UNIVERSITÀ DI TRENTO
Department of
Information Engineering and Computer Science

Doctoral Programme in
Information Engineering and Computer Science

Language-Guided Video Understanding with Foundation Models

Luca Zanella

Advisor
Prof. Elisa Ricci
Università di Trento

Co-Advisor
Dr. Massimiliano Mancini
Università di Trento

April 2026

Abstract

Video understanding systems have achieved strong performance on controlled benchmarks, yet their deployment in real-world scenarios remains limited by assumptions about supervision, training data availability, and offline access to complete video sequences. These constraints are particularly restrictive in settings such as surveillance and procedural assistance, where data is scarce, privacy-sensitive, and decisions must be made online.

Recent foundation models provide a new opportunity to rethink video understanding as a language-guided inference problem. Leveraging this shift, this thesis investigates how Vision-Language Models (VLMs) and Large Language Models (LLMs) can be used to relax key deployment constraints. The proposed methods build on frozen, language-aligned representations and increasingly shift task objectives and decision logic to inference time through natural language, rather than encoding them through task-specific training.

The first contribution shows that pre-trained VLM representations can be adapted under weak supervision for video anomaly detection and recognition by exploiting the geometric structure of vision-language embeddings. The second contribution eliminates task-specific training by reformulating anomaly detection as an inference-time reasoning problem solved using LLMs. The third contribution extends this paradigm to causal, online settings by introducing a framework for video step grounding that combines Large Multimodal Models with Bayesian filtering. Finally, the thesis addresses the reliability of language model estimates over video and explores whether synthetic videos generated by text-to-video models can improve their temporal understanding without human annotation.

By reducing reliance on task-specific data and offline access to complete videos, and by separately addressing the reliability of language model estimates, the proposed methods make video understanding systems more adaptable across tasks and environments and better suited to real-world deployment constraints.

Keywords

Video Understanding, Foundation Models, Vision-Language Models, Language-Guided Inference, Training-Free Inference, Online Inference

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Outline . . . 2
  - 1.2.1 From visual-only to vision-language representations . . . 3
  - 1.2.2 From supervised training to inference-time reasoning . . . 3
  - 1.2.3 From offline processing to online inference . . . 4
  - 1.2.4 Temporal understanding as a foundational requirement . . . 4
- 1.3 Contributions . . . 5
- 1.4 Publications . . . 6

2 Delving into CLIP latent space for Video Anomaly Recognition **9**
- 2.1 Introduction . . . 9
- 2.2 Related Work . . . 12
- 2.3 Proposed Approach . . . 14
  - 2.3.1 Selector model . . . 16
  - 2.3.2 Temporal model . . . 17
  - 2.3.3 Predictions aggregation . . . 18
  - 2.3.4 Training . . . 18
- 2.4 Experiments . . . 20
  - 2.4.1 Experiment setup . . . 20
  - 2.4.2 Evaluation against baselines . . . 22
  - 2.4.3 Ablation . . . 26
- 2.5 Chapter Summary . . . 32

3 Harnessing Large Language Models for Training-free Video Anomaly Detection **35**
- 3.1 Introduction . . . 35
- 3.2 Related Work . . . 37
- 3.3 Training-Free VAD . . . 38
  - 3.3.1 Problem formulation . . . 39
  - 3.3.2 Are LLMs good for VAD? . . . 40
  - 3.3.3 LAVAD: LAnguage-based VAD . . . 42
- 3.4 Experiments . . . 44
  - 3.4.1 Comparison with state of the art . . . 47

3.4.2 Ablation study . . . 49
3.5 Chapter Summary . . . 51

4 Training-free Online Video Step Grounding 53
4.1 Introduction . . . 53
4.2 Related Work . . . 55
4.3 On using Large Multimodal Models for Video Step Grounding . . . 56
4.3.1 Video Step Grounding . . . 56
4.3.2 Large Multimodal Models are strong baselines for VSG . . . 57
4.4 Bayesian Grounding with Large Multimodal Models . . . 59
4.4.1 Bayesian filtering . . . 60
4.4.2 PREDICT: modeling dependencies among steps via language and progress priors . . . 62
4.4.3 UPDATE: using LMM estimates to re-weigh the belief over steps 63
4.5 Experiments . . . 64
4.5.1 Comparison with state-of-the-art methods . . . 65
4.5.2 Ablation studies . . . 67
4.6 Chapter Summary . . . 69

5 Can Text-to-Video Generation help Video-Language Alignment? 71
5.1 Introduction . . . 71
5.2 Related Work . . . 74
5.3 Video-Language Alignment . . . 75
5.4 Can Synthetic Videos help VLA? . . . 77
5.5 SynViTA . . . 79
5.6 Experiments . . . 82
5.6.1 Comparison with state of the art . . . 83
5.6.2 Ablation study . . . 84
5.6.3 Analysis of synthetic data quality and diversity . . . 86
5.7 Chapter Summary . . . 88

6 Discussion 89
6.1 Applications . . . 89
6.2 Ethical Considerations . . . 90
6.3 Limitations . . . 91
6.4 Future Work . . . 92
6.5 Conclusions . . . 92

Bibliography 95

List of Tables

2.1 Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAD and VAR on ShanghaiTech. 22
2.2 Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAD and VAR on UCF-Crime. 23
2.3 Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAD and VAR on XD-Violence. 24
2.4 Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAR on UCF-Crime. The table highlights the top performers, with cells highlighted in red representing first place, cells in orange representing second place, and cells in yellow representing third place. 24
2.5 Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAR on ShanghaiTech. The table highlights the top performers, with cells highlighted in red representing first place, cells in orange representing second place, and cells in yellow representing third place. 24
2.6 Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAR on XD-Violence. The table highlights the top performers, with cells highlighted in red representing first place, cells in orange representing second place, and cells in yellow representing third place. 25
2.7 Ablation on representation and learning of the directions of abnormality. 'Finetuning' indicates that the last projection layer is fine-tuned. The final configuration of our model is represented by the row highlighted in grey in the table. 28
2.8 Comparisons of different architectural choices for the CoOp module. 'Shared' means that all the classes share a unified context, otherwise each class has a specific context. The final configuration of our model is represented by the row highlighted in grey in the table. 28
2.9 Ablation of different likelihood estimation methods, feature space transformations, and MIL selection. 'Features' indicates the transformation applied to CLIP features. The final configuration of our model is represented by the row highlighted in grey in the table. 29
2.10 Comparisons of different architectural choices for the Temporal model. The final configuration of our model is represented by the row highlighted in grey in the table. 29

2.11 Comparisons of different architectural choices for the Axial Transformer. The final configuration of our model is represented by the row highlighted in grey in the table. 30
2.12 Ablation of the losses on the Selector model. The final configuration of our model is represented by the row highlighted in grey in the table. . 30
2.13 Ablation of losses on the aggregated outputs. The final configuration of our model is represented by the row highlighted in grey in the table. . 31
2.14 Ablation on the variation of Selector model losses. The final configuration of our model is represented by the row highlighted in grey in the table. 31
2.15 Comparisons of different features. The final configuration of our model is represented by the row highlighted in grey in the table. 31

3.1 Comparison with state-of-the-art weakly-supervised, one-class, unsupervised and training-free methods on the UCF-Crime dataset. The best results among training-free methods are highlighted in bold. 45
3.2 Comparison with state-of-the-art weakly-supervised, one-class, unsupervised and training-free methods on the XD-Violence dataset. $*$ denotes results reported in [107]. The best results among training-free methods are highlighted in bold. 46
3.3 Results of LAVAD variants w/o each proposed component on the UCF-Crime Dataset. 51
3.4 Results of LAVAD on UCF-Crime with different priors in the context prompt when querying the LLM for anomaly scores. 51
3.5 Results of LAVAD on UCF-Crime with different BLIP-2 model variants in our Image-Text Caption Cleaning technique. 52
3.6 Results of LAVAD on XD-Violence with different BLIP-2 model variants in our Image-Text Caption Cleaning technique. 52

4.1 Comparison between state-of-the-art offline methods and our online method BAGLM. 66
4.2 Ablation study on the transition model. 66
4.3 Ablation study on varying the LLM. 66
4.4 Results with oracle dependencies and step progress. 67

5.1 Results of the preliminary study on using synthetic videos generated by different text-to-video models. Increases (↑) and decreases (↓) are measured relative to the model fine-tuned without synthetic videos (*i.e.*, NONE). * indicates our reproduced results using the mPLUG-Owl 7B model checkpoint released in the original VideoCon repository. 76
5.2 Average results of the preliminary study on using synthetic videos generated by different text-to-video models, for each type of misalignment. Increases (↑) and decreases (↓) are measured relative to the model fine-tuned without synthetic videos (*i.e.*, NONE). * indicates our reproduced results using the mPLUG-Owl 7B model checkpoint released in the original VideoCon repository. 76

5.3 Comparison of SynViTA with both discriminative and generative VLMs. For the video-language entailment task, we report AUC-ROC, for zero-shot text-to-video retrieval, we report mAP, and for video question-answering, we report accuracy. * indicates our reproduced results using the mPLUG-Owl 7B model checkpoint released in the original VideoCon repository. 83
5.4 Results of the ablation study on the weighting strategy for the synthetic videos in the objective function. 84
5.5 Ablation study on varying the text-to-video model. 85

List of Figures

2.1 Comparison of various anomaly recognition methods on the ShanghaiTech, UCF-Crime, and XD-Violence datasets in terms of the mean area under the curve (mAUC) of the receiver operating characteristic (ROC) and the mean average precision (mAP) of the precision-recall curve (PRC), which calculate the mean of binary AUC ROC and AP PRC values for all anomalous classes, respectively. A higher mAUC and mAP are crucial for video anomaly recognition as they reflect the model's ability to correctly recognize the correct abnormal class. Notably, our proposed method, AnomalyCLIP, achieves the highest performance on all datasets, surpassing both the state-of-the-art methods on video anomaly detection that are re-purposed for anomaly recognition and CLIP-based video action recognition methods. 10

2.2 (a) Illustration of the CLIP space and the effects of the re-centering transformation with features of normal. When the space is not re-centered around the normality prototype $\boldsymbol{m}$, directions $\boldsymbol{d}'$ are similar, making it difficult to discern anomaly types, and feature magnitude is not linked to the degree of anomaly, making it difficult to identify anomalous events. When re-centered, the distribution of the magnitudes of features projected on each $\boldsymbol{d}$ identifies the degree of detected anomaly of the corresponding type. (b) Illustration of our proposed framework. The Selector model learns directions $\boldsymbol{d}$ using CoOp [152], and uses them to identify the likelihood of each feature $\boldsymbol{x}$ to represent an occurrence of the corresponding anomalous class. MIL selection of the top-K and bottom-K abnormal segments is performed by considering the distribution of likelihoods along the corresponding direction. A Temporal model performs temporal aggregation of the features to produce the final prediction. 14

2.3 Qualitative results for VAR on four test videos from UCF-Crime (the top two rows), two test videos from ShanghaiTech (the third row), and two test videos from XD-Violence (the bottom row). For each video, we show at the bottom the predicted probability of each frame being anomalous by our model over the number of frames. We showcase some key frames to reflect the relevance between the predicted anomaly probability and the visual content. The red shaded areas denote the temporal ground-truth of anomalies. We also indicate the predicted anomalous class for detected abnormal frames in the red boxes, while videos without detected anomalies are indicated with blue boxes as Normal. 27

3.1 We introduce the first training-free method for video anomaly detection (VAD), diverging from state-of-the-art methods that are ALL training-based with different degrees of supervision. Our proposal, LAVAD, leverages modality-aligned vision-language models (VLMs) to query and enhance the anomaly scores generated by large language models (LLMs). 36

3.2 Bar plot of the VAD performance (AUC ROC) by querying LLMs with textual descriptions of video frames from various captioning models on the UCF-Crime test set. Different bars correspond to different variants of the captioning model BLIP-2 [56], while different colors indicate two different LLMs [111, 46]. For reference, we also plot the performance of the best-performing unsupervised method [108] in a red dashed line, and that of a random classifier in a gray dashed line. 39

3.3 The anomaly score predicted by Llama [111] over time for video *Shooting033* from UCF-Crime. We highlight some sample frames with their associated BLIP-2 captions to demonstrate that the caption can be semantically noisy or incorrect (red bounding boxes are for abnormal predictions, while blue bounding boxes are for normal predictions). Ground-truth anomalies are highlighted. In particular, the caption of the frame enclosed by a blue bounding box within the ground truth anomaly fails to accurately represent the visual content, leading to a wrong classification due to the low anomaly score given by the LLM. . 40

3.4 The architecture of our proposed LAVAD for addressing training-free VAD. For each test video $\mathbf{V}$, we first employ a captioning model to generate a caption C_i for each frame $\mathbf{I}_i \in \mathbf{V}$, forming a caption sequence $\mathbf{C}$. Our *Image-Text Caption Cleaning* component addresses noisy and incorrect raw captions based on cross-modal similarity. We replace the raw caption with a caption $\hat{C}_i \in \mathbf{C}$ whose textual embedding $\mathcal{E}_T(\hat{C}_i)$ is most aligned to the image embedding $\mathcal{E}_I(\mathbf{I}_i)$, resulting in a cleaned caption sequence $\hat{\mathbf{C}}$. To account for scene context and dynamics, our *LLM-based Anomaly Scoring* component further aggregates the cleaned captions within a temporal window centered around each $\mathbf{I}_i$ by prompting the LLM to produce a temporal summary S_i, forming a summary sequence $\mathbf{S}$. The LLM is then queried to provide an anomaly score for each frame based on its S_i, obtaining the initial anomaly scores $\mathbf{a}$ for all frames. Finally, our *Video-Text Score Refinement* component refines each a_i by aggregating the initial anomaly scores of frames whose textual embeddings of the summaries are mostly aligned to the representation $\mathcal{E}_V(\mathbf{V}_i)$ of the video snippet $\mathbf{V}_i$ centered around $\mathbf{I}_i$, leading to the final anomaly scores $\tilde{\mathbf{a}}$ for detecting the anomalies (anomalous frames are highlighted) within the video. 41

3.5 We showcase qualitative results obtained by LAVAD on four test videos, including two videos (top row) from UCF-Crime and two videos from XD-Violence (bottom row). For each video, we plot the anomaly score over frames computed by our method. We display some keyframes alongside their most aligned temporal summary (blue bounding boxes for normal frame predictions and red bounding boxes for abnormal frame predictions), illustrating the relevance among the predicted anomaly score, visual content, and description. Ground-truth anomalies are highlighted. 48

3.6 Results of LAVAD on UCF-Crime over the number of K semantically similar frames used for anomaly score refinement. 51

4.1 We tackle **Video Step Grounding** with **BaGLM**, a *training-free* approach which combines Bayesian filtering with Large Multimodal Models to enable *online* inference over video streams. 54

4.2 Comparison of VSG performance on HT-Step, CrossTask, and Ego4D Goal-Step datasets, prompting LMMs with step options and video segments in an online fashion. For reference, we also show the performance of the top two performing methods from the state of the art (dark bars). 58

4.3 **Overview of BaGLM.** Given a sequence of steps, an LLM is used to estimate a dependency matrix among them. This matrix is used to compute step transition probabilities employed during the **predict** step of a Bayesian filter. As the video progresses, the transition model is updated using estimates of each step's progress from an LMM. The **update** step of the filter merges this with the predictions from the LMM, refining the output. 60

4.4 Ablation study on varying the segment duration and on the used LMM. 65
4.5 Qualitative results of BaGLM on test videos from HT-Step (*Make Milanesa*) and CrossTask (*Make a Latte*). Ground truth step boundaries and predicted step probabilities per segment are shown for both BaGLM and the off-the-shelf LMM. Arrows point to the timestamps of selected keyframes. 70

5.1 We study the problem of video-language alignment, *i.e.*, modeling the relationship between video content and text descriptions. Top: current methods use LLM-generated negative captions, which may introduce certain concepts (*e.g.*, *wearing a sombrero*) only as negatives, as they are not associated with any video. Bottom: we study whether overcoming this issue by pairing negative captions with generated videos can improve VLA. 72
5.2 Distribution of the difference between $\bar{f}(\mathbf{V}^s, t^s)$ and $\bar{f}(\mathbf{V}^s, t^r)$ for each misalignment type, averaged over three text-to-video generators. Misalignment types that result in negative differences (*i.e.*, Flip and Hallucination) are highlighted in red. Best viewed in color. 77
5.3 Overview of **SynViTA**. Given a real video $\mathbf{V}^r$ with its description t^r and a negative caption t^s (generated by an LLM), we first generate a synthetic video $\mathbf{V}^s$ based on t^s. We weigh the importance of each video using the scoring criterion ϕ. We also find the shared semantic between t^r and t^s using the longest common subsequence, obtaining t'. We train f_θ to respond with `Yes` if the input video matches its description and `No` otherwise. Additionally, we encourage the model to focus on the semantic difference between real and synthetic videos, instead of the appearance difference, using their shared semantic (*i.e.*, t'). 80
5.4 Ablation study on the proposed losses. 84
5.5 Examples of videos generated by three text-to-video models (*i.e.*, CogVideoX, LaVie, and VideoCrafter2) from LLM-generated negative captions, along with alignment scores assigned by different image-text alignment methods (*i.e.*, InstructBLIP, LLaVA-1.5, and CLIP-FlanT5). For each synthetic video $\mathbf{V}^s$ and alignment model, we show its alignment with the corresponding caption t^s, denoted as $f(\mathbf{V}^s, t^s)$, and with the real caption t^r, denoted as $f(\mathbf{V}^s, t^r)$. 87

Chapter 1

Introduction

1.1 Motivation

The proliferation of contemporary multimedia platforms such as YouTube, Netflix, and TikTok has made video a dominant medium for communication, information sharing, and entertainment. Compared to static images, video integrates sight, sound, and motion into a multi-sensory signal that more closely reflects how humans perceive and interact with the world: as a continuous stream of events shaped by temporal dynamics, causal relationships, and evolving context, rather than as isolated visual snapshots.

While these properties align naturally with human perception, enabling machines to understand video remains a significant challenge. Video understanding tasks, including anomaly detection and procedural step grounding, require models to reason not only about spatial entities but also about how these entities evolve, interact, and give rise to events over time. Historically, progress in the field has been driven by advances in supervised learning and the availability of large-scale annotated datasets, such as Kinetics and ActivityNet, which provide dense labels for action recognition and temporal localization [49, 12]. Within this paradigm, model behavior is largely determined during training, while inference is treated as a fixed and lightweight execution phase.

In practice, however, this training-centric approach introduces assumptions that limit practical deployment. Collecting task-specific annotations at scale is often infeasible, particularly in domains such as surveillance or assistive technologies, where abnormal events are rare, privacy constraints restrict data collection, and long-tail distributions dominate [104]. Moreover, much of the existing literature assumes an offline setting, where the system has access to the complete video sequence before making a prediction. This assumption is at odds with many deployment scenarios that require online inference,

in which decisions must be made incrementally as visual evidence becomes available to provide timely feedback, early warnings, or step-by-step procedural guidance [41].

Recent advances in Vision-Language Models (VLMs) and Large Language Models (LLMs) provide an opportunity to revisit these assumptions. Prior to VLMs, video understanding systems relied on latent visual representations that, while expressive, lacked direct grounding in language, tying decision logic to trained, task-specific classifiers. In contrast, VLMs align visual and textual representations in a shared semantic space, enabling visual content to be interpreted through natural language [90, 3]. Building on this alignment, LLMs can reason over language-aligned visual representations at inference time, allowing task-specific decision logic to be specified through natural language prompts rather than encoded entirely during training [125, 1].

This shift blurs the traditional boundary between training and inference in video understanding. By building on frozen, language-aligned representations and expressing task-specific decision logic at inference time through natural language, it becomes possible to reconsider long-standing assumptions about supervision and offline processing, leading to approaches that are more flexible, data-efficient, and better suited to online settings.

The core objective of this thesis is to investigate how vision-language models and large language models can be leveraged to move task-specific decision logic from training to inference through natural language, with the goal of improving scalability and deployability. In this context, scalability refers to reducing reliance on task-specific training data and offline processing, so that the same models can be applied across tasks and environments with minimal adaptation. To this end, this thesis develops a progression of methods that move from frozen vision-language representations to fully training-free and online inference. It also addresses a key requirement for this paradigm to be viable in practice, *i.e.*, the reliability of language model estimates over video, by exploring synthetic video data as a scalable tool for improving temporal understanding without human annotation.

1.2 Outline

This thesis builds on the observation that recent VLMs offer powerful, general-purpose visual representations aligned with natural language. This alignment motivates a rethinking of video understanding pipelines, moving away from encoding task-specific logic through training toward language-guided inference over frozen, language-aligned representations. Building on this perspective, the thesis investigates how VLMs and LLMs can be leveraged to support video understanding under realistic deployment

constraints.

The thesis is organized as a sequence of contributions that progressively relax assumptions commonly made in prior work. We begin in Chapter 2 by studying whether pre-trained vision-language representations can be adapted with minimal supervision for video anomaly detection and recognition. Recognizing the limitations of even weak supervision, Chapter 3 explores the complete removal of task-specific training by shifting anomaly detection entirely to inference time with LLMs. As many real-world systems must operate without access to future observations, Chapter 4 extends this training-free paradigm to online, causal video streams through probabilistic temporal modeling. Finally, in Chapter 5, we address a fundamental requirement underlying all preceding approaches: the reliability of estimates produced by language models over video, investigating whether synthetic video data can improve their temporal understanding without human annotation.

Taken together, these chapters show that relaxing assumptions on task-specific training and offline access, while addressing estimate reliability, results in video understanding systems that are more suitable for real-world deployment.

1.2.1 From visual-only to vision-language representations

A first step is to understand whether pre-trained vision-language representations already encode semantic structure that can be exploited for video understanding, particularly in settings characterized by scarce supervision and extreme class imbalance.

In Chapter 2, we introduce **AnomalyCLIP**, a framework that adapts frozen VLMs to joint video anomaly detection and recognition under weak, video-level supervision. Rather than learning visual representations from scratch, **AnomalyCLIP** reshapes the CLIP embedding space by re-centering it around a normality prototype, allowing anomaly likelihood and semantic category information to be inferred from feature magnitude and direction, respectively. We model temporal dependencies between image embeddings using a transformer-based architecture and perform training within a multiple instance learning framework specifically suited for sparse anomalies. Experimental results show that the geometric structure of vision-language embeddings can be effectively exploited for complex video understanding tasks with minimal annotation.

1.2.2 From supervised training to inference-time reasoning

While **AnomalyCLIP** demonstrates that vision-language representations can be adapted with limited supervision, it still relies on task-specific training and data collection. To

better meet real-world deployment needs and further relax these assumptions, this chapter investigates whether video anomaly detection can be performed without any task-specific training.

In Chapter 3, we present **LAVAD**, a training-free approach to video anomaly detection that entirely shifts decision-making to inference time. The proposed method leverages VLMs to generate and align textual descriptions of video frames and employs LLMs to summarize and reason over these descriptions in a zero-shot manner. By formulating anomaly detection as an inference problem solved using language models, **LAVAD** eliminates the need for training data or task-specific model adaptation. Experimental results show that competitive performance can be achieved without task-specific supervision, demonstrating the feasibility of training-free video anomaly detection in data-constrained and privacy-sensitive scenarios.

1.2.3 From offline processing to online inference

Both previous chapters assume access to the full video sequence at inference time, an assumption that limits applicability in many real-world settings where decisions must be made as visual evidence becomes available. This chapter addresses the additional constraint of online inference.

In Chapter 4, we introduce **BaGLM**, a framework for training-free, online video step grounding that combines Bayesian filtering with Large Multimodal Models (LMMs). The proposed approach incrementally updates beliefs over procedural steps as new video segments are observed, allowing step predictions to be refined incrementally over untrimmed video streams. By combining estimates from LMMs with probabilistic temporal modeling, **BaGLM** supports online inference without task-specific training or access to future observations, further advancing the deployability of video understanding systems based on language models.

1.2.4 Temporal understanding as a foundational requirement

The preceding approaches rely on the ability of language models to produce reliable estimates when operating on language-aligned visual representations over time. In practice, however, these estimates can be inconsistent in videos involving complex or subtle temporal dynamics. This chapter examines whether synthetic data can improve the temporal understanding of language models in such settings.

In Chapter 5, we introduce **SynViTA** and explore whether synthetic videos generated by text-to-video models can be used to improve this capability without human

annotation. Rather than modifying visual representations or alignment interfaces within LMMs, the approach investigates whether fine-tuning language models on synthetic video data leads to more reliable estimates over video. The results highlight the potential of synthetic video generation as a scalable tool for improving the semantic consistency and reliability of estimates produced by LMMs over video.

1.3 Contributions

The main goal of this thesis is to make video understanding systems better aligned with real-world deployment constraints by leveraging language-guided inference over foundation models. To this end, we present a series of methodological, algorithmic, and experimental contributions that demonstrate how vision-language representations can serve as a foundation for video understanding and progressively relax key deployment assumptions, including the need for task-specific training data and offline access to complete video sequences.

- **Video anomaly detection and recognition using frozen vision-language representations.** We introduce **AnomalyCLIP**, the first framework based on VLMs that jointly detects and recognizes anomalous events under weak, video-level supervision. A key technical contribution is the transformation of the VLM feature space using a normality prototype, which allows the model to encode semantic "prompt directions" to distinguish specific anomaly types. A Selector module leverages the transformed features for robust multiple instance learning segment selection, and a Temporal model aggregates both short-term and long-term temporal dependencies, resulting in more actionable outputs than traditional video anomaly detection systems.

- **Video anomaly detection without task-specific training.** We pioneer the study of training-free video anomaly detection, motivated by real-world settings where task-specific data collection is often infeasible. We introduce **LAVAD**, a framework that shifts anomaly detection entirely to inference time by using LLMs to reason over textual scene descriptions derived from VLMs. To improve robustness, we employ cross-modal similarity techniques with pre-trained VLMs to mitigate noisy captions and refine anomaly scores. Experiments show that this training-free paradigm achieves competitive performance compared to traditional unsupervised methods, providing an alternative that does not require task-specific training data.

- **Online video step grounding without offline access.** We present the first study of online video step grounding in a training-free setting. We introduce **BaGLM**, a framework demonstrating that zero-shot LMMs can outperform specialized, training-based, offline methods when combined with Bayesian filtering. The main innovation is the integration of probabilistic temporal modeling with LMM-based estimates, allowing the system to incorporate priors from past video frames into LMM predictions and to operate without access to future observations.

- **Improving language models' temporal understanding with synthetic data.** We propose a methodology that exploits synthetic videos generated by text-to-video models to improve the reliability of estimates produced by the language model component in LMMs over video, without human annotation. Through **SynViTA**, we study the benefits and limitations of current text-to-video generative models as sources of supervision. Our approach introduces a sample-weighting strategy to mitigate noisy generations and a regularization term that encourages focus on semantic rather than visual differences between videos. The proposed method is model-agnostic and improves the semantic consistency of LMM estimates across multiple architectures.

1.4 Publications

In the following list, we give an overview of the publications authored during the PhD, with entries marked with an * not being discussed in this manuscript:

- Luca Zanella, Massimiliano Mancini, Yiming Wang, Alessio Tonioni, Elisa Ricci. "Training-free Online Video Step Grounding". In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. Chapter 4 is mainly based on this publication.

- Luca Zanella, Massimiliano Mancini, Willi Menapace, Sergey Tulyakov, Yiming Wang, Elisa Ricci. "Can Text-to-Video Generation help Video-Language Alignment?". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. Chapter 5 is mainly based on this publication.

- Luca Zanella, Willi Menapace, Massimiliano Mancini, Yiming Wang, Elisa Ricci. "Harnessing Large Language Models for Training-free Video Anomaly Detection". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. Chapter 3 is mainly based on this publication.

- Luca Zanella, Benedetta Liberatori, Willi Menapace, Fabio Poiesi, Yiming Wang, Elisa Ricci. "Delving into CLIP latent space for Video Anomaly Recognition". In *Computer Vision and Image Understanding*, 2024. Chapter 2 is mainly based on this publication.

- Bartłomiej Leporowski, Amir Bakhtiarnia, Nicholas Bonnici, Adrian Muscat, Luca Zanella, Yiming Wang. "MAVAD: Audio-Visual Dataset and Method for Anomaly Detection in Traffic Videos". In *IEEE International Conference on Image Processing (ICIP)*, 2024.*

- Giulio Mattolin, Luca Zanella, Elisa Ricci, Yiming Wang. "ConfMix: Unsupervised Domain Adaptation for Object Detection via Confidence-based Mixing". In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023.*

Chapter 2

Delving into CLIP latent space for Video Anomaly Recognition

Recent works have shown that Vision-Language Models (VLMs) can be successfully extended to video and applied to supervised action recognition tasks [119]. However, it remains unclear to what extent these representations are suitable for settings characterized by scarce supervision and extreme class imbalance. In this chapter, we take a first step toward scalable video understanding by studying how pre-trained VLMs, originally trained on large-scale image-text corpora, can be adapted to such settings. Focusing on video anomaly detection and recognition, we investigate whether minimal, targeted adaptation of the VLM latent space is sufficient to support reliable anomaly detection and classification, without retraining the underlying model.

2.1 Introduction

Video anomaly detection (VAD) is the task of automatically identifying activities that deviate from normal patterns in videos [103]. VAD has been widely studied by the computer vision and multimedia communities [9, 29, 77, 83, 105, 124, 136] for several important applications, such as surveillance [104] and industrial monitoring [91].

VAD is challenging because data is typically highly imbalanced, *i.e.*, normal events are many, whilst abnormal events are rare and sporadic. VAD can be addressed as an out-of-distribution detection problem, *i.e.*, one-class classification (OOC) [68, 73, 85, 136]: only visual data corresponding to the normal state is used as training data, and an input test video is classified as normal or abnormal based on its deviation from the learned normal state. However, OOC methods can be particularly ineffective in

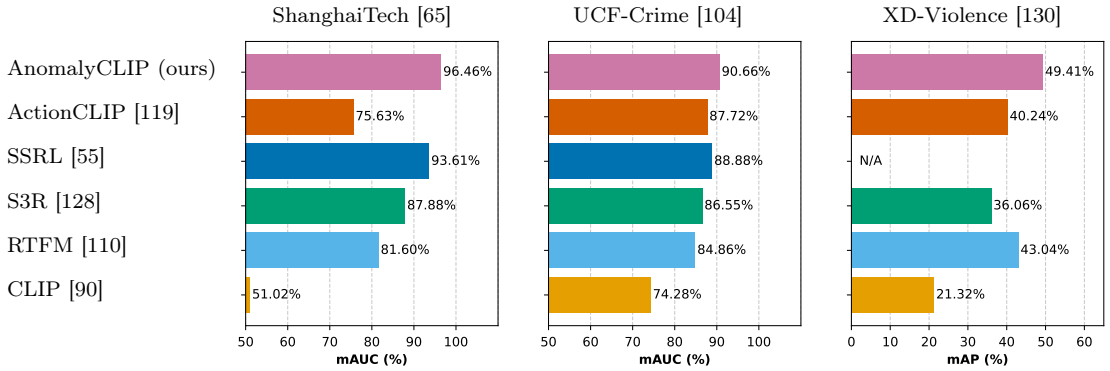


Figure 2.1: Comparison of various anomaly recognition methods on the ShanghaiTech, UCF-Crime, and XD-Violence datasets in terms of the mean area under the curve (mAUC) of the receiver operating characteristic (ROC) and the mean average precision (mAP) of the precision-recall curve (PRC), which calculate the mean of binary AUC ROC and AP PRC values for all anomalous classes, respectively. A higher mAUC and mAP are crucial for video anomaly recognition as they reflect the model's ability to correctly recognize the correct abnormal class. Notably, our proposed method, AnomalyCLIP, achieves the highest performance on all datasets, surpassing both the state-of-the-art methods on video anomaly detection that are re-purposed for anomaly recognition and CLIP-based video action recognition methods.

complex real-world applications where normal activities are diverse. An uncommon normal activity may cause a false alarm because it differs from the learned normal activities. Alternatively, VAD can be addressed with fully-supervised approaches based on frame-level annotations [6, 117]. Despite their good performance, they are considered impractical because annotations are costly to produce. Unsupervised approaches can also be used, but their performance in complex settings is not yet satisfactory [142]. For these reasons, the most recent approaches are designed for weakly-supervised learning scenarios [55, 104, 110, 129]: they exploit video-level supervision.

Whilst existing weakly-supervised VAD methods have shown to be effective in anomaly detection [55], they are generally not designed for recognizing anomaly types (*e.g.*, shooting vs. explosion). Performing Video Anomaly Recognition (VAR) in addition to VAD, which is not only detecting anomalous events but also recognizing the underlying activities, is desirable as it provides more informative and actionable insights. However, addressing VAR in a weakly-supervised setting is highly challenging due to the extreme data imbalance and the limited samples representing each anomaly [104].

We have recently experienced the emergence of powerful deep learning models that are trained on massive web-scale datasets [95]. These models, commonly referred to as Vision-Language Models (VLMs) or foundation models [90, 100], have shown strong generalization capabilities in several downstream tasks and have become a key

ingredient of modern computer vision and multimedia systems. These pre-trained models are publicly available and can be seamlessly integrated into any recognition system. VLMs can also be effectively applied to videos and to supervised action recognition tasks [119, 134].

In this chapter, we introduce the first method that jointly addresses VAD and VAR with VLMs. We argue that by leveraging representations derived from VLMs, we can obtain more discriminative features for recognizing and classifying abnormal behaviors. However, as supported by our experiments (Fig. 2.1), a naive application of existing VLMs to VAR-VAD does not suffice due to the imbalance of the training data and the subtle differences between frames of the same video containing and non-containing anomalous contents.

Therefore, we propose AnomalyCLIP, a novel solution for VAR based on the CLIP model [90], achieving state-of-the-art anomaly recognition performance as shown in Fig. 2.1.

AnomalyCLIP produces video representations that can be mapped to the textual description of the anomalous event. Rather than directly operating on the CLIP feature space, we re-center it around a normality prototype, as shown in Fig. 2.2 (a). In this way, the space assumes important semantics: the magnitude of the features indicates the degree of anomaly, while the direction from the origin indicates the anomaly type. To learn the directions that represent the desired anomaly classes, we propose a Selector model that employs prompt learning and a projection operator tailored to our new space to identify the parts in a video that better match the textual description of the anomaly. This ability is instrumental in addressing the data imbalance problem. We use the predictions of the Selector model to implement a semantically-guided Multiple Instance Learning (MIL) strategy that aims to widen the gap between the most anomalous segments of anomalous videos and normal ones. Differently from the features typically employed in VAD that are extracted using temporal-aware backbones [12, 67], CLIP visual features do not bear any temporal semantics as it operates at the image level. We thus propose a Temporal model, implemented as an Axial Transformer [40], which models both short-term relationships between successive frames and long-term dependencies between parts of the video.

As illustrated in Fig. 2.1, we evaluate the proposed approach on three benchmark datasets, ShanghaiTech [65], UCF-Crime [104], and XD-Violence [130], and empirically show that our method achieves state-of-the-art performance in VAR.

The contributions of this chapter are summarized as follows:

- we propose the first method for VAR that is based on VLMs to detect and classify

the type of anomalous events;
- we introduce a transformation of the VLM model feature space driven by a normality prototype to effectively learn the prompt directions for anomaly types;
- we propose a novel Selector model that uses semantic information imbued in the transformed VLM feature space as a robust way to perform MIL segment selection and anomaly recognition;
- we design a Temporal model to better aggregate temporal information by modeling both the short-term relationships between neighboring frames and the long-term dependencies among segments.

2.2 Related Work

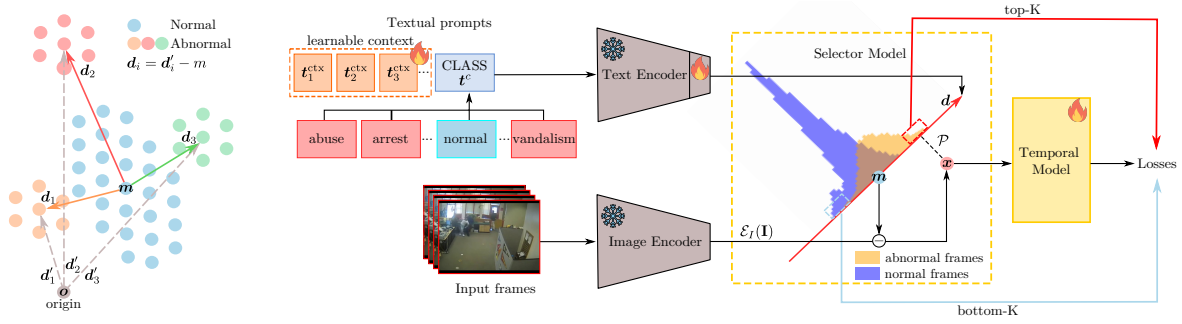
Video Anomaly Detection. Recognizing anomalous behaviors in video surveillance streams is a traditional task in computer vision and multimedia analysis. Existing methods for VAD can be grouped into four main categories based on the level of supervision available during training. The first group includes fully-supervised methods that assume available frame-level annotations in the training set [6, 117]. The second group includes weakly-supervised approaches that only require video-level normal/abnormal annotations [55, 57, 104, 110, 129]. The third group includes one-class classification methods that assume the availability of only normal training data [68, 73, 85]. The fourth group includes unsupervised models that do not use training data annotations [82, 142].

Among these types of methods, weakly-supervised approaches have gained higher popularity, as they typically yield good results while limiting the annotation effort. [104] were the first to formulate weakly-supervised VAD as a multiple-instance learning (MIL) task, dividing each video into short segments that form a set, known as *bag*. Bags generated from abnormal videos are called positive bags, and those generated from normal videos are called negative bags. Since this pioneering work, MIL has become a paradigm for VAD, and several subsequent works have proposed refining the associated ranking model to more robustly predict anomaly scores. For example, [110] proposed a Robust Temporal Feature Magnitude (RTFM) loss that is applied to a deep network consisting of a pyramid of dilated convolutions and a self-attention mechanism to model both short-term and long-term relationships between video snippets close in time and events in the whole video. [128] introduced Self-Supervised Sparse Representation Learning, an approach that combines dictionary-based representation with self-supervised learning techniques to identify abnormal events. [15] introduced Magnitude-Contrastive Glance-and-Focus Network, a neural network that uses a feature

amplification mechanism and a magnitude contrastive loss to enhance the importance of feature discriminative for anomalies. Motivated by the fact that anomalies can occur at any location and at any scale of the video, [55] proposed Scale-Aware Spatio-Temporal Relation Learning (SSRL), an approach that extends RTFM by not only learning short-term and long-term temporal relationships but also learning multi-scale region-aware features. While SSRL achieves state-of-the-art results in common VAD benchmarks, its high computational complexity limits its applicability. To the best of our knowledge, no previous works have explored foundation models [90] for VAD, as we propose in this chapter.

Video Anomaly Recognition. Before our work, few approaches attempted to simultaneously detect anomalies and identify their types [74, 114, 75]. Among these, [74] proposed a weakly-supervised method using a two-level attention mechanism. They begin by extracting spatio-temporal features with a 3D convolutional network, then employ an LSTM with an initial attention mechanism to capture temporal information. Finally, they perform detection and classification through two separate branches, linking them with a second-level attention mechanism. [114] introduced Vision Transformer Anomaly Recognition Network (ViT-ARN), which first detects anomalies using a one-class classification network. It then uses a vision transformer to extract frame features from the detected anomalies and models the temporal information between frames with a multi-reservoir echo state network. Finally, a prediction layer recognizes the anomalies. Other works jointly performing detection and recognition fall into the less practical fully supervised setting [75]. Unlike these methods that rely on unimodal backbones, our approach is the first to utilize vision and language models for joint anomaly detection and recognition.

Vision-Language Models. The emergence of novel large multimodal neural networks [90, 95, 94, 100], which can learn joint visual-text embedding spaces, has enabled unprecedented results in several image and video understanding tasks. Current VLMs adopt modality-specific encoders and are trained via contrastive techniques to align the data representations from different modalities [45, 90]. Despite their simplicity, these methods have been shown to achieve impressive zero-shot generalization capabilities. While earlier approaches, such as CLIP [90], operate on images, VLMs have recently and successfully been extended to the video domains. VideoCLIP [134] is an example of this, and it is designed to align video and textual representations by contrasting temporally overlapping video-text pairs with mined hard negatives. VideoCLIP can achieve strong zero-shot performance in several video understanding tasks. ActionCLIP [119] models action recognition as a video-text matching problem rather than a classical 1-out-of-N



(a) (b)

Figure 2.2: (a) Illustration of the CLIP space and the effects of the re-centering transformation with features of normal. When the space is not re-centered around the normality prototype $\boldsymbol{m}$, directions $\boldsymbol{d}'$ are similar, making it difficult to discern anomaly types, and feature magnitude is not linked to the degree of anomaly, making it difficult to identify anomalous events. When re-centered, the distribution of the magnitudes of features projected on each $\boldsymbol{d}$ identifies the degree of detected anomaly of the corresponding type. (b) Illustration of our proposed framework. The Selector model learns directions $\boldsymbol{d}$ using CoOp [152], and uses them to identify the likelihood of each feature $\boldsymbol{x}$ to represent an occurrence of the corresponding anomalous class. MIL selection of the top-K and bottom-K abnormal segments is performed by considering the distribution of likelihoods along the corresponding direction. A Temporal model performs temporal aggregation of the features to produce the final prediction.

majority vote task. Similarly to ours, their method uses the feature space of CLIP to learn semantically-aware representations of videos. However, a direct exploitation of the CLIP feature space fails in capturing information on anomalous events for which a specific adaptation, proposed in this chapter, is necessary. In addition, action recognition methods often fall short in weakly-supervised VAD tasks due to data imbalance between normal and abnormal events, coupled with the need for frame-level evaluation at test time, despite only having video-level supervision. To the best of our knowledge, no prior work has specifically utilized VLMs to tackle the VAD problem.

2.3 Proposed Approach

Weakly-supervised VAD is the task of learning to classify each frame in a video as either normal or anomalous using a dataset of tuples in the form $(\mathbf{V}, y)$, where $\mathbf{V}$ is a video and y is a binary label indicating whether the video contains an anomaly in any of its frames. With respect to VAD, VAR introduces the additional task of recognizing the *type* of the detected anomaly in each frame. Therefore, VAR considers a dataset

of tuples $(\mathbf{V}, c)$, where c indicates the type of anomaly in the video ($c = \emptyset$ means no anomaly is present, thus being *Normal*). In the following, we omit the subscripts for the purpose of readability.

To address the video-level supervision and the imbalance between normal videos and abnormal ones in VAD, the Multiple Instance Learning (MIL) framework [104] is widely used. MIL models each video as a bag of segments $\mathbf{V} = [\mathbf{S}_1, ..., \mathbf{S}_T] \in \mathbb{R}^{T \times F \times D}$, where T is the number of segments, F is the number of frames in each segment, and D is the number of features associated to each frame. Each segment can be seen as $\mathbf{S} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_F] \in \mathbb{R}^{F \times D}$ where $\boldsymbol{x} \in \mathbb{R}^D$ is the feature corresponding to each frame. MIL computes the likelihood of each frame being anomalous, selects the most anomalous ones based on it, and maximizes the difference in the predicted likelihood between the normal frames and the ones selected as the most anomalous.

In this chapter, we propose to leverage the CLIP model [90] to address VAR and show that:

i) the alignment between the visual and textual modalities in the CLIP feature space can be used as an effective likelihood estimator for anomalies; ii) such estimator, not only can detect anomalous occurrences, but also their types; iii) such estimator is effective only when adopting our proposed CLIP space re-centering transformation (see Fig. 2.2 (a)). Our method is composed of two models as shown in Fig. 2.2 (b): a *Selector model* and a *Temporal model*. The Selector model $\mathcal{S}$ produces the likelihood that each frame belongs to an anomalous class $\mathcal{S}(\boldsymbol{x}) \in \mathbb{R}^C$, where C is the number of anomalous classes. We exploit the vision-text alignment in the CLIP feature space and the CoOp prompt learning approach [152] to estimate this likelihood. The Temporal model $\mathcal{T}$ assigns a binary likelihood to each frame of a video, indicating whether the frame is anomalous or normal. Unlike $\mathcal{S}$, $\mathcal{T}$ exploits temporal information to improve predictions, and we implement it with a Transformer network [40]. The predictions from $\mathcal{S}$ and $\mathcal{T}$ are then aggregated to produce a distribution indicating the probability of a frame being normal or abnormal, and which abnormal class it belongs to. We train our model using a combination of MIL and regularization losses. Importantly, as $\mathcal{T}$ is randomly initialized, the likelihood scores are less reliable, thus we always use the likelihoods produced by $\mathcal{S}$ to perform segment selection in MIL.

We describe the proposed Selector model and Temporal model in detail in Sec. 2.3.1 and Sec. 2.3.2, respectively. In Sec. 2.3.3, we show how we aggregate the predictions of both models for estimating the final probability distribution. Finally, we describe the training and inference in Sec. 2.3.4.

2.3.1 Selector model

It is crucial for VAD and VAR to reliably distinguish anomalous and normal frames in anomalous videos, given only video-level weak supervision. Motivated by the recent findings in applying VLMs to video action recognition tasks [119, 134], we propose a novel likelihood estimator, encapsulated by our Selector model, that combines the CLIP [90] feature space and the CoOp [152] prompt learning approach to learn a set of directions in this space that identify each type of anomaly and their likelihood.

Our main intuition (see Fig. 2.2 (a)) is that the CLIP feature space presents an underlying structure where the set of CLIP features extracted for each frame in the dataset forms a space that is clustered around a central point, which we call the normality prototype. Consequently, the difference between a feature and the normal prototype determines important characteristics: the magnitude of the distance reflects the likelihood of it being abnormal, while its direction indicates the type of anomaly. Such important characteristics would not be exploited by a naive application of the CLIP feature space to VAR (see Tab. 2.9). Unleashing the potential of this space in detecting anomalies thus requires a re-centering transformation, a main contribution of this work.

Following this intuition, we define the normal prototype $\boldsymbol{m}$ as the average feature extracted by the CLIP image encoder $\mathcal{E}_I$ on all N frames $\mathbf{I}$ contained in videos labeled as normal in the dataset:

$$\boldsymbol{m} = \frac{1}{N} \sum_{j=1}^{N} \mathcal{E}_I(\mathbf{I}_j). \tag{2.1}$$

For each frame $\mathbf{I}$ in the dataset, we produce frame features $\boldsymbol{x}$ by subtracting the normality prototype from the CLIP encoded feature, *i.e.*, $\boldsymbol{x} = \mathcal{E}_I(\mathbf{I}) - \boldsymbol{m}$.

We then exploit the visual-text aligned CLIP feature space and learn the textual prompt embeddings whose directions are used to indicate the anomalous classes. In particular, we employ the prompt learning CoOp method [152], which we find ideal to find such directions as empirically demonstrated by our experiments (see Sec. 2.4.3).

Given a class c and the textual description of the corresponding label $\boldsymbol{t}^c$ expressed as a sequence of token embeddings, we consider a sequence of learnable context vectors $\boldsymbol{t}^{\text{ctx}}$ and derive the corresponding direction for the class $\boldsymbol{d}_c \in \mathbb{R}^D$ as:

$$\boldsymbol{d}_c = \mathcal{E}_T([\boldsymbol{t}^{\text{ctx}}, \boldsymbol{t}^c]) - \boldsymbol{m}, \tag{2.2}$$

where $\mathcal{E}_T$ indicates the CLIP text encoder. The use of the textual description acts as a

prior for the learned direction to match the corresponding type of anomaly, while the context vectors are jointly optimized during training as part of the parameters of $\mathcal{S}$ in order to enable the refinement of the direction. A different direction is learned for each class.

The learned directions serve as the base for our Selector model $\mathcal{S}$. As shown in Fig. 2.2(b), the magnitude of the projection of frame feature $\boldsymbol{x}$ on direction $\boldsymbol{d}_c$ indicates the likelihood of the anomalous class c:

$$\mathcal{S}(\boldsymbol{x}) = [\mathcal{P}(\boldsymbol{x}, \boldsymbol{d}_1), ..., \mathcal{P}(\boldsymbol{x}, \boldsymbol{d}_C)] \in \mathbb{R}^C, \tag{2.3}$$

where $\mathcal{P}$ indicates our projection operation. However, simply projecting the feature vector on the direction would make the magnitude of the projection susceptible to scale, where anomalous features of one class can potentially have a different magnitude from features of another anomalous class. To mitigate this issue, we perform a batch normalization [44] after the projection, which produces a distribution of projected features with zero mean and unitary variance:

$$\mathcal{P}(\boldsymbol{x}, \boldsymbol{d}_i) = \mathrm{BN}\left(\frac{\boldsymbol{x} \cdot \boldsymbol{d}_i}{||\boldsymbol{d}_i||}\right), \tag{2.4}$$

where BN indicates batch normalization without an affine transformation. As such, we expect within a batch the dominant normal features to be close to the origin and the abnormal features to be at the right side tail of the distribution.

The definition of likelihood can be extended to segments by summing the likelihoods of each frame:

$$\mathcal{S}(\mathbf{S}) = \sum_{i=1}^{F} \mathcal{S}(\boldsymbol{x}_i) \in \mathbb{R}^C \tag{2.5}$$

2.3.2 Temporal model

The Selector model only learns an initial *time-independent* separation between anomalous and normal frames as the CLIP model operates at the image frame level. However, the temporal information is an important piece of information for VAR that we can exploit. We thus propose the Temporal model $\mathcal{T}$ to model the relationships among frames in both short-term and long-term, to enrich the visual features and to produce the predictions that indicate the likelihood of whether a frame is anomalous:

$$\mathcal{T}(\mathbf{V}) \in \mathbb{R}^{T \times F}. \tag{2.6}$$

We use a Transformer architecture to capture the short-term temporal dependencies between frames in a segment and the long-term temporal dependencies between all segments in a video, motivated by their success in relevant sequence modeling tasks [115]. As all the video segments of $\mathbf{V}$ are received as the input, the large number of segments T and frames F increases the computational requirements for self-attention. To reduce this cost, we implement $\mathcal{T}$ as an Axial Transformer [40] that computes attention separately for the two axes corresponding to the segments and the features in each segment. As suggested by experiments in Sec. 2.4.3, Axial Transformer is also less prone to over-fitting, a likely case in VAR, as compared to a standard Transformer. We terminate the model with a sigmoid activation so that the output likelihood can also be interpreted as a probability.

2.3.3 Predictions aggregation

We combine the predictions from $\mathcal{S}$ and $\mathcal{T}$ to obtain the final output: the probabilities indicating whether a frame is normal or anomalous ($p_N(\boldsymbol{x})$ and $p_A(\boldsymbol{x})$) and the probability that a frame presents an anomaly of a certain class ($p_{A,c}(\boldsymbol{x})$).

Given an input frame feature $\boldsymbol{x}$, we define its probability of being anomalous $p_A(\boldsymbol{x})$ as its corresponding output from the Temporal model $\mathcal{T}$. The probability of the frame being normal is $p_N(\boldsymbol{x}) = 1 - p_A(\boldsymbol{x})$. To obtain the probability distribution of the frame to present an anomaly of a specific class $p_{A,c}(\boldsymbol{x})$, we employ the predictions of the Selector model that can be seen as the conditional distribution over the anomalous classes $p_{c|A}(\boldsymbol{x}) = \texttt{softmax}(\mathcal{S}(\boldsymbol{x}))$. From the definition of conditional probability it follows that $p_{A,c}(\boldsymbol{x}) = p_A(\boldsymbol{x}) * p_{c|A}(\boldsymbol{x})$.

2.3.4 Training

We train the model following the MIL framework. Specifically, MIL considers a batch with an equal number of normal and anomalous videos, uses the predicted likelihoods to identify the top-K most abnormal segments in anomalous videos, and imposes separation from the other, normal ones [104]. Due to the higher capacity of $\mathcal{T}$ with respect to $\mathcal{S}$ and its initial random initialization, $\mathcal{T}$ can not directly perform this selection since the predicted likelihoods would be excessively noisy. Instead, we use the likelihood predictions from $\mathcal{S}$ to perform MIL segment selection.

Our framework is trained end-to-end using losses on anomalous videos, losses on normal videos, and regularization losses, which we describe in the following.

Given an anomalous video $\mathbf{V}_A$ of class c, we define the set of top-K most anomalous segments $\mathcal{V}_A^+ = \{\mathbf{S}_{A1}^+, ..., \mathbf{S}_{AK}^+\}$ and, symmetrically, of bottom-K least anomalous segments $\mathcal{V}_A^- = \{\mathbf{S}_{A1}^-, ..., \mathbf{S}_{AK}^-\}$ according to the likelihood assigned by the frame-level model $\mathcal{S}$ on the direction corresponding to class c. We consider all frames in $\mathcal{V}_A^+$ and maximize the likelihood of the corresponding class being predicted by $\mathcal{S}$ by minimizing the loss $\mathcal{L}_A^{\text{DIR}}$:

$$\mathcal{L}_A^{\text{DIR}} = -\frac{\sum_{i=1}^{K} \mathcal{S}(\mathbf{S}_{Ai}^+)_c}{KF}, \tag{2.7}$$

where the likelihood tensor is indexed using the class c. To provide gradients to the Temporal model, we also maximize $p_{A,c}(\boldsymbol{x})$ for each frame contained in the segments using cross entropy:

$$\mathcal{L}_{A^+} = -\frac{\sum_{i=1}^{K}\sum_{j=1}^{F} \log(p_{A,c}(\mathbf{S}_{Ai,j}^+))}{KF}. \tag{2.8}$$

Distinguishing normal and anomalous frames in anomalous videos is a challenging problem in VAR due to the appearance similarity between frames of the same video. To foster a better separation between these frames, we additionally consider $\mathcal{V}_A^-$ and maximize $p_N(\boldsymbol{x})$ for each frame in the segments using cross entropy:

$$\mathcal{L}_{A^-} = -\frac{\sum_{i=1}^{K}\sum_{j=1}^{F} \log(p_N(\mathbf{S}_{Ai,j}^-))}{KF}, \tag{2.9}$$

To leverage the information in normal videos, for each segment $\mathbf{S}_i$ in normal video $\mathbf{V}_N$, we minimize the likelihood predicted by the Selector model:

$$\mathcal{L}_N^{\text{DIR}} = \frac{\sum_{i=1}^{T}\sum_{c=1}^{C} \mathcal{S}(\mathbf{S}_i)_c}{TFC}. \tag{2.10}$$

Following the VAD literature [28, 104, 110] we also require the model to maximize the probability of each frame in its top-K most abnormal segments $\mathcal{V}_N^+ = \{\mathbf{S}_{N1}^+, ..., \mathbf{S}_{NK}^+\}$ to be normal :

$$\mathcal{L}_{N^+} = -\frac{\sum_{i=1}^{K}\sum_{j=1}^{F} \log(p_N(\mathbf{S}_{Ni,j}^+))}{KF}. \tag{2.11}$$

We regularize training with two additional losses [104] on all frames of anomalous videos only. One is a sparsity loss on the predicted scores and encourages the minimal amount of frames to be predicted as abnormal:

$$\mathcal{L}_{\text{spa}} = \frac{\sum_{i=1}^{T}\sum_{j=1}^{F} p_A(\mathbf{V}_{i,j})}{TF} \tag{2.12}$$

The other is a smoothness term that regularises the predictions along the temporal dimension:

$$\mathcal{L}_{\text{smo}} = \sum_{i=2}^{TF} (p_A(\mathbf{V}_i) - p_A(\mathbf{V}_{i-1})), \tag{2.13}$$

where indexing is performed on the flattened sequence of frames in the video.

We jointly train the Selector and Temporal models using as final training objective:

$$\mathcal{L} = \mathcal{L}_A^{\text{DIR}} + \mathcal{L}_{A^+} + \mathcal{L}_{A^-} + \mathcal{L}_N^{\text{DIR}} + \mathcal{L}_{N^+} + \lambda_1 \mathcal{L}_{\text{spa}} + \lambda_2 \mathcal{L}_{\text{smo}}. \tag{2.14}$$

2.4 Experiments

In this section, we validate our method against a range of baselines taken from state-of-the-art VAD and action recognition methods that we adapt to the VAR task. After introducing the metrics for the VAR task, we perform evaluation on three datasets and perform a comparison in both the VAD and VAR tasks. An extensive ablation study is performed to justify our main design choices. Sec. 2.4.1 describes our experiment setup in terms of datasets and evaluation protocols. We then present and discuss the results in comparison against state-of-the-art methods in Sec. 2.4.2 and the ablation study in Sec. 2.4.3.

2.4.1 Experiment setup

Datasets. We perform our study using three widely-used VAD datasets, *i.e.*, ShanghaiTech [65], UCF-Crime [104], and XD-Violence [130]. *ShanghaiTech* consists of 437 videos, recorded from multiple surveillance cameras in a university campus. A total of 130 abnormal events of 17 anomaly classes are captured in 13 different scenes. We adopt the dataset in the configuration of [151], which adapts it to the weakly-supervised setting by organizing it into 238 training videos and 199 testing videos. *UCF-Crime* is a large-scale dataset of real-world surveillance videos, containing 1900 long untrimmed videos that cover 13 real-world anomalies with significant impacts on public safety. The training set consists of 800 normal and 810 anomalous videos, and the testing set includes the remaining 150 normal and 140 anomalous videos. *XD-Violence* is a large-scale violence detection dataset comprising 4754 untrimmed videos with audio signals and weak labels, divided into a training set of 3954 videos and a test set of 800 videos. With a total duration of 217 hours, the dataset covers various scenarios and captures 6 categories of anomalies. Notably, each violent video may have multiple labels,

ranging from 1 to 3. To accommodate our training setup, where only one anomaly type per video is considered, we select the subset of 4463 videos containing at most one anomaly.

Performance Metrics. We perform evaluation in terms of both VAD and VAR. Following previous works, we measure the performance regarding VAD using the area under the curve (AUC) of the frame-level receiver operating characteristics (ROC) as it is agnostic to thresholding for the detection task. A larger frame-level AUC means a better performance in classifying between normal and anomalous events. To measure the VAR performance, we extend the AUC metric to the multi-classification scenario. For each anomalous class, we measure the AUC by considering the anomalous frames of the class as positive and all other frames as negative. Successively, the mean AUC (mAUC) is computed over all the anomalous classes. Similarly, for the XD-Violence dataset, we follow the established evaluation protocol [130] and present VAD results using the average precision (AP) of the precision-recall curve (PRC), while for VAR results we report the mean AP (mAP), which is calculated by averaging the binary AP values across all anomalous classes.

Implementation details. At training time, each video is divided into T non-overlapping blocks. From each block, a random start index is sampled from which segments of F consecutive frames are considered. If the raw video has a length smaller than $T \times F$, we adopt loop padding and repeat the video from the start until the minimum length of $T \times F$ is reached. Each mini-batch of size B used for training is composed of $B/2$ normal clips and $B/2$ anomalous clips. This is a simple but effective way to balance the mini-batch formation, which otherwise will contain mainly normal clips. At inference, to handle videos covering arbitrary temporal windows, we first divide each video $\mathbf{V}$ into T non-overlapping blocks, where each block contains frames whose number is a multiple of F, *i.e.*, $J \times F$, where J depends on the length of $\mathbf{V}$[1]. We process $\mathbf{V}$ with J inferences to classify all frames in the video. At each j^{th} inference, we extract the j^{th} consecutive F frames from each block, forming segments with a total of $T \times F$ that span the whole video. We then feed the segments into our approach so that our Temporal model can reason the long-term temporal relationships among segments.

For a fair comparison with previous works in VAD [110, 128, 55], we use $K = 3$ for the MIL selection of the top-K and bottom-K abnormal segments, $T = 32$ number of segments, $F = 16$ frames per segment, and $B = 64$ batch size.

[1]We perform loop padding to ensure that each video is of length $J \times T \times F$

Table 2.1: Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAD and VAR on ShanghaiTech.

Supervision	Method	Features	VAD	VAR	AUC(%)	mAUC(%)
One-class	MNAD [85]		✓		70.50	
	MPN [73]		✓		73.80	
	HF^2VAD [68]		✓		76.20	
	[142]	ResNext	✓		79.62	
Unsupervised	[142]	ResNext	✓		78.93	
Zero-shot	CLIP [90]	ViT-B/16		✓	49.17	51.02
Weakly-supervised	[104]	C3D-RGB	✓		86.30	
	IBL [150]	C3D-RGB	✓		82.50	
	[142]	ResNext	✓		86.21	
	GCN [151]	TSN-RGB	✓		84.44	
	MIST [28]	I3D-RGB	✓		94.83	
	[130]	I3D-RGB	✓			
	CLAWS [144]	C3D-RGB	✓		89.67	
	RTFM [110]	I3D-RGB	✓		97.21	81.60
	[129]	I3D-RGB	✓		97.48	
	MSL [57]	I3D-RGB	✓		96.08	
	MSL [57]	VideoSwin-RGB	✓		97.32	
	S3R [128]	I3D-RGB	✓		97.48	87.88
	MGFN [15]	I3D-RGB	✓			
	MGFN [15]	VideoSwin-RGB	✓			
	SSRL [55]	I3D-RGB	✓		97.98	93.61
	ActionCLIP [119]	ViT-B/16		✓	96.36	75.63
	AnomalyCLIP (ours)	ViT-B/16	✓	✓	**98.07**	**96.46**

2.4.2 Evaluation against baselines

Regarding VAD, we compare AnomalyCLIP against state-of-the-art methods with different supervision setups, including one-class [85, 68, 73], unsupervised [142] and weakly-supervised [55, 110, 128]. As none of the above-mentioned methods address the VAR task, we produce baselines by re-purposing some best-performing VAD methods including RTFM [110], S3R [128] and SSRL [55]

- *Multi-classification with RTFM [110], S3R [128] and SSRL [55] (weakly-supervised).* We keep the original pretrained model frozen and add a multi-class classification head that we train to predict the class using a cross-entropy objective on the top-K most anomalous segments selected as in the original method. These baselines are weakly-supervised.

Table 2.2: Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAD and VAR on UCF-Crime.

Supervision	Method	Features	VAD	VAR	AUC(%)	mAUC(%)
One-class	SVM Baseline [104]		✓		50.00	
	SSV [101]		✓		58.50	
	BODS [118]	I3D-RGB	✓		68.26	
	GODS [118]	I3D-RGB	✓		70.46	
	[142]	ResNext	✓		74.20	
Un-supervised	[142]	ResNext	✓		71.04	
Zero-shot	CLIP [90]	ViT-B/16		✓	58.63	74.28
Weakly-supervised	[104]	C3D-RGB	✓		75.41	
	[104]	I3D-RGB	✓		77.92	
	IBL [150]	C3D-RGB	✓		78.66	
	[142]	ResNext	✓		79.84	
	GCN [151]	TSN-RGB	✓		82.12	
	MIST [28]	I3D-RGB	✓		82.30	
	[130]	I3D-RGB	✓		82.44	
	CLAWS [144]	C3D-RGB	✓		83.03	
	RTFM [110]	VideoSwin-RGB	✓		83.31	
	RTFM [110]	I3D-RGB	✓		84.03	84.86
	[129]	I3D-RGB	✓		84.89	
	MSL [57]	I3D-RGB	✓		85.30	
	MSL [57]	VideoSwin-RGB	✓		85.62	
	S3R [128]	I3D-RGB	✓		85.99	86.55
	MGFN [15]	VideoSwin-RGB	✓		86.67	
	MGFN [15]	I3D-RGB	✓		86.98	
	SSRL [55]	I3D-RGB	✓		**87.43**	88.88
	ActionCLIP [119]	ViT-B/16		✓	82.30	87.72
	AnomalyCLIP (ours)	ViT-B/16	✓	✓	86.36	**90.66**

- *CLIP [90] (zero-shot).* We achieve the classification by soft-maxing the cosine similarities of the input frame feature $\boldsymbol{x}$ with vectors corresponding to the embedding of the textual prompt *"a video from a CCTV camera of a {class}"* using the pre-trained CLIP model.
- *ActionCLIP [119] (weakly-supervised).* We retrain ActionCLIP [119] on our datasets by propagating the video-level anomaly labels to each frame of the corresponding video.

Tab. 2.1 presents the results on ShanghaiTech [65]. Although ShanghaiTech is a rather saturated dataset for VAD due to its simplicity in scenarios, AnomalyCLIP scores the state-of-the-art results on both VAD and VAR, with +0.09% and +2.85% in terms of

Table 2.3: Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAD and VAR on XD-Violence.

Supervision	Method	Features	VAD	VAR	AP(%)	mAP(%)
Zero-shot	CLIP [90]	ViT-B/16		✓	27.21	21.32
Weakly-supervised	[130]	C3D-RGB	✓		67.19	
	[130]	I3D-RGB	✓		73.20	
	MSL [57]	C3D-RGB	✓		75.53	
	[129]	I3D-RGB	✓		75.90	
	RTFM [110]	I3D-RGB	✓		77.81	43.04
	MSL [57]	I3D-RGB	✓		78.28	
	MSL [57]	VideoSwin-RGB	✓		78.58	
	S3R [128]	I3D-RGB	✓		**80.26**	36.06
	MGFN [15]	I3D-RGB	✓		79.19	
	MGFN [15]	VideoSwin-RGB	✓		80.11	
	ActionCLIP [119]	ViT-B/16		✓	61.01	40.24
	AnomalyCLIP (ours)	ViT-B/16	✓	✓	78.51	**49.41**

Table 2.4: Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAR on UCF-Crime. The table highlights the top performers, with cells highlighted in red representing first place, cells in orange representing second place, and cells in yellow representing third place.

Method	Class													mAUC
	Abuse	Arrest	Arson	Assault	Burglary	Explosion	Fighting	RoadAcc.	Robbery	Shooting	Shoplifting	Stealing	Vandalism	
RTFM [110]	79.99	62.57	90.53	82.27	85.53	92.76	85.21	90.31	81.17	82.82	92.56	90.23	87.20	84.86
S3R [128]	86.38	68.45	92.19	93.55	86.91	93.55	81.69	85.03	82.07	85.32	91.64	94.59	83.82	86.55
SSRL [55]	95.33	79.26	93.27	91.74	89.06	92.25	87.36	80.24	87.75	84.50	92.31	94.22	88.17	88.88
CLIP zero-shot [90]	57.37	80.65	93.72	80.83	74.34	90.31	83.54	87.46	70.22	63.99	71.21	45.49	66.45	74.28
ActionCLIP [119]	91.88	90.47	89.21	86.87	81.31	94.08	83.23	94.34	82.82	70.53	91.60	94.06	89.89	87.72
AnomalyCLIP	75.03	94.56	96.66	94.80	90.08	94.79	88.76	93.30	86.85	87.45	89.47	97.00	89.78	90.66

Table 2.5: Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAR on ShanghaiTech. The table highlights the top performers, with cells highlighted in red representing first place, cells in orange representing second place, and cells in yellow representing third place.

Method	Class														mAUC
	Car	Chasing	Circuit	Fall	Fighting	Jumping	Monocycle	Push	Robbery	Running	Skateboard	Stoop	ThrowingObj.	Vaudeville	Vehicle	
RTFM [110]	99.70	95.41	99.83	70.19	97.36	89.14	37.99	35.28	67.01	90.59	96.81	64.11	97.93	91.75	90.85	81.60
S3R [128]	98.71	96.80	99.97	85.63	95.93	69.33	96.82	54.76	61.19	94.43	96.92	75.46	97.63	97.78	96.84	87.88
SSRL [55]	99.35	97.31	99.95	91.24	96.88	93.07	89.74	90.62	91.81	94.47	97.73	71.81	98.44	96.32	95.49	93.61
CLIP zero-shot [90]	61.65	77.88	5.95	61.73	79.37	23.68	77.78	63.36	37.71	54.39	76.15	8.47	44.10	65.97	27.08	51.02
ActionCLIP [119]	98.50	93.86	98.59	16.38	97.45	89.63	98.05	8.14	67.36	78.25	97.10	0.76	97.70	98.65	93.97	75.63
AnomalyCLIP	98.08	96.66	97.97	96.69	98.03	95.48	86.89	97.99	95.00	97.95	97.29	98.62	96.50	96.97	96.79	96.46

AUC ROC and mAUC ROC, respectively. ActionCLIP [119] performs poorly in terms of mAUC, which we attribute to the low proportion of abnormal events in ShanghaiTech that makes the MIL selection strategy of particular importance to avoid incorrect supervisory signals on normal frames of abnormal videos. In contrast, our proposal has

Table 2.6: Results of the state-of-the-art methods and our AnomalyCLIP in terms of VAR on XD-Violence. The table highlights the top performers, with cells highlighted in red representing first place, cells in orange representing second place, and cells in yellow representing third place.

Method	**Class**						**mAP**
	Abuse	CarAccident	Explosion	Fighting	Riot	Shooting	
RTFM [110]	9.25	25.36	53.53	61.73	90.38	18.01	43.04
S3R [128]	2.63	23.82	45.29	49.88	90.41	4.34	36.06
CLIP zero-shot [90]	0.32	12.21	22.26	25.25	66.60	1.26	21.32
ActionCLIP [119]	2.73	25.15	55.28	58.09	87.31	12.87	40.24
AnomalyCLIP	6.10	31.31	68.75	71.44	92.74	26.13	49.41

a better recognition of the positive instances of abnormal videos, thus achieving better performance even when anomalies are rare. AnomalyCLIP achieves a large improvement of +45.44% in terms of mAUC against zero-shot CLIP, demonstrating that a naive application of a VAR pipeline in the CLIP space does not yield satisfactory results. A revision of this space, implemented as our proposed transformation, is necessary to use it effectively.

Tab. 2.2 reports the results on UCF-Crime [104]. Our method exhibits the best discrimination of the anomalous classes, achieving the highest mAUC ROC among baselines. Similar to ShanghaiTech, it also achieves an improvement in terms of mAUC against zero-shot CLIP, verifying the importance of our proposed adaptation of the CLIP space. Compared to ActionCLIP [119], our AnomalyCLIP obtains +2.94% in terms of mAUC, highlighting the need for a MIL framework to mitigate mis-assignment of anomalous class labels to normal frames of anomalous videos. It is also worth noting that the higher mAUC obtained by ActionCLIP does not result in a competitive AUC ROC on VAD, which implies a worse separation between normal and abnormal frames. When compared to the best performing method, SSRL [55] on VAD, our method obtains an improvement of +1.78% in terms of mAUC on VAR, while being slightly worse with −1.07% in terms of AUC ROC on VAD.

Tab. 2.3 shows the results on XD-Violence [130]. AnomalyCLIP outperforms other state-of-the-art methods on VAR, achieving the highest mAP. Compared to the VAD baselines' models, AnomalyCLIP outperforms RTFM [110] and demonstrates performance close to S3R [128].

Tabs. 2.4 to 2.6 display the multi-class AUC and AP for each abnormal class. The proposed method has a clear advantage when applied to the UCF-Crime and XD-

Violence datasets, which are generally considered to be complex benchmarks in anomaly detection. Our method achieves the best mAUC and mAP on average, while it is less advantageous when dealing with anomalies that exhibit slight deviations from normal patterns, such as Shoplifting in UCF-Crime. The advantage of our proposed method is less noticeable when applied to the ShanghaiTech dataset, which captures simple scenes where most methods have achieved a saturated performance.

Fig. 2.3 presents the qualitative results of our proposed AnomalyCLIP in detecting and recognizing anomalies within a set of UCF-Crime, ShanghaiTech, XD-Violence test videos. The model is capable of predicting both the presence of anomalies in test videos and the category of the anomalous event. In video *Normal_ Video_ 246* from UCF-Crime (Row 2, Column 2), it can be seen how some frames have a higher-than-expected probability of being abnormal. It is interesting to note how in the video *RoadAccidents133* from UCF-Crime (Row 1, Column 2), the anomaly score remains high even in the aftermath of the accident. It is also interesting to note that for Normal videos, AnomalyCLIP is able to obtain a relatively low anomaly probability all over the frames, meaning our model has learned a robust normal representation among Normal videos. For a more intuitive understanding of the results presented in the chapter, we invite readers to access the website `https://lucazanella.github.io/AnomalyCLIP`, where easily accessible qualitative results are available.

2.4.3 Ablation

In this section, we perform ablations of our method to validate our main design choices with UCF-Crime: the way in which we represent and learn directions, the transformations applied to the CLIP space and the employed way for estimating the likelihood of anomaly, the choice of architecture for the Temporal model, training objectives, and the impact of using features extracted from different backbones. Finally, we discuss how the notion of normality generalizes when normal behavior is highly diverse.

Representation and Learning of the Directions. In the ablation shown in Tab. 2.7, we evaluate the choice of the CoOp [152] framework to learn directions in the CLIP space. When CoOp is removed, we directly learn the directions from randomly initialized points in the CLIP space (Row 1) or make use of fixed engineered prompts of the form *"a video from a CCTV camera of a {class}"* (Row 2). Both choices result in degradation of the results, indicating that text-guided initialization of the directions and directions finetuning are both necessary. Furthermore, we show that unfreezing the last projection of the text encoder (Row 4) enables a greater freedom in finetuning the discovered

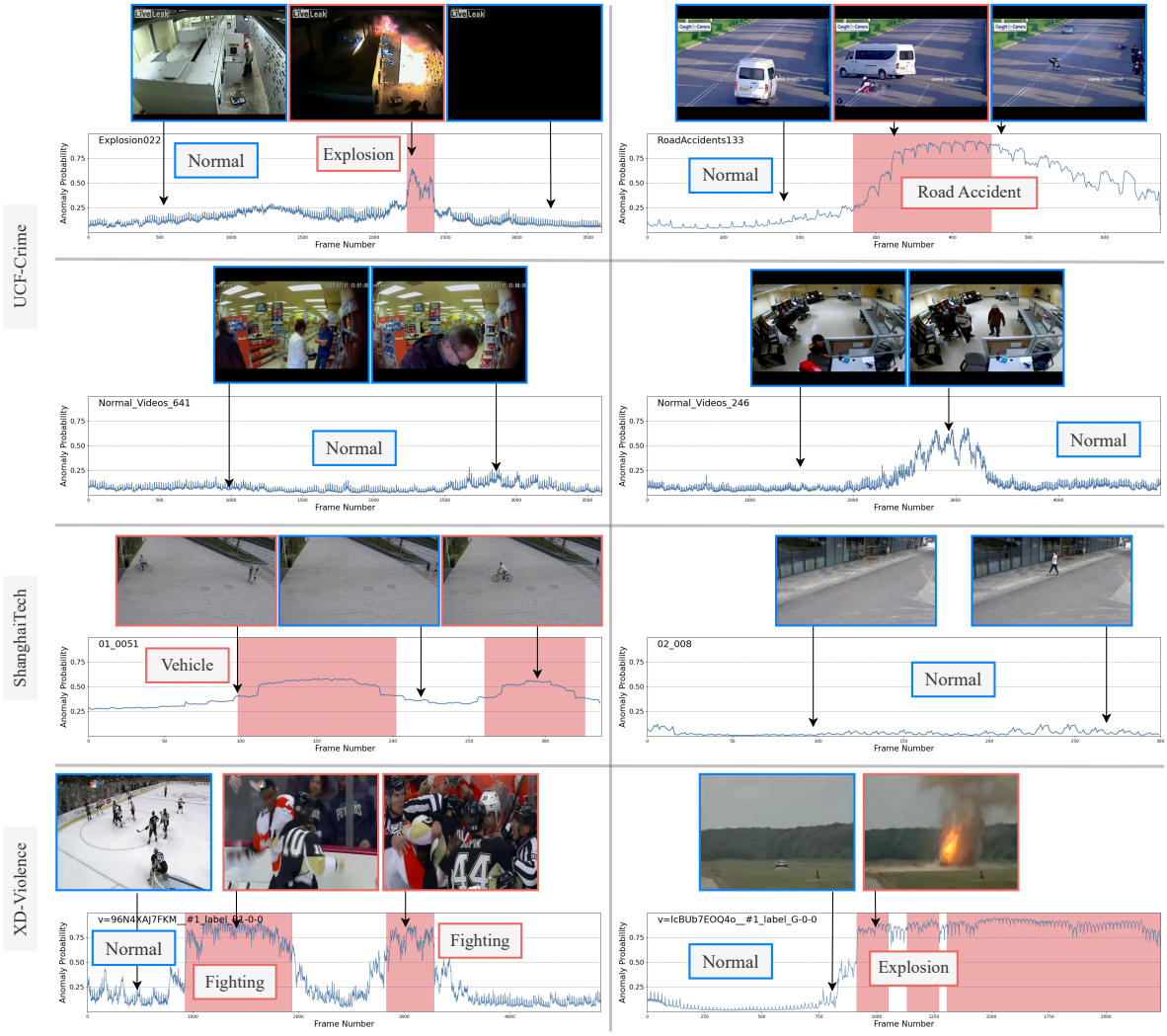


Figure 2.3: Qualitative results for VAR on four test videos from UCF-Crime (the top two rows), two test videos from ShanghaiTech (the third row), and two test videos from XD-Violence (the bottom row). For each video, we show at the bottom the predicted probability of each frame being anomalous by our model over the number of frames. We showcase some key frames to reflect the relevance between the predicted anomaly probability and the visual content. The red shaded areas denote the temporal ground-truth of anomalies. We also indicate the predicted anomalous class for detected abnormal frames in the red boxes, while videos without detected anomalies are indicated with blue boxes as Normal.

Table 2.7: Ablation on representation and learning of the directions of abnormality. 'Finetuning' indicates that the last projection layer is fine-tuned. The final configuration of our model is represented by the row highlighted in grey in the table.

Text encoder	Directions	AUC	mAUC
No	Direct Optimization	84.98	69.86
Frozen	Engineered Prompts	84.66	81.35
Frozen	CoOp	85.88	87.39
Finetuning	CoOp	**86.36**	**90.66**

Table 2.8: Comparisons of different architectural choices for the CoOp module. 'Shared' means that all the classes share a unified context, otherwise each class has a specific context. The final configuration of our model is represented by the row highlighted in grey in the table.

Context vectors	Shared	AUC	mAUC
4		86.16	**91.05**
8		**86.36**	90.66
16		85.82	90.65
8	✓	85.97	90.01

directions, yielding the best results.

In the ablation shown in Tab. 2.8, we evaluate the architectural choices on the CoOp module to learn directions in the CLIP space. Specifically, we experimented by varying the number of context vectors $\boldsymbol{t}^{\text{ctx}}$ used from 4 to 8 to 16, and using shared or class-specific context vectors. Although using 4 context vectors results in a slightly higher mAUC score, we eventually opted to use 8 context vectors because they produce a higher AUC score. Results (Rows 2 and 4) show that learning a specific set of context vectors for each class is more tailored to fine-grained categories, rather than relying on more generic shared context vectors for all classes.

Likelihood Estimation and CLIP Latent Space Transformation. The way in which the extracted CLIP features are transformed and the chosen likelihood estimation method play a crucial role in the quality of segment selection. We evaluate several choices in this procedure in Tab. 2.9. Directly using the CLIP space and cosine similarities with the learned directions as likelihood estimators (Row 1) produces the worst VAR results, indicating that the use of the normality prototype $\boldsymbol{m}$ is of high importance in the context of anomaly detection. Second, Row 2 shows that MIL segment selection as a function of the feature magnitude without accounting for the direction is not as effective, given that the large magnitude could be attributed to irrelevant factors.

Table 2.9: Ablation of different likelihood estimation methods, feature space transformations, and MIL selection. 'Features' indicates the transformation applied to CLIP features. The final configuration of our model is represented by the row highlighted in grey in the table.

Likelihood	Features	MIL Selection	AUC	mAUC
cosine sim.	CLIP	cosine sim.	85.59	83.69
$\mathcal{S}$	CLIP - $\boldsymbol{m}$	feature magnitude	84.92	89.82
$\mathcal{S}$	CLIP - $\boldsymbol{m}$	$\mathcal{S}$	**86.36**	**90.66**

Table 2.10: Comparisons of different architectural choices for the Temporal model. The final configuration of our model is represented by the row highlighted in grey in the table.

Temporal Model	Short-term	Long-term	AUC	mAUC
MLP			74.86	84.46
Transformer	✓		84.69	88.38
Transformer		✓	85.10	89.29
MTN		✓	82.71	87.65
Axial Transformer	✓	✓	**86.36**	**90.66**

Temporal Model Architecture. Capturing temporal information is an essential aspect of VAR since it provides insights into the behavior of objects and scenes over time. Tab. 2.10 shows results for different architectures of $\mathcal{T}$ *i.e.*, a 3-layer MLP, two Transformer Encoders [115], the multi-scale temporal network (MTN), designed in RTFM and used in S3R and SSRL, and the employed Axial Transformer. In particular, one transformer encoder (Row 2) performs self-attention on each independent 16-frame segment, solely modeling short-term dependencies. The other (Row 3) applies self-attention on segment embeddings, which are obtained by averaging 16-frame feature embeddings within each segment, thereby only modeling long-term dependencies. To ensure a fair comparison, both transformers are designed to have a capacity similar to that of the Axial Transformer. The reduced performance of the MLP baseline (Row 1) indicates the necessity of considering temporal information that is not readily available in the extracted CLIP features. The Axial transformer can capture temporal dependencies and outperform the compared architectures.

Tab. 2.11 shows the results for different values of the embedding size and the number of layers. In the final architecture, we use 1 layer and an embedding size of 256, for a total of 10.4 M trainable parameters.

Losses. Tab. 2.12 illustrates the contribution of the losses on the Selector model's

Table 2.11: Comparisons of different architectural choices for the Axial Transformer. The final configuration of our model is represented by the row highlighted in grey in the table.

Embedding size	Number of layers	AUC	mAUC
64	1	82.83	90.10
128	1	84.97	90.53
256	1	**86.36**	**90.66**
512	1	85.51	89.28
256	2	85.89	89.67
256	3	85.15	88.14

Table 2.12: Ablation of the losses on the Selector model. The final configuration of our model is represented by the row highlighted in grey in the table.

$\mathcal{L}_A^{\text{DIR}}$	$\mathcal{L}_N^{\text{DIR}}$	AUC	mAUC
		85.89	89.34
	✓	85.91	87.26
✓		**86.46**	**90.75**
✓	✓	86.36	90.66

outputs, where we progressively remove the losses from the full training objective. The loss on abnormal videos contributes to improved VAD and VAR results on UCF-Crime.

Tab. 2.13 similarly shows the contribution of the losses on the aggregated model's output, where we remove each from the complete training objective. We validate that each of the proposed losses promotes performance on both the VAD and VAR tasks.

The bottom-K least anomalous segments $\mathcal{V}_A^- = \{\mathbf{S}_{A1}^-, ..., \mathbf{S}_{AK}^-\}$ of anomalous videos proved to be beneficial for learning the Temporal Model. Inspired by this, we analyze the impact of incorporating this set of frames into the Selector Model loss by minimizing the loss:

$$\mathcal{L}_{A^-}^{\text{DIR}} = \frac{\sum_{i=1}^{K} \mathcal{S}(\mathbf{S}_{Ai}^-)_c}{KF}, \tag{2.15}$$

Moreover, instead of using all segments of normal videos in the Selector Model loss, we evaluate the impact of using only the top-K most abnormal segments $\mathcal{V}_N^+ = \{\mathbf{S}_{N1}^+, ..., \mathbf{S}_{NK}^+\}$ by minimizing the likelihood predicted by the Selector Model:

$$\mathcal{L}_{N^+}^{\text{DIR}} = \frac{\sum_{i=1}^{K} \mathcal{S}(\mathbf{S}_{Ni}^+)_c}{KF} \tag{2.16}$$

In Tab. 2.14, we present our findings, which indicate that modifying the loss function in

Table 2.13: Ablation of losses on the aggregated outputs. The final configuration of our model is represented by the row highlighted in grey in the table.

$\mathcal{L}_{A^+}$	$\mathcal{L}_{A^-}$	$\mathcal{L}_{N^+}$	**AUC**	**mAUC**
	✓	✓	45.23	69.57
✓		✓	84.50	**90.88**
✓	✓		80.96	86.10
✓	✓	✓	**86.36**	90.66

Table 2.14: Ablation on the variation of Selector model losses. The final configuration of our model is represented by the row highlighted in grey in the table.

$\mathcal{L}_A^{\text{DIR}}$	$\mathcal{L}_N^{\text{DIR}}$	$\mathcal{L}_{N^+}^{\text{DIR}}$	$\mathcal{L}_{A^-}^{\text{DIR}}$	**AUC**	**mAUC**
✓	✓		✓	**86.41**	88.29
✓		✓		86.17	90.53
✓	✓			86.36	**90.66**

Table 2.15: Comparisons of different features. The final configuration of our model is represented by the row highlighted in grey in the table.

Selector Model	**Temporal Model**	**AUC**	**mAUC**
I3D-RGB	I3D-RGB	65.05	84.24
ViT-B/16	I3D-RGB	78.11	88.26
ViT-B/16	$\mathcal{S}(\boldsymbol{x})$	84.44	86.78
ViT-B/16	ViT-B/16	**86.36**	**90.66**

either of two ways causes a degradation of performance. Specifically, our experiments (Row 1) demonstrate that using the bottom-K least abnormal segments is only effective when learning the Temporal Model. This is because if there is no clear separation between the bottom-K and top-K abnormal features, the Selector Model can lead to incorrectly selected bottom-K features that prevent it from learning good directions in the feature space. However, incorporating the bottom-K least abnormal segments becomes beneficial in the Temporal Model, which has a greater capacity. Furthermore, our experiments indicate that using all normal segments (Row 3) provides a more robust estimation of the direction from normal to anomalous compared to using only the top-K most abnormal segments (Row 2).

Feature Representation. The purpose of this ablation study is to determine the most suitable feature space for the proposed method *AnomalyCLIP*. To achieve this, we first investigate whether the space learned by the Selector Model can be applied to the Temporal Model. This C-dimensional space is formed by projecting each frame feature

$\boldsymbol{x}$ onto every $\boldsymbol{d}_c$ direction, where C represents the number of anomalous classes. Our results, presented in Tab. 2.15, indicate that using only this space leads to sub-optimal model performance (Row 3). This finding highlights the necessity of incorporating the information contained in the original feature space as well. We also experimented with using I3D features for both the Selector Model and the Temporal Model (Row 1), but the results demonstrate that the model using these features performs worse. We attribute this to the fact that I3D features are mapped to a region of space that is not aligned with the text features, unlike the features generated by CLIP's image encoder. For this reason, we also experimented with using I3D features for the Temporal Model and features from CLIP's image encoder for the Selector Model (Row 2). The result of this experiment further emphasizes that the latent space of CLIP is a more semantic space in which anomalous events of different classes are more separated, which in turn leads to superior discriminative ability in detecting and recognizing anomalous events.

Normality Generalization. A core challenge in multi-scene VAD is generalizing across diverse normal behaviors without triggering false alarms. We address this by defining normality through a global *normality prototype* ($\boldsymbol{m}$), calculated as the average feature vector over all normal frames in the training set. Re-centering the CLIP latent space around this mean ($\boldsymbol{x} = \mathcal{E}_I(I) - \boldsymbol{m}$) acts as a semantic filter, removing common features shared by normal scenes and isolating residual signals that capture deviations rather than scene-specific details. While using multiple prototypes could model multi-modal normality more precisely, this would introduce assignment ambiguity during training, as weak video-level labels do not specify which prototype should serve as the reference for a given frame. Instead, global re-centering maps diverse normal behaviors around the same origin, allowing a single set of learned directions to detect anomalies consistently across different environments. This results in robust generalization, as evidenced qualitatively by the consistently low anomaly scores maintained across the highly diverse normal scenes found in UCF-Crime and XD-Violence (Fig. 2.3).

2.5 Chapter Summary

In this chapter, we addressed the challenging task of Video Anomaly Recognition that extends the scope of Video Anomaly Detection by further requiring the classification of the anomalous activities. We proposed *AnomalyCLIP*, the first method that leverages VLMs in the context of VAR. Our work sheds light on the fact that a naive application of existing VLMs [90, 119] to VAR leads to unsatisfactory performance, and we demonstrated that several technical design choices are required to build a multi-modal deep

network for detecting and classifying abnormal behaviors. We also performed an extensive experimental evaluation showing that *AnomalyCLIP* achieves state-or-the-art VAR results on the benchmark ShanghaiTech [65], UCF-Crime [104], and XD-Violence [130] datasets.

While these results demonstrate that vision-language representations can be effectively adapted under weak supervision for video anomaly detection and recognition, *AnomalyCLIP* still relies on a task-specific training phase to capture normality and anomaly semantics. In practice, even weakly supervised training can limit deployment in scenarios where data collection is constrained or infeasible. This limitation motivates the investigation of whether video anomaly detection can be performed without any task-specific training, which is the focus of the next chapter.

Chapter 3

Harnessing Large Language Models for Training-free Video Anomaly Detection

The methodology introduced in Chapter 2 shows that frozen vision-language representations already encode rich semantic structure that can be exploited for video anomaly detection and recognition. However, approaches that rely on a dedicated training phase remain difficult to deploy in many real-world scenarios, particularly when data collection is constrained by privacy concerns or rapidly changing environments. Motivated by this limitation, this chapter investigates whether video anomaly detection can be performed without any task-specific training. We explore the zero-shot reasoning capabilities of Large Language Models (LLMs) to evaluate scene descriptions derived from Vision-Language Models (VLMs), resulting in a training-free formulation that does not require manual annotation or task-specific model adaptation.

3.1 Introduction

Video anomaly detection (VAD) aims to temporally localize events that deviate significantly from the normal pattern in a given video, *i.e.*, the anomalies. VAD is challenging as anomalies are often undefined and context-dependent, and they rarely occur in the real world. The literature [47] often casts VAD as an out-of-distribution detection problem and learns the normal distribution using training data with different levels of supervision (see Fig. 3.1), including fully-supervised (*i.e.*, frame-level supervision of both normal and abnormal videos) [6, 117], weakly-supervised (*i.e.*, video-level supervision of both normal and abnormal videos) [104, 129, 110, 55, 57, 48], one-class (*i.e.*, only normal videos) [85, 68, 73, 106, 137, 143], and unsupervised (*i.e.*, unlabeled

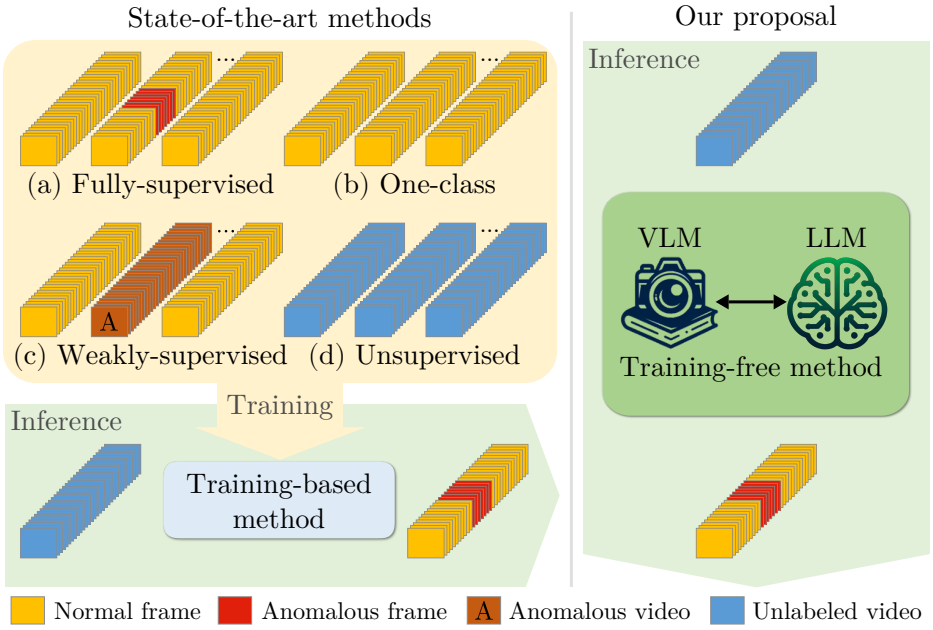


Figure 3.1: We introduce the first training-free method for video anomaly detection (VAD), diverging from state-of-the-art methods that are ALL training-based with different degrees of supervision. Our proposal, LAVAD, leverages modality-aligned vision-language models (VLMs) to query and enhance the anomaly scores generated by large language models (LLMs).

videos) [142, 112, 113]. While more supervision leads to better results, the cost of manual annotation is prohibitive. On the other hand, unsupervised methods assume abnormal videos to constitute a certain portion of the training data, a fragile assumption in practice without human intervention.

Crucially, every existing method necessitates a training procedure to establish an accurate VAD system, and this entails some limitations. One primary concern is generalization: a VAD model trained on a specific dataset tends to underperform in videos recorded in different settings (*e.g.*, *daylight* versus *night* scenes). Another aspect, particularly relevant to VAD, is the challenge of data collection, especially in certain application domains (*e.g.*, video surveillance) where privacy issues can hinder data acquisition. These considerations led us to explore a novel research question: *Can we develop a training-free VAD method?*

In this chapter, we aim to answer this challenging question. Developing a training-free VAD model is hard due to the lack of explicit visual priors on the target setting. However, such priors might be drawn using large foundation models, renowned for their generalization capability and wide knowledge encapsulation. Thus, we investigate the potential of combining existing vision-language models (VLMs) with large language

models (LLMs) in addressing training-free VAD. On top of our preliminary findings, we propose the first training-free **LA**nguage-based **VAD** method (**LAVAD**), that jointly leverages pre-trained VLMs and LLMs for VAD. LAVAD first exploits an off-the-shelf captioning model to generate a textual description for each video frame. We address potential noise in the captions by introducing a cleaning process based on the cross-modal similarity between captions and frames in the video. To capture the dynamics of the scene, we use an LLM to summarize captions within a temporal window. This summary is used to prompt the LLM to provide an anomaly score for each frame, which is further refined by aggregating the anomaly scores among frames with semantically similar temporal summaries. We evaluate LAVAD on two benchmark datasets: UCF-Crime [104] and XD-Violence [130], and empirically show that our training-free proposal outperforms unsupervised and one-class VAD methods on both datasets, demonstrating that it is possible to address VAD with *no training and no data collection*.

Contributions. In summary, our contributions are:

- We investigate, for the first time, the problem of training-free VAD, advocating its importance for the deployment of VAD systems in real settings where data collection may not be possible.
- We propose LAVAD, the first language-based method for training-free VAD using LLMs to detect anomalies exclusively from a scene description.
- We introduce novel techniques based on cross-modal similarity with pre-trained VLMs to mitigate noisy captions and refine the LLM-based anomaly scoring, effectively improving the VAD performance.
- Experiments show that, while using no task-specific supervision and no training, LAVAD achieves competitive results w.r.t. unsupervised and one-class VAD methods, opening new perspectives for future VAD research.

3.2 Related Work

Video Anomaly Detection. Existing literature on *training-based* VAD methods can be categorized into four groups, depending on the level of supervision: supervised, weakly-supervised, one-class classification, and unsupervised. *Supervised VAD* relies on frame-level labels to distinguish normal from abnormal frames [6, 117]. However, this scenario has received little attention due to its prohibitive annotation effort. *Weakly-supervised VAD* methods have access to video-level labels (the entire video is labeled as abnormal if at least one frame is abnormal, otherwise is regarded as normal) [104, 129, 110, 55, 57, 48]. Most of these methods utilize 3D convolutional neural networks for feature learning and

employ a multiple instance learning (MIL) loss for training. *One-class VAD* methods train only on normal videos, although manual verification is necessary to ensure the normality of the collected data. Several methods [85, 68, 73, 106, 137, 143] have been proposed, *e.g.*, considering generative models [137] or pseudo-supervised methods, where pseudo-anomalous instances are synthesized from normal training data [143]. Finally, *Unsupervised VAD* methods do not rely on predefined labels, leveraging both normal and abnormal videos with the assumption that most videos contain normal events [142, 112, 113, 108, 107]. Most methods in this category exploit generative models to capture normal data patterns in videos. In particular, generative cooperative learning (GCL) [142] employs alternating training: an autoencoder reconstructs input features, and pseudo-labels from reconstruction errors guide a discriminator. Tur *et al.* [112, 113] use a diffusion model to reconstruct the original data distribution from noisy features, calculating anomaly scores based on the reconstruction error between denoised and original samples. Other approaches [108, 107] train a regressor network from a set of pseudo-labels generated using OneClassSVM and iForest [62].

Instead, we completely sidestep the need for collecting data and training the model by exploiting existing large-scale foundation models to design a training-free pipeline for VAD.

LLMs for VAD. Recently, LLMs have been explored in detecting visual anomalies across diverse application domains. Kim *et al.* [50] propose an unsupervised method that mainly leverages VLMs for detecting anomalies, where ChatGPT is only utilized to produce textual descriptions that characterize normal and anomalous elements. However, the method involves human-in-the-loop to refine the LLM's outputs according to specific application contexts and requires further training to adapt the VLM. Other examples include exploiting LLMs for spatial anomaly detection in images addressing specific applications in robotics [25] or industry [33].

Differently, we leverage LLMs together with VLMs to address temporal anomaly detection on videos and propose the first *training-free* method for VAD, requiring no training and no data collection.

3.3 Training-Free VAD

In this section, we first formalize the VAD problem and the proposed training-free setting (Sec. 3.3.1). We then analyze the capabilities of LLMs in scoring anomalies in video frames (Sec. 3.3.2). Finally, we describe LAVAD, our proposed VAD method (Sec. 3.3.3).

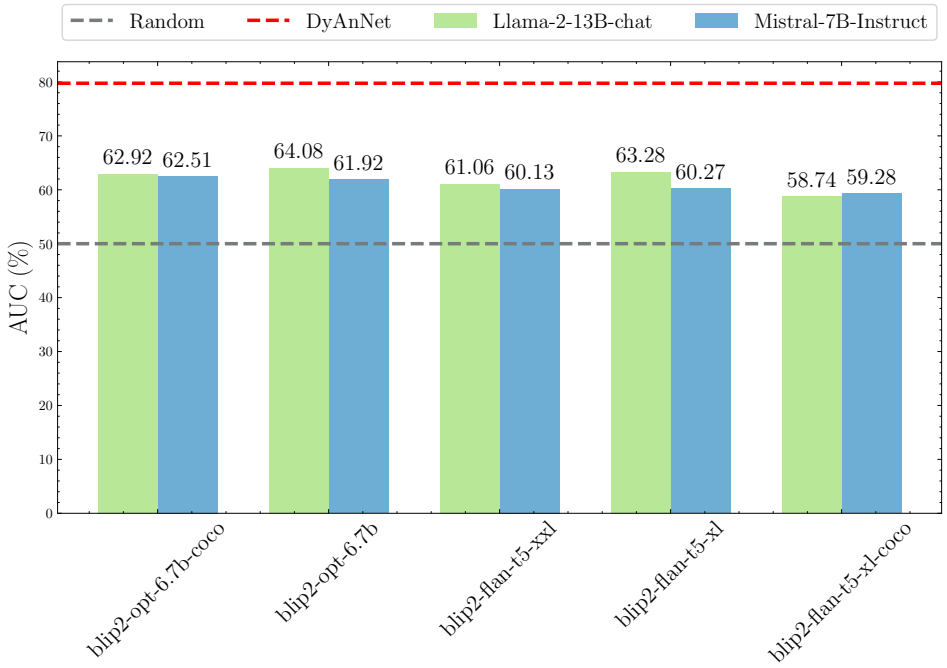


Figure 3.2: Bar plot of the VAD performance (AUC ROC) by querying LLMs with textual descriptions of video frames from various captioning models on the UCF-Crime test set. Different bars correspond to different variants of the captioning model BLIP-2 [56], while different colors indicate two different LLMs [111, 46]. For reference, we also plot the performance of the best-performing unsupervised method [108] in a red dashed line, and that of a random classifier in a gray dashed line.

3.3.1 Problem formulation

Given a test video $\mathbf{V} = [\mathbf{I}_1, \ldots, \mathbf{I}_M]$ of M frames, traditional VAD methods aim to learn a model f, which can classify each frame $\mathbf{I} \in \mathbf{V}$ as either normal (score 0) or anomalous (score 1), *i.e.* $f : \mathcal{I}^M \rightarrow [0, 1]^M$ with $\mathcal{I}$ being the image space. f is usually trained on a dataset $\mathcal{D}$ that consists of tuples in the form $(\mathbf{V},\ y)$. Depending on the supervision level, y can be either a binary vector with frame-level labels (fully-supervised), a binary video-level label (weakly-supervised), a default one (one-class), or absent (unsupervised). However, in practice, it can be costly to collect y as anomalies are rare, and $\mathbf{V}$ itself due to potential privacy concerns. Moreover, both label and video data may need regular updates due to evolving application contexts.

Differently, in this chapter, we introduce a novel setup for VAD, termed as *training-free VAD*. Under this setting, we aim to estimate the anomaly score of each $\mathbf{I} \in \mathbf{V}$ using only pre-trained models at inference time, *i.e.*, without any training or fine-tuning involving a training dataset $\mathcal{D}$.

Figure 3.3: The anomaly score predicted by Llama [111] over time for video *Shooting033* from UCF-Crime. We highlight some sample frames with their associated BLIP-2 captions to demonstrate that the caption can be semantically noisy or incorrect (red bounding boxes are for abnormal predictions, while blue bounding boxes are for normal predictions). Ground-truth anomalies are highlighted. In particular, the caption of the frame enclosed by a blue bounding box within the ground truth anomaly fails to accurately represent the visual content, leading to a wrong classification due to the low anomaly score given by the LLM.

3.3.2 Are LLMs good for VAD?

We propose to address training-free VAD by exploiting recent advances in LLMs. As the use of LLMs in VAD is still in its infancy [50], we first analyze the capabilities of LLMs in producing an anomaly score based on a textual description of a video frame.

To achieve this, we first exploit a state-of-the-art captioning model $\Phi_{\texttt{C}}$, *i.e.* BLIP-2 [56], to generate a textual description for each frame $\mathbf{I} \in \mathbf{V}$. We then treat anomaly score estimation as a classification task, asking an LLM $\Phi_{\texttt{LLM}}$ to select only one score from a list of 11 uniformly sampled values in the interval $[0, 1]$, where 0 means normal and 1 anomalous. We get the anomaly score as:

$$\Phi_{\texttt{LLM}}(\texttt{P}_C \circ \texttt{P}_F \circ \Phi_{\texttt{C}}(\mathbf{I})) \tag{3.1}$$

where $\texttt{P}_C$ is a context prompt that provides priors to the LLM regarding VAD, $\texttt{P}_F$ instructs the LLM on the desired output format to facilitate automated text parsing[1], and $\circ$ is the text concatenation operation. We devise $\texttt{P}_C$ to simulate a potential end user of a VAD system, *e.g.*, law enforcement agency, as we empirically observe that impersonation can be an effective way of guiding the output generation of the LLM. For example, we can form $\texttt{P}_C$ as: "*If you were a law enforcement agency, how would you*

[1]The exact form of $\texttt{P}_F$ can be found in the Appendix

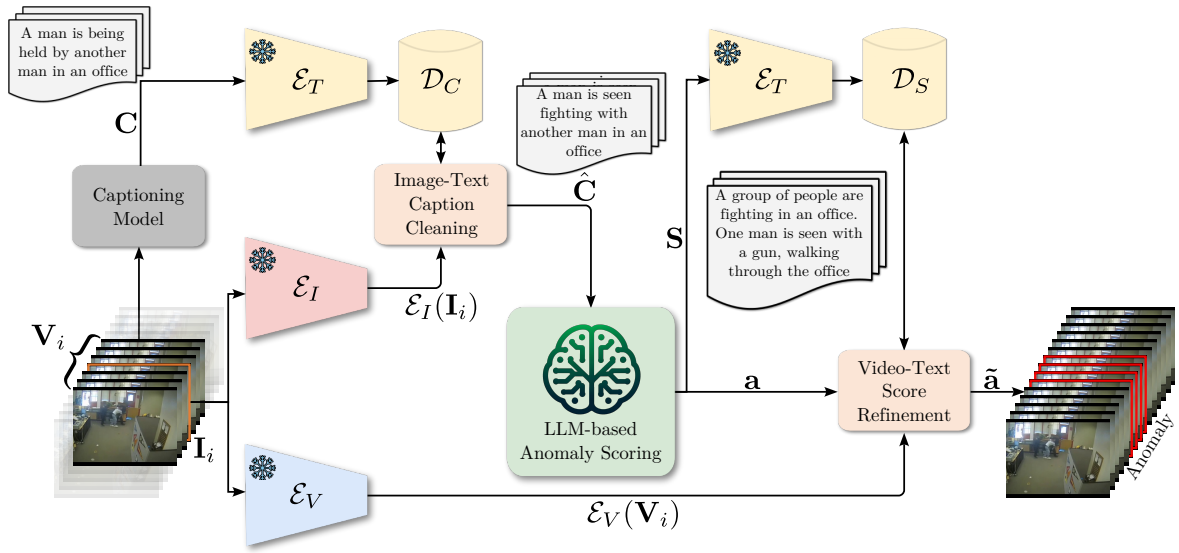


Figure 3.4: The architecture of our proposed LAVAD for addressing training-free VAD. For each test video $\mathbf{V}$, we first employ a captioning model to generate a caption C_i for each frame $\mathbf{I}_i \in \mathbf{V}$, forming a caption sequence $\mathbf{C}$. Our *Image-Text Caption Cleaning* component addresses noisy and incorrect raw captions based on cross-modal similarity. We replace the raw caption with a caption $\hat{C}_i \in \mathbf{C}$ whose textual embedding $\mathcal{E}_T(\hat{C}_i)$ is most aligned to the image embedding $\mathcal{E}_I(\mathbf{I}_i)$, resulting in a cleaned caption sequence $\hat{\mathbf{C}}$. To account for scene context and dynamics, our *LLM-based Anomaly Scoring* component further aggregates the cleaned captions within a temporal window centered around each $\mathbf{I}_i$ by prompting the LLM to produce a temporal summary S_i, forming a summary sequence $\mathbf{S}$. The LLM is then queried to provide an anomaly score for each frame based on its S_i, obtaining the initial anomaly scores $\mathbf{a}$ for all frames. Finally, our *Video-Text Score Refinement* component refines each a_i by aggregating the initial anomaly scores of frames whose textual embeddings of the summaries are mostly aligned to the representation $\mathcal{E}_V(\mathbf{V}_i)$ of the video snippet $\mathbf{V}_i$ centered around $\mathbf{I}_i$, leading to the final anomaly scores $\tilde{\mathbf{a}}$ for detecting the anomalies (anomalous frames are highlighted) within the video.

rate the scene described on a scale from 0 to 1, with 0 representing a standard scene and 1 denoting a scene with suspicious activities?". Note that $\mathtt{P}_C$ does not encode any prior on the type of anomalies itself, but just on the context.

Finally, with the estimated anomaly score from Eq. (3.1), we measure the VAD performance using the standard area under the curve of the receiver operating characteristic (AUC ROC). Fig. 3.2 reports the results obtained on the test set of the UCF-Crime dataset [104] with different variants of BLIP-2 for obtaining the frame captions, and with different LLMs, including Llama [111] and Mistral [46], for computing the frame-level anomaly scores. For reference, we also provide the state-of-the-art performance under

the unsupervised setting (the closest setting to ours) [108], and the random scoring as lower-bound. The plot demonstrates that state-of-the-art LLMs possess anomaly detection capabilities, largely outperforming random scoring. However, this performance is much lower w.r.t. trained state-of-the-art methods, even in an unsupervised setting.

We observe that two aspects might be the limiting factors in LLMs' performance. Firstly, the frame-level captions can be very noisy: the captions might be broken or may not fully reflect the visual content (see Fig. 3.3). Despite the use of BLIP-2 [56], the best off-the-shelf captioning model, some captions appear corrupted, thus leading to unreliable anomaly scores. Secondly, the frame-level caption lacks details about the global context and the dynamics of the scene, which are key elements when modeling videos. In the following, we address these two limitations and propose LAVAD, the first training-free method for VAD that leverages LLMs for anomaly scoring together with modality-aligned VLMs.

3.3.3 LAVAD: LAnguage-based VAD

LAVAD decomposes the VAD function f into five elements (see Fig. 3.4). As in the preliminary study, the first two are the captioning module $\Phi_{\texttt{C}}$ mapping images to textual descriptions in the language space $\mathcal{T}$, *i.e.*, $\Phi_{\texttt{C}} : \mathcal{I} \to \mathcal{T}$, and the LLM $\Phi_{\texttt{LLM}}$ generating text from language queries, *i.e.*, $\Phi_{\texttt{LLM}} : \mathcal{T} \to \mathcal{T}$. The other elements involve three encoders mapping input representations to a shared latent space $\mathcal{Z}$. Specifically we have the image encoder $\mathcal{E}_I : \mathcal{I} \to \mathcal{Z}$, the textual encoder $\mathcal{E}_T : \mathcal{T} \to \mathcal{Z}$, and the video encoder $\mathcal{E}_V : \mathcal{V} \to \mathcal{Z}$ for videos. Note that all five elements involve only off-the-shelf frozen models.

Following the positive findings of the preliminary analysis, LAVAD leverages $\Phi_{\texttt{LLM}}$ and $\Phi_{\texttt{C}}$ to estimate the anomaly score for each frame. We design LAVAD to address the limitations related to noise and lack of scene dynamics in frame-level captions by introducing three components: i) Image-Text Caption Cleaning through the vision-language representations of $\mathcal{E}_I$ and $\mathcal{E}_T$, ii) LLM-based Anomaly Scoring, encoding temporal information via $\Phi_{\texttt{LLM}}$ and iii) Video-Text Score Refinement of the anomaly scores via video-text similarity, using $\mathcal{E}_V$ and $\mathcal{E}_T$. In the following, we describe each component in detail.

Image-Text Caption Cleaning. For each test video $\mathbf{V}$, we first employ $\Phi_{\texttt{C}}$ to generate a caption C_i for each frame $\mathbf{I}_i \in \mathbf{V}$. Specifically, we denote as $\mathbf{C} = [C_1, \ldots, C_M]$ the sequence of captions, where $C_i = \Phi_{\texttt{C}}(\mathbf{I}_i)$. However, as shown in Sec. 3.3.2, the raw captions can be noisy, with broken sentences or incorrect descriptions. To mitigate this

issue, we rely on the captions of the whole video $\mathbf{C}$ assuming that in this set there exist captions that are unbroken and better capture the content of their respective frames, an assumption often verified in practice as the video features a scene captured by static cameras at a high frame rate. Thus, semantic content among frames can overlap regardless of their temporal distances. From this perspective, we treat caption cleaning as finding the *semantically* closest caption to a target frame $\mathbf{I}_i$ within $\mathbf{C}$.

Formally, we make use of vision-language encoders and form a set of caption embeddings by encoding each caption in $\mathbf{C}$ via $\mathcal{E}_T$, *i.e.* $\{\mathcal{E}_T(C_1), \dots, \mathcal{E}_T(C_M)\}$. For each frame $\mathbf{I}_i \in \mathbf{V}$, we compute its closest semantic caption as:

$$\hat{C}_i = \arg\max_{C \in \mathbf{C}} \langle \mathcal{E}_I(\mathbf{I}_i) \cdot \mathcal{E}_T(C) \rangle, \tag{3.2}$$

where $\langle \cdot, \cdot \rangle$ is the cosine similarity, and $\mathcal{E}_I$ the image encoder of the VLM. We then build the cleaned set of captions as $\hat{\mathbf{C}} = [\hat{C}_1, \dots, \hat{C}_M]$, replacing each initial caption C_i with its counterpart $\hat{C}_i$ retrieved from $\mathbf{C}$. By performing the caption cleaning process, we can propagate the captions of frames that are semantically more aligned to the visual content, regardless of their temporal positioning, to improve or correct noisy descriptions.

LLM-based Anomaly Scoring. The obtained caption sequence $\hat{\mathbf{C}}$, while being cleaner than the initial set, lacks temporal information. To overcome this, we leverage the LLM to provide temporal summaries. Specifically, we define a temporal window of T seconds, centered around $\mathbf{I}_i$. Within this window, we uniformly sample N frames, forming a video snippet $\mathbf{V}_i$, and a caption sub-sequence $\hat{\mathbf{C}}_i = \{\hat{C}_n\}_{n=1}^{N}$. We can then query the LLM with $\hat{\mathbf{C}}_i$ and a prompt $\mathtt{P}_S$ to get the temporal summary S_i centered on frame $\mathbf{I}_i$:

$$S_i = \Phi_{\texttt{LLM}}(\mathtt{P}_S \circ \hat{\mathbf{C}}_i) \tag{3.3}$$

where the prompt $\mathtt{P}_S$ is formed as "*Please summarize what happened in few sentences, based on the following temporal description of a scene. Do not include any unnecessary details or descriptions.*"[2].

Coupling Eq. (3.3) with the refinement process of Eq. (3.2), we obtain a textual description of the frame (S_i) which is semantically and temporally richer than C_i. With S_i, we can then query the LLM for estimating an anomaly score. Following the same prompting strategy described in Sec. 3.3.2, we ask $\Phi_{\texttt{LLM}}$ to assign to each temporal

[2] $\hat{\mathbf{C}}_i$ is represented as an ordered list, with items separated by `\n`.

summary S_i a score a_i in the interval $[0, 1]$. We get the score as:

$$a_i = \Phi_{\texttt{LLM}}(\texttt{P}_C \circ \texttt{P}_F \circ S_i) \tag{3.4}$$

where, as in Sec. 3.3.2, $\texttt{P}_C$ is a context prompt containing VAD contextual priors, and $\texttt{P}_F$ provides information on the desired output format.

Video-Text Score Refinement. By querying the LLM for each frame in the video with Eq. (3.4), we obtain the initial anomaly scores of the video $\mathbf{a} = [a_1, \ldots, a_M]$. However, $\mathbf{a}$ is purely based on the language information encoded in their summaries, without taking into account the whole set of scores. Thus, we further refine them by leveraging the visual information to aggregate scores from semantically similar frames. Specifically, we encode the video snippet $\mathbf{V}_i$ centered around $\mathbf{I}_i$ using $\mathcal{E}_V$ and all the temporal summaries using $\mathcal{E}_T$. Let us define $\mathbf{K}_i$ as the set of indices of the K-closest temporal summaries to $\mathbf{V}_i$ in $\{S_1, \ldots, S_M\}$, where the similarity between $\mathbf{V}_i$ and a caption S_j is the cosine similarity, *i.e.* $\langle \mathcal{E}_V(\mathbf{V}_i), \mathcal{E}_T(S_j) \rangle$. We obtain the refined anomaly score $\tilde{a}_i$:

$$\tilde{a}_i = \sum_{k \in \mathbf{K}_i} a_k \cdot \frac{e^{\langle \mathcal{E}_V(\mathbf{V}_i), \mathcal{E}_T(S_k) \rangle}}{\sum_{k \in \mathbf{K}_i} e^{\langle \mathcal{E}_V(\mathbf{V}_i), \mathcal{E}_T(S_k) \rangle}} \tag{3.5}$$

where $\langle \cdot, \cdot \rangle$ is the cosine similarity. Note that Eq. (3.5) exploits the same principles of Eq. (3.2), refining frame-level estimations (*i.e.*, score/captions) using their visual-language similarity (*i.e.*, video/image) with other frames in the video. Finally, with the refined anomaly scores for the test video $\tilde{\mathbf{a}} = [\tilde{a}_1, \ldots, \tilde{a}_M]$, we identify the anomalous temporal windows via thresholding.

3.4 Experiments

We validate our training-free proposal LAVAD on two datasets in comparison with state-of-the-art VAD methods that are trained with different levels of supervision, as well as training-free baselines. We conduct an extensive ablation study to justify our main design choices regarding the proposed components, prompt design, and score refinement. In the following, we first describe our experimental setup in terms of datasets and performance metrics. We then present and discuss the results in Sec. 3.4.1, followed by the ablation study in Sec. 3.4.2.

Datasets. We evaluate our method using two commonly used VAD datasets featuring real-world surveillance scenarios, *i.e.*, UCF-Crime [104] and XD-Violence [130].

Table 3.1: Comparison with state-of-the-art weakly-supervised, one-class, unsupervised and training-free methods on the UCF-Crime dataset. The best results among training-free methods are highlighted in bold.

METHOD	BACKBONE	AUC(%)
SULTANI *ET AL.* [104]	C3D-RGB	75.41
SULTANI *ET AL.* [104]	I3D-RGB	77.92
IBL [150]	C3D-RGB	78.66
GCL [142]	ResNext	79.84
GCN [151]	TSN-RGB	82.12
MIST [28]	I3D-RGB	82.30
WU *ET AL.*[130]	I3D-RGB	82.44
CLAWS [144]	C3D-RGB	83.03
RTFM [110]	VideoSwin-RGB	83.31
RTFM [110]	I3D-RGB	84.03
WU & LIU [129]	I3D-RGB	84.89
MSL [57]	I3D-RGB	85.30
MSL [57]	VideoSwin-RGB	85.62
S3R [128]	I3D-RGB	85.99
MGFN [16]	VideoSwin-RGB	86.67
MGFN [16]	I3D-RGB	86.98
SSRL [55]	I3D-RGB	87.43
CLIP-TSA [48]	ViT	87.58
SVM [104]	-	50.00
SSV [101]	-	58.50
BODS [118]	I3D-RGB	68.26
GODS [118]	I3D-RGB	70.46
GCL [142]	ResNext	74.20
TUR *ET AL.* [112]	ResNet	65.22
TUR *ET AL.* [113]	ResNet	66.85
DYANNET [108]	I3D	79.76
ZS CLIP [90]	ViT	53.16
ZS IMAGEBIND (IMAGE) [30]	ViT	53.65
ZS IMAGEBIND (VIDEO) [30]	ViT	55.78
LLAVA-1.5 [63]	ViT	72.84
LAVAD	ViT	**80.28**

UCF-Crime is a large-scale dataset that is composed of 1900 long untrimmed real-world surveillance videos, covering 13 real-world anomalies. The training set consists of 800 normal and 810 anomalous videos, while the test set includes 150 normal and 140 anomalous videos.

XD-Violence is another large-scale dataset for violence detection, comprising 4754

Table 3.2: Comparison with state-of-the-art weakly-supervised, one-class, unsupervised and training-free methods on the XD-Violence dataset. $*$ denotes results reported in [107]. The best results among training-free methods are highlighted in bold.

Method	Backbone	AP(%)	AUC(%)
Wu *et al.* [130]	C3D-RGB	67.19	-
Wu *et al.* [130]	I3D-RGB	73.20	-
MSL [57]	C3D-RGB	75.53	-
Wu and Liu[129]	I3D-RGB	75.90	-
RTFM [110]	I3D-RGB	77.81	-
MSL [57]	I3D-RGB	78.28	-
MSL [57]	VideoSwin-RGB	78.58	-
S3R[128]	I3D-RGB	80.26	-
MGFN [16]	I3D-RGB	79.19	-
MGFN [16]	VideoSwin-RGB	80.11	-
Hasan *et al.* [35]	AE^{RGB}	-	50.32*
Lu *et al.* [69]	Dictionary	-	53.56*
BODS [118]	I3D-RGB	-	57.32*
GODS[118]	I3D-RGB	-	61.56*
RareAnom [107]	I3D-RGB	-	68.33*
ZS CLIP [90]	ViT	17.83	38.21
ZS ImageBind (Image) [30]	ViT	27.25	58.81
ZS ImageBind (Video) [30]	ViT	25.36	55.06
LLaVA-1.5 [63]	ViT	50.26	79.62
LAVAD	ViT	**62.01**	**85.36**

untrimmed videos with audio signals and weak labels that are collected from both movies and YouTube. XD-Violence captures 6 categories of anomalies and it is divided into a training set of 3954 videos and a test set of 800 videos.

Performance Metrics. We measure the VAD performance using the area under the curve (AUC) of the frame-level receiver operating characteristics (ROC) as it is agnostic to thresholding for the detection task. For the XD-Violence dataset, we also report the average precision (AP), *i.e.*, the area under the frame-level precision-recall curve, following the established evaluation protocol in [130].

Implementation Details. We sample each video every 16 frames for computational efficiency. We employ BLIP-2 [56] as the captioning module $\Phi_{\texttt{C}}$. Particularly, we consider an ensemble of BLIP-2 model variants in our Image-Text Caption Cleaning technique. We use Llama-2-13b-chat [111] as our LLM module $\Phi_{\texttt{LLM}}$. We use multimodal encoders provided by ImageBind [30]. Specifically, the temporal window is $T = 10$ seconds, in

line with the pre-trained video encoder of ImageBind. We employ $K = 10$ in Video-Text Score Refinement.

3.4.1 Comparison with state of the art

We compare LAVAD against state-of-the-art approaches, including unsupervised methods [142, 112, 113, 108, 107], one-class methods [35, 69, 118, 104, 101], and weakly-supervised methods [104, 150, 142, 151, 28, 130, 144, 110, 129, 57, 57, 128, 16, 55, 48]. In addition, as none of the above methods specifically address VAD in a training-free setup, we further introduce a few training-free baselines with VLMs, *i.e.*, CLIP [90], ImageBind [30], and LLaVa [63].

Specifically, we introduce Zero-shot CLIP [90] (ZS CLIP) and Zero-shot ImageBind [30] (ZS ImageBind). For both baselines, we exploit their pre-trained encoders to compute the cosine similarities of each frame embedding against the textual embeddings of two prompts: *a standard scene* and *a scene with suspicious or potentially criminal activities*. We then apply a softmax function to the cosine similarities to obtain the anomaly score for each frame. Since ImageBind also supports the video modality, we include ZS ImageBind (Video) using the cosine similarities of the video embeddings against the two prompts. We choose ViT-B/32 [23] as the visual encoder for ZS-CLIP, ViT-H/14 [23] as the visual encoders for ZS-ImageBind (Image, Video), and both utilize CLIP's text encoder [90]. Finally, we introduce a baseline based on LLaVA-1.5, where we directly query LLaVa [63] to generate an anomaly score for each frame, using the same context prompt as in ours. LLaVA-1.5 uses CLIP ViT-L/14 [90] as the visual encoder and Vicuna-13B as the LLM.

Tab. 3.1 presents the results of the full comparison against the state-of-the-art methods, as well as our introduced training-free baselines, on the UCF-Crime dataset [104]. Notably, our method without any training demonstrates superior performance compared to both the one-class and unsupervised baselines, achieving a higher AUC ROC, with a significant improvement of +6.08% when compared to GCL [142] and a minor improvement of +0.52% against the current state of the art obtained by DyAnNet [108].

Moreover, it is evident that training-free VAD is a challenging task as a naive application of VLMs to VAD, such as ZS CLIP, ZS ImageBind (Image) and ZS ImageBind (Video), leads to poor VAD performance. VLMs are mostly trained to attend to foreground objects, rather than actions or the background information in an image that contributes to the judgment of anomalies. This might be the main reason for the poor generalization of VLMs on the VAD task. The baseline LLaVA-

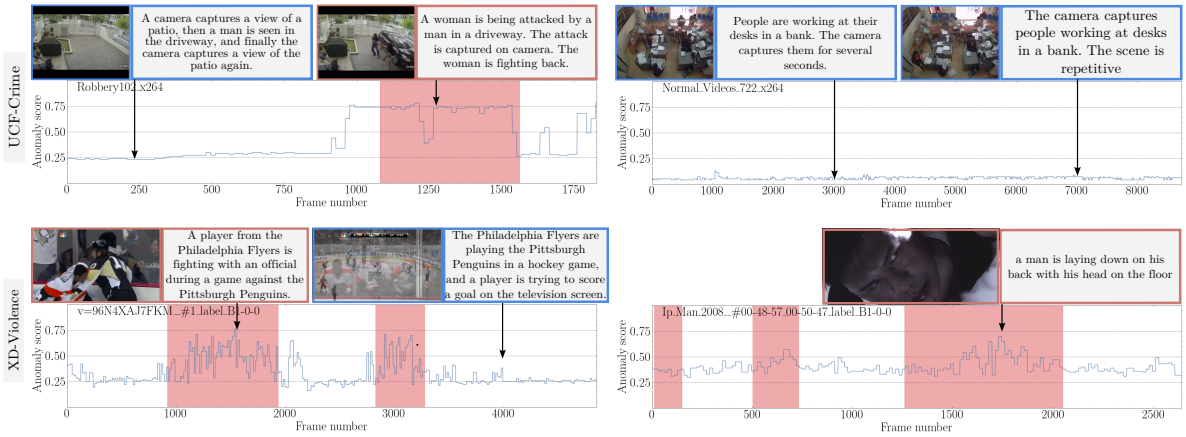


Figure 3.5: We showcase qualitative results obtained by LAVAD on four test videos, including two videos (top row) from UCF-Crime and two videos from XD-Violence (bottom row). For each video, we plot the anomaly score over frames computed by our method. We display some keyframes alongside their most aligned temporal summary (blue bounding boxes for normal frame predictions and red bounding boxes for abnormal frame predictions), illustrating the relevance among the predicted anomaly score, visual content, and description. Ground-truth anomalies are highlighted.

1.5, which directly prompts for the anomaly score for each frame, achieves a much higher VAD performance than directly exploiting VLMs in a zero-shot manner. Yet, its performance is still inferior to ours, where we leverage a richer temporal scene description for anomaly estimation, instead of a single-frame basis. The similar effect of the temporal summary is also confirmed by our ablation study as presented in Tab. 3.3. We also report the comparison against state-of-the-art methods and our baselines evaluated on XD-Violence in Tab. 3.2. Ours achieves superior performance compared to all one-class and unsupervised methods. In particular, LAVAD outperforms RareAnom [107], the best-scoring unsupervised method, by a substantial margin of +17.03% in terms of AUC ROC.

Qualitative Results. Fig. 3.5 shows qualitative results of LAVAD with sample videos from UCF-Crime and XD-Violence, where we highlight some keyframes with their temporal summaries. In the three abnormal videos (Row 1, Column 1, and Row 2), we can see that the temporal summaries of the keyframes during the anomalies accurately portray the visual content regarding the anomalous situations, which in turn benefits LAVAD to correctly identify the anomalies. In the case of *Normal_ Videos_ 722* (row 1, column 2), we can see that LAVAD consistently predicts a low anomaly score throughout the video.

3.4.2 Ablation study

In this section, we present the ablation study conducted with the UCF-Crime dataset. We first ablate the effectiveness of each proposed component of LAVAD. Then, we demonstrate the impact of task-related priors in the context prompt $\texttt{P}_C$ when prompting the LLM for estimating the anomaly scores. We also show the effect of K when aggregating the K semantically closest frames in the Video-Text Score Refinement component. Finally, we investigate the sensitivity of LAVAD to the choice of the captioning model used to generate frame descriptions.

Effectiveness of each proposed component. We ablate different variants of our proposed method LAVAD to prove the effectiveness of the three proposed components, including Image-Text Caption Cleaning, LLM-based Anomaly Score, and Video-Text Score Refinement. Tab. 3.3 shows the results of all ablated variants of LAVAD. When the Image-Text Caption Cleaning component is omitted (Row 1), *i.e.*, the LLM only exploits the raw captions to perform temporal summary and obtain the anomaly scores with refinement, the VAD performance degrades by -3.8% compared to LAVAD in terms of AUC ROC (Row 4). If we do not perform temporal summary, and only rely on the cleaned captions with refinement (Row 2), we observe a significant performance drop of -7.58% compared to LAVAD in AUC ROC, indicating that the temporal summary is an effective booster for LLM-based anomaly scoring. Finally, if we only use the anomaly scores obtained with the temporal summary on cleaned captions, without the final aggregation of semantically similar frames (Row 3), we can see that the AUC ROC decreases with a significant margin of -7.49% compared to LAVAD, proving that Video-Text Score Refinement also plays an important role in improving the VAD performance.

Task priors in the context prompt. We investigate the impact of different priors in the context prompt $\texttt{P}_C$ and present the results in Tab. 3.4. In particular, we experimented on two aspects, *i.e.*, impersonation and anomaly prior, which we believe can potentially benefit the estimation of LLM. Impersonation may help the LLM to process the input from the perspective of potential end users of a VAD system, while anomaly prior, *e.g.*, anomalies are criminal activities, may provide the LLM with a more relevant semantic context. Specifically, we ablate LAVAD with various context prompts $\texttt{P}_C$. We begin with a base context prompt: "*How would you rate the scene described on a scale from 0 to 1, with 0 representing a standard scene and 1 denoting a scene with suspicious activities?*" (Row 1). We inject only the anomaly prior by appending "*suspicious activities*" with "*or potentially criminal activities*" (Row 2). We incorporate only impersonation by

adding "*If you were a law enforcement agency,*" at the beginning of the base prompt (Row 3). Finally, we integrate both priors into the base context prompt (Row 4). As shown in Tab. 3.4, for videos within UCF-Crime, the anomaly prior appears to have a negligible effect on the LLM's assessment for anomalies, while impersonation improves the AUC ROC by +0.96% compared to the one obtained with only the base context prompt. Interestingly, incorporating both priors does not further boost the AUC ROC. We hypothesize that a more stringent context might limit the detection of a wider range of anomalies.

Effect of K on refining anomaly score. In this experiment, we investigate how the VAD performance changes in relation to the number of semantically similar temporal summaries, *i.e.*, K, used for refining the anomaly score of each frame. As depicted in Fig. 3.6, the AUC ROC metric consistently increases as K increases, and saturates when K approaches 9. The plot confirms the contribution of accounting for semantically similar frames in obtaining more reliable anomaly scores of the video.

Impact of different captioning models. To understand the sensitivity of LAVAD to the choice of the captioning model, we evaluate multiple BLIP-2 [56] variants and their ensemble on both UCF-Crime [104] and XD-Violence [130] (see Tabs. 3.5 and 3.6).

On UCF-Crime (Tab. 3.5), the ensemble approach (Row 6) achieves the best performance. The low-resolution nature of CCTV footage in this dataset often leads individual models to hallucinate specific actions. For instance, a model may incorrectly generate *"a person riding a skateboard down a road"* when the image only depicts a road in the absence of any specific event. By selecting the semantically closest captions from an ensemble of candidates, the model more effectively filters out these hallucinations in favor of more accurate scene descriptions.

Conversely, on XD-Violence (Tab. 3.6), the *flan-t5-xxl* variant (Row 3) performs best. We observe that while some variants prioritize foreground objects, *flan-t5-xxl* better captures background elements that often constitute anomalies in this dataset, such as *"a vehicle enveloped in smoke on a busy street."*. In this context, the ensemble approach can be detrimental, as the cleaning step may favor generic foreground captions over the more specific, anomaly-related descriptions produced by a single specialized variant. These findings suggest that while ensembling provides robustness to low-resolution noise, selecting a captioning model that can capture subtle scene-level changes (*e.g.*, smoke) is preferable when anomalies are not defined by salient foreground objects or actions.

Table 3.3: Results of LAVAD variants w/o each proposed component on the UCF-Crime Dataset.

Image-Text Caption Cleaning	LLM-based Anomaly Scoring	Video-Text Score Refinement	AUC (%)
✗	✓	✓	76.48
✓	✗	✓	72.70
✓	✓	✗	72.79
✓	✓	✓	**80.28**

Table 3.4: Results of LAVAD on UCF-Crime with different priors in the context prompt when querying the LLM for anomaly scores.

Anomaly prior	Impersonation	AUC (%)
✗	✗	79.32
✓	✗	79.38
✗	✓	**80.28**
✓	✓	79.77

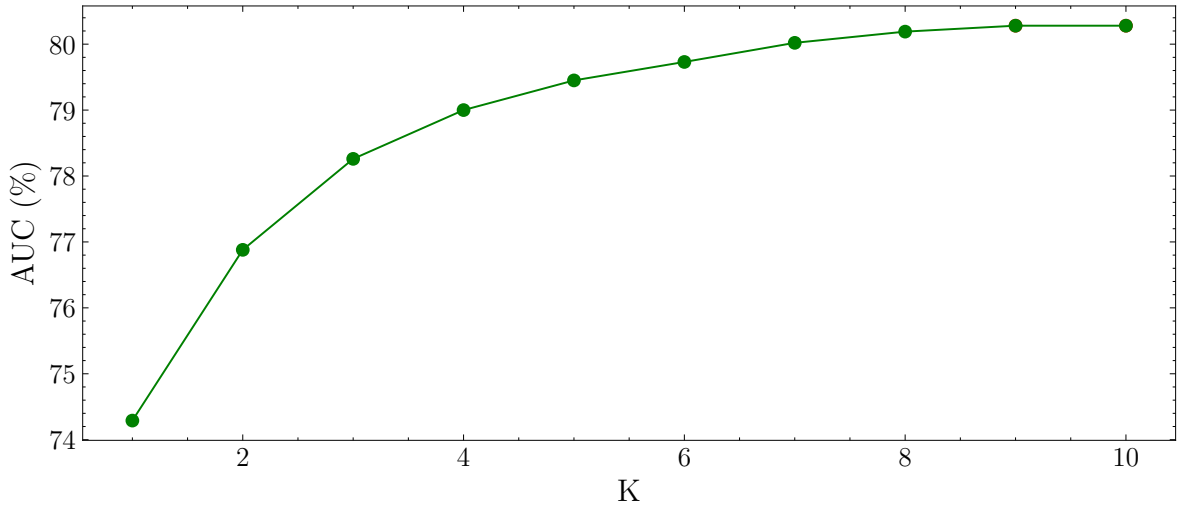


Figure 3.6: Results of LAVAD on UCF-Crime over the number of K semantically similar frames used for anomaly score refinement.

3.5 Chapter Summary

In this chapter, we introduced LAVAD, a pioneering method to address training-free VAD. LAVAD follows a language-driven pathway for estimating the anomaly scores, leveraging off-the-shelf LLMs and VLMs. LAVAD has three main components, where the first uses image-text similarities to clean the noisy captions provided by a captioning

Table 3.5: Results of LAVAD on UCF-Crime with different BLIP-2 model variants in our Image-Text Caption Cleaning technique.

BLIP-2					AUC
FLAN-T5-XL	FLAN-T5-XL-COCO	FLAN-T5-XXL	OPT-6.7B	OPT-6.7B-COCO	(%)
✓	✗	✗	✗	✗	74.19
✗	✓	✗	✗	✗	74.49
✗	✗	✓	✗	✗	74.38
✗	✗	✗	✓	✗	75.50
✗	✗	✗	✗	✓	73.94
✓	✓	✓	✓	✓	**80.28**

Table 3.6: Results of LAVAD on XD-Violence with different BLIP-2 model variants in our Image-Text Caption Cleaning technique.

BLIP-2					AP	AUC
FLAN-T5-XL	FLAN-T5-XL-COCO	FLAN-T5-XXL	OPT-6.7B	OPT-6.7B-COCO	(%)	(%)
✓	✗	✗	✗	✗	61.09	85.16
✗	✓	✗	✗	✗	57.41	82.78
✗	✗	✓	✗	✗	**62.01**	**85.36**
✗	✗	✗	✓	✗	56.55	82.42
✗	✗	✗	✗	✓	54.71	82.93
✓	✓	✓	✓	✓	59.62	84.90

model; the second leverages an LLM to aggregate scene dynamics over time and estimate anomaly scores; and the final component refines the latter by aggregating scores from semantically close frames according to video-text similarity. We evaluated LAVAD on both UCF-Crime and XD-Violence, demonstrating superior performance compared to training-based methods in the unsupervised and one-class setting, without the need for training and additional data collection.

Although LAVAD removes the need for task-specific training by shifting decision-making entirely to inference time, it operates under an offline assumption, requiring access to the complete video sequence. This limits its applicability in real-world settings where systems must process streaming data and provide timely feedback. Addressing this constraint requires extending training-free video understanding to causal, online inference, which is the focus of the next chapter.

Chapter 4

Training-free Online Video Step Grounding

While the training-free approach in Chapter 3 represents a step toward scalable video understanding, it assumes an offline setting in which the entire video sequence is available for global reasoning. In many real-world applications, however, systems must operate causally on live video streams, providing immediate feedback without access to future observations. This chapter extends the training-free paradigm to online inference by integrating Large Multimodal Models (LMMs) with Bayesian filtering, allowing beliefs over procedural steps to be updated incrementally as visual evidence becomes available.

4.1 Introduction

Grounding procedural steps in videos is crucial for enabling machines to follow along and assist humans in complex tasks like cooking a recipe, assembling furniture, or performing maintenance work. This ability is particularly valuable for real-time procedural guidance in AR/XR applications, where recognizing task progress allows users wearing headsets or smart glasses to receive timely, step-specific instructions. Specifically, the task of Video Step Grounding (VSG) takes as input a list of procedural steps extracted from an instructional article (*e.g.*, a recipe or how-to guide), and a video performing the same task, with the goal of identifying which of the steps are performed in the video.

Existing VSG approaches align procedural steps descriptions with their corresponding video frames [17, 34, 58, 76]. However, these strategies face two key limitations. First, they need a training set, entailing the cost of collecting (and potentially annotating) it. Moreover, a training set could bias models toward the specific videos and procedural

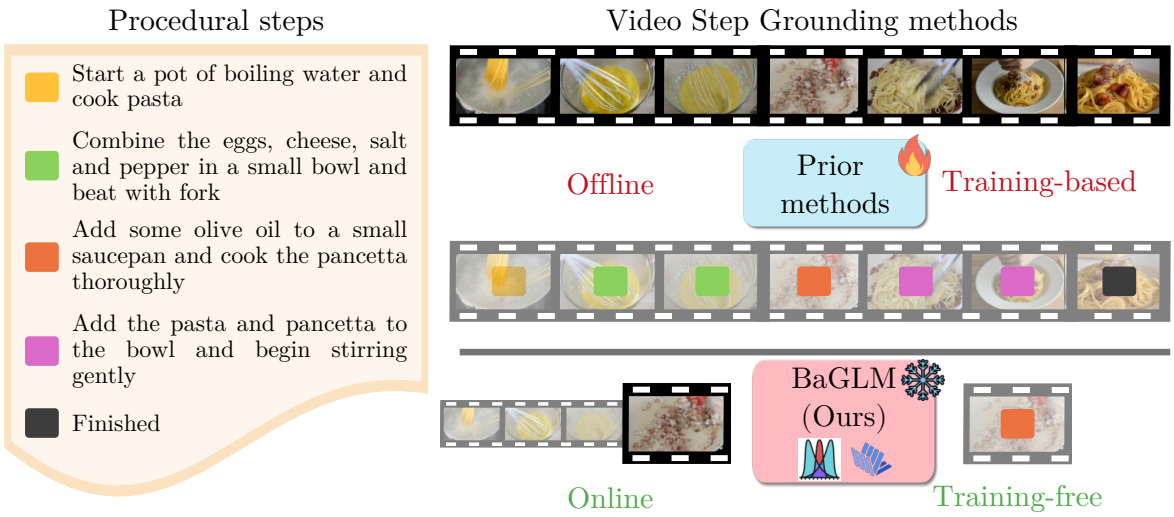


Figure 4.1: We tackle **Video Step Grounding** with **BaGLM**, a *training-free* approach which combines Bayesian filtering with Large Multimodal Models to enable *online* inference over video streams.

tasks that are depicted, limiting their generalization capability. Second, they assume offline processing, where the entire video is available ahead of time. This makes them unsuitable for real-world applications processing a live video stream.

To overcome these limitations, we explore how to address VSG *online* and *without training*, *i.e.*, operating on a streaming video (Fig. 4.1). This objective is challenging as it requires performing predictions with partial evidence (*i.e.*, having access to only a subset of the video frames) and without the possibility of extracting task-specific representations that would be typically learned from a dedicated training set. In this scenario, we explore the zero-shot capabilities of powerful Large Multimodal Models (LMMs) [5, 54, 19] for VSG. Specifically, given the set of steps, we prompt LMMs to predict the step corresponding to the current video segment. Surprisingly, models such as InternVL [19] could already surpass state-of-the-art training-based methods [58, 76], despite only having access to a single segment at a time. This highlights a strong potential for addressing VSG without training.

Building on the zero-shot capabilities of LMMs, we show how performance can be further improved by modeling temporal dependencies across steps and leveraging past information. To this end, we propose Bayesian Grounding with Large Multimodal Models (BaGLM), a training-free approach for VSG that combines Bayesian probabilistic modeling with LMMs. We harness a Large Language Model (LLM) to estimate a dependency matrix, capturing whether one step is a prerequisite of another. From this matrix, we compute transition probabilities to each step in the current video segment

(*i.e.*, the *predict* step of the Bayesian filter). This prior refines the LMM direct prediction (*i.e.*, the *update* step), injecting past temporal knowledge into the model's output. The transition model is updated over time, following the progress of each step estimated by the LMM. On three publicly available datasets (HT-Step [2], CrossTask [155], Ego4D Goal-Step [102]), BAGLM consistently outperforms existing methods with significant margins, achieving state-of-the-art results on this challenging task.

We make four key **contributions**: ① We present the first study to address VSG in an online, training-free setting, eliminating the need for data collection and better aligning with practical application needs; ② We show that the zero-shot LMMs can surpass specialized, training-based methods, revealing their potential for addressing VSG. ③ We propose a method, BAGLM, which incorporates priors from past video frames into LMMs through Bayesian filtering, modeling the temporal dependencies across steps via LLM queries; ④ We extensively evaluate BAGLM on three datasets, showing that it outperforms state-of-the-art offline methods.

4.2 Related Work

Video Step Grounding. Earlier works in VSG adopted weakly-supervised learning approaches. For instance, Zhukov *et al.* [155] proposed to learn a model from instructional narrations and a list of steps derived from temporal constraints, sharing components across tasks with similar actions or objects. Han *et al.* [34] proposed a co-training framework that combines a Temporal Alignment Network (TAN) with a dual-encoder architecture, predicting step boundaries by aligning videos and narrations, using pseudo-labels derived from cross-modal agreement. VINA [76] considered step descriptions from WikiHow [126] and learned to temporally ground them in videos without manual supervision. Recent methods have increasingly leveraged language models and large-scale pretraining. MPTVA [17] introduced a multi-pathway alignment strategy using LLM-filtered narration summaries and multiple sources of alignment, merging them to create robust pseudo-labels for training. NaSVA [58] addressed multi-sentence grounding by leveraging LLMs to transform noisy ASR transcripts into procedural steps, aligning them with video content using a narration-based similarity score.

Together, these approaches illustrate the progress from weakly supervised models leveraging narrations to more sophisticated ones that integrate language models, multimodal alignment, and robust pseudo-labeling. However, most methods still rely on extensive training, domain-specific fine-tuning, and access to the full video, assumptions that limit their real-world applicability. To the best of our knowledge, BAGLM is the

first training-free online solution that addresses these limitations.

Video-Language Alignment refers to the task of measuring the semantic consistency between a video and its corresponding textual description. Early approaches [38, 99] tackled this by relying on the cosine similarity between video frames and captions within the embedding space of CLIP [90]. However, these methods are inherently limited by the well-known shortcoming of CLIP, *i.e.*, by its inability to effectively capture temporal dynamics in text descriptions. As a result, recent works have shifted toward leveraging LMMs [60, 127, 131, 53, 146] and adopting metrics such as VQAScore [61], which are obtained from video question answering to better account for the temporal dimension.

Building on these recent studies, we propose to employ an LMM to assess the alignment between instructional steps and temporal video segments, requiring fine-grained video understanding. VSG is particularly challenging because key steps in instructional tasks often involve similar objects or scenes. For instance, "inserting a screw" or "aligning parts" may look similar in tasks like "Assemble a chair" and "Fix a table", making them hard to distinguish without nuanced semantic understanding.

4.3 On using Large Multimodal Models for Video Step Grounding

Large Multimodal Models are powerful pretrained models that have demonstrated impressive zero-shot performance on a wide variety of tasks without further tuning. As previous solutions for VSG are training-based, we wonder whether off-the-shelf LMMs can address VSG. This section begins by providing a formal definition of the VSG task, clearly distinguishing between its offline and online settings (Sec. 4.3.1). We then present the findings of our preliminary empirical investigation into the use of LMMs for addressing VSG, along with key insights gained from this study (Sec. 4.3.2).

4.3.1 Video Step Grounding

Given a video of a task composed of a series of actions, video step grounding aims to detect which actions (or steps) appear in the video. Formally, let us denote the set of steps composing a task as $\mathcal{A} = \{a_i\}_{i=1}^{K}$, where each step $a_i \in \mathcal{A}$ is expressed in the natural language space $\mathcal{T}$, and K is the number of steps. Moreover, let us denote with $\mathbf{V}$ a video in the space $\mathcal{V}$, split into T non-overlapping segments $\mathcal{S} = \{\mathbf{S}_t\}_{t=1}^{T}$. VSG

aims to identify which steps in $\mathcal{A}$ are shown in the video.

The offline setting of VSG, as addressed by previous works [58, 76], assumes access to the whole video (*i.e.*, the full set $\mathcal{S}$) when performing the task. In this chapter, we focus on *online* VSG, assuming a video stream where segments arrive one after the other, and we perform the task having only access to the current segment and the segments preceding it, *i.e.*, $\mathcal{S}_{1:t} = \{\mathbf{S}_1, \dots, \mathbf{S}_t\}$. Thus, online VSG aims to predict whether a segment $\mathbf{S}_t \in \mathcal{S}$ depicts a step $a_i \in \mathcal{A}$, given only the previous video segments $\mathcal{S}_{1:t}$. While we do not exploit this possibility, a model may also use the current segment to update predictions on past ones, contrary to the single prediction of the offline setting.

4.3.2 Large Multimodal Models are strong baselines for VSG

Let us define a large multimodal model $f_{\texttt{LMM}}$ via three elements: the visual encoder $f_{\texttt{vid}}$, the text encoder $f_{\texttt{txt}}$, and a text decoder $f_{\texttt{dec}}$. The encoders map their respective inputs into a shared d-dimensional embedding space, *i.e.*, $f_{\texttt{vid}} : \mathcal{V} \to \mathbb{R}^d$ and $f_{\texttt{txt}} : \mathcal{T} \to \mathbb{R}^d$. The decoder maps the visual and textual inputs into a probability simplex $\Delta^{|\mathcal{W}|}$, over the LLM vocabulary $\mathcal{W}$[1], *i.e.*, $f_{\texttt{dec}} : \mathbb{R}^d \times \mathbb{R}^d \to \Delta^{|\mathcal{W}|}$. The next token is sampled from this probability vector.

Preliminary experiment. We propose to frame the problem of VSG as a multi-choice question answering task where the LMM, prompted with the current video segment and all possible steps, has to predict as answer either one of them or "none". Formally, given the current segment $\mathbf{S}_t$, we prompt the LMM $f_{\texttt{LMM}}$ using information regarding the task and step, obtaining the score for a step a_i as:

$$f_{\texttt{LMM}}(\mathbf{S}_t, \pi_{\texttt{VSG}})[i] = f_{\texttt{dec}}\left(f_{\texttt{vid}}(\mathbf{S}_t), f_{\texttt{txt}}(\pi_{\texttt{VSG}})\right)[i] \tag{4.1}$$

where $\pi_{\texttt{VSG}}$ is the task prompt and $f_{\texttt{dec}}(\cdot,\cdot)[i]$ denotes the probability that the next predicted token is i (corresponding to step a_i), normalizing the scores across the multi-choice options. Note that, to account for no-step occurring, we include an additional option "none of the above" ($f_{\texttt{dec}}(\cdot,\cdot)[K+1]$). By normalizing the LMMs' probabilities over each choice, we map the segment into a probability simplex over the steps and the "none" option, *i.e.*, $f^{\texttt{VSG}}_{\texttt{LMM}} : \mathcal{V} \times \mathcal{A} \to \Delta^{K+1}$.

Datasets & metrics. We evaluate methods on three public datasets: *CrossTask* [155], *HT-Step* [2], and *Ego4D Goal-Step* [102]. *HT-Step* is a benchmark for procedural step grounding [2], where the goal is to align steps from an instructional article with an

[1]For simplicity, we omit the words' tokenization, assuming text prompts and videos are encoded equally.

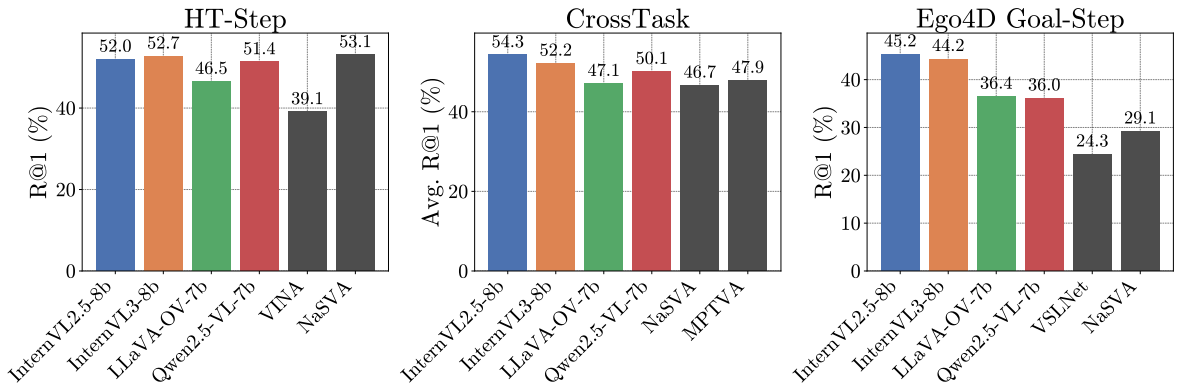


Figure 4.2: Comparison of VSG performance on HT-Step, CrossTask, and Ego4D Goal-Step datasets, prompting LMMs with step options and video segments in an online fashion. For reference, we also show the performance of the top two performing methods from the state of the art (dark bars).

input how-to video. The dataset provides two types of test sets: one for seen activities and another for unseen activities. The seen validation and test splits follow [76], each containing 600 videos in total, with 5 videos per activity across 120 activities. We evaluate with the validation set of seen classes, as the evaluation server hosting the test sets for the seen and unseen classes is unavailable.

CrossTask is an established instructional video benchmark for zero-shot step localization [155]. It contains about 4.8k instructional videos, covering 18 primary tasks and 65 related tasks. Only videos in the primary tasks are annotated as steps with temporal segments from a predefined taxonomy. We follow the same evaluation set as indicated in [58], using videos from primary tasks.

Ego4D Goal-Step [102], a subset of the Ego4D [32], includes 851 videos averaging 26 minutes in length. Unlike CrossTask and HT-Step, Ego4D videos are collected without predefined tasks. Annotators label them hierarchically, first identifying goals (*e.g.*, *Makes the bread*), then steps (*e.g.*, *Prepares the bread*), and finally substeps (*e.g.*, *weigh the dough*). We evaluate on its validation split.

For both HT-Step and CrossTask, we follow the standard evaluation protocol [58, 17], providing for each video the full set of steps for its task as multiple-choice options in the prompt. On average, each task includes about 10 steps in HT-Step and 7.5 in CrossTask. For Ego4D Goal-Step, we use step-level descriptions only (excluding substeps) and apply text normalization with `spaCy`[2]: we lowercase, lemmatize (preserving plural nouns and verbal adjectives), and normalize whitespace and punctuation. After preprocessing, the

[2]We use the model available at `https://spacy.io/models/en#en_core_web_sm`

average number of steps per video is 17.

Following [58, 17], we report Recall@1 (R@1) on HT-Step and Ego4D Goal-Step, measuring whether the top-scoring timestamp for each step falls within the ground-truth interval. For CrossTask, we report Average Recall@1 (Avg.R@1), computed by averaging per-task R@1 across all primary tasks.

Discussion. Fig. 4.2 shows the results on the three datasets, using four LMMs with strong performance in video understanding benchmarks [88]: LLAVA-OneVision-Qwen2-7B [54], Qwen2.5-VL-7B-Instruct [5], InternVL2.5-8B [19], and InternVL3-8B [154]. For reference, we also include the top two state-of-the-art methods on each dataset, considering both in-domain ones, *i.e.*, trained and evaluated on the same dataset (VINA [76] and NaSVA [58] on HT-Step, VSLNet [149] on Ego4D Goal-Step) and out-of-domain ones (*i.e.*, NaSVA on CrossTask and Ego4D Goal-Step, MPTVA [17] on CrossTask). Among the LMMs, InternVL2.5-8B scores the best performance on CrossTask and Ego4D Goal-Step, outperforming MPTVA by 6.4% and NaSVA by 16.1%, respectively. However, on HT-Step, NaSVA slightly surpasses InternVL2.5-8B (53.1 vs. 52.0). Overall, the LMMs outperform prior methods on CrossTask and Ego4D Goal-Step, while showing comparable performance on HT-Step.

Remarks. Considering the lack of task-specific tuning, these results demonstrate that zero-shot LMMs perform surprisingly well on VSG, accessing only the current segment. A natural question is whether we can (i) inject information from past segments into LMM's predictions, refining them, while (ii) maintaining the zero-shot, training-free advantages of LMMs. In the following, we explore how we can achieve this by drawing inspiration from Bayesian filtering principles.

4.4 Bayesian Grounding with Large Multimodal Models

In Sec. 4.3 we have shown that LMMs are effective at VSG without any task-specific tuning. However, they act without memory of past knowledge, performing step prediction by only looking at the current segment. Therefore, uncertain predictions (*e.g.*, due to the segment acquisition) cannot benefit from past evidence (*e.g.*, step performed in the previous segment), leading to potential model mistakes.

Formally, Eq. (4.1) provides an estimate for $P(\mathtt{A} = a_t|\mathbf{S}_t)$, where $\mathtt{A}$ is a discrete random variable taking values in the set $\mathcal{A} \cup \mathtt{none}$ with $\mathtt{none}$ denoting any action outside the set of steps. Differently, we would like to perform step prediction conditioned on all

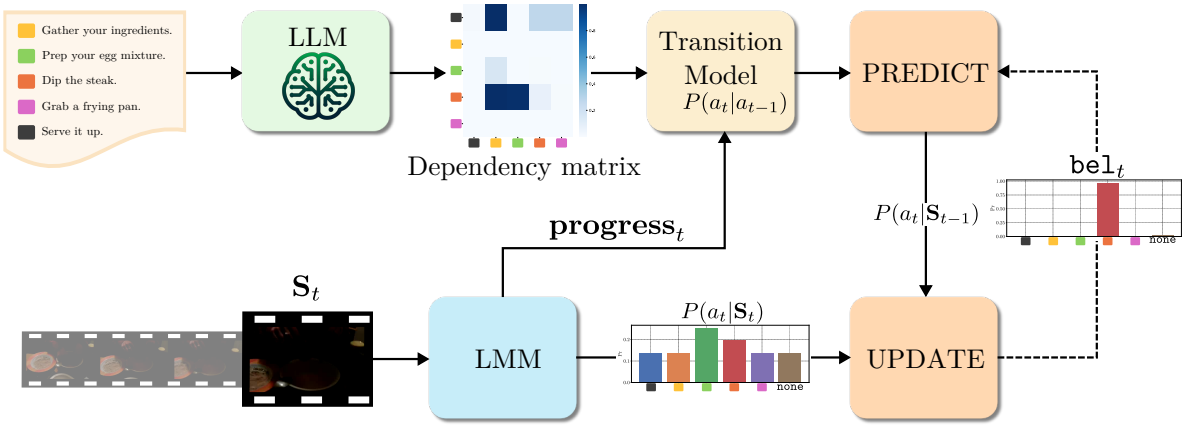


Figure 4.3: **Overview of BAGLM.** Given a sequence of steps, an LLM is used to estimate a dependency matrix among them. This matrix is used to compute step transition probabilities employed during the **predict** step of a Bayesian filter. As the video progresses, the transition model is updated using estimates of each step's progress from an LMM. The **update** step of the filter merges this with the predictions from the LMM, refining the output.

previous segments, estimating $P(\mathtt{A} = a_t|\mathcal{S}_{[1:t]})$, without the need of storing the whole history. Inspired by Bayesian filtering, a probabilistic technique addressing sequence modeling from past observations [18], we propose *Bayesian Grounding with Large Multimodal Models (BAGLM)*. BAGLM (Fig. 4.3) estimates the transition probabilities across steps through the step dependencies estimated by an LLM, to refine LMMs' predictions.

In the following, we will first present the generic formulation of Bayesian filtering in Sec. 4.4.1, and how we revisit the predict (Sec. 4.4.2) and update (Sec. 4.4.3) steps for VSG using LMMs and LLMs.

4.4.1 Bayesian filtering

Let $\mathbf{x}$ be a state of a process, $\mathbf{X}_t$ be its corresponding random variable at time t, and $\mathbf{z}_i$ be the observation at time $i \leq t$. The goal of Bayesian filtering [18] is to compute the posterior:

$$\texttt{bel}_t(\mathbf{x}) = P(\mathbf{X}_t = \mathbf{x}|\mathbf{z}_1, \cdots, \mathbf{z}_t),$$

via two essential steps: the **predict** step first computes a prior over the possible predictions using past estimations; the **update** step then estimates the prior with the current observations. In the following, we will denote $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \cdots, \mathbf{z}_t\}$ for simplicity.

Using the chain rule, we can write the posterior as:

$$\texttt{bel}_t(\mathbf{x}) = P(\mathbf{X}_t = \mathbf{x}|\mathbf{z}_{1:t}) = \frac{1}{P(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \cdot P(\mathbf{z}_t|\mathbf{X}_t = \mathbf{x}, \mathbf{z}_{1:t-1}) \cdot P(\mathbf{X}_t = \mathbf{x}|\mathbf{z}_{1:t-1}), \quad (4.2)$$

where the first term is a normalization factor, the second is the likelihood of the current observation given the past ones and the current state, and the last is the prior over the states from the observations.

To further simplify Eq. (4.2), we consider two *assumptions*: (i) we have a hidden Markov observation model, and thus $P(\mathbf{z}_t|\mathbf{X}_t = \mathbf{x}, \mathbf{z}_{1:t-1}) = P(\mathbf{z}_t|\mathbf{X}_t = \mathbf{x})$; (ii) we have an initial prior over the states independent from the first observation, *i.e.*, $P(\mathbf{X}_0 = \mathbf{x}|\mathbf{z}_1) = P(\mathbf{X}_0 = \mathbf{x})$.

Adding the first assumption to the second term and using the Chapman-Kolmogorov equation on $P(\mathbf{X}_t = \mathbf{x}|\mathbf{z}_{1:t-1})$, we obtain:

$$\texttt{bel}_t(\mathbf{x}) = \overbrace{\frac{1}{\underbrace{P(\mathbf{z}_t|\mathbf{z}_{1:t-1})}_{\text{normalization factor}}} \cdot \underbrace{P(\mathbf{z}_t|\mathbf{X}_t = \mathbf{x})}_{\text{likelihood}}}^{\textbf{update product}} \cdot \overbrace{\sum_{\mathbf{x}_i \in \mathcal{X}} \underbrace{P(\mathbf{X}_t = \mathbf{x}|\mathbf{X}_{t-1} = \mathbf{x}_i)}_{\text{transition model}} \cdot \underbrace{\texttt{bel}_{t-1}(\mathbf{x}_i)}_{\text{accumulated belief}}}^{\textbf{predict step}}, \quad (4.3)$$

where $\mathcal{X}$ is the set of possible states. The second term corresponds to the **predict** step, computing an estimate of the current state using the prior belief. The transition model describes the likelihood of a state given the previous one, while accumulated belief denotes the likelihood of the previous state as recursively accumulated via Eq. (4.3). The first term refers to the **update** step, where the predicted state probability is multiplied by the likelihood and normalization factor to obtain the final estimate.

A Bayesian filtering view on VSG. To adapt Eq. (4.3) to VSG we must define our states and observations. The state is what we want to estimate, *i.e.*, the step a in the current segment. The observation is the input we receive from the environment: the segment $\mathbf{S}$ itself. Thus, we obtain the update step as:

$$\texttt{bel}_t(a) = P(\texttt{A}_t = a|\mathcal{S}_{1:t}) = \frac{P(\mathbf{S}_t|\texttt{A}_t = a, \mathcal{S}_{1:t-1})}{P(\mathbf{S}_t|\mathcal{S}_{1:t-1})} \cdot \sum_{a_i \in \mathcal{A}} P(\texttt{A}_t = a|\texttt{A}_{t-1} = a_i) \cdot \texttt{bel}_{t-1}(a_i), \quad (4.4)$$

where $\texttt{A}_j$ is a random variable over the possible steps for the j^{th} segment. We keep the original model's assumptions: an initial prior over steps independent of the first segment, and conditional independence of the current segment given the step (which holds for step-level semantics). In the following, we detail the implementation of each component

in Eq. (4.4).

4.4.2 PREDICT: modeling dependencies among steps via language and progress priors

A peculiarity of VSG is that actions depend on each other: these dependencies provide priors on the actions performed in future segments, allowing us to build a transition model, needed in Eq. (4.4). We exploit the internal knowledge of an LLM to estimate such dependencies. Specifically, we query the LLM to identify when a step must be completed before another can occur (*i.e.*, is a prerequisite). As the dependency might be ambiguous, we instruct the LLM to estimate a probability rather than a binary score, resulting in a matrix $\mathbf{D} \in \mathbb{R}^{K \times K}$, where each entry $\mathbf{D}_{i,j}$ is the probability that step a_j is a prerequisite of step a_i. We initialize the transition matrix as $\mathbf{T} = \mathbf{D}^\top$, allowing for self-transitions (*i.e.*, $\mathbf{T}_{i,i} = 1$), and transitions from all steps to those with no prerequisites (*i.e.*, $\mathbf{T}_{i,j} = 1$ if $\sum_{j=1}^{K} \mathbf{T}_{i,j} = 0$), normalizing $\mathbf{T}$ across rows.

Modeling the action progress. The transition matrix $\mathbf{T}$ is static and purely based on the task description, without reflecting how the likelihood of a step evolves during the video. For example, if *boil water* is a prerequisite for *cook pasta* and *boil water* has not yet finished, the video cannot transition to *cook pasta*. Instead, it is more likely to continue showing *boil water* or switch to a parallel steps like *prepare the sauce*. Conversely, once *cook pasta* is completed, it becomes unlikely that the video will return to an earlier step as *boil water*. We therefore adjust $\mathbf{T}$ accounting for both step dependencies and the estimated step progress, introducing two scores: *readiness* and *validity*. Intuitively, a step is ready when its prerequisites are sufficiently complete, while a step is a valid candidate for a segment if its successors have not yet been completed [97].

To achieve this, we query the LMM to infer the execution progress of each step within a video segment. Given a segment $\mathbf{S}$ and the prompt $\pi_{\texttt{prog}}$, we define the estimated progress for step a_i as:

$$\mathbf{progress}_t[i] = \sum_{j=0}^{9} j \cdot f_{\texttt{LMM}}(\mathbf{S}_t, \pi_{\texttt{prog}})[j],$$

where $f_{\texttt{LMM}}(\mathbf{S}_t, \pi_{\texttt{prog}})[j]$ denotes the model's output probabilities over the vocabulary. We treat the model's probability distribution over the tokens $\{0, 1, \ldots, 9\}$, as the distribution over progress levels. From the latter and $\mathbf{D}$, we can compute whether an action is ready

to be performed.

Step readiness. For each step a_i, we compute readiness as the weighted maximum progress of its prerequisite steps, *i.e.*,

$$\mathbf{r}_t[i] = \frac{\sum_{j=1}^{K} \mathbf{D}_{i,j} \cdot \max_{\tau<t} \mathbf{progress}_\tau[j]}{\sum_{j=1}^{K} \mathbf{D}_{i,j}}, \tag{4.5}$$

where we measure the progress of a step as its maximum value across all preceding segments (*i.e.*, $\tau < t$), performing a weighted average across all steps. With Eq. (4.5), we get high values in case all predecessors of an action have a high progress, and low otherwise.

Step validity. Contrary to the readiness, the step validity is given by:

$$\mathbf{v}_t[i] = \frac{\sum_{j=1}^{K} \mathbf{D}_{j,i} \cdot (1 - \max_{\tau<t} \mathbf{progress}_\tau[j])}{\sum_{j=1}^{K} \mathbf{D}_{j,i}}. \tag{4.6}$$

This value is high when no successors of a_i in $\mathbf{D}$ have been executed yet, and low otherwise. Finally, we adjust the transition matrix $\mathbf{T}$ by accounting for readiness and validity of each step, *i.e.*,

$$\tilde{\mathbf{T}}_t[i,j] = \frac{\mathbf{T}[i,j] \cdot \mathbf{r}_t[j] \cdot \mathbf{v}_t[j]}{\sum_{k=1}^{K} \mathbf{T}[i,k] \cdot \mathbf{r}_t[k] \cdot \mathbf{v}_t[k]}. \tag{4.7}$$

Predict step. Exploiting the transition model of Eq. (4.7), the predict step of Eq. (4.4) becomes:

$$\texttt{predict}_t(a_i) = \sum_{a_j \in \mathcal{A}} \tilde{\mathbf{T}}_t[j,i] \cdot \texttt{bel}_{t-1}(a_j). \tag{4.8}$$

4.4.3 UPDATE: using LMM estimates to re-weigh the belief over steps

In the **update step**, we multiply the step prior for the observation likelihood $P(\mathbf{S}_t|\mathtt{A}_t = a)$ and a normalization factor independent from the steps. However, due the cardinality of $\mathcal{V}$, computing $P(\mathbf{S}_t|a_t, \mathcal{S}_t)$ is intractable. On the other hand, we follow Eq. (4.1), estimating $(\mathbf{S}_t|\mathtt{A}_t = a)$ directly from the LMM. Formally, we use the Bayes rule and write:

$$P(\mathbf{S}_t|\mathtt{A}_t = a_i) = \frac{P(\mathbf{S}_t)}{P(\mathtt{A}_t = a)} \cdot P(\mathtt{A}_t = a_i|\mathbf{S}_t) = \frac{P(\mathbf{S}_t)}{P(\mathtt{A}_t = a_i)} \cdot f_{\texttt{LMM}}(\mathbf{S}_t, \pi_{\texttt{VSG}})[i], \tag{4.9}$$

where $P(\mathbf{S}_t)$ is a prior on the segments independent from the steps, $P(\mathtt{A}_t = a_i)$ is a prior over the steps independent from the observation. We replace $P(\mathtt{A}_t = a|\mathbf{S}_t)$ with the LMM prediction. While for $P(\mathtt{A}_t = a_i)$ there are various possible choices, in our approach, we consider the prior to be uniform.

Final filtering model. Considering the uniform prior over the states and merging Eq. (4.8) into Eq. (4.9), we obtain the final belief $\mathtt{bel}_t(a_i)$ for step a_i and segment $\mathbf{S}_t$ as:

$$\mathtt{bel}_t(a_i) = \frac{1}{\mathcal{Z}} \cdot f_{\mathtt{LMM}}(\mathbf{S}_t, \pi_{\mathtt{VSG}})[i]\mathcal{Z} \cdot \sum_{a_j \in \mathcal{A}} \tilde{\mathbf{T}}_t[j,i] \cdot \mathtt{bel}_{t-1}(a_j), \tag{4.10}$$

where $\mathcal{Z}$ is a normalization factor containing all elements independent of the specific step a_i.

4.5 Experiments

In this section, we describe our experimental protocol and present the comparison w.r.t. the state of the art (Sec. 4.5.1). We then provide a qualitative analysis of BaGLM to demonstrate its robustness against visual ambiguity and overlapping steps. Finally, we perform a detailed study on BaGLM (Sec. 4.5.2). We use the same datasets and metrics described in Sec. 4.3 in our experiments.

Implementation details. Our method is implemented considering InternVL2.5-8B [19] as our LMM, based on the results of Sec. 4.3. We employ LLaMA3-70B-Instruct [31] as our LLM of choice to derive our transition model. To test our model, we split videos into sequences of non-overlapping 2-second segments, providing them as input to the LMM one after the other. We ran all experiments on a single NVIDIA H100 64GB GPU, except for LLaMA3-70B-Instruct [31], which required 4 H100 GPUs.

Baselines. We compare our method with several state-of-the-art approaches for VSG: Zhukov *et al.* [155], HT100M [80], VideoCLIP [134], MCN [13], DWSA [98], MIL-NCE [79], VT-TWINS [51], UniVL [71], VINA [76], TAN* [34, 2], NaSVA [58], and MPTVA [17]. We also implement an online variant of NaSVA [58], which introduces causal masking in the transformer encoder's self-attention layers to restrict attention to past segments only. Reported results are taken from the original papers, except for NaSVA on HT-Step and Ego4D Goal-Step, and VSLNet [149] on Ego4D Goal-Step, where we use the authors' released code. These are marked with † in the tables.

All baselines are training-based and offline, except for our online variant of NaSVA, and use HowTo100M [80] as training set or pre-training, except for VSLNet (trained on

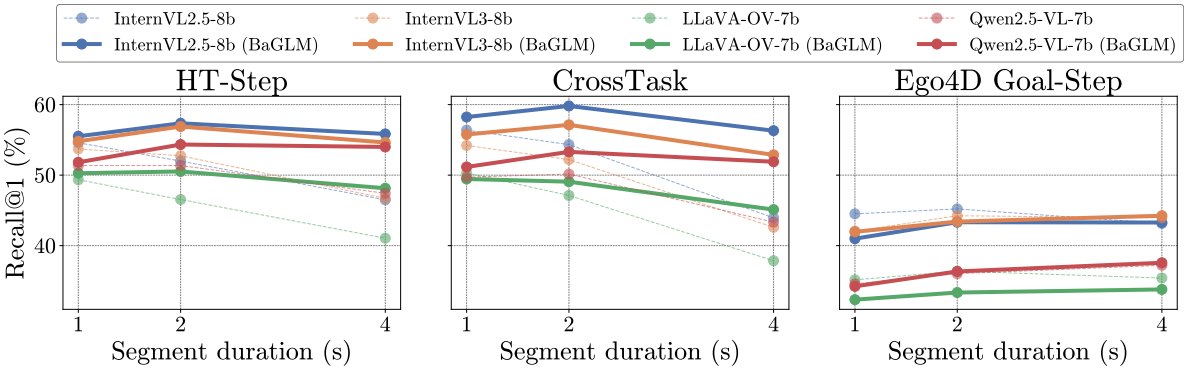


Figure 4.4: Ablation study on varying the segment duration and on the used LMM.

Ego4D Goal-Step), for Zhukov *et al.* [155] and DSWA [98] (trained on CrossTask) and for MPTVA [17] (trained on a subset of HowTo100M). For CrossTask, VideoCLIP [134] and UniVL [71] perform a further fine-tuning step on CrossTask data.

4.5.1 Comparison with state-of-the-art methods

Tab. 4.1a and Tab. 4.1b report the results of our evaluation in the three considered datasets. Overall, BAGLM, built on top of InternVL2.5-8B, outperforms all offline methods that rely on noisy supervision from narrations without manual annotations. Notably, it outperforms the current state-of-the-art method, NaSVA, by 4.3% on HT-Step, which consists of videos from HowTo100M, the same dataset used for NaSVA's self-training. The improvements of BAGLM over NaSVA are especially significant on CrossTask and Ego4D Goal-Step, with gains of 13.1% and 14.2%, respectively. The margins in CrossTask are remarkable as there exist methods (*i.e.*, VideoCLIP, UniVL) that are specifically fine-tuned for this domain. The same applies to Ego4D Goal-Step. This dataset is particularly challenging for models trained on HowTo100M due to the distribution shift introduced by its egocentric videos. On this dataset, BAGLM also outperforms VSLNet by a significant margin (+19%), despite VSLNet being trained on data from the same domain. Notably, all these results have been achieved in an online setting, with all competitors having access to the whole video, contrary to BAGLM. When evaluated under the same online setting, NaSVA's performance drops to 46.1 R@1 on HT-Step and 24.2 on Ego4D Goal-Step (-7% and -4.9% compared to its offline variant), remaining well below BAGLM (-11.3% and -19.1%). These results further highlight the effectiveness of our approach in the challenging online scenario.

Qualitative results. Fig. 4.5 showcases qualitative results produced by BAGLM on test videos from HT-Step (*Make Milanesa*) and CrossTask (*Make a Latte*). For each

Table 4.1: Comparison between state-of-the-art offline methods and our online method BaGLM.

(a) HT-Step and Ego4D Goal-Step

Method	HT-Step ↑ R@1	Ego4D Goal-Step ↑ R@1
Offline		
VSLNet [149]	-	24.3†
TAN* [34]	30.7	-
VINA [76]	39.1	-
NaSVA [58]	53.1†	29.1†
Online		
NaSVA [58]	46.1†	24.2†
BaGLM	**57.4**	**43.3**

(b) CrossTask

Method	↑ Avg. R@1
Offline	
Zhukov *et al.* [155]	22.4
HT100M [80]	33.6
VideoCLIP [134]	33.9
MCN [13]	35.1
DWSA [98]	35.3
MIL-NCE [79]	40.5
VT-TWINS [51]	40.7
UniVL [71]	42.0
VINA [76]	44.8
NaSVA [58]	46.7
MPTVA [17]	47.9
Online	
BaGLM	**59.8**

Table 4.2: Ablation study on the transition model.

Readiness	Validity	HT-Step	CrossTask	Ego4D
		55.9	58.0	42.1
✓		57.0	58.8	42.0
	✓	56.4	58.8	43.1
✓	✓	**57.4**	**59.8**	**43.3**

Table 4.3: Ablation study on varying the LLM.

Dataset	LLaMA-3.3-70B	GPT-4.1-mini
HT-Step	**57.4**	57.1
CrossTask	59.8	**60.9**
Ego4D	**43.3**	40.6

video, we plot ground truth boundaries alongside predictions from the off-the-shelf LMM and our method.

These results illustrate how BaGLM addresses scenarios where procedural steps are visually ambiguous or temporally overlapping. By leveraging Bayesian filtering, the model maintains a continuous belief distribution (bel_t) over all possible steps rather than forcing a hard binary choice. In the *Make Milanesa* video, the off-the-shelf LMM incorrectly assigns a high probability to *Dip the steak* at the start and *Prep your egg mixture* around `00:20` due to deceptive visual cues in those specific frames. In contrast, BaGLM manages this ambiguity by injecting temporal priors through the predict step of the filter (as per Eq. (4.8)). This occurs because the Bayesian filter does not process frames in isolation. It maintains a temporal memory of the task's progress, utilizing the transition model to recognize that an active step, such as dipping the steak, is likely to be ongoing despite momentary visual fluctuations.

This mechanism is particularly beneficial in the *Make a Latte* video for distinguishing visually similar actions like *pour milk* and *pour espresso*. For example, around the 1-minute mark, the model sees a liquid being poured. Because these actions look

Table 4.4: Results with oracle dependencies and step progress.

Progress Oracle	Dep. Matrix Oracle	HT-Step	CrossTask	Ego4D
		57.4	59.8	43.3
✓		44.9	38.6	36.0
	✓	54.6	58.3	45.2
✓	✓	**62.6**	**66.9**	**82.2**

visually similar in a short segment, the off-the-shelf LMM fluctuates between both steps. However, BAGLM is more certain that the action is *pour milk* because it accounts for the logical progress of the task and the previously observed steps. By resolving this visual confusion through the probabilistic recursive update of the filter, the framework maintains high precision and temporal consistency even in challenging causal, online settings.

4.5.2 Ablation studies

In this section, we analyze the key components of BAGLM, evaluating different configurations of the transition model, different LLMs for generating the dependency matrix, and experiments with oracle step dependencies and progress.

Transition Model. The transition model estimates the conditional probability of moving from one step to the next and is used in the predict step of Bayesian filtering, as per Eq. (4.8). We first evaluate a static transition matrix, initialized from the dependency matrix. We then study the effect of including the readiness score (*i.e.*, if prerequisites are not completed, Eq. (4.5)), the validity one (*i.e.*, if the step should not be re-executed, Eq. (4.6)), and both. As shown in Tab. 4.2, readiness improves the static model by 1.1% and 0.8% on two datasets and performs similarly to the static model on the third (-0.1%). Thus, accounting for the dependency matrix (and if prerequisites are met) tends to improve performance by reducing the score of non-executable actions.

Validity improves the static model by +0.5%, +0.8%, and +1.0% across the three datasets. By considering the status of actions that occur at a later time, it influences the predictions of their prerequisites (*e.g.*, *turn on the stove*) and prevents them from being repeated unnecessarily.

The best results come from combining both, improving performance by 1.5%, 1.8%, and 1.2%. This highlights the benefit of including both types of estimated priors when updating the transition model.

Choice of LLMs. Tab. 4.3 analyzes how BAGLM is affected by the LLM used to

generate dependencies between steps. We compare LLaMA-3.3-70B-Instruct [31] and GPT-4.1-mini [84]. The two models achieve comparable overall performance: LLaMA-3.3-70B performs better on HT-Step (+0.3%) and Ego4D GoalStep (+2.7%) but worse on CrossTask (-1.1%). This trend suggests that LLaMA-3.3-70B performs better on datasets with more generic step descriptions, whereas proprietary models like GPT-4.1-mini are more effective at capturing dependencies among more atomic actions.

LMMs and segment duration. In Fig. 4.4, we show how BAGLM's performance varies w.r.t. the segment duration (from 1 to 4 seconds), considering different LMMs: InternVL2.5 8B [19], InternVL3 8B [154], LLaVA-OneVision 7B [54], and Qwen2.5 7B [5].

From the figure, we can see three trends. First, BAGLM consistently improves the performance of the underlying LMM it is applied to across all segment durations on both HT-Step and CrossTask, with gains becoming more pronounced as segment duration increases. This is related to the second trend, the impact of the segment duration. Segments that are too short may lack sufficient visual cues to evaluate the video-step alignment or the progress, while longer ones may span over multiple actions, making it harder to localize steps precisely. Using 2-second segments offers the best trade-off, improving performance by +1.9%, +1.6%, +0.3%, and +2.3% across the three datasets compared to 1-second segments.

Third, BAGLM does not provide clear advantages on Ego4D Goal-Step (*i.e.*, -1.9%, -0.8%, -3%, and +0.3%). This can be attributed to challenges specific to this dataset. Ego4D Goal-Step videos are significantly longer on average (28 minutes vs. 6 minutes and less than 5 minutes for HT-Step and CrossTask, respectively) and contain coarser step annotations (53 seconds per segment on average vs. 16 seconds and 10.7 seconds for HT-Step and CrossTask, respectively). These longer videos and broader annotations result in more generic step descriptions, which in turn make it harder to infer accurate dependencies between steps and to estimate progress, as we will analyze in the following.

Use of annotated step boundaries from datasets. Given the different impact of BAGLM across datasets, we analyze the performance of the Bayesian filtering formulation without potential noise from the dependency matrix and/or the estimation of step progresses. With this aim, we conduct an analysis similar to Tab. 4.2, this time using step dependencies and action progress derived directly from the original datasets. For each video, we construct a chain of steps based on its ground truth temporal order. Note that while these dependencies reflect the observed execution order, *they are not* true semantic constraints, as the same steps may occur in different orders in other videos. Progress is computed as the normalized fraction of completion between a step's start and end timestamps.

Results of this oracle experiment are presented in Tab. 4.4, showing consistent gains across all datasets (*e.g.*, +5.2% on HT-Step), even the challenging Ego4D Goal-Step (*i.e.*, +38.9%). This confirms that the Bayesian filtering formulation is effective, and that jointly improving the elements used to estimate the transition matrix (*i.e.*, progress, dependency) would further boost the results of BAGLM.

4.6 Chapter Summary

In this chapter, we introduced BAGLM, a novel training-free approach for online video step grounding. BAGLM uses Bayesian filtering to integrate information from past video frames into LMM predictions. It consists of two key components: a transition model, initialized from step dependency matrices generated by LLMs and updated over time using action progress and dependency constraints; and an observation model, implemented as an LMM, which refines predictions as the video unfolds. We evaluated BAGLM on HT-Step, CrossTask, and Ego4D Goal-Step, showing that it outperforms training-based methods without requiring additional training or data collection.

The results obtained with BAGLM show that training-free video understanding can be extended to online settings through Bayesian filtering. However, the performance of the proposed framework depends on the reliability of the estimates produced by language models when conditioned on video. In particular, inconsistencies in these estimates directly affect the quality of the observations provided to the Bayesian filter and propagate to downstream inference by influencing subsequent belief updates. This observation motivates a deeper investigation into how language models can be better calibrated to produce reliable estimates over video, which is the focus of the next chapter.

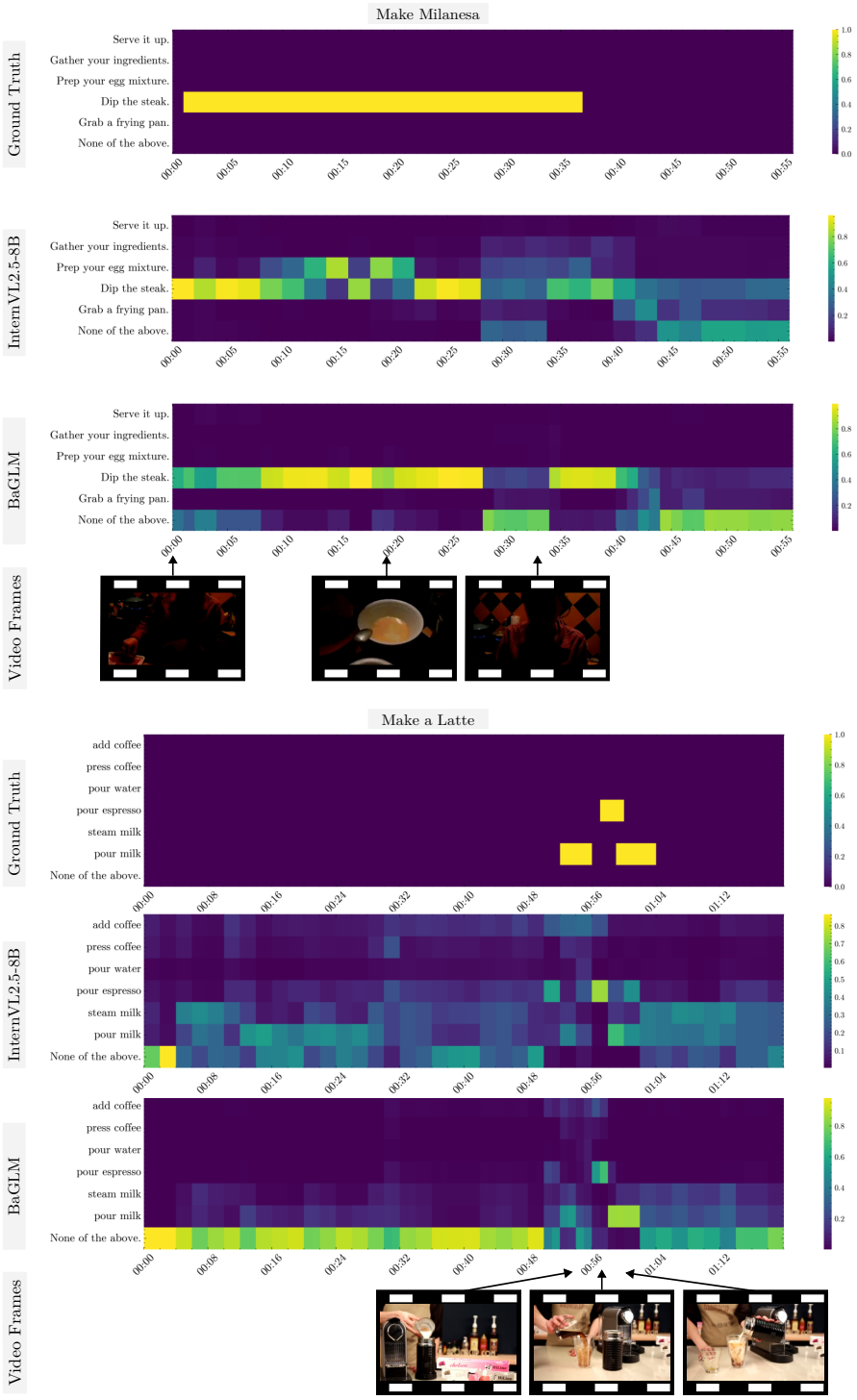


Figure 4.5: Qualitative results of BAGLM on test videos from HT-Step (*Make Milanesa*) and CrossTask (*Make a Latte*). Ground truth step boundaries and predicted step probabilities per segment are shown for both BAGLM and the off-the-shelf LMM. Arrows point to the timestamps of selected keyframes.

Chapter 5

Can Text-to-Video Generation help Video-Language Alignment?

The effectiveness of the zero-shot and training-free methods presented in the previous chapters ultimately depends on the reliability of the estimates produced by Large Multimodal Models (LMMs) when operating on video. In practice, these estimates can become inconsistent in the presence of complex or subtle temporal dynamics, directly affecting downstream inference, particularly in online settings. In this final research chapter, we explore synthetic data as a scalable means of improving the temporal understanding of these models. Specifically, we investigate whether synthetic videos generated by text-to-video models can be used to improve the reliability of the language model estimates within LMMs over video, without human annotation.

5.1 Introduction

Video-language alignment (VLA) aims to model the relationship between video content and natural language descriptions [134], a fundamental multimodal task that enables various applications, such as video captioning [27] and video-text retrieval [116]. This task is challenging because it requires the models to recognize not only the entities but also their spatial and temporal relationships.

Recent approaches exploit Large Multimodal Models (LMMs) to address VLA [7, 60, 61] by tasking the LMM to answer whether a given video and description are aligned. While effective, such LMMs often lack sufficient understanding of temporal dynamics, such as action types or temporal orders [66, 21]. This limitation also stems from the video-and-language datasets used for the LMM pre-training, as they are biased

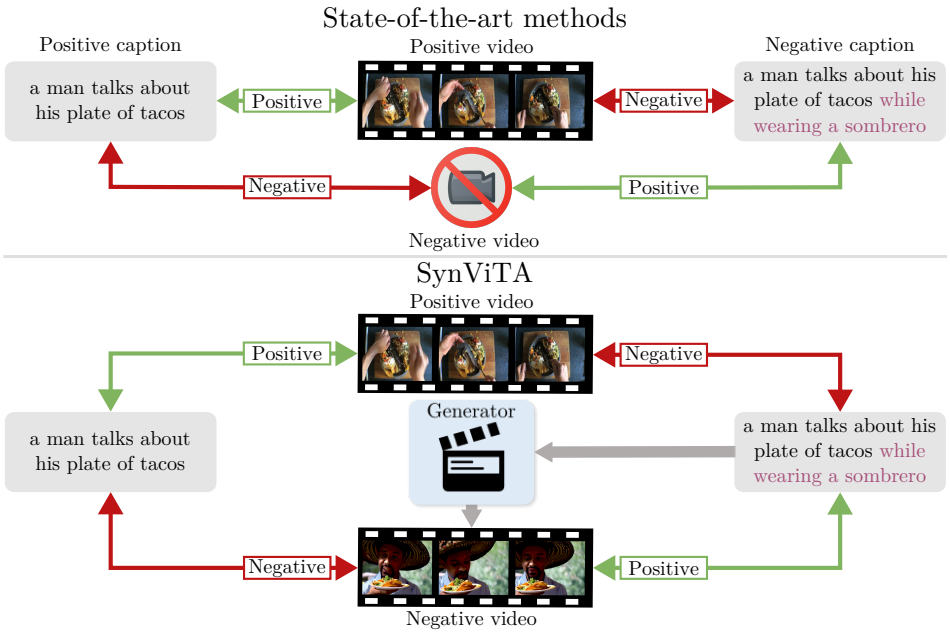


Figure 5.1: We study the problem of video-language alignment, *i.e.*, modeling the relationship between video content and text descriptions. Top: current methods use LLM-generated negative captions, which may introduce certain concepts (*e.g.*, *wearing a sombrero*) only as negatives, as they are not associated with any video. Bottom: we study whether overcoming this issue by pairing negative captions with generated videos can improve VLA.

towards frame-level semantics: the appearance of a single frame is often sufficient to infer the alignment with the textual caption [52, 81].

While a possible solution is to augment datasets with negative captions, *e.g.*, captions from other videos, these negatives can be "easy" for VLA models to distinguish simply by focusing on nouns; therefore, recent works focus on LLM-generated captions as hard negatives [7]. However, relying solely on textual negatives may cause the LMM to encounter concepts only as negatives, thus developing incorrect linguistic biases. For instance, in the VideoCon dataset [7], words like *sombrero*, *marshmallow*, and *bland* appear as textual negatives but not as positives. While a remedy is to augment the training set with videos corresponding to the hard negative captions, retrieving such videos from existing databases is not a feasible solution as they lack sufficient videos that vary only w.r.t. actions or temporal order while remaining similar in all the other semantic aspects [81]. An alternative pathway is generating synthetic videos by feeding hard negative captions to text-to-video generative models [138, 121, 14, 78]. While this idea has been investigated in the image domain [86], it remains largely unexplored for videos.

In this chapter, we aim to fill this gap and investigate, for the first time, the use of synthetic videos to improve VLA in temporal understanding. Specifically, we propose to leverage negative captions generated by existing models [7] and recent open-source text-to-video generators [138, 121, 14] to produce the corresponding synthetic videos (see Fig. 5.1). We first conduct a preliminary study to evaluate whether these generated videos can augment the training set of real videos and enhance performance on various video-related tasks. Our analysis shows that, while adding synthetic videos shows some promise, *it does not* consistently improve performance on temporally challenging downstream tasks, regardless of the generator. We also analyze the effects of different misalignment types (*i.e.*, semantically plausible changes in the video captions) on the generated videos. We notice that videos generated by, *e.g.*, introducing hallucination into the captions or reversing event order, align more with positive captions than with their target captions. Such noisy supervision signals may lead to ineffective learning, limiting improvements on downstream tasks.

Motivated by these preliminary findings, we argue that, when using synthetic videos for VLA we should account for (i) potential semantic inconsistency between input text and the generated videos and (ii) appearance biases, as synthetic videos may contain artifacts. We design SYNTHETIC VIDEOS FOR VIDEO-TEXT ALIGNMENT (SYNVITA), a model-agnostic method that can effectively tackle both challenges. SYNVITA addresses the semantic inconsistency problem by making the contribution of each synthetic video in the training objective proportional to their video-text alignment estimates [61]. Moreover, it accounts for appearance biases via a semantic regularization objective that (i) takes the common parts between the original and negative caption; (ii) encourages the model to focus on semantic changes rather than on the visual appearance difference between synthetic and real videos. We evaluate SYNVITA on the VideoCon [7] test sets with different LMMs [140, 59], and on temporally challenging downstream tasks, *i.e.*, text-to-video retrieval on SSv2-Temporal [96] and SSv2-Events [4] and video question answering on ATP-Hard [10]. On average, SYNVITA improves over state-of-the-art methods that do not use synthetic videos, demonstrating that synthetic videos can help VLA.

Contributions. To summarize, our contributions are:

- We pioneer the research problem in how to effectively leverage synthetic videos for VLA learning to improve temporal understanding;
- We conduct extensive analysis, shedding light on the potential benefits and limitations of using videos generated by state-of-the-art text-to-video generative

models;

- We propose a new learning method for VLA with synthetic videos, SynViTA, with a sample weighting strategy to mitigate noisy generations and a regularization term to enforce semantic understanding, instead of visual differences between synthetic and real videos.
- We evaluate SynViTA on different benchmarks with different LMMs, proving its model-agnostic effectiveness in aiding VLA for better temporal understanding.

5.2 Related Work

Video-language models for video understanding. Recent approaches for video understanding exploit the capabilities of foundation models. For instance, several works adapted CLIP [90], a model trained to compare images and texts, for video-language tasks, such as retrieval [26], captioning [72] or anomaly detection [145]. Other studies leveraged LLMs for reasoning over video captions [147, 123] or directly decode video features in natural language [133, 148, 59]. While these models heavily rely on pre-training on large-scale video-text pairs [134, 133], they still lack robustness in modeling temporal dynamics [66, 21]. Previous works addressed this by, *e.g.*, using LLMs to generate hard negatives [81], reversing the action sequence [4], or finer-grained objectives [122].

The closest work to ours is VideoCon [7], which finetunes an LMM using temporally challenging hard *textual* negatives. However, our focus is different, as we explore whether generated videos can improve video-text alignment, complementing negative captions.

Video-language alignment evaluation. A main challenge in VLA is quantifying the semantic alignment between text and video frames. Early attempts used metrics based on the CLIPScore [38, 99, 93], which computes video-text alignment by measuring the similarity between video frames and their captions in the CLIP embedding space [90]. However, as VLMs struggle with temporal changes in captions [141, 81, 4, 122], recent approaches have started measuring video-text alignment using LMMs for video question answering [7, 60, 127, 53, 131], such as the VQAScore in [61].

In this chapter, we use these models to evaluate the quality of the alignment and for the new objective of evaluating how much a synthetic video aligns with its textual counterpart.

Using synthetic visual data as training data. Recent works showed how augmenting training sets with synthetically generated images can improve the performance of

discriminative models [109, 87, 153, 36]. Diffusion models, known for their ability to generate highly realistic images and for their flexibility in dealing with different conditioning signals (text, depth, etc.), have significantly fostered this research trend [20]. While most works focused on image recognition tasks [109, 153, 86], recent approaches explored more challenging tasks such as few-shot recognition [36, 92] or out-of-distribution detection [24].

Our work follows a similar underlying idea and is motivated by recent advances in text-to-video generation [138, 121, 14, 78]. However, we are the first to explore synthetic videos for improving video understanding models.

5.3 Video-Language Alignment

Video-language alignment aims to rate how well the content of a video matches a given text in natural language. Formally, let us define t as the given textual input in the language space $\mathcal{T}$, and $\mathbf{V}$ as a video in the space $\mathcal{V}$. The goal is to learn a function f parameterized by θ, mapping videos and texts to their alignment scores, *i.e.*, $f : \mathcal{V} \times \mathcal{T} \rightarrow [0, 1]$, where 1 means high alignment and 0 the opposite.

Given the fine-grained nature of language, this task requires video-language models with compositional and temporal order understanding and recent approaches use LMMs for this task, where an LLM is used as decoder [7, 61]. Formally, let us define an LLM-based video-language model f via three functions: the visual encoder $f_{\texttt{vid}}$, the text encoder $f_{\texttt{txt}}$, and a decoder $f_{\texttt{dec}}$. The two encoders map their respective inputs into a shared d-dimensional embedding space, *i.e.*, $f_{\texttt{vid}} : \mathcal{V} \rightarrow \mathbb{R}^d$ and $f_{\texttt{txt}} : \mathcal{T} \rightarrow \mathbb{R}^d$. The decoder maps the visual and textual inputs into a vector in the probability simplex $\Delta^{|\mathcal{W}|}$ defined over the LLM vocabulary[1] $\mathcal{W}$, *i.e.*, $f_{\texttt{dec}} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \Delta^{|\mathcal{W}|}$. This probability vector is then used to sample the next token for the generative process.

Within this formulation, the alignment task becomes the probability of predicting `Yes` or `No` as the next word after the question π_q = `Does this video entail the description [`t`]?`, where $[t]$ is the target caption. Formally, this translates as f being:

$$f(\mathbf{V}, t) = \frac{P_{\mathcal{W}}(\texttt{Yes}|\mathbf{V}, t)}{P_{\mathcal{W}}(\texttt{Yes}|\mathbf{V}, t) + P_{\mathcal{W}}(\texttt{No}|\mathbf{V}, t)} \tag{5.1}$$

where $P_{\mathcal{W}}(\mathbf{w}|\mathbf{V}, t) = f_{\texttt{dec}}^{[\mathbf{w}]}\left(f_{\texttt{vid}}(\mathbf{V}), f_{\texttt{txt}}(\pi_q \circ t)\right)$, with π_q the shared question, $\circ$ string

[1]For simplicity, we omit the words' tokenization and we assume textual prompts and videos to be treated equally and encoded in the same space.

Table 5.1: Results of the preliminary study on using synthetic videos generated by different text-to-video models. Increases (↑) and decreases (↓) are measured relative to the model fine-tuned without synthetic videos (*i.e.*, NONE). * indicates our reproduced results using the mPLUG-Owl 7B model checkpoint released in the original VideoCon repository.

TEXT-TO-VIDEO GENERATOR	VIDEO-LANGUAGE ENTAILMENT (VIDEOCON)			TEXT-TO-VIDEO RETRIEVAL		VIDEO QA
	LLM	Human	Human-Hard	SSv2-Temporal	SSv2-Events	ATP-Hard
NONE* [7]	**88.39**	77.16	74.76	13.00	10.37	35.46
COGVIDEOX [138]	83.93 (↓ 4.46)	76.89 (↓ 0.27)	75.10 (↑ 0.34)	11.76 (↓ 1.24)	8.79 (↓ 1.58)	35.30 (↓ 0.16)
LAVIE [121]	85.26 (↓ 3.13)	76.96 (↓ 0.20)	74.63 (↓ 0.13)	**14.26** (↑ 1.26)	**10.80** (↑ 0.43)	34.82 (↓ 0.64)
VIDEOCRAFTER2 [14]	85.82 (↓ 2.57)	**77.33** (↑ 0.17)	**75.15** (↑ 0.39)	13.80 (↑ 0.80)	10.27 (↓ 0.10)	**35.79** (↑ 0.33)

Table 5.2: Average results of the preliminary study on using synthetic videos generated by different text-to-video models, for each type of misalignment. Increases (↑) and decreases (↓) are measured relative to the model fine-tuned without synthetic videos (*i.e.*, NONE). * indicates our reproduced results using the mPLUG-Owl 7B model checkpoint released in the original VideoCon repository.

MISALIGNMENT	VIDEO-LANGUAGE ENTAILMENT (VIDEOCON)			TEXT-TO-VIDEO RETRIEVAL		VIDEO QA
	LLM	Human	Human-Hard	SSv2-Temporal	SSv2-Events	ATP-Hard
NONE* [7]	**88.39**	77.16	74.76	13.00	10.37	35.46
ACTION	86.10 (↓ 2.29)	77.43 (↑ 0.27)	74.83 (↑ 0.07)	**15.04** (↑ 2.04)	10.66 (↑ 0.29)	36.28 (↑ 0.82)
ATTRIBUTE	86.51 (↓ 1.88)	77.61 (↑ 0.45)	**75.50** (↑ 0.74)	13.67 (↑ 0.67)	11.47 (↑ 1.10)	35.25 (↓ 0.21)
COUNT	86.10 (↓ 2.29)	**77.66** (↑ 0.50)	75.27 (↑ 0.51)	14.27 (↑ 1.27)	10.97 (↑ 0.60)	36.16 (↑ 0.70)
FLIP	85.69 (↓ 2.70)	76.04 (↓ 1.12)	73.53 (↓ 1.23)	14.94 (↑ 1.94)	10.73 (↑ 0.36)	36.06 (↑ 0.60)
HALLUCINATION	85.46 (↓ 2.93)	76.55 (↓ 0.61)	74.77 (↑ 0.01)	13.89 (↑ 0.89)	10.14 (↓ 0.23)	**36.37** (↑ 0.91)
OBJECT	**86.28** (↓ 2.11)	77.36 (↑ 0.20)	74.15 (↓ 0.61)	14.54 (↑ 1.54)	**11.54** (↑ 1.17)	35.48 (↑ 0.02)
RELATION	86.22 (↓ 2.17)	77.46 (↑ 0.30)	74.59 (↓ 0.17)	14.99 (↑ 1.99)	11.38 (↑ 1.01)	34.65 (↓ 0.81)

concatenation, and $f_{\texttt{dec}}^{[\mathbf{w}]}$ the likelihood of the word $\mathbf{w} \in \mathcal{W}$ from the decoder's output.

VLA learning. Usually, the parameters θ of f are updated using a dataset D of n video-language triplets $D = \{(\mathbf{V}_1, t_1^+, t_1^-), \cdots, (\mathbf{V}_n, t_n^+, t_n^-)\}$, where t_i^+ and t_i^- are the positive and negative text captions for the video $\mathbf{V}_i$, *i.e.*, captions that respectively represent (t_i^+) and do not represent (t_i^-) the video content. Exploiting the probability distribution, output of $f_{\texttt{dec}}$, we can define the following objective:

$$\mathcal{L}_{\texttt{real}} = -\sum_{i=1}^{n} \log f(\mathbf{V}_i, t_i^+) + \log\left(1 - f(\mathbf{V}_i, t_i^-)\right). \tag{5.2}$$

This loss function forces f to sample `Yes` with a higher probability if the text represents the video and `No` otherwise.

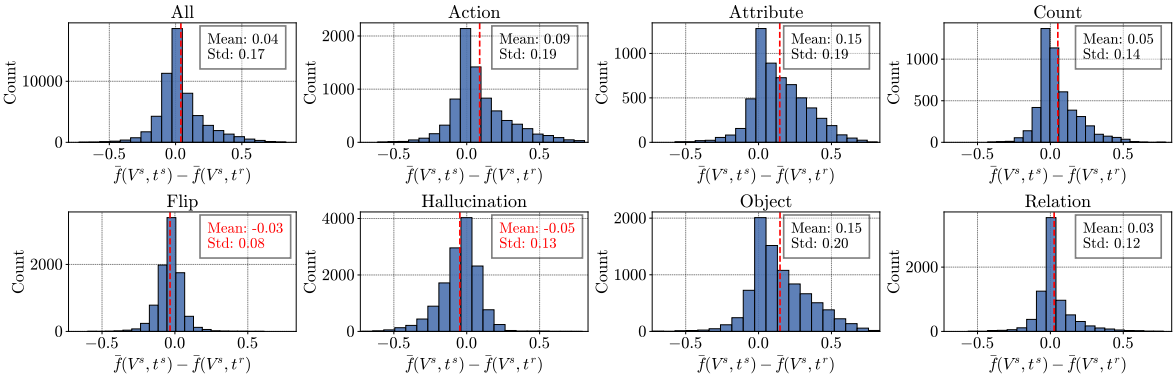


Figure 5.2: Distribution of the difference between $\bar{f}(\mathbf{V}^s, t^s)$ and $\bar{f}(\mathbf{V}^s, t^r)$ for each misalignment type, averaged over three text-to-video generators. Misalignment types that result in negative differences (*i.e.*, Flip and Hallucination) are highlighted in red. Best viewed in color.

5.4 Can Synthetic Videos help VLA?

The main loss function in VLA learning, as expressed in Eq. (5.2), considers as negative only textual inputs for a given "anchor" video. For each positive caption t_i^+, there is no negative video example associated with t_i^-. As such, linguistic biases might be induced in the LMM because some concepts appear only as textual negatives. Thus, we wonder: *Can generated videos of negative captions help learning a VLA function?* To answer this question, we consider different text-to-video generator models and use them to generate synthetic videos associated to negative captions.

VLA learning with generated videos. Formally, a text-to-video generator G maps natural language expressions in $\mathcal{T}$ and noise in the space $\mathcal{N}$ to videos, *i.e.*, $G : \mathcal{T} \times \mathcal{N} \rightarrow \mathcal{V}$. For simplicity, we define $t^r = t^+$ (*i.e.*, text positively associated with the *real* video) and $t^s = t^-$ (*i.e.*, negative text for the real video, positively associated with the *synthetic* one). We propose to use the generator to define an objective over the dataset D:

$$\mathcal{L}_{\texttt{syn}} = -\sum_{i=1}^{n} \log f\left(\mathbf{V}_i^s, t_i^s\right) + \log\left(1 - f\left(\mathbf{V}_i^s, t_i^r\right)\right) \tag{5.3}$$

where $\mathbf{V}_i^s = G(t_i^s, \eta_i)$, with $\eta_i \sim \mathcal{N}$ being the sampled noise. The negative text t_i^s is the input text to the generator, thus serving as the positive for the synthetic video, while the positive text t_i^r for the real video $\mathbf{V}_i^r$ serves as negative for the generated video.

Experimental analysis. To better understand the potential of synthetic videos, we first conduct a preliminary experimental analysis and leverage three state-of-the-art open-source video generators, *i.e.*, CogVideoX [138], LaVie [121], and VideoCrafter2

[14], to generate synthetic videos for each negative caption in the VideoCon dataset [7]. We augment the dataset with these generated videos and fine-tune an LMM, mPLUG-Owl 7B [140], using the objective functions defined in Eq. (5.2) and Eq. (5.3) for real and synthetic videos, respectively. We measure the performance with the VLA scores estimated from Eq. (5.1), following the established evaluation protocol [7] across multiple tasks and datasets. Specifically, we consider video-language entailment on the VideoCon dataset, text-to-video retrieval on SSv2-Temporal [96] and SSv2-Events [4] datasets, and video question answering (VQA) on the ATP-Hard dataset [10]. The evaluation metrics include the area under the receiver operating characteristic curve (AUC ROC) on video-language entailment, mean average precision (mAP) on text-to-video retrieval, and accuracy on VQA.

We report the results in Tab. 5.1, including baseline performance without synthetic video data (NONE). From the table, it is clear that synthetic videos harm the performance on the task closest to the training set (*i.e.*, average drop higher than 3% AUC on VideoCon LLM). One core reason for this drop is the distribution of the negatives being more similar to the one of the training set. Thus performance may decrease when a model sees them as positives. On the other hand, the results on downstream tasks suggest that synthetic videos hold promise. For instance, VideoCrafter2 improves the result of the baseline in 4/6 settings, while LaVie boosts performance on SSv2-Temporal (*i.e.*, +1.26 mAP). However, even with state-of-the-art video generators, not all of them guarantee improvements, and no single generator consistently outperforms the others across the tested downstream tasks. This can be seen with CogVideoX, which provides slight improvements on one of the tasks (*i.e.*, entailment on Human-Hard) while harming the representations on the others (*e.g.*, -1.58 mAP on SSv2-Events).

Are some negative captions challenging? The VideoCon dataset [7] includes negative captions that differ from positive ones by specific types of misalignment, including modifications in actions, attributes, objects, relations, counts, event orders (flipping), and adding hallucinations. Therefore, we also analyze whether certain types of captions are particularly challenging for the generators to produce corresponding videos. We achieve this by fine-tuning mPLUG-Owl 7B with synthetic videos specific to each misalignment type. The results averaged over the three video generators are reported in Tab. 5.2. As shown in the table, different types of misalignment have different impacts on the downstream tasks. For instance, ACTION is the misalignment that results in the largest overall improvement (*e.g.*, +2.04 mAP on SSv2-Temporal, +0.82% accuracy on ATP-hard), while FLIP and HALLUCINATIONS misalignments lead to some severe decrease on the VideoCon benchmarks (*e.g.*, -1.12 and -0.61 respectively on VideoCon

Human).

We hypothesize that such a performance drop is due to the alignment quality of synthetic videos. To evaluate our hypothesis, we measure the quality of a synthetic video $\mathbf{V}^s$, generated from a caption t^s, as a negative example for the caption t^r as $\bar{f}(\mathbf{V}^s, t^s) - \bar{f}(\mathbf{V}^s, t^r)$, where $\bar{f}(\mathbf{V}, t)$ is computed using an ensemble of VQAScores [61], obtained by averaging the scores from three VQA models [64, 22, 61], *i.e.*, their average likelihood of answering `Yes` to the question: `Does this figure show [`t`]?` across four uniformly sampled frames from the video. The higher the difference between the two scores, the higher the similarity of the synthetic video to its caption t^s than its negative t^r and, intuitively, the more relevant the synthetic video for the VLA learning process. Fig. 5.2 shows the distribution of this difference for different types of misalignments. Notably, only Flip and Hallucinations misalignments yield mean differences that are below zero (*i.e.*, -0.03 and -0.05, respectively), while the others are above (*e.g.*, 0.09 Actions, 0.15 Attribute and Object). This indicates that synthetic videos corresponding to Flip and Hallucinations negative captions are not well aligned, which worsens the VLA learning process, as confirmed in Tab. 5.2.

Finding summary. Our preliminary analysis reveals that:

(i) Synthetic videos show potential for enhancing VLA, though improvements are not consistent among different generators.

(ii) Different types of misalignment influence various downstream tasks in distinct ways.

(iii) Synthetic videos that align closer to the positive captions of real videos rather than the negative captions result in poor training samples, which negatively impact learning.

5.5 SynViTA

As shown in the previous section, some generated videos are closer to real captions than their target ones (Fig. 5.2). This contradicts a key assumption of Eq. (5.3): that generated videos fully represent the content described by their input caption t^s. This often happens due to semantic inconsistency, *i.e.*, generated videos fail to follow the semantic instruction given by the input text [53, 8]. Such synthetic videos introduce noisy supervision signals, leading to degraded VLA performance (Tab. 5.2) [70]. Moreover, even semantically consistent synthetic videos may be distinguished using visual differences

Figure 5.3: Overview of **SynViTA**. Given a real video $\mathbf{V}^r$ with its description t^r and a negative caption t^s (generated by an LLM), we first generate a synthetic video $\mathbf{V}^s$ based on t^s. We weigh the importance of each video using the scoring criterion ϕ. We also find the shared semantic between t^r and t^s using the longest common subsequence, obtaining t'. We train f_θ to respond with `Yes` if the input video matches its description and `No` otherwise. Additionally, we encourage the model to focus on the semantic difference between real and synthetic videos, instead of the appearance difference, using their shared semantic (*i.e.*, t').

(*e.g.*, artifacts [127, 89]) rather than intended semantic ones. In this chapter, we propose a model-agnostic method to better use Synthetic videos for Video-Text Alignment (SynViTA), modeling them via two strategies: alignment-based weighting and semantic consistency regularization (see Fig. 5.3).

Alignment-based weighting. To mitigate the impact of harmful synthetic videos and maximize the impact of valuable ones, we weigh the importance of each video based on a scoring criterion ϕ. Given a synthetic video $\mathbf{V}^s$, its corresponding caption t^s and the real counterpart t^r, ϕ maps them to a binary score in $[0, 1]$ depending on their level of alignment, *i.e.*, $\phi : \mathcal{V} \times \mathcal{T} \times \mathcal{T} \rightarrow [0, 1]$. A simple choice for ϕ is to directly use the alignment scores given by our model f. In this case, $\phi(\mathbf{V}^s, t^s, t^r) = f(\mathbf{V}^s, t^s)$. However, this might ignore the cases where, erroneously, $f(\mathbf{V}^s, t^r) > f(\mathbf{V}^s, t^s)$, *i.e.*, the generated video is closer to the real caption t^r than to the target one t^s. This phenomenon frequently happens (*i.e.*, Fig. 5.2), due to *e.g.*, wrong attribute/action binding [43]. For instance, if we ask the model to generate *a horse watching a person running*, it may erroneously generate *a person watching a horse running*, swapping the two actions. As shown in Sec. 5.4, this type of mistakes harms the learning of f and its capability to distinguish fine-grained details.

Thus, we define ϕ to account for how well the generated video $\mathbf{V}^s_i$ represents t^s_i in comparison to its real, negative, counterpart t^r_i, defining the weight for a synthetic video as:

$$\omega^\phi_i = \phi(\mathbf{V}^s_i, t^s_i, t^r_i) = \max(0, \bar{f}(\mathbf{V}^s_i, t^s_i) - \bar{f}(\mathbf{V}^s_i, t^r_i)) \tag{5.4}$$

where $\bar{f}$ is an ensemble of VQAScores, as in Sec. 5.4. Note that the more the video is aligned with the target text w.r.t. its negative one, the higher its weight from Eq. (5.4).

Given this scoring criterion, we define a loss function on synthetic videos, where ϕ acts as a dynamic weight giving higher relevance to videos better aligned with text:

$$\mathcal{L}^{\phi}_{\texttt{syn}} = -\sum_{i=1}^{n} \omega_i^{\phi} \cdot \left(\log f\left(\mathbf{V}_i^s, t_i^s\right) + \log\left(1 - f\left(\mathbf{V}_i^s, t_i^r\right)\right)\right). \tag{5.5}$$

Semantic consistency regularization. A positive aspect of having synthetic videos for negative textual inputs is that we can make the model focus on the semantic changes between videos rather than those in appearance. Suppose we are given a text t^r, its negative version t^s, a real video $\mathbf{V}^r$, and its generated negative version $\mathbf{V}^s$. If the difference between t^r and t^s is fine-grained, it will focus on specific properties of the video (*e.g.*, action, temporal order, etc.). This implies that the two texts share most of the content *but* for those fine-grained characteristics. We can thus define a text t', whose semantic is shared between t^r and t^s, thus not being specific to $\mathbf{V}^r$ or $\mathbf{V}^s$. We achieve this by finding the intersection between the two texts via longest-common subsequence [39], *i.e.*, $t' = \text{LCS}(t^r, t^s)$[2].

Note that t' has a specific property: given the real (synthetic) video, t' is a less accurate description than the original caption t^r (t^s), but a better one than the negative t^s (t^r). Ideally, our model should capture this relationship, modeling t' as semantically closer to the video than its negative caption, but farther w.r.t. its positive. We can achieve this by computing a triplet loss, defined as:

$$\begin{aligned}\mathcal{L}^{\phi}_{\texttt{scr}} = \sum_{i=1}^{n} \sum_{z\in\{s,r\}} \omega_i^{\phi} \cdot \big(&\max\left(0, \gamma + f(\mathbf{V}_i^z, t_i') - f(\mathbf{V}_i^z, t_i^z)\right) \\ &+ \max\left(0, \gamma + f(\mathbf{V}_i^z, t_i^{\bar{z}}) - f(\mathbf{V}_i^z, t_i')\right)\big).\end{aligned} \tag{5.6}$$

where the margin term γ enforces the desired separation between the alignment probabilities, and when $z = r$, $\bar{z} = s$ and vice versa. The first term promotes better alignment of the positive caption w.r.t. the generic caption t' and the second promotes better alignment of the latter w.r.t. the negative caption. This encourages f to focus on the semantic differences between the two visual inputs, ignoring their differences in appearance due to the synth-to-real gap.

Full objective. Considering all learning objectives together, we obtain the following

[2]Note that, in practice, t' is not implemented via token removal but via attention-level masking.

final function:

$$\mathcal{L} = \mathcal{L}_{\texttt{real}} + \mathcal{L}^{\phi}_{\texttt{syn}} + \lambda_{\texttt{scr}} \cdot \mathcal{L}^{\phi}_{\texttt{scr}}. \tag{5.7}$$

where $\lambda_{\texttt{scr}}$ is a hyperparameter that regulates the losses. We use Eq. (5.7) to learn the set θ of parameters in f. Remarkably, our framework has only two hyperparameters, *i.e.*, the margin γ of $\mathcal{L}^{\phi}_{\texttt{scr}}$ and the weight $\lambda_{\texttt{scr}}$ of $\mathcal{L}^{\phi}_{\texttt{scr}}$.

5.6 Experiments

In this section, we describe our experimental protocol and present the comparison w.r.t. the state of the art (Sec. 5.6.1). We then perform a detailed study on SynViTA (Sec. 5.6.2). Finally, we provide a qualitative analysis of the synthetic data quality and its impact on the training signal (Sec. 5.6.3).

Datasets. For training SynViTA, we use the VideoCon dataset [7], which includes temporally-challenging video-text triplets from MSR-VTT [135], VATEX [120], and TEMPO [37] for two tasks: *Video-Language Entailment (VLE)* and *Natural Language Explanation (NLE)*. In VLE, the model outputs a score of 1 if the video entails the description and 0 otherwise, while in NLE, it outputs the explanation of the differences between a video and a caption. For each negative caption in the VideoCon VLE training set, we generate a corresponding video using three text-to-video models: CogVideoX [138], LaVie [121], and VideoCrafter2 [14].

For evaluation, we use the VideoCon VLE test sets: (i) **VideoCon (LLM)**, with 27K video-text pairs from the same source datasets; (ii) **VideoCon (Human)**, with 570 pairs from ActivityNet [11] and human annotated negative captions; and (iii) **VideoCon (Human-Hard)**, a subset of 290 temporally challenging instances. Following [7], we also evaluate our model on various downstream tasks: (i) text-to-video retrieval with **SSv2-Temporal** [96], which includes 18 action classes, each with 12 videos (in total 216 videos), requiring temporal understanding; (ii) **SSv2-Events** [4], with 49 action classes, each with 12 videos, featuring multi-event actions; and (iii) video question answering on **ATP-Hard** [10], a subset of questions of NExT-QA [132] that require causal and temporal understanding of videos. We measure the performance using AUC for entailment, mAP for retrieval, and accuracy for VQA.

Implementation details. We implement SynViTA on two LMMs, mPLUG-Owl 7B [140] and Video-LLaVA [59], trained on 4 NVIDIA A100 GPUs. Both models share most of the hyperparameters with VideoCon [7] to ensure a fair comparison, and fine-tune the projection layers of the attention blocks of the LLM with low-rank adaptation (LoRA)

Table 5.3: Comparison of SynViTA with both discriminative and generative VLMs. For the video-language entailment task, we report AUC-ROC, for zero-shot text-to-video retrieval, we report mAP, and for video question-answering, we report accuracy. * indicates our reproduced results using the mPLUG-Owl 7B model checkpoint released in the original VideoCon repository.

	Video-language Entailment (VideoCon)			Text-to-Video Retrieval		Video QA
	LLM	Human	Human-Hard	SSv2-Temporal	SSv2-Events	ATP-Hard
VideoCLIP [134]	53.2	47.3	47.5	9.8	6.4	23.4
ImageBind (Video-Text) [30]	57.1	65.2	63.0	10.5	5.5	25.4
TACT [4]	-	-	-	-	7.8	27.6
VFC [81]	-	-	-	-	-	31.4
End-to-End VNLI [139]	67.0	72.4	65.0	14.6	10.4	39.0
mPLUG-Owl 7B [140]	57.24	67.02	64.39	11.08	6.75	37.96
Video-LLaVA [59]	62.98	70.37	65.99	11.64	7.11	38.56
VideoCon (mPLUG-Owl 7B)* [7]	**88.39**	77.16	74.76	13.00	10.37	35.46
VideoCon (Video-LLaVA)	85.86	80.09	75.74	19.77	10.01	38.76
SynViTA (mPLUG-Owl 7B)	86.45	77.48	74.54	17.32	**12.54**	37.31
SynViTA (Video-LLaVA)	85.43	**80.86**	**76.86**	**20.10**	11.21	**39.88**

[42], with $r = 32$, $\alpha = 32$, and dropout $= 0.05$. For both models, we set γ to 0.2, while $\lambda_{\texttt{scr}}$ to 10^{-2} for mPLUG-Owl 7B and 1.0 for Video-LLaVA.

Baselines. We compare SynViTA (mPLUG-Owl 7B) and SynViTA (Video-LLaVA) against two sets of models. The first set includes off-the-shelf VLMs such as VideoCLIP [134], ImageBind (Video-Text) [30], End-to-End VNLI [139], mPLUG-Owl 7B [140], and Video-LLaVA [59], as well as models fine-tuned for improved understanding of actions and event order, *i.e.*, VFC [81] and TACT [4]. The second set consists of models trained on video-text triplets from the VideoCon dataset, namely VideoCon (mPLUG-Owl 7B) and VideoCon (Video-LLaVA) [7].

5.6.1 Comparison with state of the art

Tab. 5.3 presents the results of our comparison on the VideoCon evaluation sets and the downstream tasks. Overall, our proposed method outperforms all previous baselines in five tasks out of six. For the entailment task, on the VideoCon Human dataset SynViTA (Video-LLaVA) improves its counterpart VideoCon (Video-LLaVA), trained without synthetic video-caption pairs, by 0.77%, and achieves a 1.12% improvement on its temporally challenging subset, Human-Hard. Similarly, SynViTA (mPLUG-Owl 7B) shows a 0.32% improvement on the VideoCon Human dataset. As expected from Sec. 5.3, on the VideoCon (LLM) test set, both SynViTA (Video-LLaVA) and SynViTA (mPLUG-Owl 7B) underperform compared to their counterparts without synthetic videos, due to the similar distribution of negatives w.r.t. those present in the training set. Thus, synthetic pairs harm the performance in this setting.

For text-to-video retrieval tasks, SynViTA (mPLUG-Owl 7B) outperforms VideoCon

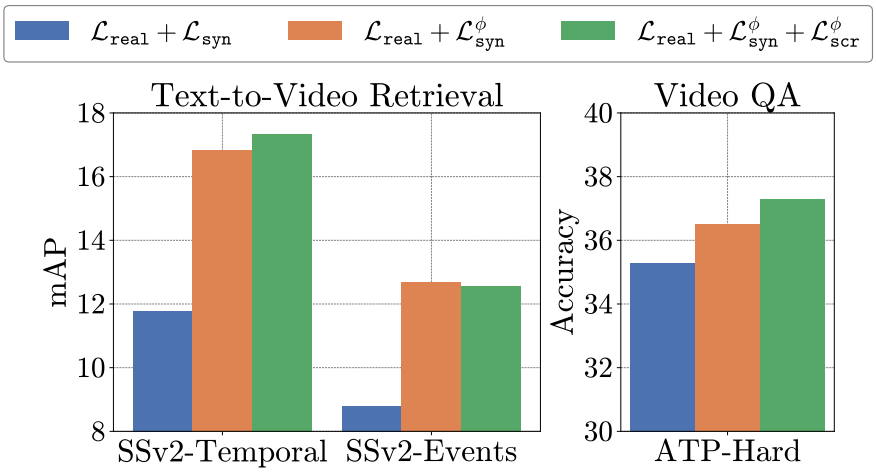


Figure 5.4: Ablation study on the proposed losses.

Table 5.4: Results of the ablation study on the weighting strategy for the synthetic videos in the objective function.

Alignment-based weighting strategy	Video-Language Entailment (VideoCon) LLM	Human	Human-Hard	Text-to-Video Retrieval SSv2-Temporal	SSv2-Events	Video QA ATP-Hard
1) Fixed weighting-1.00	83.95	76.91	**75.05**	12.54	8.48	36.23
2) $\bar{f}(\mathbf{V}^s, t^s)$	84.87	76.46	74.12	13.43	9.40	35.95
3) $\bar{f}(\mathbf{V}^s, t^s) \cdot (1 - \bar{f}(\mathbf{V}^s, t^r))$	85.57	76.79	74.11	15.52	10.74	36.06
4) $\mathbb{1}\left[\bar{f}(\mathbf{V}^s, t^s) > \bar{f}(\mathbf{V}^s, t^r)\right]$	84.88	76.79	73.17	14.17	10.38	37.15
5) $\max(0, \bar{f}(\mathbf{V}^s, t^s) - \bar{f}(\mathbf{V}^s, t^r))$	**86.45**	**77.48**	74.54	**17.32**	**12.54**	**37.31**

(mPLUG-Owl 7B) by 4.32% on SSv2-Temporal and 2.17% on SSv2-Events. Similarly, SynViTA (Video-LLaVA) shows improvements of 0.33% on SSv2-Temporal and 1.20% on SSv2-Events compared to VideoCon (Video-LLaVA). These results suggest that our model is model-agnostic and more effective at ranking similar but semantically different text descriptions than the baseline, which does not associate corresponding video data with negative captions. Finally, for the challenging video question-answering task on the ATP-Hard dataset, models fine-tuned with only textual negatives see performance drops or minimal improvement compared to their non-finetuned version. Despite this, SynViTA (mPLUG-Owl 7B) improves upon VideoCon (mPLUG-Owl 7B) by 1.85%, and SynViTA (Video-LLaVA) shows a 1.12% improvement over VideoCon (Video-LLaVA).

5.6.2 Ablation study

In this section, we analyze the components of SynViTA considering the mPLUG-Owl 7B version. We first examine the different parts of our learning objective. We then show the benefits of alignment-based weighting over fixed weights and of different alignment-based scoring criteria. Finally, we show the effect of using different text-to-video models for generating synthetic videos.

Table 5.5: Ablation study on varying the text-to-video model.

	Text-to-Video Generator				
	None	CogVideoX	LaVie	VideoCrafter2	All
LLM	**88.39**	86.45	86.45	86.43	85.82
Human	77.16	77.48	**77.51**	77.48	77.15
Human-Hard	**74.76**	74.54	74.73	74.74	73.79
SSv2-Temporal	13.00	**17.32**	15.98	15.47	14.06
SSv2-Events	10.37	**12.54**	12.36	11.72	10.90
ATP-Hard	35.46	**37.31**	36.55	36.50	36.44

Learning objectives. We first analyze the effectiveness of the two proposed components in our learning objective: the alignment-based loss function $\mathcal{L}^{\phi}_{\texttt{syn}}$ (Eq. (5.5)) and the semantic consistency regularization $\mathcal{L}^{\phi}_{\texttt{scr}}$ (Eq. (5.6)). As shown in Fig. 5.4, excluding $\mathcal{L}^{\phi}_{\texttt{syn}}$ leads to a drop in performance (blue vs. orange bar). Without this loss, the objective is solely the traditional language modeling loss. As a result, synthetic videos that are not aligned with their captions introduce a noisy training signal. Adding $\mathcal{L}^{\phi}_{\texttt{scr}}$ (green bar), further boosts the performance on 2/3 datasets, suggesting that the model better captures the video semantics. As our model is trained on triplets with single-event differences [7], $\mathcal{L}^{\phi}_{\texttt{scr}}$ is less effective for SSv2-Events, where captions involve multiple events. However, current open-source video generators struggle to generate multi-event videos.

Alignment-based weighting strategy. In this section, we evaluate our alignment-based weighting strategy (*i.e.*, Eq. (5.4)) against other alternatives, reporting the results in Tab. 5.4. As a reference, row (1) reports the results of a fixed weight (*i.e.*, 1) for all synthetic videos. Assigning weights based only on alignment with the target text (*i.e.*, $f(\mathbf{V}^s, t^s)$) improves performance on retrieval (*e.g.*, +0.92 mAP on SSv2-Events) but degrades performance on others (*e.g.*, on ATP-Hard, -0.28%), as it overlooks cases where synthetic videos align more with real captions. In row (3), we multiply the synthetic scores by the inverse similarity with the real counterpart (*i.e.*, $(1 - f(\mathbf{V}^s, t^r))$). Introducing the real captions into the score improves the results in various settings, especially on retrieval, achieving +2.98 mAP on SSv2-Temporal, and +2.26 mAP on SSv2-Events. As an alternative, row (4) considers weighing all synthetic videos as 1 if they are closer to their target caption than the real one. This strategy shows a general degradation w.r.t. the previous, except for ATP-Hard (+1.9%). This denotes that a soft-weighting scheme is still more effective as it accounts for different levels of semantic fidelity across videos. Our proposed strategy (row (5)) combines the advantages of

the two, enforcing that synthetic videos are truly negative examples, *i.e.*, being more similar to their caption than the original one of the real videos. This strategy obtains the highest results in almost all settings. For Video-LLaVA, we use (3) as it performs slightly better.

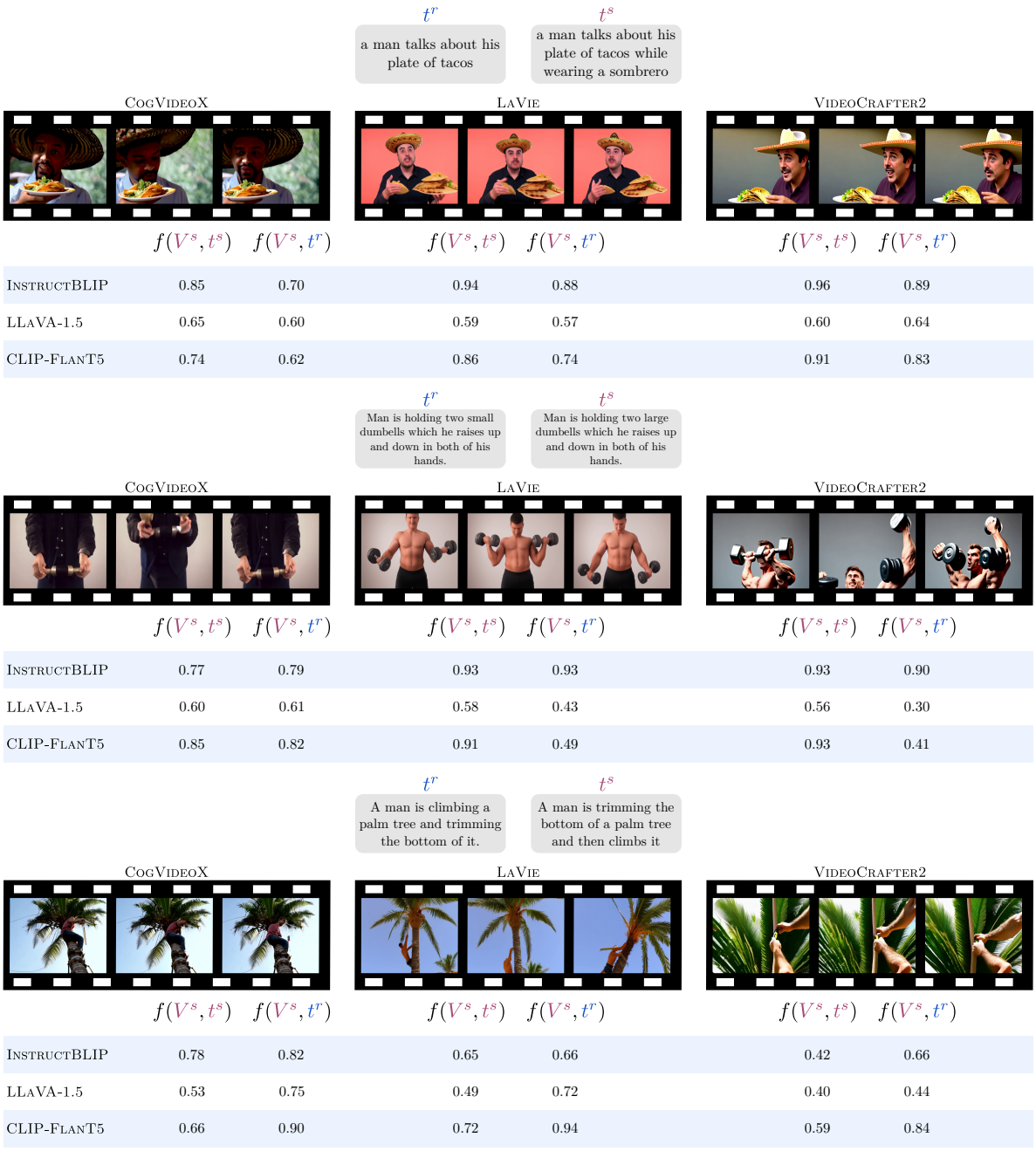
Text-to-video generators. We analyze this aspect by comparing three text-to-video generators when used with our method: CogVideoX [138], LaVie [121], and VideoCrafter2 [14]. As shown in Tab. 5.5, SynViTA (mPLUG-Owl 7B) fine-tuned on videos generated by CogVideoX outperforms the other alternatives across all downstream tasks and achieves comparable results on the video-language entailment task. While it can be challenging to determine *a-priori* the optimal generator for a downstream task, one possibility could be to generate videos from multiple generators and let the model filter them. Using all generated videos performs better than using none on the downstream tasks (*e.g.*, +1.06% on SSv2-Temporal), but underperforms CogVideoX (*e.g.*, -3.26% on SSv2-Temporal). This is likely due to the high synth-to-real video ratio, introducing a significant domain shift that requires careful handling. Nevertheless, we expect that the better the text-to-video models released, the more beneficial they will be for SynViTA.

5.6.3 Analysis of synthetic data quality and diversity

To understand how the quality and diversity of generated videos impact the training signal, Fig. 5.5 presents examples of synthetic data along with alignment scores assigned by InstructBLIP [22], LLaVA-1.5 [64], and CLIP-FlanT5 [61]. Specifically, we show the alignment of each synthetic video $\mathbf{V}^s$ with its corresponding synthetic caption t^s, denoted as $f(\mathbf{V}^s, t^s)$, and with its real counterpart t^r, denoted as $f(\mathbf{V}^s, t^r)$.

These examples illustrate that while models often produce semantically consistent videos (*e.g.*, the *sombrero* example), they can also fail to capture specific attributes like scale or complex event orders. For the caption *"Man is holding two large dumbells which he raises up and down in both of his hands."*, CogVideoX fails to depict the size of the dumbbells, leading to alignment scores that are lower than those of the real counterpart for two out of three models. Finally, for inherently nonsensical prompts or complex categories like *event order flip* (*e.g.*, the *palm tree* example), none of the generated videos achieve higher alignment scores with the input caption than with the real caption. This is likely due to the nonsensical nature of the LLM-generated prompt or artifacts introduced by the generators, such as VideoCrafter2 only rendering the top of the tree.

Overall, these cases confirm that synthetic data can introduce noise or bias, particularly when generators struggle with fine-grained semantic details or complex temporal



	$f(V^s, t^s)$	$f(V^s, t^r)$	$f(V^s, t^s)$	$f(V^s, t^r)$	$f(V^s, t^s)$	$f(V^s, t^r)$
InstructBLIP	0.85	0.70	0.94	0.88	0.96	0.89
LLaVA-1.5	0.65	0.60	0.59	0.57	0.60	0.64
CLIP-FlanT5	0.74	0.62	0.86	0.74	0.91	0.83

	$f(V^s, t^s)$	$f(V^s, t^r)$	$f(V^s, t^s)$	$f(V^s, t^r)$	$f(V^s, t^s)$	$f(V^s, t^r)$
InstructBLIP	0.77	0.79	0.93	0.93	0.93	0.90
LLaVA-1.5	0.60	0.61	0.58	0.43	0.56	0.30
CLIP-FlanT5	0.85	0.82	0.91	0.49	0.93	0.41

	$f(V^s, t^s)$	$f(V^s, t^r)$	$f(V^s, t^s)$	$f(V^s, t^r)$	$f(V^s, t^s)$	$f(V^s, t^r)$
InstructBLIP	0.78	0.82	0.65	0.66	0.42	0.66
LLaVA-1.5	0.53	0.75	0.49	0.72	0.40	0.44
CLIP-FlanT5	0.66	0.90	0.72	0.94	0.59	0.84

Figure 5.5: Examples of videos generated by three text-to-video models (*i.e.*, CogVideoX, LaVie, and VideoCrafter2) from LLM-generated negative captions, along with alignment scores assigned by different image-text alignment methods (*i.e.*, InstructBLIP, LLaVA-1.5, and CLIP-FlanT5). For each synthetic video $\mathbf{V}^s$ and alignment model, we show its alignment with the corresponding caption t^s, denoted as $f(\mathbf{V}^s, t^s)$, and with the real caption t^r, denoted as $f(\mathbf{V}^s, t^r)$.

dependencies. However, as hypothesized in Sec. 5.4, the negative mean alignment differences observed in Fig. 5.2 provide a reliable metric to identify these unreliable sam-

ples. By incorporating these scores into our alignment-based weighting (w_i^{ϕ}), SynViTA effectively down-weights or filters out lower-quality synthetic videos during training. This ensures the model learns from generated videos without being compromised by the limitations or hallucinations of current text-to-video generators.

5.7 Chapter Summary

In this chapter, we explored whether videos generated by text-to-video models can help learning a better video-language alignment (VLA) model. Our initial analysis shows that synthetic videos can boost performance on certain downstream tasks, but harm others. We attribute this to (i) semantic inconsistency, as synthetic videos may not follow the input text, and (ii) appearance bias, where the model focuses on visual differences in the videos rather than semantic differences. To address these limitations, we introduced SynViTA, the first VLA method exploiting synthetic videos. SynViTA includes an alignment-based sample weighting strategy to mitigate noisy video generations and a semantic consistency regularization to make the model focus on semantic, rather than visual, differences. SynViTA outperforms baselines that do not use synthetic videos across different LMMs on five out of six tasks, demonstrating its potential to improve VLA across diverse models.

Chapter 6

Discussion

This thesis presents a practical path toward language-guided video understanding systems that are better aligned with real-world deployment constraints by shifting from visual-only representations to vision-language representations and progressively relaxing assumptions about task-specific training and offline access during inference. In this final chapter, we discuss the implications of this approach, including applications (see Sec. 6.1), ethical considerations (see Sec. 6.2), limitations (see Sec. 6.3), future research directions (see Sec. 6.4), and final remarks (see Sec. 6.5).

6.1 Applications

This thesis introduces video understanding techniques designed to operate at inference time under relaxed assumptions on task-specific training data and offline access to complete video sequences. While the proposed methods are evaluated on standard research benchmarks, their design is motivated by practical deployment constraints and enables a range of real-world applications.

The design principles explored in this thesis (training-free inference and online operation over foundation models) make video understanding systems suitable for real-time surveillance and safety monitoring scenarios. In these settings, systems must react promptly to anomalous events, often without access to task-specific training data or the ability to process entire video streams offline. The contributions presented in Chapters 3 and 4 offer a path toward practical use of such systems in privacy-sensitive or highly dynamic environments where traditional supervised training is impractical.

The proposed techniques are also relevant to augmented and extended reality (AR/XR) applications. In particular, the online video step grounding framework intro-

duced in Chapter 4 can support interactive assistants that provide real-time, step-by-step guidance for complex procedural tasks, such as industrial maintenance or medical workflows. By combining Bayesian filtering with Large Multimodal Models (LMMs), the system updates its step predictions online as new visual evidence becomes available, maintaining consistency with previously inferred steps during task execution.

Finally, the research on synthetic data in Chapter 5 has implications for applications that require reliable temporal understanding over video, such as fine-grained video retrieval and temporal localization. By leveraging text-to-video generation, this approach provides a scalable mechanism for improving the semantic consistency of estimates produced by LMMs, while reducing reliance on costly human annotation.

6.2 Ethical Considerations

The methodologies presented in this thesis improve the accessibility and scalability of video understanding systems, but they also raise important ethical considerations that must be addressed before real-world deployment.

Video anomaly detection systems, such as **LAVAD** (Chapter 3), are typically applied in safety-critical contexts, including surveillance and monitoring for public or private use. While this work focuses on the technical feasibility of training-free anomaly detection using LLMs, deployment in such settings requires careful analysis of model behavior. In particular, LLM-based systems may inherit biases from pre-training data and may produce decisions that are difficult to interpret or justify. Understanding these decision behaviors and improving the transparency of inference with language models are necessary steps before such systems can be responsibly deployed.

Similarly, the **BAGLM** framework (Chapter 4) enables online understanding of procedural video content and can provide real-time feedback in applications such as remote learning, assisted rehabilitation, or domestic tasks. While these capabilities can support assistive technologies in data-scarce settings, reliance on large pre-trained multimodal models introduces the risk of uneven performance across domains, cultures, or user populations. Future work must explore strategies to mitigate these biases and ensure consistent behavior across diverse scenarios.

More broadly, reducing the barriers to deploying video understanding systems increases their potential for dual use. Techniques designed for safety monitoring or procedural assistance could also be misused for pervasive or intrusive surveillance if deployed without appropriate safeguards. In addition, the use of synthetic data in **SYNVITA** (Chapter 5) relies on generative models, which may produce misleading

content or be misused. Such synthetic data may influence the behavior of language models fine-tuned on it, amplifying biases, reinforcing spurious correlations, or introducing unintended failure modes. Therefore, progress in deployable video understanding should be accompanied by the development of automated detection techniques and robust governance frameworks to protect privacy and ensure content integrity.

6.3 Limitations

Despite the progress toward video understanding systems that are better aligned with real-world deployment constraints presented in this thesis, several limitations remain.

A first limitation is the performance gap between training-free methods and fully supervised systems. While approaches such as **LAVAD** (Chapter 3) and **BaGLM** (Chapter 4) outperform unsupervised baselines, they generally underperform compared to models fine-tuned using task-specific annotated data. This gap is closely tied to the quality of the underlying foundation models. In **LAVAD**, the accuracy of anomaly detection is constrained by the quality of video captioning models and the LLM's ability to interpret textual descriptions into anomaly scores. In **BaGLM**, online step grounding depends on the reliability of estimates produced by language models when conditioned on visual and linguistic cues, making it sensitive to errors or inconsistencies in these estimates.

A second limitation is computational cost. Training-free inference over LMMs remains expensive and typically requires high-end hardware. This limits deployment on edge-constrained platforms such as wearable devices or low-power cameras, which are common targets for real-time assistance and monitoring. Therefore, while these techniques are conceptually online and do not require task-specific training, their practical application is currently restricted to environments with sufficient computational resources.

A third limitation is long-range temporal understanding. While **SynViTA** (Chapter 5) leverages synthetic supervision to improve temporal understanding, it relies on text-to-video generators that produce only short clips. This restriction can create a synth-to-real domain gap, limiting the temporal complexity of the generated videos. As a result, synthetic data can only partially mitigate errors arising from long-range temporal dependencies, leaving long-horizon understanding an open challenge.

A final limitation is the reliance on prompting as the primary interface for guiding model behavior. While prompting is effective in practice, it remains difficult to optimize systematically. For example, in **BaGLM**, task progress is inferred through LMM queries rather than explicit duration or speed priors, which can lead to biased estimates.

6.4 Future Work

A key direction for future work is the development of efficient video understanding techniques that facilitate deployment on edge-constrained platforms, such as wearable devices and low-power cameras. While these platforms are natural targets for real-time assistance and monitoring, they currently cannot sustain the computational demands of large-scale multimodal models.

Practical deployment will therefore require both smaller, more efficient LMMs and inference strategies that reduce computational load. One promising approach is to reduce the number of visual tokens processed by the model. This can be achieved in two complementary ways. First, by removing redundancy at the frame level and filtering out visually uninformative frames rather than relying on fixed-rate frame sampling, which is particularly effective in surveillance and procedural videos with high temporal redundancy. Second, by selectively using only the tokens that are semantically relevant to the task or user query.

Together, these strategies reduce the number of tokens processed by transformer-based architectures and make better use of the limited context of the LLM. This directly addresses the quadratic cost of attention and makes training-free, online video understanding more feasible on resource-constrained, real-world devices. Importantly, these techniques should remain computationally lightweight and avoid introducing additional pre-processing stages that must run sequentially before inference, so as not to compromise interactive latency or the time to first token experienced by the user.

A related direction worth exploring is compressed-domain video understanding, where videos are processed directly in their compressed form (*e.g.*, motion vectors and residuals) rather than decoded into dense frame sequences. A key open challenge is how such compressed representations can be aligned with the language space of LLMs to enable reasoning over encoded video content through natural language prompts at inference time. Addressing this could reduce decoding overhead and better align with deployment scenarios involving continuous video streams and resource-constrained devices.

6.5 Conclusions

This thesis advances the state of the art in video understanding systems by leveraging language to relax assumptions that traditionally limit their real-world applicability, better aligning them with deployment constraints. Across four research contributions, we progressively shift from visual-only to vision-language representations, remove the

need for task-specific training data, and eliminate reliance on offline access to complete video sequences. Finally, we study how synthetic video data can be leveraged to improve the reliability of estimates produced by language models over video, a key requirement for training-free and online video understanding.

Chapter 2 introduced **AnomalyCLIP**, showing that the geometric structure of vision-language embeddings can be adapted with minimal video-level supervision to jointly detect and classify anomalous events. Chapter 3 extended this direction with **LAVAD**, demonstrating that LLMs can perform fully training-free video anomaly detection. To enable online inference over streaming video, Chapter 4 proposed **BaGLM**, which combines Bayesian filtering with LMMs to ground procedural steps in real time without access to future frames. Finally, recognizing the reliability of language model estimates over video as a shared requirement across the preceding approaches, Chapter 5 introduced **SynViTA** and explored whether synthetic videos generated by text-to-video models can improve their semantic consistency without human annotation.

Collectively, these contributions provide a roadmap for scalable video understanding systems that are better aligned with real-world deployment constraints. They show that expressing task-specific decision logic at inference time through language models can approach or surpass traditional training-based methods, highlighting the potential for vision systems that operate with minimal supervision, online, and under realistic deployment conditions. While challenges remain, particularly in computational efficiency and long-range temporal reasoning, the results of this thesis establish that training-free, online video understanding is both feasible and a promising direction for future research and real-world applications.

Bibliography

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv*, 2023.

[2] Triantafyllos Afouras, Effrosyni Mavroudi, Tushar Nagarajan, Huiyu Wang, and Lorenzo Torresani. Ht-step: Aligning instructional articles with how-to videos. *NeurIPS*, 2023.

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 2022.

[4] Piyush Bagad, Makarand Tapaswi, and Cees GM Snoek. Test of time: Instilling video-language models with a sense of time. In *CVPR*, 2023.

[5] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv*, 2025.

[6] Shuai Bai, Zhiqun He, Yu Lei, Wei Wu, Chengkai Zhu, Ming Sun, and Junjie Yan. Traffic anomaly detection via perspective map based on spatial-temporal information matrix. In *CVPRW*, 2019.

[7] Hritik Bansal, Yonatan Bitton, Idan Szpektor, Kai-Wei Chang, and Aditya Grover. Videocon: Robust video-language alignment via contrast captions. In *CVPR*, 2024.

[8] Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. *arXiv*, 2024.

[9] Qianyue Bao, Fang Liu, Yang Liu, Licheng Jiao, Xu Liu, and Lingling Li. Hierarchical scene normality-binding modeling for anomaly detection in surveillance videos. In *ACM Multimedia*, 2022.

[10] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the "video" in video-language understanding. In *CVPR*, 2022.

[11] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[12] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[13] Brian Chen, Andrew Rouditchenko, Kevin Duarte, Hilde Kuehne, Samuel Thomas, Angie Boggust, Rameswar Panda, Brian Kingsbury, Rogerio Feris, David Harwath, et al. Multimodal clustering networks for self-supervised learning from unlabeled videos. In *ICCV*, 2021.

[14] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *CVPR*, 2024.

[15] Yingxian Chen, Zhengzhe Liu, Baoheng Zhang, Wilton Fok, Xiaojuan Qi, and Yik-Chung Wu. Mgfn: Magnitude-contrastive glance-and-focus network for weakly-supervised video anomaly detection. *arXiv*, 2022.

[16] Yingxian Chen, Zhengzhe Liu, Baoheng Zhang, Wilton Fok, Xiaojuan Qi, and Yik-Chung Wu. Mgfn: Magnitude-contrastive glance-and-focus network for weakly-supervised video anomaly detection. In *AAAI*, 2023.

[17] Yuxiao Chen, Kai Li, Wentao Bao, Deep Patel, Yu Kong, Martin Renqiang Min, and Dimitris N Metaxas. Learning to localize actions in instructional videos with llm-based multi-pathway text-video alignment. In *ECCV*, 2024.

[18] Zhe Chen. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 2003.

[19] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance

boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv*, 2024.

[20] Sanjoy Chowdhury, Sayan Nag, and Dinesh Manocha. Apollo: Unified adapter and prompt learning for vision language models. In *EMNLP*, 2023.

[21] Daniel Cores, Michael Dorkenwald, Manuel Mucientes, Cees GM Snoek, and Yuki M Asano. Tvbench: Redesigning video-language evaluation. *arXiv*, 2024.

[22] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Albert Li, Pascale Fung, and Steven CH Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv*, 2023.

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[24] Xuefeng Du, Yiyou Sun, Jerry Zhu, and Yixuan Li. Dream the impossible: Outlier imagination with diffusion models. *NeurIPS*, 2024.

[25] Amine Elhafsi, Rohan Sinha, Christopher Agia, Edward Schmerling, Issa AD Nesnas, and Marco Pavone. Semantic anomaly detection with large language models. *Autonomous Robots*, 2023.

[26] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. In *arXiv*, 2021.

[27] Hao Fei, Shengqiong Wu, Meishan Zhang, Min Zhang, Tat-Seng Chua, and Shuicheng Yan. Enhancing video-language representations with structural spatio-temporal alignment. *TPAMI*, 2024.

[28] Jia-Chang Feng, Fa-Ting Hong, and Wei-Shi Zheng. Mist: Multiple instance self-training framework for video anomaly detection. In *CVPR*, 2021.

[29] Xinyang Feng, Dongjin Song, Yuncong Chen, Zhengzhang Chen, Jingchao Ni, and Haifeng Chen. Convolutional transformer based dual discriminator generative adversarial networks for video anomaly detection. In *ACM Multimedia*, 2021.

[30] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023.

[31] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv*, 2024.

[32] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

[33] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Anomalygpt: Detecting industrial anomalies using large vision-language models. *arXiv*, 2023.

[34] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *CVPR*, 2022.

[35] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *CVPR*, 2016.

[36] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv*, 2022.

[37] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. In *EMNLP*, 2018.

[38] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

[39] Daniel S Hirschberg. Algorithms for the longest common subsequence problem. *JACM*, 1977.

[40] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv*, 2019.

[41] Minh Hoai and Fernando De la Torre. Max-margin early event detectors. *IJCV*, 2014.

[42] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2021.

[43] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *NeurIPS*, 2023.

[44] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[45] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[46] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv*, 2023.

[47] Runyu Jiao, Yi Wan, Fabio Poiesi, and Yiming Wang. Survey on video anomaly detection in dynamic scenes with moving cameras. *Artificial Intelligence Review*, 2023.

[48] Hyekang Kevin Joo, Khoa Vo, Kashu Yamazaki, and Ngan Le. Clip-tsa: Clip-assisted temporal self-attention for weakly-supervised video anomaly detection. In *ICIP*, 2023.

[49] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv*, 2017.

[50] Jaehyun Kim, Seongwook Yoon, Taehyeon Choi, and Sanghoon Sull. Unsupervised video anomaly detection based on similarity with predefined text descriptions. *Sensors*, 2023.

[51] Dohwan Ko, Joonmyung Choi, Juyeon Ko, Shinyeong Noh, Kyoung-Woon On, Eun-Sol Kim, and Hyunwoo J Kim. Video-text representation learning via differentiable weak temporal alignment. In *CVPR*, 2022.

[52] Jie Lei, Tamara Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. In *ACL*, 2023.

[53] Baiqi Li, Zhiqiu Lin, Deepak Pathak, Jiayao Li, Yixin Fei, Kewen Wu, Xide Xia, Pengchuan Zhang, Graham Neubig, and Deva Ramanan. Evaluating and improving compositional text-to-visual generation. In *CVPRW*, 2024.

[54] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025.

[55] Guoqiu Li, Guanxiong Cai, Xingyu Zeng, and Rui Zhao. Scale-aware spatio-temporal relation learning for video anomaly detection. In *ECCV*, 2022.

[56] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

[57] Shuo Li, Fang Liu, and Licheng Jiao. Self-training multi-sequence learning with transformer for weakly supervised video anomaly detection. In *AAAI*, 2022.

[58] Zeqian Li, Qirui Chen, Tengda Han, Ya Zhang, Yanfeng Wang, and Weidi Xie. Multi-sentence grounding for long-term instructional video. In *ECCV*, 2024.

[59] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv*, 2023.

[60] Zhiqiu Lin, Xinyue Chen, Deepak Pathak, Pengchuan Zhang, and Deva Ramanan. Revisiting the role of language priors in vision-language models. *arXiv*, 2023.

[61] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *ECCV*. Springer, 2024.

[62] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation-based anomaly detection. *ACM TKDD*, 2012.

[63] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv*, 2023.

[64] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

[65] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *CVPR*, 2018.

[66] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? *arXiv*, 2024.

[67] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, 2022.

[68] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *ICCV*, 2021.

[69] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *ICCV*, 2013.

[70] Haoyu Lu, Mingyu Ding, Nanyi Fei, Yuqi Huo, and Zhiwu Lu. Lgdn: Language-guided denoising network for video-language modeling. *NeurIPS*, 2022.

[71] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv*, 2020.

[72] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 2022.

[73] Hui Lv, Chen Chen, Zhen Cui, Chunyan Xu, Yong Li, and Jian Yang. Learning normal dynamics in videos with meta prototype network. In *CVPR*, 2021.

[74] Snehashis Majhi, Srijan Das, François Brémond, Ratnakar Dash, and Pankaj Kumar Sa. Weakly-supervised joint anomaly detection and classification. In *FG 2021*, 2021.

[75] Ramna Maqsood, Usama Ijaz Bajwa, Gulshan Saleem, Rana Hammad Raza, and Muhammad Waqas Anwar. Anomaly recognition from surveillance videos using 3d convolution neural network. *Multimedia Tools and Applications*, 2021.

[76] Effrosyni Mavroudi, Triantafyllos Afouras, and Lorenzo Torresani. Learning to ground instructional articles in videos through narrations. In *ICCV*, 2023.

[77] Tao Mei and Cha Zhang. Deep learning for intelligent video analysis. In *ACM Multimedia*, 2017.

[78] Willi Menapace, Aliaksandr Siarohin, Ivan Skorokhodov, Ekaterina Deyneka, Tsai-Shien Chen, Anil Kag, Yuwei Fang, Aleksei Stoliar, Elisa Ricci, Jian Ren, et al. Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. In *CVPR*, 2024.

[79] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020.

[80] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[81] Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. In *ICCV*, 2023.

[82] Medhini G Narasimhan. Dynamic video anomaly detection and localization using sparse denoising autoencoders. *Multimedia Tools and Applications*, 2018.

[83] Rashmiranjan Nayak, Umesh Chandra Pati, and Santos Kumar Das. A comprehensive review on deep learning-based methods for video anomaly detection. *Image and Vision Computing*, 2021.

[84] OpenAI. Gpt-4o mini: advancing cost-efficient intelligence. `https://openai.com/index/gpt-4-1/`, 2024.

[85] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *CVPR*, 2020.

[86] Maitreya Patel, Abhiram Kusumba, Sheng Cheng, Changhoon Kim, Tejas Gokhale, Chitta Baral, and Yezhou Yang. Tripletclip: Improving compositional reasoning of clip via synthetic vision-language negatives. *NeurIPS*, 2024.

[87] Wujian Peng, Sicheng Xie, Zuyao You, Shiyi Lan, and Zuxuan Wu. Synthesize, diagnose, and optimize: Towards fine-grained vision-language understanding. *arXiv*, 2023.

[88] Chiara Plizzari, Alessio Tonioni, Yongqin Xian, Achin Kulshrestha, and Federico Tombari. Omnia de egotempo: Benchmarking temporal understanding of multi-modal llms in egocentric videos. *CVPR*, 2025.

[89] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models, 2025.

[90] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[91] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *CVPR*, 2022.

[92] Dvir Samuel, Rami Ben-Ari, Simon Raviv, Nir Darshan, and Gal Chechik. Generating images of rare concepts using pre-trained diffusion models. In *AAAI*, 2024.

[93] Sara Sarto, Manuele Barraco, Marcella Cornia, Lorenzo Baraldi, and Rita Cuc-

chiara. Positive-augmented contrastive learning for image and video captioning evaluation. In *CVPR*, 2023.

[94] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv*, 2022.

[95] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv*, 2021.

[96] Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only time can tell: Discovering temporal data for temporal modeling. In *WACV*, 2021.

[97] Yuhan Shen and Ehsan Elhamifar. Progress-aware online action segmentation for egocentric procedural task videos. In *CVPR*, 2024.

[98] Yuhan Shen, Lu Wang, and Ehsan Elhamifar. Learning to segment actions from visual and language instructions via differentiable weak sequence alignment. In *CVPR*, 2021.

[99] Yaya Shi, Xu Yang, Haiyang Xu, Chunfeng Yuan, Bing Li, Weiming Hu, and Zheng-Jun Zha. Emscore: Evaluating video captioning via coarse-grained and fine-grained embedding matching. In *CVPR*, 2022.

[100] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *CVPR*, 2022.

[101] Fahad Sohrab, Jenni Raitoharju, Moncef Gabbouj, and Alexandros Iosifidis. Subspace support vector data description. In *ICPR*, 2018.

[102] Yale Song, Eugene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4d goal-step: Toward hierarchical understanding of procedural activities. *NeurIPS*, 2023.

[103] Jessie James P Suarez and Prospero C Naval Jr. A survey on deep learning techniques for video anomaly detection. *arXiv*, 2020.

[104] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *CVPR*, 2018.

[105] Che Sun, Yunde Jia, and Yuwei Wu. Evidential reasoning for video anomaly detection. In *ACM Multimedia*, 2022.

[106] Shengyang Sun and Xiaojin Gong. Hierarchical semantic contrast for scene-aware video anomaly detection. In *CVPR*, 2023.

[107] Kamalakar Vijay Thakare, Debi Prosad Dogra, Heeseung Choi, Haksub Kim, and Ig-Jae Kim. Rareanom: A benchmark video dataset for rare type anomalies. *Pattern Recognition*, 2023.

[108] Kamalakar Vijay Thakare, Yash Raghuwanshi, Debi Prosad Dogra, Heeseung Choi, and Ig-Jae Kim. Dyannet: A scene dynamicity guided self-trained video anomaly detection network. In *WACV*, 2023.

[109] Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *NeurIPS*, 2024.

[110] Yu Tian, Guansong Pang, Yuanhong Chen, Rajvinder Singh, Johan W Verjans, and Gustavo Carneiro. Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In *ICCV*, 2021.

[111] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv*, 2023.

[112] Anil Osman Tur, Nicola Dall'Asen, Cigdem Beyan, and Elisa Ricci. Exploring diffusion models for unsupervised video anomaly detection. In *ICIP*, 2023.

[113] Anil Osman Tur, Nicola Dall'Asen, Cigdem Beyan, and Elisa Ricci. Unsupervised video anomaly detection with diffusion models conditioned on compact motion representations. In *ICIAP*, 2023.

[114] Waseem Ullah, Tanveer Hussain, and Sung Wook Baik. Vision transformer attention with multi-reservoir echo state network for anomaly recognition. *Information Processing & Management*, 2023.

[115] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[116] Lucas Ventura, Antoine Yang, Cordelia Schmid, and Gül Varol. Covr: Learning composed video retrieval from web video captions. In *AAAI*, 2024.

[117] Gaoang Wang, Xinyu Yuan, Aotian Zheng, Hung-Min Hsu, and Jenq-Neng Hwang. Anomaly candidate identification and starting time estimation of vehicles from traffic videos. In *CVPRW*, 2019.

[118] Jue Wang and Anoop Cherian. Gods: Generalized one-class discriminative subspaces for anomaly detection. In *ICCV*, 2019.

[119] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv*, 2021.

[120] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, 2019.

[121] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv*, 2023.

[122] Zhenhailong Wang, Ansel Blume, Sha Li, Genglin Liu, Jaemin Cho, Zineng Tang, Mohit Bansal, and Heng Ji. Paxion: Patching action knowledge in video-language foundation models. *NeurIPS*, 2024.

[123] Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, et al. Language models with image descriptors are strong few-shot video-language learners. *NeurIPS*, 2022.

[124] Ziming Wang, Yuexian Zou, and Zeming Zhang. Cluster attention contrast for video anomaly detection. In *ACM Multimedia*, 2020.

[125] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 2022.

[126] wikiHow. wikihow: How-to instructions you can trust. `https://www.wikihow.com/`.

[127] Jay Zhangjie Wu, Guian Fang, Haoning Wu, Xintao Wang, Yixiao Ge, Xiaodong Cun, David Junhao Zhang, Jia-Wei Liu, Yuchao Gu, Rui Zhao, et al. Towards a better metric for text-to-video generation. *arXiv*, 2024.

[128] Jhih-Ciang Wu, He-Yen Hsieh, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. Self-supervised sparse representation for video anomaly detection. In *ECCV*, 2022.

[129] Peng Wu and Jing Liu. Learning causal temporal relation and feature discrimination for anomaly detection. *IEEE TIP*, 2021.

[130] Peng Wu, Jing Liu, Yujia Shi, Yujia Sun, Fangtao Shao, Zhaoyang Wu, and Zhiwei Yang. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *ECCV*, 2020.

[131] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv*, 2024.

[132] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021.

[133] Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, et al. mplug-2: A modularized multi-modal foundation model across text, image and video. In *ICML*, 2023.

[134] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *EMNLP*, 2021.

[135] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016.

[136] Ke Xu, Tanfeng Sun, and Xinghao Jiang. Video anomaly detection and localization based on an adaptive intra-frame classification network. *IEEE Transactions on Multimedia*, 2019.

[137] Cheng Yan, Shiyu Zhang, Yang Liu, Guansong Pang, and Wenjun Wang. Feature prediction diffusion model for video anomaly detection. In *ICCV*, 2023.

[138] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv*, 2024.

[139] Michal Yarom, Yonatan Bitton, Soravit Changpinyo, Roee Aharoni, Jonathan Herzig, Oran Lang, Eran Ofek, and Idan Szpektor. What you see is what you read? improving text-image alignment evaluation. *NeurIPS*, 2024.

[140] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv*, 2023.

[141] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *ICLR*, 2022.

[142] M Zaigham Zaheer, Arif Mahmood, M Haris Khan, Mattia Segu, Fisher Yu, and Seung-Ik Lee. Generative cooperative learning for unsupervised video anomaly detection. In *CVPR*, 2022.

[143] Muhammad Zaigham Zaheer, Jin-ha Lee, Marcella Astrid, and Seung-Ik Lee. Old is gold: Redefining the adversarially learned one-class classifier training paradigm. In *CVPR*, 2020.

[144] Muhammad Zaigham Zaheer, Arif Mahmood, Marcella Astrid, and Seung-Ik Lee. Claws: Clustering assisted weakly supervised learning with normalcy suppression for anomalous event detection. In *ECCV*, 2020.

[145] Luca Zanella, Benedetta Liberatori, Willi Menapace, Fabio Poiesi, Yiming Wang, and Elisa Ricci. Delving into clip latent space for video anomaly recognition. *arXiv*, 2023.

[146] Luca Zanella, Massimiliano Mancini, Willi Menapace, Sergey Tulyakov, Yiming Wang, and Elisa Ricci. Can text-to-video generation help video-language alignment? *CVPR*, 2025.

[147] Luca Zanella, Willi Menapace, Massimiliano Mancini, Yiming Wang, and Elisa Ricci. Harnessing large language models for training-free video anomaly detection. In *CVPR*, 2024.

[148] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *EMNLP*, 2023.

[149] Hao Zhang, Aixin Sun, Wei Jing, Liangli Zhen, Joey Tianyi Zhou, and Rick Siow Mong Goh. Natural language video localization: A revisit in span-based question answering framework. *IEEE TPAMI*, 2021.

[150] Jiangong Zhang, Laiyun Qing, and Jun Miao. Temporal convolutional network with complementary inner bag loss for weakly supervised anomaly detection. In *ICIP*, 2019.

[151] 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.

[152] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 2022.

[153] Yongchao Zhou, Hshmat Sahak, and Jimmy Ba. Training on thin air: Improve image classification with generated data. *arXiv*, 2023.

[154] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv*, 2025.

[155] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019.