



RAGME: Retrieval Augmented Video Generation for Enhanced Motion Realism

Elia Peruzzo
University of Trento
Trento, Italy
elia.peruzzo@unitn.it

Dejia Xu
University of Texas at Austin
Austin, USA
dejia@utexas.edu

Xingqian Xu
Picsart AI Reserach
Atlanta, USA
xu16@illinois.edu

Humphrey Shi
Georgia Tech
Atlanta, USA
shihonghui3@gmail.com

Nicu Sebe
University of Trento
Trento, Italy
niculae.sebe@unitn.it

Abstract

Video generation is experiencing a rapid growth, driven by advances in diffusion models and the development of better and larger datasets. However, producing high-quality videos remains challenging, due to the high dimensional data and the complexity of the task. Recent efforts have primarily focused on enhancing visual quality and addressing temporal inconsistencies, such as flickering. Despite progress in these areas, the generated videos often fall short in terms of motion complexity and physical plausibility, with many outputs either appearing static or exhibiting unrealistic motion. In this work, we propose a framework aimed at improving the realism of motion in generated videos, exploring a complementary direction to much of the existing literature. Specifically, we advocate for the incorporation of a retrieval mechanism during the generation phase. The retrieved videos act as grounding signals, providing the model with demonstrations of how the objects move. Our pipeline is designed to apply to any text-to-video diffusion model, conditioning a pre-trained model on the retrieved samples with minimal fine-tuning. We demonstrate the superiority of our approach through established metrics, recently proposed benchmarks, and qualitative results, and we highlight additional applications of the framework. Code available at: <https://github.com/helia95/ragme>.

CCS Concepts

• **Computing methodologies** → **Computer vision**; **Animation**.

Keywords

Video Generation, Retrieval Augmented Generation, Motion Customization, Video Editing.

ACM Reference Format:

Elia Peruzzo, Dejia Xu, Xingqian Xu, Humphrey Shi, and Nicu Sebe. 2025. RAGME: Retrieval Augmented Video Generation for Enhanced Motion Realism. In *Proceedings of the 2025 International Conference on Multimedia Retrieval (ICMR '25)*, June 30-July 3, 2025, Chicago, IL, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3731715.3733417>



This work is licensed under a Creative Commons Attribution 4.0 International License. *ICMR '25, Chicago, IL, USA*

© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1877-9/2025/06
<https://doi.org/10.1145/3731715.3733417>

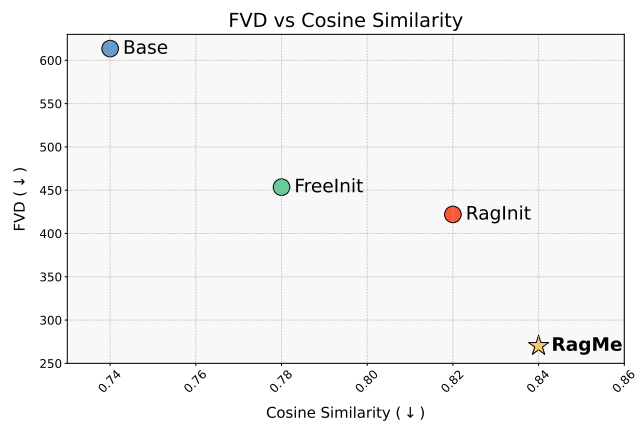


Figure 1: We evaluate the Fréchet Video Distance (FVD) using the captions and videos from the validation set of the WebVid10M [1] dataset. We plot it against the cosine similarity with respect to the retrieved examples in the DINOv2 embedding space. Ideally, the best model should produce high-quality videos (indicated by low FVD) while avoiding direct copying from the grounding examples (indicated by low cosine similarity).

1 Introduction

Text-to-video (T2V) generation is rapidly advancing, with large-scale models trained on vast datasets achieving increasingly impressive results. Notably, SORA [7] has established a new state-of-the-art, showcasing the remarkable potential of massive data and computational scaling. However, a significant limitation of current models lies in the realism and motion complexity of the objects in the output results. The generated videos often result in static scenes with simplistic or physically implausible motion [60]. Some works tackle this issue by improving the data curation pipeline [2] or proposing a different architecture that scales better with the computation [36]. However, all these models seem to suffer from similar failure cases, suggesting that scaling data and computing power are not sufficient to solve the problem.

In this work, we explore a complementary approach, i.e., incorporating grounding information to guide the network toward a more realistic and plausible motion. We propose a retrieval augmented

generation (RAG) pipeline – a technique that has demonstrated impressive results in Natural Language Processing (NLP) [29, 41]. However, it remains underutilized in computer vision, particularly in video generation. We retrieve (real) examples from an external database to guide the model and enhance the temporal dynamics of the generated samples. We term our method RAGME, Retrieval Augmented Generation for Motion Enhancement.

Our approach is inspired by the related tasks of video editing and motion transfer [14, 35, 39, 57]. In these settings, the goal is to synthesize an output video given one (or more) input video and a prompt describing the edit. The input videos are crucial for preserving motion, serving as an anchor for the video editing algorithm. We draw from these techniques but apply them to the broader problem of video generation. Our goal is to transfer the high-level action from the retrieved examples without preserving their specific details. Specifically, our design choices focus on preventing the transfer of low-level details, such as the background, the subject’s identity, or the spatial arrangement of the scene. For example, when generating a video of a person walking, we can gather samples from an external database where the action is performed in various ways. People have distinct identities and walk in different ways, in different directions, and across different environments. However, the underlying action remains consistent across these examples, and all of these variations can guide the model to produce a video with a more realistic motion. In this work, we aim to preserve only high-level information, allowing the model to generate new content without directly copying specific instances from the retrieved examples. When evaluating Fréchet Video Distance (FVD), our method significantly reduces this metric compared to the base model while ensuring that the generated video is not a replica of the retrieved samples, as indicated by a slight increase in cosine similarity between them (see Fig. 1).

We build our pipeline in a general manner, without specific assumptions about the architecture or the application (e.g., humans). We use the WebVid10M as a large-scale text-to-video dataset and use it to build a retrieval mechanism, which is used to condition a pre-trained T2V model by inserting cross-attention layers that fuse information from retrieved samples. Additionally, we propose a novel mechanism to initialize the random noise for the denoising process leveraging the retrieved samples. We evaluate our model through standard metrics like FVD, but also on the recently proposed video generation benchmarks. We demonstrate superior results compared to baselines and training-free methods for enhancing video quality and consistency. The core contribution of this work is to apply for the first time a RAG pipeline to video generation as a first step to guide the model towards more realistic motion generation.

2 Related Works

Text-to-Video Diffusion Models. In the last years, there have been several efforts to expand the achievements of text-to-image models to the video domain [3, 15, 20, 45, 50, 52]. ImagenVideo [20] and Make-A-Video [45] propose a deep cascade of temporal and spatial upsamplers to generate videos and jointly train their models on image and video datasets. A consistent line of works focus on extending powerful pre-trained text-to-image (T2I) models introducing new layers to model the time dimension and exploiting the

powerful prior learned on the spatial domain [50, 52]. Blattmann *et al.* [4] initially explored this direction by leveraging a pre-trained Stable Diffusion model [42], which was later extended to image-to-video generation and longer videos by Stable Video Diffusion [2]. AnimateDiff [18] proposes to freeze the spatial layers and train only the temporal module and introduce MotionLoRA [22] as a lightweight finetuning technique to learn specific motion patterns. Nevertheless, all these methods rely on 3DUNet with separable spatial and temporal computation which poses a limitation on motion modeling capabilities. SnapVideo [36] proposes to use a transformer-based FIT [30] architecture which can jointly model the space and time components, by exploiting a compressed video latent representation. Other works introduce fully transformer-based architectures [33], culminating in the state-of-the-art results achieved by SORA [7]. While the open-source community is working to replicate these outcomes, the generated quality still lags behind [28, 60].

Concurrently, some approaches have explored not only the architectural modeling choices but also the noising policy. Pyoco [15] introduces a noise-correlated sampling strategy, based on the intuition that frames shouldn’t be sampled from independent noise. Recently, FreeInit [54] proposed a training-free technique to optimize the initial noise of the denoising process. The model predicts a sample that is diffused back according to the noising schedule, mixing the low-frequency components with randomly initialized high-frequency components. While this approach results in improved sample consistency, it requires repeating the sampling process multiple times, which is often impractical.

We build on the recent advancement of T2V models, leveraging the strengths of powerful pre-trained models and extending their capabilities with minimal architecture modifications. Additionally, we propose a noise initialization strategy that enhances the final result without incurring the high computational costs associated with existing methods.

Motion Transfer and Video Editing. One line of work exploits pre-trained T2I models and adapts them to the task in a zero-shot manner [9, 16, 27, 39, 57]. The temporal consistency of the generated frames is typically obtained by extending the self-attention operation across frames [27, 53]. Tune-A-Video [53] involves fine-tuning the model on the video to be edited, enabling test-time edits through text prompts or cross-attention control [31]. Pix2Video [9] and FateZero [39] propose a training-free approach, exploiting the attention maps extracted during an initial inversion step and blended with those generated during the editing process, confining the edit to a specific region. TokenFlow [16] and FLATTEN [10] propose to propagate features of the base T2I model leveraging the optical flow extracted from the source video. In contrast, other methods opt for pretraining on video datasets, typically employing an inflated 3DUNet architecture and incorporating explicit dense conditioning signals (e.g., optical flow, depth maps, or sketches) to preserve motion and structure from the guiding video [14, 17, 18, 38, 51]. Animate-A-Story [19] utilizes a similar technique for guiding generation, but instead of relying on user-provided input, it retrieves a single video from a database to serve as the anchor. Other works have explored the broader task of motion transfer. Yatim *et al.* [58] addresses motion transfer between objects of different categories that may not share the same motion characteristics. They enforce

the transfer through an inference-time optimization, introducing a loss to match the correlation of features of the input with the output video. Similarly, [35, 59] propose a DreamBooth-like [43] training strategy to learn motion patterns from a set of videos with the same action.

Our work is inspired by this line of research but differs fundamentally because we do not aim to replicate the conditioning video, nor do we rely on a manually curated set of examples. Furthermore, we seek a practical implementation that avoids costly test-time training procedures.

Retrieval Augmented Generation (RAG). It represents a well established technique in Natural Language Processing as a powerful way to improve model performances, by integrating information from an external database that acts as a memory bank [6, 29, 41]. Early attempts to adapt similar retrieval mechanisms for image and video generation were introduced within the context of GANs [8, 48]. More recently, [5, 44] have applied these concepts to image diffusion models. Their approach involves a semi-parametric generative model that combines a learnable module with an external database, allowing for post-hoc conditioning based on labels, prompts, or specific styles. Re-Imagen [6] extends this concept to text-to-image (T2I) models, and [56] propose an in-context learning strategy to integrate retrieved samples and enhance generation results.

To the best of our knowledge, RAG has not yet been applied to video generation, which presents additional challenges in both the retrieval mechanism and the model's conditioning component.

3 Method

We describe the technical details of RAGME, formalize the task, and outline its applications. We begin by defining the notation used throughout the paper. We assume to have access to a database $\mathcal{D} = \{X_i\}_{i=1}^N$. Each data-point represents a video, with $X_i \in \mathbb{R}^{T \times 3 \times H \times W}$ denotes the T frames of the video with spatial resolution $H \times W$.

We define a *Retrieval Mechanism* (RM) as a non-learnable function to retrieve from the database given a query q , i.e. $f_K : (q, \mathcal{D}) \rightarrow \mathcal{Z}$, with $\mathcal{Z} = \{(X_j, T_j)\}_{j=1}^K$, $\mathcal{Z} \subseteq \mathcal{D}$ and $K = |\mathcal{Z}|$ represents the number of retrieved samples. Next, we define $g_\theta : \mathcal{T}_i \rightarrow \mathcal{Y}_i$ as a (pretrained) *T2V Generative Model* that synthesizes an output video $\mathcal{Y}_i \in \mathbb{R}^{T \times 3 \times H \times W}$ given a textual prompt $\mathcal{T}_i$.

In this work, we propose to learn a *semi-parametric* T2V model, which can incorporate relevant retrieved samples via conditioning, i.e. $g_\theta : (\mathcal{T}_i, \mathcal{Z}) \rightarrow \mathcal{Y}_i$. As discussed in Sec. 1, our final goal is to produce videos with better temporal dynamics, *without copy-pasting* artifacts from the retrieved examples.

T2V Diffusion Models Preliminaries. Diffusion models are probabilistic models that approximate distributions by iteratively denoising data. Starting with a sample of Gaussian noise, the model learns to progressively remove noise in steps until the sample approximates the target distribution [21, 46]. Our framework builds upon a pre-trained *latent* T2V model [2, 42]. Instead of learning the distribution directly in the complex, high-dimensional video space, this model projects the video into a compressed latent representation and learns a conditional distribution based on text. Architecturally, it consists of three main components: The VAE Encoder $\mathcal{E}(\cdot)$, which projects the raw input pixels to the latent space i.e. $z = \mathcal{E}(X)$, and

the correspondent Decoder $\mathcal{D}(\cdot)$. The text encoder $\tau_\theta(\cdot)$, which maps the input textual prompt to a conditioning vector; and the denoiser $\epsilon_\theta(\cdot)$, which takes the text embedding and a noisy version of the latent as input and predicts (with the correct reparametrization [21]) the added noise.

The training is performed by sampling a noise $\epsilon \sim \mathcal{N}(0, 1)$ and diffusing the original sample z_0 according to a noise scheduler function and a time-step $t \sim f(t)$ [11, 21, 24]. The diffused sample z_t is computed as

$$z_t = \sqrt{\alpha_t} \cdot z_0 + \sqrt{1 - \alpha_t} \cdot \epsilon \quad (1)$$

where α_t is a parameter controlled by the noise scheduler function that dictates the amount of noise at timestep t . At the final timestep $t = T$, the original sample is completely destroyed to pure noise, i.e. $z_T \sim \mathcal{N}(0, 1)$, which allows sampling from the model at inference time.

The parameters of the denoiser network are trained to recover the added noise. Specifically, the training loss is defined as:

$$\mathcal{L}_{\text{simple}} := \mathbb{E}_{\mathcal{E}(x), \epsilon \sim \mathcal{N}(0,1), t} \left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(c))\|_2^2 \right] \quad (2)$$

In this work, we focus on the denoiser network $\epsilon_\theta(\cdot)$. Although purely transformer based architecture are emerging, we rely on the widespread 3DUNet models [2, 3, 50, 52]. From an architectural perspective, combines convolutional layers with attention operations. The attention blocks can be further categorized into the:

- *Cross-Attention* blocks, which integrate information from the text encoder.
- *Spatial Attention* blocks, which operate on the spatial dimension treating each frame independently, the activation of the network are reshaped as $x_{\text{spatial}} \in \mathbb{R}^{(b \cdot T) \times (h \cdot w) \times \text{dim}}$.
- *Temporal Attention* blocks, which operate solely on the temporal axis, the activation of the network are reshaped as $x_{\text{temp}} \in \mathbb{R}^{(b \cdot h \cdot w) \times T \times \text{dim}}$.

In this work, we concentrate on the *temporal attention* blocks, as our primary goal is to enhance the temporal dynamics of the generated video.

Retrieval Mechanism (RM). The retrieval mechanism processes a query q and retrieve K samples from a database $\mathcal{D}$. The retrieval is performed by minimizing a distance function $d(q, \cdot)$ between the query and the other entries in the database. In practice, it is composed of three non-learnable blocks: the pre-trained text encoder f_{txt} , the pre-trained visual encoder f_{vis} and an indexing mechanism f_{index} . Following the previous works, we use CLIP to implement the visual and textual encoders. Our choice is motivated by three factors: (i) previous works on video-action recognition show that frame-wise CLIP encodings are powerful for the task, and can be used to recognize the action with high accuracy [1, 32, 34?] (ii) the embedding space is compact and reduces the dimensionality ($\text{dim} = 512$), with advantages in memory and computational requirements, (iii) the shared textual-visual embedding space allows to search the database in a multi-model manner at inference time (i.e. using the prompt of the T2V model as the query for the retrieval).

First, we preprocess the database $\mathcal{D}$. For each video X_i , we encode the frames independently and compute the average along the

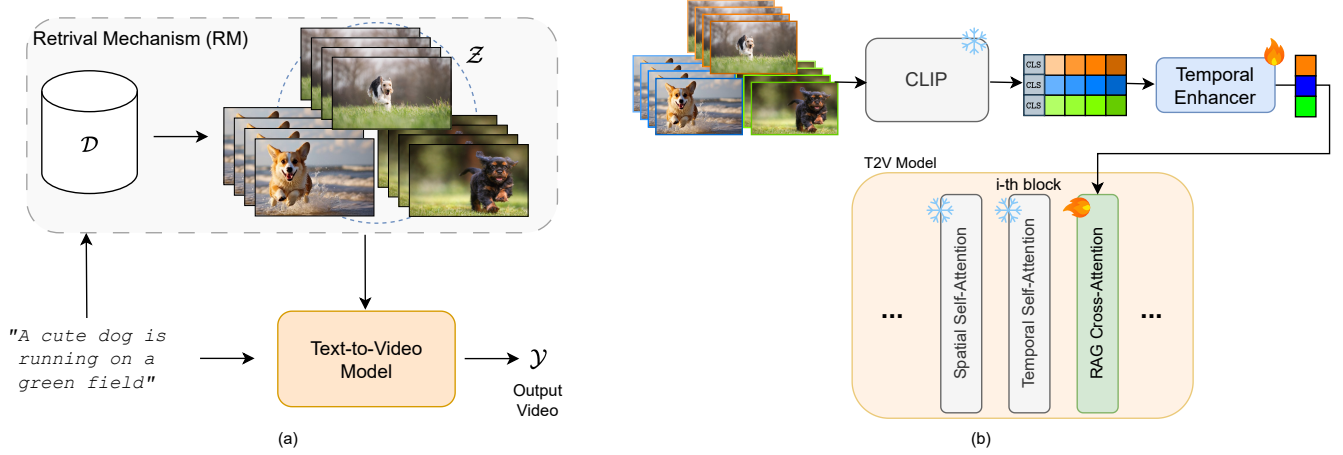


Figure 2: Pipeline of RAGME. (a) We show a general T2V pipeline with RAG capabilities. Given a textual prompt, we retrieve related videos from a database and use it to enhance the generation capabilities of a T2V model. (b) We detail the specific implementation. Each video frame from the retrieved videos is encoded using CLIP and then processed by a transformer temporal enhancer module to obtain the final conditioning vector. This vector is used to condition a T2V model through cross-attention layers. Each video is color-coded, with different frames represented by varying shades of the base color.

temporal dimension to aggregate the information. This results in a per-video representation, after L2 normalization:

$$\mathbf{x}_i = \left\| \frac{1}{T} \sum_{j=1}^T f_{\text{vis}}(\mathcal{X}_{i,j}) \right\|_2. \quad (3)$$

Second, we efficiently store the compressed video representations in the index using the FAISS library [13]. Next, we search over the index, returning K samples from the database, which maximize the *cosine similarity* d_{cos} with the query:

$$\mathbf{Z} = \underset{\mathbf{Z}_j \in \mathcal{D}}{\text{top-}K} \quad d_{\text{cos}}(q, \mathbf{Z}_j) \quad (4)$$

with $\mathbf{Z} = \{\mathbf{Z}_0, \dots, \mathbf{Z}_K\}$, $q \notin \mathbf{Z}$.

During training, we compute the averaged temporal CLIP representation for the current video $\mathcal{X}_i$ as described in Eq. (3). Then, we search the dataset using Eq. (4), setting the query $q = \mathbf{x}_i$. Conversely, at test time, we encode the given textual prompt $\mathcal{T}_i$ using the CLIP textual encoder, *i.e.*, $\mathbf{t}_i = \|f_{\text{txt}}(\mathcal{T}_i)\|_2$. Finally, we leverage the multimodal nature of the CLIP latent space and retrieve from the dataset using Eq. (4), setting the query $q = \mathbf{t}_i$. We refer to Fig. 2 (a) for a visual representation of the process.

Note that, for the sake of generality, we assume the database to contain *only* videos, but the pipeline can be applied to text-video database as well. We explore other choices for the retrieval system and discuss the result in the Sec. 4. Lastly, we apply a deduplication strategy to prevent returning (multiple) similar elements in a dataset with redundant entries. Further details on the implementation and post-processing are provided in the *Supp. Mat.*.

Retrieval Augmented Conditioning (RagCA). After developing the retrieval mechanism, we explain how to condition the T2V model using this retrieved information. For a visual representation of the process, refer to Fig. 2 (b). The first step involves representing

the conditioning videos within an appropriate embedding space. Consistent with our guiding principle, our goal is to condition the main network in a way that enhances temporal dynamics, while avoiding direct copies of the conditioning signals. The CLIP visual encoder emerges as a strong candidate for this purpose, as it effectively encodes high-level semantic without retaining low-level information [40]. Additionally, it offers a practical solution since we can directly utilize the embeddings returned by the retrieval mechanism. However, since f_{vis} operates on independent frames, we introduce a module specifically designed to handle the temporal dimension, which we term the *transformer time enhancer* model. In practice, we pack the per-frame CLIP embedding into a sequence of tokens:

$$\bar{\mathbf{z}}_i = [\text{CLS}; f_{\text{vis}}(\mathbf{Z}_{i,0}); \dots; f_{\text{vis}}(\mathbf{Z}_{i,T})] \quad (5)$$

with $\bar{\mathbf{z}}_i \in \mathbb{R}^{(T+1) \times \text{dim}}$, $[\dots; \dots]$ represents the concatenation operation and [CLS] is a class token appended at the beginning of the sequence [12]. We apply the transformer time enhancer independently on each retrieved videos and pool the [CLS] token in output. In this way, we obtain the final conditioning signal $\mathbf{z} = \tau(\bar{\mathbf{z}})$, with $\mathbf{z} \in \mathbb{R}^{b \times K \times \text{dim}}$ (see Fig. 2 (b)).

Next, we condition the pre-trained T2V model retaining the generation capabilities learned during the pertaining stage. Following previous works, we initialize new multi-head cross attention layers and inject them after every temporal attention layer of the base model. In practice, let $\mathbf{x}_{\text{temp}} \in \mathbb{R}^{(b \cdot h \cdot w) \times T \times \text{ch}}$ be the 3DUNet activation after a temporal layer, we compute a residual:

$$\mathbf{x}_{\text{temp}} = \mathbf{x}_{\text{temp}} + \text{MCA}(\mathbf{x}_{\text{temp}}, \mathbf{z}) \quad (6)$$

where $\text{MCA}(\cdot)$ represent the multi-head cross-attention operation with queries computed from $\mathbf{x}_{\text{temp}}$ and keys/values from the $\mathbf{z}$ signals respectively.

RAG Noise Initialization (RagInit). As explored in previous works [25, 54, 55], noise initialization plays an important role in diffusion models and can greatly affect the quality of the generated result. We further leverage the retrieved videos and propose to initialize the noise averaging the latents. We diffuse the result following Eq. (1) and setting $t = T$:

$$z_T^{\text{RAG}} = \sqrt{\alpha_T} \cdot \frac{1}{K} \sum_{i=1}^K \mathcal{E}(\mathcal{Z}_i) + \sqrt{1 - \alpha_T} \cdot \epsilon \quad (7)$$

This strategy is very fast, as it doesn't require inversion, and comes at the additional cost of running the VAE encoder on the retrieved videos. Nevertheless, it has the advantage of providing a good initialization for the noise which is likely to be aligned with the conditioning videos.

Implementation Details. We build our framework on Zeroscope [47], a latent T2V model based on an inflated 3DUNet architecture with factorized spatial and temporal layers. We develop the retrieval system using the WebVid10M dataset [1]; our choice is motivated by the large scale and the general-purpose nature of its videos, which cover a wide range of scenarios. For the retrieval mechanism, we implement the CLIP ViT-B-32 [40] as our feature extractor to handle both f_{vis} and f_{txt} . This model, pre-trained with a contrastive loss on images and captions from a large-scale dataset, outputs a 512-dimensional embedding representing the respective input. Although the choice of the encoder for the retrieval mechanism could, in principle, be independent of the conditioning process, we find it easier and more convenient to use the same encoder.

Next, we leverage the FAISS library [13] to create an index for efficient retrieval. The WebVid10M dataset contains duplicate or highly similar videos; to prevent the model from processing redundant information, given a query $\mathbf{q}$, we apply a deduplication strategy based on the cosine similarity between samples. We empirically set the deduplication threshold at $\delta_{\text{dedup}} = 0.965$ and maintain this value across all experiments. Additionally, to ensure that the retrieved videos are relevant to the query, we set a minimum cosine similarity threshold of $\delta_{\text{min}} = 0.6$ and remove samples from the retrieval set $\mathcal{Z}$ that do not meet this criterion. This filtering is particularly applied when retrieving a large number of samples (*i.e.* $K = 20$, $K = 50$). In such cases, padding is used to match the required length. From an architectural point of view, we introduce the transformer temporal enhancer module to improve the temporal representation of the video. It is composed of 6 layers of transformer blocks with a hidden dimension of $\text{dim} = 512$. A learnable token [CLS] is added at the beginning of the sequence and pooled in output to represent the video. Lastly, we add multi-head cross-attention layers to the base T2V model ZeroScope. We introduce a point-wise convolution initialized with zero-weights, to act as the identity when the model is initialized.

The added modules are finetuned (while keeping the rest of the network frozen) for 200K iterations on the WebVid10M dataset, at resolution 448×256 and 12 frames. Training is performed with an effective batch size of 16, distributed on 4 Nvidia A100 GPUs.

Method	FVD ($\downarrow$)	DINO-S ($\downarrow$)	Latency (s) ($\downarrow$)
Retrieved Videos	117.22	1.00	-
ZeroScope	613.15	0.74	17.78
FreeInit	453.50	0.79	68.88
RAGInit	422.10	0.82	19.86
RAGME	270.26	0.84	22.43

Table 1: Comparison between the baseline methods and RAGME on the WebVid10M validation set.

4 Experiments

In this section, we qualitatively and quantitatively analyze the performance of RAGME. We start by evaluating established metrics in the video generation field on the validation set of WebVid10M [1]. Moreover, we follow VBench [23], a benchmark recently introduced, which exploits an array of pre-trained models to evaluate the generated videos under multiple angles. Next, we present a series of ablation studies to understand the role of each component in our pipeline. Lastly, we showcase several qualitative results comparing our method with the baselines. All the results are available as videos on the supplementary website.

Baselines and Setting. We compare RAGME with videos produced by the base T2V model, ZeroScope [47]. Next, we enhance the videos generated by the base model using FreeInit [54], a training-free technique that optimizes the starting noise of the diffusion process through repeated denoising. Finally, we compare our full model with another baseline, which uses our proposed RagInit technique to initialize the noise.

We perform inference from all the models using the DDIM sampler [46] with 50 denoising steps, and classifier-free guidance with scale of $s = 7.5$.

4.1 Quantitative Results

WebVid10M Results. Our end goal is to develop a system with better video quality, especially in the temporal dynamics, while avoiding leakage of the conditioning videos (see Sec. 1). To capture the first aspect, we rely on the Fréchet Video Distance (FVD) [49], which is well-established in the video generation literature. To estimate the second factor, *i.e.* possible copy-paste artifact from the retrieved videos, we compute the cosine similarity on the DINOv2 [37] embedding space. Specifically, given a generated video $\mathcal{Y}$ and a set of retrieved videos $\mathcal{Z}$, the metric is computed as $\max_{\mathcal{Z}} \cos\text{-sim}(\mathcal{Y}, \mathcal{Z}_i)$. In this case, a model that achieves a lower cosine similarity is considered better. Lastly, we compare the methods on the latency, *i.e.* the time to generate a single video. We take into account the time of retrieving the videos and encoding them with CLIP when computing the latency of our model. We refer to the *Supp. Mat.* for more detailed discussion.

We conduct the experiments on the WebVid10M validation set, which comprises 5000 videos with the associated captions. We report the results in Tab. 1, wherein the first row we report the results of the *retrieved videos* (*i.e.* videos from the WebVid10M training set) as a reference. RAGME drastically outperforms the base diffusion model in terms of FVD, resulting in videos of higher

Method	Human Action	Subject Consistency	Background Consistency	Motion Smoothness	Temporal Flickering	Dynamic Degree
ZeroScope	0.922	0.962	0.984	0.985	0.986	0.367
FreeInit	0.912	0.978	0.990	0.988	0.994	0.242
RagInit (Our)	0.952	0.961	0.985	0.985	0.990	0.467
RAGME	0.974	0.911	0.972	0.968	0.982	0.692

Table 2: Comparison between RAGME and the baselines on VBench [23]. We report the metrics related to motion dynamics and temporal consistency. Our method outperforms the competitors in the quality of motion while slightly decreasing the consistency-related metrics.

quality. While applying FreeInit does lead to some improvement, it remains inferior in comparison. RagInit achieves comparable performance to FreeInit. However, a notable difference emerges in latency: our proposed noise initialization method does not require costly denoising steps and instead uses the retrieved samples for noise initialization.

Analyzing the DINO-similarity metric, we observe that RAGME shows an increase compared to both the baseline and FreeInit. However, compared to RagInit, the full model’s improvement is minimal, suggesting that the primary issue may lie in the noise initialization procedure rather than the cross-attention conditioning. It is important to note that a *very low* DINO cosine similarity is not desirable as well, and would indicate: either a lack of relevance between the retrieved videos and the final video or a failure of the T2V model to align with the prompt. We recommend examining the qualitative results in the next section and the supplementary material for a comprehensive assessment.

VBench Results. While the FVD metric is well established, it is difficult to interpret as improvements over it can be due to multiple factors. To get a better understanding of *what aspects* our method is improving, we follow VBench [23] for a more detailed evaluation. VBench is a recently proposed benchmark for T2V models, which comprises a suite of roughly 900 prompts and a list of 16 dimensions for evaluations. In the main paper, we report only the metrics related to the temporal consistency and quality of motion, as these represent our main target for improvement. However, we refer the reader to the *Supp. Mat.* for full comparison between the methods, and to the original paper for detailed explanation of how each metric is computed. We report the results in Tab. 2. Our method strongly outperforms the baseline in two aspects: the Human Action and the Dynamic Degree metrics. This reflects our design goals of having less static videos with better motion. At the same time comes at the price of a slight decrease in background and subject consistency, which is nevertheless expected (a static video would achieve a perfect score in these metrics). Comparing with the noise initialization stargeies of FreeInit and RagInit, it is interesting to notice that a better action can be primarily explained by a better noise initialization, but the dynamic degree is mostly due to the cross-attention layers which incorporates the retrieved videos.

4.2 Ablations

Role of the database $\mathcal{D}$. We ablate the role of the retrieval database $\mathcal{D}$ in our system, specifically focusing on the types of videos we retrieve. In the previous section, we used a general retrieval mechanism without making strong assumptions about the task.

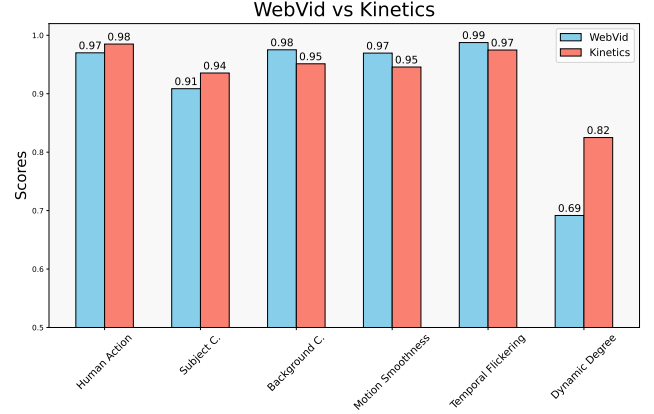


Figure 3: We compare the role of different retrieval databases on the person-related subset of VBench [23]. We retrieve from the Kinetics [26] and the WebVid10M [1].

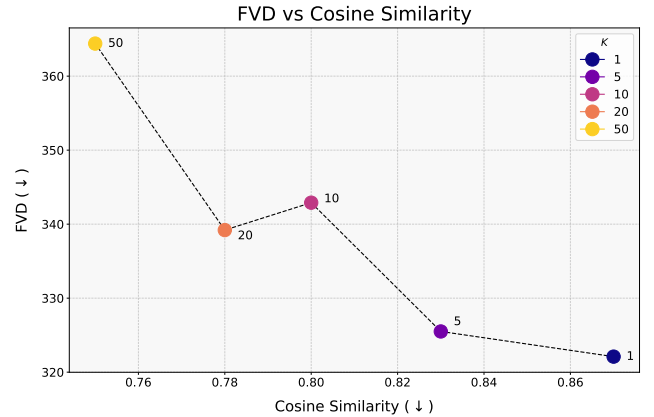


Figure 4: We study the impact of the retrieved samples K on the FVD vs Cosine Similarity trade-off. We select $K = 5$ as a good trade-off between the two.

The retrieval database consisted of general videos from WebVid, and we did not exploit the textual components. However, the proposed mechanism is highly flexible, allowing different databases to be used at inference time to retrieve videos tailored to specific applications. Hence, we assume access to an application-specific database for human-related prompts, specifically the Kinetics [26]

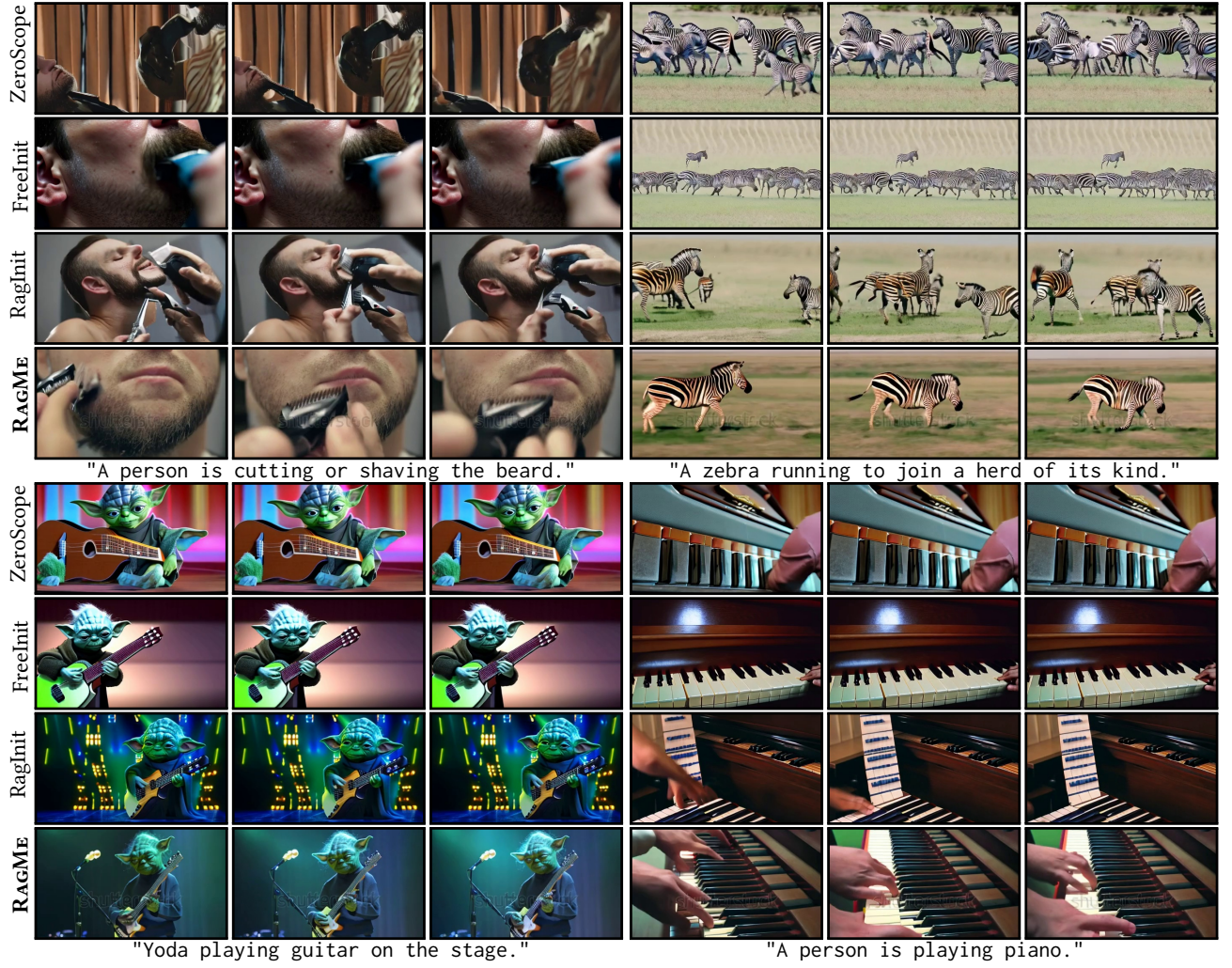


Figure 5: Visual comparison of the different methods. We report the prompt at the bottom.

video dataset, and plug it into our pipeline without further fine-tuning. This dataset, commonly used for action recognition tasks, contains a large set of actions performed by people. We replace our base dataset, derived from WebVid10M, with Kinetics and evaluate how this change affects performance on the VBench metrics. The results, shown in Fig. 3, demonstrate a relative improvement in both the Human Action and Dynamic Degree metrics. These findings highlight the importance of the retrieved videos in the process and suggest that the mechanism can be specialized for specific applications to achieve better performance.

Number of retrieved examples K . We study the impact of the number of retrieved samples on the final generated videos, comparing the FVD vs DINO-similarity trade-off. Specifically, we train different versions of the models to use different numbers of K . We use a reduced computation budget and train the models for 25k iterations. We report the results in Fig. 4. We can observe that $K = 1$, *i.e.* retrieving a single sample, achieves good FVD but incurs very high DINO-similarity (*i.e.* undesired copy-paste-effects). Vice-versa,

increasing K too much, results in progressively worse FVD probably because it becomes harder for the model to get meaningful information (besides incurring additional computational costs). We observe that $K = 5$ represents a good trade-off. We set this value and use it throughout all our experiments. In principle, nothing prevents us from training a model with a given K and adopting a different K' at inference time. However, we observed slightly reduced performances. We add a more detailed discussion, along with qualitative results for different K in the supplementary material.

4.3 Qualitative Results

In this section, we present a qualitative comparison between different methods, moreover, we explore an additional use case of our method *i.e.* *motion transfer*. In Fig. 5, we display frames from the generated videos based on prompts from the VBench suite. Our methods produce better videos in terms of both motion and scene composition. Additionally, in Fig. 6, we show the first frame of the generated video alongside the first frames of the five videos used



Figure 6: We show the first frame of the generated video and the first frame of the 5 retrieved samples used during the generation phase. No clear leakage is present, *i.e.* the model is not simply copy-pasting the output but using it to improve the result.



Figure 7: Results for the motion transfer task. The top row displays the reference video, followed by a comparison of Motion Director (MD) [59] and our method using two distinct prompts (shown at the bottom). Our approach achieves qualitatively similar results with $8\times$ fewer fine-tuning iterations.

for conditioning. We observe that no clear leakage is present, indicating that RAGME effectively integrates the retrieved information to achieve better results. The generated videos from our method contain watermarks due to the training dataset, WebVid10M [1]. However, training on higher-quality datasets would eliminate this artifact. Full video results are available on the supplementary web-site.

Motion Transfer. While our method is designed for flexible conditioning on multiple retrieved videos, a key application in video editing is motion transfer. This involves transferring motion from a reference video while controlling the appearance and overall style of the output, for instance, through a textual prompt.

Our approach is specifically designed to avoid explicit copy-paste artifacts, extracting only high-level motion semantics from the retrieved videos – aligning with our goal of enhancing generated motion in a generalizable way. However, for motion transfer, we can adapt our method accordingly. In practice, given a driving video, we overfit the controller network to that specific video to achieve the desired effect. Notably, the design of our architecture and pre-training on WebVid-10M facilitate this adaptation process, making it more efficient compared to other methods that require fine-tuning on the target video. Compared to Motion Director [59] (which relies on the same backbone video generator), our method achieves similar performance while requiring eight times less fine-tuning (50 vs 400 iterations), demonstrating the efficiency of our RAG-first design. Additional details are provided in the supplementary material.

5 Conclusions

In this work, we propose RAGME a framework for retrieval augmented text-to-video generation. We exploit retrieved videos to enhance the motion realism of the final result, showing superior performance both qualitatively and quantitatively. Moreover, we showcase how this framework can be adapted to specific tasks such as Motion Transfer, obtaining results on par with state-of-the-art at a fraction of the computational costs.

Our work opens the door to several future improvements. First, exploring the use of alternative encoders, such as video models, could provide more robust representations of actions. Extending our approach to other diffusion models and transformer-based architectures could further generalize the method, making it suitable for a wider range of applications. Lastly, expanding the model to handle the composition of multiple actions—rather than assuming a single action—would also broaden its applicability.

Acknowledgments

This work is supported by the European Commission under Horizon Europe Programme, grant number 101120237 - ELIAS and by the MUR PNRR project FAIR (PE00000013) founded by the NextGenerationEU.

References

- [1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. 2021. Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval. In *IEEE International Conference on Computer Vision*.

- [2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127* (2023).
- [3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. 2023. Align Your Latents: High-Resolution Video Synthesis With Latent Diffusion Models. In *CVPR*.
- [4] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. 2023. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22563–22575.
- [5] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. 2022. Retrieval-augmented diffusion models. *Advances in Neural Information Processing Systems* 35 (2022), 15309–15324.
- [6] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving Language Models by Retrieving from Trillions of Tokens. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 2206–2240.
- [7] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. 2024. Video generation models as world simulators. (2024). <https://openai.com/research/video-generation-models-as-world-simulators>
- [8] Arantxa Casanova, Marlene Careil, Jakob Verbeek, Michal Drozdal, and Adriana Romero Soriano. 2021. Instance-conditioned gan. *Advances in Neural Information Processing Systems* 34 (2021), 27517–27529.
- [9] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. 2023. Pix2video: Video editing using image diffusion. In *ICCV*.
- [10] Yuren Cong, Mengmeng Xu, Christian Simon, Shoufa Chen, Jiawei Ren, Yanping Xie, Juan-Manuel Perez-Rua, Bodo Rosenhahn, Tao Xiang, and Sen He. 2023. Flatten: optical flow-guided attention for consistent text-to-video editing. *arXiv preprint arXiv:2310.05922* (2023).
- [11] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.
- [12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xi-aohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR* (2021).
- [13] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The Faiss library. (2024). [arXiv:2401.08281](https://arxiv.org/abs/2401.08281) [cs.LG]
- [14] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. 2023. Structure and content-guided video synthesis with diffusion models. In *ICCV*.
- [15] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. 2023. Preserve your own correlation: A noise prior for video diffusion models. In *ICCV*.
- [16] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. 2023. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint* (2023).
- [17] Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. 2023. Sparsectrl: Adding sparse controls to text-to-video diffusion models. *arXiv preprint arXiv:2311.16933* (2023).
- [18] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. 2023. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint* (2023).
- [19] Yingqing He, Menghan Xia, Haoxin Chen, Xiaodong Cun, Yuan Gong, Jinbo Xing, Yong Zhang, Xintao Wang, Chao Weng, Ying Shan, et al. 2023. Animate-a-story: Storytelling with retrieval-augmented video generation. *arXiv preprint arXiv:2307.06940* (2023).
- [20] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. 2022. Imagen video: High definition video generation with diffusion models. *arXiv preprint* (2022).
- [21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
- [22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).
- [23] Ziqi Huang, Yinan He, Jiahao Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. 2024. VBench: Comprehensive Benchmark Suite for Video Generative Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- [24] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems* 35 (2022), 26565–26577.
- [25] Shyamgopal Karthik, Karsten Roth, Massimiliano Mancini, and Zeynep Akata. 2023. If at First You Don't Succeed, Try, Try Again: Faithful Diffusion-based Text-to-Image Generation by Selection. *arXiv preprint arXiv:2305.13308* (2023).
- [26] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950* (2017).
- [27] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. 2023. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint* (2023).
- [28] PKU-Yuan Lab and Tuzhan AI etc. 2024. *Open-Sora-Plan*. doi:10.5281/zenodo.10948109
- [29] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.
- [30] Lala Li and Ting Chen. 2023. FIT: Far-reaching Interleaved Transformers. (2023).
- [31] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. 2023. Video-p2p: Video editing with cross-attention control. *arXiv preprint* (2023).
- [32] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. 2022. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing* 508 (2022), 293–304.
- [33] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. 2024. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048* (2024).
- [34] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. 2022. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM International Conference on Multimedia*. 638–647.
- [35] Joanna Materzynska, Josef Sivic, Eli Shechtman, Antonio Torralba, Richard Zhang, and Bryan Russell. 2023. Customizing motion in text-to-video diffusion models. *arXiv preprint arXiv:2312.04966* (2023).
- [36] Willi Menapace, Aliaksandr Siarohin, Ivan Skorokhodov, Ekaterina Deyneka, Tsai-Shien Chen, Anil Kag, Yuwei Fang, Aleksei Stoliar, Elisa Ricci, Jian Ren, et al. 2024. Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7038–7048.
- [37] Maxime Quab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023).
- [38] Elia Peruzzo, Vidit Goel, DeJia Xu, Xingqian Xu, Yifan Jiang, Zhangyang Wang, Humphrey Shi, and Nicu Sebe. 2024. Vase: Object-centric appearance and shape manipulation of real videos. *arXiv preprint arXiv:2401.02473* (2024).
- [39] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. 2023. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv preprint* (2023).
- [40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.
- [41] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-Context Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics* 11 (11 2023), 1316–1331. doi:10.1162/tacl_a_00605 [arXiv:https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00605/2178834/tacl_a_00605.pdf](https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00605/2178834/tacl_a_00605.pdf)
- [42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.
- [43] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 22500–22510.
- [44] Shelly Sheynin, Oron Ashual, Adam Polyak, Uriel Singer, Oran Gafni, Eliya Nachmani, and Yaniv Taigman. 2022. Knn-diffusion: Image generation via large-scale retrieval. *arXiv preprint arXiv:2204.02849* (2022).
- [45] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. 2022. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint* (2022).
- [46] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).
- [47] Spencer Sterling. 2023. *Zeroscope*. https://huggingface.co/cerspense/zeroscope_v2_576w

- [48] Hung-Yu Tseng, Hsin-Ying Lee, Lu Jiang, Ming-Hsuan Yang, and Weilong Yang. 2020. RetrieveGAN: Image synthesis via differentiable patch retrieval. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII* 16. Springer, 242–257.
- [49] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. 2018. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717* (2018).
- [50] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. 2023. Modelscope text-to-video technical report. *arXiv preprint arXiv:2308.06571* (2023).
- [51] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. 2023. VideoComposer: Compositional Video Synthesis with Motion Controllability. *arXiv preprint* (2023).
- [52] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yanan He, Jiahuo Yu, Peiqing Yang, et al. 2023. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103* (2023).
- [53] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2023. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*.
- [54] Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. 2023. Freeinit: Bridging initialization gap in video diffusion models. *arXiv preprint arXiv:2312.07537* (2023).
- [55] Katherine Xu, Lingzhi Zhang, and Jianbo Shi. 2024. Good Seed Makes a Good Crop: Discovering Secret Seeds in Text-to-Image Diffusion Models. *arXiv preprint arXiv:2405.14828* (2024).
- [56] Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and CUI Bin. 2024. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. In *Forty-first International Conference on Machine Learning*.
- [57] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. 2023. Rerender a video: Zero-shot text-guided video-to-video translation. In *SIGGRAPH Asia 2023 Conference Papers*.
- [58] Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. 2024. Space-Time Diffusion Features for Zero-Shot Text-Driven Motion Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8466–8476.
- [59] Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jia-Wei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. 2024. Motiondirector: Motion customization of text-to-video diffusion models. In *European Conference on Computer Vision*. Springer, 273–290.
- [60] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. 2024. *Open-Sora: Democratizing Efficient Video Production for All*. <https://github.com/hpcaitech/Open-Sora>