Tora: Trajectory-oriented Diffusion Transformer for Video Generation

Zhenghao Zhang[1*], **Junchao Liao**[1*], **Menghao Li**[1], **ZuoZhuo Dai**[1],
Bingxue Qiu[1], **Siyu Zhu**[2], **Long Qin**[1], **Weizhi Wang**[1]

[1]Alibaba Group [2]Fudan University
projectpage: https://ali-videoai.github.io/tora_video/

Figure 1: Tora is capable of generating videos guided by trajectories, images, texts, or combinations thereof. Leveraging the scalability of DiT, the generated movement not only adheres precisely to the trajectory but also effectively emulates physical world dynamics. Notably, when generating videos at a 720p resolution, Tora maintains stable motion control for up to 204 frames. Due to limited space, we summarize the captions. Highly recommend viewing videos in project page.

Abstract

Recent advancements in Diffusion Transformer (DiT) have demonstrated remarkable proficiency in producing high-quality video content. Nonetheless, the potential of transformer-based diffusion models for effectively generating videos with controllable motion remains an area of limited exploration. This paper introduces Tora, the first trajectory-oriented DiT framework that concurrently integrates textual, visual, and trajectory conditions, thereby enabling scalable video generation with effective motion guidance. Specifically, Tora consists of a Trajectory Extractor (TE), a Spatial-Temporal DiT, and a Motion-guidance Fuser (MGF). The TE encodes arbitrary trajectories into hierarchical spacetime motion patches with a 3D video compression network. The MGF integrates the motion patches into the DiT blocks to generate consistent videos that accurately follow designated trajectories. Our design aligns seamlessly with DiT's scalability, allowing precise control of video content's dynamics with diverse durations, aspect ratios, and resolutions. Extensive experiments demonstrate Tora's excellence in achieving high motion fidelity, while also meticulously simulating the intricate movement of the physical world.

[*]These authors contributed equally.

Introduction

Diffusion models (Dhariwal and Nichol 2021; Ramesh et al. 2022) have demonstrated their capability to generate diverse and high-quality images or videos. Previously, video diffusion models (Ho et al. 2022b; Blattmann et al. 2023; Zhang et al. 2023a) predominantly employed U-Net architectures (Olaf Ronneberger 2015), focusing primarily on synthesizing videos of limited duration, typically around two seconds, and were constrained to fixed resolutions and aspect ratios. Recently, Sora (Brooks et al. 2024), a text-to-video generation model leveraging Diffusion Transformer (DiT) (Peebles and Xie 2023), has showcased video generation capabilities that significantly outstrip current state-of-the-art methods. Sora excels not only in the production of high-quality videos ranging from 10 to 60 seconds, but also distinguishes itself through its capacity to handle

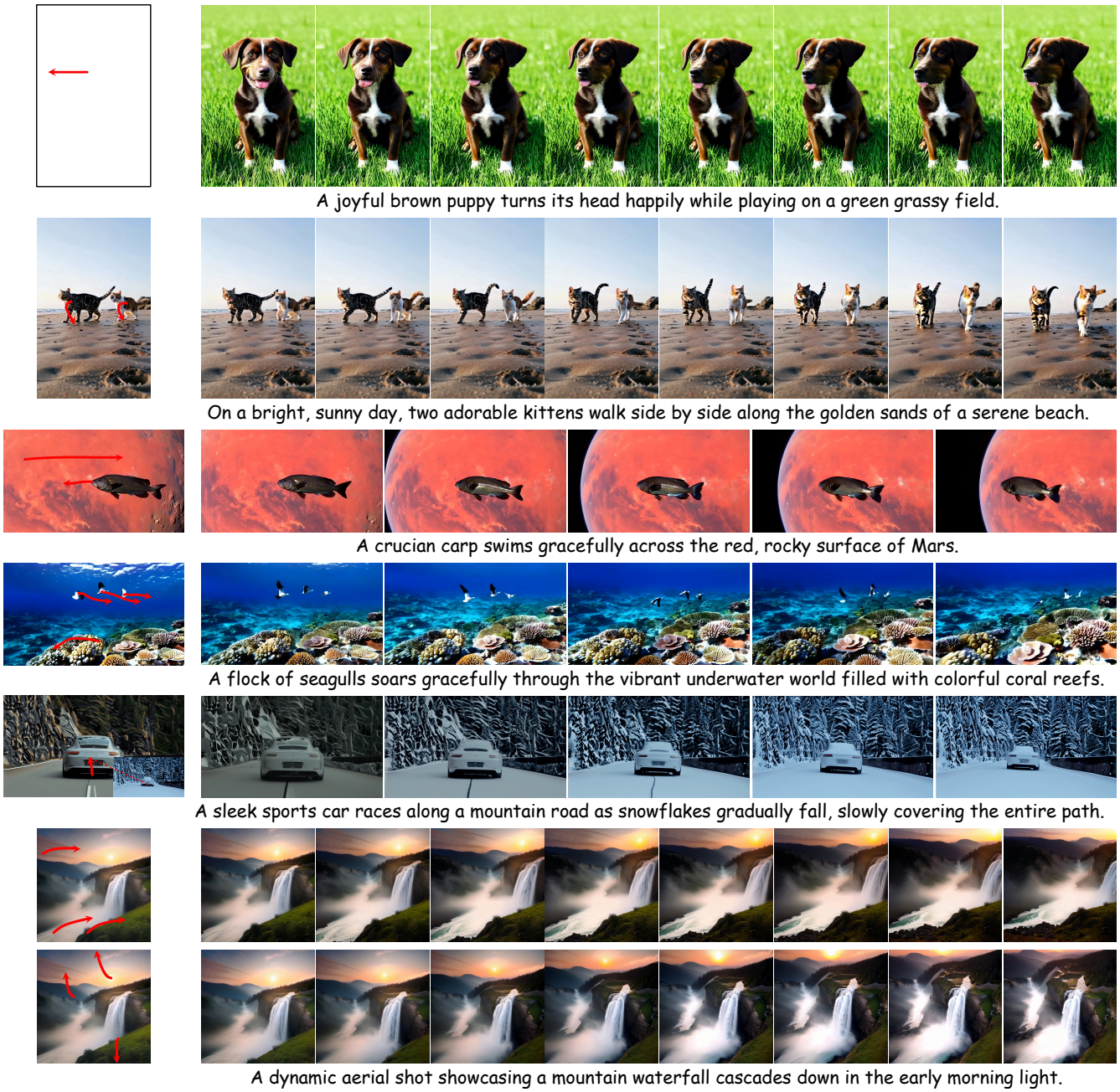


Figure 2: More generated samples. Tora accommodates diverse conditions such as text-only inputs, single starting frames, and combinations of initial and final frames (as illustrated in the fifth row). It effectively manages multiple trajectories, allowing for precise manipulation of several objects. Furthermore, Tora supports video generation across different aspect ratios, resolutions, and durations, ensuring flexible content creation. For video demonstrations, please refer to our project page.

diverse resolutions, various aspect ratios, adherence to the laws of actual physics.

Video generation requires consistent motion across image sequences, underscoring the importance of motion control. Previous works, such as VideoComposer (Wang et al. 2023) and DragNUWA (Yin et al. 2023), have implemented generalized motion manipulation through motion vectors and trajectories. Building on this foundation, MotionCtrl (Wang et al. 2024b) innovates by independently managing camera and object motions, thereby expanding the diversity of achievable motion patterns. Despite their promising controllable motion quality, U-Net methods are restricted to generating videos of only 16 frames at a fixed, lower resolution. This limitation hinders the smooth portrayal of motion, par-

ticularly during significant positional shifts in the provided trajectory, leading to distortion and unnatural movements, such as parallel drifting, which diverge from real-world dynamics. Consequently, there is an urgent need for a model capable of producing longer videos with robust motion control and detailed physical representations.

To address these challenges, we present Tora, the first DiT model that simultaneously integrates text, images, and trajectories, enabling scalable video generation with robust motion control. Notably, our work adopts OpenSora (Zheng et al. 2024), an open-source version of Sora, as the foundational DiT model. To align motion control with the scalability of the DiT framework, we propose two novel modules: the Trajectory Extractor (TE), which converts arbitrary trajectories into hierarchical spacetime motion patches, and the Motion-guidance Fuser (MGF), designed to seamlessly integrate these patches within the stacked DiT blocks. More specifically, TE initially converts positional displacements along trajectory into the RGB domain via flow visualization techniques. These visualized displacements undergo Gaussian filtering to mitigate scattered issues. Subsequently, a 3D Variational Autoencoder (VAE) (Kingma and Welling 2013) encodes trajectory visualizations into spacetime motion latents, which share the same latent space with video patches. The motion latents are then decomposed into multiple levels of motion conditions via stacked lightweight modules. Our VAE architecture is inspired by MAGVIT-v2 (Yu et al. 2023b) but simplified by foregoing codebook dependencies. The MGF integrates adaptive normalization layers (Perez et al. 2018) to infuse multi-level motion conditions into the corresponding DiT blocks. We explored various adaptations of transformer blocks including adaptive layer normalization, cross-attention, and extra channel connections to inject the motion conditions. Among these, adaptive layer normalization emerged as the most effective to generate consistent videos following trajectory.

During training, we adapt OpenSora's workflow to generate high-quality video-text pairs and utilize an optical flow estimator (Xu et al. 2023) for trajectory extraction. We also integrate a motion segmentor (Zhao et al. 2022) with a camera detector[1] to filter out instances dominated by camera motion, thereby improving our tracking of specific object trajectories. This careful selection process results in a dataset of 630k high-quality videos with consistent motion. With an adapter-like strategy (Mou et al. 2024), we solely train the temporal blocks, together with the TE and MGF. This strategy seamlessly integrates DiT's inherent generative knowledge with external motion signals.

The main contributions of our work are as follows:

- We introduce Tora, the first trajectory-oriented DiT model for scalable video generation with strong motion guidance. As illustrated in Figure 2, Tora seamlessly integrates various text, visual and trajectory instructions, enabling the creation of motion-manipulable videos.
- We propose a novel trajectory extractor and a motion-guidance fusion mechanism to facilitate motion control that aligns with the scalability of DiT. Additionally, we ablate several architecture choices and offer empirical baselines for future research.
- Experiments demonstrate that Tora is capable of generating 720p resolution videos with varying aspect ratios, extending up to 204 frames, all guided by the specified trajectories. Furthermore, it demonstrates superiority in simulating movements within the physical world.

[1]https://github.com/antiboredom/camera-motion-detector

Related Work

Diffusion models for Video Generation

Diffusion models have demonstrated an impressive capability to generate high-quality video samples. Previous research (Ho et al. 2022b,a; Singer et al. 2022; Khachatryan et al. 2023; Zhang et al. 2023b) commonly used video diffusion models (VDMs) that incorporated temporal convolutional and attention layers into the pre-trained image diffusion models. Subsequently, VideoCrafter (Chen et al. 2023) and SVD (Blattmann et al. 2023) expand the application of video diffusion models to larger datasets, while TF-T2V (Wang et al. 2024a) directly learn from extensive text-free videos. Nonetheless, these methods encounter limitations in generating long videos, owing to the inherent constraints on capacity and scalability within the U-Net design. Conversely, DiT-based models (Brooks et al. 2024; Zheng et al. 2024; Bao et al. 2024) can directly generate videos extending up to tens of seconds. Sora (Brooks et al. 2024) converts visual data into a unified representation, facilitating large-scale training and enabling the generation of 1-minute high-definition video. Vidu (Bao et al. 2024) is capable of generating both realistic and imaginative videos in various aspect ratios and resolutions. For our study, we adopt OpenSora (Zheng et al. 2024) as the foundational model, which is an open-source alternative to Sora.

Motion control in Video Generation

To better control motion in generated video, a multitude of studies have endeavored to introduce diverse motion signals in VDMs. Pioneering works like MotionDirector (Zhao et al. 2023) and VMC (Jeong, Park, and Ye 2024) have utilized reference videos to extract motion patterns applicable to diverse video generations. VideoComposer (Wang et al. 2023) expands upon this by adopting depth maps, sketches, or motion vectors from references as conditional inputs for motion control. Nonetheless, these methodologies are limited to reproducing existing motion patterns. Conversely, approaches that leverage trajectories or bounding boxes (Yin et al. 2023; Dai et al. 2023; Wang et al. 2024b) in video generation promise greater adaptability and user accessibility. DragNUWA (Yin et al. 2023) breaks new ground by integrating trajectory-based conditioning into VDMs, facilitating complex camera and object movements. AnimateAnything (Dai et al. 2023) employs motion masks for precise control over the moving regions. TrailBlazer (Wan-Duo Kurt Ma 2023), employs explicit attention mechanisms to maneuver generated objects along precise trajectories. MotionCtrl (Wang et al. 2024b) facilitates more flexible control, allowing separate adjustment of both camera movements and individual

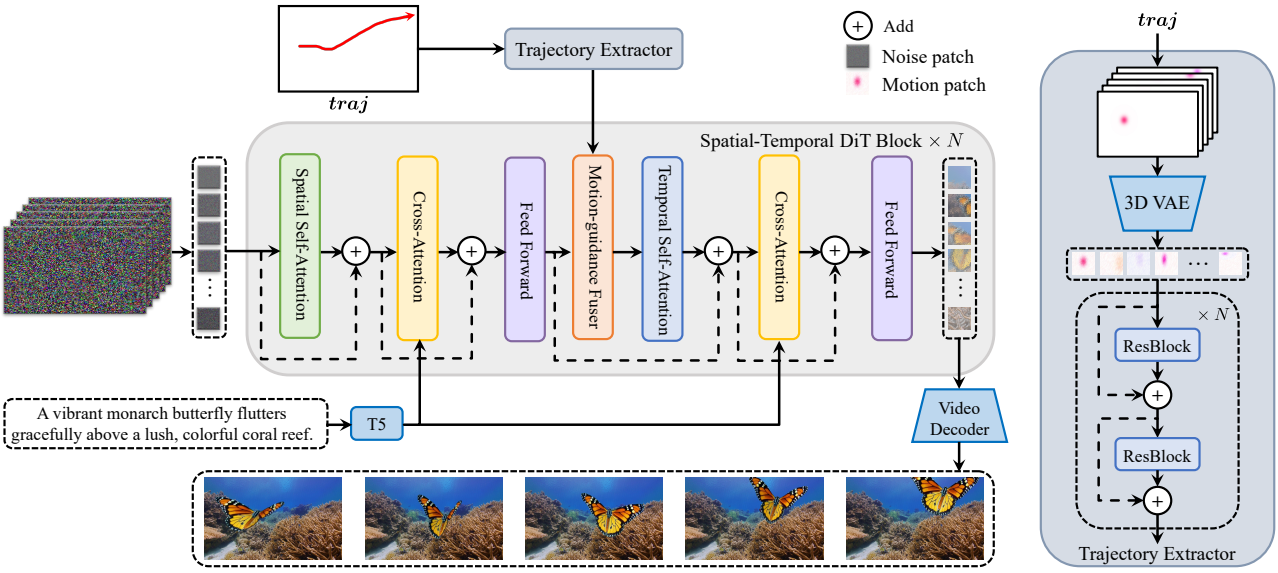


Figure 3: Overview of the Tora Architecture. We introduce two novel modules: the Trajectory Extractor and the Motion-guidance Fuser. The Trajectory Extractor uses a 3D motion VAE to embed trajectory vectors into the same latent space as video patches, preserving motion information across frames. It then employs stacked convolutional layers to extract hierarchical motion features. The Motion-guidance Fuser utilizes adaptive normalization layers to integrate these multi-level motion conditions into the corresponding DiT blocks, ensuring that generated videos consistently follow defined trajectories. Our method leverages the scalability of DiT, enabling the creation of motion-controllable videos of extended duration.

object motions. However, all of them yield noticeable artifacts in both motion consistency and visual presentation when applied to longer sequences. In contrast, our method first integrates trajectories into DiT architecture, specifically designed to accommodate scaling properties. This enables closer adherence to the physical world.

Methodology

Preliminary

Latent Video Diffusion Model (LVDM). The LVDM enhances the stable diffusion model (Ramesh et al. 2022) by integrating a 3D U-Net, thereby empowering efficient video data processing. This 3D U-Net design augments each spatial convolution with an additional temporal convolution and follows each spatial attention block with a corresponding temporal attention block. It is optimized employing a noise-prediction objective function:

$$l_\epsilon = ||\epsilon - \epsilon_\theta(z_t, t, c)||_2^2, \tag{1}$$

Here, $\epsilon_\theta(\cdot)$ signifies the 3D U-Net's noise prediction function. The condition c is guided into the U-Net using cross-attention for adjustment. Meanwhile, z_t denotes the noisy hidden state, evolving like a Markov chain that progressively adds Gaussian noise to the initial latent state z_0:

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{2}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t}(1 - \beta_t)$ and β_t is a coefficient that controls the noise strength in step t.

Diffusion Transformer (DiT). The DiT (Peebles and Xie 2023) introduces a novel architecture that merges the strengths of diffusion models with transformer architectures (Vaswani et al. 2017). This integration aims to address the limitations of traditional U-Net-based latent diffusion models (LDMs), improving their performance, versatility, and scalability. While keeping the overall framework consistent with existing LDMs, the key shift lies in replacing the U-Net with a transformer architecture for learning the denoising function $\epsilon_\theta(\cdot)$, thereby marking a pivotal advance in the realm of generative modeling.

Tora

Tora employs the Spatial-Temporal Diffusion Transformer (ST-DiT) from OpenSora as its foundational model. To facilitate user-friendly motion control while aligning with the scalability of DiT, Tora integrates two novel motion-processing components: the Trajectory Extractor (TE) and the Motion-guidance Fuser (MGF). An overview of Tora's workflow is illustrated in Figure 3.

Spatial-Temporal DiT. The ST-DiT architecture incorporates two distinct block types: the Spatial DiT Block (S-DiT-B) and the Temporal DiT Block (T-DiT-B), arranged in an alternating sequence. The S-DiT-B comprises two attention layers, each performing Spatial Self-Attention (SSA) and Cross-Attention sequentially, succeeded by a point-wise feed-forward layer that serves to connect adjacent T-DiT-B block. Notably, the T-DiT-B modifies this schema solely by substituting SSA with Temporal Self-Attention (TSA), preserving architectural coherence. Within each block, the input, upon undergoing normalization, is concatenated back to the block's output via skip-connections. By leveraging the

ability to process variable-length sequences, the denoising ST-DiT can handle videos of variable durations.

During processing, a video autoencoder (Yu et al. 2023a) is first employed to diminish both spatial and temporal dimensions of videos. To elaborate, it encodes the input video $X \in \mathbb{R}^{L\times H\times W\times 3}$ into video latent $z_0 \in \mathbb{R}^{l\times h\times w\times 4}$, where L denotes the video length and $l = L/4, h = H/8, w = W/8$. z_0 is next "patchified", resulting in a sequence of input tokens $I \in \mathbb{R}^{l\times s\times d}$. Here, $s = hw/p^2$ and p denotes the patch size. In both SSA and TSA, standard Attention is performed using Query (Q), Key (K), and Value (V) matrices:

$$Q = W_Q \cdot I_{norm}; K = W_K \cdot I_{norm}; V = W_V \cdot I_{norm}, \quad (3)$$

Here, I_{norm} is the normalized I, W_Q, W_K, W_V are learnable matrices. The textual prompt is embedded with a T5 encoder and integrated using a cross-attention mechanism.

Trajectory Extractor. Trajectories have proven to be a more user-friendly method for controlling the motion of generated videos. Specifically, given a trajectory $traj = \{(x_i, y_i)\}_{i=0}^{L-1}$, where (x_i, y_i) denotes the spatial position (x, y) at the i-th frame the trajectory passes through. Previous studies primarily encode the horizontal offset $u(x_i, y_i)$ and the vertical offset $v(x_i, y_i)$ as the motion condition:

$$u(x_i, y_i) = x_{i+1} - x_i;\ v(x_i, y_i) = y_{i+1} - y_i, \quad (4)$$

However, the DiT model employs a video autoencoder and a patchification process to convert the video into patches. Here, each patch is derived across multiple frames, rendering it inappropriate to straightforwardly employ frame-to-frame offsets. To address this, our TE converts the trajectory into motion patches, which inhabit the same latent space as the video patches. Particularly, we begin by transforming the $traj$ into a trajectory map $g \in \mathbb{R}^{L\times H\times W\times 2}$, enhanced with a Gaussian Filter to mitigate scatter. Notably, the first frame employs a fully-zero map. Afterward, the trajectory map g is translated into the RGB color space, producing $g_{vis} \in \mathbb{R}^{L\times H\times W\times 3}$ through a flow visualization technique. We use a 3D VAE to compress trajectory maps, achieving an 8x spatial and 4x temporal reduction, aligning with OpenSora framework. Our 3D VAE is based on the Magvit-v2 architecture, with spatial compression initialized using the VAE of SDXL (Podell et al. 2023) to accelerate convergence. We train the model using only reconstruction loss to obtain the compact motion latent representation $g_m \in \mathbb{R}^{l\times h\times w\times 4}$ from the g_{vis}.

To match the size of the video patches, we use the same patch size on g_m and encode it using a series of convolutional layers, resulting in spacetime motion patches $f \in \mathbb{R}^{l\times s\times d'}$. Here d' is the dimension of motion patches. The output of each convolutional layer is skip-connected to the input of the next layer to extract multi-level motion features:

$$f_i = Conv^i\ (f_{i-1}) + f_{i-1}, \quad (5)$$

where f_i is the motion condition for i-th ST-DiT block.

Motion-guidance Fuser. To incorporate DiT-based video generation with the trajectory, we explore three variants of fusion architectures that inject motion patches into each ST-DiT block. These designs are illustrated in Figure 4.

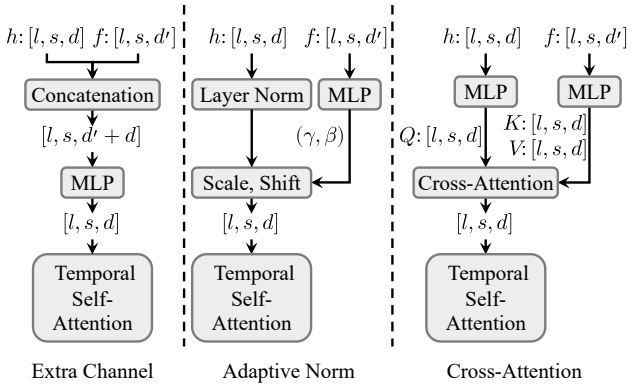


Figure 4: Different designs of the Motion-guidance Fuser for incorporating trajectory conditioning. Adaptive Norm demonstrates the best performance.

- Extra channel connections. Denote $h_i \in \mathbb{R}^{l\times s\times d}$ as the resultant output from the i-th block of the ST-DiT. Following the widespread use of concatenation in GAN-based LVDM, the motion patches are simply concatenated with the previous hidden state h_{i-1} along the channel dimension. An additional convolution layer is then added to maintain the same latent size:

$$h_i = Conv\ ([h_{i-1}, f_i]) + h_{i-1}, \quad (6)$$

- Adaptive Norm layer. Inspired by the adaptive normalization layers employed in GANs, we initially convert f_i into scale γ_i and shift β_i by adding two zero-initialized convolution layers into the ST-DiT block. Subsequently, γ_i and β_i are integrated into h_i through a straightforward linear projection:

$$h_i = \gamma_i \cdot h_{i-1} + \beta_i + h_{i-1}, \quad (7)$$

- Cross-Attention layer. The ST-DiT block has been modified to include an additional Cross-Attention layer following the SSA or TSA, with the motion patches serving as the key and value to integrate with the hidden state h:

$$h_i = CrossAttn\ ([h_{i-1}, f_i]) + h_{i-1}, \quad (8)$$

We evaluate three types of fusion architectures and find that the adaptive norm yields the best performance and computational efficiency. For the remainder of the paper, MGF employs the adaptive norm layer unless otherwise specified.

Data Processing and Training Strategy

Data Processing. We employ a structured data processing method to obtain high-quality training videos with consistent object motion. Initially, raw videos are segmented into shorter clips based on scene detection[2]. Subsequently, we remove invalid videos, such as those with encoding errors, a duration of zero, and low resolution. Furthermore, we utilize aesthetic[3] and optical flow scores (Xu et al. 2023) to filter out

[2]https://github.com/Breakthrough/PySceneDetect

[3]https://github.com/christophschuhmann/improved-aesthetic-predictor

Method	FVD (↓)			CLIPSIM (↑)			TrajError (↓)		
	16-frame	64-frame	128-frame	16-frame	64-frame	128-frame	16-frame	64-frame	128-frame
VideoComposer (Wang et al. 2023)	529	668	856	0.2335	0.2284	0.2236	15.11	29.14	58.76
DragNUWA (Yin et al. 2023)	475	593	784	0.2385	0.2341	0.2305	10.04	17.33	41.25
AnimateAnything (Dai et al. 2023)	487	602	775	0.2399	0.2342	0.2313	13.39	27.28	51.33
TrailBlazer (Wan-Duo Kurt Ma 2023)	459	581	756	0.2403	0.2351	0.2322	11.68	19.47	44.10
MotionCtrl (Wang et al. 2024b)	463	572	731	0.2412	0.2376	0.2331	9.42	16.46	38.39
Tora(Ours)	**438**	**460**	**494**	**0.2447**	**0.2435**	**0.2418**	**7.23**	**8.45**	**11.72**

Table 1: Quantitative comparisons with state-of-the-art motion-controllable video generation models. As the number of generated frames increases, Tora demonstrates a growing performance advantage over the UNet-based methods, maintaining a high degree of stability in trajectory control.

low-quality videos. To concentrate on the motion of primary objects, we implement camera motion filtering, using results from motion segmentation (Zhao et al. 2022) and camera detection to exclude instances predominantly exhibiting camera movement. Dramatic object motions in certain videos can lead to significant optical flow deviations, which may interfere with trajectory training. To address this, we retain these videos based on a probability of $(1 - score/100)$. For eligible videos, we generate captions using the PLLaVA model (Xu et al. 2024). During inference, we utilize GPT-4o (OpenAI 2023) to refine prompts, ensuring alignment with training process for consistency. More detailed information about the filtering and the refinement process can be found in the supplementary materials.

Motion condition training. Inspired by DragNUWA and MotionCtrl, we adopt a two-stage training approach for trajectory learning. In the first stage, we extract dense optical flow (Xu et al. 2023) from the training video as the trajectory, providing richer information to enhance motion learning. In the second stage, we adjust the model from complete optical flow to more user-friendly trajectories by randomly selecting 1 to N object trajectories based on motion segmentation results and flow scores. To improve the scattered nature of sparse trajectories, we apply a Gaussian filter for refinement. After completing the two-stage training, Tora facilitates flexible motion control using arbitrary trajectories.

Image condition training. We follow the mask strategy employed by OpenSora to support visual conditioning. Specifically, we randomly unmask frames during training, and the video patches of the unmasked frames are not subjected to any noise. This enables our Tora model to seamlessly integrate text, images, and trajectories into a unified model.

Experiments

Experimental Setup

Implementation Details. Tora is initialized with OpenSora v1.2 weights, and training videos have resolutions from 144p to 720p and frame counts ranging from 51 to 204. To balance training FLOP and memory usage, we adjust the batch size from 1 to 50. We use Adam Optimizer (Kingma and Ba 2015) with a learning rate of 2×10^{-5} on 4 NVIDIA A100. The 3D VAE in TE is initially trained on datasets (Mehl et al. 2023; Mayer et al. 2016; Ranjan et al. 2020; Cabon, Murray, and Humenberger 2020) annotated with optical flow and then frozen during Tora training. We train Tora for 2 epochs with dense optical flow and fine-tune for 1 epoch with sparse trajectories. The maximum number of sampling trajectories N is set to 16. The inference step and the guidance scale are set to 30 and 7.0, respectively.

Dataset. Our training videos are sourced from three datasets: 1) Panda-70M (Chen et al. 2024), from which we use the training-10M subset containing high-quality videos; 2) Mixkit (Envato 2024); and 3) Internal videos. The internal videos are manually annotated, with each clip labeled to include object masks and camera movement. Following our data processing pipeline, we select about 630k eligible videos for the training dataset. For inference, we curate 185 clips with diverse motion trajectories and scenes, to serve as a new benchmark for evaluating the motion controllability.

Metrics. We leverage standard metrics such as Fréchet Video Distance (FVD) (Unterthiner et al. 2018), and CLIP Similarity (CLIPSIM) (Wu et al. 2021) to quantitatively evaluate video quality. For assessing motion controllability, we leverage Trajectory Error (TrajError), which computes the average distance between the generated and pre-defined trajectories. Human evaluation is also introduced in supplementary materials due to space limitations.

Results

We compare our method with popular motion-guided video generation approaches in three settings: 16, 64, and 128 frames, all at a resolution of 512 × 512 for a fair evaluation. The provided trajectories are adjusted to fit the different video lengths. For most U-Net-based methods, we use sequenced inference, where the last frame generated from one batch serves as the visual condition for the next, aligning with their inference strategies. As shown in Table 1, in the 16-frame setting typical for U-Net methods, MotionCtrl and DragNUWA align better with the provided trajectories but still fall short compared to our proposed Tora. With an increasing number of frames, U-Net methods exhibit significant deviations, leading to misalignment errors that result in deformations, motion blur, or object disappearance in later sequences. In contrast, Tora remains highly robust across varying frame counts due to the transformer's scaling abilities. In the 128-frame test setting, Tora's trajectory accuracy surpasses other methods by 3 to 5 times, showcasing its outstanding motion control capabilities. Figure 5 presents

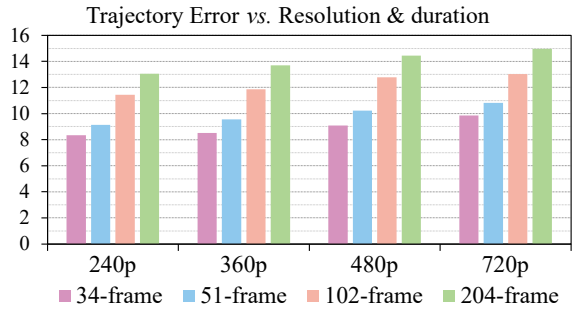


Figure 5: Comparison of Trajectory Error across various resolutions and durations. Unlike U-Net models, our method maintains motion control with a gradual increase in error.

an analysis of Trajectory Error across different resolutions and durations. Unlike U-Net models, which exhibit substantial trajectory errors over time, Tora shows only a gradual increase in error as duration extends. This aligns with the decrease in video quality observed in the DiT model. The results demonstrate that our method effectively maintains trajectory control over longer durations.

Ablation study

We conduct several ablation studies to analyze the effects of our design choices. All models are evaluated using 480p resolution, a 16:9 aspect ratio, and 204 frames.

Trajectory Compression. To integrate the trajectory vector into the same latent space as video patches, we explore three different methods for trajectory compression, as summarized in Table 2. The first method samples the mid-frame as a keyframe for successive 4-frame intervals and uses patch-unshuffle for spatial compression. While simple, this approach is sub-optimal for motion control due to potential flow estimation errors during rapid movements or occlusions, and the increased dissimilarity between patches complicates learning. The second method employs average pooling to gather information from successive frames. Although this captures a general sense of movement, it sacrifices precision by averaging the trajectory's direction and magnitude, diluting important motion details. To better preserve trajectory information between consecutive frames, we utilize a 3D VAE to extract the global context of successive trajectory intervals. The trajectory data is converted into RGB images to leverage existing 3D VAE weights. Extensive training on a large dataset of trajectory videos with this method yields the best results, highlighting the effectiveness of our customized 3D VAE approach for trajectory compression.

Method	FVD (↓)	CLIPSIM (↑)	TrajError (↓)
Sampling Frame	581	0.2304	27.61
Average Pooling	558	0.2325	20.97
3D VAE	513	0.2358	14.25

Table 2: Evaluations of the impact of different trajectory compression methods.

Block design and integrated position of MGF. We train the three variant MGF blocks as previously described, with the results presented in Table 3. Notably, the adaptive norm block achieves lowest FVD and Trajectory Error, while also exhibiting the highest computational efficiency. This advantage stems from its ability to dynamically adapt features based on varying conditions without needing strict alignment, a common challenge with cross-attention. Additionally, it maintains temporal consistency by modulating conditional information over time, which is essential for incorporating motion cues. In contrast, channel concatenation can lead to information congestion, making motion signals less effective. We find that initializing the normalization layer as the identity function is vital for optimal performance. Additionally, placing the MGF module within the Temporal DiT block significantly enhances trajectory motion control, evidenced by a drop in Trajectory Error from 23.39 to 14.25.

Method	FVD (↓)	CLIPSIM (↑)	TrajError (↓)
Extra Channel	542	0.2329	21.07
Cross Attention	526	0.2354	18.36
Adaptive Norm	513	0.2358	14.25

Table 3: Different variants of motion fusion blocks employed in MGF. Adaptive Norm works best.

Training Strategies. We evaluate the two-stage training approach, with results in Table 4. Training only with dense optical flow is ineffective, as it fails to capture the details of sparse trajectories. Conversely, using only sparse trajectories provides limited information, complicating the learning process. By first training with dense flows and then fine-tuning with sparse trajectories, our model demonstrates better adaptability and versatility in managing various motion patterns, leading to improved overall performance.

Motion-guidance	FVD (↓)	CLIPSIM (↑)	TrajError (↓)
Dense Flow	601	0.2307	39.34
Sparse Flow	556	0.2334	24.73
Hybrid	513	0.2358	14.25

Table 4: Ablation of the type of training trajectories. "Hybrid" denotes the two-stage training strategy.

Conclusion

This paper introduces Tora, the first trajectory-oriented Diffusion Transformer framework for video generation. Tora effectively encodes arbitrary trajectories into spacetime motion patches, which align with the scaling properties of DiT, thereby enabling more realistic simulations of physical world movements. By employing a two-stage training process, Tora achieves motion-controllable video generation across a wide range of durations, aspect ratios, and resolutions. Remarkably, it can generate high-quality videos that adhere to specified trajectories, producing up to 204 frames at 720p resolution. This capability underscores Tora's versatility and robustness in handling diverse motion patterns

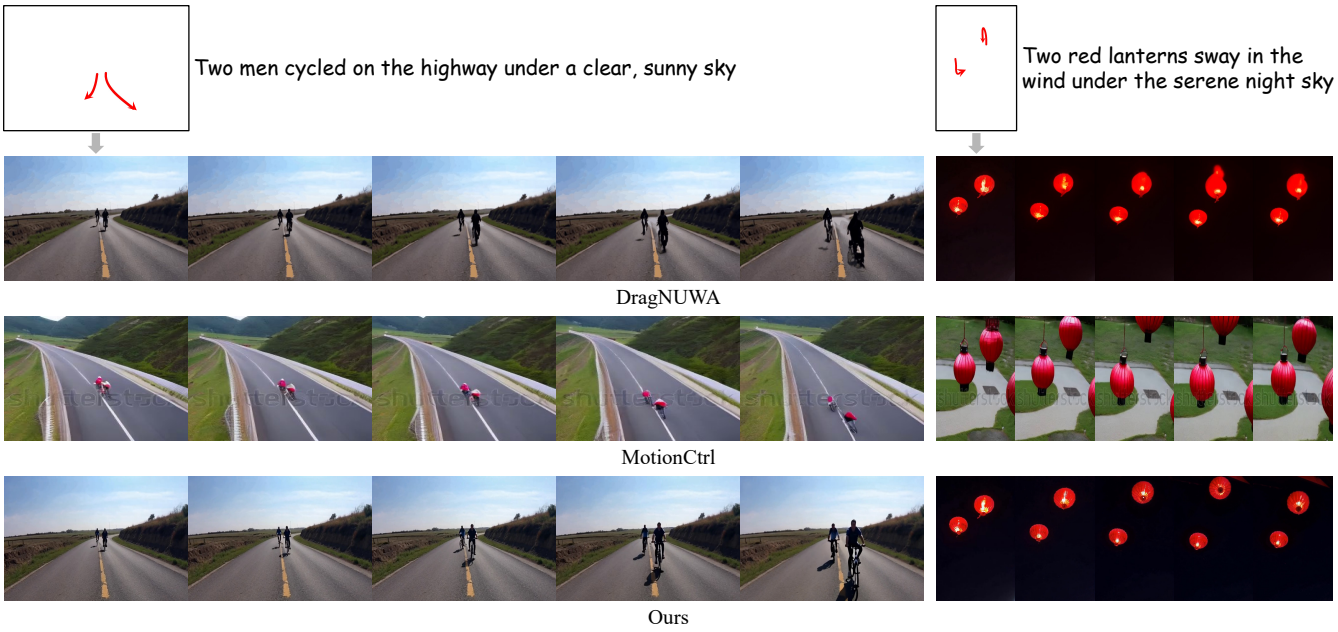


Figure 6: Qualitative Comparisons on Trajectory Control. All methods are capable of generating objects that follow the given trajectory. However, Tora not only adheres precisely to the specified trajectory but also produces smoother movement that conforms to the physics world.

while maintaining high visual fidelity. We hope our work provides a strong baseline for future research in motion-guided Diffusion Transformer methods.

Appendix

This supplementary material offers additional results, comprehensive dataset information, and thorough analyses that bolster the findings and conclusions outlined in the main text. It is organized as follows:

- Additional qualitative results.
- Data pre-processing method.
- Dataset details, regarding total quantity, total durations, etc.
- Prompt refinement method.
- Motion VAE training.

Qualitative Comparisons

While the main text focuses on quantitative comparisons with the motion-controllable video generation models and ablation studies on different designs for TE and MGF, here we provide further visual comparisons.

Compare with motion-controllable methods

Figure 6 provides a comparative analysis of our proposed method against mainstream motion control techniques. In the first scenario involving the coordinated movement of two individuals, all methods manage to generate motion trajectories that are relatively accurate. However, our approach stands out for its superior visual quality. This advantage is largely attributed to the use of longer sequence frames, which contribute to smoother motion trajectories and more realistic background rendering. For example, in our generated bicycle scenario, the human legs exhibit lifelike pedaling motions, while the output from DragNUWA shows legs that appear to float almost horizontally, compromising physical realism. Moreover, both DragNUWA and MotionCtrl encounter significant motion blur towards the end of their videos. Additionally, MotionCtrl introduces unintended camera movements during the riding sequence, despite the absence of any intended camera movement conditions. In another instance, DragNUWA suffers from severe deformation of the lantern as the provided trajectory oscillates up and down. Although MotionCtrl's trajectory is relatively accurate, the resulting video does not align with the expected portrayal of two lanterns. Overall, our method not only adheres closely to the provided trajectories but also minimizes object deformation, thereby ensuring higher fidelity in motion representation.

Compare with OpenSora

Despite OpenSora's impressive accomplishments, it faces challenges when creating long videos featuring complex motions, such as simultaneous movement of multiple objects, swinging, or circling. This often leads to incoherent or distorted foreground objects, negatively impacting visual quality. To our delight, we discovered that incorporating appropriate trajectory control into the DiT model offers a more effective constraining signal. This improvement markedly enhances video fluidity and preserves object fi-

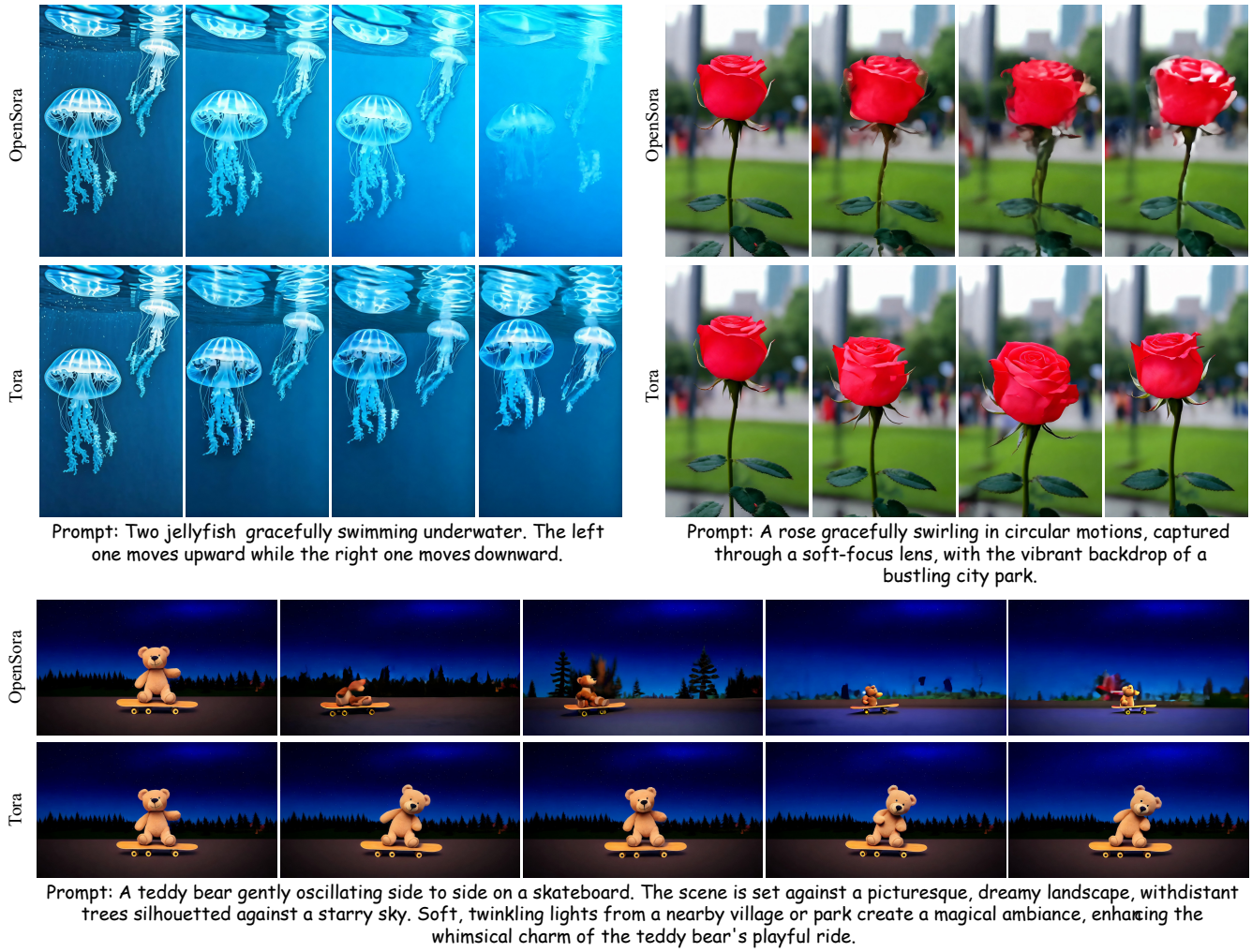


Figure 7: Qualitative comparison between Tora and OpenSora. All results are generated under the same text and image conditions. Tora employs an appropriate trajectory that simulates real-world physics, leading to more coherent and stable motion.

delity, as demonstrated in Figure 7.

In scenarios where a teddy bear is oscillating side to side on a skateboard or a rose is swirling in circular motions, OpenSora, which relies solely on textual directives for motion control, exhibits noticeable object deformations. In contrast, Tora excels at maintaining the inherent shape of the objects. Additionally, when managing the motion of multiple entities, such as a pair of jellyfish—one moving upward while the other moves downward, OpenSora demonstrates noticeable flickering, underscoring its limitations in handling complex movements. In conclusion, the integration of Tora's motion signaling mechanism enhances both the controllability and stability of the synthesized videos.

Comparison of Different Trajectory Compression Methods

We train our proposed trajectory extractor using the various trajectory compression methodologies previously discussed. The comparisons of these methods are visually illustrated in Figure 8.

In key-frame sampling, while it successfully captures essential motion, it frequently leads to misalignment between video patches and motion patches, especially during rapid motion sequences. This misalignment hinders the generated objects from accurately tracking their trajectories, negatively impacting visual fluidity and overall quality. On the other hand, average pooling smooths out minor variations, resulting in a more consistent motion representation. However, in complex trajectories, such as S-shaped turns where consecutive frame directions are inconsistent, this approach may introduce artifacts because the physical relevance of optical flow decreases. In contrast, our proposed 3D VAE approach effectively compresses trajectory information into the video's latent space. By training the 3D VAE on the large dataset with flow annotations, it successfully extracts the most relevant motion features for guidance, preserving the movement of successive frames to a significant extent. As evidenced in the results, this method significantly enhances

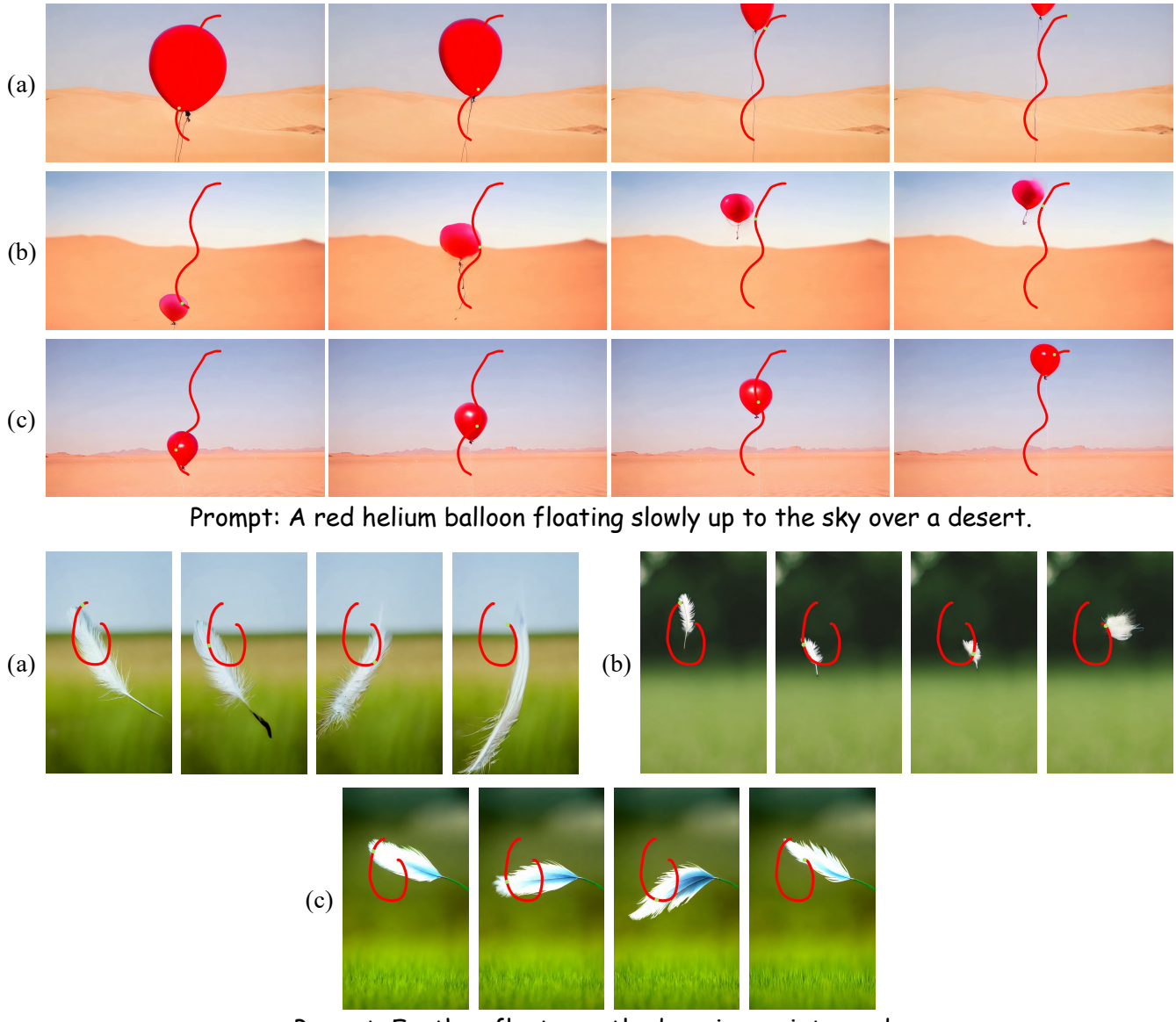


Figure 8: Generated videos employing different trajectory compression methods: (a) Sampling Keyframe; (b) Average Pooling; (c) 3D VAE.

the fluidity and coherence of the generated movements, producing visually compelling sequences that closely resemble natural motion.

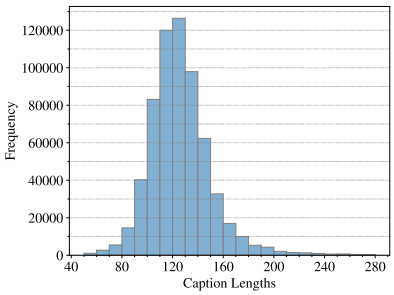
Data Pre-processing

During the processing of the video datasets, constructing a high-quality training set is crucial as it significantly impacts the quality of the generated videos. The following is a detailed description of our data processing workflow, which includes steps such as invalid videos removal, resolution filtering, camera motion filtering, and assessing the degree of object motion.

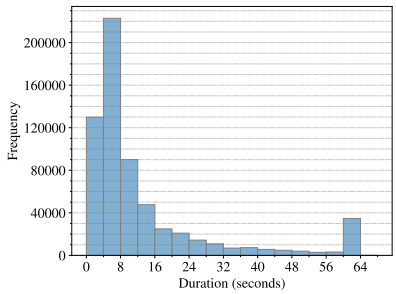
Initially, during the dataset preparation phase, we remove invalid videos. This step aims to identify and discard videos that do not meet our established criteria, including those with encoding errors, a duration of zero, or low quality. We identify encoding errors and zero-duration videos by directly decoding them. Furthermore, we predict both the aesthetic score[4] and the optical flow score (Xu et al. 2023) for each video. A video is deemed valid only if its aesthetic score exceeds 5.5 and its flow score is greater than 2.

Next, we perform resolution filtering. To ensure the effectiveness of subsequent study, we establish a minimum reso-

[4]https://github.com/christophschuhmann/improved-aesthetic-predictor



(a) Histogram of Caption Lengths.



(b) Histogram of Video Durations.

resolution \ # frames	51	102	204	408
144p	50	25	12	6
240p	20	10	5	2
360p	8	4	2	1
480p	4	2	1	-
720p	2	1	-	-

(c) The training batch size of every bucket (resolution, duration).

Figure 9: Overview of the training data distributions and batch sizes.

lution standard of 720p. By checking the resolution of each video, we can eliminate those that fall below this threshold, thereby ensuring that the videos in our dataset possess adequate clarity and detail.

Subsequently, we perform camera motion filtering using a camera motion detector[5] and a motion segmentor (Zhao et al. 2022) to filter out videos with significant camera movement, which may distort the model's ability to focus on the motion of the primary subjects. More specifically, the zoom detection threshold is set between 0.4 and 0.6. The detected camera movement angles, calculated based on the background from the motion segmentation results, are valid as follows:$[0°, 20°], [160°, 200°], [340°, 360°]$.

Finally, we analyze the magnitude score of the optical flow within the foreground, excluding those scenes that are mostly static or exhibit minimal movement. Moreover, dramatic object motions in some videos can cause significant optical flow deviations, interfering with trajectory training. Consequently, we retain these videos with a probability of $(1 - score/100)$.

Through these rigorous filtering and processing steps, we successfully construct a high-quality video dataset suitable for subsequent training, providing a solid foundation for our study.

Dataset Details

This section offers an overview of the dataset used in this study, covering its origin and composition. We employ histograms and descriptive statistics to illustrate the dataset's structure and distribution.

Training Data

The video data is sourced from the Panda-70M subset, Mixart, and internal videos. We initially collect 2.6M videos and apply the data preprocessing pipeline to filter the content, resulting in 631k eligible videos for training. An overview of the training dataset is presented in Table 5, which details the durations, resolution and FPS.

Additionally, Table 6 summarizes the mean and standard deviation for the durations, number of frames, and caption lengths. We also present histogram to show the distribution of the caption lengths and the durations of all video clips, as shown in the Figure 9a and Figure 9b.

[5]https://github.com/antiboredom/camera-motion-detector

# Videos Clips	631053
Total Durations (hours)	2952.93
Average Shorter Edge Length	965.11
Average FPS	29.23

Table 5: Statistical information about the training data.

	mean	std
Durations (seconds)	16.85	19.58
#Frames	506.22	644.38
Caption Length (#word)	125.52	24.22

Table 6: Statistics of training set, regarding durations, number of frames, and caption lengths.

Drawing inspiration from OpenSora, we employ a multi-scale and mixed-duration training strategy, which involves training videos of various resolutions and lengths together. Specifically,we establish predefined buckets, each defined by a unique combination of (video resolution, duration). Videos are then assigned to the appropriate bucket according to their specific attributes. Note that videos of any aspect ratio will fall into these buckets if their total pixel count is within the specified statistical intervals. The parameter settings for the buckets adhere to the principle that lower resolutions correspond to longer durations, enabling Tora to adapt to videos of varying lengths. Notably, our preprocessing steps ensure that the shorter edge of each training video exceeds 720 pixels. To enable training across various scales, we shuffle the dataset and randomly select videos for downsampling to lower resolutions. Additionally, we employ different batch sizes for each bucket to balance the GPU load. The details of the buckets are presented in Figure 9c.

Evaluation Data

Our evaluation dataset is primarily sourced from video object segmentation datasets (Xu et al. 2021; Qi et al. 2022;

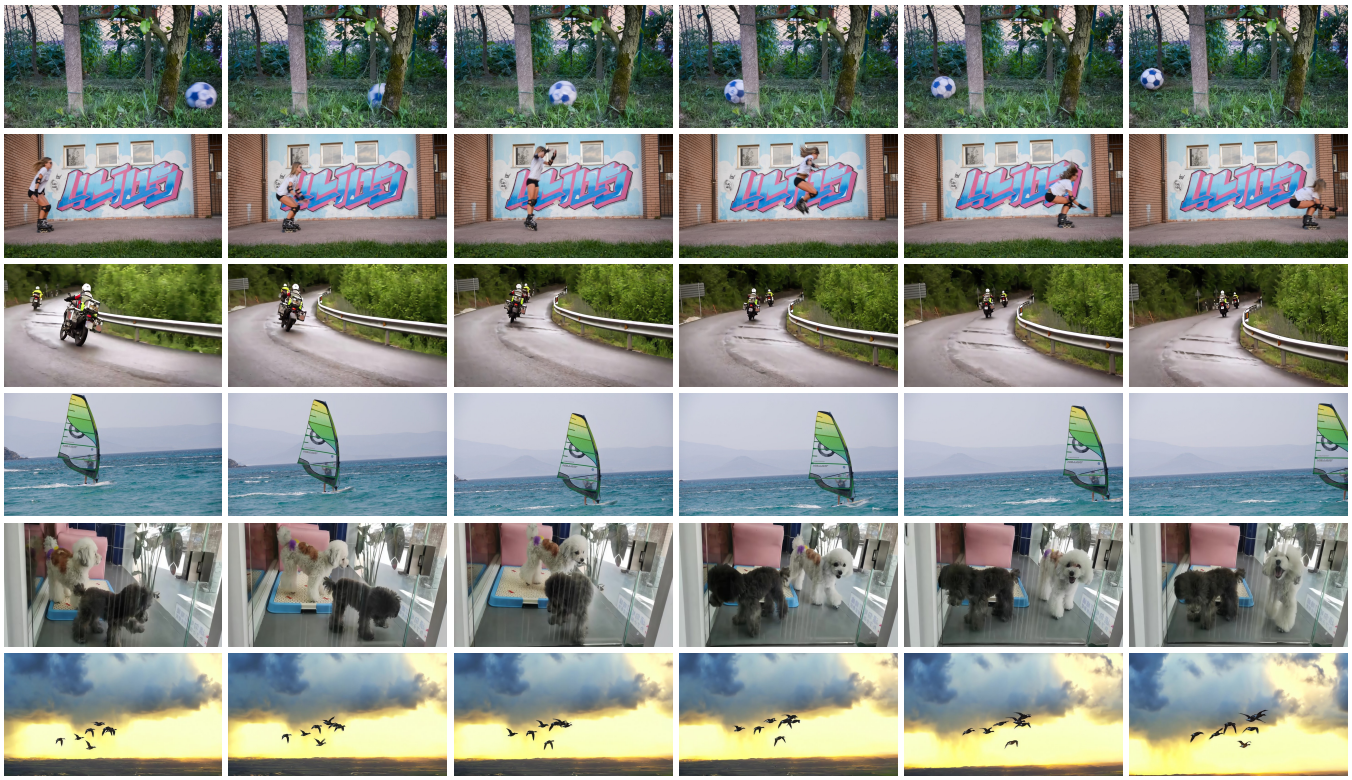


Figure 10: Visualization of Our Evaluation Dataset, highlighting 0%, 20%, 40%, 60%, 80%, and 100% of the total duration. Each center point of the annotated object masks is treated as a trajectory point. The number of trajectories in the tested video matches the number of annotated objects.

Pont-Tuset et al. 2017), which offer robust object motion critical for our analysis. To enhance the quality of our evaluation, we implement a camera motion filtering technique to select videos where the camera remains predominantly stable. This filtering process allows us to concentrate on where object motion is distinctly pronounced, thereby improving the reliability of our assessments. For each frame, we utilize the center of the annotated object masks as trajectory points, providing precise references for evaluating motion dynamics. Figure 10 presents several examples from our evaluation dataset, highlighting the diversity and relevance of the selected video sequences.

Prompt Refinement

We encourage users to provide detailed text prompts to achieve satisfactory video results. To ensure consistency in the distribution of text prompts during both training and testing phases, we utilize GPT-4o to refine simple testing prompts. The process of learning refined prompts for GPT-4o involves two key components. The first component is the task description, which clearly outlines the objectives for the model in generating enhanced content:

```
You need to refine user's input
prompt. The user's input prompt is used
for video generation task. You need
to refine the user's prompt to make it
more suitable for the task. Here are
some examples of refined prompts: ↓
a close-up shot of a woman applying
makeup. she is using a black brush to
apply a dark powder to her face. the
woman has blonde hair and is wearing
a black top. the background is black,
which contrasts with her skin tone and
the makeup. the focus is on her face
and the brush, with the rest of her
body and the background being out of
focus. the lighting is soft and even,
highlighting the texture of the makeup
and the woman's skin. there are no
texts or other objects in the video.
the woman's expression is neutral,
and she is looking directly at the
camera. the video does not contain
any action, as it is a still shot of
a woman applying makeup. the relative
position of the woman and the brush is
such that the brush is in her hand and
is being used to apply the makeup to
her face. the video does not contain
any other objects or actions. the woman
is the only person in the video, and
she is the main subject. the video does
```

not contain any sound. the description is based on the visible content of the video and does not include any assumptions or interpretations. ↓
a professional setting where a woman is presenting a slide from a presentation. she is standing in front of a projector screen, which displays a bar chart. the chart is colorful, with bars of different heights, indicating some sort of data comparison. the woman is holding a pointer, which she uses to highlight specific parts of the chart. she is dressed in a white blouse and black pants, and her hair is styled in a bun. the room has a modern design, with a sleek black floor and a white ceiling. the lighting is bright, illuminating the woman and the projector screen. the focus of the image is on the woman and the projector screen, with the background being out of focus. there are no texts visible in the image. the relative positions of the objects suggest that the woman is the main subject of the image, and the projector screen is the object of her attention. the image does not provide any information about the content of the presentation or the context of the meeting. ↓
a serene scene in a park. the sun is shining brightly, casting a warm glow on the lush green trees and the grassy field. the camera is positioned low, looking up at the towering trees, which are the main focus of the image. the trees are dense and full of leaves, creating a canopy of green that fills the frame. the sunlight filters through the leaves, creating a beautiful pattern of light and shadow on the ground. the overall atmosphere of the video is peaceful and tranquil, evoking a sense of calm and relaxation. ↓
a moment in a movie theater. a couple is seated in the middle of the theater, engrossed in the movie they are watching. the man is dressed in a casual outfit, complete with a pair of sunglasses, while the woman is wearing a cozy sweater. they are seated on a red theater seat, which stands out against the dark surroundings. the theater itself is dimly lit, with the screen displaying the movie they are watching. the couple appears to be enjoying the movie, their attention completely absorbed by the on-screen action. the theater is mostly empty, with only a few other seats visible in the background. the video does not contain any text or additional objects. the relative positions of the objects are such that the couple is in the foreground, while the screen and the other seats are in the background. the focus of the video is clearly on the couple and their shared experience of watching a movie in a theater. ↓
a scene where a person is examining a dog. the person is wearing a blue shirt with the word "volunteer" printed on it. the dog is lying on its side, and the person is using a stethoscope to listen to the dog's heartbeat. the dog appears to be a golden retriever and is looking directly at the camera. the background is blurred, but it seems to be an indoor setting with a white wall. the person's focus is on the dog, and they seem to be checking its health. the dog's expression is calm, and it seems to be comfortable with the person's touch. the overall atmosphere of the video is calm and professional. ↓
The refined prompt should pay attention to all objects in the video. The description should be useful for AI to re-generate the video. The description should be no more than six sentences. The refined prompt should be in English.

Following that, GPT-4o is supplied with the testing captions for processing. This allows it to refine the prompts based on the initial task description, ensuring that the provided captions are more detailed and aligned with our objectives:

Generate the refined prompts for following inputs: ↓
A man rides on a huge fish, flying from the water into the sky. ↓
Two Jedi cats are fighting with each other in the forest. ↓
A polar bear with a black hat is walking on the Great Wall. ↓
A woman and a golden retriever are playing on the beach at sunset. ↓
Two roses sway together before a snow-covered mountain range.

Motion VAE Training

Given the absence of pre-existing networks tailored for video optical flow compression, training such a network from scratch presents significant challenges. Directly transferring the motion vectors to the image domain and applying a pretrained 3D VAE may hinder the model's ability to effectively encode motion features, primarily due to domain

discrepancies. To overcome this issue, we refine a motion-specific 3D VAE that is initialized from a pretrained model. Specifically, our motion 3D VAE is specifically initialized using the architecture of OpenSora's VAE, which adapts the structure of Magvit-v2. This VAE has a substantial parameter count of 384 million, effectively leveraging the capabilities of a well-established network. Our training data is sourced from a combination of datasets annotated with optical flow information (Mehl et al. 2023; Mayer et al. 2016; Ranjan et al. 2020; Cabon, Murray, and Humenberger 2020). We fine-tune the motion 3D VAE for 200,000 iterations with a batch size of 1. The training video size is set to a random number of frames, capped at 34. This setting aligns with the OpenSora video VAE, improving compatibility between the motion VAE and the video VAE and ensuring a cohesive training process. We utilize PSNR, SSIM and trajectory error to evaluate reconstruction quality and motion-controllable ability. The performance differences between the pure video VAE and our fine-tuned model are presented in Table 7.

Model	PSNR↑	SSIM↑	TrajError↓
Pure Video VAE	27.34	0.842	17.09
Our VAE	28.76	0.860	14.25

Table 7: The performance comparison of different 3D VAE.

References

Bao, F.; Xiang, C.; Yue, G.; He, G.; Zhu, H.; Zheng, K.; Zhao, M.; Liu, S.; Wang, Y.; and Zhu, J. 2024. Vidu: a Highly Consistent, Dynamic and Skilled Text-to-Video Generator with Diffusion Models. *arxiv*, abs/2405.04233.

Blattmann, A.; Dockhorn, T.; Kulal, S.; Mendelevitch, D.; Kilian, M.; Lorenz, D.; Levi, Y.; English, Z.; Voleti, V.; Letts, A.; Jampani, V.; and Rombach, R. 2023. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets. *arXiv preprint arXiv:2311.15127*.

Brooks, T.; Peebles, B.; Holmes, C.; DePue, W.; Guo, Y.; Jing, L.; Schnurr, D.; Taylor, J.; Luhman, T.; Luhman, E.; Ng, C.; Wang, R.; and Ramesh, A. 2024. Video generation models as world simulators.

Cabon, Y.; Murray, N.; and Humenberger, M. 2020. Virtual KITTI 2. *CoRR*, abs/2001.10773.

Chen, H.; Xia, M.; He, Y.-Y.; Zhang, Y.; Cun, X.; Yang, S.; Xing, J.; Liu, Y.; Chen, Q.; Wang, X.; Weng, C.-L.; and Shan, Y. 2023. VideoCrafter1: Open Diffusion Models for High-Quality Video Generation. *ArXiv*, abs/2310.19512.

Chen, T.; Siarohin, A.; Menapace, W.; Deyneka, E.; Chao, H.; Jeon, B. E.; Fang, Y.; Lee, H.; Ren, J.; Yang, M.; and Tulyakov, S. 2024. Panda-70M: Captioning 70M Videos with Multiple Cross-Modality Teachers. *arxiv*, abs/2402.19479.

Dai, Z.; Zhang, Z.; Yao, Y.; Qiu, B.; Zhu, S.; Qin, L.; and Wang, W. 2023. Fine-Grained Open Domain Image Animation with Motion Guidance. *arxiv*, abs/2311.12886.

Dhariwal, P.; and Nichol, A. 2021. Diffusion Models Beat GANs on Image Synthesis. *ArXiv*, abs/2105.05233.

Envato, E. 2024. mixkit:Free assets for your next video project.

Ho, J.; Chan, W.; Saharia, C.; Whang, J.; Gao, R.; Gritsenko, A.; Kingma, D. P.; Poole, B.; Norouzi, M.; Fleet, D. J.; et al. 2022a. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*.

Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022b. Video diffusion models.

Jeong, H.; Park, G. Y.; and Ye, J. C. 2024. VMC: Video Motion Customization using Temporal Attention Adaption for Text-to-Video Diffusion Models. In *CVPR*.

Khachatryan, L.; Movsisyan, A.; Tadevosyan, V.; Henschel, R.; Wang, Z.; Navasardyan, S.; and Shi, H. 2023. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Mayer, N.; Ilg, E.; Häusser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; and Brox, T. 2016. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 4040–4048. IEEE Computer Society.

Mehl, L.; Schmalfuss, J.; Jahedi, A.; Nalivayko, Y.; and Bruhn, A. 2023. Spring: A High-Resolution High-Detail Dataset and Benchmark for Scene Flow, Optical Flow and Stereo. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, 4981–4991. IEEE.

Mou, C.; Wang, X.; Xie, L.; Wu, Y.; Zhang, J.; Qi, Z.; and Shan, Y. 2024. T2I-Adapter: Learning Adapters to Dig Out More Controllable Ability for Text-to-Image Diffusion Models. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024*, 4296–4304.

Olaf Ronneberger, T. B., Philipp Fischer. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 234–241. Springer.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4195–4205.

Perez, E.; Strub, F.; de Vries, H.; Dumoulin, V.; and Courville, A. C. 2018. FiLM: Visual Reasoning with a General Conditioning Layer. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024*, 3942–3951.

Podell, D.; English, Z.; Lacey, K.; Blattmann, A.; Dockhorn, T.; Müller, J.; Penna, J.; and Rombach, R. 2023. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. *arxiv*, abs/2307.01952.

Pont-Tuset, J.; Perazzi, F.; Caelles, S.; Arbeláez, P.; Sorkine-Hornung, A.; and Van Gool, L. 2017. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*.

Qi, J.; Gao, Y.; Hu, Y.; Wang, X.; Liu, X.; Bai, X.; Belongie, S.; Yuille, A.; Torr, P. H.; and Bai, S. 2022. Occluded video instance segmentation: A benchmark. *International Journal of Computer Vision*, 130(8): 2022–2039.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents, 2022. *URL https://arxiv. org/abs/2204.06125*, 7.

Ranjan, A.; Hoffmann, D. T.; Tzionas, D.; Tang, S.; Romero, J.; and Black, M. J. 2020. Learning Multi-human Optical Flow. *Int. J. Comput. Vis.*, 128(4): 873–890.

Singer, U.; Polyak, A.; Hayes, T.; Yin, X.; An, J.; Zhang, S.; Hu, Q.; Yang, H.; Ashual, O.; Gafni, O.; et al. 2022. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*.

Unterthiner, T.; van Steenkiste, S.; Kurach, K.; Marinier, R.; Michalski, M.; and Gelly, S. 2018. Towards Accurate Generative Models of Video: A New Metric & Challenges. *arXiv:1812.01717*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Wan-Duo Kurt Ma, W. B. K., J. P. Lewis. 2023. TrailBlazer: Trajectory Control for Diffusion-Based Video Generation. *arXiv preprint arXiv:2401.00896*.

Wang, X.; Yuan, H.; Zhang, S.; Chen, D.; Wang, J.; Zhang, Y.; Shen, Y.; Zhao, D.; and Zhou, J. 2023. VideoComposer: Compositional Video Synthesis with Motion Controllability. *arXiv preprint arXiv:2306.02018*.

Wang, X.; Zhang, S.; Yuan, H.; Qing, Z.; Gong, B.; Zhang, Y.; Shen, Y.; Gao, C.; and Sang, N. 2024a. A Recipe for Scaling up Text-to-Video Generation with Text-free Videos.

Wang, Z.; Yuan, Z.; Wang, X.; Chen, T.; Xia, M.; Luo, P.; and Shan, Y. 2024b. MotionCtrl: A Unified and Flexible Motion Controller for Video Generation. In *SIGGRAPH*.

Wu, C.; Huang, L.; Zhang, Q.; Li, B.; Ji, L.; Yang, F.; Sapiro, G.; and Duan, N. 2021. Godiva: Generating open-domain videos from natural descriptions. *arXiv preprint arXiv:2104.14806*.

Xu, H.; Zhang, J.; Cai, J.; Rezatofighi, H.; Yu, F.; Tao, D.; and Geiger, A. 2023. Unifying Flow, Stereo and Depth Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(11): 13941–13958.

Xu, L.; Zhao, Y.; Zhou, D.; Lin, Z.; Ng, S.; and Feng, J. 2024. PLLaVA : Parameter-free LLaVA Extension from Images to Videos for Video Dense Captioning. *arxiv*, abs/2404.16994.

Xu, N.; Yang, L.; Yang, J.; Yue, D.; Fan, Y.; Liang, Y.; and Huang, T. S. 2021. Youtubevis dataset 2021 version. In *https://youtube-vos.org/dataset/vis*.

Yin, S.; Wu, C.; Liang, J.; Shi, J.; Li, H.; Ming, G.; and Duan, N. 2023. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*.

Yu, L.; Cheng, Y.; Sohn, K.; Lezama, J.; Zhang, H.; Chang, H.; Hauptmann, A. G.; Yang, M.-H.; Hao, Y.; Essa, I.; et al. 2023a. Magvit: Masked generative video transformer.

Yu, L.; Lezama, J.; Gundavarapu, N. B.; Versari, L.; Sohn, K.; Minnen, D.; Cheng, Y.; Gupta, A.; Gu, X.; Hauptmann, A. G.; Gong, B.; Yang, M.; Essa, I.; Ross, D. A.; and Jiang, L. 2023b. Language Model Beats Diffusion - Tokenizer is Key to Visual Generation. *arXiv*, abs/2310.05737.

Zhang, S.; Wang, J.; Zhang, Y.; Zhao, K.; Yuan, H.; Qin, Z.; Wang, X.; Zhao, D.; and Zhou, J. 2023a. I2VGen-XL: High-Quality Image-to-Video Synthesis via Cascaded Diffusion Models. *arXiv preprint arXiv:2311.04145*.

Zhang, Y.; Wei, Y.; Jiang, D.; Zhang, X.; Zuo, W.; and Tian, Q. 2023b. ControlVideo: Training-free Controllable Text-to-Video Generation. *arXiv preprint arXiv:2305.13077*.

Zhao, R.; Gu, Y.; Wu, J. Z.; Zhang, D. J.; Liu, J.; Wu, W.; Keppo, J.; and Shou, M. Z. 2023. MotionDirector: Motion Customization of Text-to-Video Diffusion Models. *arxiv*, abs/2310.08465.

Zhao, W.; Liu, S.; Guo, H.; Wang, W.; and Liu, Y. 2022. ParticleSfM: Exploiting Dense Point Trajectories for Localizing Moving Cameras in the Wild. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXII*, volume 13692, 523–542.

Zheng, Z.; Peng, X.; Yang, T.; Shen, C.; Li, S.; Liu, H.; Zhou, Y.; Li, T.; and You, Y. 2024. Open-Sora: Democratizing Efficient Video Production for All.