

MotionStone: Decoupled Motion Intensity Modulation with Diffusion Transformer for Image-to-Video Generation

Shuwei Shi^{1*}, Biao Gong^{2†}, Xi Chen³, Dandan Zheng², Shuai Tan², Zizheng Yang²,
Yuyuan Li⁴, Jingwen He⁵, Kecheng Zheng², Jingdong Chen², Ming Yang², Yinqiang Zheng^{1‡}

¹The University of Tokyo ²Ant Group ³Tongyi Lab
⁴Zhejiang University ⁵The Chinese University of Hong Kong

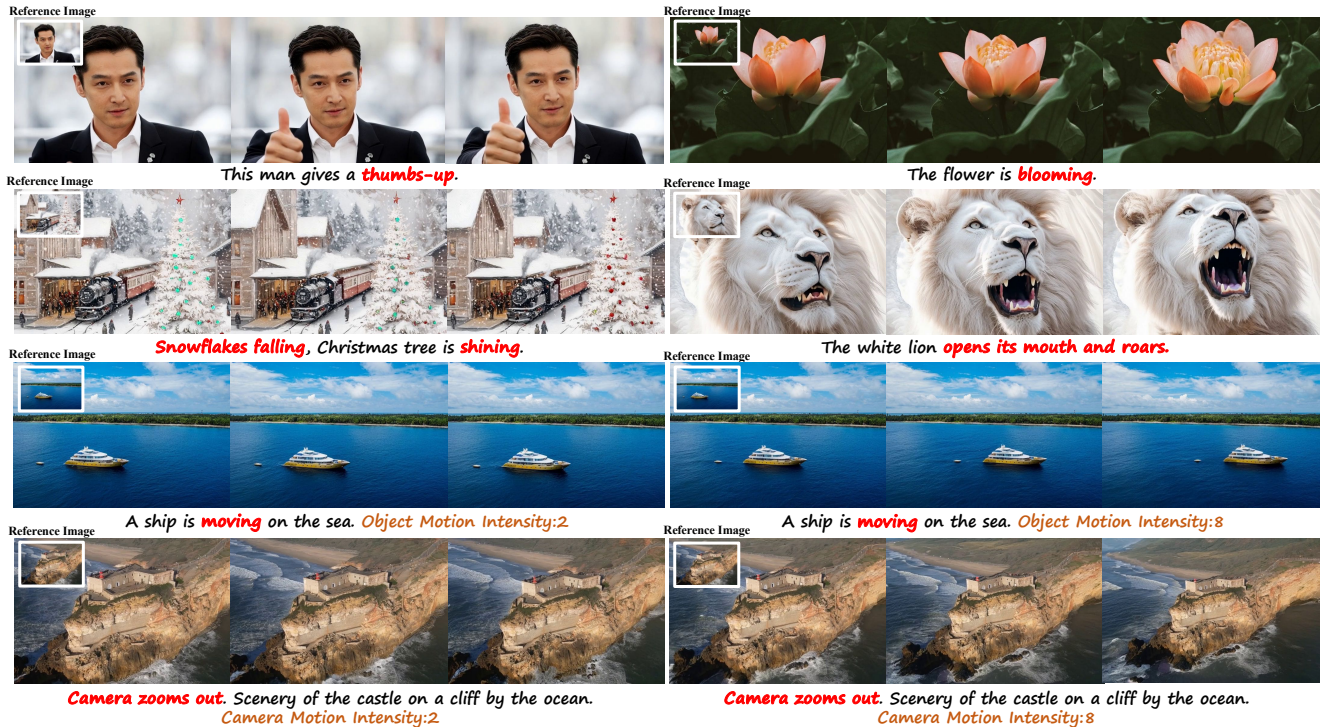


Figure 1. Samples generated by MotionStone. Our model achieves accurate motion instruction following (rows-1 and rows-2), and is controllable, easily adapting to specified object motion intensities (row-3) and camera motion intensities (row-4).

Abstract

The image-to-video (I2V) generation is conditioned on the static image, which has been enhanced recently by the motion intensity as an additional control signal. These motion-aware models are appealing to generate diverse motion patterns, yet there lacks a reliable motion estimator for training such models on large-scale video set in the wild. Traditional metrics, e.g., SSIM or optical flow, are hard to generalize to arbitrary videos, while, it is very tough for human annotators to label the abstract motion intensity neither. Furthermore, the motion intensity shall reveal both local object motion and global camera movement,

which has not been studied before. This paper addresses the challenge with a new motion estimator, capable of measuring the decoupled motion intensities of objects and cameras in video. We leverage the contrastive learning on randomly paired videos and distinguish the video with greater motion intensity. Such a paradigm is friendly for annotation and easy to scale up to achieve stable performance on motion estimation. We then present a new I2V model, named MotionStone, developed with the decoupled motion estimator. Experimental results demonstrate the stability of the proposed motion estimator and the state-of-the-art performance of MotionStone on I2V generation. These advantages warrant the decoupled motion estimator to serve as a general plug-in enhancer for both data processing and video generation training.

*Work done during internship at Ant Group. †Project lead.

‡Corresponding author.

1. Introduction

Image-to-Video (I2V) generation animates static images into fun creative videos which has attracted broad interests in research and industry [16, 18, 34, 44, 51, 54]. The key to achieving high-quality I2V results lies in synthesizing sufficient temporal dynamics, which requires effective frame-to-frame motion modeling. Some methods [5, 12, 24, 29, 30, 42, 47, 55] introduce additional conditions into diffusion models, *e.g.*, optical flow, motion trajectories, or depth maps, to better capture motion dynamics. However, these methods require complex and hard-to-obtain control conditions as inputs, and the training data must be meticulously preprocessed for model training, preventing them from reliably generalizing to videos in the wild.

Recently, several I2V works [8, 11, 26] explore text-based motion control and introduce motion intensity as the essential control signal on motion patterns. For example, LivePhoto [8] and Cinemo [26] leverage text prompts to direct motion and integrate SSIM [46] to modulate motion intensity. Although these motion-aware models demonstrate improved controllability in motion and enhanced generation quality, the estimation of motion intensity remains inadequate due to the discrepancy between their motion modeling strategy and human motion perception. As a result, the diffusion model is unable to accurately capture the real motion intensity in a video clip during training, which negatively impacts the convergence process.

Furthermore, as shown in Fig. 2, motion patterns in real-world videos could be very complicated including both object motion and camera movement. Applying traditional motion extractors, which are not specifically designed for video motion modeling, to estimate motion across entire videos, is unattainable to distinguish between different types of motion, thereby limiting precise control over motion dynamics. A straightforward way is to learn a motion estimator to predict human perception of object and camera motion intensity in videos.

In this paper, we introduce *MotionStone*, a general I2V diffusion model to enable decoupled modeling and control of video motion. The core of *MotionStone* is the independent motion estimator, comprising a motion modeling backbone and dual heads to disentangle object and camera motion. Specifically, we first propose a video motion annotation method, which requires human annotators to distinguish the relative motion intensity of objects and cameras in randomly selected video pairs. Then the proposed motion estimator is trained using a contrastive learning strategy with these relatively annotated video pairs. For the structure of the motion estimator, we employ a learnable TAdaConv [19] as the motion feature extractor, integrating the pairwise ranking loss and MLP-based motion heads to facilitate motion disentanglement.

During the training phase of the diffusion model, we

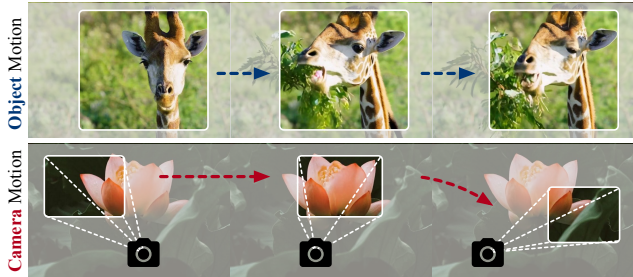


Figure 2. **Illustration of the motion decoupling.** Decoupling these two types of motion helps the diffusion model learn specific motion patterns, thereby improving the dynamics and controllability of the generated video.

freeze the pre-trained motion estimator and use its prediction result as an additional input for noise prediction at each step. In particular, we design a decoupled motion score injection method that allows the model to discern whether each motion intensity control signal originates from the camera or the object, thus achieving decoupled motion modeling in training. Extensive quantitative and qualitative results demonstrate that *MotionStone* achieves state-of-the-art performance in text-guided motion control through its decoupled motion intensity guidance and conditional injection, as shown in Fig. 1. *MotionStone* animates diverse real-world images across various domains, skillfully decomposing motion into object and camera components. With its decoupled motion guidance, *MotionStone* allows users to customize motion intensity, enabling a wide range of motion effects.

2. Related Work

Image Animation. Image animation aims to generate controllable videos using a static image as content conditioning. Early methods [10, 38] focus on modeling motion patterns for specific object types, limiting their ability to generalize motion control to other scenarios. To capture realistic motion from real videos, some methods [9, 35, 37, 44, 56, 57] use a set of videos with various motion patterns as references, transferring these motion patterns to images within the same category. Other approaches model motion for specific scenes, such as fluids [28, 31] and human hair [48]. Additionally, some methods [6, 18, 40, 43] convert the human pose into additional conditions, such as depth maps or skeletal points, to guide video generation. Although these methods achieve continuous motion control within a specific domain, their applicability remains limited due to the restricted training data and the frequent need for side control signals. Subsequently, some generalizable I2V models [2, 53] typically train on video data based on pre-trained I2V models. However, the generated videos often exhibit limited motion diversity due to structural limitations and conditions [8].

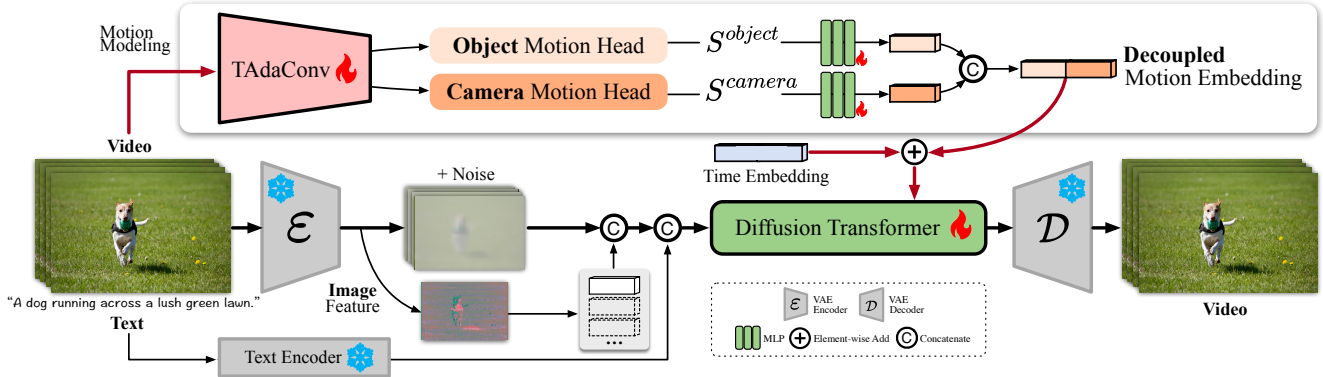


Figure 3. **The framework of MotionStone.** The first frame of the video serves as the conditioning image, while object and camera motion intensities (ranging from 1 to 10) are predicted by the motion estimator and can be customized by users during inference. At the top, the object and camera motion intensities predicted by the motion estimator are processed through an MLP respectively to obtain corresponding embeddings, which are then concatenated along the channel dimension to form the Decoupled Motion Embedding. This embedding is added to the time embedding and injected into the Diffusion Transformer to generate videos.

Recent methods [8, 11, 26, 54] explore using text as a condition to control motion in video. For instance, PIA [54] attempts to animate specific domain images using text descriptions of motion. Other methods [8, 11, 26] further incorporate coarse-grained motion intensity estimates to generate videos with varying intensities or speeds. However, these approaches lack alignment with human perception, leading to suboptimal results in motion control. In this work, we propose a generalizable framework that uses flexible text as a guiding condition, enabling precise motion modeling in generated videos.

Text-to-Video Generation. The text-to-video (T2V) models have made significant progress along the emerging diffusion models [15, 17, 36, 39]. Early T2V models [3, 16, 27, 45, 47] harness the strong priors of existing text-to-image (T2I) models, adapting temporal modules trained on video data to enable video generation. For instance, Tune-A-Video [47] fine-tunes a pretrained T2I diffusion model with a temporal attention mechanism in a one-shot manner. AnimateDiff [16] introduces a plug-and-play motion module that integrates seamlessly into existing personalized T2I diffusion models to animate images in a similar way. However, these models rely on U-Net-based denoising networks, which have limited their performance.

Recently, some works [7, 14, 27, 52] have shifted the denoising network from U-Net to Diffusion Transformer, inspired by DiT [32]. Video generation models with Transformers have strong spatiotemporal modeling abilities. Powered by large-scale training data, they can generate videos with rich content and motion. CogVideoX [52] utilizes a 3D Variational Autoencoder and an expert Transformer with adaptive LayerNorm to produce coherent, extended-duration videos from text prompts. However, while these methods allow text to control content, they limit the fine-grained control over object and camera movement.

3. Method

Our method is built on a pretrained video diffusion model [52], consisting primarily of a diffusion transformer [32] and a 3D VAE [52]. We first give a brief introduction to the process of the video diffusion model in Sec. 3.1, followed by presenting the overall pipeline in Sec. 3.2. In Sec. 3.3, we provide a detailed explanation of motion intensity estimation, and in Sec. 3.4, we propose a new scheme for injecting motion intensity.

3.1. Preliminaries

Diffusion Transformer demonstrates superior capabilities of spatiotemporal modeling in video generation compared to U-Net architecture. In this work, we select CogVideoX [52] as the pre-trained model. Given a video $\mathbf{x} \in \mathbb{R}^{F \times H \times W \times 3}$, the 3D VAE encoder \mathcal{E} compresses video frames along the spatiotemporal dimensions to obtain a latent representation $\mathbf{z}_0 = \mathcal{E}(\mathbf{x})$, where $\mathbf{z}_0 \in \mathbb{R}^{(\frac{F-1}{4}+1) \times H' \times W' \times C}$. After that, the forward diffusion and reverse denoising processes are performed in the latent space. During the forward phase, noise is incrementally added to the latent vector \mathbf{z}_0 over a total of T steps. At each time step t , the diffusion process is defined as follows:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t} \mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad (1)$$

where $\epsilon \in \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\bar{\alpha}_t$ is the cumulative products of noise coefficient α_t at each time step. For the backward pass, a diffusion model performs iterative noise reduction, guided by the text prompt c_{text} and time step t . The objective of this stage can be formulated as:

$$L = \mathbb{E}_{\mathcal{E}(x), c_{text}, \epsilon \sim \mathcal{N}(0,1), t} \left[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, c_{text})\|_2^2 \right]. \quad (2)$$

3.2. Overall Pipeline

The framework of our model is shown in Figure 3. The model takes a reference image, a text prompt, and two disentangled motion intensities predicted by a motion intensity estimator as inputs. During training, we first extract the first frame from the input video to use as a conditioning reference for generation. The trained motion intensity estimator then predicts the camera and object motion intensities of the input video, providing two motion scores that guide the video generation process. During inference, users can specify the desired motion intensities for the object and camera, if available, to customize the generated video. The model takes a latent $\mathbf{z} \in \mathbb{R}^{B \times T \times C \times H \times W}$ and concatenates the first frame latent of the video along the channel dimension to guide video generation. For frames beyond the first in the video sequence, zeros are padded in place. Subsequently, the model uses a text encoder to extract the features of the text prompt, which are then concatenated with the latent and fed into the diffusion transformer. Meanwhile, the two motion intensities predicted by the motion estimator are mapped to high-dimensional embeddings by MLP, then concatenated and added to the time step t . This combined representation serves as a modulation condition for the vision and text features, enabling fine-grained control over the motion of video generation.

3.3. Motion Intensity Estimation

To achieve precise control over motion intensity, we train an independent motion estimator to predict the intensity of object and camera motion within a video. We provide a detailed explanation covering three aspects: the construction of training data, the design of the motion estimator architecture, and the training configuration.

Training Data Construction. Training a motion estimator to accurately predict video motion typically requires labeling object and camera motion intensities for each video—a highly challenging task. Due to the complexity of video motion, assigning specific scores to object and camera movement is impractical, as people find it difficult to consistently rate motion intensities. To address this issue, we develop a simple and intuitive labeling approach. Rather than assigning precise scores, annotators compare video pairs, indicating which video exhibits stronger object or camera motion. This method largely streamlines the annotation process. We construct 5,000 video pairs, with annotators labeling the relative motion intensities for object and camera motion within each pair.

Motion Estimator. The motion estimator needs to simultaneously predict both object motion and camera motion for video. Therefore, when designing the structure of the motion estimator, the first requirement is a backbone that can effectively represent the motion in the video. Based on this motion representation, two heads (an object

motion head and a camera motion head) are introduced to map the representation to two corresponding motion intensities. Given that our video generation network is quite large, it is crucial to limit the overall parameters and computational cost of the additional motion estimator. To achieve this goal, we use TAda [19] as the backbone for video motion representation. Given an input video \mathbf{x} , the motion representation of the video can be obtained through spatiotemporal modeling with TAdaConv. This process can be formulated by the following equation:

$$M = \text{TAdaConv}(\mathbf{x}; \phi), \quad (3)$$

where M represents the extracted motion representation features and ϕ represents the parameters of TAdaConv. Afterward, we apply global average pooling over the spatiotemporal dimensions on the extracted features, followed by two separate heads: one for object motion scoring and another for camera motion scoring. Both heads are composed of MLPs. Each head predicts the respective scores for object and camera motion in the video. This process can be formulated as follows:

$$s^{object, camera} = \text{MLP}_{object, camera}(\text{GAP}(M); \theta), \quad (4)$$

where s^{object} and s^{camera} represent the object motion score and camera motion score, respectively, θ represents the parameters of the object or camera motion prediction head and GAP denotes global average pooling.

Training Configuration. Since the training dataset only contains relative motion comparison labels in the video pair, we design a contrastive learning approach to train the motion intensity estimator. This method helps the motion intensity estimator predict the relative magnitude of object and camera motion in video pairs. We train the motion estimator using the ranking loss [25].

Specifically, given a video \mathbf{x} as input to the motion estimator, we obtain object motion score s^{object} and camera motion score s^{camera} through Eq. 3 and Eq. 4. Since our goal is to learn the relative rank in the video pair, we introduce the pairwise ranking loss to train the motion estimator:

$$L_o = \max(0, s_2^{object} - s_1^{object}), \quad (5)$$

$$L_c = \max(0, s_2^{camera} - s_1^{camera}), \quad (6)$$

here we assume that the object and camera motion of \mathbf{x}_1 is higher than \mathbf{x}_2 .

However, training only with the ranking loss, the predicted scores from the motion estimator tend to cluster closely together. Such an estimator can distinguish relative motion between videos but is not practically usable as it lacks sufficient differentiation. Ideally, the predicted score should reflect clear distinctions (ranging from 1 to 10).

To make the motion estimator practically applicable, we randomly sample a subset of videos from our training dataset. Using the tracking method [49] combined with object masks extracted by [50], we calculate tracking

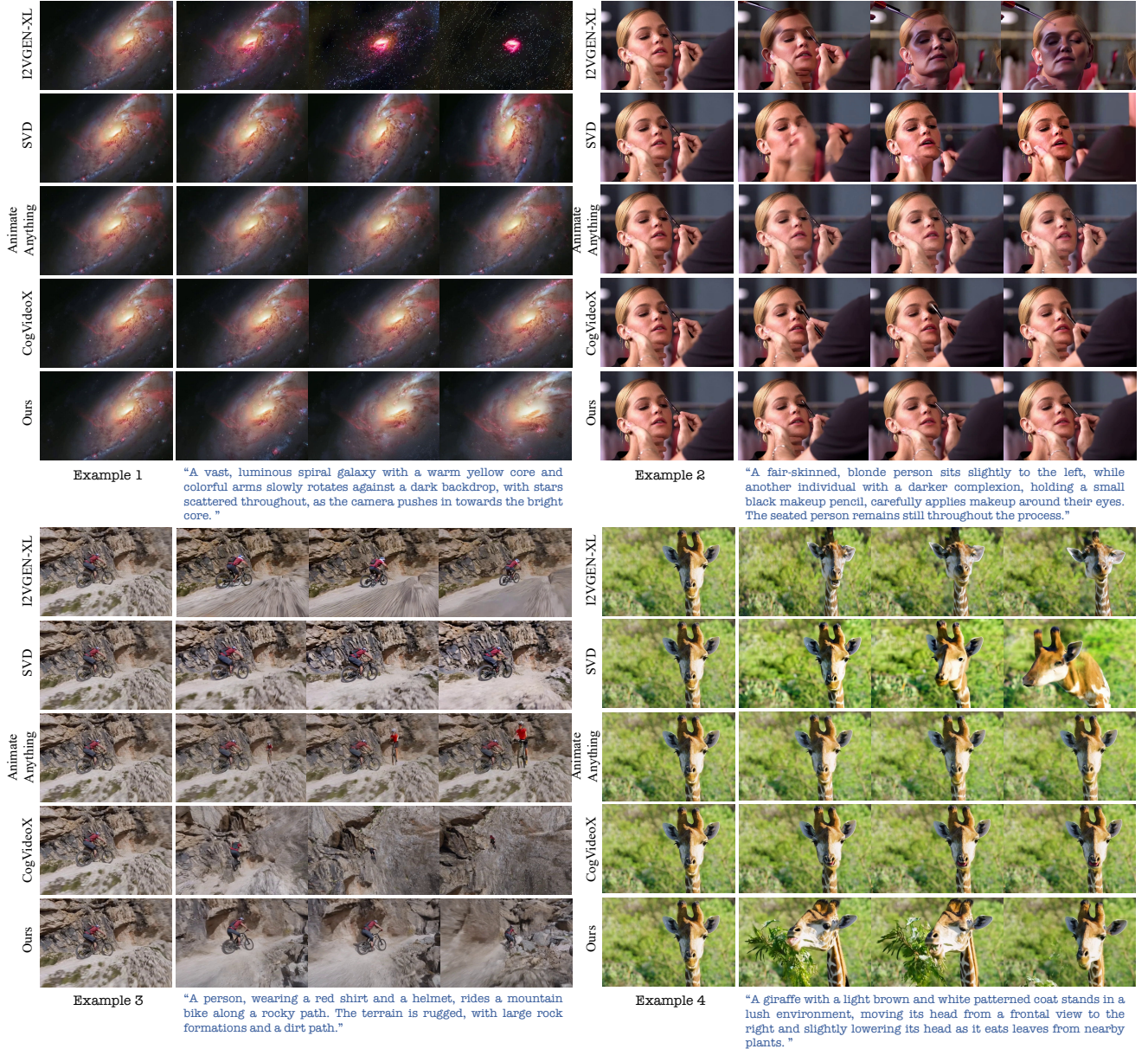


Figure 4. **Qualitative comparison with other methods.** We compare our MotionStone with I2VGEN-XL [53], SVD [2], AnimateAnything [11] and CogvideoX [52]. MotionStone demonstrates superior alignment with text and image inputs compared to other methods (*Example 2* and *Example 4*). Additionally, as shown in *Example 1*, it highlights the ability of camera controlling, while other methods tend to remain static frames. *Example 3* showcases the capacity of MotionStone to control object movements, whereas other methods either remain static frames or produce unrealistic scenes that defy physical principles.

trajectories for the object and camera motion in each video. From these trajectories, we can approximate the average motion intensity of the object y^{object} and camera y^{camera} . We use them as pseudo-labels of video motion to conduct regression training for the motion estimator. The regression loss of the motion estimator training can be formulated as:

$$\mathcal{L}_r = \|s^{object} - y^{object}\|_2^2 + \|s^{camera} - y^{camera}\|_2^2. \quad (7)$$

We then jointly train the estimator using both the ranking

loss and regression loss with pseudo-labels derived from the tracking results. The overall training loss can be defined:

$$\mathcal{L}_{total} = \mathcal{L}_o + \mathcal{L}_c + \lambda \mathcal{L}_r, \quad (8)$$

where λ denotes the balancing parameter.

3.4. Motion Condition Injection Design

After training the motion estimator, it is crucial to inject the predicted motion intensity values into the backbone

Table 1. **Quantitative comparison** with state-of-the-art methods. We use Background Consistency to assess temporal quality, while aesthetics and imaging quality metrics are used to evaluate the visual quality of each frame.

Method	Background Consistency	Aesthetic Quality	Imaging Quality
I2VGen-XL [53]	90.93%	40.14%	58.35%
SVD [2]	93.17%	42.38%	59.61%
AnimateAnything [11]	93.89%	46.04%	61.69%
CogVideoX-5B [52]	94.91%	45.88%	61.99%
MotionStone	95.76%	46.78%	62.29%

network as conditions. Due to the distinct meanings, these two motion types can’t be directly compared and combined in the same way, so we propose a decoupled injection approach during the diffusion model training.

Specifically, we use two separate MLPs to learn high-dimensional mappings for the predicted object and camera motion vectors. These vectors are then concatenated and added to t , allowing two conditions to remain disentangled, and preventing ambiguity in condition injection. As shown in Figure 3, the input motion intensities are first mapped to high-dimensional vectors, similar to t , and then processed through two MLPs with the same channel dimensions, respectively. The outputs are concatenated and added to t , collectively modulating the scaling coefficients.

4. Experiments

4.1. Implementation Details

Training Configurations. We implement `MotionStone` using the CogVideoX [52] framework. Our model training is conducted on 100,000 high-quality videos collected by ourselves, utilizing 8 A100 GPUs with batch size 16. The training is performed using Supervised Fine-Tuning (SFT). For each training video, we sample 49 frames and apply center cropping and resizing to standardize their resolution to 480×720 . We condition the Image-to-Video model training on the first frame of each video alongside its associated text prompt.

Evaluation Metrics. We conduct user studies to compare our approach with previous methods. Please refer to the appendix for detailed results. For quantitative analysis, we use the WebVID validation set [1], where the first frame and corresponding prompt serve as conditions to generate videos. We employ specific metrics from VBench [20] to evaluate the generated videos, using Background Consistency to assess temporal quality, while aesthetics and imaging quality metrics are used to evaluate the visual quality of each frame.

4.2. Comparisons with Existing Alternatives

We compare `MotionStone` with several recent Image-to-Video (I2V) methods. I2VGEN-XL [53] and AnimateAnything [11] are classic I2V approaches that enable

Table 2. **Ablation Study for proposed modules.** Motion Estimator (M), Decoupled injection strategy (D). SSIM and S mean previous motion modeling methods: inter-frame SSIM [8] and feature difference [11] respectively.

Method	Background Consistency	Aesthetic Quality	Imaging Quality
MotionStone w/o M	95.13%	45.61%	60.15%
MotionStone w/ S	94.97%	46.13%	60.73%
MotionStone w/ SSIM	92.99%	45.72%	54.75%
MotionStone w/o D	94.03%	46.27%	58.73%
MotionStone	95.76%	46.78%	62.29%

Table 3. **Comparison with previous motion intensity estimation methods.** We calculate and compare object and camera motion scores for each video pair, then validate these predictions against manually annotated ground truth. Correct predictions score 1 point. Evaluation is conducted on the validation set of the video pair dataset introduced in Sec. 3.3.

Method	Motion Estimation Accuracy
SSIM	44.56%
Ours	72.80%

video generation conditioned on a given image and text. AnimateAnything also supports coarse control over motion intensity. SVD [2] is a widely used I2V model that employs U-Net as its denoising network. Additionally, CogVideoX [52] is an open-source video generation model based on the Diffusion Transformer.

Quantitative Results. We conduct quantitative experiments using WebVID [22] validation dataset. Specific metrics from VBench [20] are selected to evaluate the experimental results. Among these, the Background Consistency metric from the CLIP Score [33] effectively reflects the generative quality in the temporal dimension of video. Meanwhile, Aesthetic Quality and Imaging Quality respectively assess the aesthetic appeal and the quality of each individual frame in the generated video. It is important to note that we do not utilize the prompt suite from VBench; instead, we only employ their evaluation procedures and models. As shown in Table 1, compared to U-Net-based generative models like SVD, I2VGEN-XL, and AnimateAnything, our approach demonstrates significant improvements in both temporal consistency and visual quality of the generated videos. Even relative to our baseline, CogVideoX, our method achieves noticeable enhancements. This is largely due to our model’s integration of motion intensity prediction and decoupled conditional injection, which effectively reduces the ambiguity between the motion described in text prompts and the actual motion intensity in the generated videos. This also demonstrates that precise motion intensity control signals can help video models converge more effectively.

Qualitative Analysis. In Figure 4, we select a set of representative samples to qualitatively compare `MotionStone` with I2VGEN-XL [53], SVD [2], AnimateAnything [11],

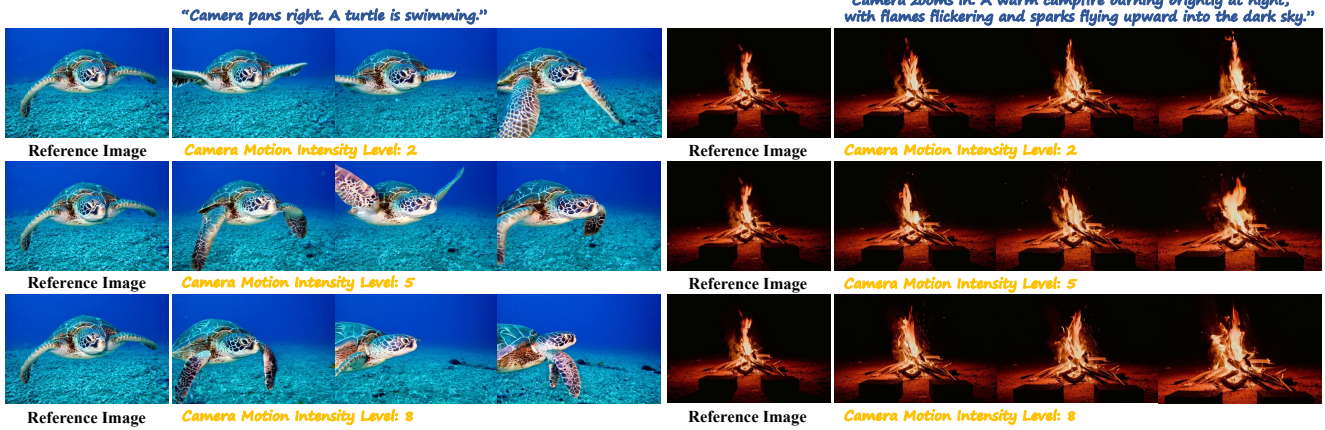


Figure 5. **Illustrations of camera motion intensity guidance.** We present two common camera movements: Zoom and Pan. Since the camera movement often impacts object motion in scenes with moving subjects, we fix the object motion intensity at 5 to isolate and highlight the effect of varying camera motion intensity. The camera movement becomes significant when the score increases.



Figure 6. **Illustrations of object motion intensity guidance.** To emphasize control over object motion intensity and speed, we exclude camera motion prompts from the text and set the camera motion intensity to its minimum value of 1 while varying the object motion intensity. As the given object motion intensity increases, the generated video reflects a corresponding increase in object motion intensity.

and CogVideoX [52]. We select cases involving people, animals, natural scenes, and fast-moving scenarios. As can be seen, the identity of the subject in the videos generated by I2VGEN-XL is not well-preserved, and there are occasional discrepancies between the motion and the text prompt. SVD also appears to face issues with preserving the identity of objects, and in fast-moving scenarios, the generated video exhibits limited motion intensity (e.g., the bicycle remains stationary). Although the video frames generated by AnimateAnything are well-aligned with the input image, in most scenes, the generated video is almost static, and there are occasional interruptions from other objects that interfere with the main subject. Compared to previous methods, CogVideoX shows some improvements in motion continuity. However, it occasionally fails to align with the content of the input image and exhibits limited motion. In Example 3, it generates content that does not adhere to the physical rules of the real world.

In contrast, MotionStone is capable of generating videos that align well with both the input image and text. The generated videos exhibit substantial camera and object motion, producing visually appealing shots and motion that adhere to the laws of the physical world.

4.3. Ablation Studies

In this section, we provide detailed analyses of proposed modules. We begin with quantitative experiments on all of the proposed modules, followed by a qualitative analysis of the controllability of object and camera motion intensities. Finally, we evaluate the motion magnitude between video pairs in the validation set. Specifically, we compare the accuracy of our motion estimator against SSIM to verify its predictive effectiveness and its ability to decouple motion.

Motion Estimator. To validate the overall effectiveness of the proposed motion estimator, we compare the quantitative performance of MotionStone trained with fixed motion

intensity (set to a default value of 5) versus `MotionStone` trained with motion intensities estimated by the motion estimator. The experiments are conducted on the WebVID validation set. As shown in Table 2, the quality of videos generated by `MotionStone` without the motion estimator (`MotionStone w/o M`) shows a noticeable decline. This is due to the variability in object and camera motion within the training data, using a fixed intensity value confuses the model’s understanding of video motion dynamics.

Decoupled Motion Condition Injection. To quantitatively demonstrate the effectiveness of the proposed decoupled injection method, we compare the performance of `MotionStone` trained with decoupled versus non-decoupled motion intensity injection. In the non-decoupled injection approach, object and camera motion are not specifically separated along feature channels but are instead mixed and injected together. As shown in Table 2, the performance of `MotionStone` with non-decoupled injection (`MotionStone w/o D`) is inferior to that of the decoupled injection approach. This is primarily because object and camera motion occur in different spatial dimensions; mixing them together obscures their distinct contributions, making it challenging for the model to discern each type of motion, thus complicating the training process.

Comparison with Previous Motion Intensity Estimation Methods. We further compare our method with previous methods for modeling motion intensity, which uses inter-frame SSIM [8] (`MotionStone w/ SSIM`) or feature difference [11] (`MotionStone w/ S`). Models are trained following these methods; however, since neither approach can decouple object and camera motion, we use a single motion intensity guidance to train the model. As shown in Table 2, both methods exhibit varying degrees of performance decline. This is because neither SSIM nor inter-frame feature difference aligns well with human perception of video motion intensity. Additionally, both methods fail to decouple complex video motion dynamics, instead modeling the motion intensity of the entire scene as a coarse, unified value, which leads to inaccurate estimations.

We further evaluate the trained motion estimator and the SSIM-based motion intensity estimation method on the validation set of the manually constructed video pair dataset introduced in Sec. 3.3. Since SSIM cannot decouple object and camera motion, we use its predicted overall motion intensity as a proxy for both object and camera motion intensities. We calculate the object and camera motion scores for both videos in a video pair and compare their relative magnitudes. The obtained comparison is then compared with the manually annotated ground truth (GT). If the predicted object or camera motion relationship is correct, it is scored as 1 point. The final motion intensity relationship prediction accuracy is calculated using this method, as shown in Table 3. Our motion estimator achieves excellent

accuracy in predicting motion relationships, surpassing SSIM-based method by 28%. This demonstrates that the trained motion estimator effectively decouples object and camera motion in videos.

Motion Intensity Estimation. As demonstrated in Sec. 3.3, we represent the intensity of object and camera motion in a video as a score, reflecting the magnitude and speed of both object and camera movements in the video. We perform ablation analysis of camera and object motion intensities in Figure 5 and Figure 6, respectively. In Figure 5, we control for camera motion and explore the impact of intensity control on two representative types of camera movements: zoom and pan. Since camera movement often influences object motion in scenes with moving subjects, we fix the object motion intensity at 5 to highlight the effect of controlling camera movement by varying its intensity level. From the left set of images, we observe that when the camera motion intensity level is set to 2, the rightward pan of the camera is limited. As the motion intensity level increases, the amplitude of the rightward pan significantly grows, resulting in a more substantial shift in perspective. For the images on the right, as the camera motion intensity level increases, the degree of camera zoom-in also increases. Figure 5 demonstrates that `MotionStone` while maintaining the intensity and speed of object motion, is capable of customizing the intensity and speed of camera motion. In Figure 6, to highlight the control over object motion intensity and speed, we do not include camera motion prompts in the text and set the camera motion intensity level to the minimum value of 1, while varying the object motion intensity level. From the two sets of images, it is clear that as the given object motion intensity level increases, the speed and intensity of object motion in the generated video both become faster and larger.

5. Conclusion

In this work, we propose `MotionStone`, a general image-to-video (I2V) generation framework that enables decoupled modeling and control of video motion. To achieve this, we train a dedicated motion estimator that directly predicts object and camera motion intensities in line with human perception. To address the challenge that human annotators cannot directly label absolute motion intensities, we develop a novel annotation method for video pairs specifically for training the motion estimator. We design a motion estimator with a backbone for video motion representation and disentangled heads to predict object and camera motion, trained using a contrastive learning strategy. Finally, we inject the predicted motion intensities into a diffusion model, thereby improving training convergence and user customization ability. This entire pipeline demonstrates impressive performance across diverse domains and task instructions.

References

- [1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021. 6
- [2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2, 5, 6
- [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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 3
- [4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3
- [5] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. Stablevideo: Text-driven consistency-aware diffusion video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23040–23050, 2023. 2
- [6] Di Chang, Yichun Shi, Quankai Gao, Hongyi Xu, Jessica Fu, Guoxian Song, Qing Yan, Yizhe Zhu, Xiao Yang, and Mohammad Soleymani. Magicpose: Realistic human poses and facial expressions retargeting with identity-aware diffusion. In *Forty-first International Conference on Machine Learning*, 2023. 2
- [7] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James T. Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- α : Fast training of diffusion transformer for photorealistic text-to-image synthesis. *ArXiv*, abs/2310.00426, 2023. 3
- [8] Xi Chen, Zhiheng Liu, Mengting Chen, Yutong Feng, Yu Liu, Yujun Shen, and Hengshuang Zhao. Livephoto: Real image animation with text-guided motion control. In *European Conference on Computer Vision*, pages 475–491. Springer, 2025. 2, 3, 6, 8, 4, 5
- [9] Chia-Chi Cheng, Hung-Yu Chen, and Wei-Chen Chiu. Time flies: Animating a still image with time-lapse video as reference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5641–5650, 2020. 2
- [10] Yung-Yu Chuang, Dan B Goldman, Ke Colin Zheng, Brian Curless, David H Salesin, and Richard Szeliski. Animating pictures with stochastic motion textures. In *ACM SIG-GRAPH 2005 Papers*, pages 853–860. 2005. 2
- [11] 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 e-prints*, pages arXiv–2311, 2023. 2, 3, 5, 6, 8, 4
- [12] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023. 2
- [13] Yuming Fang, Hanwei Zhu, Yan Zeng, Kede Ma, and Zhou Wang. Perceptual quality assessment of smartphone photography. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3677–3686, 2020. 3
- [14] Peng Gao, Le Zhuo, Ziyi Lin, Chris Liu, Junsong Chen, Ruoyi Du, Enze Xie, Xu Luo, Longtian Qiu, Yuhang Zhang, et al. Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers. *arXiv preprint arXiv:2405.05945*, 2024. 3
- [15] Biao Gong, Siteng Huang, Yutong Feng, Shiwei Zhang, Yuyuan Li, and Yu Liu. Check locate rectify: A training-free layout calibration system for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6624–6634, 2024. 3
- [16] 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. *arXiv preprint arXiv:2307.04725*, 2023. 2, 3
- [17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3
- [18] Li Hu. Animate anyone: Consistent and controllable image-to-video synthesis for character animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8153–8163, 2024. 2
- [19] Ziyuan Huang, Shiwei Zhang, Liang Pan, Zhiwu Qing, Mingqian Tang, Ziwei Liu, and Marcelo H Ang Jr. Tada! temporally-adaptive convolutions for video understanding. *arXiv preprint arXiv:2110.06178*, 2021. 2, 4
- [20] Ziqi Huang, Yanan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024. 6, 1, 3, 4
- [21] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021. 3
- [22] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. WebVision Database: Visual Learning and Understanding from Web Data. *ArXiv*, abs/1708.02862, 2017. 6
- [23] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9801–9810, 2023. 3
- [24] Jun Hao Liew, Hanshu Yan, Jianfeng Zhang, Zhongcong Xu, and Jiashi Feng. Magicedit: High-fidelity and temporally

- coherent video editing. *arXiv preprint arXiv:2308.14749*, 2023. 2
- [25] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Rankiqq: Learning from rankings for no-reference image quality assessment. In *Proceedings of the IEEE international conference on computer vision*, pages 1040–1049, 2017. 4
- [26] Xin Ma, Yaohui Wang, Gengyu Jia, Xinyuan Chen, Yuanfang Li, Cunjian Chen, and Yu Qiao. Cinemo: Consistent and controllable image animation with motion diffusion models. *arXiv preprint arXiv:2407.15642*, 2024. 2, 3
- [27] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuanfang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024. 3
- [28] Aniruddha Mahapatra and Kuldeep Kulkarni. Controllable animation of fluid elements in still images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3667–3676, 2022. 2
- [29] Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. Revideo: Remake a video with motion and content control. *arXiv preprint arXiv:2405.13865*, 2024. 2
- [30] Muyao Niu, Xiaodong Cun, Xintao Wang, Yong Zhang, Ying Shan, and Yinqiang Zheng. Mofa-video: Controllable image animation via generative motion field adaptations in frozen image-to-video diffusion model. *arXiv preprint arXiv:2405.20222*, 2024. 2
- [31] Makoto Okabe, Ken Anjyo, Takeo Igarashi, and Hans-Peter Seidel. Animating pictures of fluid using video examples. In *Computer Graphics Forum*, pages 677–686. Wiley Online Library, 2009. 2
- [32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 3
- [33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6, 2
- [34] Weiming Ren, Huan Yang, Ge Zhang, Cong Wei, Xinrun Du, Wenhao Huang, and Wenhui Chen. Consisti2v: Enhancing visual consistency for image-to-video generation. *arXiv preprint arXiv:2402.04324*, 2024. 2
- [35] Yoav Shalev and Lior Wolf. Image animation with perturbed masks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3647–3656, 2022. 2
- [36] Shuwei Shi, Wenbo Li, Yuechen Zhang, Jingwen He, Biao Gong, and Yinqiang Zheng. Resmaster: Mastering high-resolution image generation via structural and fine-grained guidance. *arXiv preprint arXiv:2406.16476*, 2024. 3
- [37] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in neural information processing systems*, 32, 2019. 2
- [38] Aliaksandr Siarohin, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13653–13662, 2021. 2
- [39] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3
- [40] Shuai Tan, Biao Gong, Xiang Wang, Shiwei Zhang, Dandan Zheng, Ruobing Zheng, Kecheng Zheng, Jingdong Chen, and Ming Yang. Animate-x: Universal character image animation with enhanced motion representation. *arXiv preprint arXiv:2410.10306*, 2024. 2
- [41] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 3
- [42] 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. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [43] Xiang Wang, Shiwei Zhang, Changxin Gao, Jiayu Wang, Xiaoqiang Zhou, Yingya Zhang, Luxin Yan, and Nong Sang. Unianimate: Taming unified video diffusion models for consistent human image animation. *arXiv preprint arXiv:2406.01188*, 2024. 2
- [44] Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. Latent image animator: Learning to animate images via latent space navigation. *arXiv preprint arXiv:2203.09043*, 2022. 2
- [45] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *arXiv preprint arXiv:2309.15103*, 2023. 3
- [46] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2
- [47] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7623–7633, 2023. 2, 3
- [48] Wenpeng Xiao, Wentao Liu, Yitong Wang, Bernard Ghanem, and Bing Li. Automatic animation of hair blowing in still portrait photos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22963–22975, 2023. 2
- [49] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20406–20417, 2024. 4

- [50] Junyu Xie, Charig Yang, Weidi Xie, and Andrew Zisserman. Moving object segmentation: All you need is sam (and flow). *arXiv preprint arXiv:2404.12389*, 2024. [4](#)
- [51] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*, pages 399–417. Springer, 2025. [2](#)
- [52] 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. [3](#), [5](#), [6](#), [7](#)
- [53] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023. [2](#), [5](#), [6](#)
- [54] Yiming Zhang, Zhening Xing, Yanhong Zeng, Youqing Fang, and Kai Chen. Pia: Your personalized image animator via plug-and-play modules in text-to-image models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7747–7756, 2024. [2](#), [3](#)
- [55] Zhenghao Zhang, Junchao Liao, Menghao Li, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. *arXiv preprint arXiv:2407.21705*, 2024. [2](#)
- [56] Jian Zhao and Hui Zhang. Thin-plate spline motion model for image animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3657–3666, 2022. [2](#)
- [57] Ruiqi Zhao, Tianyi Wu, and Guodong Guo. Sparse to dense motion transfer for face image animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–2000, 2021. [2](#)

MotionStone: Decoupled Motion Intensity Modulation with Diffusion Transformer for Image-to-Video Generation

Supplementary Material

A. Implementation Details

We supplement more details of the training of motion estimator. For training the motion estimator, we utilize 8 A100 GPUs with batch size 64. The learning rate is set to 5×10^{-6} . To align with the training configuration of `MotionStone`, input videos are cropped to a resolution of 480×720 and sampled to 49 frames. The motion estimator is trained for 10,000 steps using the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We set the weight of regression loss λ to 0.1.

B. Details on the Training Data for Motion Estimator

In this section, we provide more details on the training data for the motion estimator. We ask 15 annotators to participate in this annotation process. The annotators are asked to label video pairs from several aspects: First, they are asked to determine whether the two videos in a pair contain a moving object. A video is considered to have a moving object only if it features a foreground object in motion. Meanwhile, camera motion focuses on the global motion in the scene. If a video in the pair contains a moving object, it is labeled as 1; otherwise, it is labeled as 0. Note that comparisons of object motion between the two videos are only made when at least one video in the pair features a moving object. Next, annotators are tasked with labeling the relative magnitude of the object and camera motion in each video pair. If both videos contain object or camera motion, the corresponding item is annotated based on the annotators’ subjective judgment. If only one video in the pair exhibits object or camera motion, the video with motion is considered significantly greater in the respective category. Specifically, we define the annotations as follows: if the first video shows significantly greater camera or object motion than the second one, it is labeled as 2; if it is only slightly greater, it is labeled as 1. Conversely, if the first video shows significantly or slightly less motion, it is labeled as -2 or -1, respectively. If neither video exhibits object or camera motion, the corresponding item is labeled as 0. During the training process using contrastive learning, this label is employed to amplify the motion differences between two videos. If a specific motion in the first video is significantly greater than that in the second, the corresponding loss is set to twice that of cases with a smaller difference.

After completing one round of annotation, we conduct

a sampling check on 5,000 video pairs, reviewing 20% of them. The investigation achieves an accuracy rate of 95%, meeting the annotation standards. This demonstrates that the annotated data aligns well with human perception of the relative magnitude of object and camera motion in videos.

C. User Study on Comparisons with Existing Alternatives

Since the metrics in VBench [20] cannot fully evaluate the performance of the model, we conduct user studies. We ask 10 annotators to participate in this process. To ensure the generalization of the evaluation, we select a wide variety of real and animated images, including elements such as people, animals, camera movement, plants, and natural landscapes. Twenty image-text prompts are selected and processed by each compared method, including `MotionStone`, generating a total of 100 video clips. Each participant is presented with two videos generated by different methods for the same prompts and asked to choose the one that performed better in four aspects: *Text Consistency* evaluates if the motion and content follow the text prompt. *Image Consistency* assesses the ability to preserve the identity of the reference image. *Content Quality* determines the overall quality of video generation, including visual appeal, definition, and the logical coherence of the generated content. *Motion Quality* evaluates the plausibility and richness of the motion. The pairwise comparison is repeated for all combinations of videos, resulting in C_2^5 comparisons.

As shown in Tab. 4, our method demonstrates superior performance, particularly in terms of Text Consistency, Content Quality and Motion Quality. This highlights the effectiveness of our approach in text-based motion control and the generation of videos with content and motion that align more closely with human perception.

D. Evaluation Metrics

We select several metrics from VBench [20] for quantitative evaluation experiments, including *Background Consistency*, *Aesthetic Quality*, *Imaging Quality*, *Subject Consistency*, *Motion Smoothness*, *Dynamic Degree* and *Temporal Flickering*. It is important to note that, we utilize only its models and evaluation processes, excluding its prompt suite. Consequently, some metrics that strictly require the use of the prompt suite are omitted. The detailed information on each metric is introduced as follows.

Table 4. **Results of user study.** The best results for each column are **bold**. We ask annotators to rate videos based on four aspects: Text Consistency, which assesses how well the motion and content adhere to the textual descriptions; Image Consistency, which evaluates the ability to preserve the identity of the reference image; Content Quality, which focuses on inter-frame coherence and definition; and Motion Quality, which measures the plausibility and richness of the motion.

Method	I2VGEN-XL	SVD	AnimateAnything	CogVideoX-5B	MotionStone
Text Consistency \uparrow	32.50%	39.38%	25.00%	63.13%	90%
Image Consistency \uparrow	27.50%	36.88%	56.25%	62.50%	66.88%
Content Quality \uparrow	31.25%	45.63%	33.13%	63.13%	76.88%
Motion Quality \uparrow	26.25%	48.13%	39.38%	61.25%	75.00%



Figure 7. **Qualitative ablation for proposed modules.** Using inter-frame SSIM [8] and feature difference [11] (*MotionStone w/ SSIM* and *MotionStone w/ S*) causes varying degrees of unnatural background motion (In the first row, the snow block in the upper left corner of the third column appears. In the second row, background motion blur is observed.) and does not follow the camera motion described in the text prompt. Omitting the proposed motion estimator (*MotionStone w/o M*) and the decoupled injection method (*MotionStone w/o D*) results in issues such as generating static video and confusion or overlap between camera motion and object motion control, respectively. These approaches also fail to follow the camera motion described in the text prompt successfully.

Background Consistency. This metric measures the temporal consistency of the background scenes by calculating CLIP [33] feature similarity across frames.

Aesthetic Quality. This metric assesses the human-perceived artistic and aesthetic value of each video frame utilizing the LAION aesthetic predictor. This tool captures

various aesthetic dimensions, including composition, color richness and harmony, photorealism, naturalness, and the artistic quality of the video frames.

Imaging Quality. Imaging quality pertains to distortions such as over-exposure, noise, and blur observed in the generated frames. This metric measures this using the MUSIQ [21] image quality predictor, which is trained on the SPAQ [13] dataset.

Subject Consistency. This metric calculates the DINO [4] feature similarity across frames to evaluate the consistency of a subject’s appearance throughout the video.

Motion Smoothness. Evaluating the smoothness of motion in generated videos and its adherence to real-world physical laws is crucial. To assess this, this metric leverages motion priors from the video frame interpolation model [23].

Dynamic Degree. As a completely static video might perform well in the previously mentioned temporal quality metrics, it is essential to assess the level of dynamics (i.e., the presence of significant motions) in the generated videos. To achieve this, this metric uses RAFT [41] to estimate the extent of dynamics in the synthesized outputs.

Temporal Flickering. Generated videos may display imperfect temporal consistency, particularly in local and high-frequency details. To quantify this, this metric extracts static frames and calculates the mean absolute difference between them.

E. Limitation

Although MotionStone has made notable progress in I2V generation and motion intensity control, it still faces several limitations. First, MotionStone is built upon CogVideoX, and due to constraints in memory and computational resources, it can only generate videos of approximately 6 seconds in length at a specific resolution. We believe that as the computational demands of foundational video generation models decrease in the future, MotionStone will be able to generate longer videos with higher resolutions. Furthermore, with reduced computational resource requirements, it will be feasible to design a larger motion estimator and leverage more extensive training datasets to develop a more powerful model. The enhanced motion estimator could better assist I2V generation, and we are confident that such advancements will lead to superior performance.

F. More Experiments

In this section, we first present additional ablation studies, including more detailed qualitative and quantitative experiments, as well as an evaluation of the motion strength error of our proposed motion estimator compared to previous motion intensity estimation methods. Subsequently, we provide more specific quantitative comparison results.

Finally, we provide additional cases to showcase the generative capabilities of MotionStone.

F.1. More Ablations

More Quantitative and Qualitative Results. We first supplement additional quantitative metrics on VBench [20] to demonstrate the superiority of MotionStone. As shown in Tab. 5, benefiting from the support of the motion estimator and the decoupled injection method, MotionStone outperforms other motion intensity modulation approaches and models without these strategies in terms of generated quality, inter-frame consistency of subjects and backgrounds, motion magnitude, and temporal quality.

Furthermore, we conduct qualitative ablation studies. As shown in Fig. 7, we generate videos using prompts containing both camera and object motions. We observe that MotionStone w/ S and MotionStone w/ SSIM fail to follow the camera motion described in the text prompt. Additionally, MotionStone w/ S exhibits unnatural motion in background objects (e.g., the snow block in the upper left corner of the third column), while MotionStone w/ SSIM displays motion blur issues. These problems are common to non-decoupled motion intensity modulation methods, as they inadvertently cause undesirable background motion while animating the subject. We observe that the MotionStone w/o M model, which does not utilize the motion estimator, generates static frames without responding to the specified motion intensity. This issue arises because, during training, the model does not receive varying signals corresponding to different motion intensities but rather a constant signal. As a result, the model fails to interpret the provided intensity control signals and is unable to model motion intensity accordingly. MotionStone w/o D exhibits excessive motion, affecting both the object and the camera motion. Moreover, it fails to follow the text prompt to perform a zoom-out motion, instead generating an opposite camera motion. This issue stems from the lack of decoupled injection of camera and object motion intensity signals. Without clear separation, the model struggles to associate the signals with the specific motion components they are meant to control, leading to unpredictable overlap or confusion. Consequently, the generated video lacks coherent and orderly control. In contrast, MotionStone accurately follows the object and camera motion descriptions provided in the text prompt and generates visually appealing and motion-consistent videos based on the specified motion intensities. This demonstrates the effectiveness of the proposed modules.

Motion Intensity Guidance. We provide an additional example to demonstrate the decoupled control capabilities of MotionStone for object motion and camera motion intensities. As shown in Fig. 8, in the first two rows, the text prompt does not specify camera motion, so the camera

Table 5. **More quantitative ablation results on VBench [20].** The best results for each column are **bold**. Motion Estimator (M), Decoupled injection strategy (D). SSIM and S mean previous motion modeling methods: inter-frame SSIM [8] and feature difference [11] respectively.

Method	Background Consistency	Aesthetic Quality	Imaging Quality	Subject Consistency	Motion Smoothness	Dynamic Degree	Temporal Flickering
MotionStone w/o M	95.13%	45.61%	60.15%	93.34%	98.51%	43%	96.51%
MotionStone w/o S	94.97%	46.13%	60.73%	92.99%	98.48%	42%	96.42%
MotionStone w/ SSIM	92.99%	45.72%	54.75%	88.96%	97.51%	47%	93.54%
MotionStone w/o D	94.03%	46.27%	58.73%	92.54%	97.59%	48%	95.20%
MotionStone	95.76%	46.78%	62.29%	94.56%	98.96%	48%	97.41%



Figure 8. **Illustrations of object and camera motion intensity guidance.** MotionStone can decouple and independently control camera motion and object motion intensities. When either camera motion or object motion is increased, the generated videos exhibit excellent adherence to the respective motion changes.

motion intensity is set to the minimum. By increasing the control of object motion intensity, it is evident that the camel moves faster. In contrast, in the last two rows, we introduce camera motion descriptions in the text prompt and adjust the camera motion intensity while reducing the control of object motion intensity. It is observable that as the object motion intensity decreases from 7 to 4, the camel

slows down. Meanwhile, as the camera motion intensity increases from 3 to 9, the camera pans to the right more quickly. These examples strongly demonstrate the ability of the MotionStone to decouple and independently control camera and object motions in generated videos.

Furthermore, we compare the performance of different motion intensity guidance methods. Using predefined mo-

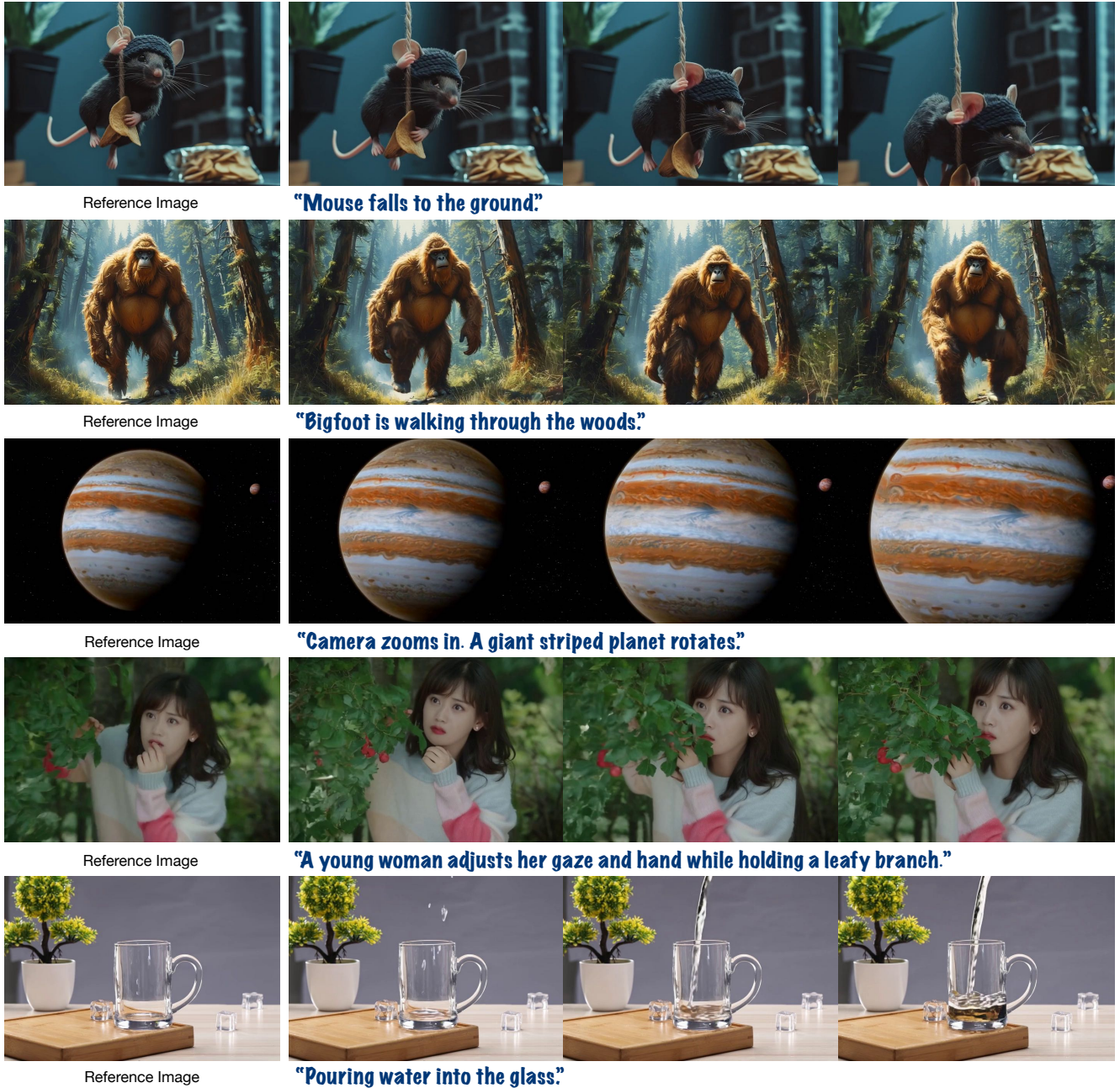


Figure 9. **More cases generated by MotionStone.** MotionStone demonstrates impressive generation quality across various scenarios. Here, the default object motion intensity or camera motion intensity (if applicable) is set to 5.

Table 6. **Ablation on motion intensity guidance.** Compared to previous methods, our motion estimator achieves more precise control over motion intensity, generating videos with camera or object motion that better aligns with user requirements.

Method	Motion Strength Error
Feature Difference (S) [11]	11.55
SSIM [8]	11.27
Ours	2.52

tion intensity values, we generate videos and subsequently apply a motion estimator to obtain the corresponding motion intensities. The mean squared error (MSE) between the generated video intensities and the input values is then calculated. As shown in Tab. 6, the motion estimator proposed in this work provides more stable motion guidance and ensures that the motion intensities in the generated videos align more closely with the user-specified values.

Table 7. **More quantitative comparison results on VBench [20].** The best results for each column are **bold**.

Method	Background Consistency	Aesthetic Quality	Imaging Quality	Subject Consistency	Motion Smoothness	Dynamic Degree	Temporal Flickering
I2VGen-XL [53]	90.93%	40.14%	58.35%	86.97%	97.02%	44%	95.24%
SVD [2]	93.17%	42.38%	59.61%	93.23%	97.39%	40%	94.70%
AnimateAnything [11]	93.89%	46.04%	61.69%	93.72%	97.58%	4%	95.48%
CogVideoX-5B [52]	94.91%	45.88%	61.99%	94.39%	98.76%	36%	96.73%
MotionStone	95.76%	46.78%	62.29%	94.56%	98.96%	48%	97.41%

F.2. More Results

We supplement additional quantitative comparison results across more evaluation dimensions on VBench [20], as shown in Tab. 7. MotionStone demonstrates superior performance in terms of temporal quality and motion magnitude of the generated videos compared to previous methods.

We also provide additional examples generated by MotionStone, as shown in Fig. 9. These include real human figures, anime-style characters, animals, and natural scenes. MotionStone demonstrates remarkable capabilities in conjuring entirely new content out of thin air.

We provide the original video cases showcased in the paper within the supplementary materials. The detailed video effects can be found in the designated folder.