Video Motion Transfer with Diffusion Transformers

Alexander Pondaven[1] Aliaksandr Siarohin[2] Sergey Tulyakov[2] Philip Torr[1] Fabio Pizzati[1,3]
[1]University of Oxford [2]Snap Inc. [3]MBZUAI

ditflow.github.io

Figure 1. **Overview of DiTFlow.** We propose a motion transfer method tailored for video Diffusion Transformers (DiT). We exploit a training-free strategy to transfer the motion of a reference video (top) to newly synthesized video content with arbitrary prompts (bottom). By optimizing DiT-specific positional embeddings, we can also synthesize new videos in a zero-shot manner.

Abstract

We propose DiTFlow, a method for transferring the motion of a reference video to a newly synthesized one, designed specifically for Diffusion Transformers (DiT). We first process the reference video with a pre-trained DiT to analyze cross-frame attention maps and extract a patch-wise motion signal called the Attention Motion Flow (AMF). We guide the latent denoising process in an optimization-based, training-free, manner by optimizing latents with our AMF loss to generate videos reproducing the motion of the reference one. We also apply our optimization strategy to transformer positional embeddings, granting us a boost in zero-shot motion transfer capabilities. We evaluate DiTFlow against recently published methods, outperforming all across multiple metrics and human evaluation.

Corr.: pondaven@robots.ox.ac.uk fabio.pizzati@mbzuai.ac.ae

1. Introduction

Diffusion models have rapidly emerged as the global standard for visual content synthesis, largely due to their performance at scale. By scaling the model size, it has been possible to train on increasingly large datasets, even including billions of samples, incredibly boosting synthesis capabilities [4, 12, 61, 66]. This trend is especially pronounced in video synthesis, where generating realistic, frame-by-frame visuals with coherent motion relies heavily on extensive data and large models. In this context, a particularly promising development is the introduction of diffusion transformers (DiTs) [37]. Inspired by transformers, DiTs propose a new class of diffusion model that allows for improved scalability, ultimately achieving impressive realism in generation, as demonstrated by their adoption in many scale-oriented open source [53, 61] and commercial systems [34, 38].

However, realism alone is insufficient for real-world use of synthesized videos. *Control* over the generated video is essential for smooth integration into video creation and editing workflows. Most current models offer text-to-video (T2V) control through prompts, by synthesizing videos aligned with a user's textual description. However, this is rarely sufficient for achieving the desired result. While text may condition the appearance of objects in a scene, it is extremely challenging to control *motion i.e.* how the elements move in the scene, since text is inherently ambiguous when describing how fine-grained content evolves over time. To overcome this challenge, *motion transfer* approaches have used existing reference videos as a guide for the dynamics of the scene. The aim is to capture realistic motion patterns and transfer them to synthesized frames. However, most existing approaches are UNet-based [45] and do not take advantage of the superior performance of DiTs, which jointly process spatio-temporal information through their attention mechanism. We believe this opens up opportunities to extract high-quality motion information from the internal mechanics of DiTs.

In this paper, we propose DiTFlow, the first motion transfer method tailored for DiTs. We leverage the global attention token-based processing of the video, inherent to DiTs, to extract motion patterns directly from the analysis of attention blocks. With this representation, referred to as Attention Motion Flow (AMF), we are able to condition the motion of the synthesized video content, as we show in Figure 1. We exploit an optimization-based, training-free strategy, coherently with related literature [59, 62]. In practice, we optimize the latent representation of the video across different denoising steps to minimize the distance to a reference AMF. While employing a separate optimization process for each video yields the best performance, we also discover that optimizing the positional embeddings within DiTs enables the transfer of learned motion to new generations without further optimization, hence in a fully zero-shot scenario not previously possible with UNet-based approaches. This potentially lowers the computational cost of transferring motion on multiple synthesized videos. Overall, our novel contributions are the following:

1. We propose Attention Motion Flow as guidance for motion transfer on DiTs.
2. We propose an extension to our optimization objective in this DiT setting, demonstrating zero-shot motion transfer when training positional embeddings.
3. We test DiTFlow on state-of-the-art large-scale DiT baselines for T2V, providing extensive comparisons across multiple metrics and user studies.

2. Related works

Text-to-video approaches. Following the success of diffusion models [19, 48, 50, 51] in generating images from text [41, 42, 44], methods to handle the extra temporal dimension in videos were developed [2, 3, 5, 15, 20, 25]. These approaches commonly rely on the UNet [45] architecture with separate temporal attention modules operating on solely the temporal dimension for cross-frame consistency. Recently, Diffusion Transformer (DiT) based approaches for text-to-image (T2I) [6, 12, 37] and text-to-video (T2V) [4, 7, 16, 32, 33, 61, 66] have shown superior performance in quality and motion consistency. In particular, VDT [32] highlights the transformer's ability to capture long-range temporal patterns and its scalability.

Motion transfer. Motion transfer consists of synthesizing novel videos following the motion of a reference one. Unlike video-to-video translation [10, 31, 46], motion transfer approaches aim for complete disentanglement of the original video structure, focusing on motion alone. Some methods use training to condition on motion signals like trajectories, bounding boxes and motion masks [8, 11, 56, 57, 60, 63, 64], but this implies significant costs. Other approaches train motion embeddings [23] or finetune model parameters [15, 22, 58, 65]. However, these methods use separate attention for temporal and spatial information, making them unsuitable for DiTs. Optimization-based approaches extract a motion representation at inference [14, 21, 59, 62], which is more suitable for cross-architecture applications. TokenFlow [14] has a nearest-neighbor based approach on diffusion features, employing expensive sliding window analysis. SMM [62] employ spatial averaging, while MOFT [59] discover motion channels in diffusion features.

Attention control in diffusion models. Attention features containing semantic correspondences can be manipulated to control generation [13, 17, 40]. Video editing approaches modify the style or subject with feature injection [1, 28, 31, 55] or gradients [10, 36]. MotionClone [30] is a UNet-based method that transfers motion by computing a loss on attention. However, this assumes *separate temporal attention* with easily separable motion. This is unavailable for DiTs which use *full spatio-temporal attention* where disentangling motion patterns from content becomes more challenging and requires suboptimal adaptation.

3. Preliminaries

In this section, we introduce the basic formalism and concepts necessary for DiTFlow. We begin by reviewing the inference mechanics of T2V diffusion models (Section 3.1). We then introduce DiTs for video generation (Section 3.2).

3.1. Text-to-video diffusion models

Let us consider a pre-trained T2V diffusion model. We aim to map sampled Gaussian noise to an output video x_0 using a denoising network ϵ_θ over $t \in [0, T]$ denoising operations [19]. To reduce the computational cost of

multi-frame generation, video generators typically use Latent Diffusion Models (LDM) [44], which operate on latent video representations defined by a pre-trained autoencoder [27] with encoder $\mathcal{E}$ and decoder $\mathcal{D}$. We map a sampled Gaussian $z_T \sim \mathcal{N}(0, I)$ to $z_0 \in \mathbb{R}^{F \times C \times W \times H}$, where F, C, W, H represent the number of frames, latent channels, width, and height, respectively. Noisy latents z_t at each step t maintain the same shape until decoded to the output video $x_0 = \mathcal{D}(z_0)$. We formalize the basic denoising iteration as:

$$z_{\text{t-1}} = f(z_t, \epsilon_\theta(z_t, C, t)) \tag{1}$$

where C is the textual prompt describing the desired output video. This textual signal can condition the network using cross-attentions [44], or by being directly concatenated with the video latent representation [61]. The function f describes how noise is removed from z_t following a specific noise schedule over T steps [19, 49].

3.2. Video generation with DiT

Unlike U-Net diffusion models [44], DiT-based systems [37] treat the noisy latent as a sequence of tokens, taking inspiration from patching mechanisms typical of Vision Transformers [9]. The denoising network ϵ_θ is replaced with a transformer-based architecture. Latent patches of size $P \times P$ are encoded into tokens with dimension D and reorganized into a sequence of shape $(F \cdot \frac{H}{P} \cdot \frac{W}{P}) \times D$. The denoising network ϵ_θ is composed of N DiT blocks [37] consisting of multi-head self-attention [54] and linear layers. To encode positional information between patches during attention, a positional embedding ρ, consisting of values dependent on the patch location in the sequence, conditions the denoising network $\epsilon_\theta(z_t, C, t, \rho)$. Various position encoding schemes exist [47, 52, 54] where ρ is most commonly either added directly to patches at the start of ϵ_θ or augment the queries and keys at each attention block.

4. Method

Our core idea is to take advantage of the attention mechanism in DiTs to extract motion patterns across video frames in a zero-shot manner. Building on the intuition behind motion cue extraction discussed in Section 4.1, we then describe the creation of AMFs in Section 4.2. The AMF extracted from a reference video can be used as an optimization objective at a particular transformer block in the denoising network, as illustrated in Figure 2. We define how we use the AMF for optimization in Section 4.3. Note that the extracted motion patterns are independent from the input content, enabling the application of motion from a reference video to arbitrary target conditions.

4.1. Cross-frame Attention Extraction

We aim to extract the diffusion features of a reference video x_{ref} with a pretrained T2V DiT model in order to obtain a signal for motion. The motion of subjects in a video may be described by highly correlated content that changes spatially over time, so *tokens with similar spatial content will intuitively attend to each other across frames*. Hence, we can benefit from extracted token dependencies between frames to reconstruct how a specific element will move over time. We start by computing the latent $z_{\text{ref}} = \mathcal{E}(x_{\text{ref}})$ and pass it through the transformer at denoising step $t = 0$ with an empty textual prompt. Previous work [59] have observed a cleaner motion signal at lower denoising steps and we found $t = 0$ to be suitable for feature analysis of the input video. This also avoids the need for expensive DDIM inversion [49] for our task. Let us consider the n-th DiT block of ϵ_θ. The self-attention layer computes keys and queries for M attention heads. We average over the heads for feature extraction to reduce noise and memory consumption when optimizing. We denote K and Q as the keys and queries, averaged across heads, with shape $(F \cdot S, D_h)$, where $S = \frac{H}{P} \cdot \frac{W}{P}$ represents the spatial token length and $D_h = D/M$. We represent this operation as:

$$\{Q, K\} \overset{\text{n}}{\leftarrow} \epsilon_\theta(z_{\text{ref}}, \varnothing, 0, \rho) \tag{2}$$

Hence, for two frames $(i, j), i, j \in [1, F]$ we can calculate the *cross-frame attention* $A^{\otimes}_{i,j}$ as follows:

$$A^{\otimes}_{i,j} = \sigma\left(\tau \frac{Q_i K_j^T}{\sqrt{d_k}}\right) \in \mathbb{R}^{S \times S} \tag{3}$$

In Equation (3), Q_i and K_j refer to the query and key matrices of the i-th and j-th frames with shape $S \times D_h$. Here, σ is the softmax operation over the final dimension *i.e.* over tokens in the j-th frame, $\sqrt{d_k}$ is the attention scaling term [54] and τ is a temperature parameter. Intuitively, $A^{\otimes}_{i,j}$ encodes the relationship between patches of frames i and j from x_{ref}, serving as a signal to capture the reference video motion.

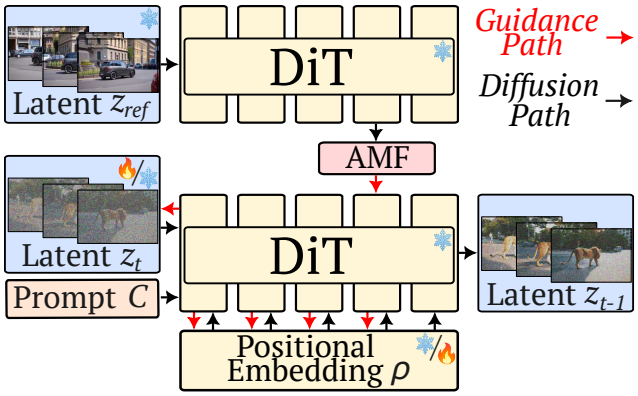


Figure 2. **Core idea of DiTFlow.** We extract the AMF from a reference video and we use that to guide the latent representation z_t towards the motion of the reference video. In our experiments, we also tested optimizing positional embeddings for improved zero-shot performance.

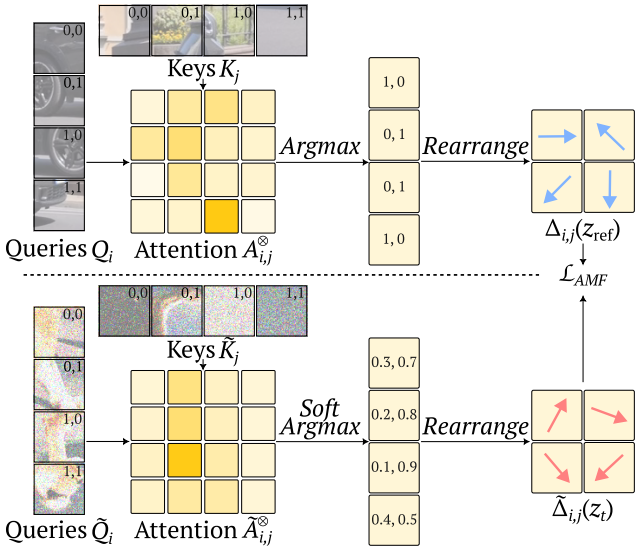


Figure 3. **Guidance.** We compute the reference displacement by processing cross-frame attentions with an argmax operation and rearranging them into displacement maps, identifying patch-aware cross-frame relationships. For video synthesis, we do the same operation with a soft argmax to preserve gradients, and impose reconstruction with the reference displacement.

4.2. Attention Motion Flows

We subsequently capture the AMF of a video by estimating a displacement map of spatial patches across all frame combinations. Each frame is composed of $\frac{H}{P} \times \frac{W}{P}$ patches. Our goal is to understand how the content of all patches in frame i moves to obtain frame j. To achieve this, we first process $A^{\otimes}_{i,j}$ using an argmax operation, which assigns each patch in frame i to the index of the most attended patch in frame j. We denote this result as $\hat{A}^{\otimes}_{i,j}$ where each entry $\hat{A}^{\otimes}_{i,j}[(u,v)]$ stores the assigned coordinate (u',v'). Empirically, selecting a single index using argmax results in a cleaner displacement map, leading to more reliable motion guidance. Using the obtained index pairs, we construct a patch displacement matrix $\Delta_{i,j}$ of size $S \times 2$ where $\Delta_{i,j}[(u,v)] = (u'-u, v'-v)$. This process is illustrated in Figure 3 (top). Finally, we aggregate the displacement matrices for all frame pairs (i,j) to construct the reference AMF, which serves as the motion guidance signal:

$$\text{AMF}(z_{\text{ref}}) = \{\Delta_{i,j} \mid i,j \in [1,F]\} \tag{4}$$

Our extracted AMF follows the idea of motion vectors used in MPEG-4 patch-based video compression [43], but is applied to DiT latent representations. In summary, for each patch, we calculate a motion vector in a two-dimensional coordinate space that indicates where the patch will move from frame i to frame j, effectively capturing the motion.

Algorithm 1 DiTFlow inference pipeline

Input: Reference video x_{ref}, trained DiT model ϵ_θ, encoder $\mathcal{E}$, decoder $\mathcal{D}$, prompt C, positional embedding ρ.
Output: Generated video x_0 with transferred motion

Extract latent representation: $z_{\text{ref}} \leftarrow \mathcal{E}(x_{\text{ref}})$
Compute attention: $\{Q, K\} \xleftarrow{n} \epsilon_\theta(z_{\text{ref}}, \varnothing, 0, \rho)$
for each (i,j) where $i,j \in [1,F]$ **do**
  Calculate cross-frame attention $A^{\otimes}_{i,j}$
  Construct displacement matrix $\Delta_{i,j}$
end for
Construct reference AMF: $\text{AMF}(z_{\text{ref}}) \leftarrow \Delta_{i,j}$
Initialize $z_T \sim \mathcal{N}(0, I)$
Initialize $\rho_T = \rho$
for denoising step $t = T$ to 0 **do**
  if $t > T_{\text{opt}}$ **then**
    for optimization step $k = 0$ to K_{opt} **do**
      Extract $\tilde{Q}$ and $\tilde{K}$: $\{\tilde{Q}, \tilde{K}\} \xleftarrow{n} \epsilon_\theta(z_t, C, t, \rho_t)$
      for each (i,j) where $i,j \in [1,F]$ **do**
        Calculate cross-frame attention $\tilde{A}^{\otimes}_{i,j}$
        Compute soft displacement matrix $\tilde{\Delta}_{i,j}$
      end for
      Construct $\text{AMF}(z_t) \leftarrow \tilde{\Delta}_{i,j}$
      Get $\mathcal{L}_{\text{AMF}} \leftarrow ||\text{AMF}(z_{\text{ref}}) - \text{AMF}(z_t)||_2^2$
      Update z_t or ρ_t by minimizing $\mathcal{L}_{\text{AMF}}$
    end for
  else
    $\rho_t = \rho$
  end if
  $z_{t-1} = f(z_t, \epsilon_\theta(z_t, C, t, \rho_t))$
end for
return $x_0 = \mathcal{D}(z_0)$

4.3. Guidance and Optimization

Once the reference AMF is obtained from x_{ref}, we use it to guide the generation of new video content with a T2V DiT model. Specifically, we aim to guide the denoising of $z_T \sim \mathcal{N}(0, I)$ in such a way that $x_0 = \mathcal{D}(z_0)$ reproduces the same motion patterns as x_{ref}. Here, we enforce guidance with an optimization-based strategy, aimed to reproduce the same extracted AMF at a given transformer block for the generated video in intermediate denoising steps t. For a given t, we consider key $\tilde{K}$ and query $\tilde{Q}$ extracted by the n-th DiT block while processing the input latent z_t:

$$\{\tilde{Q}, \tilde{K}\} \xleftarrow{n} \epsilon_\theta(z_t, C, t, \rho_t) \tag{5}$$

We follow the procedure described in Equation (3) to extract the corresponding cross-frame attention $\tilde{A}^{\otimes}_{i,j}$:

$$\tilde{A}^{\otimes}_{i,j} = \sigma\left(\tau \frac{\tilde{Q}_i \tilde{K}_j^T}{\sqrt{d_k}}\right) \tag{6}$$

We then calculate *soft* displacement matrices $\tilde{\Delta}_{i,j}$ as illustrated in Figure 3 (bottom). Rather than using argmax, we perform a weighted sum of attention values to identify continuous displacement values that preserve gradients where:

$$\tilde{\Delta}_{i,j}[(u,v)] = \sum_{(u',v')} \tilde{A}^{\otimes}_{i,j}[(u,v),(u',v')] \cdot (u'-u, v'-v) \tag{7}$$

We then build the soft AMF for the current step t as $\text{AMF}(z_t) = \{\tilde{\Delta}_{i,j} \mid i,j \in [1,F]\}$. Finally, we minimize the element-wise Euclidean distance between the AMF displacement vectors of the reference and current denoising step. This equates to minimizing the following loss:

$$\mathcal{L}_{\text{AMF}}(z_{\text{ref}}, z_t) = ||\text{AMF}(z_{\text{ref}}) - \text{AMF}(z_t)||_2^2 \tag{8}$$

We follow previous U-Net-based approaches [59, 62] and optimize z_t by backpropagating this loss in specific denoising steps. An alternative approach, benefiting from the DiT-specific presence of positional encodings, is to backpropagate the loss to optimize positional embeddings ρ_t. Note that positional embeddings are responsible for encoding the spatiotemporal locations of content. Intuitively, by manipulating the positional information of latent patches, we *guide the reorganization of patches* for motion transfer. This is also disentangled from latents that encode content. Hence, we empirically observed that while latent optimization leads to better overall performance, optimizing ρ_t enables better generalization of the learned embeddings to new prompts without repeating the optimization, allowing for fully zero-shot inference. In practice, we optimize z_t or ρ_t for K_{opt} steps, up until a given denoising step $T_{\text{opt}} \in [0,T)$ of the diffusion process, while denoising normally for the remaining steps. We report our full inference scheme in Algorithm 1.

5. Experiments

5.1. Experimental Setup

Dataset and metrics. For evaluation, we use 50 unique videos from the DAVIS dataset [39] coherently with the state-of-the-art [59, 62]. To allow for a fine-grained motion transfer assessment, we test each video with three different prompts in order of similarity from the original video: (1) a *Caption* prompt, created by simply captioning the video. This allows us to verify that the network disentangles the content from that of the original frames. (2) A *Subject* prompt, obtained by changing the subject while keeping the background the same. (3) Lastly, a *Scene* prompt, describing a completely different scene. This makes 150 motion-prompt pairs in total. For the evaluation, we use an *Image Quality* (IQ) metric for frame-wise prompt fidelity assessment based on CLIPScore [18] and a *Motion Fidelity* (MF) metric for motion tracklet consistency following [62]. The MF metric [62] compares the similarity of tracklets on x and x_{ref} obtained from off-the-shelf tracking [24].

Baselines and networks. For video synthesis, we use the state-of-the-art DiT CogVideoX [61] with both 2 billion (CogVideoX-2B) and 5 billion (CogVideoX-5B) parameter variants. We compare against four baselines. First, we present a *Backbone* method, which simply prompts the T2V model with C. This will act as a lower bound on MF. Following common practices [59], we define an *Injection* baseline injecting extracted attention features during inference. Specifically, we inject keys K and values V obtained by processing x_{ref} with the DiT at inference time. This loosely directs the structure of the synthesized scene, allowing it to roughly follow the spatial organization of elements in x_{ref} without DDIM inversion [49]. Given the simplicity and effectiveness of this technique, *we apply it to all baselines* except for the Backbone. We evaluate against three optimization-based guidance methods: SMM [62], MOFT [59] and MotionClone [30]. We adapt them to CogVideoX for a fair comparison. For SMM, we replace the expensive DDIM inversion operation, taking over one hour per video on CogVideoX-5B, with KV injection. We adapt MotionClone for DiTs by operating on the temporal axis of attention. For fairness, we normalize the number of optimization steps. Due to the unavailability of the evaluation videos and prompts used in related works [59, 62], we test all baselines on our DAVIS setup.

Inference settings We employ 50 denoising steps for all baselines and optimize for 5 steps using Adam [26] in the first 20% of denoising timesteps with a linearly decreasing learning rate following [62] from 0.002 to 0.001. We evaluate our AMF-based loss on the 20th block for CogVideoX-5B and the 15th for 2B. We apply KV injection on the first DiT block. Temperature τ is set to 2, emphasizing higher similarity tokens. On an NVIDIA A40 GPU, CogVideoX-2B generates a 21-frame video in around 3.5 minutes and 4 minutes with DiTFlow. CogVideoX-5B takes on average 5 minutes by default and 8 minutes with DiTFlow guidance. Unless explicitly specified, we optimize the latents z_t.

5.2. Benchmarks

Quantitative evaluation We evaluate the effectiveness of DiTFlow when optimizing z_t in order to directly compare our AMF-based guidance with baselines. In the results presented in Table 1, we consistently outperform all baselines. In particular, we considerably improve MF in both 5B and 2B models, scoring **0.785** and **0.726** respectively. In comparison, the best baseline, SMM, achieves **0.766** and **0.688**, demonstrating our superior capabilities to capture motion. Let us also highlight that when tested on *Subject* prompts, SMM on CogVideoX-5B reports considerably lower MF

Method	*CogVideoX-5B*								*CogVideoX-2B*							
	Caption		**Subject**		**Scene**		**All**		**Caption**		**Subject**		**Scene**		**All**	
	MF ↑	IQ ↑	MF ↑	IQ ↑	MF ↑	IQ ↑	MF ↑	IQ ↑	MF ↑	IQ ↑	MF ↑	IQ ↑	MF ↑	IQ ↑	MF ↑	IQ ↑
Backbone	0.524	0.315	0.502	**0.321**	0.544	0.318	0.523	0.318	0.521	0.313	0.495	0.312	0.523	0.314	0.513	0.313
Injection [59]	0.608	0.315	0.581	**0.321**	0.635	**0.320**	0.608	**0.319**	0.546	**0.315**	0.524	0.317	0.563	**0.321**	0.544	**0.318**
MotionClone [30]	0.635	0.313	0.640	**0.321**	0.628	**0.320**	0.634	0.318	0.564	0.303	0.557	0.304	0.574	0.301	0.565	0.303
SMM [62]	0.782	0.313	0.741	0.317	0.776	0.316	0.766	0.315	**0.687**	0.312	0.682	0.312	0.694	0.317	0.688	0.312
MOFT [59]	0.728	0.313	0.728	**0.321**	0.722	0.319	0.726	0.318	0.503	0.312	0.502	0.313	0.508	0.315	0.504	0.312
DiTFlow	**0.790**	**0.316**	**0.775**	**0.321**	**0.789**	0.319	**0.785**	**0.319**	0.685	0.311	**0.753**	**0.322**	**0.739**	0.320	**0.726**	0.317

Table 1. **Metrics evaluation.** We compare DiTFlow across 3 different caption setups (Caption, Subject, Scene) and against 4 baselines. We consistently score first or second in all metrics for almost all scenarios, advocating the quality of our motion transfer. Performance is consistent across two backbones with 5B and 2B parameters respectively. Best results are in **bold** and second best are underlined.

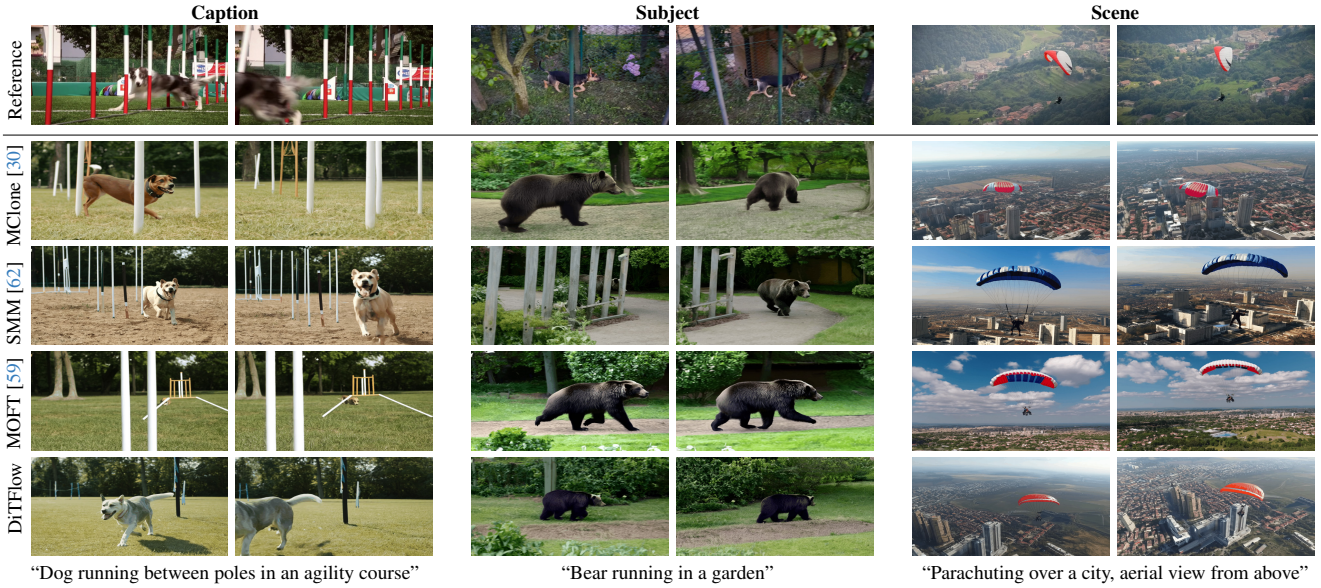


Figure 4. **Baseline comparison.** Baselines associate motion to wrong elements due to poor layout representation typical of UNet-based approaches that do spatial averaging or only consider deviations at each location. DiTFlow captures the spatio-temporal motion of each patch, resulting in correct spatial positioning and sizing of moving elements, *e.g.* the dog (left), the bear (middle), the parachute (right).

(**0.741**) with respect to performance on Caption (**0.782**) and Scene (**0.776**). We attribute this to the entanglement of the guidance signal with x_{ref}. Notably, SMM is based on spatially-averaged global features extracted from the reference video. This makes it challenging to tackle semantic modifications impacting only part of the scene, such as Subject prompts. This is less evident for CogVideoX-2B due to the inferior overall performance. Conversely, DiTFlow and MOFT preserve local guidance, allowing for a more fine-grained semantic control on generated scenes. We outperform MotionClone as it does not take full advantage of the spatiotemporal information in attention. We observe that IQ values exhibit lower variability compared to setups with heterogeneous backbones [62]. This shows that architecture is the main factor impacting image quality, further justifying our investigation of DiTs.

Qualitative comparison. We visually compare video generations of CogVideoX-5B against baselines in Figure 4. MotionClone has less fine-grained disentanglement of motion as it directly guides attention values, while *we guide the evolution of attentions over time*, regardless of their value. SMM suffers in Subject prompts, as observed quantitatively. SMM is based on a spatial averaging operation over features, which limits feature disentanglement as shown in the Subject column, where the rendered bars resemble those of the reference video. MOFT also tends to move the wrong elements in the scene as seen in the Caption example where the poles are moved instead of the dog. Their extracted MOFT feature guides the deviation of each spatial location from the temporal average, which can easily target the wrong content. DiTFlow, on the other hand, improves motion assignment by guiding the *explicit relationship of*

Figure 5. **Qualitative results of DiTFlow.** We are able to perform motion transfer in various conditions. Note how varying the prompt completely changes the scene's appearance while maintaining consistent motion. We map motion to correct elements even in cases where the motion changes drastically in positioning and size (bottom right).

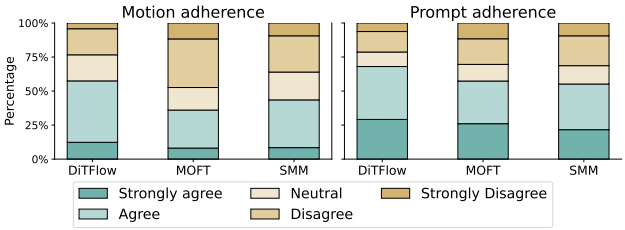
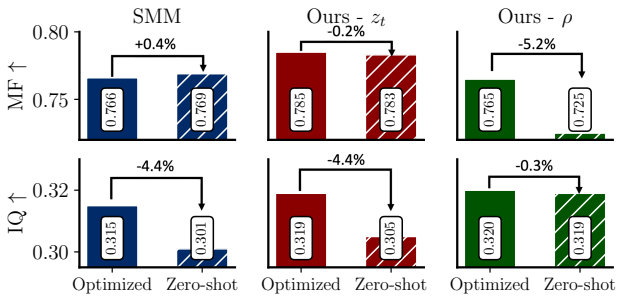


Figure 6. **Human evaluation.** We asked humans to evaluate agreement on the quality of generated samples in terms of motion (left) and prompt (right) adherence. DiTFlow consistently outperforms baselines in both evaluations.

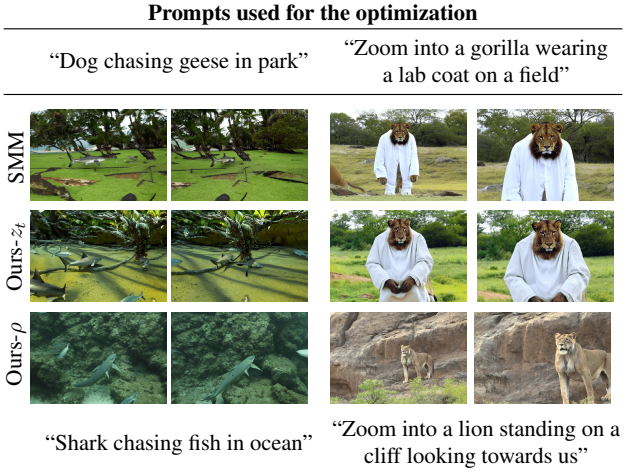
patches across both space and time through our AMF feature rather than guiding each location independently as in MOFT. Injection results are available in appendix.

We present additional qualitative results of DiTFlow in Figures 5 and 1 with realistic rendering of motion. We preserve motion in scenarios very different from the reference, such as in the top row. Motion is correctly mapped to specific elements in the scene even if they change drastically in size across frames, as in the challenging example in the bottom right. We attribute this to our patch-wise motion understanding, allowing it to capture fine-grained signals.

Human evaluation. We compare the best baselines on a study involving human judgment. In the first experiment (*Motion Adherence*), we show users the reference video together with videos synthesized by DiTFlow and the best baselines on CogVideoX-5B. We ask in a Likert-5 [29] preference scale how much they agree with the statement "The motion of the reference video is similar to the motion of the generated video". Users were instructed to focus on motion dynamics and ignore content differences. In the second experiment, users assessed *Prompt Adherence*, thus providing a video-level IQ score. They were asked to rate their agreement with the statement "The text accurately describes the video". We collect 66 preferences across 30 individuals, accounting for 1,980 unique answers. We report results in Figure 6. DiTFlow significantly outperforms both MOFT and SMM, consistent with the results in Table 1.



(a) Quantitative evaluation. SMM and Ours-z_t follow Table 1 (All)



(b) Qualitative evaluation

Figure 7. **Zero-shot performance.** In (a), we quantify zero-shot effectiveness. We compare performance by optimizing each prompt (Optimized) or using pre-optimized representations with new prompts (Zero-shot). Overall, optimizing ρ allows for better preservation of IQ. This results in better zero-shot disentanglement when changing the prompt, as shown in (b).

5.3. Zero-shot generation

We evaluate the zero-shot capabilities of running DiTFlow with our optimized ρ and a new prompt *without further optimization*. We quantify performance across all prompt categories. We optimize ρ on one prompt and infer with a different prompt. For instance, the optimized representation on the Caption prompt is injected into the generation of Subject and Scene prompts. Using the 5B model, we compare our methods to SMM [62], the best baseline method on MF in Table 1. In Figure 7a, we report average zero-shot performance across all prompts. We define Ours-z_t as DiTFlow with z_t optimization, where the prompt is changed after optimizing $z_{0...t}$. Ours-ρ is DiTFlow with positional embedding optimization following Section 4.3.

While optimizing z_t leads to better zero-shot motion preservation (**-0.2%**) compared to ρ (**-5.2%**), we report a significant drop in prompt adherence (**-4.4%** vs **-0.3%**) as seen in Figure 7b. Injecting the optimized z_t partially leaks the content of the optimization prompt. For example, elements of the lab coat prompt appear in the lion generation, and park features carry over into the ocean setting on the left. Optimizing ρ in the final row avoids this issue while rendering the desired motion, as the positional embedding has no impact on generated semantics, thus disentangling motion from content. Ultimately, with our findings on DiTs, we provide a novel adaptable strategy for prioritizing absolute performance through latent optimization or zero-shot capabilities using positional embeddings.

Block	MF ↑	IQ ↑
0	0.620	0.321
10	0.670	0.309
20	**0.797**	**0.313**
30	0.558	0.315

(a) Guidance block

T_{opt}	MF ↑	IQ ↑
0%	0.623	0.317
20 %	0.797	0.313
40 %	**0.813**	**0.314**
80 %	0.803	0.311
100 %	0.799	0.312

(b) Denoising steps

K_{opt}	MF ↑	IQ↑
1	0.769	0.318
5	0.797	0.313
10	**0.803**	**0.313**

(c) Optimization steps

Table 2. **Ablation studies.** We investigate our inference setups. In (a), we highlight that early blocks in DiTs contribute more to motion. In (b) and (c), we show that DiTFlow performance can be further boosted by increasing computational power.

5.4. Ablation studies

We ablate our design choices for DiTFlow on CogVideoX-5B. In Table 2, we show the impact of 1) the DiT block index where we optimize, 2) the denoising iterations with guidance and 3) the number of optimization steps by varying each in isolation. We run DiTFlow on 14 unique videos and corresponding prompts. In Table 2a, we show that the first blocks of CogVideoX-5B have incremental importance in motion guidance. We select the 20th block yielding best metrics. We notice that increasing the number of denoising (Table 2b) and optimization steps (Table 2c) positively impacts motion transfer at the cost of more computation. In particular, we report best performance for 40% of T (MF **0.813**) and 10 optimization steps (MF 0.803). Each of these setups double the optimization time for each generation, hence we selected 20% of T and 5 optimization steps for the best speed/quality tradeoff. Nevertheless, this enables improved motion transfer by investing in computation.

6. Conclusions

We present DiTFlow, the first DiT-specific method for motion transfer, based on our novel AMF formulation. Through extensive experiments and multiple evaluations, we demonstrate the effectiveness of DiTFlow against baselines in motion transfer, with extensions to zero-shot generation. Improved video action representations in DiTs may lower costs for generating robotic simulations and enable intuitive control of subject actions in content creation.

7. Acknowledgments

The authors extend their thanks to Constantin Venhoff and Runjia Li for their helpful feedback on the first draft. We appreciate the time put in by the participants of our user study. Alexander Pondaven is generously supported by a Snap studentship and the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems [EP/S024050/1]. We would also like to thank Ivan Johnson and Adam Lugmayer for their endless support with the compute server.

References

[1] Jianhong Bai, Tianyu He, Yuchi Wang, Junliang Guo, Haoji Hu, Zuozhu Liu, and Jiang Bian. Uniedit: A unified tuning-free framework for video motion and appearance editing. *arXiv preprint arXiv:2402.13185*, 2024. 2

[2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arxiv:2311.15127*, 2023. 2

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 2

[4] 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. Video generation models as world simulators, 2024. 1, 2

[5] 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. 2

[6] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2023. 2

[7] Shoufa Chen, Mengmeng Xu, Jiawei Ren, Yuren Cong, Sen He, Yanping Xie, Animesh Sinha, Ping Luo, Tao Xiang, and Juan-Manuel Perez-Rua. Gentron: Diffusion transformers for image and video generation. In *CVPR*, 2024. 2

[8] Zuozhuo Dai, Zhenghao Zhang, Yao Yao, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Animateanything: Fine-grained open domain image animation with motion guidance. *arXiv preprint arXiv:2311.12886*, 2023. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[10] Dave Epstein, Allan Jabri, Ben Poole, Alexei A. Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. In *NeurIPS*, 2023. 2

[11] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *ICCV*, 2023. 2

[12] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 1, 2

[13] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2023. 2

[14] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. In *ICLR*, 2024. 2, 1

[15] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024. 2

[16] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. *arXiv preprint arxiv:2312.06662*, 2023. 2

[17] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *ICLR*, 2023. 2

[18] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021. 5

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 3

[20] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *NeurIPS*, 2022. 2

[21] Teng Hu, Jiangning Zhang, Ran Yi, Yating Wang, Hongrui Huang, Jieyu Weng, Yabiao Wang, and Lizhuang Ma. Motionmaster: Training-free camera motion transfer for video generation. *arXiv preprint arXiv:2404.15789*, 2024. 2

[22] Hyeonho Jeong, Geon Yeong Park, and Jong Chul Ye. Vmc: Video motion customization using temporal attention adaption for text-to-video diffusion models. In *CVPR*, 2024. 2

[23] Manuel Kansy, Jacek Naruniec, Christopher Schroers, Markus Gross, and Romann M. Weber. Reenact anything: Semantic video motion transfer using motion-textual inversion. *arXiv preprint arXiv:2408.00458*, 2024. 2

[24] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *ECCV*, 2024. 5

[25] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *ICCV*, 2023. 2

[26] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 3

[28] Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. Anyv2v: A tuning-free framework for any video-to-video editing tasks. *arXiv preprint arXiv:2403.14468*, 2024. 2

[29] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932. 7

[30] Pengyang Ling, Jiazi Bu, Pan Zhang, Xiaoyi Dong, Yuhang Zang, Tong Wu, Huaian Chen, Jiaqi Wang, and Yi Jin. Motionclone: Training-free motion cloning for controllable video generation. *arXiv preprint arXiv:2406.05338*, 2024. 2, 5, 6

[31] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *CVPR*, 2024. 2

[32] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. Vdt: General-purpose video diffusion transformers via mask modeling. In *ICLR*, 2024. 2

[33] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arxiv:2401.03048*, 2024. 2

[34] 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. 1

[35] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, 2023. 2

[36] Saman Motamed, Wouter Van Gansbeke, and Luc Van Gool. Investigating the effectiveness of cross-attention to unlock zero-shot editing of text-to-video diffusion models. In *CVPR Workshops*, 2024. 2

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *CVPR*, 2023. 1, 2, 3

[38] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 1

[39] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2018. 5, 1

[40] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *ICCV*, 2023. 2

[41] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *ICML*, 2021. 2

[42] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arxiv:2204.06125*, 2022. 2

[43] Iain E Richardson. *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*. John Wiley & Sons, 2004. 4

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[46] Pratim Saha and Chengcui Zhang. Translation-based video-to-video synthesis. *arXiv prepring arXiv:2404.04283*, 2024. 2

[47] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL-HLT*, 2018. 3

[48] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ICML*, 2015. 2

[49] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3, 5

[50] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *arXiv preprint arXiv:1907.05600*, 2020. 2

[51] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 2

[52] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 2024. 3, 1

[53] Genmo Team. Mochi 1. `https://github.com/genmoai/models`, 2024. 1

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 1

[55] Wen Wang, Yan Jiang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2024. 2

[56] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. In *NeurIPS*, 2023. 2

[57] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *SIGGRAPH*, 2024. 2

[58] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, 2023. 2

[59] Zeqi Xiao, Yifan Zhou, Shuai Yang, and Xingang Pan. Video diffusion models are training-free motion interpreter and controller. *arXiv preprint arXiv:2405.14864*, 2024. 2, 3, 5, 6

[60] Jinbo Xing, Menghan Xia, Yuxin Liu, Yuechen Zhang, Yong Zhang, Yingqing He, Hanyuan Liu, Haoxin Chen, Xiaodong Cun, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Make-your-video: Customized video generation using textual and structural guidance. *IEEE TVCG*, 2024. 2

[61] 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 preprint arXiv:2408.06072*, 2024. 1, 2, 3, 5

[62] Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *CVPR*, 2024. 2, 5, 6, 8, 1

[63] Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023. 2

[64] Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. *arXiv preprint arXiv:2407.21705*, 2024. 2

[65] Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jiawei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. Motiondirector: Motion customization of text-to-video diffusion models. In *ECCV*, 2024. 2

[66] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024. 1, 2