the following, we first present the design of the two proposed generation branches, and then introduce the co-attention fusion module. Lastly, we present the proposed two discriminators, the overall optimization objective and implementation details.

The inputs of the proposed Xing generator are the source image I_s, the source pose P_s, and the target pose P_t. The goal is to translate the pose of the person in the source image I_s from the source pose P_s to the target pose P_t, thus synthesizing a photo-realistic person image $I_t^{'}$. In this way, the source image I_s provides the appearance information and the poses (P_s, P_t) provide the shape information to the Xing generator for synthesizing the desired person image.

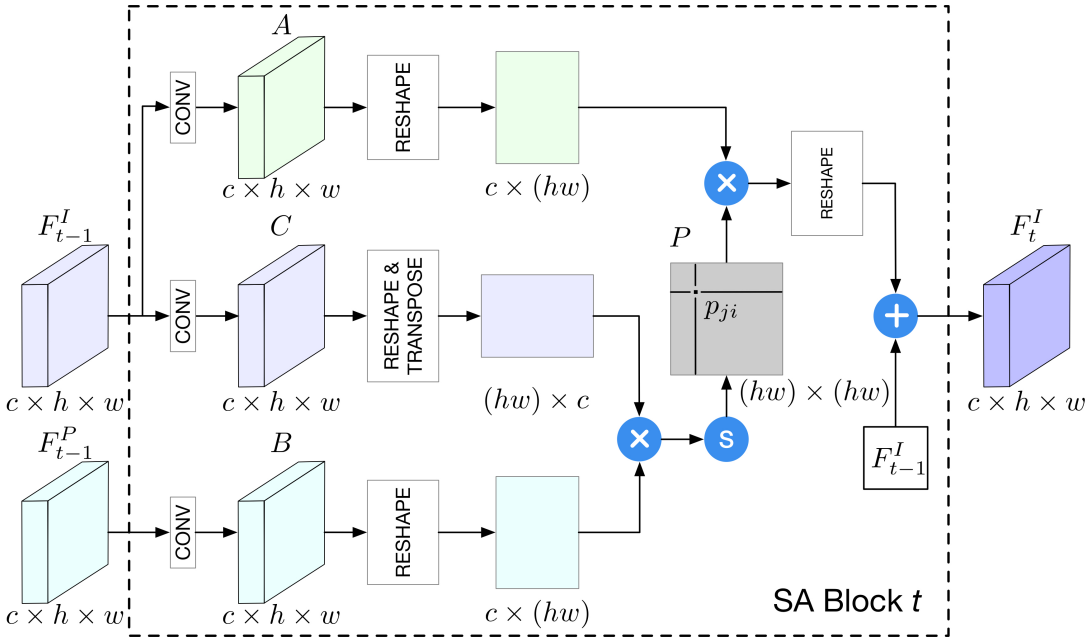


Figure 4.3: Structure of the proposed SA block which takes the previous appearance code F^I_{t-1} and the previous shape code F^P_{t-1} as input and obtains the appearance code F^I_t in a crossed non-local way. The symbols $\oplus$, $\otimes$ and Ⓢ and Ⓒ denote element-wise addition, element-wise multiplication, Softmax activation, and channel-wise concatenation, respectively.

4.3.1 Shape-Guided Appearance-Based Generation

The proposed Shape-guided Appearance-based generation (SA) branch consists of an image encoder and a series of the proposed SA blocks. The source image I_s is first fed into the image encoder to produce the appearance code F^I_0, as shown in Figure 4.1. The encoder consists of two convolutional layers in our experiments. The SA branch contains several cascaded SA blocks which progressively update the initial appearance code F^I_0 to the final appearance code F^I_T under the guidance of the AS branch. As we can see in Figure 4.1, all SA blocks have an identical network structure. Consider the t-th block in Figure 4.3, whose inputs are the appearance code $F^I_{t-1} \in \mathbb{R}^{c \times h \times w}$ and the shape code $F^P_{t-1} \in \mathbb{R}^{c \times h \times w}$. The output is the refined appearance code $F^I_t \in \mathbb{R}^{c \times h \times w}$. Specifically, given the appearance code F^I_{t-1}, we first feed it into a convolution layer to generate a new appearance code

C, where $C \in \mathbb{R}^{c \times h \times w}$. Then we reshape C to $\mathbb{R}^{c \times (hw)}$, where $n=hw$ is the number of pixels. At the same time, the SA block receives the shape code F_{t-1}^{P} from the AS branch, which is also fed into a convolution layer to produce a new shape code $B \in \mathbb{R}^{c \times h \times w}$ and then reshape to $\mathbb{R}^{c \times (hw)}$. After that, we perform a matrix multiplication between the transpose of C and B, and apply a Softmax layer to produce a correlation matrix $P \in \mathbb{R}^{(hw) \times (hw)}$,

$$p_{ji} = \frac{\exp(B_i C_j)}{\sum_{i=1}^{n} \exp(B_i C_j)}, \tag{4.1}$$

where p_{ji} measures the impact of the i-th position of B on the j-th position of the appearance code C. In this crossing way, the SA branch can capture more joint influence between the appearance code F_{t-1}^{I} and shape code F_{t-1}^{P}, producing a richer appearance code F_{t}^{I}.

Note that Equation (4.1) has a close relationship with the non-local operator proposed by Wang et al. [187]. The major difference is that the non-local operator in [187] computes the pairwise similarity within the same feature map to incorporate global information, whereas the proposed crossing way computes the pairwise similarity between different feature maps, i.e., the person appearance and shape feature maps.

After that, we feed F_{t-1}^{I} into a convolution layer to produce a new appearance code $A \in \mathbb{R}^{c \times h \times w}$ and reshape it to $\mathbb{R}^{c \times (hw)}$. We then perform a matrix multiplication between A and the transpose of P and reshape the result to $\mathbb{R}^{c \times h \times w}$. Finally, we multiply the result by a scale parameter α and conduct an element-wise sum operation with the original appearance code F_{t-1}^{I} to obtain the refined appearance code $F_{t}^{I} \in \mathbb{R}^{c \times h \times w}$,

$$F_{t}^{I} = \alpha \sum_{i=1}^{n} (p_{ji} A_i) + F_{t-1}^{I}, \tag{4.2}$$

where α is 0 in the beginning and but is gradually updated. By doing

so, each position of the refined appearance code F_t^I is a weighted sum of all positions of the shape code F_{t-1}^P and the previous appearance code F_{t-1}^I. Thus, it has a global contextual view between F_{t-1}^P and F_{t-1}^I, and it selectively aggregates useful contexts according to the correlation matrix P.

4.3.2 Appearance-Guided Shape-Based Generation

In our preliminary experiments, we observe that only the SA generation branch is not sufficient to learn such a complex deformable translation process. Intuitively, since the shape features can guide the appearance features, we believe the appearance features can also be used to guide the shape features in turn. Therefore, we also propose an Appearance-guided Shape-based generation (AS) branch. The proposed AS branch mainly consists of a pose encoder and a sequence of AS blocks, as shown in Figure 4.1. The source pose P_s and target pose P_t are first concatenated along the channel dimension and then fed into the pose encoder to produce the initial shape representation F_0^P. The pose encoder has the same network structure as the image encoder. Note that to capture the dependency between the two poses, we only adopt one pose encoder.

The AS branch contains several cascaded AS blocks, which progressively update the initial shape code F_0^P to the final shape code F_T^P under the guidance of the SA branch. All AS blocks have the same network structure, as illustrated in Figure 4.1. Consider the t-th block in Figure 4.4, whose inputs are the shape code $F_{t-1}^P \in \mathbb{R}^{c\times h\times w}$ and the appearance code $F_{t-1}^I \in \mathbb{R}^{c\times h\times w}$. The output is the refined shape code $F_t^P \in \mathbb{R}^{c\times h\times w}$.

Specifically, given the shape code F_{t-1}^P, we first feed it into a convolution layer to generate a new shape code H, where $H \in \mathbb{R}^{c\times h\times w}$. We then reshape H to $\mathbb{R}^{c\times(hw)}$. At the same time, the AS block receives the appearance code F_{t-1}^I from the SA branch, which is also fed into a convolution layer to produce a new appearance code E and then reshape it to $\mathbb{R}^{c\times(hw)}$. Af-

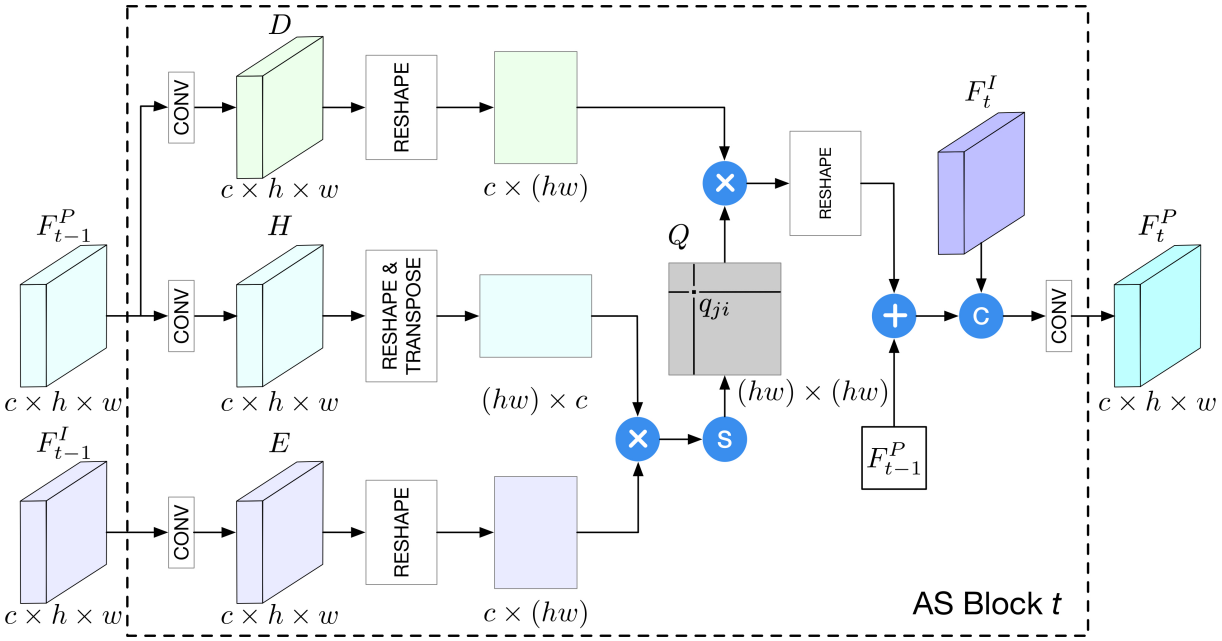


Figure 4.4: Structure of the proposed AS block, which takes the previous shape code F_{t-1}^P and the previous appearance code F_{t-1}^I as inputs and obtains the shape code F_t^P in a crossing way. The symbols $\oplus$, $\otimes$ and Ⓢ and Ⓒ denote element-wise addition, element-wise multiplication, Softmax activation, and channel-wise concatenation, respectively.

ter that, we perform a matrix multiplication between the transpose of H and E, and apply a Softmax layer to produce another correlation matrix $Q \in \mathbb{R}^{(hw)\times(hw)}$,

$$q_{ji} = \frac{\exp(E_i H_j)}{\sum_{i=1}^{n} \exp(E_i H_j)}, \tag{4.3}$$

where q_{ji} measures the impact of i-th position of E on the j-th position of the shape code H. $n{=}hw$ is the number of pixels.

Meanwhile, we feed F_{t-1}^P into a convolution layer to produce a new shape code $D \in \mathbb{R}^{c\times h\times w}$ and reshape it to $\mathbb{R}^{c\times(hw)}$. We then perform a matrix multiplication between D and the transpose of Q and reshape the result to $\mathbb{R}^{c\times h\times w}$. Finally, we multiply the result by a scale parameter β and conduct an element-wise sum operation with the original shape code F_{t-1}^P. The result is then concatenated with the appearance code F_t^I and fed into

a convolution layer to obtain the updated shape code $F_t^P \in \mathbb{R}^{c\times h\times w}$,

$$F_t^P = \text{Concat}(\beta \sum_{i=1}^{n}(q_{ji}D_i) + F_{t-1}^P, F_t^I), \tag{4.4}$$

where Concat$(\cdot)$ denotes the channel-wise concatenation operation and β is a parameter. By doing so, each position in the refined shape code F_t^P is a weighted sum of all positions in the appearance code F_{t-1}^I and previous shape code F_{t-1}^P.

4.3.3 Co-Attention Fusion

The proposed Co-Attention Fusion (CAF) module consists of two parts, i.e., generating intermediate results and co-attention maps. These co-attention maps are used to spatially select from both the intermediate generations and the input image, and are combined to synthesize a final output. This idea of the proposed CAF module comes from the multi-channel attention selection module in SelectionGAN [165]. However, there are three differences: (1) We use two generation branches to generate intermediate results, i.e., SA branch and AS branch. (2) Attention maps are generated by the combination of both shape and appearance features, so the model learns more correlations between the two features. (3) We also produce the input attention map, which aims to select useful content from the input image for generating the final image.

We consider two directions to generate intermediate results. One is generating multiple intermediate image synthesis results from the final appearance code F_T^I, and the other is generating multiple intermediate image synthesis results from the final shape code F_T^P. Specifically, the appearance code F_T^I is fed into a decoder to generate N intermediate results $I^I = \{I_i^I\}_{i=1}^N$, and followed by a Tanh activation function. Meanwhile, the final shape code F_T^P is fed into another decoder to generate another N

intermediate results I^P=$\{I_i^P\}_{i=1}^N$, and also followed by a Tanh activation function. Both can be formulated as,

$$\begin{aligned} I_i^I &= \text{Tanh}(F_T^I W_i^I + b_i^I), \qquad \text{for } i = 1, \cdots, N \\ I_i^P &= \text{Tanh}(F_T^P W_i^P + b_i^P), \qquad \text{for } i = 1, \cdots, N \end{aligned} \tag{4.5}$$

where two convolution operations are performed with N convolutional filters $\{W_i^I, b_i^I\}_{i=1}^N$ and $\{W_i^P, b_i^P\}_{i=1}^N$. Thus, the $2N$ intermediate results and the input image I_s can be regarded as the candidate image pool.

To generate the co-attention map which reflects the correlation between the appearance F_T^I and shape F_T^P codes, we first stack both F_T^I and F_T^P along the channel axes, and then feed them into a group of filters $\{W_i^A, b_i^A\}_{i=1}^{2N+1}$ to generate the corresponding $2N$+1 co-attention maps,

$$I_i^A = \text{Softmax}(\text{Concat}(F_T^I, F_T^P) W_i^A + b_i^A), \quad \text{for } i = 1, \cdots, 2N{+}1 \tag{4.6}$$

where Softmax is a channel-wise Softmax function used for the normalization, and Concat$(\cdot)$ denotes the channel-wise concatenation operation. Finally, the learned co-attention maps are used to perform a channel-wise selection from each intermediate generation and the input image as follows,

$$I_t^{'} = (I_1^A \otimes I_1^I) \oplus \cdots (I_{2N}^A \otimes I_{2N}^P) \oplus (I_{2N+1}^A \otimes I_s), \tag{4.7}$$

where $I_t^{'}$ represents the final synthesized person image selected from the multiple diverse results and the input image. $\otimes$ and $\oplus$ denote the element-wise multiplication and addition, respectively.

4.3.4 Optimization Objective

We use three different losses as our full optimization objective, i.e., adversarial loss $\mathcal{L}_{gan}$, pixel loss $\mathcal{L}_{l1}$, and perceptual loss $\mathcal{L}_p$,

$$\min_{G} \max_{D_I, D_P} \mathcal{L} = \lambda_{gan}\mathcal{L}_{gan} + \lambda_{l1}\mathcal{L}_{l1} + \lambda_p\mathcal{L}_p, \tag{4.8}$$

where λ_{gan}, λ_{l1} and λ_p are the weights, measuring corresponding contributions of each loss to the total loss $\mathcal{L}$. The total adversarial loss is derived from the appearance-guided discriminator D_I and the shape-guided discriminator D_P, which aims to judge how likely is that $I_t^{'}$ contains the same person in I_s and how well $I_t^{'}$ aligns with the target pose P_t, respectively. The $L1$ pixel loss is used to compute the difference between the generated image $I_t^{'}$ and the real target image I_t, i.e., $\mathcal{L}_{l1}$=$||I_t - I_t^{'}||_1$. The perceptual loss $\mathcal{L}_p$ is used to reduce pose distortions and make the generated images look more natural and smooth, i.e., $\mathcal{L}_p$=$||\phi(I_t) - \phi(I_t^{'})||_1$, where ϕ denotes the outputs of several layers in the pre-trained VGG19 network [148].

4.3.5 Implementation Details

We follow the training procedures of GANs and alternatively train the proposed Xing generator G and two discriminators (D_I, D_P). During training, G takes I_s, P_s and P_t as input and outputs a translated person image $I_t^{'}$ with target pose P_t. Specifically, I_s is fed to the SA branch, and P_s, P_t are fed to the AS branch. For the adversarial training, (I_s, I_t) and (I_s, $I_t^{'}$) are fed to the appearance-guided discriminator D_P for ensuring appearance consistency. (P_t, I_t) and (P_t, $I_t^{'}$) are fed to the shape-guided discriminator D_P for ensuring shape consistency.

The Adam optimizer [68] is used to train the proposed XingGAN for around 90K iterations with β_1=0.5 and β_2=0.999. We set T=9 in the proposed Xing generator and N=10 in the proposed co-attention fusion

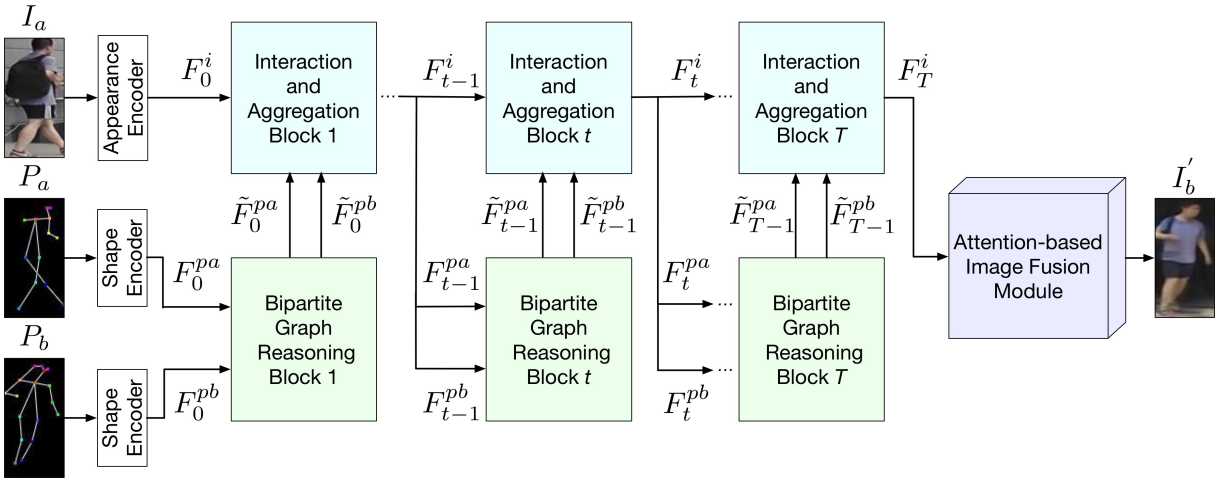


Figure 4.5: Overview of the proposed graph generator, which consists of a sequence of Bipartite Graph Reasoning (BGR) blocks, a sequence of Interaction-and-Aggregation (IA) blocks and an Attention-based Image Fusion (AIF) module. BGR blocks aim to reason the crossing long-range relations between the source pose and the target pose in a bipartite graph. IA blocks aim to interactively update person's appearance and shape feature representations. AIF module aims to selectively generate the final result via an attention network. The symbols $F^i=\{F_j^i\}_{j=0}^{T}$, $F^{pa}=\{F_j^{pa}\}_{j=0}^{T-1}$, $F^{pb}=\{F_j^{pb}\}_{j=0}^{T-1}$, $\tilde{F}^{pa}=\{\tilde{F}_j^{pa}\}_{j=0}^{T-1}$, and $\tilde{F}^{pb}=\{\tilde{F}_j^{pb}\}_{j=0}^{T-1}$ denote the appearance codes, the source shape codes, the target shape codes, the updated source shape codes, and the updated target shape codes, respectively.

module on both datasets. λ_{gan}, λ_{l1} and λ_p in Equation (4.8) are set to 5, 50 and 50, respectively. For the decoders, the kernel size of convolutions for generating the intermediate images and co-attention maps are 3×3 and 1×1, respectively.

4.4 Model Description

We start by introducing the details of the proposed Bipartite Graph Reasoning GAN (BiGraphGAN), which consists of a graph generator G and two discriminators (i.e., appearance discriminator D_a and shape discriminator D_s). An illustration of the proposed graph generator G is shown in Figure 4.5, which mainly contains three parts, i.e., a sequence of Bipartite Graph Reasoning (BGR) blocks modeling the crossing long-range relations between the source pose P_a and the target pose P_b, a sequence

of Interaction-and-Aggregation (IA) blocks interactively enhancing both person's shape and appearance feature representations, and an Attention-based Image Fusion (AIF) module attentively generating the final result $I_b^{'}$. In the following, we first present the proposed blocks and then introduce the optimization objective and implementation details of the proposed Bi-GraphGAN.

Figure 4.5 shows the proposed graph generator G, whose inputs are the source image I_a, the source pose P_a and the target pose P_b. The generator G aims to transfer the pose of the person in the source image I_a from the source pose P_a to the target pose P_b, generating the desired image $I_b^{'}$. Firstly, I_a, P_a and P_b are separately fed into three encoders to obtain the appearance code F_0^i, the source shape code F_0^{pa} and the target shape code F_0^{pb}. Note that we used the same shape encoder to learn both P_a and P_b, i.e., the two shape encoders for learning the two different poses are sharing the weights.

4.4.1 Pose-to-Pose Bipartite Graph Reasoning

The proposed Bipartite Graph Reasoning (BGR) block aims to reason the crossing long-range relations between the source pose and the target pose in a bipartite graph. All BGR blocks have an identical structure as illustrated in Figure 4.5. Consider the t-th block given in Figure 4.6, whose inputs are the source shape code F_{t-1}^{pa} and the target shape code F_{t-1}^{pb}. The BGR block aims to reason these two codes in a bipartite graph via Graph Convolutional Networks (GCNs) and outputs new shape codes. The proposed BGR block contains two symmetrical branches (i.e., B2A branch and A2B branch) because a bipartite graph is a bidirectional graph. As shown in Figure 4.2(c), each node in the source nodes connects all the target nodes; at the same time, each node in the target nodes connects all the source nodes. In the following, we mainly describe the detailed

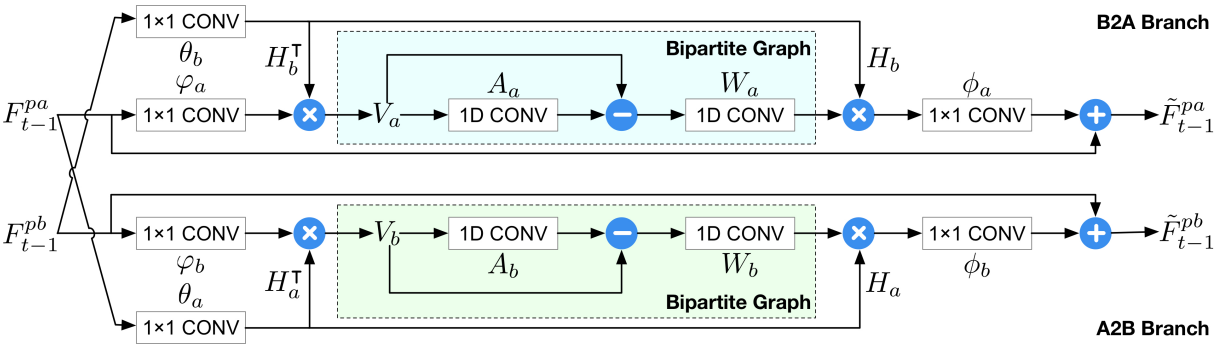


Figure 4.6: Illustration of the proposed Bipartite Graph Reasoning (BGR) Block t, which consists of two branches, i.e., B2A and A2B. Each of them aims to model cross-contextual information between shape features F_{t-1}^{pa} and F_{t-1}^{pb} in a bipartite graph via Graph Convolutional Networks (GCNs).

modeling process of the B2A branch, and another A2B branch is similar to this.

From Coordinate Space to Bipartite-Graph Space. Firstly, we reduce the dimension of the source shape code F_{t-1}^{pa} with function $\varphi_a(F_{t-1}^{pa}) \in \mathbb{R}^{C \times D_a}$, where C is the number of feature map channels, D_a is the number of nodes of F_{t-1}^{pa}. Then we reduce the dimension of the target shape code F_{t-1}^{pb} with function $\theta_b(F_{t-1}^{pb}) = H_b^\intercal \in \mathbb{R}^{D_b \times C}$, where D_b is the number of nodes of F_{t-1}^{pb}. Next, we project F_{t-1}^{pa} to a new feature V_a in a bipartite graph using the projection function H_b^T. Therefore we have,

$$V_a = H_b^\intercal \varphi_a(F_{t-1}^{pa}) = \theta_b(F_{t-1}^{pb}) \varphi_a(F_{t-1}^{pa}), \tag{4.9}$$

where both functions $\theta_b(\cdot)$ and $\varphi_a(\cdot)$ are implemented using 1×1 convolutional layer. This results in a new feature $V_a \in \mathbb{R}^{D_b \times D_a}$ in the bipartite graph, which represents the crossing relations between the nodes of the target pose F_{t-1}^{pb} and the source pose F_{t-1}^{pa} (see Figure 4.2(c)).

Cross Reasoning with Graph Convolution. After projection, we build a fully-connected bipartite graph with adjacency matrix $A_a \in \mathbb{R}^{D_b \times D_b}$. We then use a graph convolution to reason the crossing long-range relations between the nodes from both source and target poses, which can be for-

mulated as,

$$M_a = (\mathrm{I} - A_a)V_a W_a, \tag{4.10}$$

where $W_a \in \mathbb{R}^{D_a \times D_a}$ denotes the trainable edge weights. We follow [21, 218] and use Laplacian smoothing [21, 85] to propagate the node features over the bipartite graph. The identity matrix I can be viewed as a residual sum connection to alleviate optimization difficulties. Note that we randomly initialize both adjacency matrix A_a and the weights W_a, and then train both by gradient descent in an end-to-end manner.

From Bipartite-Graph Space to Coordinate Space. After the cross-reasoning, the updated new feature M_a is mapped back to the original coordinate space for further processing. Next, we add the result to the original source shape code F_{t-1}^{pa} to form a residual connection [47]. This process can be expressed as,

$$\tilde{F}_{t-1}^{pa} = \phi_a(H_b M_a) + F_{t-1}^{pa}, \tag{4.11}$$

where we reuse the projection matrix H_b and perform a linear projection $\phi_a(\cdot)$ to project M_a back to the original coordinate space. Therefore, we obtain the new source feature $\tilde{F}_{t-1}^{pa}$, which has the same dimension with the original one F_{t-1}^{pa}.

Similarly, the A2B branch outputs the new target shape feature $\tilde{F}_{t-1}^{pb}$. Note that the idea of the proposed BGR block is inspired by the GloRe unit proposed by [21]. The main difference is that the GloRe unit reasons the relations within the same feature map via a standard graph, but the proposed BGR block reasons the crossing relations between feature maps of different poses using a bipartite graph.

4.4.2 Pose-to-Image Interaction and Aggregation

As shown in Figure 4.5, the proposed Interaction-and-Aggregation (IA) block receives the appearance code F_{t-1}^{i}, the new source shape code $\tilde{F}_{t-1}^{pa}$ and the new target shape code $\tilde{F}_{t-1}^{pb}$ as inputs. IA block aims to simultaneously and interactively enhance F_t^i, F_t^{pa} and F_t^{pb}. Specifically, both shape codes firstly concatenated and fed into two convolutional layers to produce the attention map A_p. Mathematically,

$$A_p = \sigma(\text{Conv}(\text{Concat}(\tilde{F}_{t-1}^{pa}, \tilde{F}_{t-1}^{pb}))), \tag{4.12}$$

where $\sigma(\cdot)$ denotes the element-wise Sigmoid function.

Appearance Code Enhance. After obtaining A_p, the appearance F_{t-1}^{i} is enhanced by,

$$F_t^i = A_p \otimes F_{t-1}^{i} + F_{t-1}^{i}, \tag{4.13}$$

where $\otimes$ denotes element-wise product. By multiplying with the attention map A_p, the new appearance code F_t^i at certain locations can be either preserved or suppressed.

Shape Code Enhance. Next, we concatenate F_t^i, F_{t-1}^{pa} and F_{t-1}^{pb}, and go through two convolutional layers to obtain the updated shape code F_t^{pa} and F_t^{pb} by splitting the result along the channel axis. This process can be performed by,

$$F_t^{pa}, F_t^{pb} = \text{Conv}(\text{Concat}(F_t^i, F_{t-1}^{pa}, F_{t-1}^{pb})). \tag{4.14}$$

In this way, both new shape codes F_t^{pa} and F_t^{pb} can synchronize the changes caused by the new appearance code F_t^i.

4.4.3 Attention-Based Image Fusion

At the T-th IA block, we obtain the final appearance code F_T^i. We then feed F_T^i to an image decoder to generate the intermediate result $\tilde{I}_b$. At the same time, we feed F_T^i to an attention decoder to produce the attention mask A_i.

The attention encoder consists of several deconvolutional layers and a Sigmoid activation layer. Thus, the attention encoder aims to generate a one-channel attention mask A_i, in which each pixel value is between 0 to 1. The attention mask A_i aims to selectively pick useful content from both the input image I_a and the intermediate result $\tilde{I}_b$ for generating the final result $I_b^{'}$. This process can be expressed as,

$$I_b^{'} = I_a \otimes A_i + \tilde{I}_b \otimes (1 - A_i), \tag{4.15}$$

where $\otimes$ denotes element-wise product. In this way, both the image decoder and the attention decoder can interact with each other and ultimately produce better results.

4.4.4 Optimization Objective

Appearance and Shape Discriminators. We adopt two discriminators for adversarial training. Specifically, we feed image-image pair (I_a, I_b) and (I_a, $I_b^{'}$) into the appearance discriminator D_a to ensure appearance consistency. Meanwhile, we feed pose-image pair (P_b, I_b) and (P_b, $I_b^{'}$) into the shape discriminator D_s for shape consistency. Both discriminators (i.e., D_a and D_s), and the proposed graph generator G are trained in an end-to-end way, aiming to enjoy mutual benefits from each other in a joint framework.

Optimization Objectives. We follow [234, 156] and use the adversarial loss $\mathcal{L}_{gan}$, the pixel-wise $L1$ loss $\mathcal{L}_{l1}$ and the perceptual loss $\mathcal{L}_{per}$ as our

optimization objectives,

$$\mathcal{L}_{full} = \lambda_{gan}\mathcal{L}_{gan} + \lambda_{l1}\mathcal{L}_{l1} + \lambda_{per}\mathcal{L}_{per}, \tag{4.16}$$

where λ_{gan}, λ_{l1} and λ_{per} control the relative importance of the three objectives. For the perception loss, we follow [234, 156] and use the $Conv1\_2$ layer.

4.4.5 Implementation Details

In our experiments, we follow previous work [234, 156] and represent the source pose P_a and the target pose P_b as two 18-channel heat maps that encode the locations of 18 joints of a human body. Adam optimizer [68] is employed to learn the proposed BiGraphGAN for around 90K iterations with β_1=0.5 and β_2=0.999.

In preliminary experiments, we found that as T increases, the performance is getting better and better. When T is equal to 9, the proposed model achieves the best results, and then the performance begins to decline. Thus we set T=9 in the proposed graph generator. Moreover, λ_{gan}, λ_{l1}, λ_{per} in Equation (4.16), and the number of feature map channels C are set to 5, 10, 10, and 128, respectively. The proposed BiGraphGAN is implemented in PyTorch [120].

4.5 XingGAN Experiments

Datasets. We follow [98, 146, 234] and conduct experiments on two challenging datasets, i.e., Market-1501 [224] and DeepFashion [97]. Images on Market-1501 and DeepFashion are rescaled to 128×64 and 256×256, respectively. To generate human skeletons as training data, we employ OpenPose [13] to extract human joints. In this way, both P_s and P_t con-

Method	Market-1501					DeepFashion		
	SSIM	IS	Mask-SSIM	Mask-IS	PCKh	SSIM	IS	PCKh
PG2 [98]	0.253	3.460	0.792	3.435	-	0.762	3.090	-
DPIG [99]	0.099	3.483	0.614	3.491	-	0.614	3.228	-
PoseGAN [146]	0.290	3.185	0.805	3.502	-	0.756	3.439	-
C2GAN [164]	0.282	3.349	0.811	3.510	-	-	-	-
BTF [2]	-	-	-	-	-	0.767	3.220	-
PG2* [98]	0.261	3.495	0.782	3.367	0.73	0.773	3.163	0.89
PoseGAN* [146]	0.291	3.230	0.807	3.502	**0.94**	0.760	3.362	0.94
VUnet* [36]	0.266	2.965	0.793	3.549	0.92	0.763	3.440	0.93
Pose-Transfer* [234]	0.311	3.323	0.811	3.773	**0.94**	0.773	3.209	**0.96**
XingGAN (Ours)	**0.313**	**3.506**	**0.816**	**3.872**	0.93	**0.778**	**3.476**	0.95
Real Data	1.000	3.890	1.000	3.706	1.00	1.000	4.053	1.00

Table 4.1: Quantitative results on Market-1501 and DeepFashion. For all metrics, higher is better. (∗) denotes the results tested on our testing set.

sist of an 18-channel heat map encoding the positions of 18 joints of a human body. We also filter out images where no human is detected. Thus, we collect 101,966 training pairs and 8,570 testing pairs on DeepFashion. For Market-1501, we have 263,632 training and 12,000 testing pairs. Note that to better evaluate the proposed XingGAN, the person identities of the training set do not overlap with those of the testing set.

Evaluation Metrics. We follow [98, 146, 234] and adopt Structure Similarity (SSIM) [191], Inception Score (IS) [139], and their masked versions, i.e., Mask-SSIM and Mask-IS, as the evaluation metrics. Moreover, we adopt the PCKh score proposed in [234] to explicitly assess the shape consistency.

4.5.1 State-of-the-Art Comparisons

Quantitative Comparisons. We compare the proposed XingGAN with several leading person image generation methods, i.e., PG2 [98], DPIG [99], VUnet [36], PoseGAN [146], C2GAN [164], BTF [2] and Pose-Transfer [234]. Quantitative results measured by SSIM, IS, Mask-SSIM, Mask-IS, and PCKh metrics are shown in Table 4.1. Note that previous works [98,

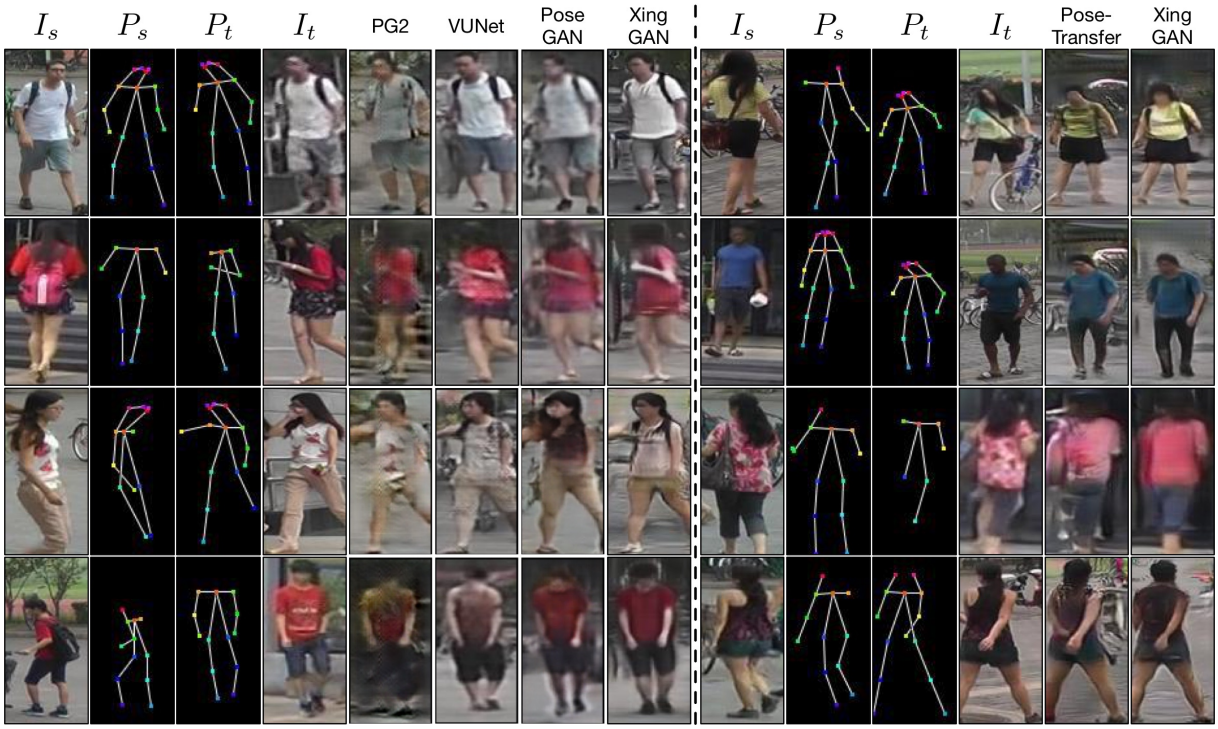


Figure 4.7: Qualitative comparison with PG2 [98], VUnet [36], PoseGAN [146] and Pose-Transfer [234] on Market-1501.

146] did not release the train/test split, thus we use their well-trained models and re-evaluate their performance on our testing set as in Pose-Transfer [234]. Although our testing set inevitably includes some of their training samples, the proposed XingGAN still achieves the best results in terms of SSIM, IS, Mask-SSIM, and Mask-IS metrics on both datasets. For the PCKh metric, [234] obtains slightly better results than XingGAN. However, we observe that the images generated by XingGAN are more realistic and have less visual artifacts than those generated by [234] (see Figures 4.7 and 4.8).

Qualitative Comparisons. Results compared with PG2 [98], VUnet [36] and PoseGAN [146] are shown on the left of Figures 4.7 and 4.8. We can see that the proposed XingGAN achieves much better results than PG2, VUnet, and PoseGAN on both datasets, especially at appearance details and the integrity of generated persons. Moreover, to evaluate the

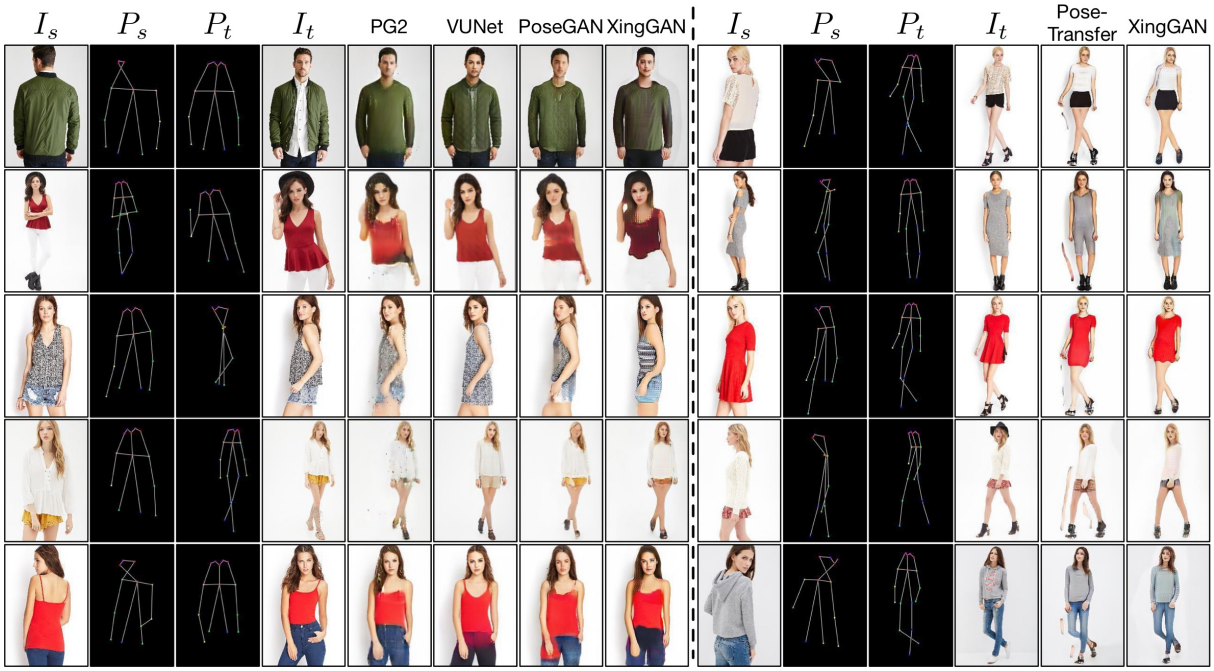


Figure 4.8: Qualitative comparison with PG2 [98], VUnet [36], PoseGAN [146] and Pose-Transfer [234] on DeepFashion.

effectiveness of XingGAN, we compare it with a stronger baseline, i.e., Pose-Transfer [234]. Results are shown on the right of Figures 4.7 and 4.8. We can see that XingGAN also generates much better person images having fewer visual artifacts than Pose-Transfer. For instance, Pose-Transfer [234] always generates a lot of visual artifacts in the background as shown in Figure 4.8.

Human Evaluation. We follow the evaluation protocol of [98, 146, 234] and recruited 30 volunteers to conduct a user study. Participants were shown a sequence of images and asked to give an instant judgment about each image within a second. Specifically, we randomly select 55 real and 55 fake images (generated by our model) and shuffle them. The first 10 of them are used for practice and the remaining 100 images are used for evaluation. Results compared with PG2 [98], PoseGAN [146], Pose-Transfer [234] and C2GAN [164] are shown in Table 4.2. We observe that the proposed XingGAN achieves the best results on all measurements compared with the

Method	Market-1501		DeepFashion	
	R2G	G2R	R2G	G2R
PG2 [98]	11.2	5.5	9.2	14.9
PoseGAN [146]	22.67	50.24	12.42	24.61
C2GAN [164]	23.20	46.70	-	-
Pose-Transfer [234]	32.23	63.47	19.14	31.78
XingGAN (Ours)	**35.28**	**65.16**	**21.61**	**33.75**

Table 4.2: User study of person image generation (%). R2G means the percentage of real images rated as generated w.r.t. all real images. G2R means the percentage of generated images rated as real w.r.t. all generated images. The results of other methods are reported from their papers.

leading methods, further validating that the generated images by our model are more sharp and photo-realistic.

4.5.2 Ablation Study

Variants of XingGAN. We conduct extensive ablation studies on Market-1501 [224] to evaluate different components of our XingGAN. XingGAN has four baselines as shown in Table 4.3: (i) 'SA' means only using the proposed Shape-guided Appearance-based generation branch. (ii) 'AS' means only adopting the proposed Appearance-guided Shape-based generation branch. (iii) 'SA+AS' combines both branches to produce the final person images. (iv) 'SA+AS+CAF' is our full model and employs the proposed Co-Attention Fusion module.

Effect of Dual-Branch Generation. The results of the ablation study are shown in Table 4.3. We see that the proposed SA branch achieves only 0.239 and 0.768 in SSIM and Mask-SSIM, respectively. When we only use the proposed AS branch, the values of SSIM and Mask-SSIM are improved to 0.286 and 0.798, respectively. Thus we conclude that the AS branch is more effective than the SA branch for generating photo-realistic person images. The AS branch takes the person poses as input and aims to learn person appearance representations, while the SA branch takes the

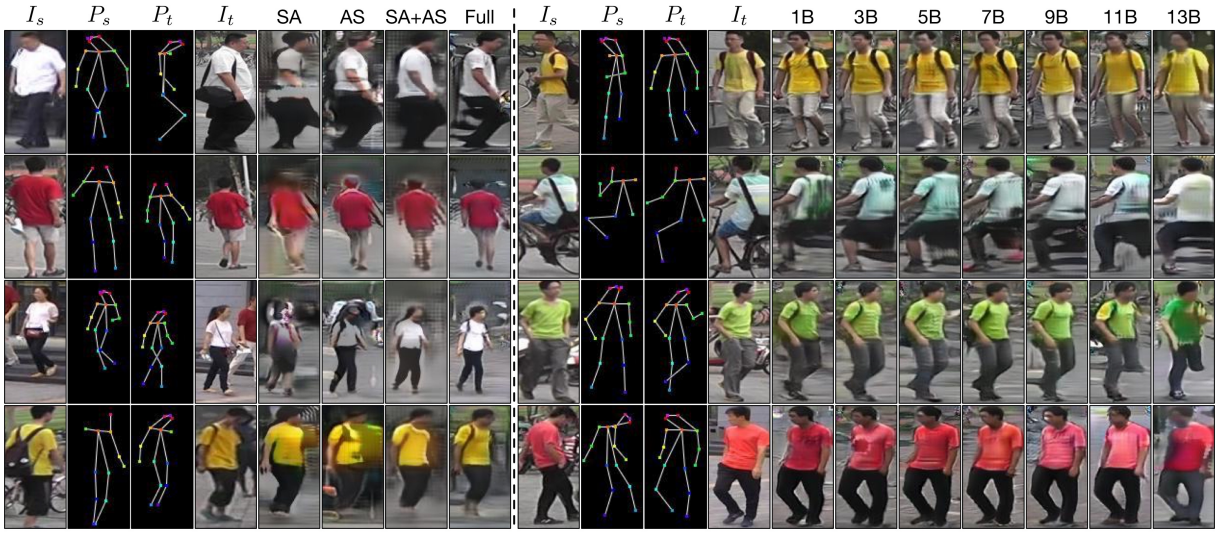


Figure 4.9: Ablation study of the proposed XingGAN on Market-1501. (left) Results of different variants of the proposed XingGAN. (right) Results of varying the number of the proposed Xing blocks. 'B' stands for the proposed Xing Blocks.

Variants of XingGAN	IS	Mask-IS	SSIM	Mask-SSIM
SA	**3.849**	3.645	0.239	0.768
AS	3.796	3.810	0.286	0.798
SA + AS	3.558	3.807	0.310	0.807
SA + AS + CAF (Full)	3.506	**3.872**	**0.313**	**0.816**

Table 4.3: Quantitative comparison of different variants of the proposed XingGAN on Market-1501. For all metrics, higher is better. 'SA', 'AS' and 'CAF' stand for the proposed SA branch, AS branch and co-attention fusion module, respectively.

person image as input and targets to learn person shape representations. Learning the appearance representations is much easier than learning the shape representations since there are shape deformations between the input person image and the desired person image, leading the AS branch to achieve better results than the SA branch.

Next, when adopting the combination of the proposed SA and AS branches, the performance in terms of SSIM and Mask-SSIM further boosts. However, the results in terms of IS and Mask-IS do not decline too much. Moreover, Figure 4.9 (left) shows some qualitative examples of the ablation study. We observe that the visualization results of 'SA', 'AS', and

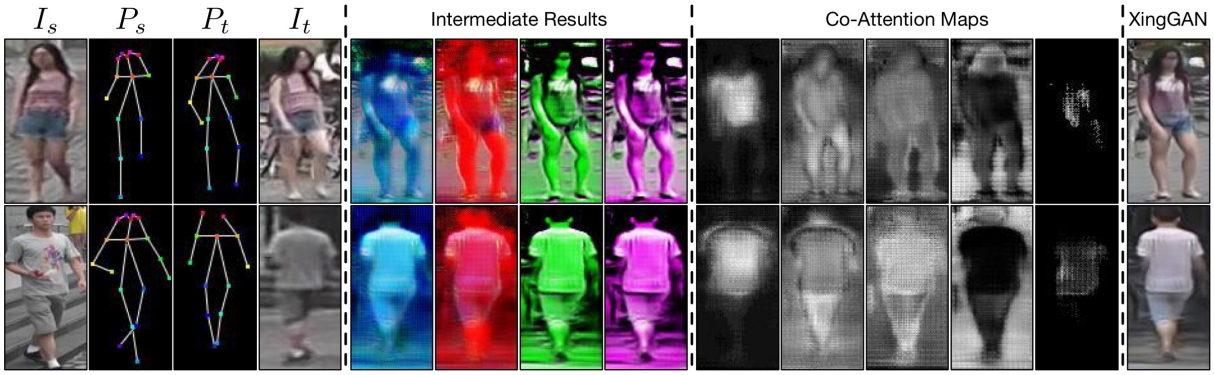


Figure 4.10: Visualization of intermediate results and co-attention maps generated by the proposed XingGAN on Market-1501. We randomly show four intermediate results, the corresponding four co-attention maps and the input attention map. Attention maps are normalized for better visualization.

'SA+AS' are consistent with the quantitative results. Therefore, both quantitative and qualitative results confirm the effectiveness of the proposed dual-branch generation strategy.

Effect of Co-Attention Fusion. 'SA+AS+CAF' outperforms the 'SA+AS' baseline with around 0.065, 0.003, and 0.009 gain on Mask-IS, SSIM, and Mask-SSIM, respectively. This means that the proposed co-attention fusion model indeed learns more correlations between the appearance and shape representations for generating the targeted person images, confirming our design motivation. Moreover, the proposed CAF module obviously improves the quality of the visualization results, as shown in the column 'Full' of Figure 4.9.

Lastly, we show the learned co-attention maps and the generated intermediate results. These co-attention maps are complementary, which could be qualitatively verified by visualizing the results in Figure 4.10. It is clear that they have learned different activated content between the generated intermediate results and the input image for generating the final person images.

Effect of The Xing Generator. The proposed Xing generator has two important network designs. One is the carefully designed Xing block, con-

Method	IS	Mask-IS	SSIM	Mask-SSIM
Xing Generator (1 blocks)	3.378	3.713	0.310	0.812
Xing Generator (3 blocks)	3.241	3.866	**0.316**	0.813
Xing Generator (5 blocks)	3.292	3.860	0.313	0.812
Xing Generator (7 blocks)	3.293	3.871	0.310	0.810
Xing Generator (9 blocks)	3.506	**3.872**	0.313	**0.816**
Xing Generator (11 blocks)	3.428	3.712	0.286	0.793
Xing Generator (13 blocks)	**3.708**	3.679	0.257	0.774
Resnet Generator (5 blocks)	3.236	3.807	0.297	0.802
Resnet Generator (9 blocks)	3.077	3.862	0.301	0.802
Resnet Generator (13 blocks)	3.134	3.731	0.300	0.797
PATN Generator (5 blocks)	3.273	3.870	0.309	0.809
PATN Generator (9 blocks)	3.323	3.773	0.311	0.811
PATN Generator (13 blocks)	3.274	3.797	0.314	0.808

Table 4.4: Quantitative comparison and ablation study of the proposed Xing generator on Market-1501. For all metrics, higher is better.

sisting of two sub-blocks, i.e., SA block and AS block. The Xing blocks jointly model both shape and appearance representations in a crossing way and enjoying the mutually improved benefits from each other. The other one is the cascaded network design, which deals with the complex and deformable translation problem progressively. Thus, we further conduct two experiments, one is to show the advantage of the progressive generation strategy by varying the number of the proposed Xing blocks, and the other is to explore the advantage of the Xing block by replacing it with the residual block [60] and PATB [234] resulting in two generators named Resnet generator and PATN generator in Table 4.4, respectively.

Quantitative and qualitative results are shown in Table 4.4 and Figure 4.9 (right). We observe that the proposed Xing generator with 9 blocks works the best. However, increasing the number of blocks further reduces generation performance. This could be attributed to the proposed Xing block. Only a few blocks are needed to capture the useful appearance and shape representations and the connection between them. Thus, we adopt 9 Xing blocks as default in our experiments for both datasets. Moreover, we

Method	Market-1501					DeepFashion		
	SSIM	IS	Mask-SSIM	Mask-IS	PCKh	SSIM	IS	PCKh
PG2 [98]	0.253	3.460	0.792	3.435	-	0.762	3.090	-
DPIG [99]	0.099	3.483	0.614	3.491	-	0.614	3.228	-
Deform [146]	0.290	3.185	0.805	3.502	-	0.756	3.439	-
C2GAN [164]	0.282	3.349	0.811	3.510	-	-	-	-
BTF [2]	-	-	-	-	-	0.767	3.220	-
PG2* [98]	0.261	**3.495**	0.782	3.367	0.73	0.773	3.163	0.89
Deform* [146]	0.291	3.230	0.807	3.502	**0.94**	0.760	3.362	0.94
VUnet* [36]	0.266	2.965	0.793	3.549	0.92	0.763	**3.440**	0.93
PATN* [234]	0.311	3.323	0.811	**3.773**	**0.94**	0.773	3.209	0.96
BiGraphGAN	**0.325**	3.329	**0.818**	3.695	**0.94**	**0.778**	3.430	**0.97**
Real Data	1.000	3.890	1.000	3.706	1.00	1.000	4.053	1.00

Table 4.5: Quantitative comparison of different methods on Market-1501 and DeepFashion. For all metrics, higher is better. (∗) denotes the results tested on our testing set.

see that the proposed Xing generator with only 5 Xing blocks outperforms both ResNet and PATN generators with 13 blocks on most metrics, which further certifies that our Xing generator has a good appearance and shape modeling capabilities with a very few blocks.

4.6 BiGraphGAN Experiments

Datasets. We follow previous works [98, 146, 234] and conduct extensive experiments on two public datasets, i.e., Market-1501 [224] and DeepFashion [97]. Specifically, we adopt the train/test split used in [234, 156] for a fair comparison. In addition, images are resized to 128×64 and 256×256 on Market-1501 and DeepFashion, respectively.

Evaluation Metrics. We follow [98, 146, 234] and employ Inception score (IS) [139], Structure Similarity (SSIM) [191] and their masked versions (i.e., Mask-IS and Mask-SSIM) as our evaluation metrics to quantitatively measure the quality of the generated images by different approaches. Moreover, we employ the PCKh score proposed in [234] to explicitly evaluate the shape consistency of the generated person images.

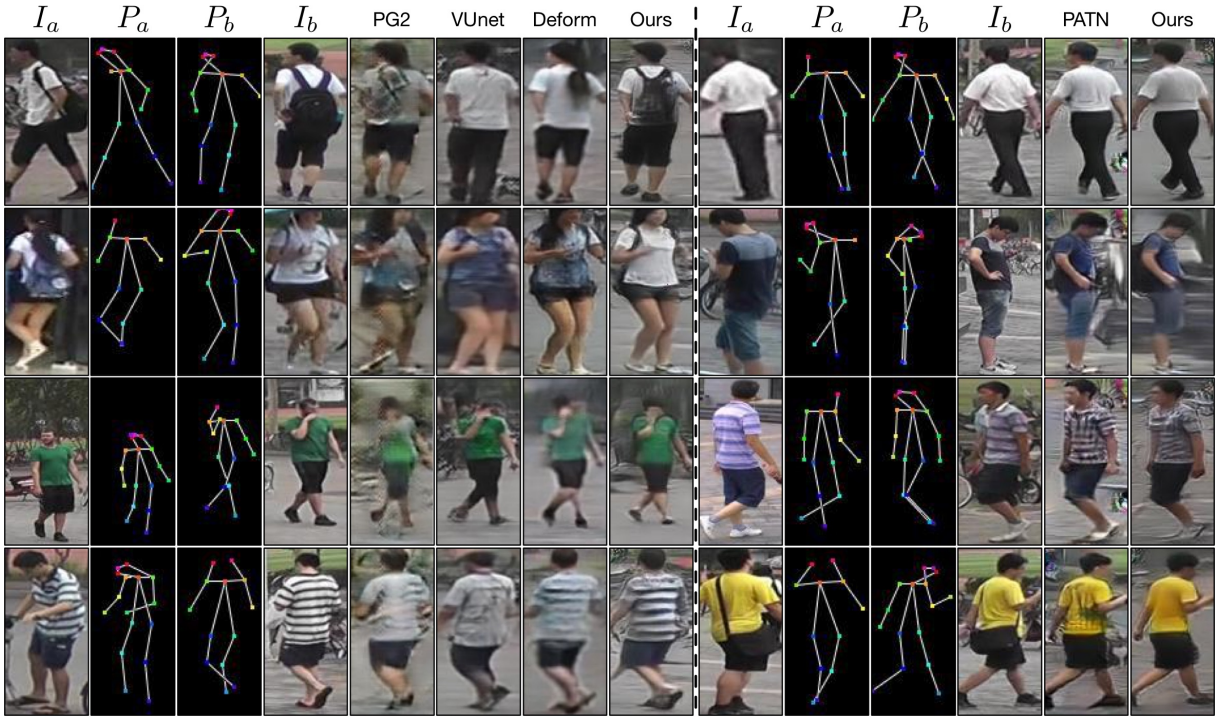


Figure 4.11: Qualitative comparisons of different methods on Market-1501.

4.6.1 State-of-the-Art Comparisons

Quantitative Comparisons. We compare BiGraphGAN with several leading person image synthesis methods, i.e., PG2 [98], DPIG [99], Deform [146, 145], C2GAN [164], BTF [2], VUnet [36], and PATN [234]. Quantitative comparison results are shown in Table 4.5, we can see that the proposed method achieves the best results on most metrics such as SSIM, Mask-SSIM and PCKh on Market-1501, and SSIM and PCKh on DeepFashion. For other metrics such as IS, the proposed method still achieves better results than the most related model PATN on both datasets. These results validate the effectiveness of our method.

Qualitative Comparisons. We also provide visualization comparison results on both datasets in Figures 4.11 and 4.12. As shown in the left of both figures, the proposed BiGraphGAN generates remarkably better results than PG2 [98], VUnet [36] and Deform [146] on both datasets.

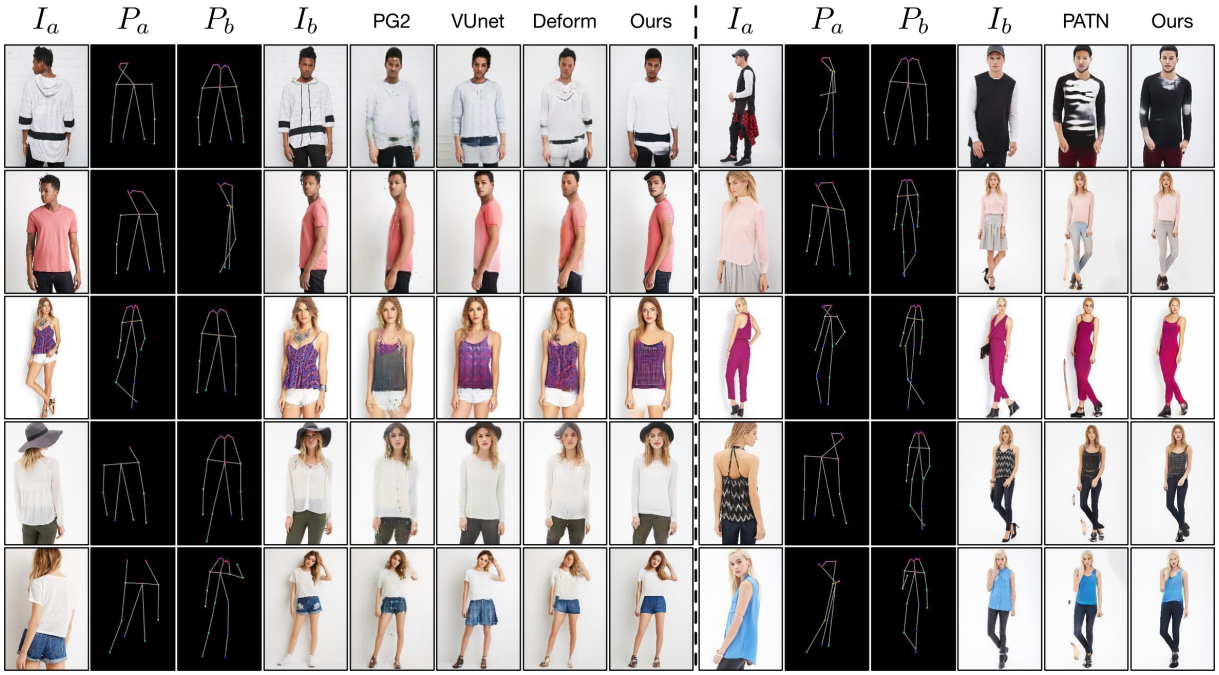


Figure 4.12: Qualitative comparisons of different methods on DeepFashion.

Method	Market-1501		DeepFashion	
	R2G	G2R	R2G	G2R
PG2 [98]	11.20	5.50	9.20	14.90
Deform [146]	22.67	50.24	12.42	24.61
C2GAN [164]	23.20	46.70	-	-
PATN [234]	32.23	63.47	19.14	31.78
BiGraphGAN	**35.76**	**65.91**	**22.39**	**34.16**

Table 4.6: Quantitative comparison of user study (%) on Market-1501 and DeepFashion. 'R2G' and 'G2R' represent the percentage of real images rated as fake w.r.t. all real images, and the percentage of generated images rated as real w.r.t. all generated images, respectively.

To further evaluate the effectiveness of the proposed method, we compare the proposed BiGraphGAN with the most state-of-the-art model, i.e., PATN [234], in the right of both figures. We still observe that our proposed BiGraphGAN generates more clear and visually plausible person images than PATN on both datasets.

User Study. We also follow [98, 146, 234] and conduct a user study to evaluate the quality of the generated images. Specifically, we follow the

Baselines of BiGraphGAN	SSIM ↑	Mask-SSIM ↑
B1: Our Baseline	0.305	0.804
B2: B1 + B2A	0.310	0.809
B3: B1 + A2B	0.310	0.808
B4: B1 + A2B + B2A (Sharing)	0.322	0.813
B5: B1 + A2B + B2A (Non-Sharing)	0.324	0.813
B6: B5 + AIF	**0.325**	**0.818**

Table 4.7: Ablation study of the proposed BiGraphGAN on Market-1501. For both metrics, higher is better.

evaluation protocol used in [234] for a fair comparison. Comparison results of different methods are shown in Table 4.6, we can see that the proposed method achieves the best results on all metrics, which further validates that the generated images by the proposed BiGraphGAN are more photo-realistic.

4.6.2 Ablation Study

Baselines of BiGraphGAN. We perform extensive ablation studies to validate the effectiveness of each component of the proposed BiGraphGAN on Market-1501. The proposed BiGraphGAN has 6 baselines (i.e., B1, B2, B3, B4, B5, B6) as shown in Table 4.7 and Figure 4.13 (left). B1 is our baseline. B2 uses the proposed B2A branch for modeling the crossing relations from the target pose to the source pose. B3 adopts the proposed A2B branch to model the crossing relations from the source pose to the target pose. B4 uses the combination of both A2B and B2A branches to model the crossing relations between the source pose and the target pose. Note that both GCNs in B4 are sharing the parameters. B5 employs a non-sharing strategy between the two GCNs to model the crossing relations. B6 employs the proposed AIF module to make the graph generator attentively select which part is more useful for generating the final person image.

Ablation Analysis. The results of the ablation study are shown in Table

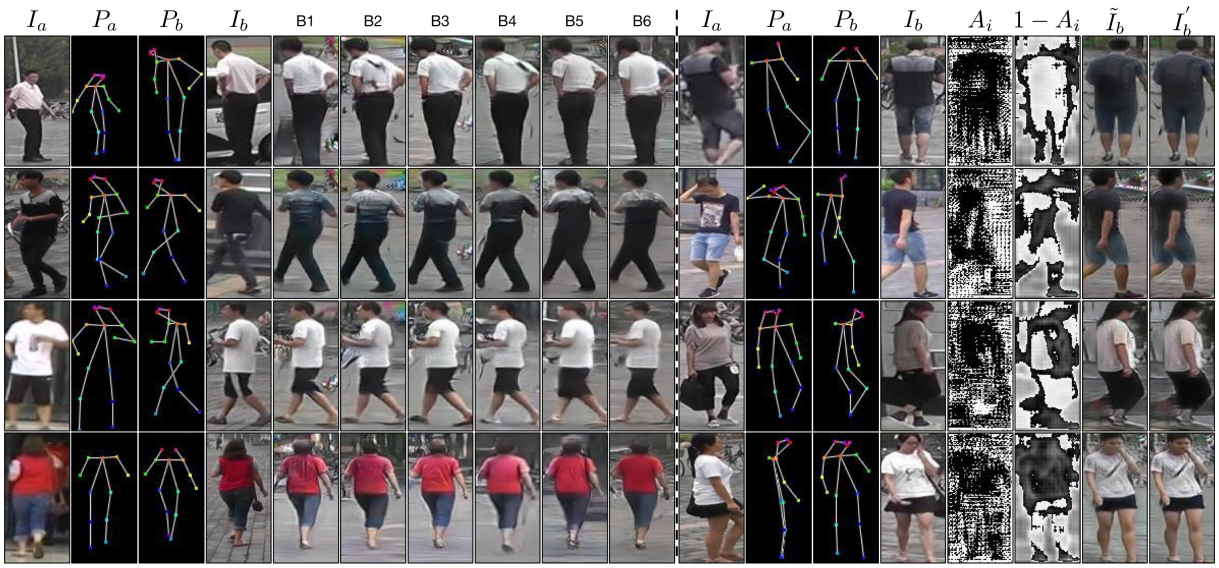


Figure 4.13: (left) Qualitative comparisons of ablation study on Market-1501. (right) Visualization of the learned attention masks and intermediate results.

4.7 and Figure 4.13 (left). We observe that both B2 and B3 achieve significantly better results than B1, which proves our initial motivation that modeling the crossing relations between the source pose and the target pose in a bipartite graph will boost the generation performance. In addition, we see that B4 performs better than B2 and B3, demonstrating the effectiveness of modeling the symmetric relations between the source and target poses. B5 achieves better results than B4, which means that two GCNs are constructed separately to model the symmetric relations will improve the generation performance in the joint network. B6 is better than B5, which clearly proves the effectiveness of the proposed attention-based image fusion strategy.

Moreover, we show several examples of the learned attention masks and intermediate results in Figure 4.13 (right) We can see that the proposed module attentively selects useful content from both the input image and intermediate result to generate the final result, thus verifying our design motivation.

4.7 Conclusion

We propose a novel XingGAN for the challenging person image generation task. It uses cascaded guidance with two different generation branches, and learns a deformable translation mapping from both person shape and appearance features. Moreover, we propose two novel blocks to effectively update person shape and appearance features in a crossing way. Extensive experiments based on human judgments and automatic evaluation metrics show that XingGAN achieves new state-of-the-art results on two challenging datasets. Lastly, we believe that the proposed blocks and the XingGAN framework can be easily extended to address other GAN-based generation and even multi-modality fusion tasks.

We also propose a novel Bipartite Graph Reasoning GAN (BiGraphGAN) framework for the challenging person image generation task. We introduce two novel blocks, i.e., Bipartite Graph Reasoning (BGR) block and Interaction-and-Aggregation (IA) block. The first is employed to model the crossing long-range relations between the source pose and the target pose in a bipartite graph. The second block is used to interactively enhance both person's shape and appearance features. Extensive experiments of both human judgments and automatic evaluation demonstrate that the proposed BiGraphGAN achieves remarkably better performance than the state-of-the-art approaches.

In next chapter, we will introduce a novel SelectionGAN for guided image-to-image translation task, which explores cascaded semantic guidance with a coarse-to-fine inference, and aims at producing a more detailed synthesis from richer and more diverse multiple intermediate generations.

Part II

Scene Image Generation

Chapter 5

SelectionGAN

We propose a novel model named Multi-Channel Attention Selection Generative Adversarial Network (SelectionGAN) for guided image-to-image translation, where we translate an input image into another while respecting an external semantic guidance. The proposed SelectionGAN explicitly utilizes the semantic guidance information and consists of two stages. In the first stage, the input image and the conditional semantic guidance are fed into a cycled semantic-guided generation network to produce initial coarse results. In the second stage, we refine the initial results by using the proposed multi-scale spatial pooling & channel selection module and the multi-channel attention selection module. Moreover, uncertainty maps automatically learned from attention maps are used to guide the pixel loss for better network optimization. Exhaustive experiments on four challenging guided image-to-image translation tasks (face, hand, body, and street view) demonstrate that our SelectionGAN is able to generate significantly better results than the state-of-the-art methods. Meanwhile, the proposed framework and modules are unified solutions and can be applied to solve other generation tasks such as semantic image synthesis. The source code and trained models are available at https://github.com/HaOTang/SelectionGAN.

5.1 Introduction

Guided image-to-image translation is a task aiming at synthesizing new images from an input image and several external semantic guidance. This task has been gaining a lot interest especially from the computer vision community, and has been widely investigated in recent years. Due to different forms of semantic guidance, e.g., segmentation maps, hand skeletons, facial landmarks, and pose skeleton, most of the existing methods for this class of tasks are tailored toward specific applications, i.e., they need to specifically design the network architectures and training objectives according to different generation tasks. For example, Ma et al. propose PG2 [98], which is a two-stage framework and uses the pose mask loss for generating person images based on an image of that person and human pose keypoints. Tang et al. propose GestureGAN [163], which is a forward-backward consistency architecture and adopt the proposed color loss to generate novel hand gesture images based on the input image and conditional hand skeletons. Wang et al. propose the few-shot Vid2Vid framework [183], which uses the carefully designed weight generation module to synthesize videos that realistically reflect the style of the input image and the layout of conditional segmentation maps.

Different from previous works in guided image-to-image translation, in this paper, we focus on developing a framework that is application-independent. This makes our framework and modules more widely applicable to many generation tasks with different forms of semantic guidance. To tackle this challenging problem, AlBahar and Huang [2] recently propose a bi-directional feature transformation to better utilize the constraints of the semantic guidance. Although this approach performs an interesting exploration, we observe unsatisfactory aspects mainly in the generated image layout and content details, which are due to three different reasons.

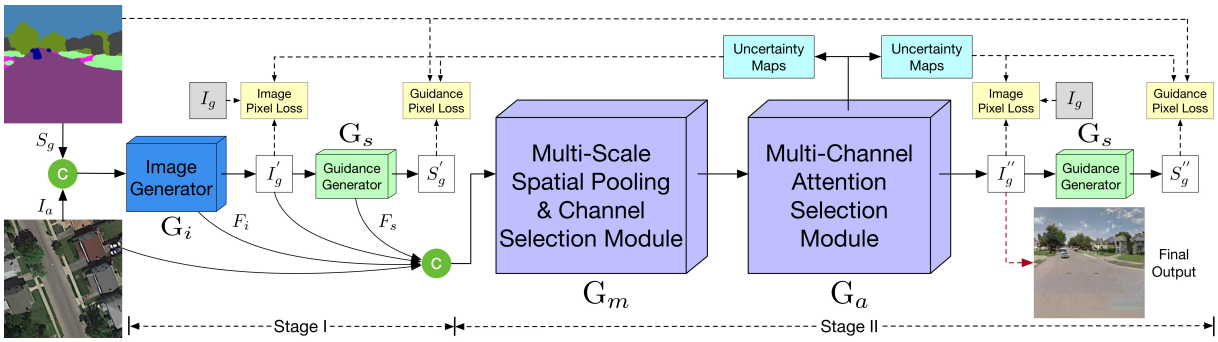


Figure 5.1: Overview of the proposed SelectionGAN. Stage I presents a cycled semantic-guided generation sub-network which accepts both the input image I_a and the conditional semantic guidance S_g, and simultaneously synthesizes the target image $I_g^{'}$ and reconstructs the semantic guidance $S_g^{'}$. Stage II takes the input image I_a, the coarse prediction $I_g^{'}$, and the learned deep features (F_i and F_s) from stage I, and performs a fine-grained generation using the proposed multi-scale spatial pooling & channel selection and the multi-channel attention selection modules. ⓒ denotes channel-wise concatenation.

First, since it is always costly to obtain manually annotated semantic guidance, the semantic guidance is usually produced from pre-trained models trained on other large-scale datasets, e.g., pose skeletons are extracted using OpenPose [4] and segmentation maps are extracted using [90, 226], leading to insufficiently accurate predictions for all the pixels, and thus misguiding the image generation process. Second, we argue that the translation with a single phase generation network is not able to capture the complex image structural relationships between the source and target domains, especially when source and target domains only have little or even no overlap, e.g., person image generation [98, 156], and cross-view image translation [165, 133]. Third, a three-channel generation space may not be suitable enough for learning a good mapping for this complex synthesis problem. Given these problems, could we enlarge the generation space and learn an automatic selection mechanism to synthesize more fine-grained generation results?

Based on these observations, in this paper, we propose a novel Multi-Channel Attention Selection Generative Adversarial Network (Selection-

GAN), which contains two generation stages. The overall framework of the proposed SelectionGAN is shown in Figure 5.1. In the first stage, we learn a cycled image-guidance generation sub-network, which accepts a pair consisting of an image and the conditional semantic guidance, and generates target images, which are further fed into a semantic guidance generation network to reconstruct the input semantic guidance. This cycled guidance generation adds stronger supervision between the image and guidance domains, facilitating the optimization of the network.

The coarse outputs from the first generation network, including the input image, together with the deep feature maps from the last layer, are input into the second stage networks. We first employ the proposed multi-scale spatial pooling & channel selection module to enhance the multi-scale features in both spatial and channel dimensions. Next, several intermediate outputs are produced, and simultaneously we learn a set of multi-channel attention maps with the same number as the intermediate generations. These attention maps are used to spatially select from the intermediate generations, and are combined to synthesize a final output. Finally, to overcome the inaccurate semantic guidance issue, the multi-channel attention maps are further used to generate uncertainty maps to guide the reconstruction loss. Through extensive experimental evaluations, we demonstrate that SelectionGAN produces remarkably better results than the existing baselines on four different guided image-to-image translation tasks, i.e., segmentation map guided cross-view image translation, hand skeleton guided gesture-to-gesture translation, facial landmark guided expression-to-expression translation, and pose guided person image generation. Moreover, the proposed framework and modules can be applied to other generation tasks such as semantic image synthesis.

Overall, the contributions of this paper are as follows:

- A novel Multi-Channel Attention Selection GAN (SelectionGAN) for

guided image-to-image translation task is presented. It explores cascaded semantic guidance with a coarse-to-fine inference, and aims at producing a more detailed synthesis from richer and more diverse multiple intermediate generations.

- A novel multi-scale spatial pooling & channel selection module is proposed, which is utilized to automatically enhance the multi-scale feature representation in both spatial and channel dimensions.
- A novel multi-channel attention selection module is proposed, which is utilized to attentively select interested intermediate generations and is able to significantly boost the quality of the final output. The multi-channel attention module also effectively learns uncertainty maps to guide the pixel loss for more robust optimization.
- Extensive experiments clearly demonstrate the effectiveness of the proposed SelectionGAN, and show state-of-the-art results on four guided image-to-image translation (including face, hand, body, and street view) tasks. Moreover, we show the proposed SelectionGAN is effective on other generation tasks such as semantic image synthesis.

Part of the material presented here appeared in [165]. The current paper extends [165] in several ways. (1) We present a more detailed analysis of related works by including recently published works dealing with guided image-to-image translation. (2) We propose a novel module, i.e., multi-scale channel selection, to automatically enhance the multi-scale feature representation in the feature channel dimension. Equipped with this new module, our SelectionGAN proposed in [165] is upgraded to SelectionGAN++. (3) We extent the proposed framework to a more robust and general framework for handling different guided image-to-image translation tasks. (4) We extend the quantitative and qualitative experiments by comparing our SelectionGAN and SelectionGAN++ with the very recent works on four guided image-to-image translation tasks and one semantic

image synthesis task with 11 public datasets.

5.2 Related Work

Generative Adversarial Networks (GANs) [41] have shown the capability of generating high-quality images [63]. A vanilla GAN model [41] has two important components: a generator G and a discriminator D. The goal of G is to generate photo-realistic images from a noise vector, while D is trying to distinguish between a real image and the image generated by G. Although it is successfully used in generating images of high visual fidelity, there are still some challenges, i.e., how to generate images in a conditional setting. To generate domain-specific images, Conditional GANs (CGANs) [103] have been proposed. One specific application of CGANs is image-to-image translation [55].

Image-to-Image Translation frameworks learn a parametric mapping between inputs and outputs. For example, Isola et al. [55] propose Pix2pix, which is a supervised model and uses a CGAN to learn a translation function from input to output image domains. Based on Pix2pix, Wang et al. [184] propose Pix2pixHD, which can turn semantic maps into photo-realistic images.

Our work builds upon the recent advances in image-to-image translation, i.e., Pix2pix, and aims to extend it to a broader set of guided image-to-image translation problem, which provides users with more input. Moreover, the proposed multi-scale spatial pooling & channel selection and the multi-channel attention selection modules are network-agnostic and can be plugged into any existing GAN-based generation architectures.

Guided Image-to-Image Translation is a variant of image-to-image translation problem aimed at translating an input image to a target image while respecting certain constrains specified by some external guidance,

such as class labels [25, 167, 162], text descriptions [78, 95], human keypoint/skeleton [163, 98, 159], segmentation maps [165, 133, 183, 154, 92, 169], and reference images [182, 2]. Given that different generation tasks need different guidance information, existing works are tailored to a specific application, i.e., with specifically designed network architectures and training objectives. For example, Ma et al. propose PG2 [98], which is a two-stage framework and uses the pose mask loss for generating person images based on an image of that person and human pose keypoints. Tang et al. propose GestureGAN [163], which is a forward-backward consistency architecture and adopt the proposed color loss to generate novel hand gesture images based on the input image and conditional hand skeletons. Wang et al. propose the few-shot Vid2Vid framework [183], which uses a carefully designed weight generation module to synthesize videos that realistically reflect the style of the input image and the layout of conditional segmentation maps.

Compared to existing works in guided image-to-image translation, we develop a unified and robust framework that is application-independent. In this way, the proposed framework can be widely applied to many generation tasks with different forms of guidance such as scene segmentation maps, hand skeletons, facial landmarks, and human body skeleton (see Figure 1.2).

Attention Learning in Image-to-Image Translation. Attention learning has been extensively exploited in computer vision and natural language processing, e.g., [196, 177, 29, 34, 33]. To improve the image generation performance, the attention mechanism has also been recently investigated in the image-to-image translation tasks [160, 66, 212, 166].

Unlike existing attention methods, we aim at a more effective network design and propose a novel SelectionGAN, which allows to automatically select from multiple diverse and rich intermediate generations, and thus sig-

nificantly improving the generation quality. To the best of our knowledge, our model is the first attempt to incorporate a multi-channel attention selection module within a GAN framework for image-to-image translation tasks.

5.3 Model Description

In this section we present the details of the proposed multi-channel attention selection GAN. An illustration of the overall network structure is depicted in Figure 5.1. In the first stage, we present a cascaded semantic-guided generation sub-network, which utilizes the input image and the conditional semantic guidance as inputs, and generate the target images while respecting the semantic guidance.

These generated images are further input into a semantic guidance generator to recover the input guidance forming a generation cycle. In the second stage, the coarse synthesis and the deep features from the first stage are combined, and then are passed to the proposed multi-scale spatial pooling & channel selection module to model the long-range multi-scale dependencies between each channel of feature representations. Thus the enhanced feature maps are fed to the proposed multi-channel attention selection module, which aims at producing more fine-grained synthesis from a larger generation space and also at generating uncertainty maps to jointly guide multiple optimization losses.

5.3.1 Cascade Semantic-Guided Generation

Semantic-Guided Generation. We target to translate an input image to another while respecting the semantic guidance. There are many strategies to incorporate the additional semantic guidance into the image-to-image translation model [2, 156] and the most straight forward scheme

is input concatenation. Specifically, as shown in Figure 5.1, we concatenate the input image I_a and the semantic guidance S_g, and feed them into the image generator G_i and synthesize the target image $I_g^{'}$ as $I_g^{'}$=$G_i(I_a, S_g)$. In this way, the semantic guidance provides stronger supervision to guide the image-to-image translation in the deep network.

Semantic-Guided Cycle. Existing guided image-to-image translation methods [98, 146, 2] only use semantic guidance as input to guide the image generation process, which actually provide a weak guidance. Different from theirs, we apply the semantic guidance not only as input but also as part of the network's output. Specifically, as shown in Figure 5.1, we propose a cycled semantic guidance generation network to benefit more the semantic guidance information in learning jointly. The conditional semantic guidance S_g together with the input image I_a are input into the image generator G_i, and produce the synthesized image $I_g^{'}$. Then $I_g^{'}$ is further fed into the semantic guidance generator G_s which reconstructs a new semantic guidance $S_g^{'}$. We can formalize the process as $S_g^{'}$=$G_s(I_g^{'})$=$G_s(G_i(I_a, S_g))$. Then the optimization objective is to make $S_g^{'}$ as close as possible to S_g, which naturally forms a semantic guidance generation cycle, i.e., $[I_a, S_g] \overset{G_i}{\rightarrow} I_g^{'} \overset{G_s}{\rightarrow} S_g^{'} \approx S_g$. The two generators are explicitly connected by the ground-truth semantic guidance, which in this way provides extra constraints on the generators to better learn the semantic structure consistency. We observe that the simultaneous generation of both the images and the semantic guidance improves the generation performance in our experiments section.

Cascade Generation. Due to the complexity of the tasks such as in pose guided person image generation [98, 155, 156], input and output domains usually have little overlap, which apparently leads to ambiguity issues in the generation process. Moreover, we observe that the image generator G_i outputs a coarse synthesis after the first stage, which yields blurred image

details and high pixel-level dissimilarity with the target images. Both inspire us to explore a coarse-to-fine generation strategy in order to boost the synthesis performance based on the coarse predictions. Cascade models have been used in several other computer vision tasks such as object detection [15] and semantic segmentation [27], and have shown great effectiveness. In this paper, we introduce the cascade strategy to deal with the guided image-to-image translation problems. In both stages we have a basic cycled semantic guidance generation sub-network, while in the second stage, we propose two novel multi-scale spatial pooling & channel selection and multi-channel attention selection modules to better utilize the coarse outputs from the first stage and to produce fine-grained final outputs. We observed significant improvement by using the proposed cascade strategy, illustrated in the experimental part.

5.3.2 Multi-Scale Spatial Pooling & Channel Selection

An overview of the proposed multi-scale spatial pooling & channel selection module is shown in Figure 5.2. The module consists of a multi-scale spatial pooling and a multi-scale channel selection components. In this way, the proposed module can learn multi-scale deep feature interdependencies in both spatial and channel dimensions.

Multi-Scale Spatial Pooling. Since there exists a large object/scene deformation between the source domain and the target domain, a single-scale feature may not be able to capture all the necessary spatial information for a fine-grained generation. Thus, we propose a multi-scale spatial pooling scheme, which uses a set of different kernel sizes and strides to perform a global average pooling on the same input features. By so doing, we obtain multi-scale features with different receptive fields to perceive different spatial contexts. More specifically, given the coarse inputs and the deep features produced from the stage I, we first concatenate all of them as new

features denoted as $\mathcal{F}_{\mathrm{c}} \in \mathbb{R}^{C \times H \times W}$ for the stage II as:

$$\mathcal{F}_{\mathrm{c}} = \mathrm{concat}(I_a, I_g^{'}, F_i, F_s), \tag{5.1}$$

where $\mathrm{concat}(\cdot)$ is a function for channel-wise concatenation operation; F_i and F_s are features from the last convolution layers of the generators G_i and G_s, respectively. H and W are width and height of the features, and C is the number of channels. We apply a set of M spatial scales $\{s_i\}_{i=1}^{M}$ in pooling, resulting in pooled features with different spatial resolution. Different from the pooling scheme used in [223] which directly combines all the features after pooling, we first select each pooled feature via an element-wise multiplication with the input feature. Since in our task the input features are from different sources, highly correlated features would preserve more useful information for the generation. Let us denote $\mathrm{pl\_up_s}(\cdot)$ as pooling at a scale s followed by an up-sampling operation to rescale the pooled feature at the same resolution, and $\otimes$ as element-wise multiplication, we can formalize the whole process as follows:

$$\mathcal{F}_{\mathrm{m}} \leftarrow \mathrm{concat}\big(\mathcal{F}_{\mathrm{c}}, \mathcal{F}_{\mathrm{c}} \otimes \mathrm{pl\_up}_1(\mathcal{F}_{\mathrm{c}}), \ldots, \mathcal{F}_{\mathrm{c}} \otimes \mathrm{pl\_up}_M(\mathcal{F}_{\mathrm{c}})\big), \tag{5.2}$$

which produces new multi-scale features $\mathcal{F}_{\mathrm{m}} \in \mathbb{R}^{4C \times H \times W}$ (we set M=3 in our experiments.) for the use in the next multi-scale channel selection module. By doing so, the 'level' of features can be enriched by combining multiple scale feature maps.

Multi-Scale Channel Selection. Each channel map of $\mathcal{F}_{\mathrm{m}}$ can be now regarded as a scale-specific response, and different scale feature maps should be associated with each other. To exploit the interdependencies between each scale features of $\mathcal{F}_{\mathrm{m}}$, we propose a multi-scale channel selection module to explicitly model interdependencies between channels of multi-scale feature $\mathcal{F}_{\mathrm{m}}$. The structure of multi-scale channel selection module is

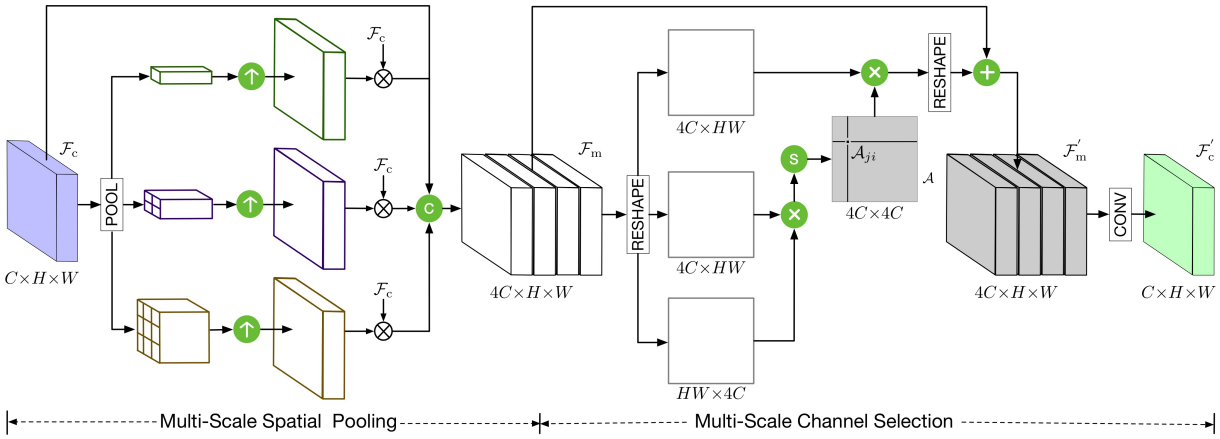


Figure 5.2: Overview of the proposed multi-scale spatial pooling & channel selection module. The multi-scale spatial pooling pools features from different receptive fields in order to have a better generation of image details. The multi-scale channel selection aims at automatically emphasizing interdependent channel maps by integrating associated features among all multi-scale channel maps to improve deep feature representation. $\oplus$, $\otimes$, ⓒ, ⓢ and ⓣ denote element-wise addition, element-wise multiplication, channel-wise concatenation, softmax, and up-sampling operation, respectively.

illustrated in Figure 5.2.

The channel attention map $\mathcal{A}$ can be obtained from the multi-scale feature $\mathcal{F}_{\mathrm{m}}$. More specific, $\mathcal{F}_{\mathrm{m}}$ is first reshaped to $\mathbb{R}^{4C\times HW}$, and then a matrix multiplication is preformed between $\mathcal{F}_{\mathrm{m}}$ and the transpose of $\mathcal{F}_{\mathrm{m}}$. Next, we employ a Softmax activation function to obtain the channel attention map $\mathcal{A}\in\mathbb{R}^{4C\times 4C}$. Each pixel $\mathcal{A}_{ji}$ in $\mathcal{A}$ measures the i^{th} channel's impact on the j^{th} channel. In this way, the correlation can be built between features from different scales. Moreover, to reshape back to $\mathbb{R}^{4C\times H\times W}$, we perform a matrix multiplication between $\mathcal{A}$ and the transpose of $\mathcal{F}_{\mathrm{m}}$. Then, the result is multiplied by a parameter α and added to the original feature $\mathcal{F}_{\mathrm{m}}$ to obtain the channel-wise enhanced feature $\mathcal{F}'_{\mathrm{m}}\in\mathbb{R}^{4C\times H\times W}$,

$$\mathcal{F}'_{\mathrm{m}} = \alpha\sum_{i=1}^{4C}(\mathcal{A}_{ji}\mathcal{F}_{\mathrm{m}i}) + \mathcal{F}_{\mathrm{m}j}. \tag{5.3}$$

By doing so, each channel in the final feature $\mathcal{F}'_{\mathrm{m}}$ is a weighted sum of all

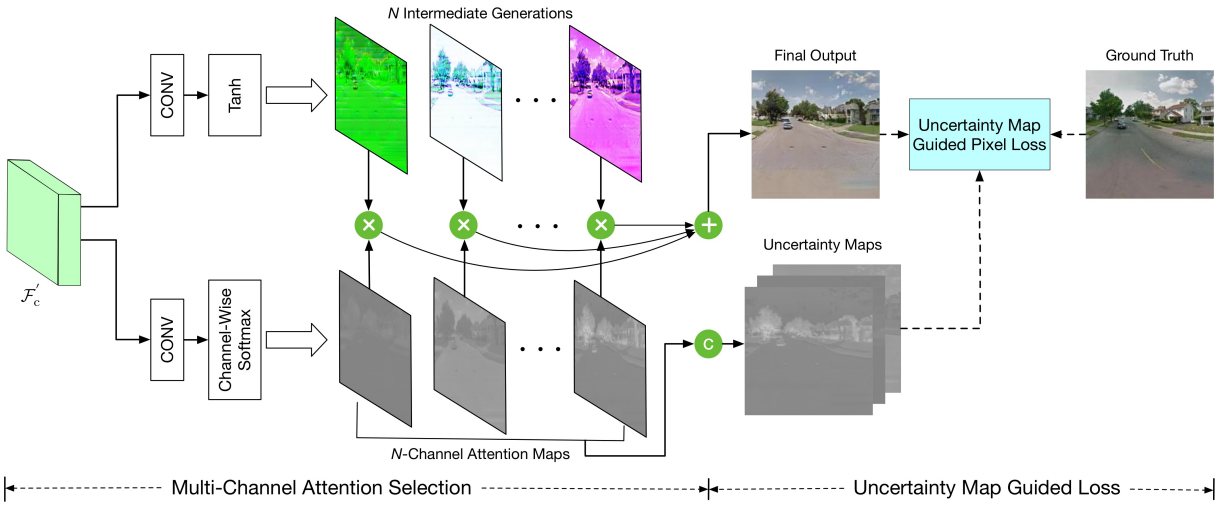


Figure 5.3: Overview of the proposed multi-channel attention selection module. This module aims to automatically select from a set of intermediate diverse generations in a larger generation space to improve the generation quality. Meanwhile, the module also effectively learns uncertainty maps to guide the pixel loss for robust joint images and guidances optimization. $\oplus$, $\otimes$ and ⓒ denote element-wise addition, element-wise multiplication, and channel-wise concatenation, respectively.

channels and it models the long-range dependencies between multi-scale feature maps. Finally, the enhanced feature $\mathcal{F}_{\mathrm{m}}^{'}$ is fed into a convolutional layer to obtain $\mathcal{F}_{\mathrm{c}}^{'} \in \mathbb{R}^{C \times H \times W}$, which has the same size as the original one $\mathcal{F}_{\mathrm{c}}$. This design ensures that the proposed multi-scale spatial pooling & channel selection module can be plugged into existing computer vision architectures.

5.3.3 Multi-Channel Attention Selection

In previous image-to-image translation works, the image was generated only in a three-channel RGB space. We argue that this is not enough for the complex translation problem we are dealing with, and thus we explore using a larger generation space to have a richer synthesis via constructing multiple intermediate generations. Accordingly, we design a multi-channel attention mechanism to automatically perform spatial and temporal selection from the generations to synthesize a fine-grained final output.

Given the enhanced multi-scale feature volume $\mathcal{F}_{\mathrm{c}}^{'} \in \mathbb{R}^{C \times H \times W}$, where H and W are width and height of the features, and C is the number of channels, we consider two directions as shown in Figure 5.3. One is for the generation of multiple intermediate image synthesis and the other is for the generation of multi-channel attention maps. To produce N different intermediate generations $I_G = \{I_G^i\}_{i=1}^N$, a convolution operation is performed with N convolutional filters $\{W_G^i, b_G^i\}_{i=1}^N$ followed by a $\tanh(\cdot)$ non-linear activation operation. For the generation of corresponding N attention maps, the other group of filters $\{W_A^i, b_A^i\}_{i=1}^N$ is applied. Then the intermediate generations and the attention maps are calculated as follows:

$$\begin{aligned} I_G^i &= \tanh(\mathcal{F}_{\mathrm{c}}^{'} W_G^i + b_G^i), \qquad \text{for} \quad i = 1, \ldots, N \\ I_A^i &= \mathrm{Softmax}(\mathcal{F}_{\mathrm{c}}^{'} W_A^i + b_A^i), \qquad \text{for} \quad i = 1, \ldots, N \end{aligned} \tag{5.4}$$

where $\mathrm{Softmax}(\cdot)$ is a channel-wise softmax function used for the normalization. Finally, the learned attention maps are utilized to perform channel-wise selection from each intermediate generation as follows:

$$I_g^{''} = (I_A^1 \otimes I_G^1) \oplus \cdots \oplus (I_A^N \otimes I_G^N) \tag{5.5}$$

where $I_g^{''}$ represents the final synthesized generation selected from the multiple diverse results, and $\oplus$ denotes the element-wise addition. We also generate a final semantic guidance in the second stage as in the first stage, i.e., $S_g^{''} = G_s(I_g^{''})$. Due to the same purpose of the two semantic guidance generators, we use a single G_s twice by sharing the parameters in both stages to reduce the network capacity.

Uncertainty-Guided Pixel Loss. As we discussed in the introduction, the semantic guidance obtained from the pretrained model is not accurate for all the pixels, leading to a wrong guidance during training. To tackle this issue, we propose to learn uncertainty maps to control the optimization

loss as shown in Figure 5.3. The uncertainty learning has been investigated in [64] for multi-task learning, and here we introduce it for solving the noisy semantic guidance problem. Assume that we have K different loss maps which need a guidance. The multiple generated attention maps are first concatenated and passed to a convolution layer with K filters $\{W_u^i\}_{i=1}^K$ to produce a set of K uncertainty maps. The reason for using the attention maps to generate uncertainty maps is that the attention maps directly affect the final generation leading to a close connection with the loss. Let $\mathcal{L}_p^i$ denote a pixel-level loss map and U_i denote the i-th uncertainty map, we have:

$$\begin{aligned} U_i &= \sigma\big(W_u^i(\text{concat}(I_A^1, \ldots, I_A^N) + b_u^i\big) \\ \mathcal{L}_p^i &\leftarrow \frac{\mathcal{L}_p^i}{U_i} + \log U_i, \quad \text{for} \quad i = 1, \ldots, K \end{aligned} \tag{5.6}$$

where $\sigma(\cdot)$ is a Sigmoid function for pixel-level normalization. The uncertainty map is automatically learned and acts as a weighting scheme to control the optimization loss.

Parameter-Sharing Discriminator. We extend the vanilla discriminator in [55] to a parameter-sharing structure. In the first stage, this structure takes the real image I_a and the generated image $I_g^{'}$ or the ground-truth image I_g as input. The discriminator D learns to tell whether a pair of images from different domains is associated with each other or not. In the second stage, it accepts the real image I_a and the generated image $I_g^{''}$ or the real image I_g as inputs. This pairwise input encourages D to discriminate the diversity of image structure and to capture the local-aware information.

5.3.4 Optimization Objective

Adversarial Loss. In the first stage, the adversarial loss of D for distinguishing synthesized image pairs $[I_a, I_g^{'}]$ from real image pairs $[I_a, I_g]$ is

formulated as follows:

$$\mathcal{L}_{cGAN}(I_a, I_g^{'}) = \mathbb{E}_{I_a,I_g}\left[\log D(I_a, I_g)\right] + \mathbb{E}_{I_a,I_g^{'}}\left[\log(1 - D(I_a, I_g^{'}))\right]. \quad (5.7)$$

In the second stage, the adversarial loss of D for distinguishing synthesized image pairs $[I_a, I_g^{''}]$ from real image pairs $[I_a, I_g]$ is formulated as follows:

$$\mathcal{L}_{cGAN}(I_a, I_g^{''}) = \mathbb{E}_{I_a,I_g}\left[\log D(I_a, I_g)\right] + \mathbb{E}_{I_a,I_g^{''}}\left[\log(1 - D(I_a, I_g^{''}))\right]. \quad (5.8)$$

Both losses aim to preserve the local structure information and produce visually pleasing synthesized images. Thus, the adversarial loss of the proposed SelectionGAN is the sum of Equations (5.7) and (5.8),

$$\mathcal{L}_{cGAN} = \mathcal{L}_{cGAN}(I_a, I_g^{'}) + \lambda\mathcal{L}_{cGAN}(I_a, I_g^{''}). \quad (5.9)$$

Overall Loss. The total optimization loss is a weighted sum of the above losses. Generators G_i, G_s, multi-scale spatial pooling & channel selection module G_m, multi-channel attention selection network G_a, and discriminator D are trained in an end-to-end fashion optimizing the following min-max function:

$$\min_{\{G_i,G_s,G_m,G_a\}} \max_{\{D\}} \mathcal{L} = \sum_{i=1}^{4} \lambda_i \mathcal{L}_p^i + \mathcal{L}_{cGAN} + \lambda_{tv}\mathcal{L}_{tv}. \quad (5.10)$$

where $\mathcal{L}_p^i$ uses the L1 reconstruction to separately calculate the pixel loss between the generated four images/guidances (i.e., $I_g^{'}$, $S_g^{'}$, $I_g^{''}$, and $S_g^{''}$) and the corresponding real images/guidances. $\mathcal{L}_{tv}$ is the total variation regularization [60] on the final synthesized image $I_g^{''}$. λ_i and λ_{tv} are the trade-off parameters to control the relative importance of different objectives. The training is performed by solving the min-max optimization problem.

5.3.5 Implementation Details

Network Architecture. For a fair comparison, we employ U-Net [55] as our generator architectures G_i and G_s. U-Net is a network with skip connections between a down-sampling encoder and an up-sampling decoder. Such architecture comprehensively retains contextual and textural information, which is crucial for removing artifacts and padding textures. Since our focus is on the image generation task, G_i is more important than G_s. Thus we use a deeper network for G_i and a shallow network for G_s. Specifically, the filters in first convolutional layer of G_i and G_s are 64 and 4, respectively. For the network G_a, the kernel size of convolutions for generating the intermediate images and attention maps are 3×3 and 1×1, respectively. We adopt PatchGAN [55] for the discriminator D.

Training Details. We mainly focus on four guided image-to-image translation tasks in this paper. For cross-view image translation, we follow [133] and use RefineNet [90] and [226] to generate segmentation maps on Dayton, SVA, and Ego2Top datasets as training data, respectively. For facial expression generation, we follow [164] and use OpenFace [4] to extract facial landmarks on Radboud Faces dataset as training data. For both hand gesture generation and human pose generation tasks, we follow [163, 98] and employ OpenPose [13] as pose joints detector and filter out images where no human hand and body are detected in the associated datasets.

We follow the optimization method in [41] to optimize the proposed SelectionGAN, i.e., one gradient descent step on discriminator and generators alternately. We first train G_i, G_s, G_m, G_a with D fixed, and then train D with G_i, G_s, G_m, G_a fixed. The proposed SelectionGAN is trained and optimized in an end-to-end fashion. We employ Adam [68] with momentum terms β_1=0.5 and β_2=0.999 as our solver. In our experiments, we set λ_{tv}=1e−6, λ_1=100, λ_2=1, λ_3=200 and λ_4=2 in Equation (5.10), and λ=4

in Equation (5.9). The number of attention channels N in Equation (5.4) is set to 10.

5.4 Experiments

We conduct extensive experiments on a variety of guided image-to-image translation tasks such as segmentation map guided cross-view image translation, facial landmark guided expression-to-expression translation, hand skeleton guided gesture-to-gesture translation, and pose skeleton guided person image generation. Moreover, to explore the generality of the proposed SelectionGAN on other generation tasks, we conduct experiments on the challenging semantic image synthesis task.

5.4.1 Cross-View Image Translation

Datasets. We follow [133, 134, 165] and perform experiments on four public cross-view image translation datasets: 1) The Dayton dataset [179] contains 76,048 images and the train/test split is 55,000/21,048. The images in the original dataset have 354×354 resolution. We resize them to 256×256. 2) The CVUSA dataset [192] consists of 35,532/8,884 image pairs in train/test split. Following [211, 133], the aerial images are center-cropped to 224×224 and resized to 256×256. For the ground level images and corresponding segmentation maps, we take the first quarter of both and resize them to 256×256. 3) The Surround Vehicle Awareness (SVA) dataset [116] is a synthetic dataset collected from Grand Theft Auto V (GTAV) video game. Following [134], we select every tenth frame to remove redundancy in this dataset since the consecutive frames in each set are very similar to each other. Thus, we collect 46,030/22,254 image pairs for training and testing, respectively. 4) The Ego2Top dataset [6] is more challenging and contains different indoor and outdoor conditions. Each

#	Setups of SelectionGAN	SSIM ↑	PSNR ↑	SD ↑	FID ↓	Inception Score ↑		
						All	Top-1	Top-5
A	$I_a \overset{G_i}{\rightarrow} I'_g$	0.4555	19.6574	18.8870	91.47	3.2359	2.1903	3.3110
B	$S_g \overset{G_i}{\rightarrow} I'_g$	0.5223	22.4961	19.2648	87.51	3.4849	2.2544	3.4217
C	$[I_a, S_g] \overset{G_i}{\rightarrow} I'_g$	0.5374	22.8345	19.2075	84.10	3.4478	2.2616	3.4668
D	$[I_a, S_g] \overset{G_i}{\rightarrow} I'_g \overset{G_s}{\rightarrow} S'_g$	0.5438	22.9773	19.4568	82.81	3.1655	2.2561	3.2401
E	D + Uncertainty-Guided Pixel Loss	0.5522	23.0317	19.5127	79.84	3.2741	2.2687	3.3063
F	E + Multi-Channel Attention Selection	0.5989	23.7562	20.0000	75.57	3.3365	**2.2749**	3.4664
G	F + Total Variation Regularization	0.6047	23.7956	20.0830	74.11	3.3172	2.1397	3.3509
H	G + Multi-Scale Spatial Pooling	**0.6167**	**23.9310**	**20.1214**	**72.23**	**3.4978**	2.1880	**3.4786**

Table 5.1: Ablations study of the proposed SelectionGAN.

case contains one top-view video and several egocentric videos captured by the people visible in the top-view camera. This dataset has more than 230,000 frames. For training data, we follow [165] and randomly select 386,357 pairs and each pair is composed of two images of the same scene but different viewpoints. We randomly select 25,600 pairs for evaluation.

Parameter Settings. For a fair comparison, we adopt the same training setup as in [55, 133]. All images are scaled to 256×256, and we enabled image flipping and random crops for data augmentation. Similar to [133], the experiments for Dayton are trained for 35 epochs with batch size of 4. For CVUSA, we follow the same setup as in [211, 133], and train our network for 30 epochs with batch size of 4. For Ego2Top, all models are trained with 10 epochs using batch size 8. For SVA, all models are trained with 20 epoch using batch size 4.

Evaluation Metrics. Similar to [133, 165], we employ Inception Score [139], top-k prediction accuracy, KL score, and Fréchet Inception Distance (FID) [48] for the quantitative analysis. These metrics evaluate the generated images from a high-level feature space. We also employ pixel-level similarity metrics to evaluate our method, i.e., Structural-Similarity (SSIM) [191], Peak Signal-to-Noise Ratio (PSNR), and Sharpness Difference (SD).

Baseline Models. We first conduct an ablation study on Dayton to evaluate the components of the proposed SelectionGAN. To reduce the training

#	Stage I	Stage II	SSIM	PSNR	SD
F	√		0.5551	23.1919	19.6311
F		√	**0.5989**	**23.7562**	**20.0000**
G	√		0.5680	23.2574	19.7371
G		√	**0.6047**	**23.7956**	**20.0830**
H	√		0.5567	23.1545	19.6034
H		√	**0.6167**	**23.9310**	**20.1214**

Table 5.2: Quantitative results of coarse-to-fine generation.

time, we randomly select 1/3 samples from the whole 55,000/21,048 samples, i.e., around 18,334 samples for training and 7,017 samples for testing. The proposed SelectionGAN considers eight baselines (A, B, C, D, E, F, G, H) as shown in Table 5.1. Baseline A uses a Pix2pix structure [55] and generates $I_g^{'}$ using a single image I_a. Baseline B uses the same Pix2pix model and generates $I_g^{'}$ using the corresponding semantic guidance S_g. Baseline C also uses the Pix2pix structure, and inputs the combination of a conditional image I_a and the semantic guidance S_g to the generator G_i. Baseline D uses the proposed cycled semantic guidance generation upon Baseline C. Baseline E represents the pixel loss guided by the learned uncertainty maps. Baseline F employs the proposed multi-channel attention selection module to generate multiple intermediate generations, and to make the neural network attentively select which part is more important for generating the target image. Baseline G adds the total variation regularization on the final result $I_g^{''}$. Baseline H employs the proposed multi-scale spatial pooling module to refine the features $\mathcal{F}_c$ from stage I. All the baseline models are trained and tested on the same data using the configuration.

Ablation Analysis. The results of the ablation study are shown in Table 5.1. We observe that Baseline B is better than baseline A since S_g contains more structural information than I_a. By comparison Baseline A with Baseline C, the semantic-guided generation improves SSIM, PSNR and SD by 8.19, 3.1771 and 0.3205, respectively, which confirms the importance

of the conditional semantic guidance information. By using the proposed cycled semantic guidance generation, Baseline D further improves over C, meaning that the proposed semantic guidance cycle structure indeed utilizes the semantic guidance information in a more effective way, confirming our design motivation. Baseline E outperforms D showing the importance of using the uncertainty maps to guide the pixel loss map which contains an inaccurate reconstruction loss due to the wrong semantic guidance produced from the pre-trained models. Baseline F significantly outperforms E with around 4.67 points gain on the SSIM metric, clearly demonstrating the effectiveness of the proposed multi-channel attention selection scheme. We can also observe from Table 5.1 that, by adding the proposed multi-scale spatial pool scheme and the TV regularization, the overall performance is further boosted. Finally, we demonstrate the advantage of the proposed two-stage strategy over the one-stage method. The results are shown in Figures 5.4, 5.12, and Table 5.2. It is obvious that the coarse-to-fine generation model is able to generate sharper results and contains more details than the one-stage model, which further confirms our motivations.

Comparisons with SENet. The proposed multi-scale spatial pooling shares a similar intuition with SENet [49] which amplifies the channels via attention based on pooling. Unlike SENet that employs positive attention via the Sigmoid function, the proposed multi-scale spatial pooling selects each pooled feature via an element-wise multiplication with the original feature. Since in our task the input features are from different sources, highly correlated features would preserve more useful information for the generation. We also conduct experiments to compare the proposed method with SENet on Dayton. Specifically, we use the SE layer proposed in [49] to replace our multi-scale spatial pooling module, obtaining the following results in terms of SSIM, PSNR, and SD: 0.5912, 23.3857, and 19.8061, respectively. We can see that our method (see the Baseline H in Table 5.1)

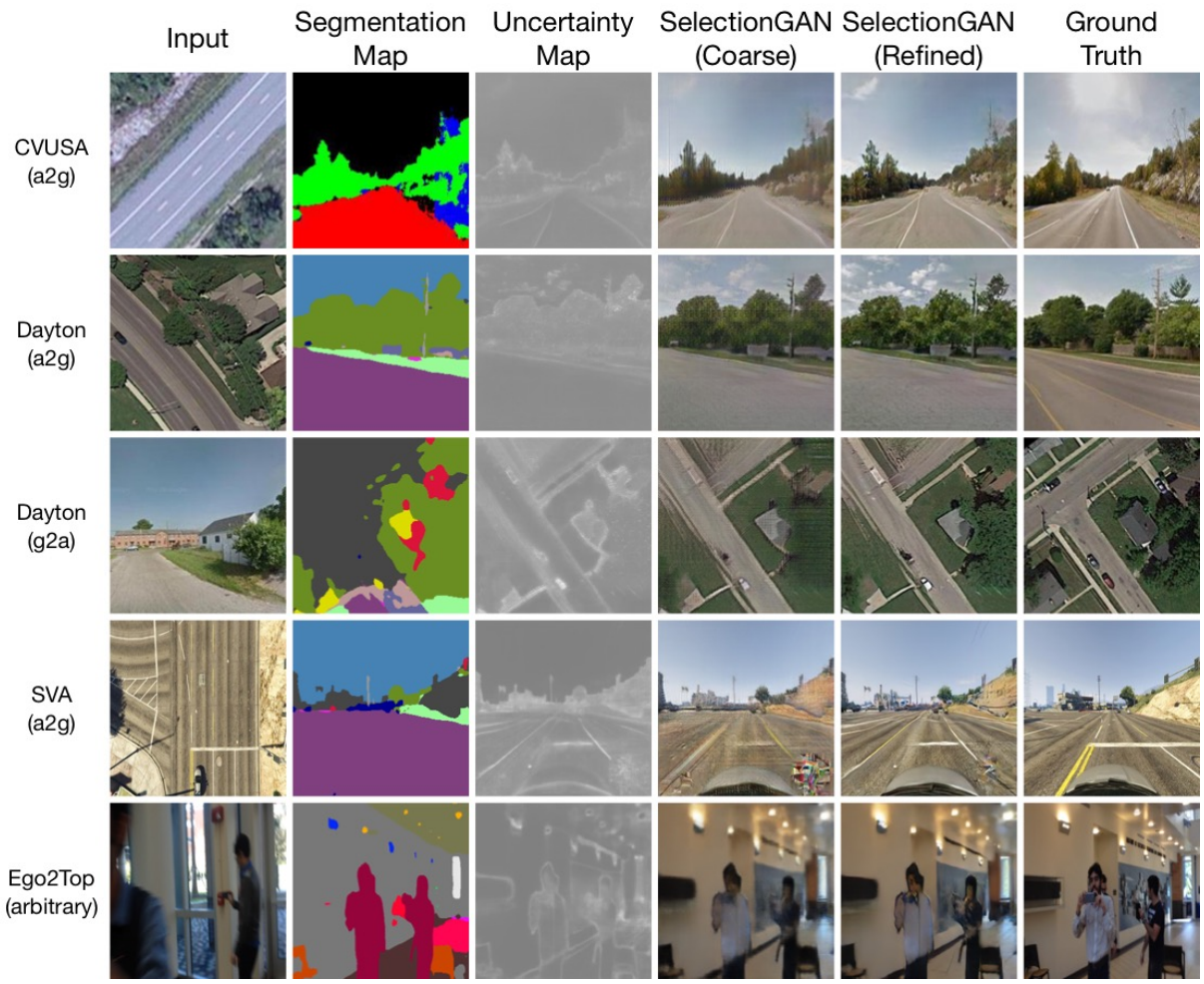


Figure 5.4: Results of cross-view image translation generated by the proposed SelectionGAN on different datasets.

still significantly outperforms [49]. Moreover, we provide the visualization results in Figure 5.5 (note that our method achieves better results than SENet).

Influence of the Number of Attention Channels. We investigate the influence of the number of attention channel N in Equation (5.4). The results are shown in Table 5.3. We observe that the performance tends to be stable after N=10. Thus, taking both performance and training speed into consideration, we set N=10 in all our experiments.

SelectionGAN vs. SelectionGAN++. We also provide comparison results of SelectionGAN [165] and SelectionGAN++ on both SVA and

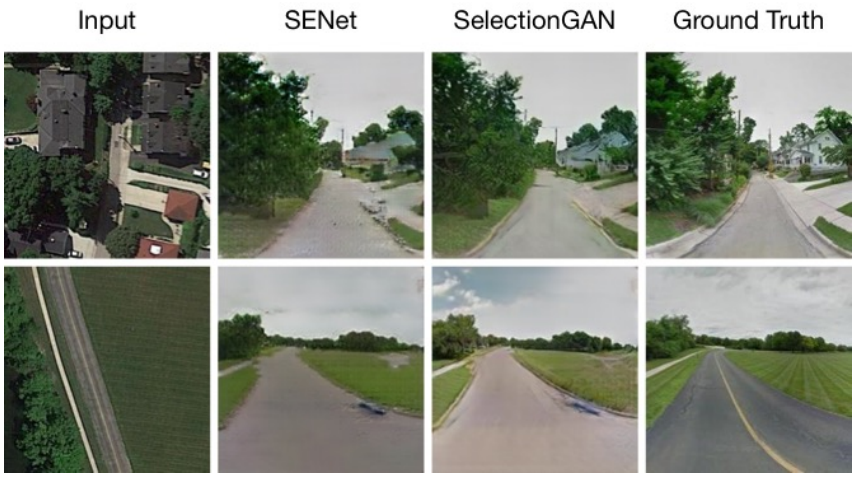


Figure 5.5: Comparison results of SENet and the proposed SelectionGAN on Dayton.

N	SSIM	PSNR	SD
0	0.5438	22.9773	19.4568
1	0.5522	23.0317	19.5127
5	0.5901	23.8068	**20.0033**
10	**0.5986**	23.7336	19.9993
32	0.5950	**23.8265**	19.9086

Table 5.3: Influence of the number of attention channels N.

Radboud Faces datasets. The results of cross-view image translation and facial expression generation are shown in Tables 5.4 and 5.9, respectively. We can see that SelectionGAN++ achieves better results on most metrics, meaning that the proposed multi-scale channel selection module indeed enhances the feature representation, and thus is improving the generation performance. Note that SelectionGAN++ generates sharper and more realistic images than SelectionGAN, but SelectionGAN has higher pixel-wise similarity scores (i.e., SSIM, PSNR, and SD). This is also observed in other image generation [98], super-resolution [60], and human perceptual judgment [220] tasks. From the visualization results in Figure 5.6, we can see that SelectionGAN++ generates more photo-realistic images with fewer visual artifacts than SelectionGAN on both tasks. For instance, SelectionGAN generates road lines in the first and second rows of Figure 5.6, but

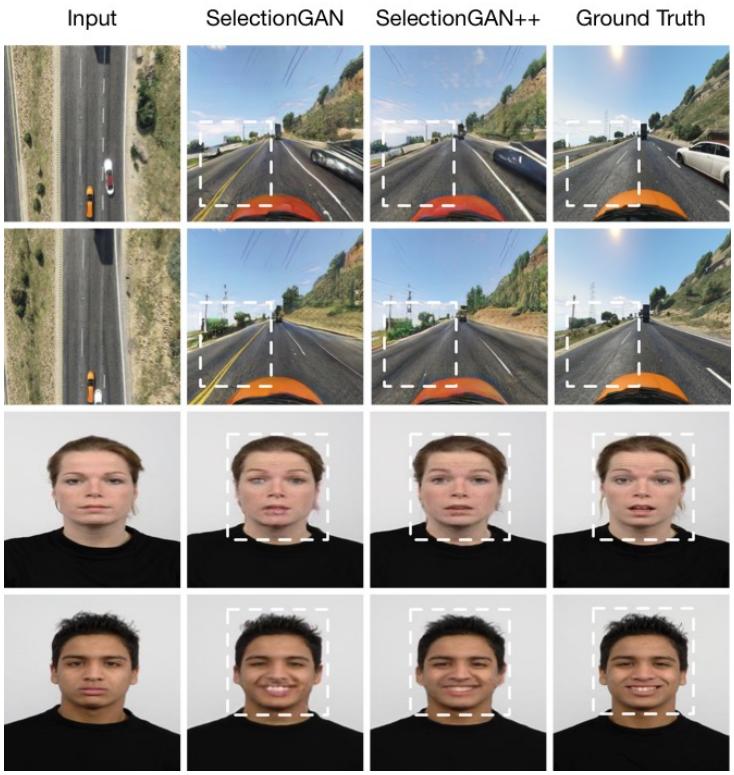


Figure 5.6: Comparison results of SelectionGAN and SelectionGAN++ on Dayton (top two rows) and Radboud Faces (bottom two rows).

Figure 5.7: Results of cross-view image translation on SVA.

there are no road lines in the corresponding ground truths. Meanwhile, SelectionGAN++ generates more realistic eyes and month than SelectionGAN as shown in the third and fourth row of Figure 5.6, respectively.

State-of-the-Art Comparisons. We compare our SelectionGAN with

Method	Accuracy (%) ↑				Inception Score* ↑			SSIM ↑	PSNR ↑	SD ↑	KL ↓	FID ↓
	Top-1		Top-5		All	Top-1	Top-5					
X-Pix2pix [55]	8.5961	30.3288	9.0260	29.9102	2.0131	1.7221	2.2370	0.3206	17.9944	17.0254	19.5533	859.66
X-SO [134]	7.5146	30.9507	10.3905	38.9822	2.4951	1.8940	2.6634	0.4552	21.5312	17.5285	12.0906	443.79
X-Fork [133]	17.3794	53.4725	23.8315	63.5045	2.1888	1.9776	2.3664	0.4235	21.2400	16.9371	4.1925	129.16
X-Seq [133]	19.5056	57.1010	25.8807	65.3005	2.2232	1.9842	2.4344	0.4638	22.3411	17.4138	3.7585	118.70
H-Pix2pix [134]	18.0706	54.8068	23.4400	62.3072	2.1906	1.9507	2.4069	0.4327	21.6860	16.9468	4.2894	117.13
H-SO [134]	5.2444	26.4697	5.2544	31.9527	2.3202	1.9410	2.7340	0.4457	21.7709	17.3876	12.8761	1452.88
H-Fork [134]	18.0182	51.0756	26.6747	62.8166	2.3202	1.9525	2.3918	0.4240	21.6327	16.8653	4.7246	109.43
H-Seq [134]	20.7391	57.5378	28.5517	67.4649	2.2394	1.9892	2.4385	0.4249	21.4770	17.5616	4.4260	95.12
H-Regions [134]	15.4803	48.0767	21.8225	56.8994	2.6328	2.0732	2.8347	0.4044	20.9848	17.6858	6.0638	88.78
Pix2pix++ [55]	8.8687	34.5434	9.2713	35.7490	2.5625	2.0879	2.7961	0.3664	17.6549	18.4015	13.1153	220.23
X-Fork++ [133]	10.2658	37.8405	11.4138	38.7976	2.4280	2.0387	2.7630	0.3406	17.3937	18.2153	10.1403	166.33
X-Seq++ [133]	11.2580	36.8018	11.9838	36.9231	2.6849	2.1325	2.9397	0.3617	17.4893	18.4122	11.8560	154.80
Pix2pixHD [184]	35.0018	72.9430	52.2181	85.6375	2.5820	2.1436	2.8730	0.5437	23.1823	18.9723	2.6322	32.79
GauGAN [119]	34.6740	71.4061	50.1152	81.4900	2.6462	**2.2112**	2.9550	0.5195	22.0174	18.7762	2.6714	27.93
SelectionGAN	33.9055	71.8779	50.8878	85.0019	2.6576	2.1279	2.9267	**0.5752**	**24.7136**	**19.7302**	2.6183	**26.09**
SelectionGAN++	**35.9008**	**73.3249**	**52.5346**	**86.9432**	**2.7370**	2.1914	**3.0271**	0.5481	24.2886	19.2001	**2.5788**	37.17

Table 5.4: Quantitative results of cross-view image translation on SVA. (∗) Inception Score for real (ground truth) data is 3.1282, 2.4932 and 3.4646 for all, top-1 and top-5 setups, respectively.

Method	Accuracy (%) ↑				Inception Score* ↑			SSIM ↑	PSNR ↑	SD ↑	KL ↓
	Top-1		Top-5		All	Top-1	Top-5				
Zhai et al. [211]	13.97	14.03	42.09	52.29	1.8434	1.5171	1.8666	0.4147	17.4886	16.6184	27.43 ± 1.63
Pix2pix [55]	7.33	9.25	25.81	32.67	3.2771	2.2219	3.4312	0.3923	17.6578	18.5239	59.81 ± 2.12
X-SO [134]	0.29	0.21	6.14	9.08	1.7575	1.4145	1.7791	0.3451	17.6201	16.9919	414.25 ± 2.37
X-Fork [133]	20.58	31.24	50.51	63.66	3.4432	2.5447	3.5567	0.4356	19.0509	18.6706	11.71 ± 1.55
X-Seq [133]	15.98	24.14	42.91	54.41	**3.8151**	2.6738	**4.0077**	0.4231	18.8067	18.4378	15.52 ± 1.73
Pix2pix++ [55]	26.45	41.87	57.26	72.87	3.2592	2.4175	3.5078	0.4617	21.5739	18.9044	9.47 ± 1.69
X-Fork++ [133]	31.03	49.65	64.47	81.16	3.3758	2.5375	3.5711	0.4769	21.6504	18.9856	7.18 ± 1.56
X-Seq++ [133]	34.69	54.61	67.12	83.46	3.3919	2.5474	3.4858	0.4740	21.6733	18.9907	5.19 ± 1.31
SelectionGAN	**41.52**	**65.51**	**74.32**	**89.66**	3.8074	**2.7181**	3.9197	**0.5323**	**23.1466**	**19.6100**	**2.96 ± 0.97**

Table 5.5: Quantitative results of cross-view image translation on CVUSA. (∗) Inception Score for real (ground truth) data is 4.8741, 3.2959 and 4.9943 for all, top-1 and top-5 setups, respectively.

Method	Accuracy (%) ↑				Inception Score* ↑			SSIM ↑	PSNR ↑	SD ↑	KL ↓
	Top-1		Top-5		All	Top-1	Top-5				
Pix2pix [55]	6.80	9.15	23.55	27.00	2.8515	1.9342	2.9083	0.4180	17.6291	19.2821	38.26 ± 1.88
X-SO [134]	27.56	41.15	57.96	73.20	2.9459	2.0963	2.9980	0.4772	19.6203	19.2939	7.20 ± 1.37
X-Fork [133]	30.00	48.68	61.57	78.84	3.0720	2.2402	3.0932	0.4963	19.8928	19.4533	6.00 ± 1.28
X-Seq [133]	30.16	49.85	62.59	80.70	2.7384	2.1304	2.7674	0.5031	20.2803	19.5258	5.93 ± 1.32
Pix2pix++ [55]	32.06	54.70	63.19	81.01	**3.1709**	2.1200	3.2001	0.4871	21.6675	18.8504	5.49 ± 1.25
X-Fork++ [133]	34.67	59.14	66.37	84.70	3.0737	2.1508	3.0893	0.4982	21.7260	18.9402	4.59 ± 1.16
X-Seq++ [133]	31.58	51.67	65.21	82.48	3.1703	2.2185	**3.2444**	0.4912	21.7659	18.9265	4.94 ± 1.18
SelectionGAN	**42.11**	**68.12**	**77.74**	**92.89**	3.0613	**2.2707**	3.1336	**0.5938**	**23.8874**	**20.0174**	**2.74 ± 0.86**

Table 5.6: Quantitative results of cross-view image translation on Dayton in a2g direction. (∗) Inception Score for real (ground truth) data is 3.8319, 2.5753 and 3.9222 for all, top-1 and top-5 setups, respectively.

several recently proposed state-of-the-art methods, which are Pix2pix [55], Zhai et al. [211], X-Fork [133], X-Seq [133] and X-SO [134]. Moreover, to

Method	Accuracy (%) ↑				Inception Score* ↑			SSIM ↑	PSNR ↑	SD ↑	KL ↓
	Top-1		Top-5		All	Top-1	Top-5				
Pix2pix [55]	1.22	1.57	5.33	6.86	2.5418	1.6797	2.4947	0.2213	15.7197	16.5949	120.46 ± 1.94
X-Fork [133]	5.91	10.22	20.98	30.29	4.6447	2.1386	3.8417	0.2740	16.3709	17.3509	22.12 ± 1.65
X-Seq [133]	4.78	8.96	17.04	24.40	4.5094	2.0276	3.6756	0.2738	16.3788	17.2624	25.19 ± 1.73
Pix2pix++ [55]	19.53	33.19	40.89	48.34	5.0833	2.4096	4.4595	0.3779	21.1346	17.8056	10.93 ± 1.87
X-Fork++ [133]	13.92	22.38	34.20	42.42	5.2266	2.4100	4.5591	0.3560	20.5788	17.6183	17.34 ± 1.98
X-Seq++ [133]	19.41	36.11	40.46	50.41	4.9890	2.3519	4.2881	0.3878	21.2327	17.9469	9.33 ± 1.64
SelectionGAN	**28.31**	**54.56**	**62.97**	**76.30**	**5.6200**	**2.5328**	**4.7648**	**0.6024**	**26.6565**	**19.7755**	**3.05 ± 0.91**

Table 5.7: Quantitative results of cross-view image translation on Ego2Top. (∗) Inception Score for real (ground truth) data is 6.4523, 2.8507 and 5.4662 for all, top-1 and top-5 setups, respectively.

study the effectiveness of SelectionGAN, we introduce five strong baselines which use both segmentation map and RGB image as inputs, including Pix2pix++ [55], X-Fork++ [133], X-Seq++ [133], Pix2pixHD [184], and GauGAN [119]. The comparison results are shown in Tables 5.4, 5.5, 5.6 and 5.7. We can observe that SelectionGAN consistently outperforms existing methods on most metrics. Qualitative results compared with the leading baselines are shown in Figures 5.7 and 5.8. We can see that our method generates more clear details on objects/scenes such as road, tress, clouds, car than the other comparison methods. Moreover, the results generated by our method are closer to the ground truths in layout and structure.

Visualization of Learned Uncertainty Maps. In Figures 5.4 and 5.9, we show some samples of the generated uncertainty maps. We can see that the generated uncertainty maps learn the layout and structure of the target images. Note that most textured regions are similar in our generation images, while the junction/edge of different regions is uncertain, and thus the model learns to highlight these parts.

Generated Semantic Guidances. Since the proposed SelectionGAN can reconstruct the semantic guidance (here, the segmentation maps), we also compare the generated semantic guidance with X-Fork [133] and X-Seq [133] on Dayton. Following [133], we compute the per-class accuracy

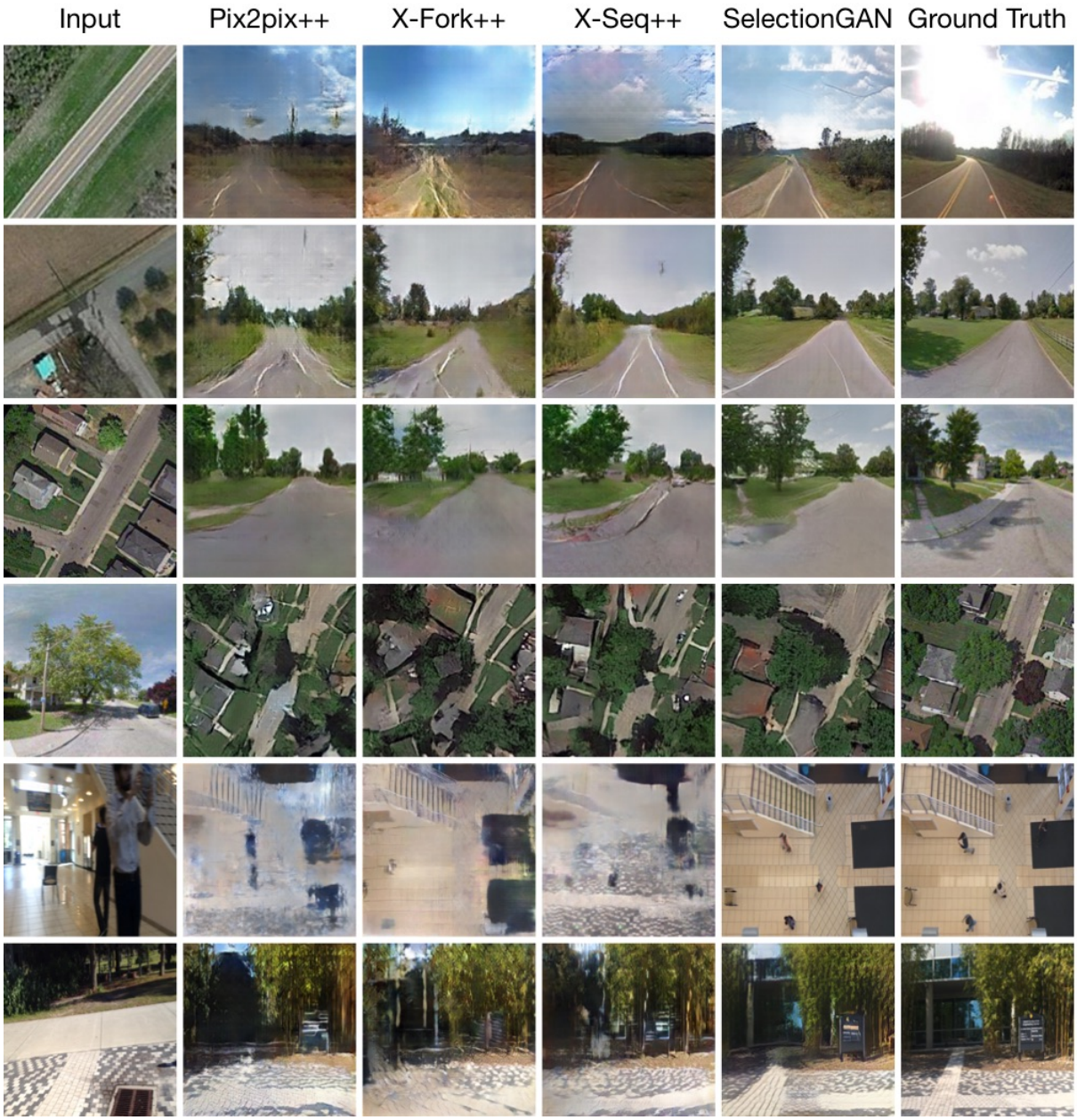


Figure 5.8: Results of cross-view image translation on CVUSA (top two rows), Dayton (middle two rows), and Ego2Top (bottom two rows).

and mean IOU for the most common classes in this dataset (see Table 5.8). We see that our SelectionGAN achieves better results than X-Fork [133] and X-Seq [133] on both metrics.

Controllable Cross-View Image Translation. We further adopt Ego2Top to conduct the controllable cross-view image translation experiments. The

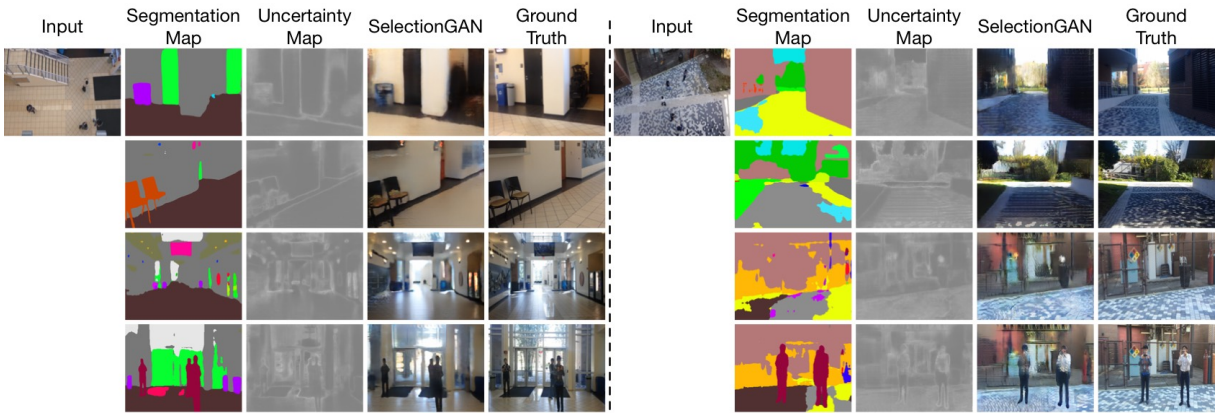


Figure 5.9: Results of controllable cross-view image translation for both indoor (left) and outdoor (right) scenes.

Method	Per-class Acc. ↑	mIOU ↑
X-Fork [133]	0.6262	0.4163
X-Seq [133]	0.4783	0.3187
SelectionGAN	**0.6415**	**0.5455**

Table 5.8: Per-class accuracy and mean IOU for the generated segmentation maps on Dayton.

results are shown in Figure 5.9. Given a single input image and some novel segmentation maps, SelectionGAN is able to generate the same scene but with different viewpoints in both indoor and outdoor environments.

5.4.2 Facial Expression Generation

Datasets. We follow C2GAN [164] and conduct facial expression generation experiments on the Radboud Faces dataset [74]. This dataset contains over 8,000 face images with eight different emotional expressions. We follow C2GAN and all the images are resized to 256×256 without any preprocessing. Then, we adopt OpenFace [4] to extract facial landmarks as the semantic guidance. Consequently, we collect 5,628 training image pairs and 1,407 testing pairs.

Parameter Settings. Following C2GAN [164], the experiments on Radboud Faces are trained for 200 epochs with batch size of 4.

Method	AMT ↑	SSIM ↑	PSNR ↑	LPIPS ↓
StarGAN [25]	24.7	0.8345	19.6451	-
Pix2pix [55]	13.4	0.8217	19.9971	0.1334
GPGAN [28]	0.3	0.8185	18.7211	0.2531
PG2 [98]	28.4	0.8462	20.1462	0.1130
Pix2pixHD [184]	20.5	0.8269	24.5621	0.1228
GauGAN [119]	10.7	0.7528	20.8430	0.2170
C2GAN [164]	34.2	0.8618	21.9192	0.0934
SelectionGAN	37.5	0.8760	**27.5671**	0.0917
SelectionGAN++	**39.1**	**0.8761**	27.5158	**0.0905**

Table 5.9: Quantitative results of facial expression generation on Radboud Faces.

Evaluation Metrics. We follow C2GAN [164] and employ Structural Similarity (SSIM) [191] and Peak Signal-to-Noise Ratio (PSNR) to evaluate the quantitative quality of generated images. Moreover, we adopt Amazon Mechanical Turk (AMT) perceptual studies to evaluate the quality of the generated images. Specifically, participants were shown a sequence of pairs of images, one a real image and one fake image, and asked to click on the image they thought was real. Finally, we also use a neural network based metric LPIPS [220] to evaluate the proposed method.

State-of-the-Art Comparisons. We compare the proposed SelectionGAN with several state-of-the-art methods, i.e., StarGAN [25], Pix2pix [55], GPGAN [28], PG2 [98], Pix2pixHD [184], GauGAN [119], and C2GAN [164]. Quantitative results of the SSIM, PSNR, LPIPS, and AMT metrics are show in Table 5.9. We can see that the proposed SelectionGAN achieves the best results on all metrics, validating the effectiveness of our method. Note that GauGAN achieves unsatisfactory results in this task since it is proposed to use segmentation maps as input. However, this task uses facial landmarks as guidances, which is quite different from segmentation maps. On the contrary, our method achieves good results in this task, which further proves the generalizability of our method. Qualitative results are shown in Figure 5.10. Clearly, the image generated by our SelectionGAN

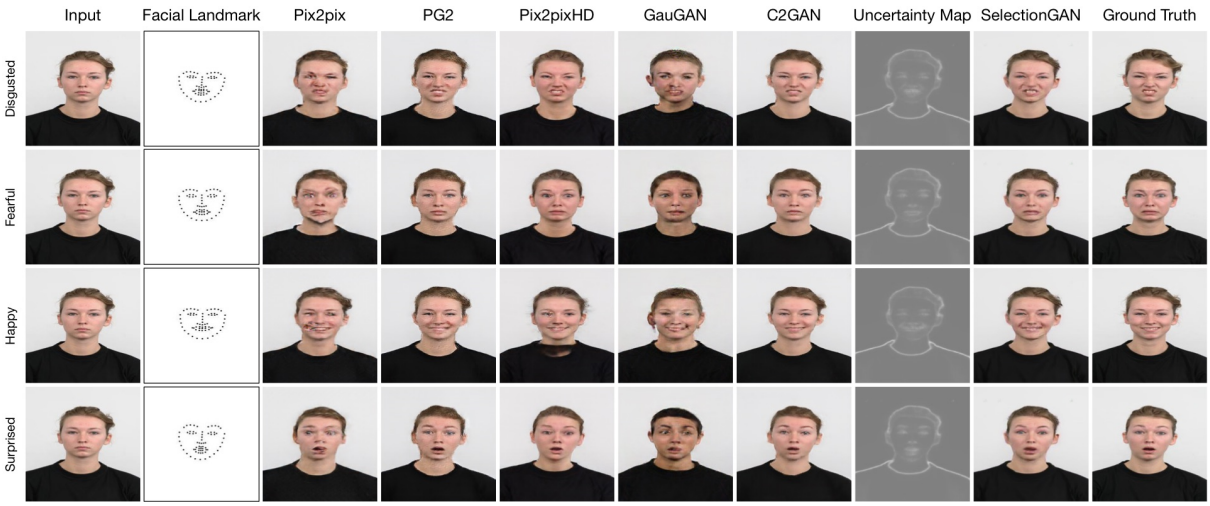


Figure 5.10: Results of facial expression generation on Radboud Faces.

are more sharper and contains more image details compared with other leading methods.

Visualization of Learned Uncertainty Maps. We also show the learned uncertainty maps in Figure 5.10. We observe that the proposed SelectionGAN can generate different uncertainty maps according to different facial expressions, which means the proposed model can learn the difference between different expression domains.

5.4.3 Hand Gesture Translation

Datasets. We follow GestureGAN [163] and conduct experiments on both NTU Hand Digit [137] and Senz3D [101] datasets. NTU Hand Digit dataset contains 75,036 and 9,600 image pairs for training and testing sets, each of which is comprised of two images of the same person but different gestures. For Senz3D, which contains 135,504 pairs and 12,800 pairs for training and testing.

Parameter Settings. Images on both datasets are resized to 256×256, and we enabled image flipping and random crops for data augmentation. Following GestureGAN [163], the experiments on both datasets are trained

Method	NTU Hand Digit					Senz3D				
	PSNR ↑	IS ↑	AMT ↑	FID ↓	FRD ↓	PSNR ↑	IS ↑	AMT ↑	FID ↓	FRD ↓
PG2 [98]	28.2403	2.4152	3.5	24.2093	2.6319	26.5138	3.3699	2.8	31.7333	3.0933
SAMG [201]	28.0185	2.4919	2.6	31.2841	2.7453	26.9545	3.3285	2.3	38.1758	3.1006
DPIG [99]	30.6487	2.4547	7.1	**6.7661**	2.6184	26.9451	3.3874	6.9	26.2713	3.0846
PoseGAN [146]	29.5471	2.4017	9.3	9.6725	2.5846	27.3014	3.2147	8.6	24.6712	3.0467
Pix2pixHD [184]	**38.1295**	2.2358	21.3	8.4003	**1.1475**	-	-	-	-	-
GauGAN [119]	32.2218	**2.6210**	13.2	18.4373	1.8229	-	-	-	-	-
GestureGAN [163]	32.6091	2.5532	**26.1**	7.5860	2.5223	27.9749	**3.4107**	**22.6**	**18.4595**	2.9836
SelectionGAN	30.6465	2.4472	15.8	16.2159	2.1560	**30.4036**	2.4595	14.1	30.9775	**2.7014**

Table 5.10: Quantitative results of hand gesture-to-gesture translation on NTU Hand Digit and Senz3D.

for 20 epochs with batch size of 4.

Evaluation Metrics. We follow [163] and employ Peak Signal-to-Noise Ratio (PSNR), Inception score (IS) [139], Fréchet Inception Distance (FID) [48], and Fréchet ResNet Distance (FRD) [163] to evaluate the generated images. Moreover, we follow the same settings as in [55, 163] to conduct the Amazon Mechanical Turk (AMT) perceptual studies.

State-of-the-Art Comparisons. We compare the proposed SelectionGAN with the leading hand gesture translation methods, i.e., PG2 [98], SAMG [201], DPIG [99], PoseGAN [146], Pix2pixHD [184], GauGAN [119], and GestureGAN [163]. Comparison results are shown in Table 5.10. We can see that our SelectionGAN achieves competitive results on both datasets. Qualitative results compared with existing methods are shown in Figure 5.11. We can see that our SelectionGAN also generates photo-realistic results on this task. Moreover, we show the learned uncertainty maps in Figures 5.11 and 5.12.

Controllable Hand Gesture Translation. In Figure 5.12, we provide results of controllable hand gesture translation. We can see that the proposed SelectionGAN can translates a single input image into several output images while each one respecting the constraints specified in the provided hand skeleton.

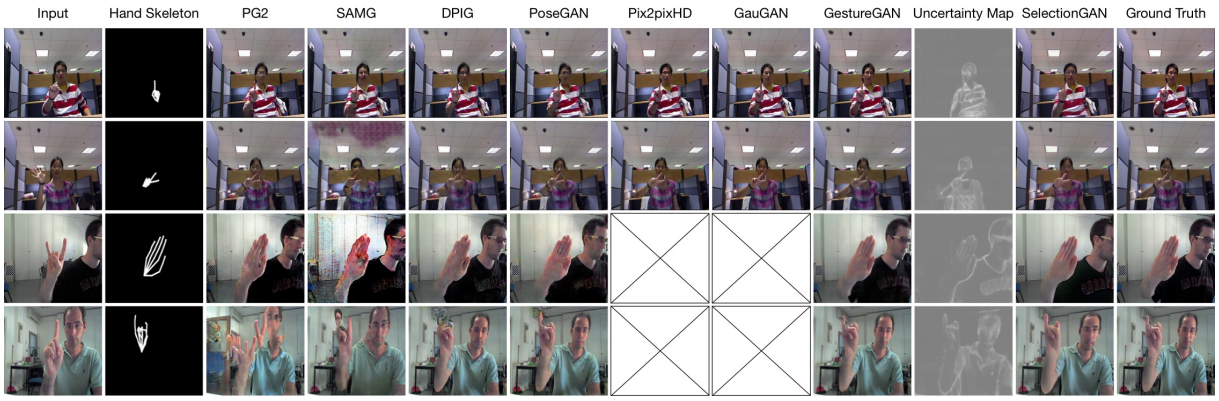


Figure 5.11: Results of hand gesture-to-gesture translation on NTU Hand Digit (top two rows) and Senz3D (bottom two rows).

5.4.4 Person Image Generation

Datasets. We follow PATN [234] and conduct person image generation experiments on both Market-1501 [224] and DeepFashion [97] datasets. Following [234], we collect 263,632 and 12,000 pairs for training and testing on Market-1501. For DeepFashion, 101,966 and 8,570 pairs are randomly selected for training and testing.

Parameter Settings. Following PATN [234], images are re-scaled to 128×64 and 256×256 on Market-1501 and DeepFashion datasets, respectively. Moreover, the experiments on both datasets are trained for around 90k iteration with batch size of 32 and 12 on Market-1501 and DeepFashion, respectively.

Evaluation Metrics. We follow previous works [98, 146, 164, 146, 234] and adopt Structure Similarity (SSIM) [191], Inception score (IS) [139] and their corresponding masked versions, i.e., M-SSIM and M-IS, as our evaluation metrics. Moreover, we recruit 30 volunteers to conduct a user study.

State-of-the-Art Comparisons. We compare the proposed SelectionGAN with several leading person image generation methods, i.e., PG2 [98], DPIG [99], PoseGAN [146], VUNet [36], C2GAN [164], BTF [2], Pix2pixHD

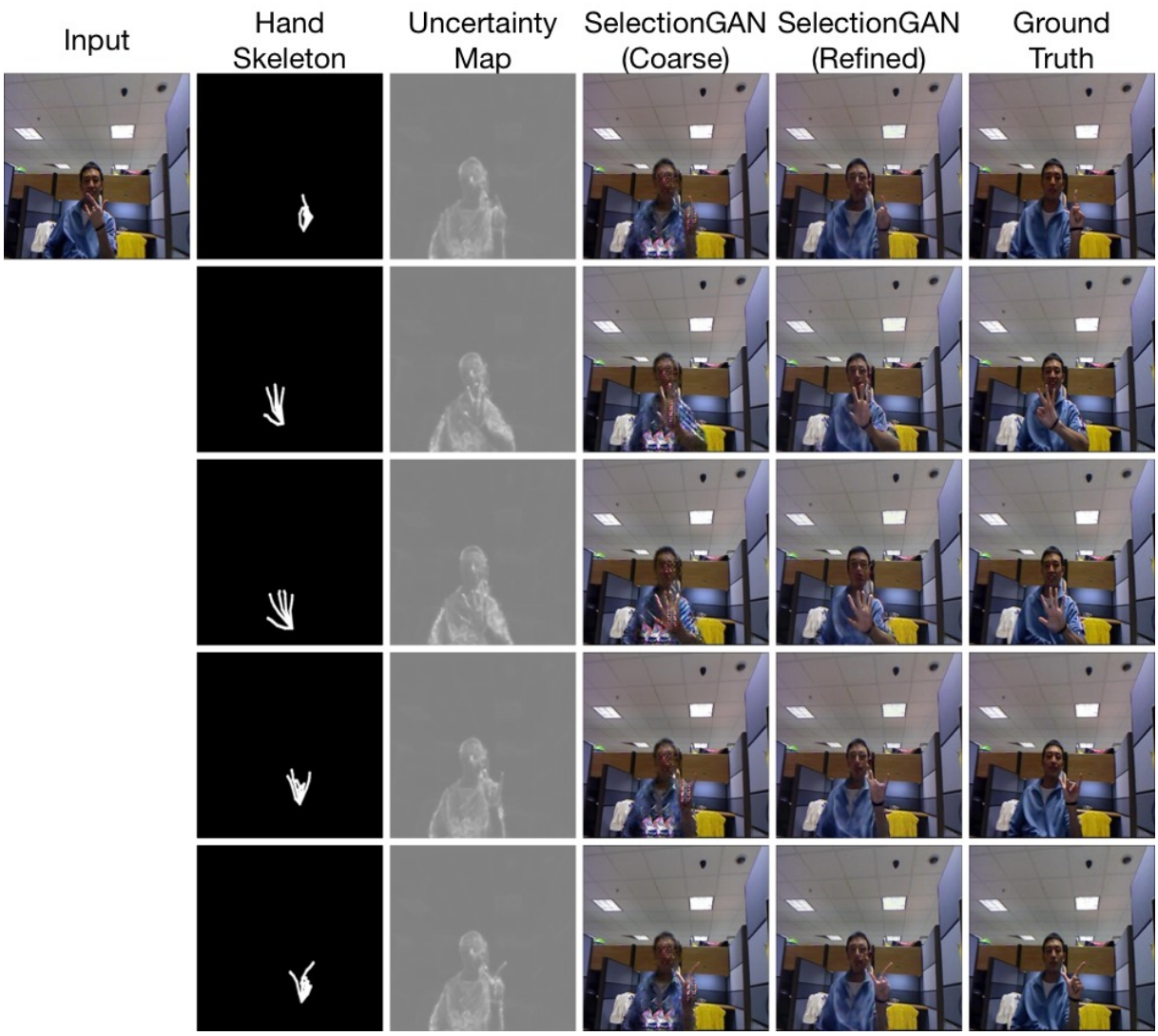


Figure 5.12: Results of controllable hand gesture translation.

[184], GauGAN [119], and PATN [234]. Quantitative results of the SSIM, IS, M-SSIM, and M-IS metrics are show in Table 5.11. We can see that the proposed SelectionGAN achieves competitive performance compared with the carefully designed methods on this task such as PATN [234] and PoseGAN [146]. Moreover, we show user study results in Table 5.12. We observe that our method achieve better results over [146, 164, 98, 234], further validating that our generated images are more photo-realistic. Qualitative results are shown in Figure 5.13. The image generated by our

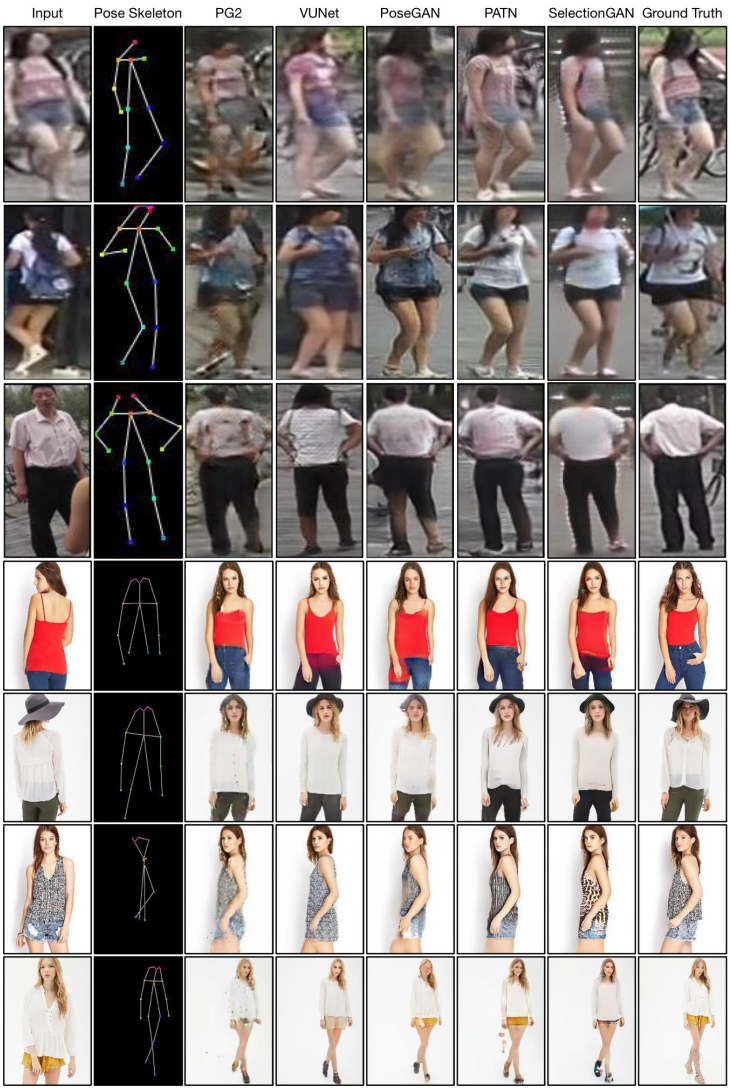


Figure 5.13: Results of person image generation on Market-1501 (top three rows) and DeepFashion (bottom four rows).

SelectionGAN are more realistic and sharp compared with other leading methods. Moreover, the person layouts of generated images by our method are closer to the target skeletons.

5.4.5 Semantic Image Synthesis

To explore the generality of SelectionGAN on other generation tasks, we also conduct experiments on the challenging semantic image synthesis task.

Method	Market-1501				DeepFashion	
	SIM ↑	IS ↑	M-SSIM ↑	M-IS ↑	SSIM ↑	IS ↑
PG2 [98]	0.253	3.460	0.792	3.435	0.762	3.090
DPIG [99]	0.099	3.483	0.614	3.491	0.614	3.228
PoseGAN [146]	0.290	3.185	0.805	3.502	0.756	3.439
C2GAN [164]	0.282	3.349	0.811	3.510	-	-
BTF [2]	-	-	-	-	0.767	3.220
PG2* [98]	0.261	**3.495**	0.782	3.367	0.773	3.163
PoseGAN* [146]	0.291	3.230	0.807	3.502	0.760	3.362
Pix2pixHD* [184]	-	-	-	-	0.762	3.224
GauGAN* [119]	-	-	-	-	0.754	3.165
VUNet* [36]	0.266	2.965	0.793	3.549	0.763	**3.440**
PATN* [234]	0.311	3.323	0.811	**3.773**	0.773	3.209
SelectionGAN	**0.331**	3.449	**0.816**	3.376	**0.776**	3.341
Real Data	1.000	3.890	1.000	3.706	1.000	4.053

Table 5.11: Quantitative results of person image generation on Market-1501 and DeepFashion.

Method	Market-1501		DeepFashion	
	R2G ↑	G2R ↑	R2G ↑	G2R ↑
PG2 [98]	11.2	5.5	9.2	14.9
PoseGAN [146]	22.67	50.24	12.42	24.61
C2GAN [164]	23.20	46.70	-	-
PATN [234]	32.23	63.47	19.14	31.78
SelectionGAN	**34.64**	**64.75**	**20.57**	**33.54**

Table 5.12: User study (%) of person image generation. R2G means the percentage of real images rated as generated w.r.t. all real images. G2R means the percentage of generated images rated as real w.r.t. all generated images.

Specifically, we adopt GauGAN [119] as our backbone network in this task and we combine it with the proposed multi-channel attention selection module to form our final model.

Datasets. We follow GauGAN [119] and conduct semantic image synthesis experiments on two challenging datasets, i.e., Cityscapes [26] and ADE20K [226]. The training and testing set sizes of Cityscapes are 2,975 and 500, respectively. For ADE20K, which contains 150 semantic classes,

Figure 5.14: Results of semantic image synthesis on Cityscapes (top two rows) and ADE20K (bottom three rows).

and has 20,210 training and 2,000 validation images.

Parameter Settings. Images are re-scaled to 512×256 and 256×256 on Cityscapes and ADE20K datasets, respectively. Following GauGAN [119], the experiments on both datasets are trained for 200 epochs with batch size of 32.

Evaluation Metrics. We Follow [119] and employ the mean Intersection-over-Union (mIoU) and pixel accuracy (Acc) to measure the segmentation accuracy. Specifically, we adopt the state-of-the-art segmentation networks to evaluate the generated images, i.e., DRN-D-105 [207] for Cityscapes and UperNet101 [195] for ADE20K. We also employ the Fréchet Inception Distance (FID) [48] to measure the distance between the distribution of generated samples and the distribution of real samples. Finally, we follow GauGAN and employ Amazon Mechanical Turk (AMT) to measure the perceived visual fidelity of the generated images.

State-of-the-Art Comparisons. We adopt several leading semantic image synthesis methods as our baselines, i.e., Pix2pixHD [184], CRN [16], SIMS [127], and GauGAN [119]. The results of mIoU, Acc, and FID are

Method	Cityscapes			ADE20K		
	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓
CRN [16]	52.4	77.1	104.7	22.4	68.8	73.3
SIMS [127]	47.2	75.5	**49.7**	-	-	-
Pix2pixHD [184]	58.3	81.4	95.0	20.3	69.2	81.8
GauGAN [119]	62.3	81.9	71.8	38.5	79.9	33.9
SelectionGAN	**63.8**	**82.4**	65.2	**40.1**	**81.2**	**33.1**

Table 5.13: Quantitative results of semantic image synthesis on Cityscapes and ADE20K.

AMT ↑	Cityscapes	ADE20K
Ours vs. CRN [16]	63.86	69.43
Ours vs. Pix2pixHD [184]	54.04	78.62
Ours vs. SIMS [127]	53.57	-
Ours vs. GauGAN [119]	52.89	55.15

Table 5.14: User preference study of semantic image synthesis on Cityscapes and ADE20K. The numbers indicate the percentage of users who favor the results of the proposed method over the competing method.

show in Table 5.13. We note that our SelectionGAN achieves better results than the existing competing methods on both mIoU and Acc metrics. For FID, SelectionGAN is only worse than SIMS on Cityscapes. However, SIMS has poor segmentation results. Moreover, we follow GauGAN and provide AMT results in Table 5.14. We observe that users favor our translated images on both datasets compared with existing leading methods. Qualitative results compared with exiting methods are shown in Figure 5.14. We observe that SelectionGAN produces much better results with fewer visual artifacts than exiting methods.

Visualization of Generated Segmentation Maps. We follow GauGAN and apply pre-trained segmentation networks on the generated images to produce segmentation maps. The intuition behind this is that if the generated images are realistic, a well-trained semantic segmentation model should be able to predict the ground truth label. The results compared with the state-of-the-art method GauGAN are shown in Figure 5.15. We

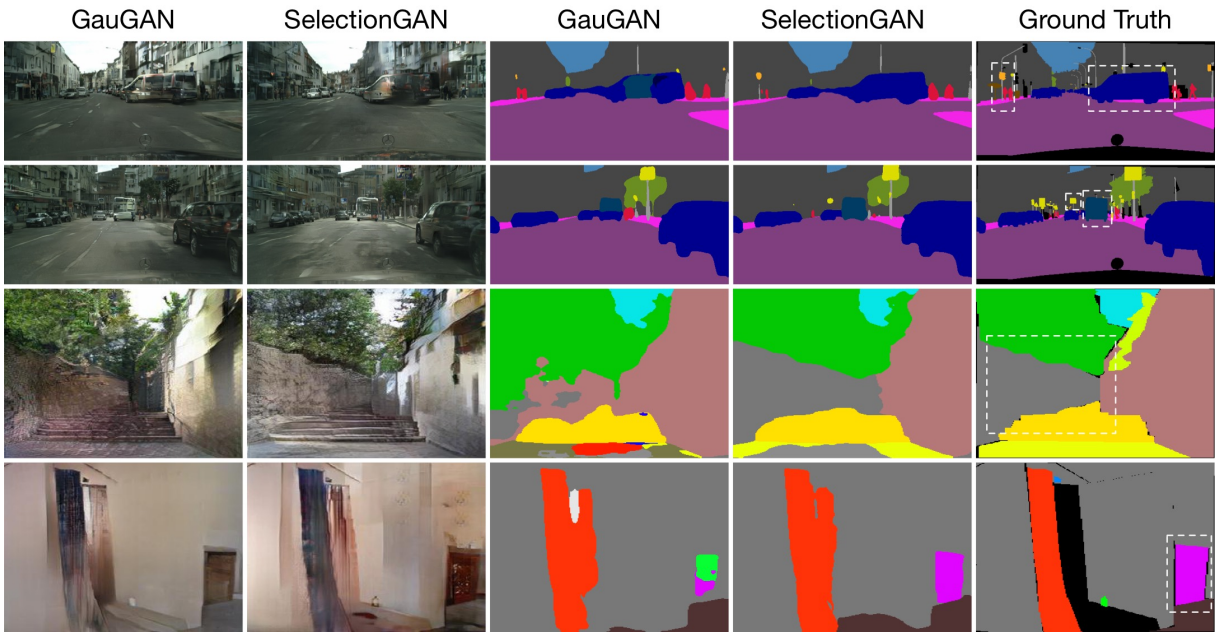


Figure 5.15: Generated segmentation maps on Cityscapes (top two rows) and ADE20K (bottom two rows).

observe that the proposed SelectionGAN generates better semantic maps than GauGAN on both datasets.

5.5 Conclusion

We propose SelectionGAN to address a novel image synthesis task by conditioning on a input image and several conditional semantic guidances. In particular, we adopt a cascade strategy to divide the generation procedure into two stages. Stage I aims to capture the semantic structure of the target image and Stage II focus on more appearance details via the proposed multi-scale spatial pooling & channel selection and the multi-channel attention selection modules. We also propose an uncertainty map guided pixel loss to solve the inaccurate semantic guidance issue for better optimization. Experimental results on four guided image-to-image translation and one semantic image synthesis tasks with 11 public datasets show that our method obtains much better results than the state-of-the-art models.

In next chapter, we will introduce LGGAN and EdgeGAN for semantic image syntesis. LGGAN explores image generation from local context, which we believe is beneficial for generating richer details compared with the existing global image-level generation methods. A new local class-specific generative structure is designed for this purpose. It can effectively handle the generation of small objects and details, which are common difficulties encountered by the global-based generation. In EdgeGAN, we propose an effective attention guided edge transfer module to selectively transfer useful edge structure information from the edge generation branch to the image generation branch. We also design a new semantic preserving module to highlight class-dependent feature maps based on the input semantic label map for generating semantically consistent results. Both ideas have not been investigated by any existing GAN-based generation works.

Chapter 6

LGGAN and EdgeGAN

In this paper, we address the task of semantic-guided image generation. One challenge common to most existing image-level generation methods is difficulty in generating small objects and detailed local textures. To tackle this issue, in this work we consider generating images using local context. As such, we design a local class-specific generative network using semantic maps as guidance, which separately constructs and learns subgenerators for different classes, enabling it to capture finer details. To learn more discriminative class-specific feature representations for the local generation, we also propose a novel classification module. To combine the advantages of both global image-level and local class-specific generation, a joint generation network is designed with an attention fusion module and a dual-discriminator structure embedded. Lastly, we propose a novel semantic-aware upsampling method, which has a larger receptive field and can take far-away pixels that are semantically related for feature upsampling, enabling it to better preserve semantic consistency for instances with the same semantic labels. Extensive experiments on two image generation tasks show the superior performance of the proposed method. State-of-the-art results are established by large margins on both tasks and on nine challenging public benchmarks. The source code and trained models are

available at https://github.com/Ha0Tang/LGGAN.

We also propose a novel Edge guided Generative Adversarial Network (EdgeGAN) for photo-realistic image synthesis from semantic layouts. Although considerable improvement has been achieved, the quality of synthesized images is far from satisfactory due to two largely unresolved challenges. First, the semantic labels do not provide detailed structural information, making it difficult to synthesize local details and structures. Second, the widely adopted CNN operations such as convolution, downsampling and normalization usually cause spatial resolution loss and thus are unable to fully preserve the original semantic information, leading to semantically inconsistent results (e.g., missing small objects). To tackle the first challenge, we propose to use edge as an intermediate representation which is further adopted to guide image generation via a proposed attention guided edge transfer module. Edge information is produced by a convolutional generator and introduces detailed structure information. Further, to preserve the semantic information, we design an effective module to selectively highlight class-dependent feature maps according to the original semantic layout. Extensive experiments on six challenging datasets show that the proposed EdgeGAN can generate significantly better results than state-of-the-art methods. The source code and trained models are available at https://github.com/Ha0Tang/EdgeGAN.

6.1 Introduction

Semantic-guided image generation is a hot topic covering several mainstream research directions, including cross-view image translation [55, 211, 133, 165] and semantic image synthesis [184, 16, 127, 119]. The cross-view image translation task, proposed in [133], is essentially an ill-posed problem due to the large ambiguity in the generation if only a single RGB image is

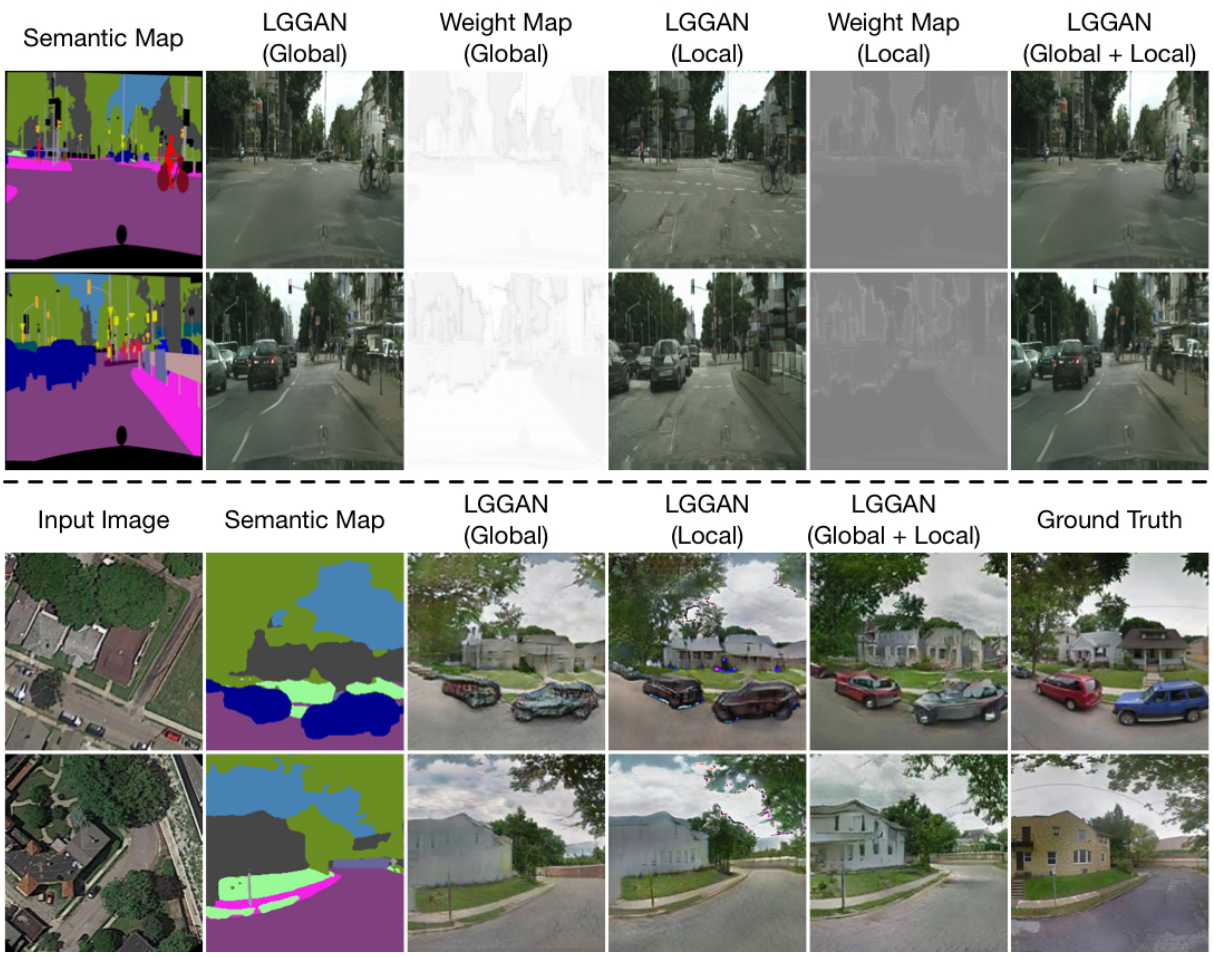


Figure 6.1: Examples of semantic image synthesis results on Cityscapes (top) and cross-view image translation results on Dayton (bottom) with different settings of our LGGAN.

given as input. To alleviate this problem, recent works, such as Selection-GAN [165], try to generate the target image based on another image of the scene and several novel semantic maps, as shown in Figure 6.1 (bottom). Adding a semantic map allows the model to learn the correspondences in the target view with appropriate object relations and transformations. On the other side, the semantic image synthesis task aims to generate a photo-realistic image from a semantic map (see Figure 6.1 (top)). This has many real-world applications and has drawn much attention from the academic research community as well as industry [184, 16, 127, 119, 96, 58, 154]. With the use of semantic information, existing methods for both tasks have

achieved promising performance in semantic-guided image generation.

However, there is still room for improvement, especially when it comes to generating local structures and details, as well as small-scale objects. We believe there are several reasons for this. First, the existing methods for both tasks are typically based on global image-level generation. In other words, they accept a semantic map containing several object classes and aim to generate the appearance of each one using the same network design or shared network parameters. In this case, all the classes are treated equally by the network. However, because different semantic classes have distinct properties, using specified network learning for each would intuitively facilitate the complex generation of multiple classes. Second, the number of training samples for different semantic classes is often imbalanced. For instance, for the Dayton dataset [179], cars and buses occupy less than 2% of all pixels in the training data. This naturally causes the model learning to be dominated by the classes with the largest number of training samples. Third, the sizes of objects in different semantic classes vary. As shown in the first row of Figure 6.1, larger-scale object classes, such as roads and sky, usually occupy a bigger area of the image than smaller-scale classes, such as poles and traffic lights. Since convolutional networks usually share parameters at different convolutional positions, the larger-scale object classes would thus take advantage during the learning process, further increasing the difficulty in accurately generating the small-scale object classes.

To tackle these issues, a straightforward solution would be to model the generation of different image classes individually using local context. By so doing, each class could have its own generation network structure or parameters, thus greatly avoiding the learning of a biased generation space. To achieve this goal, in this paper, we design a novel class-specific generation network. It consists of several subgenerators for different classes

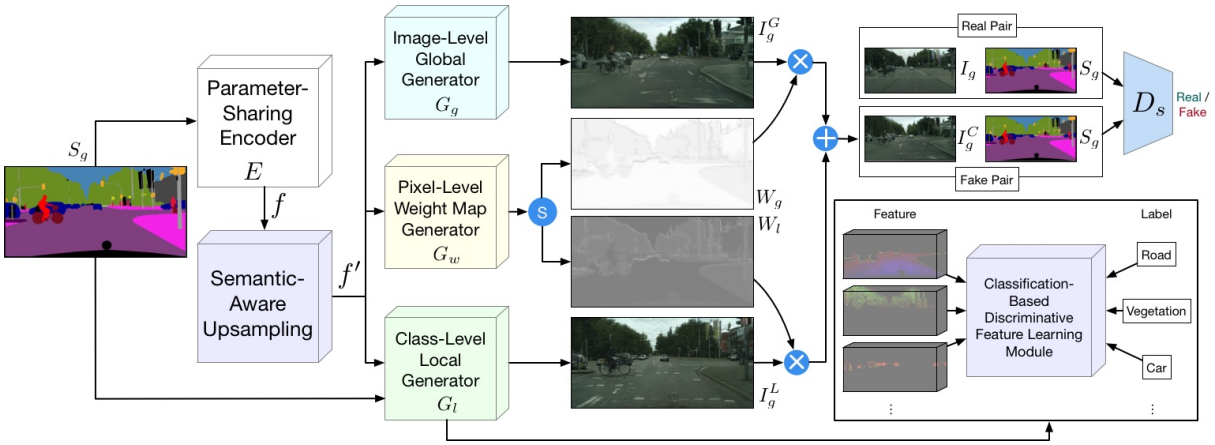
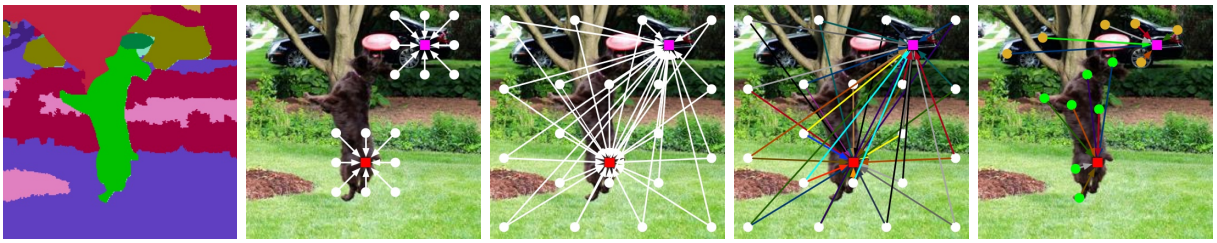


Figure 6.2: Overview of the proposed method, which contains a semantic-guided generator G and discriminator D_s. G consists of a parameter-sharing encoder E, an image-level global generator G_g, a class-level local generator G_l and a weight map generator G_w. The global generator and local generator are automatically combined by two learned weight maps from the weight map generator to reconstruct the target image. D_s tries to distinguish the generated images from two modality spaces, i.e., the image space and semantic space. Moreover, to learn a more discriminative class-specific feature representation, a novel classification module is proposed. Lastly, to better preserve semantic information when upsampling feature maps, a novel semantic-aware upsampling method is introduced. All of these components are trained in an end-to-end fashion so that the local generation and the global generation can benefit from each other. The symbols $\oplus$, $\otimes$ and Ⓢ denote element-wise addition, element-wise multiplication and channel-wise Softmax, respectively.

with a shared encoded feature map. The input semantic map is utilized as the guidance to obtain feature maps corresponding spatially to each class, which are then used to produce a separate generation for different class regions.

Due to the highly complementary properties of global and local generation, a local class-specific and global image-level generative adversarial network (LGGAN) is proposed to combine the advantages of both. It contains three main network branches (see Figure 6.2). The first is the image-level global generator, which learns a global appearance distribution using the input. The second is the proposed class-specific local generator, which aims to generate different object classes separately, using semantic-guided class-specific feature filtering. Finally, the fusion weight map generation



(a) Input Semantic Map. (b) Nearest, bilinear, etc. (c) Deconvolution. (d) Spatial Attention. (e) SAU (Ours).

Figure 6.3: Comparison with different feature upsampling and enhancement methods on the semantic-guided image generation task. Given two locations (indicated by red and magenta squares) in the output feature map, our goal is to generate these locations by selectively upsampling several points (indicated by circles) in the input feature map.

branch learns two pixel-level weight maps, which are used to fuse the local and global subnetworks in a weighted combination of their final generation results. The proposed LGGAN can be jointly trained in an end-to-end fashion, enabling the local and global generation to benefit from each other during optimization.

Moreover, existing methods, such as [119, 154], typically adopt nearest-neighbor interpolation to upsample feature maps and then generate final results, which leads to many visual artifacts and blurriness in the generated images. Feature upsampling is a key operation in the semantic-guided image synthesis task. Traditional upsampling methods, such as nearest-neighbor, bilinear, and bicubic interpolation only consider subpixel neighborhoods (indicated by white circles in Figure 6.3(b)), failing to capture the complete semantic information, e.g., the head and body of the dog, and the front part of the car. Learnable upsampling methods, such as deconvolution [111] and pixel shuffle [141], are able to obtain the global information with a larger kernel size, but learn the same kernel (indicated by the white arrows in Figure 6.3(c)) across the image, regardless of the semantic information. Other feature enhancement methods, such as spatial attention [38], can learn different kernels (indicated by different colored

arrows in Figure 6.3(d)), but they still inevitably capture a lot of redundant information, i.e., 'grass' and 'soil'. Moreover, they are prohibitively expensive since they must consider all pixels.

To address these limitations, we further propose a novel semantic-aware upsampling (SAU) for this challenging task, as shown in Figure 6.3(e). Our SAU dynamically upsamples a small subset of relevant pixels based on the semantic information, i.e., the green and tangerine circles represent the pixels within the dog and the car, respectively. In this way, SAU is more efficient than deconvolution, pixel shuffle, and spatial attention, and can capture more complete semantic information than traditional upsampling methods such as nearest-neighbor interpolation.

Overall, the contributions of this paper are as follows:

- We explore image generation from local context, which we believe is beneficial for generating richer details compared with the existing global image-level generation methods. A new local class-specific generative structure is designed for this purpose. It can effectively handle the generation of small objects and details, which are common difficulties encountered by the global-based generation.
- We propose a novel global and local generative adversarial network design able to take into account both the global and local contexts. To stabilize the optimization of the proposed joint network structure, a fusion weight map generator and a dual-discriminator are introduced. Moreover, to learn discriminative class-specific feature representations, a novel classification module is proposed.
- We introduce a novel semantic-aware upsampling (SAU) to dynamically upsample a small subset of relevant pixels based on the semantic information. SAU is more efficient than deconvolution, pixel shuffle, and spatial attention, and can capture more complete semantic information than traditional upsampling methods such as nearest-neighbor

interpolation.

- Experiments for cross-view image translation on the Dayton [179], CVUSA [192], and SVA [116] datasets, and semantic image synthesis on the Cityscapes [26], ADE20K [226], COCO-Stuff [11], DeepFashion [97], CelebAMask-HQ [75], and Facades [176] datasets demonstrate the effectiveness of the proposed method, and show significantly better results compared with state-of-the-art methods.

Part of the material presented here appeared in [169]. The current paper extends [169] in several ways. (1) We present a more detailed analysis of related works, including recently published works dealing with semantic-guided image generation and feature upsampling. (2) We propose a general and highly effective feature upsampling method, i.e., SAU. The proposed SAU has three advantages: i) Global view. Unlike traditional upsampling methods (e.g., nearest-neighbor) that only exploit local neighborhoods, SAU can aggregate semantic information in a global view. ii) Semantically adaptive. Instead of using a fixed kernel for all locations (e.g., deconvolution), SAU enables semantic class-specific upsampling by generating adaptive kernels for different locations. iii) Efficient. Unlike spatial attention, which uses a fully connected strategy to connect all pixels, SAU only considers the most relevant pixels, introducing little computational overhead. Equipped with this new upsampling method, our LGGAN proposed in [169] is upgraded to LGGAN++. (3) We conduct extensive ablation studies to demonstrate the effectiveness of the proposed SAU against other feature upsampling and enhancement methods. (4) We extend the quantitative and qualitative experiments by comparing the proposed LGGAN and SAU with very recent works on two image synthesis tasks with diverse scenarios. We observe that the proposed methods achieve consistent and substantial gains on nine public datasets.

Although existing approaches [16, 55, 119, 44, 96, 127] conducted inter-

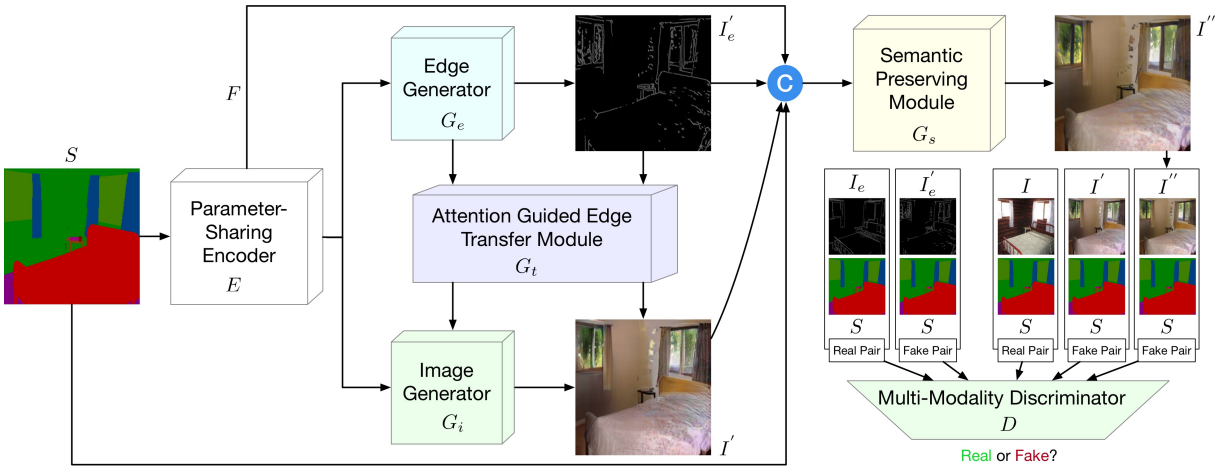


Figure 6.4: Overview of the proposed EdgeGAN. It consists of a parameter-sharing encoder E, an edge generator G_e, an image generator G_i, and a multi-modality discriminator D. The edge and image generators are connected by the proposed attention guided edge transfer module G_t from two levels, i.e., edge feature-level and edge content-level, to generate realistic images. The semantic preserving module G_s is proposed to preserve the semantic information of the input semantic labels. The discriminator D aims to distinguish the outputs from two modalities, i.e., edge and image. The whole framework can be end-to-end trained so that each component can benefit from each other. The symbol ⓒ denotes channel-wise concatenation.

esting explorations, we still observe unsatisfactory aspects mainly in the generated local structures and small-scale objects, which we believe are mainly due to two reasons. First, conventional methods [119, 184, 96] generally take the semantic label map as input directly. However, the input label map provides only structural information between different semantic-class regions and does not contain any structural information within each semantic-class region, making it difficult for synthesizing rich local structures within each class. Taking label map S in Figure 6.4 as an example, the generator does not have enough structural guidance to produce a realistic bed, window and curtain from only the input label (S). Second, the classic deep network architectures are constructed by stacking convolutional, down-sampling, normalization, non-linearity, and up-sampling layers, which will cause the problem of spatial resolution losses of the input semantic labels.

To address both issues, in this paper, we propose a novel Edge guided Generative Adversarial Network (EdgeGAN) for the semantic image synthesis. The overall framework of the proposed EdgeGAN is shown in Figure 6.4. We first propose an edge generator to produce the edge features and edge maps, and then the generated edge features and edge maps are selectively transferred to the image generator to improve the quality of the image results by using the proposed attention guided edge transfer module. Moreover, to tackle the issue of the spatial resolution losses caused by the common operations in the deep networks, we propose an effective semantic preserving module, which aims at selectively highlighting class-dependent feature maps according to the original semantic layout. Finally, we develop a multi-modality discriminator to simultaneously distinguish the output from two modal spaces, i.e., the edge and the image space. All the proposed modules are jointly optimized in an end-to-end fashion so that each module can benefit from each other in the training.

To summarize, our contributions are as follows:

- We propose a novel Edge Guided GAN (EdgeGAN) for the challenging semantic image synthesis task. To the best of our knowledge, we are the first to explore the edge generation from semantic layouts and then utilize the generated edges to guide the generation of realistic images.
- We propose an effective attention guided edge transfer module to selectively transfer useful edge structure information from the edge generation branch to the image generation branch. We also design a new semantic preserving module to highlight class-dependent feature maps based on the input semantic label map for generating semantically consistent results. Both ideas have not been investigated by any existing GAN-based generation works.
- We conduct extensive experiments on six challenging datasets under diverse scenarios, i.e., Cityscapes [26], ADE20K [226], COCO-

Stuff [11], Facades [176], CelebAMask-HQ [75], and DeepFashion [97]. Both qualitative and quantitative results show that EdgeGAN is able to produce remarkably better results than existing baseline models, regarding both the visual fidelity and the alignment with the input semantic labels. Moreover, our method can generate multi-modal images and edges, which has not been considered by existing methods. The code will be made publicly available.

6.2 Related Work

Generative Adversarial Networks (GANs) [41] have been widely used for image generation [63, 140]. A vanilla GAN has two important components, i.e., a generator and a discriminator. The goal of the generator is to synthesize photorealistic images from a noise vector, while the discriminator tries to distinguish between the real and the generated images. To create user-specific images, the conditional GAN (CGAN) [103] was proposed. A CGAN combines a vanilla GAN and external information, such as class labels [114, 25], text descriptions [213, 78], object keypoints [132, 164], human body/hand skeletons [156, 163], semantic maps [184, 165], scene graphs [61], or attention maps [100, 160].

Image-to-Image Translation aims to generate the target image based on an input image. CGANs have achieved decent results in both paired [55, 2] and unpaired [40, 231] image translation tasks. For instance, Isola et al. propose Pix2pix [55], which employs a CGAN to learn a translation mapping from input to output image domains such as map-to-photo and day-to-night. To further improve the quality of the generated images, the attention mechanism has been recently investigated in image translation tasks [165, 66, 100, 20, 197].

Different from previous attention-related image generation works, we

propose a novel attention guided edge transfer module to transfer useful edge structure information from the edge generation branch to the image generation branch at two different levels, i.e., feature level and content level. To the best of our knowledge, our module is the first attempt to incorporate both edge feature attention and edge content attention within a GAN framework for image-to-image translation tasks.

Global and Local Generation in GANs. Modeling global and local information in GANs to generate better results has been explored in various generative tasks [51, 53, 89, 84, 126, 44]. For instance, Huang et al. [51] proposed TPGAN for frontal view synthesis by simultaneously perceiving global structures and local details. Gu et al. [44] proposed MaskGAN for face editing by separately learning every face component, e.g., mouth and eye. However, these methods have only been applied to face-related tasks, such as face rotation or face editing, where all the domains have large overlap and similarity. In contrast, we propose a new local and global image generation framework for the more challenging scene image generation task, where the local context modeling is based on semantic-guided class-specific generation, which has not been explored by any existing works.

Semantic-Guided Image Generation. Scene generation tasks are a hot topic as each image can be parsed into distinctive semantic objects. In this paper, we mainly focus on two image generation tasks, i.e., cross-view image translation [211, 133, 134, 165] and semantic image synthesis [184, 16, 127, 119]. Most existing works on cross-view image translation aim to synthesize novel views of the same objects [32, 229, 172]. For instance, Dosovitskiy et al. [32] used generative models to produce unseen views of cars, chairs, and tables. Moreover, several works deal with image translation problems with drastically different views and generate a novel scene from another given scene [211, 133, 92, 165]. For instance, Zhai et al. [211] tried to generate panoramic ground-level images from aerial images

of the same location. Tang et al. proposed SelectionGAN [165] to solve the cross-view image translation task using semantic maps and CGAN models. On the other side, the semantic image synthesis task aims to generate a photorealistic image from a semantic map [184, 16, 119]. For example, Park et al. proposed GauGAN [119], which achieves the best results on this task.

With the semantic maps as guidance, existing approaches for both tasks achieve promising performance. However, the results produced by these global image-level generation methods are still often unsatisfactory, especially for detailed local textures. In contrast, our proposed approach focuses on generating a more realistic global structure/layout and local texture details. Both the local and global generation branches are jointly learned in an end-to-end fashion, enabling both to be improved through their mutually beneficial information.

Feature Upsampling. Traditional upsampling methods such as nearest-neighbor and bilinear interpolation use spatial distance and handcrafted kernels to capture the correlations between pixels. Recently, several deep learning methods, such as deconvolution [111] and pixel shuffle [141], have been proposed to upsample feature maps using learnable kernels. For instance, pixel shuffle [141] tries to reshape depth on the channel dimension into width and height on the spatial dimension. However, these methods either exploit semantic information in a small neighborhood or use a fixed kernel. Other works, including super-resolution [59, 50], inpainting [181], and denoising [102], also explore the use of learnable kernels.

Existing image generation methods, such as [119, 58, 154], typically adopt nearest-neighbor interpolation to upsample feature maps and then generate final results. However, this leads to unsatisfactory results, paticularly in the generated content details and intra-object completions. To adress this limitation, we propose a novel semantic-aware upsampling for

this task. To the best of our knowledge, we are the first to investigate the influence of feature upsampling on this challenging task.

Edge Guided Image Generation. Edge maps are usually adopted in image inpainting [136, 107, 81] and image super-resolution [108] tasks to reconstruct the missing structure information of the inputs. For example, Nazeri et al. [107] propose an edge generator to hallucinate edges in the missing regions given edges, which can be regarded as an edge completion problem. Using edge images as the structural guidance, [107] achieves good results even for some highly structured scenes. Moreover, Pix2pix [55] adopts edge maps as input and aims to generate realistic shoes and handbags, which can be seen as an edge-to-image translation problem.

Different from previous works, we propose a novel edge generator to perform a new task, i.e., semantic label-to-edge translation. To the best of our knowledge, we are the first work to generate realistic edge maps from semantic labels. Then the generated edge maps, with more local structure information, can be used to improve the quality of the image results.

Semantic Image Synthesis aims to generate a photo-realistic image from a semantic label map [184, 16, 127, 119, 96, 9, 233, 112, 235, 169]. With the semantic information as guidance, existing methods have achieved promising performance. However, we can still observe unsatisfying aspects, especially on the generation of the small-scale objects, which we believe is mainly due to the problem of spatial resolution losses associated with deep network operations such as convolution, normalization and down-sampling, etc. To solve this problem, Park et al. propose GauGAN [119], which uses the input semantic labels to modulate the activations in normalization layers through a spatially-adaptive transformation. However, the spatial resolution losses caused by other operations such as convolution and down-sampling have not been resolved. Moreover, we observe that the input label map has only a few semantic classes in the entire dataset. Thus the gener-

ator should focus more on learning these existing semantic classes rather than all the semantic classes.

To tackle both limitations, we propose a novel semantic preserving module, which aims to selectively highlight class-dependent feature maps according to the input labels for generating semantically consistent images. This idea has not been considered by existing GAN-based semantic image synthesis models.

6.3 LGGAN Model Description

We start by presenting the details of the proposed LGGAN and SAU. We first introduce the used backbone structure and then present the proposed semantic-aware upsampling, and finally introduce the design of the proposed local and global generation networks.

6.3.1 Backbone Encoding Network Structure

Semantic-Guided Generation. In this paper, we focus on two main tasks, i.e., semantic image synthesis and cross-view image translation. For the former, we follow GauGAN [119] and use the semantic map S_g as the input of the backbone encoder E, as shown in Figure 6.2. For the latter, we follow SelectionGAN [165] and concatenate the input image I_a and a novel semantic map S_g as the input of the backbone encoder E. By so doing, the semantic maps act as priors to guide the model in learning the generation of another domain.

Parameter-Sharing Encoder. As we have three different branches for three different generators, the encoder E shares parameters with all the branches to create a compact backbone network. The gradients from each branch all contribute to the learning of the encoder. We believe that, in this way, the encoder can learn both local and global information and the

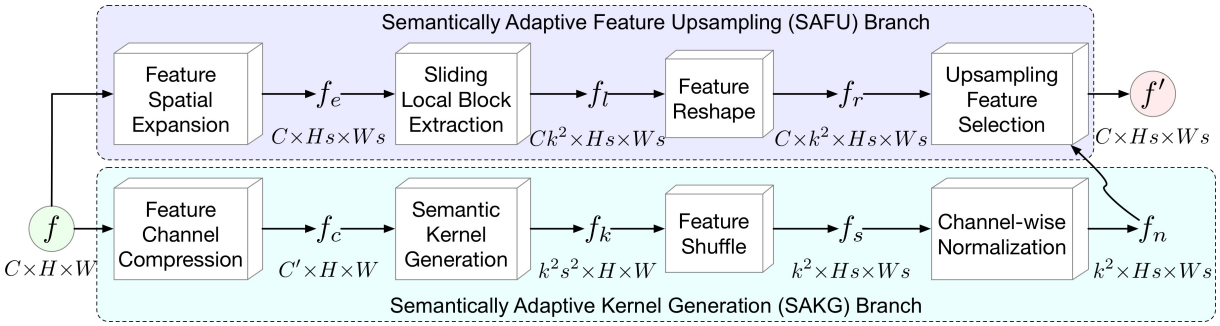


Figure 6.5: Overview of the proposed SAU, which consists of two branches, i.e., SAKG and SAFU. The SAKG branch aims to generate semantically adaptive kernels according to the input layout. The SAFU branch aims to selectively upsample the feature $f\in C\times H\times W$ to the target one $f'\in C\times Hs\times Ws$ based on the kernels learned in SAKG, where s is the expected upsample scale.

correspondence between them. Thus, the encoded deep representation of the input S_g can be represented as $f=E(S_g)$. This feature f is then fed into the proposed SAU to obtain an upsampled feature map $f'=\text{SAU}(f)$, as shown in Figures 6.2 and 6.5.

6.3.2 Semantic-Aware Upsampling

An illustration of the proposed SAU is shown in Figure 6.5. It consists of two main branches, i.e., the semantically adaptive kernel generation (SAKG) branch, which predicts upsampled kernels according to the semantic information, and the semantically adaptive feature upsampling (SAFU) branch, whcih selectively performs the feature upsampling based on the kernels learned in SAKG. All components are trained in an end-to-end fashion.

Specifically, given a feature map $f\in\mathbb{R}^{C\times H\times W}$ and an upsample scale s, SAU aims to produce a new feature map $f'\in\mathbb{R}^{C\times Hs\times Ws}$. For any target location $l'=(i',j')$ in the output f', there is a corresponding source location $l=(i,j)$ at the input f, where $i=\lfloor i'/s\rfloor$, $j=\lfloor j'/s\rfloor$. We denote $Z(l,k)$ as the $k\times k$ subregion of f centered at the location l in, e.g., the neighbor of the location l. See Figures 6.3 and 6.5 for illustration.

Semantically Adaptive Kernel Generation

The SAKG branch aims to generate a semantically adaptive kernel at each location, according to the semantic information. It consists of four modules, i.e., feature channel compression, semantic kernel generation, feature shuffle, and channel-wise normalization.

Feature Channel Compression. This module is used to reduce the network parameters and computational cost. Specifically, the input feature f is fed into a convolutional layer with a 1×1 kernel to compress the input channel from C to C', making SAU have fewer parameters.

Semantic Kernel Generation. This module receives $f_c \in \mathbb{R}^{C'\times H\times W}$ as input (where H and W denote the height and width of the feature map) and tries to generate different semantically adaptive kernels, which can be represented as $f_k \in \mathbb{R}^{k^2s^2\times H\times W}$. Here k is the semantically adaptive upsampling kernel size and s is the expected upsample scale. In our experiments, we set C'=64, k=5, and s=2, which achieve good results in most cases.

Feature Shuffle. We then feed the feature f_k through a feature shuffle layer to rearrange its elements, leading to a new feature map $f_s \in \mathbb{R}^{k^2\times Hs\times Ws}$, where k^2=$k\times k$ represents the learned semantic kernel. Note that the learned semantic kernels are quite different at different locations l', as shown in Figures 6.3 and 6.6.

Channel-Wise Normalization. Next, we apply a channel-wise softmax operation on each semantic kernel f_s to obtain the normalized kernel f_n, i.e., the sum of the weight values in k^2 is equal to 1. In this way, we can guarantee that information from the combination will not explode. Moreover, the semantically adaptive kernels can determine which regions to emphasize or suppress according to the semantic information.

Semantically Adaptive Feature Upsampling

The SAFU branch aims to upsample the input feature f based on the kernel f_n learned in the SAKG branch, in a semantically adaptive way. It contains four modules, i.e., feature spatial expansion, sliding local block extraction, feature reshape, and upsampling feature selection.

Feature Spatial Expansion. The input feature f is fed into this module to expand the spatial size from $H \times W$ to $Hs \times Ws$.

Sliding Local Block Extraction. Then, the expanded feature $f_e \in \mathbb{R}^{C \times Hs \times Ws}$ is fed into this module to extract a sliding local block of each location in f_e, leading to the new feature $f_l \in \mathbb{R}^{Ck^2 \times Hs \times Ws}$.

Feature Reshape. We next reshape f_l by adding another dimension, resulting in a new feature $f_r \in \mathbb{R}^{C \times k^2 \times Hs \times Ws}$. In this way, we can conduct a multiplication between the reshaped local block f_r and the learned kernel f_n.

Upsampling Feature Selection. Finally, the feature map f_r and the kernel f_n learned in the SAKG branch are fed into the upsampling feature selection module to generate the final feature map $f' \in \mathbb{R}^{C \times Hs \times Ws}$ in a weighted sum manner. The computational process at the location $l=(i,j)$ can be expressed as follows:

$$f' = \sum_{p=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} \sum_{q=-\lfloor k/2 \rfloor}^{\lfloor k/2 \rfloor} f_r(i+p, j+q) \times f_n(p,q). \tag{6.1}$$

In this way, the pixels in the learned kernel f_n contribute to the upsampled pixel l' differently, based on semantic information of features instead of the spatial distance between locations. The semantics of the upsampled feature map are stronger than those of the original one, since the information from relevant points in a local region can be more attended to, and the pixels with the same semantic label can achieve mutual gains, improving intra-

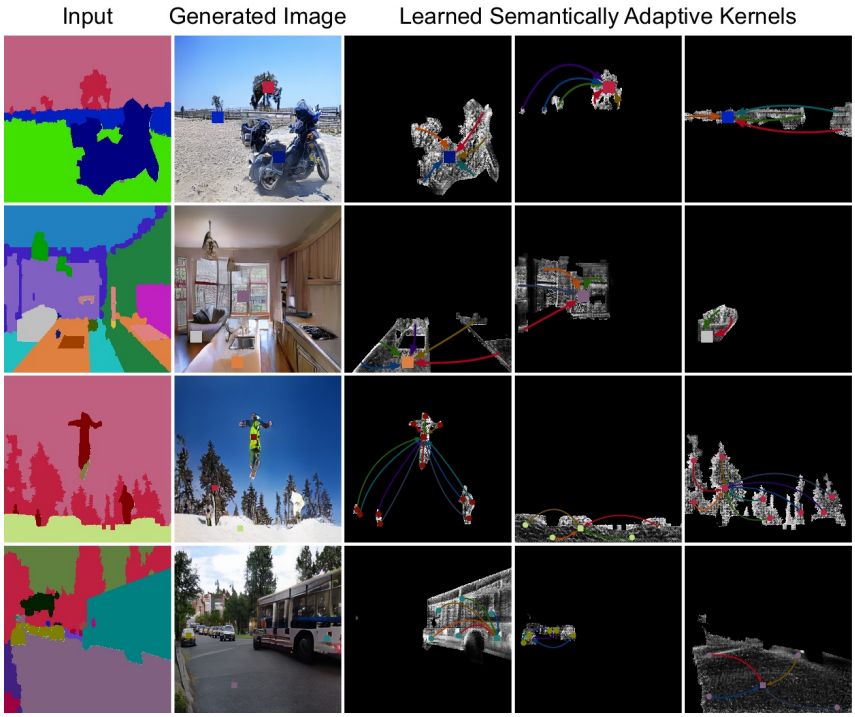


Figure 6.6: Visualization of semantically adaptive kernels learned on COCO-Stuff. In the second column, we show three representative locations in each generated image, with different colored squares. The other three columns show semantically adaptive kernels learned for those three locations, with corresponding color arrows summarizing the most-attended regions for upsampling the target location. The network learns to allocate attention according to regions with the same semantic information, rather than just spatial adjacency.

object semantic consistency.

Why Does the SAU Work Better?

A short answer is that it can better preserve semantic information compared to other common upsampling methods. Specifically, while other methods, such as nearest-neighbor interpolation and deconvolution, are essential parts in almost all state-of-the-art image generation [131] and translation [119] models, they tend to 'pollute' semantic information when performing feature upsampling, since they inevitably incorporate contaminating information from irrelevant regions (see Figure 6.3).

In contrast, the proposed SAU performs feature upsampling using itself,

i.e., it uses the pixels belonging to the same semantic label to upsample the feature maps. Hence, the generator can better preserve semantic information. It enjoys the benefit of feature upsampling without losing the input semantic information. In Figure 6.6, we show some examples of the learned semantically adaptive kernels. We can easily observe that the proposed SAU achieves upsampling by leveraging complementary features from regions with the same semantic information, rather than local regions of fixed shape to generate consistent objects/scenarios. This further confirms our motivations.

6.3.3 Local and Global GAN

An illustration of the overall framework is shown in Figure 6.2. The generation module consists of three main parts, i.e., a semantic-guided class-specific generator modeling the local context, an image-level generator modeling the global layout, and a weight-map generator for fusing the local and the global generators.

Class-Specific Local Generation Network. As shown in Figure 6.1 and discussed in the introduction, the training data imbalance between different classes and the size difference between semantic objects makes it extremely difficult to generate small object classes and fine details. To address this, we propose a novel local class-specific generation network. It constructs a separate generator for each semantic class, thus being able to largely avoid the interference from large object classes during the joint optimization. Each subgeneration branch has independent network parameters and concentrates on a specific class, therefore being capable of effectively producing similar generation quality for different classes and yielding richer local image details.

An overview of the local generation network G_l is provided in Figure 6.7. The upsampled feature map f' is multiplied by the semantic mask of each

class, i.e., M_i, to obtain a filtered class-specific feature map for each one. The mask-guided feature filtering operation can be written as:

$$\mathbf{F}_i = M_i * f', \quad i = 1, 2, ..., N, \tag{6.2}$$

where N is the number of semantic classes. Then, the filtered feature map $\mathbf{F}_i$ is fed into several convolutional layers for the corresponding i^{th} class, which generates an output image $I^l_{g_i}$. To better learn each class, we utilize a semantic mask guided pixel-wise $L1$ reconstruction loss, which can be expressed as follows:

$$\mathcal{L}^{local}_{\text{L1}} = \sum_{i=1}^{N} \mathbb{E}_{I_g, I^l_{g_i}}[||I_g * M_i - I^l_{g_i}||_1]. \tag{6.3}$$

The final output I^L_g from the local generation network can be obtained in two ways. The first is by performing an element-wise addition of all the class-specific outputs:

$$I^L_g = I^l_{g_1} \oplus I^l_{g_2} \oplus \cdots \oplus I^l_{g_N}. \tag{6.4}$$

The second is by applying a convolutional operation on all the class-specific outputs, as shown in Figure 6.7:

$$I^L_g = \text{Conv}(\text{Concat}(I^l_{g_1}, I^l_{g_2}, \cdots, I^l_{g_N})), \tag{6.5}$$

where Concat$(\cdot)$ and Conv$(\cdot)$ denote a channel-wise concatenation and convolutional operation, respectively.

Class-Specific Discriminative Feature Learning. We observe that the filtered feature map $\mathbf{F}_i$ is not able to produce very discriminative class-specific generations, leading to similar generation results for some classes, especially for small-scale object classes. To provide a more diverse generation for different object classes, we propose a novel classification-based

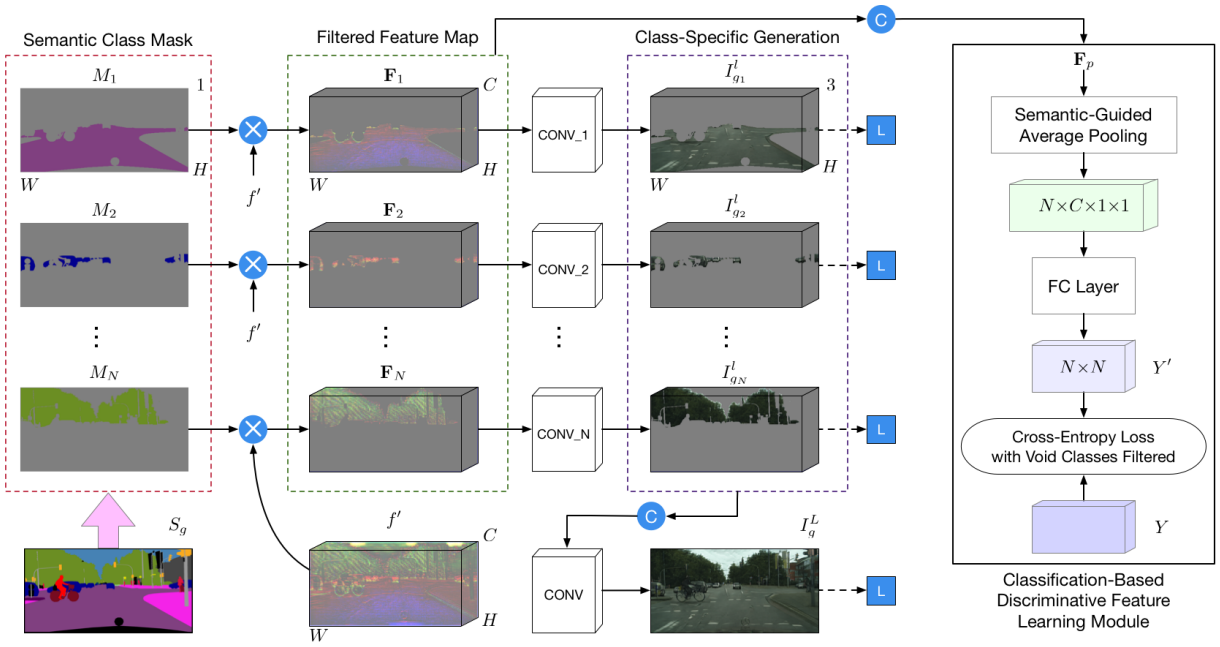


Figure 6.7: Overview of the proposed local class-specific generator G_l, which consists of four steps, i.e., semantic class mask calculation, class-specific feature map filtering, classification-based discriminative feature learning and class-specific generation. A cross-entropy loss with void classes filtered is applied to the feature representation of each class to learn more discriminative class-specific representations. A semantic-mask guided pixel-wise $L1$ loss is applied at the end for class-level reconstruction. The symbols $\otimes$ and ⓒ denote element-wise multiplication and channel-wise concatenation, respectively. Note that we assume the size of f' is $C\times H\times W$ for simplicity, which is different from the one in Figure 6.5 (i.e., $C\times Hs\times Ws$).

feature learning module to learn more discriminative class-specific feature representations, as shown in Figure 6.7. One input sample of the module is a stack of feature maps produced by different local generation branches, i.e., $\{\mathbf{F}_1, ..., \mathbf{F}_N\}$. First, the stacked feature map $\mathbf{F}_p \in \mathbb{R}^{N\times C\times H\times W}$ (with C, H, W as the number of feature map channels, height and width, respectively) is fed into a semantic-guided average pooling layer, and we obtain a pooled feature map with dimensions of $N\times C\times 1\times 1$. Then the pooled feature map is fed to a fully connected (FC) layer to predict the classification probabilities of the N object classes of the image. The output after the FC layer is $Y' \in \mathbb{R}^{N\times N}$, where N is the number of semantic classes. Since, for each filtered feature map $\mathbf{F}_i$ $(i=1, ..., N)$, we predict an $N\times 1$ one-hot

vector for the probabilities of the N classes.

Since some object classes may not exist in the input semantic mask, the features from the local branches corresponding to the void classes should not contribute to the classification loss. Therefore, we filter the final cross-entropy (CE) loss by multiplying it with a void class indicator for each input sample. The indicator is a one-hot vector $H=\{H_i\}_{i=1}^{N}$, with $H_i=1$ for a valid class and $H_i=0$ for a void class. Then, the CE loss is defined as follows:

$$\mathcal{L}_{\text{CE}} = -\sum_{m=1}^{N} H_m \sum_{i=1}^{N} 1\{Y(i)=i\} \log(f(\mathbf{F}_i)), \tag{6.6}$$

where $1\{\cdot\}$ is an indicator function having a return of 1 if $Y(i)=i$, otherwise 0. $f(\cdot)$ is a classification function which produces a prediction probability given an input feature map $\mathbf{F}_i$. Y is a label set of all object classes.

Image-Level Global Generation Network. Similar to the local generation branch, the upsampled feature map f' is also fed into the global generation subnetwork G_g for image-level generation, as shown in Figure 6.2. Global generation is capable of capturing the global structural information or layout of the target images. Thus, the global result I_g^G can be obtained through a feed-forward computation: $I_g^G=G_g(f')$. Besides the proposed G_g, many existing global generators can also be used together with the local generator G_l, making the proposed framework very flexible.

Pixel-Level Fusion Weight-Map Generation Network. To better combine the local and global generation subnetworks, we further propose a pixel-level weight map generator G_w, which aims at predicting pixel-wise weights to fuse the global generation I_g^G and the local generation I_g^L. G_w consists of two Transpose Convolution→InstanceNorm→ReLU blocks and one Convolution→InstanceNorm→ReLU block. The number of output channels for these three blocks are 128, 64, and 2, respectively. The kernel sizes are 3×3 with stride 2, except for the last layer which has a kernel size

of 1×1 with stride 1 for dense prediction. We predict a two-channel weight map W_f using the following calculation:

$$W_f = \text{Softmax}(G_w(f')), \tag{6.7}$$

where Softmax$(\cdot)$ denotes a channel-wise softmax function used for normalization, i.e., the sum of the weight values at the same pixel position is equal to 1. By so doing, we can guarantee that information from the combination will not explode. W_f is sliced to have a weight map W_g for the global branch and a weight map W_l for the local branch. The final fused generation result is calculated as follows:

$$I_g^C = I_g^G \otimes W_g + I_g^L \otimes W_l, \tag{6.8}$$

where $\otimes$ is an element-wise multiplication operation. In this way, the pixel-level weights predicted by G_w directly operate on the output of G_g and G_l. Moreover, the generators G_w, G_g and G_l affect and contribute to each other during the model optimization.

Dual-Discriminator. To exploit prior domain knowledge, i.e., the semantic map, we extend the single-domain vanilla discriminator [41] to a cross-domain structure, which we refer to as the semantic-guided discriminator D_s, as shown in Figure 6.2. It takes the semantic map S_g and generated image I_g^C (or the real image I_g) as inputs:

$$\mathcal{L}_{\text{CGAN}}(G, D_s) = \mathbb{E}_{S_g, I_g}\left[\log D_s(S_g, I_g)\right] + \mathbb{E}_{S_g, I_g^C}\left[\log(1 - D_s(S_g, I_g^C))\right], \tag{6.9}$$

aiming to preserve image layout and capture the local information.

For the cross-view image translation task, we also propose another image-guided discriminator D_i, which takes the conditional image I_a and

the final generated image I_g^C (or the ground-truth image I_g) as input:

$$\mathcal{L}_{\text{CGAN}}(G, D_i) = \mathbb{E}_{I_a, I_g}\left[\log D_i(I_a, I_g)\right] + \mathbb{E}_{I_a, I_g^C}\left[\log(1 - D_i(I_a, I_g^C))\right]. \tag{6.10}$$

In this case, the overall loss of our dual-discriminator D can be expressed as:

$$\mathcal{L}_{\text{CGAN}} = \mathcal{L}_{\text{CGAN}}(G, D_i) + \mathcal{L}_{\text{CGAN}}(G, D_s). \tag{6.11}$$

6.4 EdgeGAN Model Description

In this section, we describe the proposed Edge Guided GAN (EdgeGAN) for semantic image synthesis. We first introduce an overview of the proposed EdgeGAN, and then introduce the details of each module. Finally, we present the optimization objective.

Framework Overview. Figure 6.4 shows the overall structure of our EdgeGAN for semantic image synthesis, which consists of a semantic and edge guided generator G and a multi-modality discriminator D. The generator G consists of five components: (i) a parameter-sharing convolutional encoder E is proposed to produce deep feature maps F; (ii) an edge generator G_e is adopted to generate edge maps $I_e^{'}$ taking as input deep features from the encoder; (iii) an image generator G_i is used to produce intermediate images $I^{'}$; (iv) an attention guided edge transfer module G_t is designed to forward useful structure information from the edge generator to the image generator; and (v) the semantic preserving module G_s is developed to selectively highlight class-dependent feature maps according to the input label for generating semantically consistent images $I^{''}$. Meanwhile, to effectively train the network, we propose a multi-modality discriminator D that can simultaneously distinguish the outputs from two modalities, i.e., edge and image.

EdgeGAN takes a semantic layout as input and outputs a semantically correspondent photo-realistic image. During training, the ground truth edge map is extracted from corresponding ground truth images with a Canny edge detector [12].

6.4.1 Edge Guided Semantic Image Synthesis

Parameter-Sharing Encoder. The backbone encoder E can employ any deep network architecture, such as the commonly used AlexNet [72], VGG [148], and ResNet [47]. We directly utilize the feature maps from the last convolutional layer as deep feature representations, i.e., $F{=}E(S)$, where E represents the encoder; $S{\in}\mathbb{R}^{N\times H\times W}$ is the input label, with H and W as width and height of the input semantic labels, and N as the total number of semantic classes. Optionally, one can always combine multiple intermediate feature maps to enhance the feature representation. The encoder is shared by the edge generator and the image generator. Then, the gradients from the two generators all contribute to updating the parameters of the encoder. This compact design can potentially enhance the deep representations as the encoder can simultaneously learn structure representations from the edge generation branch and appearance representations from the image generation branch.

Edge Guided Image Generation. As discussed, the lack of detailed structure or geometry guidance makes it extremely difficult for the generator to produce realistic local structures and details. To overcome this limitation, we propose to adopt the edge as guidance. A novel edge generator G_e is designed to directly generate the edge maps from the input semantic labels. This also facilitates the shared encoder to learn more local structures of the targeted images. Meanwhile, the image generator G_i aims to generate photo-realistic images from the input labels. In this way, the encoder is boosted to learn the appearance information of the targeted

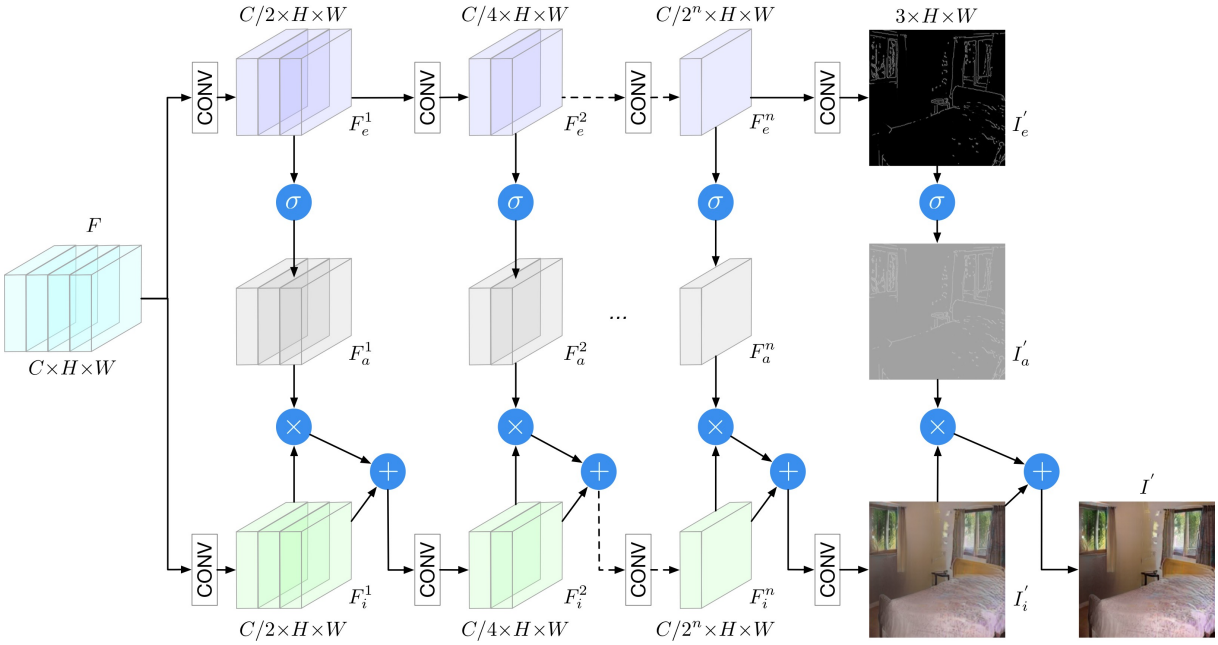


Figure 6.8: Structure of the proposed edge generator G_e (*first row*), the proposed attention guided edge transfer module G_t (*middle row*) and the proposed image generator G_i (*third row*). The edge generator G_e selectively transfers useful local structure information to the image generator G_i using the proposed attention guided transfer module G_t. The symbols $\oplus$, $\otimes$, and $\textcircled{\sigma}$ denote element-wise addition, element-wise multiplication, and Sigmoid activation function, respectively.

images.

Previous works [119, 96, 127, 16, 184] directly use deep networks to generate the target image, which is challenging since the network needs to simultaneously learn appearance and structure information from the input labels. In contrast, our EdgeGAN separately learns structure and appearance via the proposed edge generator and image generator. Moreover, the explicit guidance from the ground truth edge maps can also facilitate the training of the encoder.

The framework of the proposed edge and image generators are illustrated in Figure 6.8. Given the feature maps from the last convolutional layer of the encoder, i.e., $F \in \mathbb{R}^{C \times H \times W}$, where H and W are the width and height of the features, and C is the number of channels, the edge generator produces edge features and edge maps which are further utilized to guide

the image generator to generate the intermediate image $I^{'}$.

The edge generator G_e contains n convolution layers and correspondingly produces n intermediate feature maps F_e=$\{F_e^j\}_{j=1}^n$. After that, another convolution layer with Tanh non-linear activation is utilized to generate the edge map $I_e^{'}$∈$\mathbb{R}^{3\times H\times W}$. Meanwhile, the feature maps F is also fed into the image generator G_i to generate n intermediate feature maps F_i=$\{F_i^j\}_{j=1}^n$. Then another convolution operation with Tanh non-linear activation is adopted to produce the intermediate image $I_i^{'}$∈$\mathbb{R}^{3\times H\times W}$. In addition, the intermediate edge feature maps F_e and the edge map $I_e^{'}$ are utilized to guide the generation of the image feature maps F_i and the intermediate image $I^{'}$ via the Attention Guided Edge Transfer as detailed below.

Attention Guided Edge Transfer. We further propose a novel attention guided edge transfer module G_t to explicitly employ the edge structure information to refine the intermediate image representations. The architecture of the proposed transfer module G_t is illustrated in Figure 6.8.

To transfer useful structure information from edge feature maps F_e=$\{F_e^j\}_{j=1}^n$ to the image feature maps F_i=$\{F_i^j\}_{j=1}^n$, the edge feature maps are firstly processed by Sigmoid activation function to generate the corresponding attention maps F_a=Sigmoid(F_e)=$\{F_a^j\}_{j=1}^n$. Then, we multiply the generated attention maps with the corresponding image feature maps to obtain the refined maps which incorporate local structures and details as in Equation (6.12). Finally, the edge refined features are element-wisely summed with the original image features to produce the final edge refined features, which are further fed to the next convolution layer.

$$F_i^j = \text{Sigmoid}(F_e^j) \times F_i^j + F_i^j, \quad \text{for } j = 1, \cdots, n \tag{6.12}$$

In this way, the image feature maps also contain the local structure information provided by the edge feature maps. Similarly, to directly employ

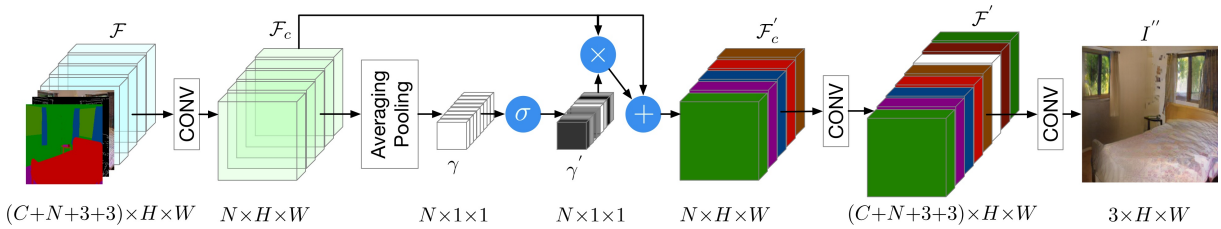


Figure 6.9: Overview of the proposed semantic preserving module G_s, which aims at capturing the semantic information and predicts scaling factors conditioned on the combined feature maps $\mathcal{F}$. These learned factors selectively highlight class-dependent feature maps, which are visualized in different colors. The symbols $\oplus$, $\otimes$, and $\textcircled{\sigma}$ denote element-wise addition, element-wise multiplication, and Sigmoid activation function, respectively.

the structure information from the generated edge map $I_e^{'}$ for image generation, we adopt the attention guided edge transfer module to refine the generated image directly with edge information as in Equation (6.13).

$$I^{'} = \text{Sigmoid}(I_e^{'}) \times I_i^{'} + I_i^{'}, \tag{6.13}$$

where $I_a^{'}$=Sigmoid($I_e^{'}$) is the generated attention map. We also provide the visualization results of Equation (6.13) in Figure 6.28 (right).

6.4.2 Semantic Preserving Image Enhancement

Due to the spatial resolution loss caused by convolution, normalization and down-sampling layers, existing models [184, 119, 127, 16] cannot fully preserve the semantic information of the input labels as illustrated in Figure 6.27. For instance, the small 'pole' is missing and the large 'fence' is incomplete. To tackle this problem, we propose a novel semantic preserving module, which aims to select class-dependent feature maps and further enhance it through the guidance of the original semantic layout. An overview of the proposed semantic preserving module G_s is shown in Figure 6.9.

Specifically, the input of the module, denoted as $\mathcal{F}$, is the concatenation of the input label S, the generated intermediate edge map $I_e^{'}$ and image $I^{'}$,

and the deep feature F produced from the shared encoder E.

Then, we apply a convolution operation on $\mathcal{F}$ to produce a new feature map $\mathcal{F}_c$ with the number of channels equal to the number of semantic categories, where each channel corresponds to a specific semantic category (a similar conclusion can be found in [38]). Next, we apply the averaging pooling operation on $\mathcal{F}_c$ to obtain the global information of each class followed by a Sigmoid activation function to derive scaling factors $\gamma^{'}$ as in Equation (6.14), where each value represents the importance of the corresponding class.

$$\gamma^{'} = \text{Sigmoid}(\text{AvgPool}(\mathcal{F}_c)). \tag{6.14}$$

Then, the scaling factor $\gamma^{'}$ is adopted to reweight the feature map $\mathcal{F}_c$ and highlight corresponding class-dependent feature maps as Equation (6.15). The reweighted feature map is further added with the original feature $\mathcal{F}_c$ to compensate information loss due to multiplication, and produces $\mathcal{F}_c^{'} \in \mathbb{R}^{N \times H \times W}$,

$$\mathcal{F}_c^{'} = \mathcal{F}_c \times \gamma^{'} + \mathcal{F}_c. \tag{6.15}$$

After that, we perform another convolution operation on $\mathcal{F}_c^{'}$ to obtain the feature map $\mathcal{F}^{'} \in \mathbb{R}^{(C+N+3+3) \times H \times W}$ to enhance the representative capability of the feature. In addition, $\mathcal{F}^{'}$ has the same size as the original input one $\mathcal{F}$, which makes the module flexible and can be plugged into other existing architectures without modifications of other parts to refine the output.

Finally, the feature map $\mathcal{F}^{'}$ is fed into a convolutional layer followed by a Tanh non-linear activation layer to obtain the final result $I^{''}$. The semantic preserving module enhances the representational power of the model by adaptively recalibrating semantic class-dependent feature maps, and shares similar spirits with style transfer [52], and recent works SENet [49] and EncNet [214]. One intuitive example of the utility of the module

is for the generation of small object classes: the small object classes are easily missed in the generation results due to spatial resolution loss while our scaling factor can put an emphasis on small objects and help preserve them.

6.4.3 Optimization Objective

Multi-Modality Discriminator. To facilitate the training of the proposed EdgeGAN for high-quality edge and image generation, a novel multi-modality discriminator is developed to simultaneously distinguish outputs from two modality spaces, i.e., edge and image. Since the edges and RGB images share the same structure, they can be learned using the multi-modality discriminator. In the preliminary experiment, we also tried to use two discriminators, but no performance improvement was observed while increasing the model complexity. Thus, we use the proposed multi-modality discriminator. The framework of the proposed multi-modality discriminator is shown in Figure 6.4, which is capable of discriminating both real/fake images and edges. To discriminate real/fake edges, the discriminator loss considering the semantic label S and the generated edge $I_e^{'}$ (or the real edge I_e) is as Equation (6.16) which guide the model to distinguish real edges from fake generated edges.

$$\mathcal{L}_{\mathrm{CGAN}}(G_e, D) = \mathbb{E}_{S,I_e}\left[\log D(S, I_e)\right] + \mathbb{E}_{S,I_e^{'}}\left[\log(1 - D(S, I_e^{'}))\right]. \quad (6.16)$$

Further, to discriminate real/fake images, the discriminator loss regarding semantic label S and the generated images $I^{'}$, $I^{''}$ (or the real image I) is as

Equation (6.17), which guide the model to discriminate real/fake images.

$$\begin{aligned}\mathcal{L}_{\text{CGAN}}(G_i, G_s, D) = (\lambda+1)\mathbb{E}_{S,I}\left[\log D(S,I)\right] + \mathbb{E}_{S,I'}\left[\log(1-D(S,I^{'}))\right] \\ + \lambda\mathbb{E}_{S,I''}\left[\log(1-D(S,I^{''}))\right],\end{aligned} \tag{6.17}$$

where λ controls the losses of the generated two images.

Optimization Objective. Equipped with the multi-modality discriminator, we elaborate on the training objective for the proposed method as follows. Three different losses, i.e., the multi-modality adversarial loss, the discriminator feature matching loss $\mathcal{L}_f$ and the perceptual loss $\mathcal{L}_p$, are used to optimize the proposed EdgeGAN,

$$\begin{aligned}\min_G \max_D \mathcal{L} = \lambda_c \underbrace{(\mathcal{L}_{\text{CGAN}}(G_e, D) + \mathcal{L}_{\text{CGAN}}(G_i, G_s, D))}_{\text{Multi-Modality Adversarial Loss}} \\ + \lambda_f \underbrace{(\mathcal{L}_f(I_e, I_e^{'}) + \mathcal{L}_f(I, I^{'}) + \lambda\mathcal{L}_f(I, I^{''}))}_{\text{Discriminator Feature Matching Loss}} \\ + \lambda_p \underbrace{(\mathcal{L}_p(I_e, I_e^{'}) + \mathcal{L}_p(I, I^{'}) + \lambda\mathcal{L}_p(I, I^{''}))}_{\text{Perceptual Loss}},\end{aligned} \tag{6.18}$$

where λ_c, λ_f and λ_p are three parameters of the corresponding loss that contributes to the total loss $\mathcal{L}$; where $\mathcal{L}_f$ matches the discriminator intermediate features between the generated images/edges and the real images/edges; where $\mathcal{L}_p$ matches the VGG [148] extracted features between the generated images/edges and the real images/edges. By maximizing the discriminator loss, the generator is promoted to simultaneously generate reasonable edge maps that can capture the local-aware structure information and generate photo-realistic images semantically aligned with the input semantic labels.

6.4.4 Implementation Details

We adopt GauGAN [119] as the structure of our encoder E, which is built on ResNet [47]. For both the image generator G_i and edge generator G_e, the kernel size and padding size of convolutions are all 3×3 and 1 for preserving the feature map size. We set n=3 for generators G_i, G_s and G_t. The channel size of feature F is set to C=64. For the semantic preserving module G_s, we adopt an adaptive average pooling operation. Spectral normalization [104] is applied to all the layers in both the generator and discriminator. We adopt the Canny edge detector [12] to extract ground truth edge maps for training. Also, we follow the training procedures of GANs [41] and alternatively train the generator G and discriminator D, i.e., one gradient descent step on discriminator and generator alternately. We use the Adam solver [68] and set β_1=0, β_2=0.999. λ_c, λ_f, and λ_p in Equation (6.18) is set to 1, 10, and 10, respectively. All λ in both Equations (6.17) and (6.18) are set to 2. We conduct the experiments on an NVIDIA DGX1 with 8 V100 GPUs.

6.5 LGGAN Experiments

6.5.1 Local and Global GAN

The proposed LGGAN can be applied to different generative tasks, such as cross-view image translation [165] and semantic image synthesis [119].

Cross-View Image Translation

Datasets and Evaluation Metrics. We follow [165, 134] and perform experiments on the SVA [116], Dayton [179], and CVUSA [192] datasets. Similar to [134, 165], we employ inception score (IS), Fréchet inception distance (FID) [48], accuracy, KL divergence score (KL), structural similarity

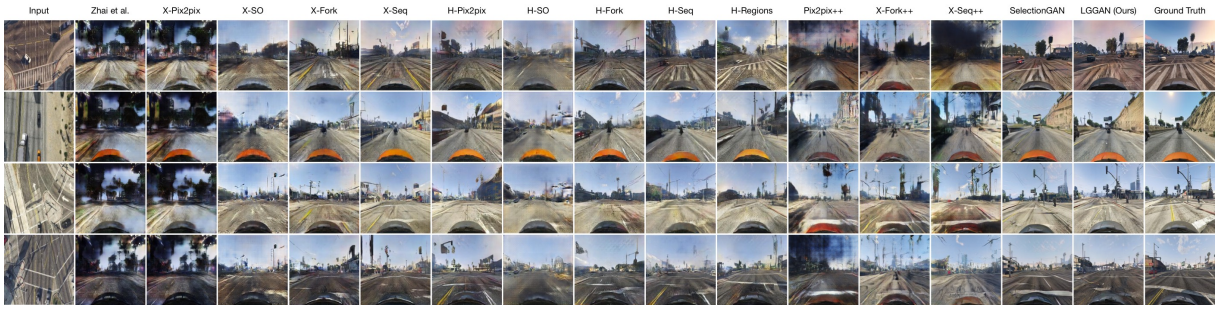


Figure 6.10: Qualitative comparison of cross-view image translation in a2g direction on SVA.

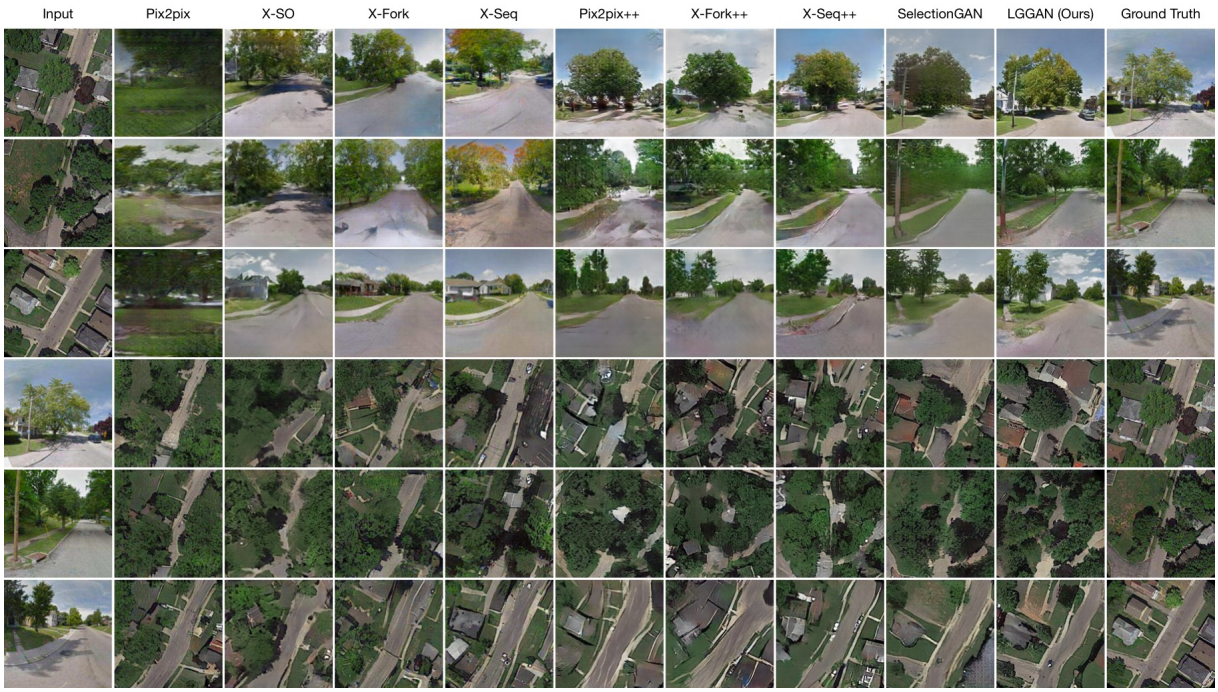


Figure 6.11: Qualitative comparison of cross-view image translation in both a2g (top three rows) and g2a (bottom three rows) directions on Dayton.

(SSIM), peak signal-to-noise ratio (PSNR), and sharpness difference (SD) to evaluate the proposed model.

State-of-the-Art Comparisons. We compare our LGGAN with several recently proposed state-of-the-art methods, i.e., Zhai et al. [211], Pix2pix [55], X-SO [134], X-Fork [133] and X-Seq [133]. The comparison results are shown in Tables 6.1, 6.2, and 6.3. We observe that the proposed LGGAN consistently outperforms the competing methods in all metrics.

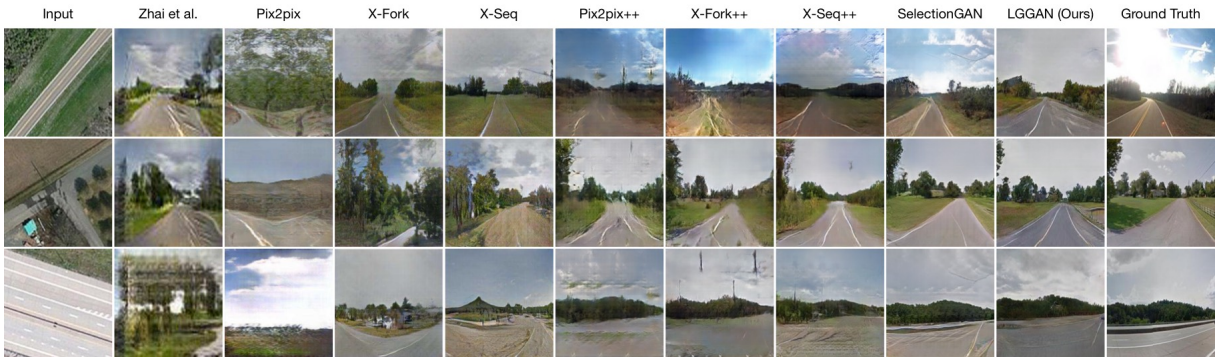


Figure 6.12: Qualitative comparison of cross-view image translation in a2g direction on CVUSA.

To study the effectiveness of our LGGAN, we conduct experiments and compare against using semantic maps and RGB images as input, Pix2pix++ [55], X-Fork++ [133], X-Seq++ [133], and SelectionGAN [165]. We implement Pix2pix++, X-Fork++, and X-Seq++ using their public source codes. The results are shown in Tables 6.1, 6.2, and 6.3. LGGAN achieves significantly better results than Pix2pix++, X-Fork++, and X-Seq++, confirming its advantage. A direct comparison with SelectionGAN is also shown in the tables, where our model provides better results on all metrics except those for pixel-level evaluation, i.e., SSIM, PSNR, and SD. SelectionGAN achieves slightly better results in these three metrics because it uses a two-stage generation strategy and an attention selection module. However, we generate much more photorealistic results than SelectionGAN, as shown in Figures 6.10, 6.11, and 6.12.

Qualitative Evaluation. The qualitative results of our model compared with the leading methods are shown in Figures 6.10, 6.11, and 6.12. The results generated by the proposed LGGAN are visually better than those provided by the existing methods. Specifically, our method generates clearer details for objects such as cars, buildings, roads, and trees.

Arbitrary Cross-View Image Translation. We also follow SelectionGAN [165] and show some results on arbitrary cross-view image translation

Method	Accuracy (%) ↑				Inception Score* ↑			SSIM ↑	PSNR ↑	SD ↑	KL ↓	FID ↓
	Top-1		Top-5		All	Top-1	Top-5					
X-Pix2pix [55]	8.5961	30.3288	9.0260	29.9102	2.0131	1.7221	2.2370	0.3206	17.9944	17.0254	19.5533	859.66
X-SO [134]	7.5146	30.9507	10.3905	38.9822	2.4951	1.8940	2.6634	0.4552	21.5312	17.5285	12.0906	443.79
X-Fork [133]	17.3794	53.4725	23.8315	63.5045	2.1888	1.9776	2.3664	0.4235	21.2400	16.9371	4.1925	129.16
X-Seq [133]	19.5056	57.1010	25.8807	65.3005	2.2232	1.9842	2.4344	0.4638	22.3411	17.4138	3.7585	118.70
H-Pix2pix [134]	18.0706	54.8068	23.4400	62.3072	2.1906	1.9507	2.4069	0.4327	21.6860	16.9468	4.2894	117.13
H-SO [134]	5.2444	26.4697	5.2544	31.9527	2.3202	1.9410	2.7340	0.4457	21.7709	17.3876	12.8761	1452.88
H-Fork [134]	18.0182	51.0756	26.6747	62.8166	2.3202	1.9525	2.3918	0.4240	21.6327	16.8653	4.7246	109.43
H-Seq [134]	20.7391	57.5378	28.5517	67.4649	2.2394	1.9892	2.4385	0.4249	21.4770	17.5616	4.4260	95.12
H-Regions [134]	15.4803	48.0767	21.8225	56.8994	2.6328	2.0732	2.8347	0.4044	20.9848	17.6858	6.0638	88.78
Pix2pix++ [55]	8.8687	34.5434	9.2713	35.7490	2.5625	2.0879	2.7961	0.3664	17.6549	18.4015	13.1153	220.23
X-Fork++ [133]	10.2658	37.8405	11.4138	38.7976	2.4280	2.0387	2.7630	0.3406	17.3937	18.2153	10.1403	166.33
X-Seq++ [133]	11.2580	36.8018	11.9838	36.9231	2.6849	2.1325	2.9397	0.3617	17.4893	18.4122	11.8560	154.80
SelectionGAN [165]	33.9055	71.8779	50.8878	85.0019	2.6576	2.1279	2.9267	**0.5752**	**24.7136**	**19.7302**	2.6183	26.09
LGGAN (Ours)	**37.0871**	**75.1314**	**56.0278**	**85.4714**	**2.8088**	**2.2804**	**3.1205**	0.5609	24.4779	19.6138	**2.2922**	**25.04**

Table 6.1: Quantitative comparison of cross-view image translation on SVA in the a2g direction. (∗) The inception scores for real (ground truth) data are 3.1282, 2.4932, and 3.4646 for the 'all', 'top-1', and 'top-5' setups, respectively.

Method	Accuracy (%) ↑				Inception Score* ↑			SSIM ↑	PSNR ↑	SD ↑	KL ↓
	Top-1		Top-5		All	Top-1	Top-5				
Pix2pix [55]	6.80	9.15	23.55	27.00	2.8515	1.9342	2.9083	0.4180	17.6291	19.2821	38.26 ± 1.88
X-SO [134]	27.56	41.15	57.96	73.20	2.9459	2.0963	2.9980	0.4772	19.6203	19.2939	7.20 ± 1.37
X-Fork [133]	30.00	48.68	61.57	78.84	3.0720	2.2402	3.0932	0.4963	19.8928	19.4533	6.00 ± 1.28
X-Seq [133]	30.16	49.85	62.59	80.70	2.7384	2.1304	2.7674	0.5031	20.2803	19.5258	5.93 ± 1.32
Pix2pix++ [55]	32.06	54.70	63.19	81.01	3.1709	2.1200	3.2001	0.4871	21.6675	18.8504	5.49 ± 1.25
X-Fork++ [133]	34.67	59.14	66.37	84.70	3.0737	2.1508	3.0893	0.4982	21.7260	18.9402	4.59 ± 1.16
X-Seq++ [133]	31.58	51.67	65.21	82.48	3.1703	2.2185	3.2444	0.4912	21.7659	18.9265	4.94 ± 1.18
SelectionGAN [165]	42.11	68.12	77.74	92.89	3.0613	2.2707	3.1336	**0.5938**	**23.8874**	**20.0174**	2.74 ± 0.86
LGGAN (Ours)	**48.17**	**79.35**	**81.14**	**94.91**	**3.3994**	**2.3478**	**3.4261**	0.5457	22.9949	19.6145	**2.18 ± 0.74**

Table 6.2: Quantitative comparison of cross-view image translation on Dayton in the a2g direction. (∗) The inception scores for real (ground truth) data are 3.8319, 2.5753, and 3.9222 for the 'all', 'top-1', and 'top-5' setups, respectively.

Method	Accuracy (%) ↑				Inception Score* ↑			SSIM ↑	PSNR ↑	SD ↑	KL ↓
	Top-1		Top-5		All	Top-1	Top-5				
Zhai et al. [211]	13.97	14.03	42.09	52.29	1.8434	1.5171	1.8666	0.4147	17.4886	16.6184	27.43 ± 1.63
Pix2pix [55]	7.33	9.25	25.81	32.67	3.2771	2.2219	3.4312	0.3923	17.6578	18.5239	59.81 ± 2.12
X-SO [134]	0.29	0.21	6.14	9.08	1.7575	1.4145	1.7791	0.3451	17.6201	16.9919	414.25 ± 2.37
X-Fork [133]	20.58	31.24	50.51	63.66	3.4432	2.5447	3.5567	0.4356	19.0509	18.6706	11.71 ± 1.55
X-Seq [133]	15.98	24.14	42.91	54.41	3.8151	2.6738	**4.0077**	0.4231	18.8067	18.4378	15.52 ± 1.73
Pix2pix++ [55]	26.45	41.87	57.26	72.87	3.2592	2.4175	3.5078	0.4617	21.5739	18.9044	9.47 ± 1.69
X-Fork++ [133]	31.03	49.65	64.47	81.16	3.3758	2.5375	3.5711	0.4769	21.6504	18.9856	7.18 ± 1.56
X-Seq++ [133]	34.69	54.61	67.12	83.46	3.3919	2.5474	3.4858	0.4740	21.6733	18.9907	5.19 ± 1.31
SelectionGAN [165]	41.52	65.51	74.32	89.66	3.8074	2.7181	3.9197	**0.5323**	**23.1466**	19.6100	2.96 ± 0.97
LGGAN (Ours)	**44.75**	**70.68**	**78.76**	**93.40**	**3.9180**	**2.8383**	3.9878	0.5238	22.5766	**19.7440**	**2.55 ± 0.95**

Table 6.3: Quantitative comparison of cross-view image translation on CVUSA in a2g direction. (∗) The inception scores for real (ground truth) data are 4.8741, 3.2959, and 4.9943 for the 'all', 'top-1', and 'top-5' setups, respectively.

(Figure 6.13). We observe that, given an aerial image and a few semantic maps, LGGAN is able to produce the same scene but with different

Figure 6.13: Examples of arbitrary cross-view image translation.

Model	Pix2pix [55]	X-Fork [133]	X-Seq [133]	SelectionGAN [165]	LGGAN
G	39.0820	39.2163	39.0820*2	55.4808	**12.1913**
D	2.7696	2.7696	2.7696*2	2.7687	**2.7678*2**
Total	41.8516	41.9859	83.7032	58.2495	**17.7269**

Table 6.4: Comparison of the number of network parameters (M). 'G' and 'D' stand for generator and discriminator, respectively.

Method	Cityscapes			ADE20K			COCO-Stuff		
	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓
CRN [16]	52.4	77.1	104.7	22.4	68.8	73.3	23.7	40.4	70.4
SIMS [127]	47.2	75.5	49.7	-	-	-	-	-	-
Pix2pixHD [184]	58.3	81.4	95.0	20.3	69.2	81.8	14.6	45.7	111.5
SelectionGAN [168]	63.8	82.4	65.2	40.1	81.2	33.1	-	-	-
PIS [35]	64.8	82.4	96.4	-	-	-	38.6	69.0	28.8
TSIT [58]	65.9	82.7	59.2	38.6	80.8	31.6	-	-	-
DAGAN [154]	66.1	82.6	60.3	40.5	81.6	31.9	-	-	-
GauGAN [119]	62.3	81.9	71.8	38.5	79.9	33.9	37.4	67.9	22.6
+ SAU (Ours)	65.5	82.5	48.3	39.8	80.7	32.0	**39.0**	**69.1**	**20.1**
LGGAN (Ours)	**68.4**	**83.0**	57.7	**41.6**	**81.8**	31.6	-	-	-
+ SAU (Ours)	67.7	82.9	**48.1**	41.4	81.5	**30.5**	-	-	-

Table 6.5: Quantitative comparison of semantic image synthesis on Cityscapes, ADE20K, and COCO-Stuff.

viewpoints.

Network Parameter Comparisons. In Table 6.4, we compare the number of network parameters in LGGAN with several state-of-the-art models. As we can see, the proposed LGGAN achieves superior model capacity and better generation performance compared with existing methods.

AMT ↑	Cityscapes	ADE20K
LGGAN (Ours) vs. CRN [16]	67.38	79.54
LGGAN (Ours) vs. Pix2pixHD [184]	56.16	85.69
LGGAN (Ours) vs. SIMS [127]	54.84	-
LGGAN (Ours) vs. GauGAN [119]	53.19	57.31

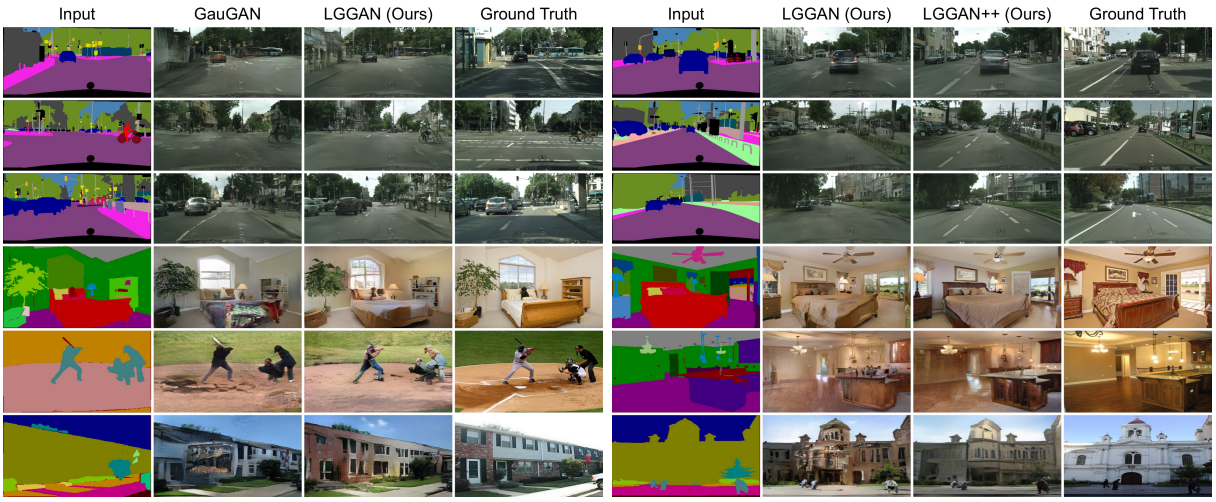
Table 6.6: User study. The numbers indicate the percentage of users who favor the results of the proposed LGGAN over the competing methods.

Semantic Image Synthesis

Datasets and Evaluation Metrics. We follow GauGAN [119] and conduct extensive experiments on Cityscapes [26], ADE20K [226], and COCO-Stuff [11]. We use mean intersection-over-union (mIoU), pixel accuracy (Acc), and Fréchet inception distance (FID) [48] as the evaluation metrics.

State-of-the-Art Comparisons. We compare the proposed LGGAN with several leading semantic image synthesis methods, i.e., Pix2pixHD [184], CRN [16], SIMS [127], GauGAN [119], SelectionGAN [165], TSIT [58], PIS [35], and DAGAN [154]. The results in terms of mIoU, Acc, and FID are shown in Table 6.5. The proposed LGGAN outperforms the existing competing methods by a large margin in both mIoU and Acc. For FID, LGGAN is only worse than SIMS on Cityscapes. However, SIMS has poor segmentation performance. This is because SIMS produces an image by searching and copying image patches from the training dataset. The generated images are more realistic since the method uses real image patches. However, SIMS tends to copy objects with mismatched patches due to the presence queries that cannot be guaranteed to have results. Finally, we follow the evaluation protocol of GauGAN and provide Amazon Mechanical Turk (AMT) results, as shown in Table 6.6. As can be seen, users favor our synthesized results on both datasets compared with other competing methods including SIMS.

Qualitative Evaluation. The qualitative results compared with the lead-



(a) GauGAN vs. LGGAN. (b) LGGAN vs. LGGAN++ (i.e., LGGAN+SAU).

Figure 6.14: Qualitative comparison of semantic image synthesis on Cityscapes (top three rows) and ADE20K (bottom three rows).

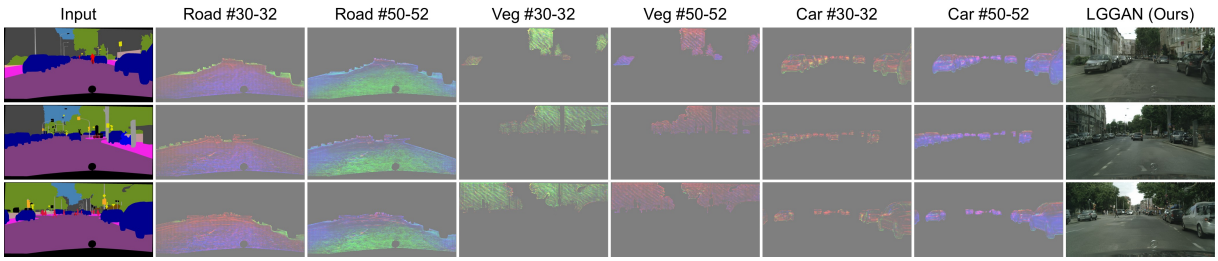


Figure 6.15: Visualization of class-specific feature maps learned for three different classes, i.e., road, vegetation, and cars.

ing method GauGAN [119] are shown in Figure 6.14(a). We can see that our LGGAN generates much better results with fewer visual artifacts than GauGAN.

Visualization of Learned Feature Maps. In Figure 6.15, we randomly show some channels from the learned class-specific feature maps (30^{th} to 32^{th}, and 50^{th} to 52^{th}) on Cityscapes to see if they clearly highlight particular semantic classes. We show the visualization results for three different classes, i.e., road, vegetation, and cars. We can easily observe that each local subgenerator effectively learns the deep class-level representations, further confirming our motivations.

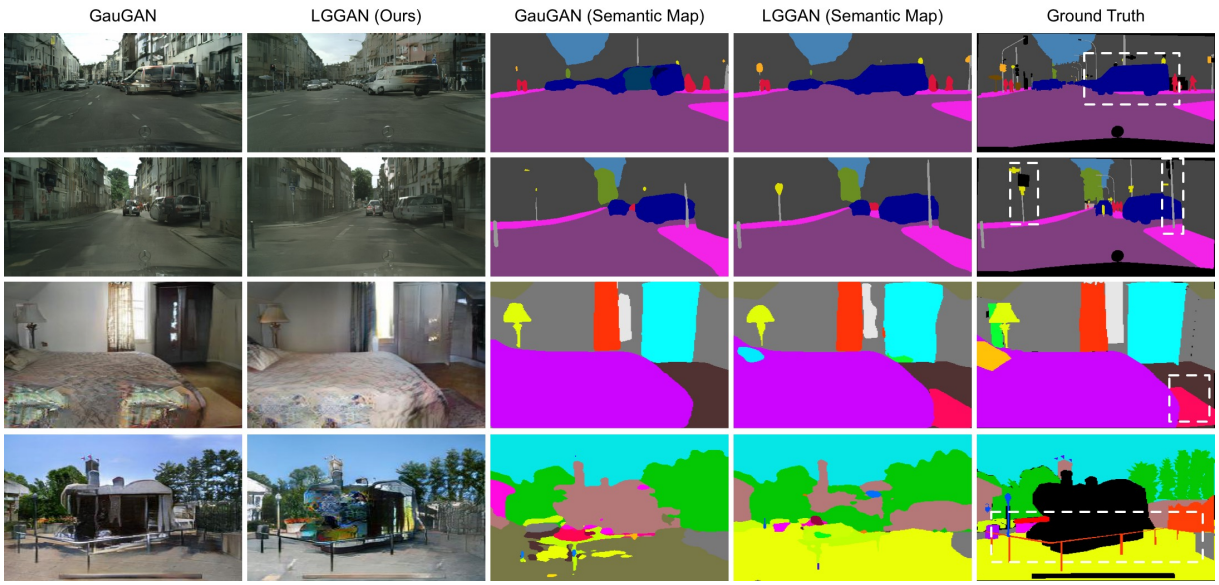


Figure 6.16: Visualization of semantic maps generated by LGGAN compared with those from GauGAN [119] on Cityscapes (top two rows) and ADE20K (bottom two rows).

Visualization of Generated Semantic Maps. We follow GauGAN [119] and apply pretrained segmentation networks on the generated images to produce semantic maps, i.e., DRN-D-105 [207] for Cityscapes and UperNet101 [195] for ADE20K. The generated semantic maps of our LGGAN, GauGAN, and ground truths are shown in Figure 6.16. As can be seen, LGGAN generates better semantic maps than GauGAN, especially on local textures ('car' in the first row) and small objects ('traffic sign' and 'pole' in the second row).

Ablation Study

Baselines. We conduct extensive ablation studies on Cityscapes to evaluate different components of our LGGAN. The proposed LGGAN has five baselines (i.e., B1, B2, B3, B4, B5), as shown in Table 6.7: (i) In B1, only the global generator is adopted. (ii) B2 combines the global generator and the proposed local generator to produce the final results, where the local results are produced using an addition operation, as proposed in

No.	Setup of LGGAN	mIoU ↑	FID ↓
B1	Ours w/ Global	62.3	71.8
B2	B1 + Local (addition)	64.6	66.1
B3	B1 + Local (convolution)	65.8	65.6
B4	B3 + Class Discriminative Loss	67.0	61.3
B5	B4 + Weight Map	**68.4**	**57.7**

Table 6.7: Ablation study of the proposed LGGAN on Cityscapes.

Equation (6.4). (iii) B3 is similar to B2 but uses a convolutional layer to generate the local results, as presented in Equation (6.5). (iv) B4 employs the proposed classification-based discriminative feature learning module. (v) B5 is our full LGGAN model and adopts the proposed weight map fusion strategy.

Effect of Local and Global Generation. The results of the ablation study are shown in Table 6.7. When using an addition operation to generate the local result, the local and global generation strategy improves the mIoU and FID by 2.3 and 5.7, respectively. When adopting a convolutional operation to produce the local results, the performance is further improved, with a 3.5 and 6.2 gain in mIoU and FID, respectively. Both results confirm the effectiveness of the proposed local and global generation framework. We also provide qualitative results in Figure 6.1. We observe that our full model, i.e., Global+Local, generates visually better results than both the individual global and local methods.

Effect of Classification-Based Feature Learning. B4 significantly outperforms B3, with gains of roughly 1.2 and 4.3 in mIoU and FID, respectively. This means that the model does indeed learn a more discriminative class-specific feature representation, confirming the superiority of our design.

Effect of Weight Map Fusion. By adding the proposed weight map fusion scheme in B5, the overall performance is further boosted, with improvements of 1.4 and 3.6 in mIoU and FID, respectively. This indicates

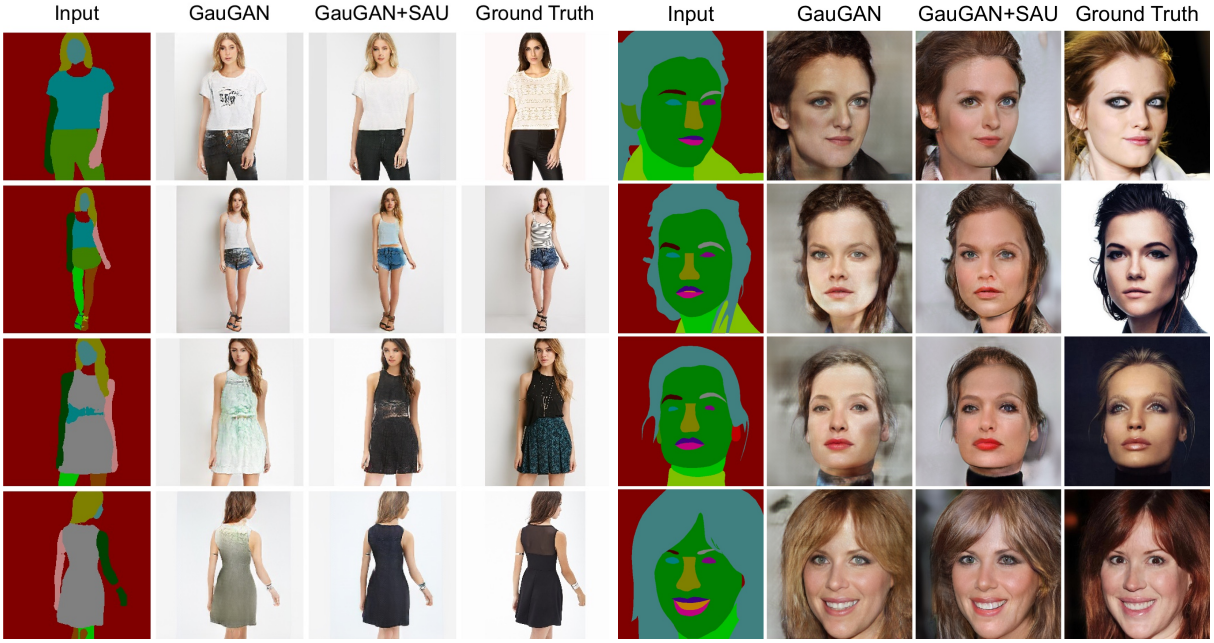
(a) Cityscapes. (b) Facades.

Figure 6.17: Qualitative comparison of semantic image sythesis on Cityscapes and Facades.

that the proposed LGGAN can in fact learn complementary information from the local and global generation branches. In Figure 6.1, we show some examples of the generated weight maps. As can be seen, the generated global weight maps mainly focus on learning the global layout and structure, while the learned local weight maps focus on the local details, especially the connection between different classes.

6.5.2 Semantic-Aware Upsampling

Datasets. We first follow GauGAN [119] and conduct experiments on Cityscapes [26], ADE20K [226], and COCO-Stuff [11]. Then we conduct experiments on three more datasets with diverse scenarios. 1) Facades [176] contains different city images with various architectural styles. The training and test sets have 378 and 228 images, respectively. We resize the images to 512×512 for high-resolution layout-to-image translation tasks. 2) CelebAMask-HQ [75] contains face images with 19 semantic facial attributes. The training and test sets are made up of 24,183 and 2,842 images, respectively. We also resize these images to 512×512. 3) DeepFashion [97] contains human body images. The number of images in the



(a) DeepFashion. (b) CelebAMask-HQ.

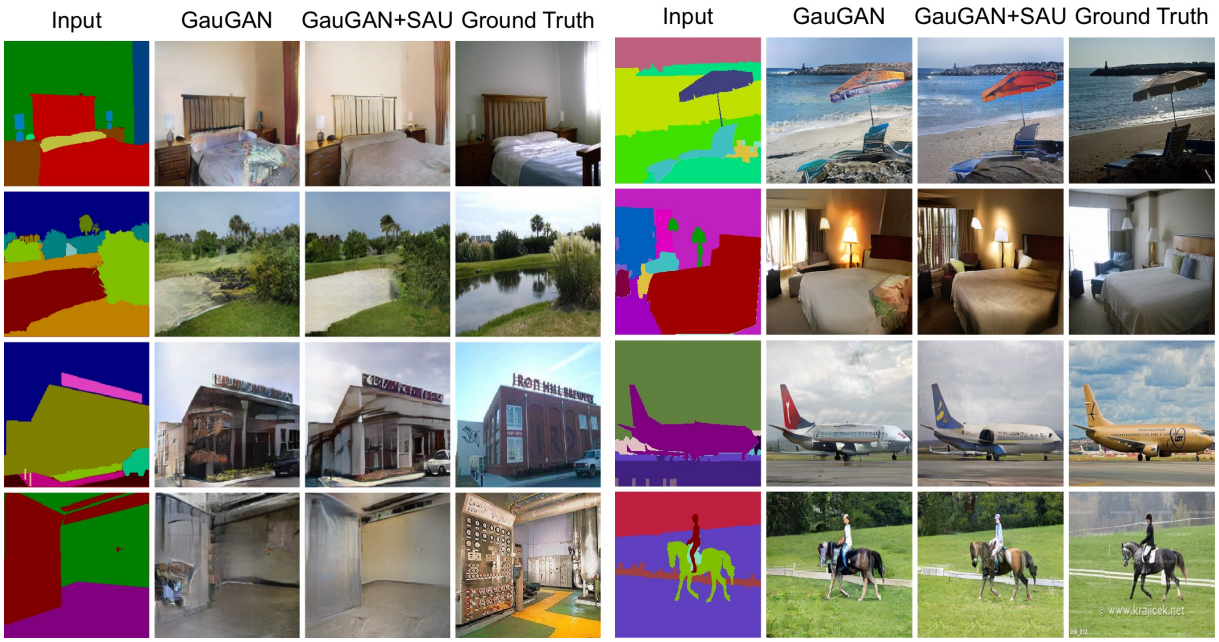
Figure 6.18: Qualitative comparison of semantic image synthesis on DeepFashion and CelebAMask-HQ.

training and test sets are 30,000 and 2,247, respectively. We resize the images to 256×256, and use a well-trained model [83] to extract input semantic layouts for this dataset.

Evaluation Metrics. We follow GauGAN [119] and use mIoU, Acc, and FID [48] as the evaluation metrics on Cityscapes, ADE20K, and COCO-Stuff. For DeepFashion, CelebAMask-HQ, and Facades, we use FID and Learned Perceptual Image Patch Similarity (LPIPS) [220].

State-of-the-Art Comparisons

Qualitative Comparisons. We first compare the proposed GauGAN+SAU with GauGAN [119] on DeepFashion, CelebAMask-HQ, and Facades. Specifically, we replace the feature upsampling layer in GauGAN with our SAU layer. Visualization results are shown in Figures 6.17(b) and 6.18. We can see that the model with SAU generates more photorealistic results than



(a) ADE20K. (b) COCO-Stuff.

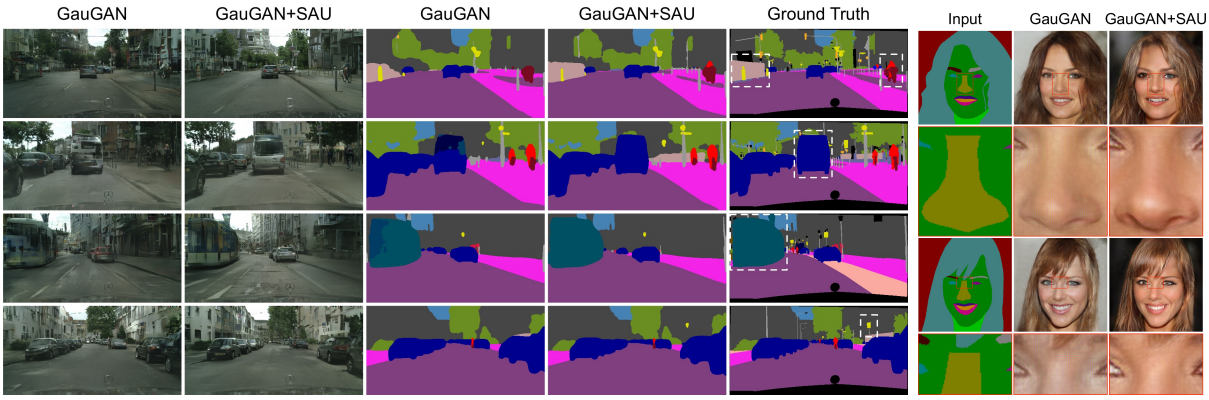
Figure 6.19: Qualitative comparison of semantic image synthesis on ADE20K and COCO-Stuff.

AMT ↑	Cityscapes	ADE20K	COCO-Stuff	DeepFashion	Facades	CelebAMask-HQ
GauGAN+SAU (Ours) vs. GauGAN	63.8	65.7	62.4	60.1	58.3	70.5

Table 6.8: User study. The numbers indicate the percentage of users who favor the results of the proposed GauGAN+SAU over GauGAN.

the original GauGAN. Moreover, we compare GauGAN and the proposed method in a zoomed-in manner on CelebAMask-HQ in Figure 6.20(b). As can be seen, the model with our SAU generates more vivid content than the original GauGAN model, further validating the effectiveness of SAU. Lastly, we compare the proposed method with GauGAN on Cityscapes, ADE20K, and COCO-Stuff. Comparison results are shown in Figures 6.17(a) and 6.19. Our method produces clearer and more visually plausible results than both leading methods, further demonstrating the benefit of our design.

User Study. We follow the same evaluation protocol as GauGAN and perform a user study. The results compared with the original GauGAN



(a) Cityscapes. (b) CelebAMask-HQ.

Figure 6.20: (a) Visualization of semantic maps generated by GauGAN+SAU compared with those from GauGAN [119] on Cityscapes. The most improved regions are highlighted in the ground truths with white dash boxes. (b) Comparison in a zoomed-in manner on CelebAMask-HQ.

Method	DeepFashion		Facades		CelebAMask-HQ	
	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
GauGAN [119]	22.8	0.2476	116.8	0.5437	42.2	0.4870
+ SAU (Ours)	**20.8**	**0.2446**	**112.4**	**0.5387**	**33.6**	**0.4788**

Table 6.9: Quantitative comparison of semantic image synthesis on DeepFashion, Facades, and CelebAMask-HQ.

are shown in Table 6.8. As can be seen, users strongly favor the results generated by our proposed method on all datasets, further validating that the images generated by our upsampling method are more photorealistic.

Quantitative Comparisons. Although the user study is most suitable for evaluating the quality of the generated images, we also follow GauGAN and use mIoU, Acc, FID, and LPIPS for quantitative evaluation. The results compared with several leading methods are shown in Tables 6.5 and 6.9. Firstly, we observe from Table 6.9 that the model with SAU achieves better results than GauGAN on DeepFashion, CelebAMask-HQ, and Facades. Moreover, from Table 6.5, we can see that our method (i.e., GauGAN+SAU) achieves competitive results compared with other leading methods on Cityscapes, ADE20K, and COCO-Stuff.

Visualization of Generated Semantic Maps. We again follow GauGAN and apply the pretrained DRN-D-105 [207] to the generated Cityscapes images to produce semantic maps. The results, compared with those produced by GauGAN, are shown in Figure 6.20(a). We see that the method with our proposed SAU generates more semantically consistent results than the original GauGAN.

LGGAN vs. LGGAN++. Our SAU is general and can be seamlessly integrated into existing GANs. To demonstrate this generalization ability, we conduct more experiments on both Cityscapes and ADE20K. Specifically, we replace the upsampling layers in LGGAN [169] with the proposed SAU. The results are shown in Table 6.5 and Figure 6.14(b). We can see from Figure 6.14(b) that LGGAN++ achieves more photorealistic results than LGGAN, validating the generalization ability of SAU. Moreover, LGGAN+SAU (i.e., LGGAN++) achieves a significantly better FID score on both datasets, as shown in Table 6.5.

Ablation Study

Baselines. We conduct an extensive ablation study on Cityscapes to evaluate the effectiveness of the proposed SAU. As shown in Table 6.10, B1, B2, and B3 are three traditional upsampling methods. B4 and B5 are two learnable upsampling methods. B6 is the spatial attention module proposed in [38]. B7 is our proposed SAU.

Ablation Analysis. We first compare the proposed SAU with different upsampling strategies (i.e., B1, B2, B3, B4, B5). The results of the ablation study are shown in Table 6.10 and Figure 6.21. We can see from Table 6.10 that the proposed SAU achieves a significantly better FID than other methods, indicating that the design of effective upsampling methods is critical for this challenging task. We also observe from Figure 6.21 that the proposed SAU generates more photorealistic and semantically consistent

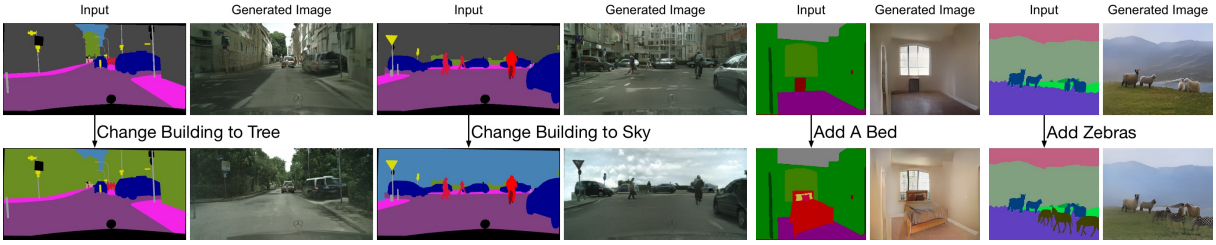
Figure 6.21: Qualitative comparison of different upsampling methods on Cityscapes. Key differences are highlighted by red boxes.

No.	Method	FID ↓	Params ↓
B1	Nearest-Neighbor Upsampling	58.7	93.0M
B2	Bilinear Upsampling	52.9	93.0M
B3	Bicubic Upsampling	54.4	93.0M
B4	Deconvolution [111]	54.0	98.6M
B5	Pixel Shuffle [141]	59.1	143.2M
B6	Spatial Attention [38]	56.2	97.4M
B7	SAU (Ours)	**48.3**	**93.4M**

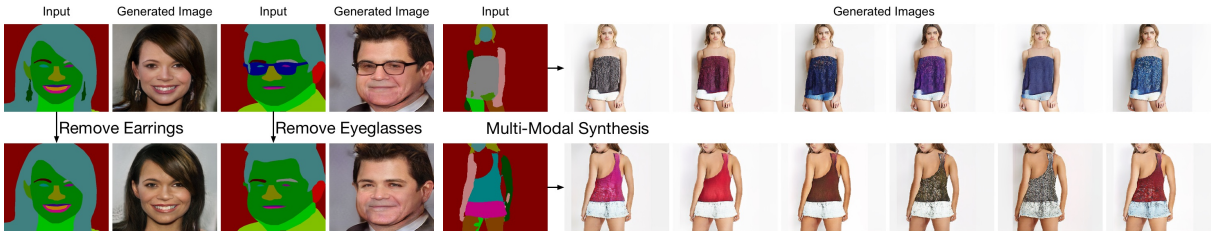
Table 6.10: Quantitative comparison of different feature upsampling and enhancement methods on Cityscapes.

results with fewer artifacts than other upsampling methods. Moreover, we add the spatial attention module [38] to GauGAN, obtaining 56.2 in FID. We can see that our method still significantly outperforms spatial attention.

Model Parameter Comparisons. We also compare the number of generator parameters with different baselines. The results are shown in Table 6.10. Traditional upsampling methods (B1, B2, and B3) have the same number of parameters. Moreover, we can see that the proposed SAU achieves superior model capacity compared to the learnable upsampling



(a) Cityscapes. (b) ADE20K. (c) COCO-Stuff.



(d) CelebAMask-HQ. (e) DeepFashion.

Figure 6.22: Exemplar applications of the proposed method on different datasets.

methods (i.e., B4 and B5) and spatial attention (i.e., B6).

Applications

Semantic Manipulation. Our model also supports semantic manipulation. For instance, we can replace a building with trees (Figure 6.22(a)), insert a bed into a room (Figure 6.22(b)), add a few zebras to some grass (Figure 6.22(c)), or remove earrings and eyeglasses from a face (Figure 6.22(d)). These applications provide users more controllability during the translation process.

Multi-Modal Synthesis. By using a random vector as the input of the generator, our model can perform multi-modal synthesis. The results are shown in Figure 6.22(e). We can see that our model generates different outputs from the same input layout.

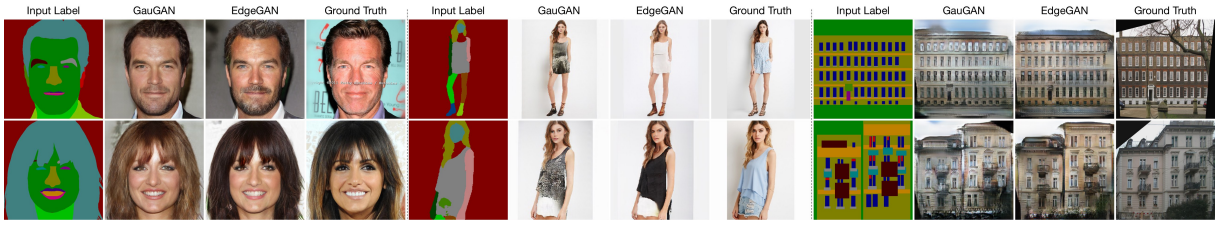


Figure 6.23: Results of GauGAN [119] and EdgeGAN on CelebAMask-HQ (left), DeepFashion (middle), and Facades (right).

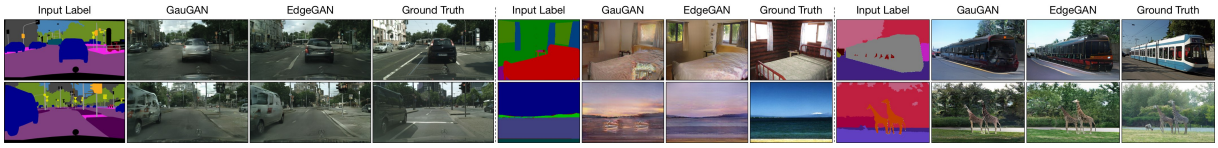


Figure 6.24: Results of GauGAN [119] and EdgeGAN on Cityscapes (left), ADE20K (middle), and COCO-Stuff (right).

6.6 EdgeGAN Experiments

Datasets and Evaluation Metrics. We first follow GauGAN [119] and conduct experiments on three datasets, i.e., Cityscapes [26], ADE20K [226], and COCO-Stuff [11]. We then conduct experiments on three more datasets with diverse scenarios to evaluate the effectiveness of the proposed EdgeGAN. (1) Facades [176] contains 378 and 228 training and test images, respectively. We resize the images to 512×512 for high-resolution image generation; (2) CelebAMask-HQ [75] contains 24,183 and 2,842 training and test images, respectively. We also resize the images to 512×512; (3) DeepFashion [97] contains 30,000 and 2,247 training and test samples, respectively. We resize the images to 256×256, and use a well-trained model [83] to obtain input semantic layouts. Moreover, we follow [119] and adopt the mean Intersection-over-Union (mIoU), Pixel Accuracy (Acc), and Fréchet Inception Distance (FID) [48] as the evaluation metrics.

AMT ↑	DeepFashion	Facades	CelebAMask-HQ
Ours vs. GauGAN [119]	69.13	73.41	78.56
AMT ↑	Cityscapes	ADE20K	COCO-Stuff
Ours vs. CRN [16]	70.28	81.35	84.71
Ours vs. Pix2pixHD [184]	60.85	85.18	89.36
Ours vs. SIMS [127]	57.67	-	-
Ours vs. GauGAN [119]	56.54	60.49	65.96
Ours vs. CC-FPSE [96]	55.81	57.75	61.52

Table 6.11: User preference study on six datasets. The numbers indicate the percentage of users who favor the results of the proposed EdgeGAN over the competing method.

6.6.1 State-of-the-Art Comparisons

Qualitative Comparisons. Visual comparison results of DeepFashion, Facades, and CelebAMask-HQ with the leading method GauGAN [119] are shown in Figure 6.23. Note that we used the source code provided by the authors to generate the results of GauGAN on these three datasets for fair comparisons. We can see that the proposed EdgeGAN achieves visually better results with fewer visual artifacts than GauGAN. To further validate the effectiveness of the proposed EdgeGAN, we compare it with GauGAN [119] on the Cityscapes, ADE20K, and COCO-Stuff datasets. The comparison results are shown in Figure 6.24. We observe that EdgeGAN generates sharper images than GauGAN, especially at local structures and details.

User Study. We follow the same evaluation protocol of GauGAN [119] and conduct a user study. Specifically, we give the participants an input semantic label and two translated images from different models and ask them to choose the generated image that looks more like a corresponding image of the semantic label. The participants are given unlimited time to make the decision. The results of DeepFashion, Facades, and CelebAMask-HQ compared with GauGAN [119] are shown in Table 6.11 (top). Moreover, we provide the results compared with Pix2pixHD [184], CRN [16], SIMS [127], GauGAN [119], and CC-FPSE [96] on the other three datasets

FID ↓	DeepFashion	Facades	CelebAMask-HQ
GauGAN [119]	22.8	116.8	42.2
EdgeGAN	**21.1**	**99.9**	**28.1**

Table 6.12: Quantitative comparison with GauGAN [119] on DeepFashion, Facades, and CelebAMask-HQ.

Method	Cityscapes			ADE20K			COCO-Stuff		
	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓
CRN [16]	52.4	77.1	104.7	22.4	68.8	73.3	23.7	40.4	70.4
SIMS [127]	47.2	75.5	49.7	-	-	-	-	-	-
Pix2pixHD [184]	58.3	81.4	95.0	20.3	69.2	81.8	14.6	45.8	111.5
PIS [35]	64.8	82.4	96.4	-	-	-	38.6	69.0	28.8
TSIT [58]	65.9	82.7	59.2	38.6	80.8	31.6	-	-	-
LGGAN [169]	68.4	83.0	57.7	41.6	81.8	31.6	-	-	-
CC-FPSE [96]	65.5	82.3	54.3	43.7	82.9	31.7	41.6	70.7	19.2
GauGAN [119]	62.3	81.9	71.8	38.5	79.9	33.9	37.4	67.9	22.6
EdgeGAN	**64.5**	**82.5**	**57.1**	**42.0**	**82.0**	**32.4**	**38.3**	**68.7**	**21.0**

Table 6.13: Quantitative comparison of different methods on Cityscapes, ADE20K, and COCO-Stuff.

(i.e., Cityscapes, ADE20K, and COCO-Stuff) in Table 6.11 (bottom). We observe that users favor our synthesized results on all the six datasets compared with other competing methods including GauGAN and CC-FPSE, further validating that the generated images by the proposed EdgeGAN are more natural and photo-realistic.

Quantitative Comparisons. Although the user study is more suitable for evaluating the quality of the generated images, we also follow previous works and use mIoU, Acc, and FID for quantitative evaluation. Results of the six datasets are shown in Tables 6.12 and 6.13. It is clear that with GauGAN as a baseline, the proposed EdgeGAN outperforms GauGAN by a large margin on all the six datasets, validating the effectiveness of the proposed method. However, we also observe that other methods such as CC-FPSE [96] achieves better results than ours, but our proposed modules are lightweight and general, and can be seamlessly integrated into CC-

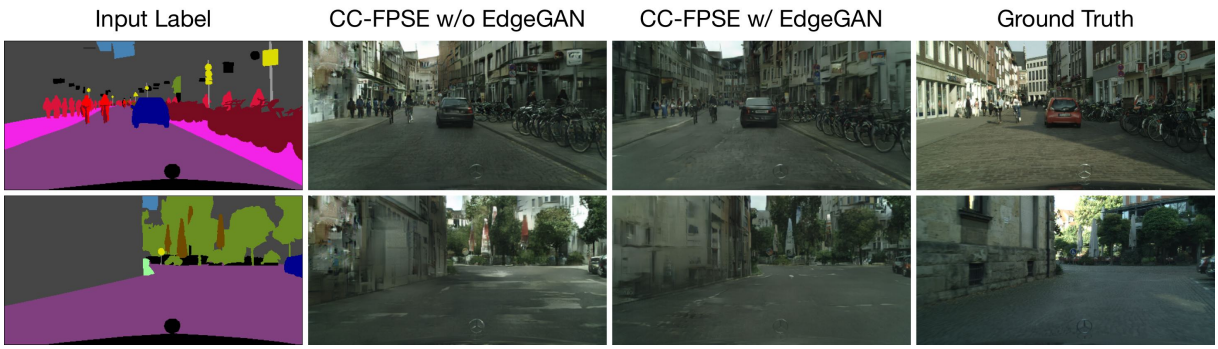


Figure 6.25: The generalization ability of EdgeGAN. We can see that CC-FPSE [96] with our model (i.e., CC-FPSE w/ EdgeGAN) generates more realistic images with fewer artifacts than CC-FPSE without using our model (i.e., CC-FPSE w/o EdgeGAN).

Method	Cityscapes			ADE20K			COCO-Stuff		
	G	*D*	Total ↓	*G*	*D*	Total ↓	*G*	*D*	Total ↓
CC-FPSE [96]	138.6M	5.2M	143.8M	151.2M	5.2M	156.4M	152.4M	5.2M	157.6M
GauGAN [119]	93.0M	5.6M	**98.6M**	96.5M	5.8M	**102.3M**	97.5M	5.9M	**103.4M**
EdgeGAN	93.2M	5.6M	98.8M	97.2M	5.8M	103.0M	98.4M	5.9M	104.3M

Table 6.14: Comparison of the number of model parameters. *G* and *D* denote Generator and Discriminator, respectively.

FPSE to further improve performance with negligible overheads. Moreover, we provide the number of model parameters in Table 6.14. We see that EdgeGAN has much fewer model parameters compared to CC-FPSE on all datasets, meaning that the proposed EdgeGAN can be trained with less training time and GPU memory.

Generalization of EdgeGAN. Our proposed framework and modules are general and can be seamlessly integrated into existing GANs. Thus, to validate the generalization ability of EdgeGAN, we further conduct more experiments on Cityscapes. Specifically, we adopt CC-FPSE [96] as our encoder *E* and keep everything unchanged. We observe that CC-FPSE with our method achieves significantly better results than the original one without using our method. Specifically, we further improve the mIoU, Acc, and FID from 65.5, 82.3, and 54.3 to 67.6, 82.9, and 50.6, respectively, validating the generalization ability of the proposed method. We can also

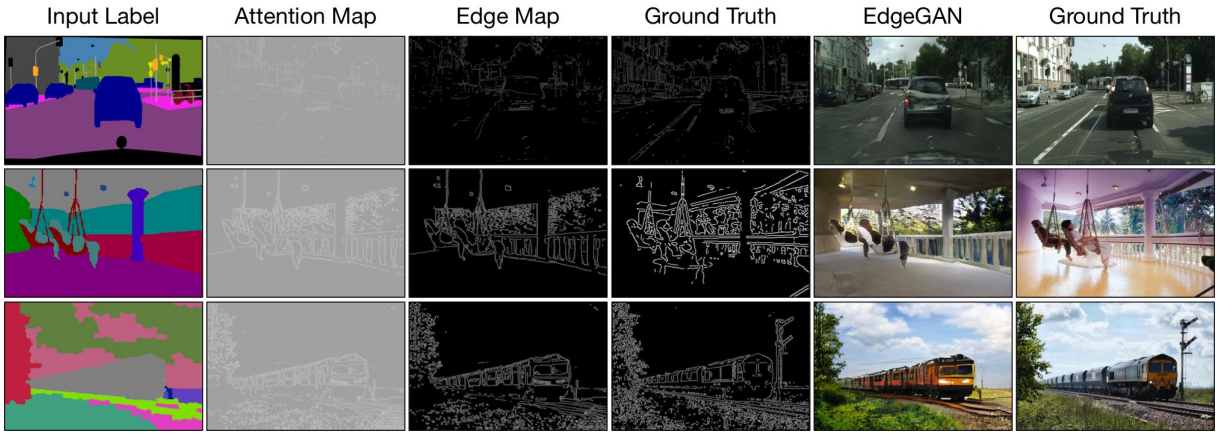


Figure 6.26: Edge and attention maps generated by EdgeGAN on Cityscapes (top), ADE20K (middle), and COCO-Stuff (bottom).

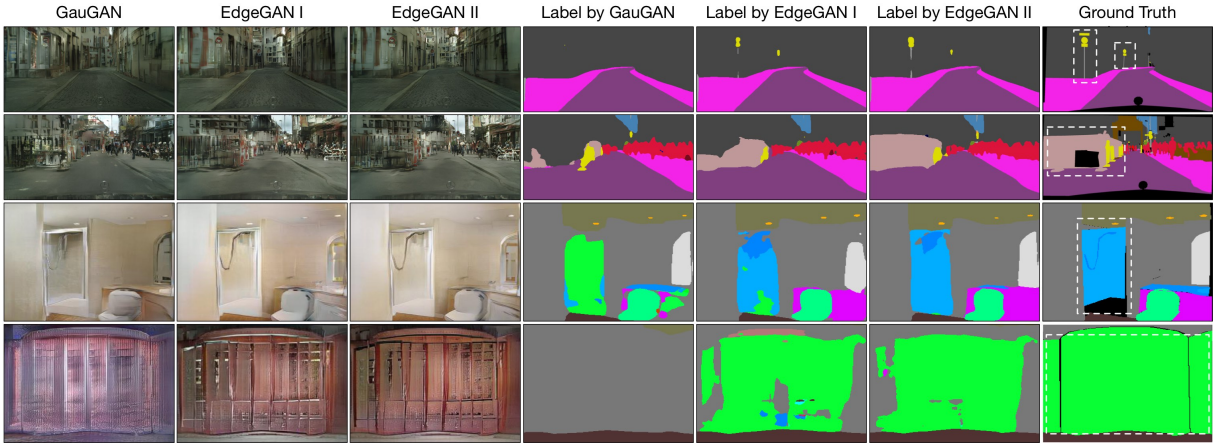


Figure 6.27: Segmentation maps generated by GauGAN [119] and EdgeGAN on Cityscapes (top two rows) and ADE20K (bottom two rows). 'EdgeGAN I' and 'EdgeGAN II' stand for $I^{'}$ and $I^{''}$, respectively.

see that the model with our EdgeGAN generates significantly better results than the model without using our EdgeGAN in Figure 6.25.

Visualization of Edge and Attention Maps. We also visualize the generated edge and attention maps in Figure 6.26. We observe that the proposed EdgeGAN can generate reasonable edge maps according to the input labels, thus the generated edge maps can be used to provide more local structure information for generating more photo-realistic images.

Visualization of Segmentation Maps. We follow GauGAN [119] and apply pre-trained segmentation networks [207, 195] on the generated im-

Figure 6.28: Results generated by EdgeGAN on the multi-modal image and edge synthesis task (left). Visualization of the differences after the edge-guided refinement in Equation (6.13) (right).

Variants of EdgeGAN	mIoU ↑	Acc ↑	FID ↓
E+G_i	58.6	81.4	65.7
E+G_i+G_e	60.2	81.7	61.0
E+G_i+G_e+G_t	61.5	82.0	59.0
E+G_i+G_e+G_t+G_s	**64.5**	**82.5**	**57.1**

Table 6.15: Quantitative comparison of different variants of the proposed EdgeGAN on Cityscapes.

ages to produce segmentation maps. Results compared with GauGAN [119] are shown in Figure 6.27. We observe that the proposed EdgeGAN consistently generates better semantic labels than GauGAN on both Cityscapes and ADE20K datasets.

Multi-Modal Image and Edge Synthesis. By using a random vector as the input of the generator, our model can perform multi-modal image and edge synthesis. Note that existing semantic image synthesis methods [119, 96, 169] can only achieve multi-modal image synthesis. The results are shown in Figure 6.28 (left). We can see that our model generates different edges and images from the same input layout, which we believe will benefit other tasks such as image inpainting and super-resolution.

6.6.2 Ablation Study

Variants of EdgeGAN. We conduct ablation studies on Cityscapes [26] to evaluate different components of EdgeGAN. Our EdgeGAN has four baselines as shown in Table 6.15: (i) 'E+G_i' means only using the encoder

	Cityscapes			ADE20K			COCO-Stuff		
	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓	mIoU ↑	Acc ↑	FID ↓
EdgeGAN I	61.7	82.1	59.1	39.6	80.9	34.2	36.8	67.3	23.1
EdgeGAN II	**64.5**	**82.5**	**57.1**	**42.0**	**82.0**	**32.4**	**38.3**	**68.7**	**21.0**

Table 6.16: Comparison of with ('EdgeGAN I') and without ('EdgeGAN II') using our semantic preserving module.

E and the proposed image generator G_i to synthesize the targeted images; (ii) 'E+G_i+G_e' means adopting the proposed image generator G_i and edge generator G_e to simultaneously produce both edge maps and images; (iii) 'E+G_i+G_e+G_t' connects the image generator G_i and the edge generator G_e by using the proposed attention guided edge transfer module G_t; (iv) 'E+G_i+G_e+G_t+G_s' is our full model and employs the proposed semantic preserving module G_s to further improve the quality of the final results.

Effect of Edge Guided Generation Strategy. The results of the ablation study are shown in Table 6.15. When using the proposed edge generator G_e to produce the corresponding edge map from the input label, performance on all evaluation metrics is improved. Specifically, 1.6, 0.3, and 4.7 point gains on the mIoU, Acc, and FID metrics, respectively, which confirms the effectiveness of the proposed edge guided generation strategy. We also provide several visualization results of the differences (see Equation (6.13)) after the edge-guided refinement in Figure 6.28 (right).

Effect of Attention Guided Edge Transfer Module. We observe that the implicitly learned edge structure information by the 'E+G_i+G_e' baseline is not enough for such a challenging task. Thus we further adopt the proposed attention guided edge transfer module G_t to transfer useful edge structure information from the edge generation branch to the image generation branch. We observe that 1.3, 0.3, and 2.0 point gains are obtained on the mIoU, Acc, and FID metrics, respectively. This means that the proposed transfer module G_t indeed learns rich feature representations with

more convincing structure cues and details, and then transfers them from the edge generator G_e to the image generator G_i, confirming our design motivation.

Effect of Semantic Preserving Module. By adding the proposed semantic preserving module G_s, the overall performance is further boosted with 3.0, 0.5, and 1.9 point improvements on the mIoU, Acc, and FID metrics, respectively. This means G_s indeed learns and highlights class-specific semantic feature maps, leading to better generation results. In Figure 6.27, we show some samples of the generated semantic maps. We observe that the semantic maps produced by the results with G_s (i.e., 'Label by EdgeGAN II' in Figure 6.27) are more accurate than those without using G_s ('Label by EdgeGAN I' in Figure 6.27). Moreover, we provide quantitative results in Table 6.16. We can see that G_s indeed learns better class-specific feature representation, leading to better performance on all datasets. Both quantitative and qualitative results confirm the effectiveness of our G_s.

6.7 Conclusion

We propose a local class-specific and global image-level generative adversarial network (LGGAN) for semantic-guided image generation. The proposed LGGAN contains three generation branches, i.e., global image-level generation, local class-level generation, and pixel-level fusion weight map generation, respectively. A new class-specific local generation network is designed to alleviate the influence of imbalanced training data and the size difference of objects for joint learning. To learn more discriminative and class-specific feature representations, a novel classification module is further proposed. Moreover, we introduce a novel semantic-aware upsampling method, which is able to aggregate semantic information in the input layout and adaptively conduct class-specific upsampling during the translation

process. Experimental results demonstrate the superiority of the proposed approach, which achieves the new state-of-the-art on both cross-view image translation and semantic image synthesis, on nine public datasets.

We also propose a novel EdgeGAN for challenging semantic image synthesis tasks. EdgeGAN introduces three core components: edge guided image generation strategy, attention guided edge transfer module, and semantic preserving module. The first one is employed to generate edge maps from input semantic labels. The second one is used to selectively transfer the useful structure information from the edge branch to the image branch. The third one is adopted to alleviate the problem of spatial resolution losses caused by different operations in the deep nets. Extensive experiments on six datasets show that EdgeGAN achieves significantly better results than existing methods. Lastly, we believe that our modules can be easily plugged into existing GANs to address other generation or computer vision tasks.

In the next chapter, we will introduce a novel DanceGAN for the challenging music-guided dance video synthesis task, which consists of two generation stages, i.e., music-to-skeleton translation, and skeleton-to-video translation.

Part III

Cross-Modal Translation

Chapter 7

DanceGAN

We propose a novel Dance Generative Adversarial Network (DanceGAN) for the music-guided dance video synthesis task, i.e., translating the input music to a dance video. The proposed DanceGAN consists of two generation stages, which are music-to-skeleton translation and skeleton-to-dance translation. We propose two novel graph attention networks to effectively construct skeleton sequences from the input music, which capture dependencies between joints in both spatial and temporal dimensions and have not been considered by any existing GAN-based generation work. Moreover, we propose a novel self-supervised regularization network to translate the generated skeletons along with a conditional image to a dance video, which considers the video generation from both forward and backward directions such that enhancing the generation performance and training stability. Lastly, we collect a new skeleton-to-dance translation dataset from the internet, which contains 54,944 video sequences. Extensive experiments demonstrate that DanceGAN achieves significantly realistic results on both music-to-skeleton and skeleton-to-dance translation tasks. The source code and trained models are available at https://github.com/HaOTang/DanceGAN.

7.1 Introduction

In this paper, we mainly focus on the music-guided dance video synthesis task, i.e., translating the input music to photo-realistic dance video as depicted in Figure 7.2. However, music/audio data tends to be less structured and thus involves more challenges to model its correlation with visual/video data, which makes generating a realistic dance video from the input music an open question. Therefore, we decompose the synthesis task into two generation stages: (1) music-to-skeleton translation, i.e., generating skeleton sequences from the input music for predicting the dance motion, and (2) skeleton-to-dance translation, i.e., generating dance video conditioned on the generated skeleton sequences in stage I along with a conditional image. In this way, we simultaneously model the motion information from the generated skeleton sequences and the appearance content (e.g., person identity) from the conditional image, leading to the final realistic dance video.

Existing music-to-skeleton translation methods such as [76, 65, 45, 62, 143, 39, 205, 3, 142, 1, 170] mainly rely on classical convolutional and recurrent neural networks. For instance, Tang et al. [170] use an LSTM-autoencoder model to generate dance pose skeletons, whilst Lee et al. [76] introduce a synthesis-by-analysis learning framework to generate dance from music. However, these networks suffer from training and variability issues due to the non-Euclidean geometry of the motion manifold structure. To fix this issue, existing methods such as [135] design a method based on graph convolutional networks (GCN) [82, 144, 200] to tackle the problem of skeleton generation from the input music. Specifically, Ren et al. [135] present a pose perceptual loss relied on a pre-trained GCN network to match intermediate features, assuming GCN contains high-level spatial structural information for the human skeleton structure. However,

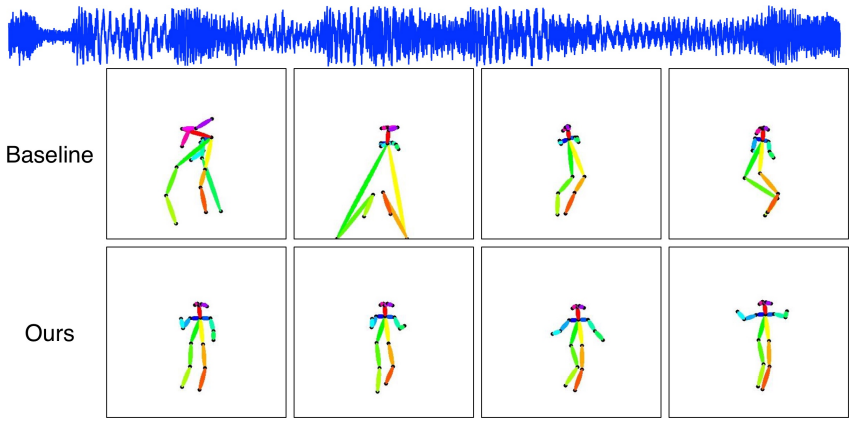


Figure 7.1: Skeleton sequence generated by baseline [135] and our proposed method according to the same input music.

modeling the skeleton as a graph solely in the level of loss calculation is insufficient to capture the dependencies and correlations between human joints, and thus results in potential distorted actions, as shown in Figure 7.1.

To address this limitation, hereby, we explicitly treat each human joint as a single node in the graph during the generation stage since the human skeleton has a graph-based representation by nature. Moreover, to better capture dependencies between joints/nodes spatially and temporally, we further propose two graph attention networks, as shown in Figure 7.2. Specifically, the proposed self-frame spatial graph attention network (SSGA-Net) aims to understand self-frame interactions between different body parts (also see Figure 7.3 (left)). While the proposed cross-frame temporal graph attention network (CTGA-Net) targets to model cross-frame correlations (also see Figure 7.3 (right)). We then sum the outputs of the two graph attention networks to further give a better constraint on which contributes to more precise joint predictions.

To translate the generated skeleton sequence to photo-realistic dance video, existing methods always directly adopt the state-of-the-art motion transfer models [14, 94, 185, 230, 183] to achieve this goal. For example,

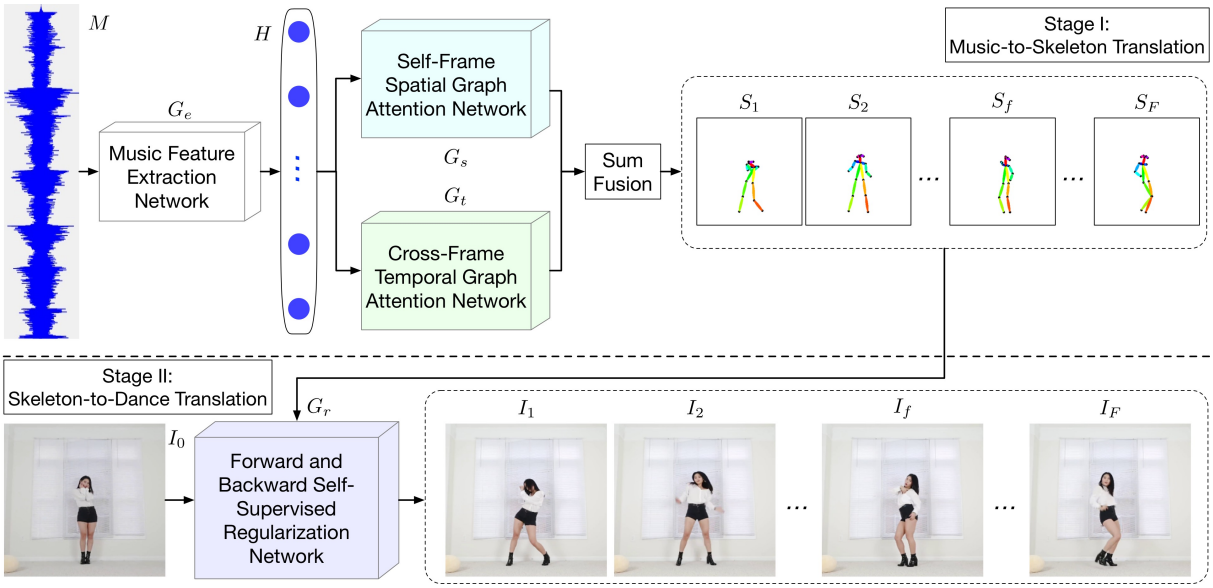


Figure 7.2: Overview of the proposed DanceGAN framework, which consists of two translation stages, i.e., music-to-skeleton translation and skeleton-to-dance translation. Stage I consists of a music feature extraction network G_e, a self-frame spatial graph attention network G_s, and a cross-frame temporal graph attention network G_t. The proposed graph attention networks G_s and G_t aim to capture dependencies between human joints from both spatial and temporal dimensions. In stage II, we propose a novel forward and backward self-supervised regularization network G_r for translating the generated skeleton sequence in stage I and an input conditional image I_0 to photo-realistic person dancing video.

Ren et al. [135] directly employ Pix2pixHD [184] to transfer the generated skeletons in stage I to the target dance video. Lee et al. [76] adopt Vid2vid [185] to convert skeleton sequences to videos. Despite their success, we still observe that existing methods such as [184, 185] produce unsatisfactory aspects and visual artifacts in the generated videos, which we believe is mainly caused by the inconsistent generation order.

To fix this issue, we propose a novel forward and backward self-supervised regularization network to generate realistic dance videos conditioned on the generated skeleton sequence and a conditional image in stage II (see Figure 7.2). The proposed self-supervised regularization network generates videos from three different strategies, i.e., baseline generation, forward generation, and backward generation (see Figure 7.4). Additionally, we in-

troduce self-supervised regularizations for forward and backward passes to facilitate video generation and enhance the training stability. As an additional contribution, we collect a new skeleton-to-dance translation dataset from the internet, which consists of 54,944 video sequences.

Overall, our contributions are summarized as follows:

- We propose a novel network architecture (DanceGAN) for the challenging music-guided dance video synthesis task, which consists of two generation stages, i.e., music-to-skeleton translation, and skeleton-to-video translation.
- We propose two graph attention networks to explicitly model dependencies across joints spatially and temporally at the same time. To the best of our knowledge, we are the first to explore the spatial and temporal graph attention networks to guide the generation of coordinated and coherent skeleton sequence from the input music. We also design a new self-supervised regularization network to enhance the video generation process from both forward and backward directions, which has not been investigated by any existing GAN-based generation works.
- We conduct extensive experiments on the music-to-skeleton translation and skeleton-to-video translation tasks. Both qualitative and quantitative results verify that the proposed DanceGAN can produce remarkably better results than existing methods. Moreover, we collected a new skeleton-to-video translation dataset, which contains 54,944 dance video sequences.

7.2 Related Work

Music-Guided Dance Video Synthesis aims to generate dance videos from the input music. Previous works such as [142, 115, 77, 37] usually use

statistical models to achieve this task. Recently, several works [3, 170, 76] use convolutional and recurrent neural networks to learn the mapping from the input music to the output dance video harnessing the development of deep learning. For instance, Tang et al. [170] introduce a music-guided dance choreography synthesis method using an LSTM-autoencoder model to learn a mapping between acoustic and motion features. Moreover, Lee et al. [76] introduce a decomposition-to-composition framework to dismantle and assemble between complex dances and basic movements conditioned on the input music. However, these methods naively treat skeleton nodes/joints as pixels of images without considering the rich self-frame and cross-frame structure information, leading to potential distorted actions.

Different from previous methods, we utilize graph convolutional networks to leverage human structure representations during the generation stage, which significantly encloses the gap towards more realistic and coherent results.

Graph-Based Models have shown to be efficient for many tasks such as skeleton-based action recognition [82, 200], semi-supervised classification [69], crowd counting [19], node classification [178], text classification [204], anomaly detection [225], face clustering [203], relation extraction [222], image segmentation [91], person image generation [155], traffic forecasting [46], and scene graph generation [17].

Unlike previous methods, we employ graph-based models to solve a new task, i.e., music-to-skeleton translation task. Ren et al. [135] present a pose perceptual loss based on a pre-trained GCN network to match the intermediate features to generate a realistic skeleton sequence from the input music. However, we discovered during experiments that solely modeling the skeleton as a graph in the level of loss calculation is not sufficient to capture the long-range dependencies and correlations between human joints, and thus potentially leads to distorted actions (see Figure 7.1). In

this paper, we consider each human joint as a node in a graph during the generation stage and further design two graph attention networks to capture the long-range dependencies between joints from both spatial and temporal dimensions. To the best of our knowledge, this spatial-temporal graph approach has not been investigated in any existing GAN-based human skeleton generation work.

Self-Supervised Learning has shown effectiveness in many tasks such as semantic segmentation [190, 118], depth estimation [153], object recognition [227], representation learning [42, 56], image generation [173, 24], action recognition [57], optical flow estimation[93], and commonsense reasoning [70].

Different from these methods, in this paper, we adopt self-supervised learning to tackle the skeleton-to-dance translation task. Specifically, we propose a novel self-supervised regularization network to generate realistic dance videos from both forward and backward passes. Moreover, the self-supervised regularizations for these two passes are proposed to facilitate the video generation process and thus enhance the training stability, which has not been considered in any existing GAN-based video generation methods.

7.3 Model Description

We start by introducing the details of the proposed DanceGAN, which consists of two generation stages (i.e., music-to-skeleton translation and skeleton-to-dance translation). An illustration of the proposed DanceGAN framework is shown in Figure 7.2. Stage I mainly contains three parts, i.e., a music feature extraction network G_e extracting the style and beat features from the input music M, a spatial graph attention network G_s modeling the long-range correlations between human parts within each frame, a temporal graph attention network G_t capturing cross-dependencies be-

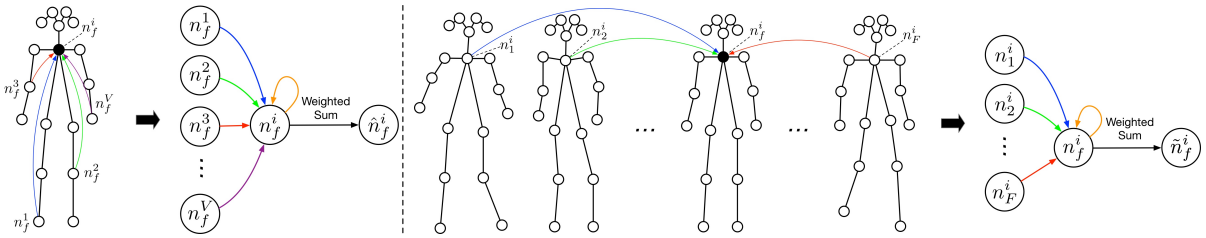


Figure 7.3: Overview of the proposed self-frame spatial graph attention network (SSGA-Net) (left) and cross-frame temporal graph attention network (CTGA-Net) (right). Different arrow colors represent independent attention computations in the same frame from the spatial dimension or cross frames from the temporal dimension. The aggregated nodes from each node are summed in a weighed way to obtain $\hat{n}_f^i$ and $\tilde{n}_f^i$ in the spatial and temporal dimension, respectively. These two results represent the strength of the connections between each pair of nodes from both spatial and temporal dimensions.

tween joints across different frames. In stage II, we take the generated skeleton sequence in stage I and a conditional image I_0 as inputs, aiming to generate a photo-realistic dance video. To this end, we introduce a new self-supervised regularization network to enhance the video generation process in both forward and backward generation directions. In the following, we first present the proposed networks in each stage and then introduce the optimization objective of the proposed method.

7.3.1 Music-to-Skeleton Translation

Overview. The overview of stage I is delineated in Figure 7.2, which aims to learn a mapping from the input music M with sample rate r per frame to a joint location vector sequence S, i.e., G_I:M∈$\mathbb{R}^{Fr}$→S∈$\mathbb{R}^{F\times 2V}$, where F is the total number of video frames; V denotes the number of joints of the human skeleton in each frame, where joints are represented by a set of 2D coordinates.

Music Feature Extraction Network. We first chunk the input music M into 0.1-second pieces. Next, these pieces are fed into the music feature extraction network G_e to extract the audio feature. The network G_e consists of an audio encoder and a bi-directional two-layer GRU [23]. The

output of G_e is the hidden state $H \in \mathbb{R}^{F \times h}$, where h denotes the depth of the hidden states. Next, the hidden state H is fed to both proposed graph attention networks to jointly produce a skeleton sequence of S.

Self-Frame Spatial Graph Attention Network. The hidden state H is first fed into three Linear-BatchNorm1d-ReLU layers to improve feature representation, leading to the output $H' \in \mathbb{R}^{F \times 4h}$. Then the output H' is connected with another Linear layer to predict the initial joint locations $S' \in \mathbb{R}^{F \times 2V}$. However, we observed that S' tends to collapse and thus fails to represent a realistic skeleton sequence in our preliminary experiments (see Figure 7.1). The reason is that the model cannot capture the spatial and temporal dependencies between joints which causes large perturbations in training and thus results in a trivial solution.

To address this limitation, we present two novel graph attention networks, i.e., SSGA-Net and CTGA-Net. The proposed SSGA-Net is manifested in Figure 7.3 (left), we only present the f-th (where $f=0, \cdots, F$) frame in this figure. Let n_f^i be the i-th (where $i=0, \cdots, V$) node in the f-th frame. The goal of SSGA-Net is to model the spatial correlations between joints/nodes within each frame. Specifically, we feed S' into two convolution layers to generate two new outputs $B \in \mathbb{R}^{F \times 2V}$ and $A \in \mathbb{R}^{F \times 2V}$, respectively. Next, we perform a matrix multiplication between the transpose of A and B, and apply a Softmax function to calculate the spatial correlation map $C^s \in \mathbb{R}^{2V \times 2V}$,

$$C^s_{ji} = \frac{\exp(B_i A_j)}{\sum_{i=1}^{2V} \exp(B_i A_j)}, \tag{7.1}$$

where the spatial correlation map C^s_{ji} measures the i-th node's impact on j-th node in each frame. Meanwhile, we feed S' into a convolution layer to generate a new output $D \in \mathbb{R}^{F \times 2V}$. Then we perform matrix multiplication between D and the transpose of C^s. Lastly, we multiply it by a scale

parameter α and perform an element-wise sum with the initial joints S' to obtain the output as follow

$$\hat{S}'_j = \alpha \sum_{i=1}^{2V} (C^s_{ji} D_i) + S'_j, \tag{7.2}$$

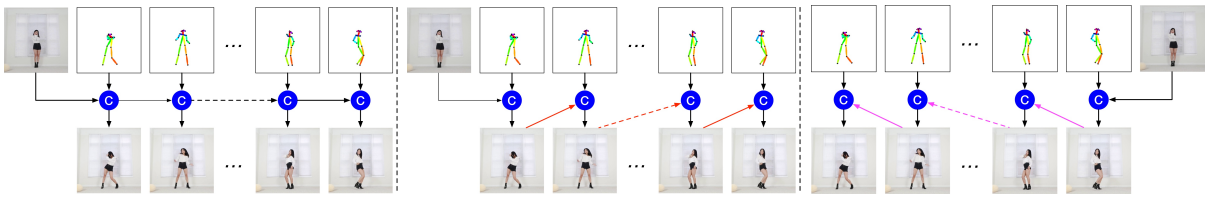
where $\hat{S}' \in \mathbb{R}^{F \times 2V}$, and α gradually learns a weight from 0.

By doing so, each predicted node in $\hat{S}'$ is a weighted sum of all the nodes in each frame. Thus, it obtains a global view of the spatial structure and can selectively adjust themselves according to the spatial correlation map, improving the representations of human structure and semantic consistency.

Cross-Frame Temporal Graph Attention Network. To generate a coherent skeleton sequence, the same node in each frame should be associated with each other. We build the CTGA-Net to explicitly model the dependencies to learn the correlations across different frames. The structure of CTGA-Net is illustrated in Figure 7.3 (right). Different from SSGA-Net, we directly calculate the temporal correlation map $C^t \in \mathbb{R}^{F \times F}$ from S'. Specifically, we perform a matrix multiplication between S' and the transpose of S'. Next, we apply a Softmax function to obtain the temporal correlation map $C^t \in \mathbb{R}^{F \times F}$ from S',

$$C^t_{ji} = \frac{\exp(S'_i S'_j)}{\sum_{i=1}^{F} \exp(S'_i S'_j)}, \tag{7.3}$$

where the temporal correlation map C^t_{ji} measures the i-th frame's impact on the j-th frame. Next, we perform a matrix multiplication between the transpose of C^t_{ji} and S'. Then we multiply the result by a scale parameter



(a) Baseline generation strategy. (b) Forward generation strategy. (c) Backward generation strategy.

Figure 7.4: Different generation strategies for the music-to-dance translation.

β and perform an element-wise sum with S' to obtain the result

$$\tilde{S}'_j = \beta \sum_{i=1}^{C} (C^t_{ji} S'_i) + S'_j, \tag{7.4}$$

where $\tilde{S}' \in \mathbb{R}^{F \times 2V}$ and β is initialized as 0.

In this way, each predicted joint/node in $\tilde{S}'$ is a weighted sum of the joints/nodes of all frames, which captures the long-range correlations across different frames to help generations of more coherent skeleton sequences.

Fusion of Graph Attention Networks. To fully harness the long-range correlations, we aggregate the results from these two graph attention networks. Specifically, we perform an element-wise summation to obtain the final joint predictions $S = \hat{S}' + \tilde{S}'$. After that, each joint/node can simultaneously perceive the joints/nodes from all spatial and temporal locations to adjust its position, leading to realistic and coherent skeleton sequence.

Optimization Objective. We follow [135] and use four different losses as our full optimization objective in the stage I,

$$\mathcal{L}_1 = \lambda_{gan}\mathcal{L}_{gan} + \lambda_p \mathcal{L}_p + \lambda_f \mathcal{L}_f + \lambda_{l1}\mathcal{L}_{l1}, \tag{7.5}$$

where $\mathcal{L}_{gan}$, $\mathcal{L}_p$, $\mathcal{L}_f$, and $\mathcal{L}_{l1}$ denote the adversarial loss, pose perceptual loss, feature matching loss, and $L1$ reconstruction loss, respectively.

7.3.2 Skeleton-to-Dance Translation

Overview. To translate the generated skeleton sequence S to a realistic dance video, we propose a new self-supervised learning regularization network G_r, as depicted in Figure 7.2. The network G_r takes S and a conditional image I_0 as inputs and aims to output video frames I_i (where i=1, 2, $\cdots$, F) by three different generation strategies outlined in Figure 7.4. In this way, the conditional image I_0 can provide appearance content while the skeleton sequence S can provide motion information to jointly generated the final video frames I_i. Moreover, the proposed method can generate different subject dancing videos with the same music as input.

Baseline Generation Strategy. As illustrated in Figure 7.4(a), we combine each skeleton and the conditional image as the input to the network to generate the corresponding images. This baseline generation strategy can be formulated as

$$I_i = G_r(\text{concat}(I_0, S_i)), \quad i=1, 2, \cdots, F. \tag{7.6}$$

However, we observe that this basic generation strategy cannot produce realistic images. Thus, we introduce two new self-supervised regularizations to boost the quality of the generated images, i.e., forward/backward self-supervised regularization (FSR/BSR).

Forward Self-Supervised Regularization. We first generate images by using the forward generation strategy (in Figure 7.4(b)). Specifically, we use the previous adjacent frame to generate the next frame, since the difference between the adjacent two frames is smaller and easier to learn. This generation strategy can be represented as

$$\hat{I}_i = G_r(\text{concat}(I_{i-1}, S_i)), \quad i=2, \cdots, F, \tag{7.7}$$

where I_{i-1}=$G_r(\text{concat}(I_0, S_{i-1}))$. We further propose the FSR to reduce

the difference between I_i and $\hat{I}_i$, this can be formulated as

$$\mathcal{L}_{fsr} = ||I_i - \hat{I}_i||_1, \quad i{=}2, \cdots, F. \tag{7.8}$$

By doing so, more constrains can be added on the network G_r to generate more realist and coherent video frames.

Backward Self-Supervised Regularization. Similar to FSR, we also propose BSR to further improve the feature representation of G_r from the backward direction. Specifically, we first generate images by using the backward generation strategy (in Figure 7.4(c)),

$$\tilde{I}_i = G_r(\text{concat}(I_{i+1}, S_i)), \quad i{=}1, 2, \cdots, F{-}1, \tag{7.9}$$

where $I_{i+1}{=}G_r(\text{concat}(I_0, S_{i+1}))$. We further propose the BSR to reduce the difference between I_i and $\tilde{I}_i$,

$$\mathcal{L}_{bsr} = ||I_i - \tilde{I}_i||_1, \quad i{=}1, 2, \cdots, F{-}1. \tag{7.10}$$

Optimization Objective. The optimization objective of the stage II can be written as

$$\mathcal{L}_2 = \lambda_{gan}\mathcal{L}_{gan} + \lambda_{l1}(\mathcal{L}_{l1} + \mathcal{L}_{fsr} + \mathcal{L}_{bsr}), \tag{7.11}$$

where $\mathcal{L}_{l1}{=}||I_i{-}I_i^{real}||_1$ $(i{=}1, 2, \cdots, F)$ and I_i^{real} are the real images. $\mathcal{L}_{gan}$ is the adversarial loss between the generated image I_i and the corresponding real one I_i^{real}.

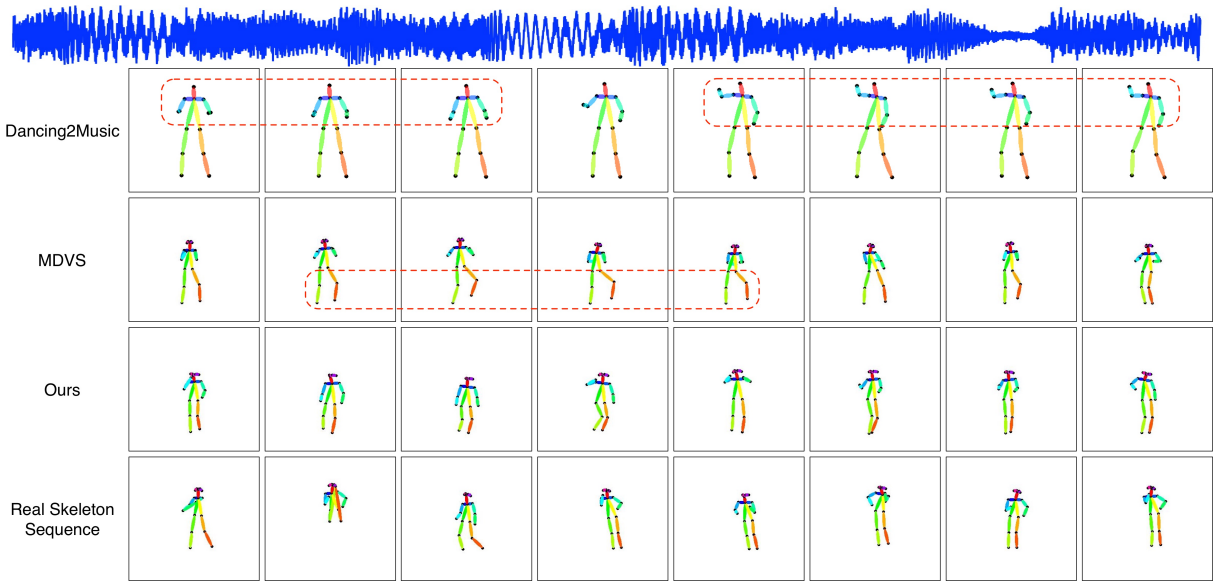


Figure 7.5: Visualization results compared with two leading methods (i.e., Dancing2Music [76], MDVS [135]) and the real skeleton sequence (ground truth). Note that we use Dancing2Music code [76] to produce its output skeletons, which are slightly different from the type of MDVS [135] and ours.

7.4 Experiments

7.4.1 Music-to-Skeleton Translation

Datasets. We use the dataset proposed in [135] for fair comparisons. Specifically, the dataset contains 100 videos, i.e., 40 k-pop videos, 20 ballet videos, and 40 popping videos. We then use OpenPose [13] to extract human skeletons, which results in 1,782 k-pop clips, 448 ballet clips, and 1518 popping clips in total. We follow [135] and select the last 10% of each type of dance for testing, and the remaining videos are used for training.

Qualitative Evaluation. We compare the proposed method with two leading music-to-skeleton methods, i.e., Dancing2Music [76] and MDVS [135]. The results are shown in Figure 7.5. It is easy to tell that Dancing2Music [76] and MDVS [135] can generate jerking dances that are prone to repeating the same movements (as shown in the red dash boxes). Compared to these two methods, our results are more realistic and coherent.

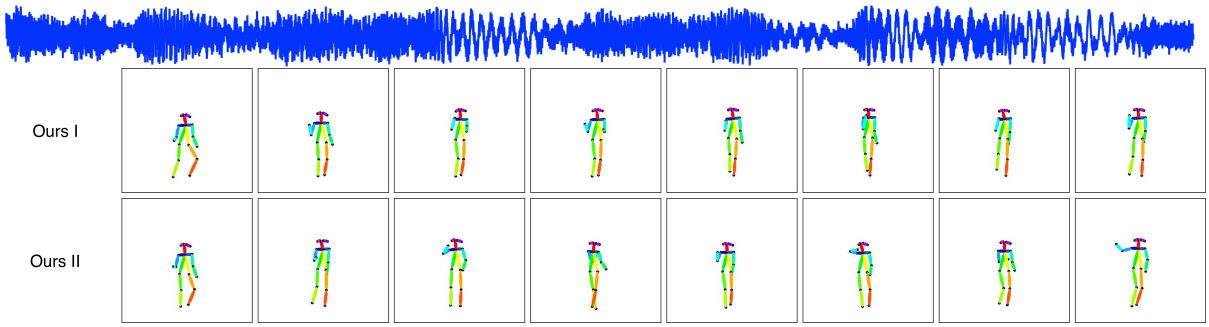


Figure 7.6: Examples of multi-modal generation by the proposed method.

Method	Motion Realism Preference ↑		Style Consistency Preference ↑	
	Expert	Non-Expert	Expert	Non-Expert
Dancing2Music [76]	3.5%	4.7%	2.9%	2.4%
MDVS [135]	28.6%	27.2%	24.2%	25.7%
DanceGAN (Ours)	**33.6%**	**32.2%**	**34.6%**	**34.7%**
Real Sequence	34.3%	35.9%	38.3%	37.2%

Table 7.1: User study compared with two leading methods (i.e., Dancing2Music [76], MDVS [135]) and the real skeleton sequence (ground truth). For each comparison, the participant is asked to answer two questions, i.e., 'which skeleton sequence is more realistic regardless of the input music', and 'which skeleton sequence matches the music style better'. The numbers indicate the preference percentage of users who favor the results of the corresponding methods or the real skeleton sequence.

Method	FID ↓	Diversity ↑
MDVS [135]	27.95	21.53
Ours	**23.42**	**26.62**

Table 7.2: State-of-the-art comparisons on the music-to-skeleton translation task.

We also show the results of the multi-modal generation in Figure 7.6. It is worth noting that, given the same input music, our method can generate different skeleton sequences.

User Study. To evaluate the quality of the generated skeleton sequences, we also conduct a user study via Amazon Mechanical Turk. Specifically, the users are first asked to answer a background question: 'Do you learn to dance or have dancing experience?'. Based on their answers, they are labeled as 'Expert' or 'None-Expert'. Then, given four skeleton se-

quences (two generated by existing methods, one generated by our proposed method, and one real skeleton sequence) with the input music, each participant needs to answer two questions: 'Which skeleton sequence is more realistic regardless of music?' and 'Which skeleton sequence matches the music better?'. The users have unlimited time to select their choices. To make this fair, we tell users in advance that there are two forms of skeletons since Dancing2Music [76] produces a slightly different form from the other three. The results compared with Dancing2Music [76], MDVS [135], and the real skeleton sequence are shown in Table 7.1. The users show more preference for our approach in terms of both motion realism and style consistency compared with the other two methods and the ground truth, which indicates the effectiveness of the proposed method.

Quantitative Evaluation. Here, we consider Fréchet Inception Distance (FID) [48] and diversity score [76] as our evaluation metrics. We generate about 70 dances based on randomly sampled music to calculate FID. Moreover, we generate around 70 dances based on randomly sampled music to calculate the diversity score the same as [135]. The results compared with MDVS [135] are shown in Table 7.2. We observe that the proposed method achieves significantly better results than MDVS in the two metrics.

7.4.2 Skeleton-to-Dance Translation

Datasets. The authors of [135] cannot release the training images due to data privacy. Therefore, to train the proposed model, we apply OpenPose [13] to collect the training data from the internet, as shown in Figure 7.7. In total, we have 54,944 video sequences and each one contains the extracted skeletons and the corresponding images. We randomly select 4,581 sequences for testing, and the remaining 50,363 sequences are used for training.

Qualitative Evaluation. Existing methods such as Dancing2Music [76]

Figure 7.7: Examples of video frames collected from the internet for the skeleton-to-dance translation task.

Method	User Preference ↑
Pix2pixHD [184]	28.9%
Vid2vid [185]	30.6%
Ours	**40.5%**

Table 7.3: User study results compared with two leading methods (i.e., Pix2pixHD [184] and Vid2vid [185]) on the skeleton-to-dance translation task.

and MDVS [135] directly employ existing motion transfer models to convert the generated skeleton sequence in stage I to realistic video frames. Specifically, Dancing2Music [76] adopts Vid2vid [185] to achieve this goal while MDVS [135] uses Pix2pixHD [184] to translate skeleton sequences to video frames. Thus, we compare the proposed method with the dominant method Vid2vid [185] in Figure 7.8, where our method generates more realistic and coherent video frames than the leading method, which further validates the effectiveness of the proposed self-supervised regularization network.

User Study. We also conduct a user study to evaluate the generated images. Participants are asked to watch a series of video triplets with the real skeleton sequence (two videos are synthesized using Pix2pixHD and

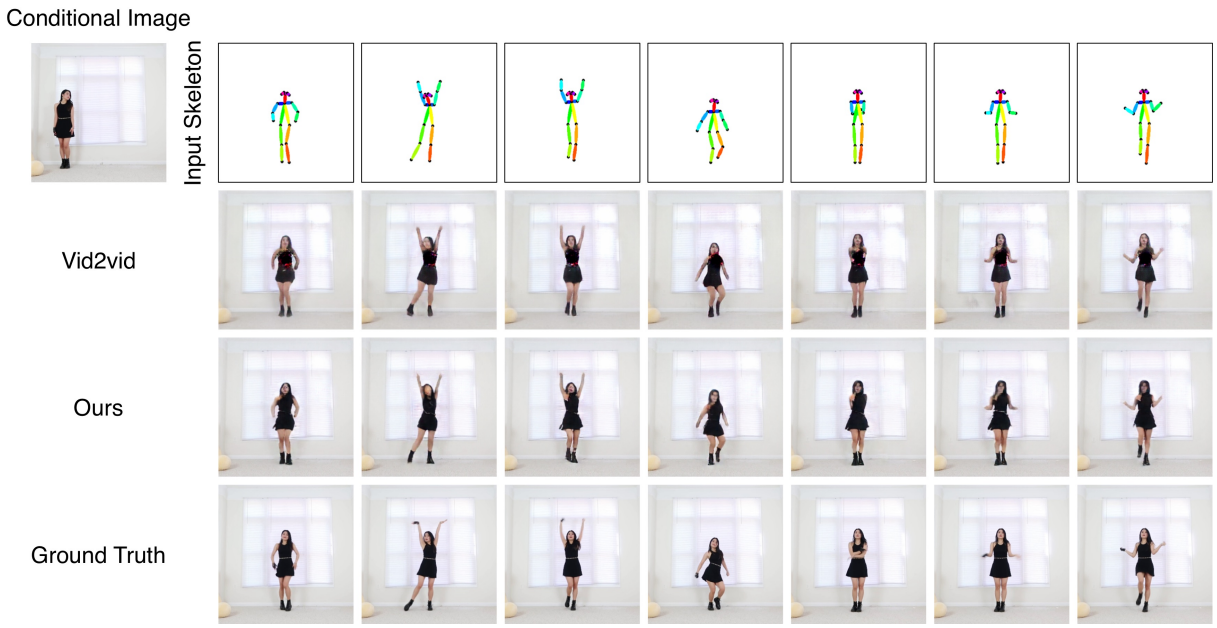


Figure 7.8: Visualization results compared with the leading method Vi2vid [185] and the ground truth. For different conditional images, the proposed method can generate different subject dancing videos with the same music.

Vid2vid, and the other by our method). They are then asked to pick the more realistic one and we give them unlimited time to respond. Each task consists of 20 video triplets and is performed by 30 distinct participants. The results are shown in Table 7.3 where more participants consider that the results of our method are more realistic than the other methods.

Quantitative Evaluation. We adopt FID [48] and LPIPS [220] to evaluate the skeleton-to-dance translation results. The comparison results with two video generation methods, i.e., Vid2vid [185] and Pix2pixHD [184] are shown in Table 7.4, from which we can see that the proposed method achieves the best results in both evaluation metrics, validating the generated video frames by our method are more realistic.

7.4.3 Ablation Study

We conduct extensive ablation studies to evaluate the effectiveness of each component of the proposed DanceGAN.

Method	FID ↓	LPIPS ↓
Pix2pixHD [184]	61.68	0.2135
Vid2vid [185]	47.65	0.1812
Ours	**35.17**	**0.1529**

Table 7.4: State-of-the-art comparisons on the skeleton-to-dance translation task.

	Setting	FID ↓
S1-1	Baseline	27.95
S1-2	S1-1 + SSGA-Net	26.14
S1-3	S1-1 + CTGA-Net	25.42
S1-4	S1-1 + SSGA-Net + CTGA-Net	**23.42**

Table 7.5: Ablation study on the music-to-skeleton task.

Baseline Models of Stage I. Stage I has four baselines as shown in Table 7.5: (i) S1-1 is our baseline using the model proposed in MDVS [135]. (ii) S1-2 means adopting the proposed SSGA-Net to model spatial correlations between different human parts. (iii) S1-3 captures long-range dependencies between joints in the temporal dimension by the proposed CTGA-Net. (iv) S1-4 is our full model and employs both SSGA-Net and CTGA-Net to further improve joint correlations.

Ablation Analysis for Stage I. The results of stage I are shown in Table 7.5 which prove that our SSGA-Net and CTGA-Net indeed improve the generation performance based on the baseline, validating the effectiveness of both proposed graph attention networks. Moreover, by adding both the proposed SSGA-Net and CTGA-Net in the baseline S1-4, the overall performance is further boosted. Several comparison results are provided in Figure 7.1.

Baseline Models of Stage II. Stage II also has four baselines as shown in Table 7.6: (i) S2-1 is our baseline, aiming to generate video frames using the baseline generation strategy. (ii) S2-2 adopts the proposed FSR to generate video frames using the proposed forward generation strategy. (iii) S2-3 generates video frames using the proposed backward generation

	Setting	FID ↓
S2-1	Baseline	27.12
S2-2	S2-1 + FSR	24.89
S2-3	S2-1 + BSR	25.15
S2-4	S2-1 + FSR + BSR	**21.53**

Table 7.6: Ablation study on the skeleton-to-dance task.

strategy and the proposed BSR. (iv) S2-4 is our full model and employs both FSR and BSR to improve the performance of video generation in a self-supervised way. To save training time, we select 11,818 and 1,182 from the whole dataset for training and testing, respectively.

Ablation Analysis of Stage II. The results of stage II are shown in Table 7.6, certifying that our FSR and BSR indeed boost the generation performance over the baseline, validating the effectiveness of proposed forward and backward self-supervised regularization. Moreover, by introducing both proposed FSR and BSR together in the baseline S2-4, the overall performance is further improved.

7.5 Conclusion

In this paper, we propose a novel DanceGAN for the challenging music-guided dance video synthesis task. The proposed DanceGAN consists of two generation stages, i.e., music-to-skeleton translation and skeleton-to-video translation. We introduce three novel networks for the two generation stages, i.e., spatial graph attention network, temporal graph attention network, and self-supervised regularization network. The two graph attention networks are employed to model correlations between joints in self-frame and cross-frame, resulting in a realistic and coherent skeleton sequence from the input music. The third network is used to enhance the video generation ability from both forward and backward directions. Lastly, we

also introduce a new skeleton-to-video translation dataset, which contains 54,944 video sequences. Extensive experiments demonstrate that DanceGAN achieves significantly better performance than the state-of-the-art approaches.

Chapter 8

Conclusion

8.1 Contribution

In this thesis, we explore Generative Adversarial Network [41] to learn to generate images including human faces, hands, bodies, and natural scenes. Specifically:

Part I. Person Image Generation. Chapter 2 describes the proposed GestureGAN [163] for hand gesture-to-gesture translation, which can generate target images with arbitrary poses, sizes, structures, and locations in the wild. We also propose three novel objective functions to better optimize the proposed GAN model, i.e., color loss, controllable structure guided cycle consistency loss, and self-content preserving loss. These optimization functions and the proposed GAN framework are jointly trained in an end-to-end fashion to improve both the quality and fidelity of the generated images. Lastly, we introduce an efficient Fréchet ResNet Distance (FRD) metric to evaluate the similarity of the real and generated images, which is more consistent with human judgment.

Chapter 3 introduces the proposed C2GAN [164] for facial expression-to-expression translation, which organizes the guidance and the image data in an interactive manner, instead of using as input only the guidance information. The proposed cycle in cycle network structure is a new de-

sign that explores the effective use of cross-modal information for guided image-to-image translation tasks. The designed cycled sub-networks connect different modalities and implicitly constraint each other, leading to extra supervision signals for better image generation. We also investigate cross-modal discriminators and cycle losses for more robust network optimization.

Chapter 4 presents the proposed XingGAN [156] and BiGraphGAN [155] for pose-guided preson image generation, respectively. XingGAN explores cascaded guidance with two different generation branches, and aims at progressively producing a more detailed synthesis from both person shape and appearance embeddings. We then propose SA and AS blocks, which effectively transfer and update person shape and appearance features in a crossing way to mutually improve each other, and are able to significantly boost the quality of the final outputs.

Moreover, the proposed BiGraphGAN aims to progressively reason the pose-to-pose and pose-to-image relations via two novel proposed blocks. We also propose a novel Bipartite Graph Reasoning (BGR) block to effectively reason the crossing long-range relations between the source pose and the target pose in a bipartite graph by using Graph Convolutional Networks (GCNs). Finally, we present a new Interaction-and-Aggregation (IA) block to interactively enhance both person's appearance and shape feature representations.

Part II. Scene Image Generation. Chapter 5 introduces the proposed SelectionGAN [165] for cross-view image translation, which explores cascaded semantic guidance with a coarse-to-fine inference, and aims at producing a more detailed synthesis from richer and more diverse multiple intermediate generations. We also propose a novel multi-scale spatial pooling & channel selection module, which is utilized to automatically enhance the multi-scale feature representation in both spatial and channel dimensions.

We introduce a novel multi-channel attention selection module, which is utilized to attentively select interested intermediate generations and is able to significantly boost the quality of the final output. The multi-channel attention module also effectively learns uncertainty maps to guide the pixel loss for more robust optimization.

Chapter 6 presents the proposed LGGAN [169] and EdgeGAN [161] for semantic image synthesis, respectively. LGGAN explores image generation from the local context, which we believe is beneficial for generating richer details compared with the existing global image-level generation methods. A new local class-specific generative structure is designed for this purpose. It can effectively handle the generation of small objects and details, which are common difficulties encountered by the global-based generation. We also propose a novel global and local generative adversarial network design able to take into account both the global and local contexts. To stabilize the optimization of the proposed joint network structure, a fusion weight map generator and a dual-discriminator are introduced. Moreover, to learn discriminative class-specific feature representations, a novel classification module is proposed. We also introduce a novel semantic-aware upsampling (SAU) to dynamically upsample a small subset of relevant pixels based on the semantic information. SAU is more efficient than deconvolution, pixel shuffle, and spatial attention, and can capture more complete semantic information than traditional upsampling methods such as nearest-neighbor interpolation.

At the same time, we propose a novel EdgeGAN for the challenging semantic image synthesis task. To the best of our knowledge, we are the first to explore the edge generation from semantic layouts and then utilize the generated edges to guide the generation of realistic images. We then propose an effective attention guided edge transfer module to selectively transfer useful edge structure information from the edge generation branch

to the image generation branch. We also design a new semantic preserving module to highlight class-dependent feature maps based on the input semantic label map for generating semantically consistent results. Both ideas have not been investigated by any existing GAN-based generation works.

Part III. Cross-Modal Translation. Chapter 7 describes the proposed DanceGAN for cross-modal audio-to-video translation, which consists of two generation stages, i.e., music-to-skeleton translation, and skeleton-to-video translation. We also propose two graph attention networks to explicitly model dependencies across joints spatially and temporally at the same time. To the best of our knowledge, we are the first to explore the spatial and temporal graph attention networks to guide the generation of coordinated and coherent skeleton sequence from the input music. We also design a new self-supervised regularization network to enhance the video generation process from both forward and backward directions, which has not been investigated by any existing GAN-based generation works.

In conclusion, we have proposed a few models for generating human faces, hands, bodies, and natural scenes. Although each method was originally proposed for a certain task, we later discovered that each method is universal and can be used to solve different tasks. For instance, GestureGAN can be used to solve both hand gesture generation and cross-view image translation tasks, as shown in Chapter 2. C2GAN can be used to solve facial expression generation, person pose generation, hand gesture generation, and cross-view image translation, as shown in Chapter 3. SelectionGAN can be used to solve cross-view image translation, facial expression generation, person pose generation, hand gesture generation, and semantic image synthesis, as shown in Chapter 5. Finally, our extensive experiments have shown that the introduced models can produce more visually better results compared to existing state-of-the-art methods.

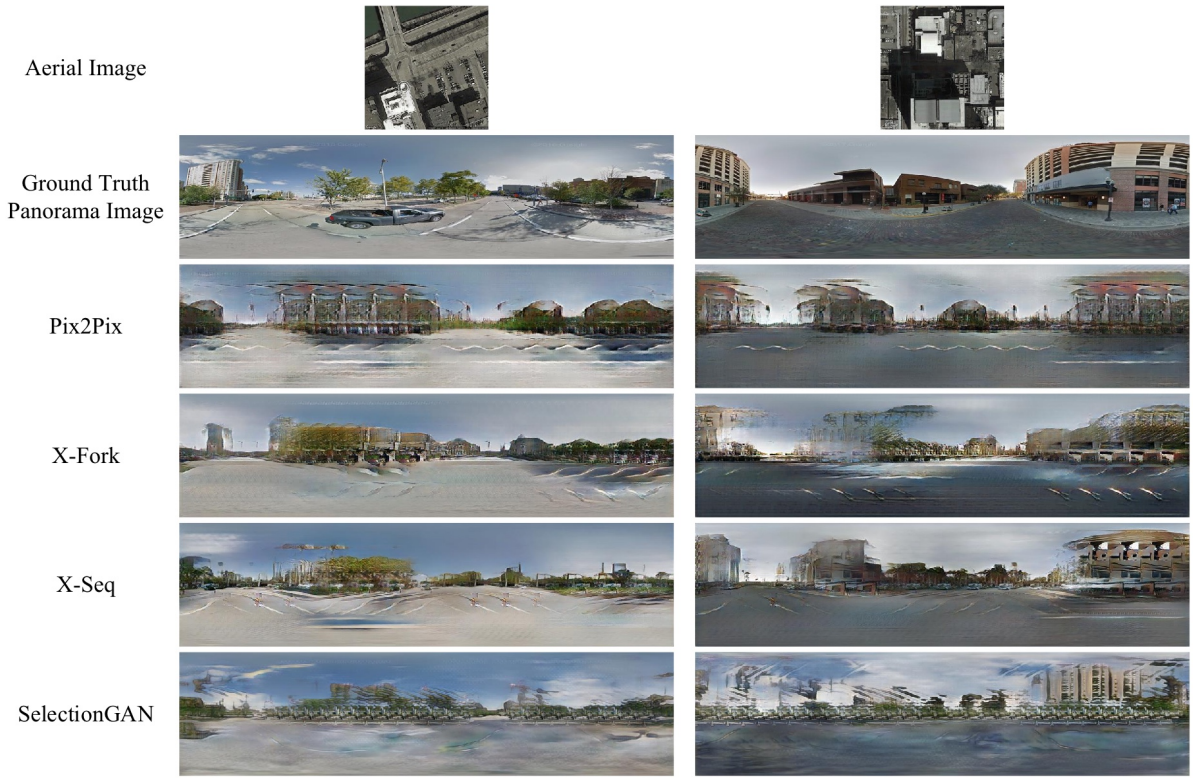


Figure 8.1: Cross-view panorama image synthesis.

8.2 Futher Work

Below, we discuss several potential future directions, building on our current visual synthesis and manipulation algorithms.

Cross-View Panorama Image Synthesis. Despite significant recent progress on cross-view image generation [165, 169, 133], it remains difficult to synthesize ground-view panorama images conditioned on top-view aerial images (Figure 8.1). Among the core challenges are the difference in image resolution between the aerial and panorama images, and the limited aside information available for viewpoint transformation. In the future, we would like to explore how to effectively and progressively generate panorama images from the top-view aerial images.

Text-Guided Image Editing. Text-guided image editing aims to manipulate source images according to given text descriptions (Figure 8.2).

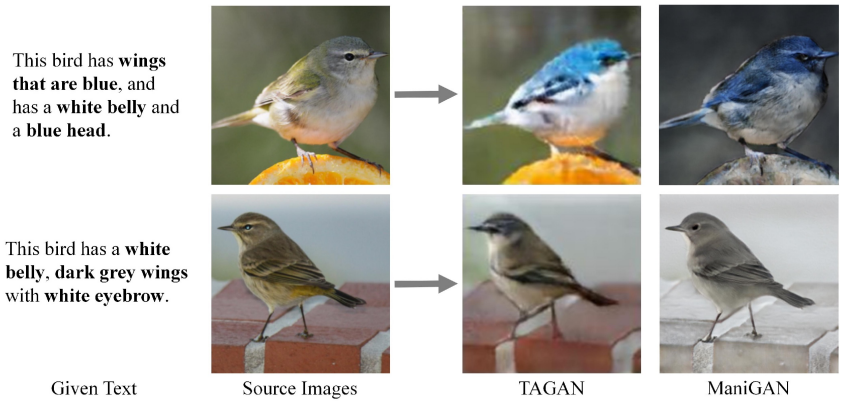


Figure 8.2: Text-guided image editing task aims to edit the source images according to a given text-guidance. Current text-guided editing models cannot manipulate the source images while reconstructing fine-grained features (e.g., TAGAN [106]) or preserving text-irrelevant source image feature (e.g., ManiGAN [79]).

It is a hard task since it should manipulate the text-required features in the image and keep the text-irrelevant parts unchanged. Although previous text-to-image works have shown remarkable progress, guaranteeing the balance between manipulation and preservation remains challenging. In the future, we would like to explore how to effectively and dynamically combine both text and image domains, and to align both low-level statistics and high-level semantics across domains to further improve the manipulation quality.

Cross-View Exocentric to Egocentric Video Synthesis. Cross-view video synthesis seeks to generate video sequences of one view from another dramatically different view. In this paper, we investigate exocentric (third-person) view to egocentric (first-person) view video generation task (Figure 8.3). This is challenging since the egocentric view sometimes is remarkably different from the exocentric view. Thus, transforming the appearances across the two views is a non-trivial task. In the future, we would like to explore how to effectively learn both the spatial and temporal information to generate egocentric video sequences from the exocentric view.

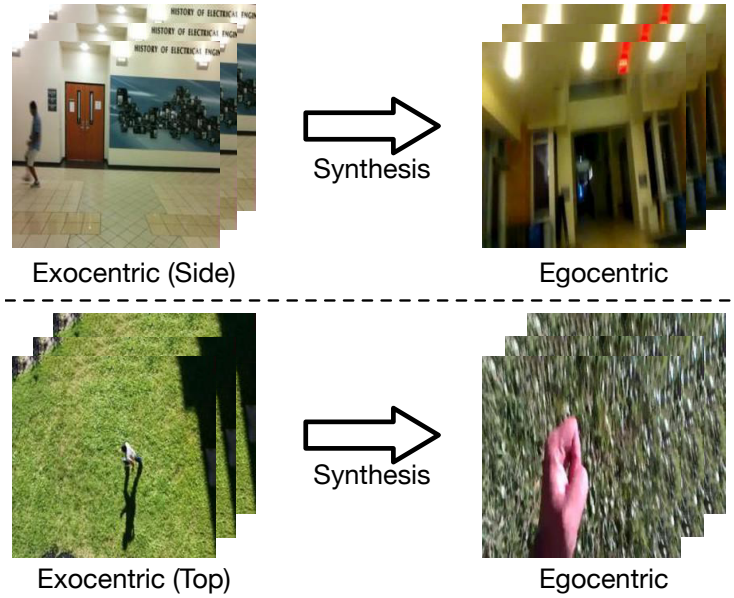


Figure 8.3: The goal of exocentric to egocentric cross-view video synthesis is to generate video sequence from exocentric perspective (Side/Top) to egocentric perspective.

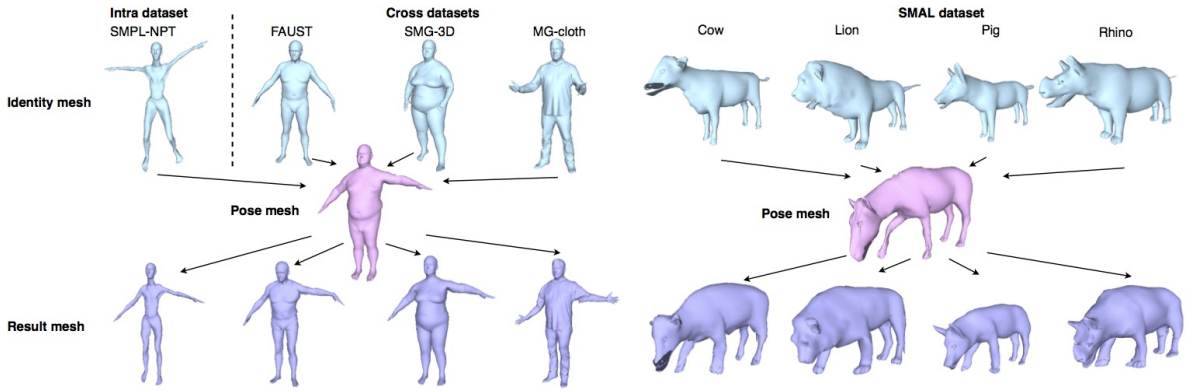


Figure 8.4: Examples of our 3D pose transfer results. Blue, pink, and purple colors stand for identity, pose, and result meshes, respectively.

3D Pose Transfer. Endowing desired poses to different identity human meshes is a promising and challenging task in computer vision. In the previous work, the correspondence of desired pose and identity body meshes should be provided as constraints for training the model, which limits the models to be generalized to other unseen domains. The intuition of this work is that the pose transfer essentially is a deformation procedure depended on the inconsistency of the given meshes. By perceiving and mining

the geometric inconsistency, the correspondence between meshes could be implicitly and efficiently learned by networks. In the future, we would like to extend the person image generation task [155, 156] to the 3D pose transfer task (Figure 8.4), and explore how to adaptively perceive the global geometric inconsistency across the given meshes.

Acknowledgments

I want to thank many important people who support and help me during my Ph.D time, who actually made this thesis possible. My supervisor Prof. Nicu Sebe, is a perfect advisor I have ever met in my academic experience. He does not perform only as a teacher, but also as a close friend to us. He always cares about our research and our life, and tries his best to help us when we meet any difficulties. Prof. Philip Torr, was my supervisor when I was doing a visting Ph.D. student at the University of Oxford. I am always impressed by the means that he thinks on the researches. Prof. Yan Yan, was my supervisor when I was doing a visting Ph.D. student at the Texas State University. He is dedicated to the reaserch and every time after the discussion with him, I would have a deeper understanding of the problems. Prof. Dan Xu is a very close collaborator during my whole Ph.D. time. I am very pleased to have worked with him during these years and I learned a lot of from him on how to do researches.

I would thank all students and postdocs at Trento who provided tremendous support for my study and research. Special thanks to MHUG folks including Elisa Ricci, Paolo Rota, Enver Sangineto, Wei Wang, Enrico Fini, Bin Ren, Aliaksandr Siarohin, Yue Song, Guanglei Yang, Jichao Zhang, Mengyi Zhao, Yiming Wang.

I would like to thank my friends at the University of Oxford TVG lab, including Song Bai, Li Zhang, Qizhu Li, Feihu Zhang, Shuyang Sun, Xiaojuan Qi. Thanks to these guys for many helpful comments and feedback.

I would like to thank my friends at the Texas State University computer vision and machine learning lab, including Bin Duan, Gaowei Liu, Songsong Wu, Yutian Lin. Thanks to these guys for many helpful comments and feedback.

I would like to thank my coauthors from other groups and institutions: Ling Shao for collaboration in the house layout generation project, and Sergey Tulyakov for helps in the music-guided dance video synthesis project.

I would like to thank all my friends, e.g., Haoyu Chen, Xinyuan Qian, Lei Ding, Ming Tao, among others. I personally learned a lot from them, both through their incredible work and their insightful ideas.

Finally, I am grateful to my parents for their love and support during this wonderful journey. The unbroken bond between us made me the person who I am today.

Last but not least, I want to thank me for believing in me. I want to thank me for doing all this hard work, I want to thank me for having no days off, I want to thank me for never quitting, I want to thank me for just being me at all times.

Bibliography

[1] Hyemin Ahn, Jaehun Kim, Kihyun Kim, and Songhwai Oh. Generative autoregressive networks for 3d dancing move synthesis from music. *IEEE RAL*, 5(2):3500–3507, 2020. 222

[2] Badour AlBahar and Jia-Bin Huang. Guided image-to-image translation with bi-directional feature transformation. In *ICCV*, 2019. 84, 86, 87, 88, 89, 105, 112, 113, 122, 127, 128, 129, 152, 155, 171

[3] Omid Alemi, Jules Françoise, and Philippe Pasquier. Groovenet: Real-time music-driven dance movement generation using artificial neural networks. *Networks*, 8(17):26, 2017. 222, 226

[4] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016. 65, 70, 123, 137, 148

[5] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Combogan: Unrestrained scalability for image domain translation. In *ICLR*, 2018. 18, 19, 23, 57

[6] Shervin Ardeshir and Ali Borji. Ego2top: Matching viewers in egocentric and top-view videos. In *ECCV*, 2016. 138

[7] Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *CVPR*, 2018. 7

[8] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *CVPR*, 2018. 82, 84, 87, 88

[9] Aayush Bansal, Yaser Sheikh, and Deva Ramanan. Shapes and context: in-the-wild image synthesis & manipulation. In *CVPR*, 2019. 174

[10] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 7, 22, 86

[11] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *CVPR*, 2018. 168, 171, 198, 202, 209

[12] John Canny. A computational approach to edge detection. *IEEE TPAMI*, (6):679–698, 1986. 186, 193

[13] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 26, 65, 88, 104, 137, 234, 236

[14] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *ICCV*, 2019. 57, 84, 88, 223

[15] Dong Chen, Shaoqing Ren, Yichen Wei, Xudong Cao, and Jian Sun. Joint cascade face detection and alignment. In *ECCV*, 2014. 130

[16] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *ICCV*, 2017. 156, 157, 162, 163, 168, 172, 173, 174, 187, 189, 197, 198, 210, 211

[17] Tianshui Chen, Weihao Yu, Riquan Chen, and Liang Lin. Knowledge-embedded routing network for scene graph generation. In *CVPR*, 2019. 226

[18] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *CVPR*, 2018. 23

[19] Xinya Chen, Yanrui Bin, Changxin Gao, Nong Sang, and Hao Tang. Relevant region prediction for crowd counting. *Elsevier Neurocomputing*, 2020. 90, 226

[20] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. Attention-gan for object transfiguration in wild images. In *ECCV*, 2018. 8, 171

[21] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, 2019. 90, 101

[22] Lu Chi, Guiyu Tian, Yadong Mu, and Qi Tian. Two-stream video classification with cross-modality attention. In *ICCV Workshops*, 2019. 89

[23] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *SSST Workshop*, 2014. 228

[24] Hyeong-Seok Choi, Changdae Park, and Kyogu Lee. From inference to generation: End-to-end fully self-supervised generation of human face from speech. In *ICLR*, 2020. 227

[25] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 8, 18, 19, 23, 27, 54, 59, 69, 70, 87, 127, 149, 171

[26] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 155, 168, 170, 198, 202, 209, 214

[27] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016. 130

[28] Xing Di, Vishwanath A Sindagi, and Vishal M Patel. Gp-gan: Gender preserving gan for synthesizing faces from landmarks. In *ICPR*, 2018. 57, 69, 70, 149

[29] Lei Ding, Hao Tang, and Lorenzo Bruzzone. Lanet: Local attention embedding to improve the semantic segmentation of remote sensing images. *IEEE TGRS*, 2020. 127

[30] Brian Dolhansky and Cristian Canton Ferrer. Eye in-painting with exemplar generative adversarial networks. In *CVPR*, 2018. 7

[31] Haoye Dong, Xiaodan Liang, Ke Gong, Hanjiang Lai, Jia Zhu, and Jian Yin. Soft-gated warping-gan for pose-guided person image synthesis. In *NeurIPS*, 2018. 88

[32] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE TPAMI*, 39(4):692–705, 2017. 172

[33] Bin Duan, Hao Tang, Wei Wang, Ziliang Zong, Guowei Yang, and Yan Yan. Audio-visual event localization via recursive fusion by joint co-attention. In *WACV*, 2021. 127

[34] Bin Duan, Wei Wang, Hao Tang, Hugo Latapie, and Yan Yan. Cascade attention guided residue learning gan for cross-modal translation. In *ICPR*, 2020. 23, 127

[35] Aysegul Dundar, Karan Sapra, Guilin Liu, Andrew Tao, and Bryan Catanzaro. Panoptic-based image synthesis. In *CVPR*, 2020. 197, 198, 211

[36] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *CVPR*, 2018. 84, 86, 87, 88, 89, 105, 106, 107, 112, 113, 152, 155

[37] Rukun Fan, Songhua Xu, and Weidong Geng. Example-based automatic music-driven conventional dance motion synthesis. *IEEE TVCG*, 18(3):501–515, 2011. 225

[38] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019. 166, 190, 206, 207

[39] Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik. Learning individual styles of conversational gesture. In *CVPR*, 2019. 222

[40] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *CVPR*, 2019. 171

[41] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio.

Generative adversarial nets. In *NeurIPS*, 2014. 7, 10, 22, 33, 60, 65, 86, 126, 137, 171, 184, 193, 243

[42] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *CVPR*, 2019. 227

[43] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Victor Lempitsky. Coordinate-based texture inpainting for pose-guided human image generation. In *CVPR*, 2019. 82

[44] Shuyang Gu, Jianmin Bao, Hao Yang, Dong Chen, Fang Wen, and Lu Yuan. Mask-guided portrait editing with conditional gans. In *CVPR*, 2019. 168, 172

[45] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *ACM MM*, 2020. 222

[46] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI*, 2019. 226

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 101, 186, 193

[48] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 34, 139, 151, 156, 193, 198, 203, 209, 236, 238

[49] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 141, 142, 190

[50] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *CVPR*, 2019. 173

[51] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *ICCV*, 2017. 172

[52] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 89, 190

[53] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM TOG*, 36(4):107, 2017. 172

[54] Mohamed Ilyes Lakhal, Oswald Lanz, and Andrea Cavallaro. Pose guided human image synthesis by view disentanglement and enhanced weighting loss. In *ECCV*, 2018. 82

[55] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 8, 18, 19, 22, 23, 27, 28, 29, 33, 38, 40, 42, 43, 44, 45, 48, 54, 57, 65, 69, 70, 74, 75, 76, 87, 126, 135, 137, 139, 140, 145, 146, 149, 151, 162, 168, 171, 174, 194, 195, 196, 197

[56] Simon Jenni, Hailin Jin, and Paolo Favaro. Steering self-supervised feature learning beyond local pixel statistics. In *CVPR*, 2020. 227

[57] Simon Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. In *ECCV*, 2020. 227

[58] Liming Jiang, Changxu Zhang, Mingyang Huang, Chunxiao Liu, Jianping Shi, and Chen Change Loy. Tsit: A simple and versatile

framework for image-to-image translation. In *ECCV*, 2020. 163, 173, 197, 198, 211

[59] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, 2018. 173

[60] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 31, 32, 38, 111, 136, 143

[61] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *CVPR*, 2018. 171

[62] Hsuan-Kai Kao and Li Su. Temporally guided music-to-body-movement generation. In *ACM MM*, 2020. 222

[63] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 7, 22, 86, 126, 171

[64] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018. 135

[65] Jae Woo Kim, Hesham Fouad, and James K Hahn. Making them dance. In *AAAI Fall Symposium: Aurally Informed Performance*, 2006. 222

[66] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *ICLR*, 2020. 87, 127, 171

[67] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jungkwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017. 18, 19, 23, 57

[68] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 33, 66, 97, 104, 137, 193

[69] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 90, 226

[70] Tassilo Klein and Moin Nabi. Contrastive self-supervised learning for commonsense reasoning. In *ACL*, 2020. 227

[71] Iryna Korshunova, Wenzhe Shi, Joni Dambre, and Lucas Theis. Fast face-swap using convolutional neural networks. In *ICCV*, 2017. 57

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 186

[73] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018. 89

[74] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. *Taylor & Francis Cognition and emotion*, 24(8):1377–1388, 2010. 69, 148

[75] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, 2020. 168, 171, 202, 209

[76] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *NeurIPS*, 2019. 222, 224, 226, 234, 235, 236, 237

[77] Minho Lee, Kyogu Lee, and Jaeheung Park. Music similarity-based approach to generating dance motion sequence. *Springer MTA*, 62(3):895–912, 2013. 225

[78] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. In *NeurIPS*, 2019. 8, 127, 171

[79] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip HS Torr. Manigan: Text-guided image manipulation. In *CVPR*, 2020. 248

[80] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Perceptual generative adversarial networks for small object detection. In *CVPR*, 2017. 7

[81] Jingyuan Li, Fengxiang He, Lefei Zhang, Bo Du, and Dacheng Tao. Progressive reconstruction of visual structure for image inpainting. In *ICCV*, 2019. 174

[82] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019. 222, 226

[83] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *arXiv preprint arXiv:1910.09777*, 2019. 203, 209

[84] Peipei Li, Yibo Hu, Qi Li, Ran He, and Zhenan Sun. Global and local consistent age generative adversarial networks. In *ICPR*, 2018. 172

[85] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 2018. 101

[86] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Joint image filtering with deep convolutional networks. *IEEE TPAMI*, 41(8):1909–1923, 2019. 89

[87] Yining Li, Chen Huang, and Chen Change Loy. Dense intrinsic appearance flow for human pose transfer. In *CVPR*, 2019. 88, 89

[88] Dong Liang, Rui Wang, Xiaowei Tian, and Cong Zou. Pcgan: Partition-controlled human image generation. In *AAAI*, 2019. 84, 88

[89] Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. Coco-gan: generation by parts via conditional coordinating. In *ICCV*, 2019. 172

[90] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 34, 66, 123, 137

[91] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, 2019. 226

[92] Gaowen Liu, Hao Tang, Hugo Latapie, and Yan Yan. Exocentric to egocentric image generation via parallel generative adversarial network. In *ICASSP*, 2020. 8, 87, 127, 172

[93] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang.

Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *CVPR*, 2020. 227

[94] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *ICCV*, 2019. 82, 84, 88, 223

[95] Xihui Liu, Zhe Lin, Jianming Zhang, Handong Zhao, Quan Tran, Xiaogang Wang, and Hongsheng Li. Open-edit: Open-domain image manipulation with open-vocabulary instructions. In *ECCV*, 2020. 127

[96] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, et al. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *NeurIPS*, 2019. 163, 168, 169, 174, 187, 210, 211, 212, 214

[97] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016. 84, 86, 104, 112, 152, 168, 171, 202, 209

[98] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NeurIPS*, 2017. 8, 18, 19, 20, 24, 25, 26, 29, 30, 33, 36, 37, 38, 54, 57, 59, 60, 64, 66, 67, 68, 69, 70, 72, 82, 84, 86, 87, 88, 89, 104, 105, 106, 107, 108, 112, 113, 114, 122, 123, 127, 129, 137, 143, 149, 151, 152, 153, 155

[99] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In

CVPR, 2018. 19, 24, 37, 38, 68, 72, 82, 84, 86, 87, 88, 89, 105, 112, 113, 151, 152, 155

[100] Youssef Alami Mejjati, Christian Richardt, James Tompkin, Darren Cosker, and Kwang In Kim. Unsupervised attention-guided image-to-image translation. In *NeurIPS*, 2018. 8, 87, 171

[101] Alvise Memo and Pietro Zanuttigh. Head-mounted gesture controlled interface for human-computer interaction. *Springer MTA*, 77(1):27–53, 2018. 36, 71, 150

[102] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *CVPR*, 2018. 173

[103] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 8, 87, 126, 171

[104] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 193

[105] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Instagan: Instance-aware image-to-image translation. In *ICLR*, 2019. 8

[106] Seonghyeon Nam, Yunji Kim, and Seon Joo Kim. Text-adaptive generative adversarial networks: Manipulating images with natural language. In *NeurIPS*, 2018. 248

[107] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *ICCV Workshops*, 2019. 174

[108] Kamyar Nazeri, Harrish Thasarathan, and Mehran Ebrahimi. Edge-informed single image super-resolution. In *ICCV Workshops*, 2019. 174

[109] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. Dense pose transfer. In *ECCV*, 2018. 88

[110] Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. In *NeurIPS*, 2017. 33

[111] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 166, 173, 207

[112] Evangelos Ntavelis, Andrés Romero, Iason Kastanis, Luc Van Gool, and Radu Timofte. Sesame: Semantic editing of scenes by adding, manipulating or erasing objects. In *ECCV*, 2020. 174

[113] Augustus Odena. Semi-supervised learning with generative adversarial networks. In *ICML Workshops*, 2016. 8

[114] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017. 171

[115] Ferda Ofli, Yasemin Demir, Yücel Yemez, Engin Erzin, A Murat Tekalp, Koray Balcı, İdil Kızoğlu, Lale Akarun, Cristian Canton-Ferrer, Joëlle Tilmanne, et al. An audio-driven dancing avatar. *Springer Journal on Multimodal User Interfaces*, 2(2):93–103, 2008. 225

[116] Andrea Palazzi, Guido Borghi, Davide Abati, Simone Calderara, and Rita Cucchiara. Learning to map vehicles into bird's eye view. In *ICIAP*, 2017. 138, 168, 193

[117] Boxiao Pan, Zhangjie Cao, Ehsan Adeli, and Juan Carlos Niebles. Adversarial cross-domain action recognition with co-attention. In *AAAI*, 2020. 7

[118] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *CVPR*, 2020. 227

[119] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 8, 20, 24, 54, 57, 58, 87, 88, 89, 145, 146, 149, 151, 153, 155, 156, 157, 162, 163, 166, 168, 169, 172, 173, 174, 175, 179, 187, 189, 193, 197, 198, 199, 200, 202, 203, 205, 209, 210, 211, 212, 213, 214

[120] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 66, 104

[121] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 19, 29

[122] Wei Peng, Jingang Shi, Zhaoqiang Xia, and Guoying Zhao. Mix dimension in poincaré geometry for 3d skeleton-based action recognition. In *ACM MM*, 2020. 90

[123] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M Álvarez. Invertible conditional gans for image editing. In *NeurIPS Workshop*, 2016. 8

[124] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 89

[125] Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. Unsupervised person image synthesis in arbitrary poses. In *CVPR*, 2018. 87

[126] Guo-Jun Qi, Liheng Zhang, Hao Hu, Marzieh Edraki, Jingdong Wang, and Xian-Sheng Hua. Global versus localized generative adversarial nets. In *CVPR*, 2018. 172

[127] Xiaojuan Qi, Qifeng Chen, Jiaya Jia, and Vladlen Koltun. Semi-parametric image synthesis. In *CVPR*, 2018. 156, 157, 162, 163, 168, 172, 174, 187, 189, 197, 198, 210, 211

[128] Xuelin Qian, Yanwei Fu, Tao Xiang, Wenxuan Wang, Jie Qiu, Yang Wu, Yu-Gang Jiang, and Xiangyang Xue. Pose-normalized image generation for person re-identification. In *ECCV*, 2018. 82

[129] Fengchun Qiao, Naiming Yao, Zirui Jiao, Zhihao Li, Hui Chen, and Hongan Wang. Geometry-contrastive generative adversarial network for facial expression synthesis. *arXiv preprint arXiv:1802.01822*, 2018. 54

[130] Tingting Qiao, Jing Zhang, Duanqing Xu, and Dacheng Tao. Mirrorgan: Learning text-to-image generation by redescription. In *CVPR*, 2019. 23

[131] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 179

[132] Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. In *NeurIPS*, 2016. 8, 19, 24, 54, 171

[133] Krishna Regmi and Ali Borji. Cross-view image synthesis using conditional gans. In *CVPR*, 2018. 8, 19, 20, 24, 25, 26, 34, 39, 40, 42, 43, 44, 54, 55, 59, 64, 66, 74, 75, 76, 123, 127, 137, 138, 139, 145, 146, 147, 148, 162, 172, 194, 195, 196, 197, 247

[134] Krishna Regmi and Ali Borji. Cross-view image synthesis using geometry-guided conditional gans. *Elsevier CVIU*, 187:102788, 2019. 8, 54, 57, 74, 75, 76, 138, 145, 172, 193, 194, 196

[135] Xuanchi Ren, Haoran Li, Zijian Huang, and Qifeng Chen. Self-supervised dance video synthesis conditioned on music. In *ACM MM*, 2020. 222, 223, 224, 226, 231, 234, 235, 236, 237, 239

[136] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *ICCV*, 2019. 174

[137] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE TMM*, 15(5):1110–1120, 2013. 36, 150

[138] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 59, 63, 65

[139] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 68, 105, 112, 139, 151, 152

[140] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *ICCV*, 2019. 7, 86, 87, 171

[141] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016. 166, 173, 207

[142] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. Dancing-to-music character animation. In *Computer Graphics Forum*, 2006. 222, 225

[143] Eli Shlizerman, Lucio Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman. Audio to body dynamics. In *CVPR*, 2018. 222

[144] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *CVPR*, 2019. 222

[145] Aliaksandr Siarohin, Stéphane Lathuilière, Enver Sangineto, and Nicu Sebe. Appearance and pose-conditioned human image generation using deformable gans. *IEEE TPAMI*, 2019. 113

[146] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *CVPR*, 2018. 8, 19, 24, 26, 37, 38, 54, 57, 59, 64, 67, 68, 72, 82, 84, 86, 87, 88, 89, 104, 105, 106, 107, 108, 112, 113, 114, 129, 151, 152, 153, 155

[147] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 33

[148] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 31, 48, 97, 186, 192

[149] Lingxiao Song, Zhihe Lu, Ran He, Zhenan Sun, and Tieniu Tan. Geometry guided adversarial facial expression synthesis. In *ACM MM*, 2018. 19, 54

[150] Sijie Song, Wei Zhang, Jiaying Liu, and Tao Mei. Unsupervised person image generation with semantic parsing transformation. In *CVPR*, 2019. 87

[151] Qianru Sun, Liqian Ma, Seong Joon Oh, Luc Van Gool, Bernt Schiele, and Mario Fritz. Natural and effective obfuscation by head inpainting. In *CVPR*, 2018. 57

[152] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. In *ICLR*, 2017. 57

[153] Feitong Tan, Hao Zhu, Zhaopeng Cui, Siyu Zhu, Marc Pollefeys, and Ping Tan. Self-supervised human depth estimation from monocular videos. In *CVPR*, 2020. 227

[154] Hao Tang, Song Bai, and Nicu Sebe. Dual attention gans for semantic image synthesis. In *ACM MM*, 2020. 127, 163, 166, 173, 197, 198

[155] Hao Tang, Song Bai, Philip HS Torr, and Nicu Sebe. Bipartite graph reasoning gans for person image generation. In *BMVC*, 2020. 11, 24, 129, 226, 244, 250

[156] Hao Tang, Song Bai, Li Zhang, Philip HS Torr, and Nicu Sebe. Xinggan for person image generation. In *ECCV*, 2020. 7, 11, 24, 87, 103, 104, 112, 123, 128, 129, 171, 244, 250

[157] Hao Tang, Xinya Chen, Wei Wang, Dan Xu, Jason J Corso, Nicu Sebe, and Yan Yan. Attribute-guided sketch generation. In *FG*, 2019. 23

[158] Hao Tang and Hong Liu. A novel feature matching strategy for large scale image retrieval. In *IJCAI*, 2016. 35

[159] Hao Tang, Hong Liu, and Nicu Sebe. Unified generative adversarial networks for controllable image-to-image translation. *IEEE TIP*, 29:8916–8929, 2020. 127

[160] Hao Tang, Hong Liu, Dan Xu, Philip HS Torr, and Nicu Sebe. Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks. *arXiv preprint arXiv:1911.11897*, 2019. 57, 127, 171

[161] Hao Tang, Xiaojuan Qi, Dan Xu, Philip HS Torr, and Nicu Sebe. Edge guided gans with semantic preserving for semantic image synthesis. *arXiv preprint arXiv:2003.13898*, 2020. 12, 54, 245

[162] Hao Tang, Wei Wang, Songsong Wu, Xinya Chen, Dan Xu, Nicu Sebe, and Yan Yan. Expression conditional gan for facial expression-to-expression translation. In *ICIP*, 2019. 127

[163] Hao Tang, Wei Wang, Dan Xu, Yan Yan, and Nicu Sebe. Gesturegan for hand gesture-to-gesture translation in the wild. In *ACM MM*, 2018. 7, 8, 10, 18, 19, 22, 24, 25, 36, 37, 38, 54, 57, 66, 71, 72, 87, 122, 127, 137, 150, 151, 171, 243

[164] Hao Tang, Dan Xu, Gaowen Liu, Wei Wang, Nicu Sebe, and Yan Yan. Cycle in cycle generative adversarial networks for keypoint-guided image generation. In *ACM MM*, 2019. 11, 24, 25, 66, 82, 84, 85, 86, 87, 88, 89, 105, 107, 108, 112, 113, 114, 137, 148, 149, 152, 153, 155, 171, 243

[165] Hao Tang, Dan Xu, Nicu Sebe, Yanzhi Wang, Jason J Corso, and Yan Yan. Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation. In *CVPR*, 2019. 12, 19, 26, 40, 42, 43, 44, 48, 51, 54, 57, 64, 74, 75, 76, 87, 88, 95, 123, 125, 127, 138, 139, 142, 162, 163, 171, 172, 173, 175, 193, 195, 196, 197, 198, 244, 247

[166] Hao Tang, Dan Xu, Nicu Sebe, and Yan Yan. Attention-guided generative adversarial networks for unsupervised image-to-image translation. In *IJCNN*, 2019. 8, 57, 87, 127

[167] Hao Tang, Dan Xu, Wei Wang, Yan Yan, and Nicu Sebe. Dual generator generative adversarial networks for multi-domain image-to-image translation. In *ACCV*, 2018. 18, 19, 23, 57, 127

[168] Hao Tang, Dan Xu, Yan Yan, Jason J Corso, Philip HS Torr, and Nicu Sebe. Multi-channel attention selection gans for guided image-to-image translation. *arXiv preprint arXiv:2002.01048*, 2020. 197

[169] Hao Tang, Dan Xu, Yan Yan, Philip HS Torr, and Nicu Sebe. Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation. In *CVPR*, 2020. 12, 24, 57, 87, 88, 127, 168, 174, 206, 211, 214, 245, 247

[170] Taoran Tang, Jia Jia, and Hanyang Mao. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In *ACM MM*, 2018. 222, 226

[171] Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Fei Wu, and Xiao-Yuan Jing. Df-gan: Deep fusion generative adversarial networks for text-to-image synthesis. *arXiv preprint arXiv:2008.05865*, 2020. 23

[172] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *ECCV*, 2016. 172

[173] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for 3d control over portrait images. In *CVPR*, 2020. 227

[174] Luan Tran, Kihyuk Sohn, Xiang Yu, Xiaoming Liu, and Manmohan Chandraker. Gotta adapt'em all: Joint pixel and feature-level domain adaptation for recognition in the wild. In *CVPR*, 2019. 7

[175] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 7

[176] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *GCPR*, 2013. 168, 171, 202, 209

[177] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 127

[178] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 226

[179] Nam N Vo and James Hays. Localizing and orienting street views using overhead imagery. In *ECCV*, 2016. 39, 74, 138, 164, 168, 193

[180] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019. 7

[181] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe: Content-aware reassembly of features. In *ICCV*, 2019. 173

[182] Miao Wang, Guo-Ye Yang, Ruilong Li, Run-Ze Liang, Song-Hai Zhang, Peter M Hall, and Shi-Min Hu. Example-guided style-consistent image synthesis from semantic labeling. In *CVPR*, 2019. 68, 70, 72, 127

[183] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. In *NeurIPS*, 2019. 122, 127, 223

[184] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 18, 20, 24, 31, 54, 126, 145, 146, 149, 151, 153, 155, 156, 157, 162, 163, 169, 171, 172, 173, 174, 187, 189, 197, 198, 210, 211, 224, 237, 238, 239

[185] Tingchun Wang, Mingyu Liu, Junyan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. 223, 224, 237, 238, 239

[186] Wei Wang, Xavier Alameda-Pineda, Dan Xu, Pascal Fua, Elisa Ricci, and Nicu Sebe. Every smile is unique: Landmark-guided diverse smile generation. In *CVPR*, 2018. 57

[187] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 92

[188] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018. 90

[189] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. In *CVPR*, 2017. 7

[190] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020. 227

[191] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 70, 105, 112, 139, 149, 152

[192] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *ICCV*, 2015. 40, 138, 168, 193

[193] Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y Chang, and Shih-Wei Liao. Relgan: Multi-domain image-to-image translation via relative attributes. In *ICCV*, 2019. 87

[194] Wenqi Xian, Patsorn Sangkloy, Varun Agrawal, Amit Raj, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Texturegan: Controlling deep image synthesis with texture patches. In *CVPR*, 2018. 88

[195] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 156, 200, 213

[196] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *CVPR*, 2018. 127

[197] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018. 171

[198] Xing Xu, Huimin Lu, Jingkuan Song, Yang Yang, Heng Tao Shen, and Xuelong Li. Ternary adversarial networks with self-supervision for zero-shot cross-modal retrieval. *IEEE TCYB*, 50(6):2400–2413, 2019. 7

[199] Xing Xu, Tan Wang, Yang Yang, Lin Zuo, Fumin Shen, and Heng Tao Shen. Cross-modal attention with semantic consistence for image-text matching. *IEEE TNNLS*, 2020. 7

[200] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018. 90, 222, 226

[201] Yichao Yan, Jingwei Xu, Bingbing Ni, Wendong Zhang, and Xiaokang Yang. Skeleton-aided articulated motion generation. In *ACM MM*, 2017. 19, 20, 24, 37, 38, 57, 72, 151

[202] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *ECCV*, 2018. 82

[203] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. In *CVPR*, 2019. 226

[204] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *AAAI*, 2019. 226

[205] Zijie Ye, Haozhe Wu, Jia Jia, Yaohua Bu, Wei Chen, Fanbo Meng, and Yanfeng Wang. Choreonet: Towards music to dance synthesis with choreographic action unit. In *ACM MM*, 2020. 222

[206] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, 2017. 18, 19, 23, 54, 57

[207] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, 2017. 156, 200, 206, 213

[208] Xiaoming Yu, Yuanqi Chen, Shan Liu, Thomas Li, and Ge Li. Multi-mapping image-to-image translation via learning disentanglement. In *NeurIPS*, 2019. 8

[209] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *ICCV*, 2019. 88

[210] Mihai Zanfir, Alin-Ionut Popa, Andrei Zanfir, and Cristian Sminchisescu. Human appearance transfer. In *CVPR*, 2018. 84, 88

[211] Menghua Zhai, Zachary Bessinger, Scott Workman, and Nathan Jacobs. Predicting ground-level scene layout from aerial imagery. In *CVPR*, 2017. 40, 43, 138, 139, 145, 162, 172, 194, 196

[212] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019. 127

[213] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 171

[214] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018. 190

[215] Jichao Zhang, Jingjing Chen, Hao Tang, Wei Wang, Yan Yan, Enver Sangineto, and Nicu Sebe. Dual in-painting model for unsupervised gaze correction and animation in the wild. In *ACM MM*, 2020. 22

[216] Jichao Zhang, Yezhi Shu, Songhua Xu, Gongze Cao, Fan Zhong, Meng Liu, and Xueying Qin. Sparsely grouped multi-task generative adversarial networks for facial attribute manipulation. In *ACM MM*, 2018. 87

[217] Jichao Zhang, Meng Sun, Jingjing Chen, Hao Tang, Yan Yan, Xueying Qin, and Nicu Sebe. Gazecorrection: Self-guided eye manipulation in the wild using self-supervised generative adversarial networks. *arXiv preprint:1906.00805*, 2019. 7

[218] Li Zhang, Xiangtai Li, Anurag Arnab, Kuiyuan Yang, Yunhai Tong, and Philip HS Torr. Dual graph convolutional network for semantic segmentation. In *BMVC*, 2019. 90, 101

[219] Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. Cross-domain correspondence learning for exemplar-based image translation. In *CVPR*, 2020. 68, 70, 72

[220] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 40, 42, 48, 70, 143, 149, 203, 238

[221] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM TOG*, 9(4), 2017. 88

[222] Yan Zhang, Zhijiang Guo, and Wei Lu. Attention guided graph convolutional networks for relation extraction. In *ACL*, 2019. 226

[223] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 131

[224] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015. 67, 84, 86, 104, 108, 112, 152

[225] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H Li, and Ge Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *CVPR*, 2019. 226

[226] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 123, 137, 155, 168, 170, 198, 202, 209

[227] Mohan Zhou, Yalong Bai, Wei Zhang, Tiejun Zhao, and Tao Mei. Look-into-object: Self-supervised structure modeling for object recognition. In *CVPR*, 2020. 227

[228] Shuchang Zhou, Taihong Xiao, Yi Yang, Dieqiao Feng, Qinyao He, and Weiran He. Genegan: Learning object transfiguration and attribute subspace from unpaired data. In *BMVC*, 2017. 57

[229] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, 2016. 172

[230] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara Berg. Dance dance generation: Motion transfer for internet videos. In *ICCV Workshop*, 2019. 223

[231] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 18, 19, 21, 23, 26, 27, 31, 32, 38, 45, 50, 54, 57, 65, 171

[232] Junyan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *NeurIPS*, 2017. 18, 19, 23

[233] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *CVPR*, 2020. 174

[234] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *CVPR*, 2019. 82, 83, 84, 85, 86, 87, 88, 89, 103, 104, 105, 106, 107, 108, 111, 112, 113, 114, 115, 152, 153, 155

[235] Zhen Zhu, Zhiliang Xu, Ansheng You, and Xiang Bai. Semantically multi-modal image synthesis. In *CVPR*, 2020. 174