2025-4-13

EquiVDM: Equivariant Video Diffusion Models with Temporally Consistent Noise

Chao Liu, Arash Vahdat

Temporally consistent video-to-video generation is essential for applications of video diffusion models in areas such as sim-to-real, style-transfer, video upsampling, etc. In this paper, we propose a video diffusion framework that leverages temporally consistent noise to generate coherent video frames without specialized modules or additional constraints. We show that the standard training objective of diffusion models, when applied with temporally consistent noise, encourages the model to be equivariant to spatial transformations in input video and noise. This enables our model to better follow motion patterns from the input video, producing aligned motion and high-fidelity frames. Furthermore, we extend our approach to 3D-consistent video generation by attaching noise as textures on 3D meshes, ensuring 3D consistency in sim-to-real applications. Experimental results demonstrate that our method surpasses state-of-the-art baselines in motion alignment, 3D consistency, and video quality while requiring only a few sampling steps in practice.

1. Introduction

Video-to-video generative models have a wide range of applications, including sim-to-real, style transfer, and video upsampling. Video diffusion models trained in conditional settings have become the de facto approach for addressing these tasks [1, 2, 3, 4, 5, 6, 7]. However, following the original formulation of diffusion models for images and videos [7, 8, 9, 10], these models use independent Gaussian noise in their noising process. To achieve temporal consistency, they often incorporate 3D convolutions [5, 11] or attention layers [7] into diffusion-based frameworks to better capture and propagate spatiotemporal information. While these designs can improve temporal consistency, they typically rely on extensive training on large-scale, high-quality video datasets [4, 12] to effectively learn to generate realistic frames with natural, coherent motion patterns from independent noise.

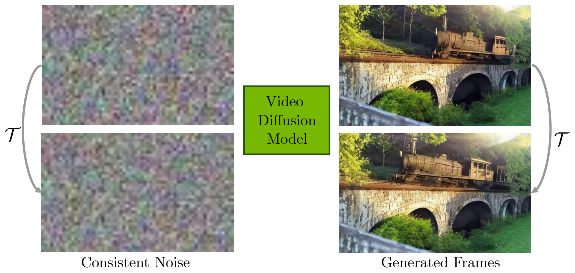

Figure 1 | A video diffusion model that is equivariant to input spatial transformations generates videos with the same spatial transformation when provided with temporally consistent noise.

An alternative line of work aims to generate temporally consistent frames by directly sampling from temporally correlated noise. This is particularly appealing for video-to-video applications where an input video can be used to drive the coherent noise. In particular, [13, 14, 15] propose methods to warp noise across frames while preserving its spatial Gaussianity, then use a pretrained *image* diffusion model to denoise the warped noise, thereby inducing consistent transformations in the generated frames. However, as discussed by [14], standard image diffusion networks are not intrinsically equivariant to noise-warping transformations, due to their highly nonlinear layers. Consequently, these approaches need sampling-time guidance or regularization strategies to achieve approximate equivariance, which can introduce additional hyperparameters and complexity into the generation pipeline.

These recent trends raise two questions: 1) Do we need warped noise in the era of big video diffusion models such as Sora, Cosmos, Wan, and CogVideoX trained on massive video datasets? 2) What is the role of equivariance in generating consistent videos? To answer these, we study the role of warped noise in training video diffusion diffusion. We show that the equivariance to the spatial warping transformation of the input is *learned* without modifying the training objective of conventional video diffusion models (VDMs) by simply switching the noising process from independent noise to warped noise. Unlike prior methods that require specialized modules [16, 17, 18, 19, 20, 21, 22], our approach introduces equivariance as an inherent property of the VDM itself during training. Thus, the temporal coherence comes at no extra cost in terms of model complexity or runtime overhead, thereby providing a straightforward yet effective solution for high-fidelity video generation. We name our video diffusion models trained with consistent noise as *EquiVDM*.

For video-to-video applications, we generate temporally consistent noise using motion vectors extracted from a reference video or input video respectively [13, 14]. Additionally, we propose a novel approach to construct 3D-consistent noise for 3D-consistent video generation. Specifically, we attach Gaussian noise as textures to 3D meshes and render the resulting noise images from various camera viewpoints as input to the diffusion model. We train EquiVDM on 2D videos without any 3D information using motion-based warped noise and we switched to the 3D consistent noise at the inference time. We demonstrate that EquiVDM leverages its equivariance properties to align the generated video frames faithfully according to the underlying 3D geometry and camera poses, in applications such as sim-to-real where a 3D mesh of the input scene is available.

We empirically demonstrate that EquiVDM excels at producing videos with better motion following and higher visual quality compared to state-of-the-art methods, even without additional modules or auxiliary loss terms. Notably, our base EquiVDM without any explicit input video conditioning outperforms specialized video-to-video models trained with independent noise. Moreover, when EquiVDM is adapted for video-to-video tasks, its performance improves even further. Finally, we showcase that EquiVDM can generate coherent video sequences in very few sampling steps—significantly reducing computation time while maintaining high-quality outputs.

In summary, our contributions are: (i) We propose EquiVDM, the video diffusion model equivariant to the warping transformation of the input noise, and show that it can be trained with warped noise using the vanilla video denoising loss, without any additional regularization. (ii) We propose to render 3D-consistent noise from meshes attached with Gaussian noise on the surface and generate 3D consistent video using EquiVDM trained with 2D videos only. (iii) We demonstrate that EquiVDM can generate videos with better motion following and higher quality compared to state-of-the-art methods. In particular, the base EquiVDM even outperforms existing models that require additional modules to encode per-frame dense conditions. Additinonal control modules using dense frame conditions such as soft-edge further improves our model. (iv) We showcase that EquiVDM with warped noise can generate videos within very few sampling steps without compromising the quality, opening up a new perspective into accelerated sampling with non-conventional noise distribution.

2. Related works

Controllable video generation Controllable video generation extends image-generation methods by leveraging additional constraints to guide generation. Prior works incorporate dense frame-wise signals such as depth or edge maps by adding modules to text-to-video backbones or by introducing temporal blocks to capture motion [23, 16, 19, 20]. For user-defined sparse trajectories (e.g., drag-and-drop), researchers encode these trajectories through auxiliary modules or flow-completion strategies, then fuse them into the diffusion model's latent features [24, 25, 17, 21]. Some approaches refine alignment with 2D Gaussian or bounding-box constraints, bypassing the need for an initial frame or applying sampling-time guidance to precisely follow the specified motion [26, 27].

Taming noise for rendering and generation Generating noise with specific properties such as independence and temporal consistency is a crucial step for diffusion model based video generation, as well as rendering in graphics. For example, [28] improve the rendering efficiency and stability by introducing a spatiotemporal noise generation pipeline for stochastic rendering. [29] propose a fast coherent noise generation method for non-photorealistic rendering. [30, 31] focus on 2D blue noise generation for more efficient ray-tracing based rendering pipeline. [32] extend the blue noise generation to the diffusion model based video generation given that the blue noise preserves more high-frequency information than Gaussian noise. [33] study the noise prior and introduce temporally correlated noise in video diffusion without any spatial transformation. [34, 35] explore the residual noise between frames for video generation with more temporal consistency. In [36] the temporal correlation of the noise for video generation is modeled directly to improve temporal consistency.

Getting consistent Gaussian noise for image sequence and video generation using diffusion models has been getting more attention recently. [13] introduce a warping-based Gaussian noise generation method based on conditional upsampling for image sequence generation. The warped noise theoretically preserves Gaussianity for each frame while being temporally consistent across frames. [15] improve the efficiency of the warping-based method by operating directly in the continuous domain thus avoiding the need for conditional upsampling. [14] proposes a consistent Gaussian noise generation method alternatively based on Gaussian process.

In concurrent works, [37] and [38] utilize the temporal consistent noise for 3D asset and video generation. More specifically, [37] propose a method for text-to-3D generation by distilling from a pretrained image diffusion model using multi-view consistent noise. [38] finetune a pretrained video diffusion model using warped noise for motion control. In this work, we focus on the video diffusion models that are equivariant to the warping operation of the input noise and show that the equivariance can be learned by using the original loss without any modification or new modules.

3. Preliminary

Video Diffusion Model Considering the task of generating a video $\mathbf{V} = (V^{(0)}, V^{(1)}, \cdots, V^{(K)})$ where $V^{(k)}$ is the k-th frame of the video, given input conditions $\mathbf{c}$ (text prompts, control frames, etc.), we train a video diffusion model $D_\theta(\mathbf{V}_t; \mathbf{c}, t)$ to predict the clean video frames $V^{(k)}$ from the noisy video frames $V_t^{(k)} = V^{(k)} + \epsilon^{(k)}$ where $\epsilon^{(k)}$ is the Gaussian noise added to the k-th frame (e.g., $\epsilon^{(k)} \sim \mathcal{N}(\mathbf{0}, t\mathbf{I})$). In the following, we drop the input conditions $\mathbf{c}$ and diffusion time t for brevity. The VDM is trained by minimizing the per-frame denoising loss:

$$\mathcal{L} = \mathbb{E}_{p(\mathbf{V}, \mathbf{V}_t)} \sum_k \left\| D_\theta^{(k)}(\mathbf{V}_t) - V^{(k)} \right\|_2^2. \tag{1}$$

which has the same minimizer as [39]:

$$\mathcal{L} = \mathbb{E}_{p(\mathbf{V}_t)} \sum_k \left\| D_\theta^{(k)}(\mathbf{V}_t) - \mathbb{E}_{p(\mathbf{V}|\mathbf{V}_t)} \left[V^{(k)} \right] \right\|_2^2. \tag{2}$$

After training, a video can be generated by iteratively denoising a randomly sampled Gaussian noise following the sampling schedule. Please refer to [5, 4] for more details.

Integral Noise Due to the temporal consistency of video frames, the transformation of the image regions visible to pairs of two frames in the video can be modeled by a warping operation:

$$\mathcal{T} \circ I(\mathbf{p}) = I\left(\mathcal{T}^{-1}(\mathbf{p})\right)$$

where $I(\mathbf{p})$ is the source image, $\mathcal{T}$ is the warping operation extracted from a driving video or derived from given 3D mesh and camera trajectory, and $\mathcal{T} \circ I(\mathbf{p})$ is the warped image, usually computed through interpolation for natural images. [13] show that the interpolation-based warping operation breaks down the Gaussianity of the noise and makes the input noise spatially correlated. To tackle this issue, the authors proposed the noise transport equation (NTE) for warping the noise while keeping its Gaussianity within each frame. In NTE, the warped noise value $\mathcal{T} \circ \epsilon(\mathbf{p})$ is

$$\mathcal{T} \circ \epsilon(\mathbf{p}) = \frac{1}{\sqrt{|\Omega_\mathbf{p}|}} \sum_{A_i \in \Omega_\mathbf{p}} \epsilon_{\text{up}}(A_i)$$

where $\Omega_\mathbf{p}$ is the set of pixels in the source noise covered by the deformed pixel after warping; $|\Omega_\mathbf{p}|$ is the number of pixels in the covered area; $\epsilon_{\text{up}}(A_i)$ is the *stochastically upsampled* noise value of the the deformed i-th pixel. Please check [13] for additional detail.

4. Method

In this section, we first introduce EquiVDM, a video diffusion model equivariant to the warping transformations of the input noise. Then we describe how EquiVDM can be used for 3D consistent video generation by attaching the Gaussian Noise to the 3D mesh surface. Last, we will show how to better train EquiVDM to account for the inconsistency in the latent frames obtained from video encoders.

4.1. Video generation with temporally consistency noise

Prior works [13, 14] have previously introduced methods for getting temporally consistent noise while preserving its Gaussianity within each frame, making it possible to generate images following the motion patterns of the input warped noise. However, image diffusion models (IDMs) are not generally equivariant to input noise warping due to the generic layers in the network. This leads to inconsistency and even abrupt changes like flickering in the generated images. To tackle this issue, [14] introduce a sampling-time guidance to regularize generated pixel values using the optical flow.

To avoid additional regularization or post-training guidance during sampling, two approaches can be applied: (1) modify the network architecture to make it equivariant to input transformations; (2) learn the equivariance from training data through specific loss or training schemes. For the first approach, switching to equivariant layers needs heavy retraining. Moreover, building an equivariant diffusion architecture for even the simplest transformations (e.g. spatial shift) is generally a challenging open problem. Thus, we focus on the second approach since it does not require any architecture changes, making it possible to finetune directly from a pretrained model.

Our key result, summarized in the following theorem, is that the conventional denoising loss function in Equation 1 is in fact training the VDM to be equivariant, as long as the input noise is also consistent. This implies that we do not need to introduce any special loss, hyperparameters, or regularization. We can train VDMs simply with consistent noise to learn equivariance from data.

Theorem 4.1. *Given the temporally consistent video with K frames $\mathbf{V} = (V^{(0)}, V^{(1)}, \cdots, V^{(K)})$ with cross-frame warping transformation $V^{(k)} = \mathcal{T}_k \circ V^{(0)}$, and the noisy video $\mathbf{V}_t$ with consistent added noise $\mathbf{N} = (\epsilon^{(0)}, \epsilon^{(1)}, \cdots, \epsilon^{(K)})$ related with the same warping transformation $\epsilon^{(k)} = \mathcal{T}_k \circ \epsilon^{(0)}$, the minimizer of the denoising loss in Equation 1, is a video diffusion model D_θ that is equivariant to the transformation $\mathcal{T}_k$, such that $D_\theta^{(k)}(\mathbf{V}_t) = \mathcal{T}_k \circ D_\theta^{(0)}(\mathbf{V}_t)$, with $D_\theta^{(k)}(\mathbf{V}_t)$ being the k-th frame of the denoised video.*

Proof. Recall, that minimizing Equation 1 with respect to θ is equivalent to minimizing Equation 2. Given the warping transformation $\mathcal{T}_k$ from frame $V^{(0)}$ to $V^{(k)}$, the expected value of the k-th frame of the video can be written as:

$$\mathbb{E}_{p(\mathbf{V}|\mathbf{V}_t)}[V^{(k)}] = \mathbb{E}_{p(\mathbf{V}|\mathbf{V}_t)}[\mathcal{T}_k \circ V^{(0)}] = \mathcal{T}_k \circ \mathbb{E}_{p(\mathbf{V}|\mathbf{V}_t)}[V^{(0)}],$$

obtained from the linearity of the expectation and warping operation respectively. Thus, we can rewrite Equation 2 as:

$$\mathbb{E}_{p(\mathbf{V}_t)} \sum_k \left\| D_\theta^{(k)}(\mathbf{V}_t) - \mathcal{T}_k \circ \mathbb{E}_{p(\mathbf{V}|\mathbf{V}_t)} \left[V^{(0)} \right] \right\|_2^2 \quad (3)$$

which is minimized when $D_\theta^{(0)}(\mathbf{V}_t) = \mathbb{E}_{p(\mathbf{V}|\mathbf{V}_t)} \left[V^{(0)} \right]$ for the first frame and $D_\theta^{(k)}(\mathbf{V}_t) = \mathcal{T}_k \circ D_\theta^{(0)}(\mathbf{V}_t)$ for all subsequent frames. □

The theorem formally shows that the common denoising loss with warped noise trains VDMs to generate temporally consistent video frames that follow the motion patterns of the input noise. This suggests a simple recipe for training EquiVDM. We use the standard denoising loss in Equation 1 where noise is constructed by warping the noise from the first frame using motion vectors obtained from a driving video.

4.2. Video generation with 3D consistent noise

Theorem 4.1 can be used for 3D consistent video generation from 3D consistent noise. To this end, we attach a Gaussian noise texture to the 3D mesh surface. We first render the Gaussian noise image given the 3D mesh, camera pose and intrinsics. The warping transformation from the UV texture map to the image planes can be determined by the rasterization process. Then we use the Noise Transport Equation to warp the Gaussian noise from the texture map to the image planes to get 3D consistent noise maps $\mathbf{N} = (\mathcal{T}_{0,uv} \circ \epsilon_{uv}, \mathcal{T}_{1,uv} \circ \epsilon_{uv}, \cdots, \mathcal{T}_{K,uv} \circ \epsilon_{uv})$, where ϵ_{uv} is the Gaussian noise texture map, and $\mathcal{T}_{k,uv}$ is the warping transformation from the UV texture map to the image plane for the k-th camera view, as shown in Figure 2.

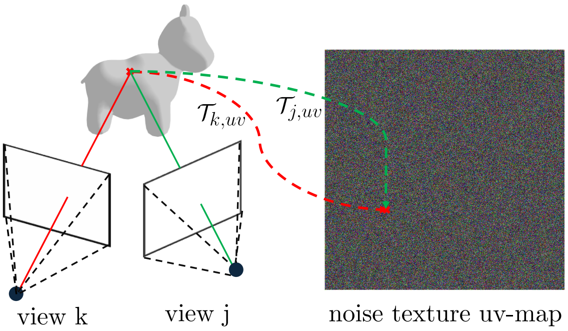

Figure 2 | Render 3D consistent noise from meshes attached with noise texture map. A Gaussian noise texture is attached to the 3D mesh surface. 3D consistent noise map for different camera views can be computed by warping the uv texture map to the image plane. The warping operation is defined by the 3D mesh, camera pose and intrinsics.

With VDMs trained with temporally consistent noise, we can generate 3D consistent video frames from the 3D consistent noise maps. Given that the uv-to-image warping transformation is bijective and intertible, we define the corresponding image-to-uv warping as $\mathcal{T}_{k,uv}^{-1}$. For a pair of close-by camera views, their 3D consistent noise maps can be related by:

$$\epsilon^{(k)} = \mathcal{T}_{k,uv} \circ \epsilon_{uv} = \mathcal{T}_{k,uv} \circ \mathcal{T}_{j,uv}^{-1} \circ \epsilon^{(j)} = \mathcal{T}_{k,j} \circ \epsilon^{(j)}$$

where $\mathcal{T}_{k,j} = \mathcal{T}_{k,uv} \circ \mathcal{T}_{j,uv}^{-1}$ is the effective warping operation from the j-th image to the k-th image. Here we assume that for close-by camera views, most regions in one view are visible in the other. This assumption is reasonable for video sequences with FPS that is high enough compared to the camera motion. Then based on Theorem 4.1, we can generate the corresponding video frames related by $F_k = \mathcal{T}_{k,j} \circ F_j$, which demonstrates the same 3D consistency as the input noise maps. Note that although the pattern of the uv-to-image warping $\mathcal{T}_{k,uv}$ is quite different from the frame-to-frame warping patterns in the training video, the 3D consistency of the generated video can still be preserved without fine-tuning, since the effective image-to-image warping operation $\mathcal{T}_{k,j}$ is similar to the ones in the training videos.

4.3. Independent noise addition

Although the theory suggests that training VDMs with temporally consistent noise encourages them to be equivariant to the spatial warping in the input, our experiments show that they can struggle with generating high-quality videos in practice. We hypothesize that several factors break the assumptions in warped noise: (1) The errors in optical flow can result in errors in warping transformation estimation. (2) Suc-

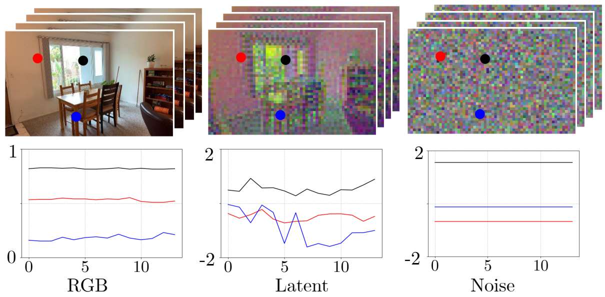

Figure 3 | The values of three tracked points in the video frames in the pixel, latent and noise videos. The variantion in the latent video is much larger than the one in the pixel and noise videos due to the compression in the latent space.

cessive frames in natural videos do not have perfect 1-to-1 mappings as the camera movement can make some pixels visible and some hidden. (3) The optical flow is estimated in the pixel space while virtually all VDMs are defined in a latent space.

In Figure 3, we study the effect of applying warp noise to a latent encoding of a video. Here we show the values of three tracked pixels in the video frames in the RGB, latent, and corresponding noise spaces. Although the values of the tracked pixels in the RGB and noise space are by construction consistent across frames, we observe a large variation in the latent space (middle figure). This indicates that latent embeddings of the tracked pixels have additional high-frequency variations in the temporal direction that are not accounted for when adding the constant warped noise. However, diffusion models require a forward process that destroys all information in all frequencies such that the generative model can learn to generate them in the reverse process [40, 41].

In order to tackle this issue, we propose to add independent noise to each frame along with the temporal consistent ones during training. More specifically, the added noise becomes

$$\epsilon = \beta \epsilon_{\text{warp}} + \sqrt{1-\beta^2}\ \epsilon_{\text{ind}}, \tag{4}$$

where $\beta \in [0, 1]$ is a hyperparameter controlling the strength of the temporal consistent noise, ϵ_{ind} is the independent noise. Another perspective on the added independent noise is that it expands the manifold of the noise, such that the expanded manifold can better cover and destroy the latent encoding compared to the one in the warped noise, which spans a much smaller manifold space due to the temporal correlation. We set β to 0.9 in all the experiments unless specified otherwise. This corresponds to a small amount of added independent noise.

5. Experiments

We evaluate our method on video generation task with static scenes with only camera motion, as well as more general in-the-wild scenes with both camera motion and dynamic scenes. Then we validate the performance of our method with 3D meshes attached with Gaussian noise as textures. Finally, we provide ablation studies to show the effectiveness of each component in our method.

5.1. Experiment Setup

Datasets and Metrics We curate our dataset for training from the training set of RealEstate-10k [42], OpenVideo-1M [43] and VidGen-1M [44] datasets. The RealEstate10K dataset contains about 80k videos of static real estate scenes, while OpenVideo-1M and VidGen-1M each contains around 1M in-the-wild videos including both static and dynamic scenes. For evaluation, we use the test set of RealEstate10K for egomotion-only video generation, and Youtube-VIS 2021 [45] for in-the-wild video generation. We use LLaVA-NeXT [46] for video captioning for datasets without captions. For efficiency, we extract the video captions for every 10 frames assuming that the videos are temporally consistent and the contents do not change too much.We train and evaluate two models separately for static and in-the-wild scenes in order to test if the VDM can learn the equivariance to warping transformations of different types and complexities.

In order to validate the performance of our method on 3D consistent video generation given meshes with Gaussian noise texture, we render the noise inputs for the evaluation set of ScanNet++ [47] dataset using the dense 3D mesh and the camera trajectory for the IPhone modality for which both the dense camera trajectory and the corresponding RGB images are available, such that we can directly compare the generated videos with the ground truth. We perform mesh pre-processing such that the texture mapping and the noise rendering are more well-conditioned and robust to the holes and sharp edges in the given meshes. We use the model trained with in-the-wild videos for this task and the following ablation experiments.

We use FID [48], FVD [49] to measure the quality of the generated videos, and CLIP [50] score to measure the similarity between the generated videos and the ground truth videos. To measure the temporal consistency and the alignment of the motions between the generate and ground truth videos, we first extract the dense optical flow from the frames in the ground truth video, then use it to warp the generated videos accordingly, and compute the PSNR and SSIM scores between the warped and the target frames in the generated video. After applying the warping operation using the optical flow from the ground truth video, if the generated video follows the motion pattern in the ground truth video and the video is temporally consistent, the corresponding pixels in the source and target frames would be close hence we have higher PSNR and SSIM scores. Please refer to the Appendix for a illustration of the temporal consistency metrics.

Training Details We train the EquiVDM by finetuning from the pre-trained VideoCrafter2 [51] model with the temporal consistent noise as the input as describe in Section 4. For the controlled video generation where the a seperate control signnal such as dense depth or edge map is available, we add and finetune the additional modules from CtrlAdapter [19] while keeping the rest of the model frozen. We use the AdamW optimizer [52] with a learning rate of 10^{-4} for the finetuning the base model, and 2×10^{-5} for the finetuning the added control modules. The model is finetuned on 64 Nvidia A100 GPUs for around 200k iterations.

5.2. Video Generation

We first evaluate whether EquiVDM with warped noise input can improve the video generation performance compared to methods using independent noise. In particular, we focus on whether it can learn to generate better videos in terms of semantic and motion alignment from the consistent noise input with motion information. To this end, we compare the performance of our method with state-of-the-art VDMs with both UNet and DiT backbones [53, 54, 55, 11, 51]. The quantitative results are listed in Table 1. The CLIP score improvements show that the noise-equivariant model can learn to infer the semantic information from the consistent noise input; while the PSNR and SSIM improvements show the noise-equivariance properties emerges with training the VDMs with the consistent noise input, thus the motions in the input noise and the generated videos are aligned. The improvement over both video quality and temporal alignment is also observed in the concurrent work [38].

We then evaluate our method on the controlled video generation task, where dense conditioning frames such as soft-edge maps are available. In particular, we use canny edge and soft-edge maps from [56] as the control frames. We compare our method with models with additional control modules [23, 16, 19]. The qualitative results are shown in Figure 4. For our method, EquiVDM-base generates videos from warped noise using text-only prompts, while EquiVDM-full has the additional input conditioning, finetuned from CtrlAdapter [19]. EquiVDM-full achieves the best temporal consistency for textures (*e.g.* the the patterns of the cows, the grass texture), as well as the motion alignment (*e.g.* the motion of legs of the cows, the orientation of rabbit's head) with the ground truth video where the warping optical flow is extracted from. In addition, EquiVDM-base without additional modules achieves on-par or better performance compared to the models with dedicated control modules.

The quantitative results are listed in Table 2. As shown, our method achieves the best performance on both semantic and motion metrics on both static and in-the-wild scenes. A key observation is that even without the additional control modules, our method (EquiVDM-base in Table 1) can already perform on-par or better than the compared methods with dedicated control modules in Table 2. This manifests that EquiVDM can learn to generate better videos by taking advantage of the temporal correlation from the warped noise input. It also indicates that the temporal correlation in the warped noise can serve as a strong prior for both the motion pattern and semantic information in addition to motions.

Another observation is that for our method, the performance of the controlled model is generally better than the base model, indicating that the benefit of equivariance is complementary to the additional conditioning modules. As a result, for video-to-video generation tasks, we can improve the performance by making the full model noise-equivariant without any architecture modification to it.

5.3. 3D Consistent Video Generation

An important application of the noise-equivariant model is to generate 3D-consistent videos by attaching noise to 3D meshes and rendering 2D noise images given the camera trajectory and 3D scene layout. The 3D consistent videos then can be generated by EquiVDM. Compared to video-to-video generation approaches using synthetic videos for the sim-to-real task, our method eliminates the need for synthetic video rendering during training and sampling, which often require detailed texture maps, lighting information in addition to 3D meshes, and more importantly paired simulation and real data. In our case EquiVDM can be trained with in-the-wild 2D videos, and used for 3D consistent video generation, without any additional finetuning on 3D data, which is challenging and expensive to obtain at scale. The quantitative results are listed in Table 3. Our method (EquiVDM-mesh) outperforms the other methods with the same dense frame control signals. Additionally, we provide results for the flow-based variant of our method

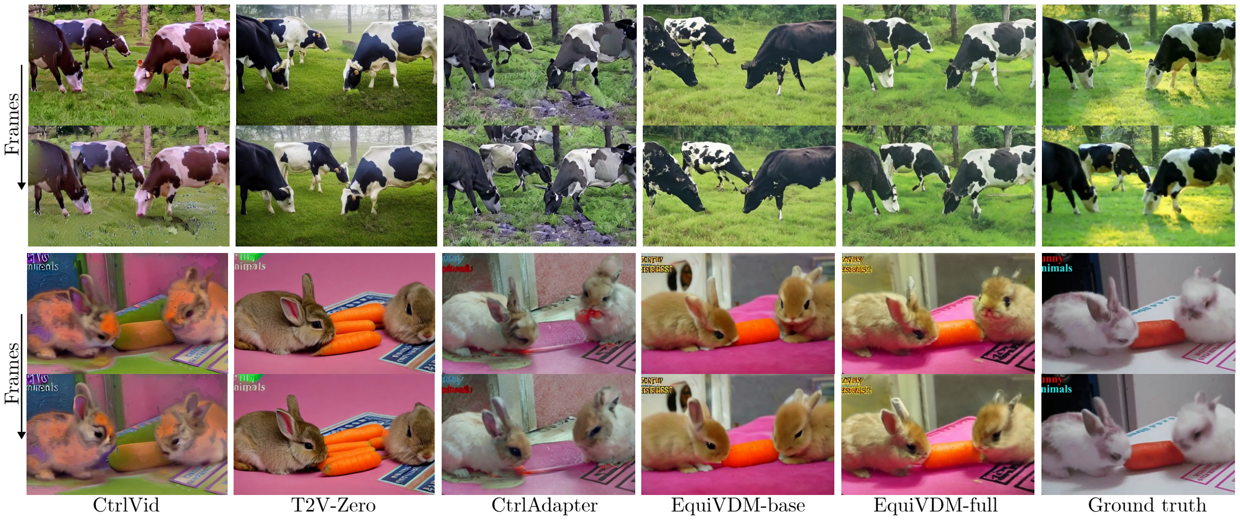

Figure 4 | Frames from the generated videos with different methods. CtrlVid [23], T2V-Zero [16], CtrlAdapter [19] and EquiVDM-full used soft-edge map as control signal for each frame along with the text prompt. EquiVDM-base only used text prompt. EquiVDM-full achieves the best consistency over the frames (*e.g.* textures of the cows), and alignment with the motion patterns in the ground truth videos. In addition, while EquiVDM-base only uses the text prompt, it achieves on-par or better performance compared to previous methods using control frames.

Table 1 | Video generation performance for models with text prompt only.

	Static Scenes				In-the-wild Scenes			
Method	**FID**	**FVD**	**CLIP**	**PSNR**	**FID**	**FVD**	**CLIP**	**PSNR**
VC2 [51]	**23.36**	1882	0.8053	25.02	41.23	4565	0.6500	19.33
Show-1 [53]	64.06	2740	0.7925	25.40	34.83	5422	0.6908	20.59
Pyramid-flow [54]	66.14	3078	0.7301	25.39	46.88	5726	0.6377	21.86
OpenSora-1.2 [55]	48.59	3032	0.7726	23.54	39.14	5733	0.6898	20.35
CogVideoX-2B [11]	42.01	3088	0.7796	22.20	36.76	5369	0.6540	18.05
EquiVDM-base	25.19	**1440**	**0.8424**	**32.69**	**26.59**	**3193**	**0.6925**	**25.65**

Table 2 | Video generation performance for models with both text prompt and control frames.

	Static Scenes				In-the-wild Scenes			
Method	**FID**	**FVD**	**CLIP**	**PSNR**	**FID**	**FVD**	**CLIP**	**PSNR**
CtrlVid canny [23]	44.00	1489	0.800	26.42	38.45	2724	0.7154	22.68
CtrlVid softedge [23]	61.90	1481	0.764	27.33	59.80	2694	0.7129	23.16
T2V-Zero canny [16]	40.10	2194	0.805	23.98	29.98	3350	0.7146	21.57
CtrlAdapter softedge [19]	47.29	1396	0.816	30.98	39.62	2789	0.7167	21.52
CtrlAdapter canny [19]	75.11	2698	0.768	25.33	36.24	2496	0.7214	23.09
EquiVDM-full softedge	**31.65**	1242	0.851	31.44	24.40	2122	0.7293	**26.86**
EquiVDM-full canny	33.46	**1142**	**0.853**	**34.52**	**22.24**	**1922**	**0.7551**	26.58

(EquiVDM-flow), where the input noise tensor is derived by warping the noise map of the initial frame using optical flow. The attached noise based variant of our method achieves similar performance to the flow-based version, although it is trained only with 2D videos without any 3D meshes.

5.4. Ablation Studies

Added noise amount We first evaluate our method with different amounts of added independent noise by adjusting the β value in Equation 4. A smaller β value indicates more noise added to the video hence less temporally consistent noise, and vice versa. In partic-

Table 3 | Sim2Real performance on ScanNet++ dataset.

Method	**FID**	**FVD**	**CLIP**	**PSNR**	**SSIM**
CtrlVid	82.63	1843	0.7384	27.08	0.8089
T2V-Zero	39.28	2078	0.7884	22.35	0.7110
CtrlAdapter	57.10	2098	0.7298	24.74	0.6881
EquiVDM-mesh	31.56	1266	**0.7917**	29.75	**0.9027**
EquiVDM-flow	**30.96**	**1244**	0.7915	**29.92**	0.8980

Table 4 | Ablations on β values controlling added noise.

	FID	**FVD**	**CLIP**	**PSNR**	**SSIM**
$\beta = 0.0$	39.92	2292	0.8126	20.81	0.6057
$\beta = 0.5$	26.66	1765	0.8509	30.77	0.9258
$\beta = 0.9$	**25.12**	**1585**	0.8575	**31.91**	**0.9343**
$\beta = 1.0$	50.03	1910	**0.9224**	28.67	0.9224

ular, for $\beta = 0.0$ the input noise is independent for each frame without any temporal consistency; while $\beta = 1.0$ indicates the input noise is fully determined by the first frame and the warping operation without any variations. We evaluate the performance on the test set of RealEstate10K dataset. As shown in Table 4, using temporally consistent noise helps in generating better videos in terms of quality, semantic alignment, and temporal consistency. On the other hand, without any added independent noise, the performance degrades since the model fails to model the high-frequency temporal variations of the corresponding pixels in the latent space; while the added independent noise expands the manifold of the input noise such that it covers the latent space better, as discussed in Section 4.3. We found that adding a small amount of independent noise with $\beta = 0.9$ achieves the best balance between quality and consistency.

Sampling steps Since the motion information about the video is already included in the warped noise, one natural question is whether the sampling steps can be reduced compared to the one using independent noise where both the motion and appearance have to be generated from scratch. To answer this question, we evaluate our method on ScanNet++ dataset with different numbers of sampling steps without changing the sampling schedule or performing the model distillation. As shown in Figure 5, using temporally consistent noise input, our method can generate videos with similar or better quality compared to the one using independent noise in much fewer sampling steps. In addition, the metrics saturate quickly, indicating that the appearance of the video can be generated from scratch with few sampling steps given the temporally consistent noise input. As shown in Figure 6, the detailed appearance-like reflection on the table surface can be generated in as few as 5 sampling steps. We believe these results open up new venues for video diffusion acceleration using warped noise.

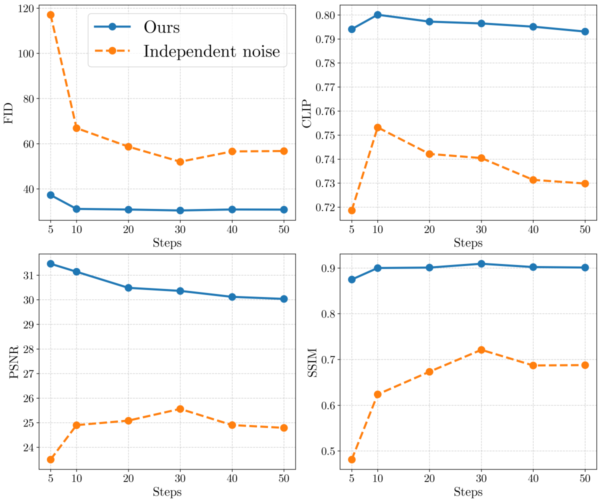

Figure 5 | Ablation on the number of sampling steps.

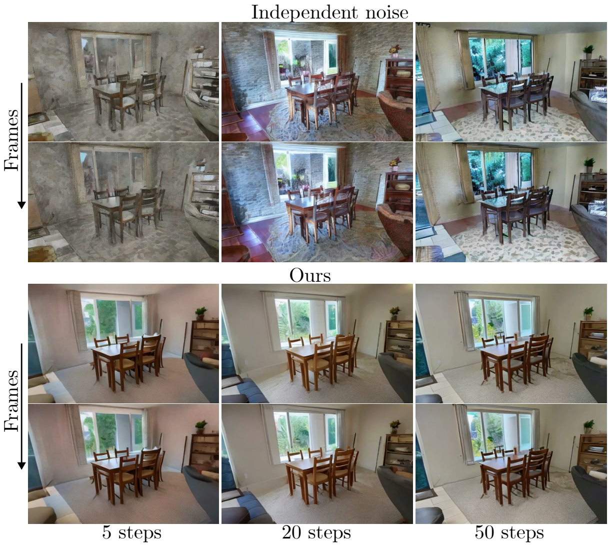

Figure 6 | Our method with temporally consistent warped noise generates videos with similar quality compared to the one using independent noise, but with much fewer sampling steps.

6. Conclusion

In this work, we propose using temporally consistent noise input for video generation with diffusion models. We show that video diffusion models can be trained to be equivariant to temporal transformations of the input by training with warped noise, without requiring additional regularization or modifications to the model architecture. We extend this approach to 3D by attaching noise to 3D mesh surfaces, enabling the generation of 3D-consistent frames from rendered noise maps. Through extensive experiments, we demonstrate that video diffusion models with consistent noise input generate more temporally consistent and higher-quality videos in significantly fewer sampling steps compared to those using independent noise input. One limitation of our method is that for long video generation, drifting is not fully alleviated by using consistent noise input. Possible solutions include utilizing auto-regressive video diffusion models along with warped noise. For 3D-grounded video generation, we can further enhance long-term cross-view consistency by attaching intermediate denoised features or images onto the 3D mesh in addition to the noise.

7. Impact Statement

This paper presents a generative model for video synthesis, contributing to advancements in generative learning. Our approach enhances the ability to generate high-quality, realistic videos, which has potential applications in content creation, simulation, and data augmentation. While this technology offers significant benefits, it also raises ethical considerations, particularly regarding the potential misuse of synthetic video generation for misinformation. We encourage responsible deployment with appropriate safeguards. Future work may explore bias mitigation and methods to ensure transparent and trustworthy generative video models.

References

[1] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.12015*, 2024.

[2] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI*, 2024.

[3] Abhishek Sharma, Adams Yu, Ali Razavi, Andeep Toor, Andrew Pierson, Ankush Gupta, Austin Waters, Aäron van den Oord, Daniel Tanis, Dumitru Erhan, Eric Lau, Eleni Shaw, Gabe Barth-Maron, Greg Shaw, Han Zhang, Henna Nandwani, Hernan Moraldo, Hyunjik Kim, Irina Blok, Jakob Bauer, Jeff Donahue, Junyoung Chung, Kory Mathewson, Kurtis David, Lasse Espeholt, Marc van Zee, Matt McGill, Medhini Narasimhan, Miaosen Wang, Mikołaj Bińkowski, Mohammad Babaeizadeh, Mohammad Taghi Saffar, Nando de Freitas, Nick Pezzotti, Pieter-Jan Kindermans, Poorva Rane, Rachel Hornung, Robert Riachi, Ruben Villegas, Rui Qian, Sander Dieleman, Serena Zhang, Serkan Cabi, Shixin Luo, Shlomi Fruchter, Signe Nørly, Srivatsan Srinivasan, Tobias Pfaff, Tom Hume, Vikas Verma, Weizhe Hua, William Zhu, Xinchen Yan, Xinyu Wang, Yelin Kim, Yuqing Du, and Yutian Chen. Veo, 2024.

[4] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, Gergely Klár, Grace Lam, Shiyi Lan, Laura Leal-Taixe, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum Reda, Xiaowei Ren, Vasanth Rao Naik Sabavat, Ed Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Varghese, Hao Wang, Haoxiang Wang, Heng Wang, Ting-Chun Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Lin Yen-Chen, Xiaohui Zeng, Yu Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[5] 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.

[6] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[7] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4172–4182, 2023.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[9] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through

stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[10] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022.

[11] 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. *CoRR*, 2024.

[12] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, and Sergey Tulyakov. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[13] Pascal Chang, Jingwei Tang, Markus Gross, and Vinicius C Azevedo. How i warped your noise: a temporally-correlated noise prior for diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024.

[14] Giannis Daras, Weili Nie, Karsten Kreis, Alex Dimakis, Morteza Mardani, Nikola Borislavov Kovachki, and Arash Vahdat. Warped Diffusion: Solving Video Inverse Problems with Image Diffusion Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[15] Yitong Deng, Winnie Lin, Lingxiao Li, Dmitriy Smirnov, Ryan Burgert, Ning Yu, Vincent Dedun, and Mohammad H Taghavi. Infinite-Resolution Integral Noise Warping for Diffusion Models. *arXiv*, 2024.

[16] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

[17] Tsai-Shien Chen, Chieh Hubert Lin, Hung-Yu Tseng, Tsung-Yi Lin, and Ming-Hsuan Yang. Motion-conditioned diffusion model for controllable video synthesis. *arXiv preprint arXiv:2304.14404*, 2023.

[18] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

[19] Han Lin, Jaemin Cho, Abhay Zala, and Mohit Bansal. Ctrl-adapter: An efficient and versatile framework for adapting diverse controls to any diffusion model. *arXiv preprint arXiv:2404.09967*, 2024.

[20] 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.

[21] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*, pages 331–348. Springer, 2024.

[22] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021.

[23] Weifeng Chen, Yatai Ji, Jie Wu, Hefeng Wu, Pan Xie, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models. *arXiv preprint arXiv:2305.13840*, 2023.

[24] Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Yuexian Zou, and Ying Shan. Image conductor: Precision control for interactive video synthesis. *arXiv preprint arXiv:2406.15339*, 2024.

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

[26] Weixi Feng, Chao Liu, Sifei Liu, William Yang Wang, Arash Vahdat, and Weili Nie. Blobgen-vid: Compositional text-to-video generation with blob video representations. *arXiv preprint arXiv:2501.07647*, 2025.

[27] Koichi Namekata, Sherwin Bahmani, Ziyi Wu, Yash Kant, Igor Gilitschenski, and David B. Lindell. Sg-i2v: Self-guided trajectory control in image-to-video generation. In *International Conference on Learning Representations (ICLR)*, 2025.

[28] Alan Wolfe, Nathan Morrical, Tomas Akenine-Möller, and Ravi Ramamoorthi. Scalar Spatiotemporal Blue Noise Masks. *arXiv*, 2021.

[29] Michael Kass and Davide Pesare. Coherent noise for non-photorealistic rendering. *ACM SIGGRAPH 2011 papers*, pages 1–6, 2011.

[30] M. Corsini, P. Cignoni, and R. Scopigno. Efficient and Flexible Sampling with Blue Noise Properties of Triangular Meshes. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):914–924, 2012.

[31] Fernando de Goes, Katherine Breeden, Victor Ostromoukhov, and Mathieu Desbrun. Blue noise through optimal transport. *ACM Transactions on Graphics (TOG)*, 31(6):1–11, 2012.

[32] Xingchang Huang, Corentin Salaun, Cristina Vasconcelos, Christian Theobalt, Cengiz Oztireli, and Gurprit Singh. Blue noise for diffusion models. *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers '24*, pages 1–11, 2024.

[33] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22930–22941, 2023.

[34] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10209–10218, 2023.

[35] Zhongwei Zhang, Fuchen Long, Yingwei Pan, Zhaofan Qiu, Ting Yao, Yang Cao, and Tao Mei. Trip: Temporal residual learning with image noise prior for image-to-video diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8671–8681, 2024.

[36] Kexin Lu, Yuxi CAI, Lan Li, Dafei Qin, and Guodong Li. Improve temporal consistency in diffusion models through noise correlations, 2024.

[37] Runjie Yan, Yinbo Chen, and Xiaolong Wang. Consistent flow distillation for text-to-3d generation. In *International Conference on Learning Representations (ICLR)*, 2025.

[38] Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, Michael Ryoo, Paul Debevec, and Ning Yu. Go-with-the-Flow: Motion-Controllable Video Diffusion Models Using Real-Time Warped Noise. *arXiv*, 2025.

[39] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[40] Karsten Kreis, Ruiqi Gao, and Arash Vahdat. Denoising diffusion-based generative modeling: Foundations and applications. CVPR 2022 Tutorial, 2022.

[41] Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. *arXiv preprint arXiv:2206.13397*, 2022.

[42] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph*, 37, 2018.

[43] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

[44] Zhiyu Tan, Xiaomeng Yang, Luozheng Qin, and Hao Li. Vidgen-1m: A large-scale dataset for text-to-video generation. *arXiv preprint arXiv:2408.02629*, 2024.

[45] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5187–5196. IEEE, 2019.

[46] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[47] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *IEEE International Conference on Computer Vision (ICCV)*, 2023.

[48] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[49] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

[50] 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 (ICML)*, pages 8748–8763. PMLR, 2021.

[51] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024.

[52] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[53] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv preprint arXiv:2309.15818*, 2023.

[54] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

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

[56] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5117–5127, 2021.

[57] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22139–22149, 2024.

Appendix

Appendix A: PSNR and SSIM used for evaluating temporal consistency

The PSNR and SSIM metrics in our paper are used for evaluting the temporal consistency of the generated frames, as well as how the motion pattern of the generated frames follows the optical flow of the input noise. As shown in Figure 7, to compute the PSNR and SSIM metrics, we first extract the 2D optical flow of the input driving video. Then given the correpsponding generated video, we warp the the source frame (the frame t in the case shown in the illustration) towards the target frame (the frame t+1) using the optical flow. Then we compute the PSNR and SSIM metrics between the warped source frame and the target frame. As a result, if the generated video follows the same motion pattern as the ground truth and maintains temporal consistency, it will yield a higher PSNR and SSIM scores—and vice versa.

Compared with the metrics in Video Benchmark [57], our metric is similar to the "Warping Error" for temporal consistency in the Sec.4.4 of that paper. The only difference is that the optical flow used for warping is estimated from the ground truth video rather than generated video.

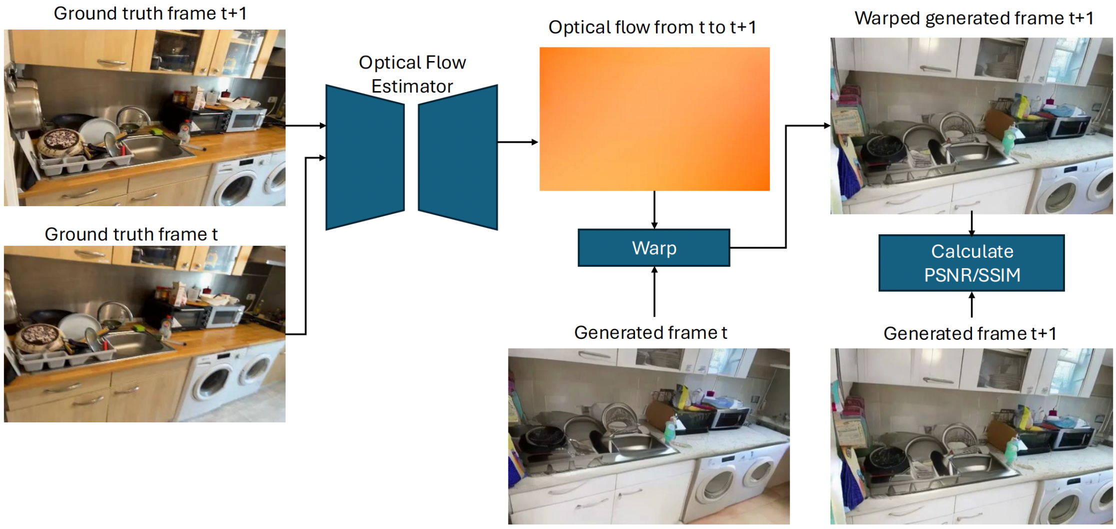

Figure 7 | Illustration of the PSNR and SSIM metrics used for evaluating temporal consistency.

Appendix B: EquiVDM for Diffusion Models with Transformers

For video diffusion models with transformer backbone [7, 11, 4], the latent space of the video where the diffusion and sampling process are performed is a set of video tokens from a video tokenizer. Unlike the VAEs in the UNet-based video diffusion models, the video tokenizer not only compress the spatial dimension of the video, but also the temporal dimension. For example, in CogVideoX [11] and CosMos [4], the tokenizer processes a video with N frames by first encoding the initial frame independently. It then encodes the subsequent $N-1$ frames into a sequence of $\lceil (N-1)/k \rceil$ temporal tokens, where k represents the temporal compression factor.

We build the warped noise frames accordingly to account for the temporal compression in the video tokenizer. For example, for the video tokenizer temporal compression scheme in CogVideoX [11] and CosMos [4], we first get the subsampled video by taking the first frame and every k-th frame from the following frames. Then we build the warped noise frames from the subsampled video. Another option is to build the warped noise frames directly from the original video, then subsample the warped noise frames accordingly. The first apporach is more efficient since it reduces the numbers of optical flow estimations. On the other hand, the second approach is more robust to videos with large motions. In our experiment, we use the second apprach for more robustness.

To add the control signal such as soft-edge maps, we use the same method as in the UNet-based video diffusion models: we add the adapter layers [19] between the frame encoder for the controlling frames and the transformer blocks in the video diffusion model. We interlace the adapter layers every 4 transformer blocks in the transformer backbone to avoid memory overflow. The qualitative results of the EquiVDM with the CogVideoX [11] model are shown in Figure 8–11.

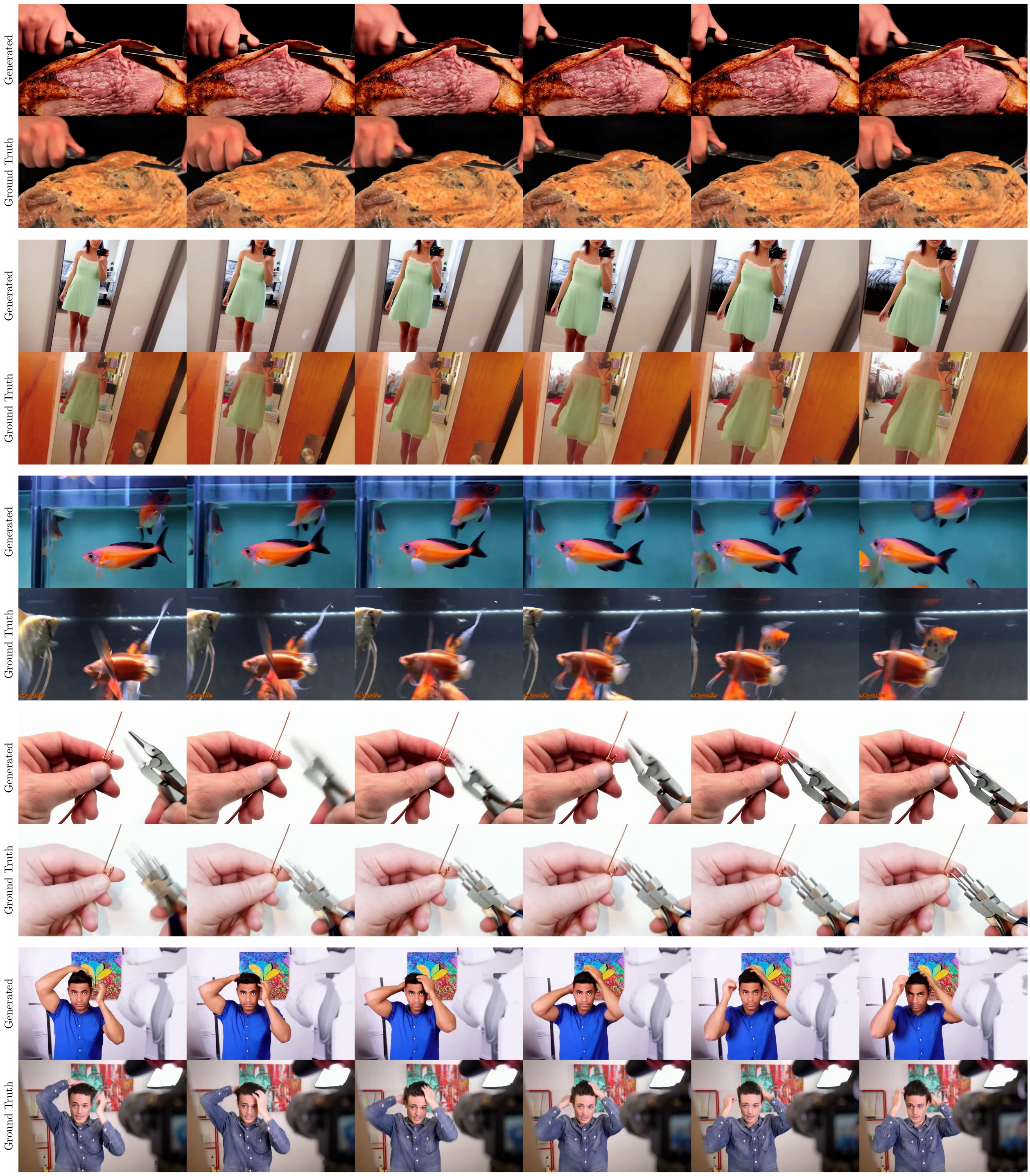

Figure 8 | The generated and driving videos of DiT-based video diffusion models.

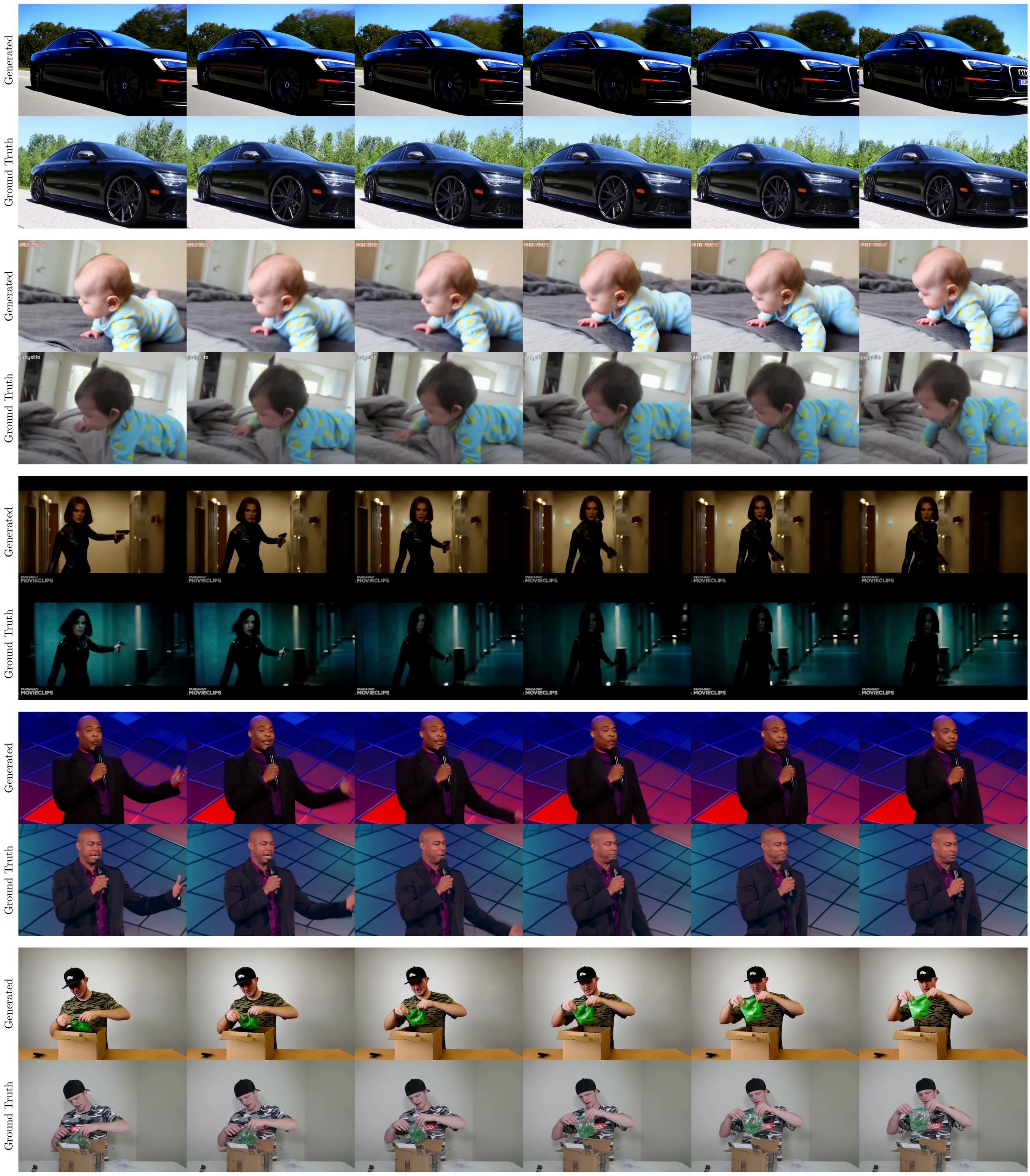

Figure 9 | The generated and driving videos of DiT-based video diffusion models.

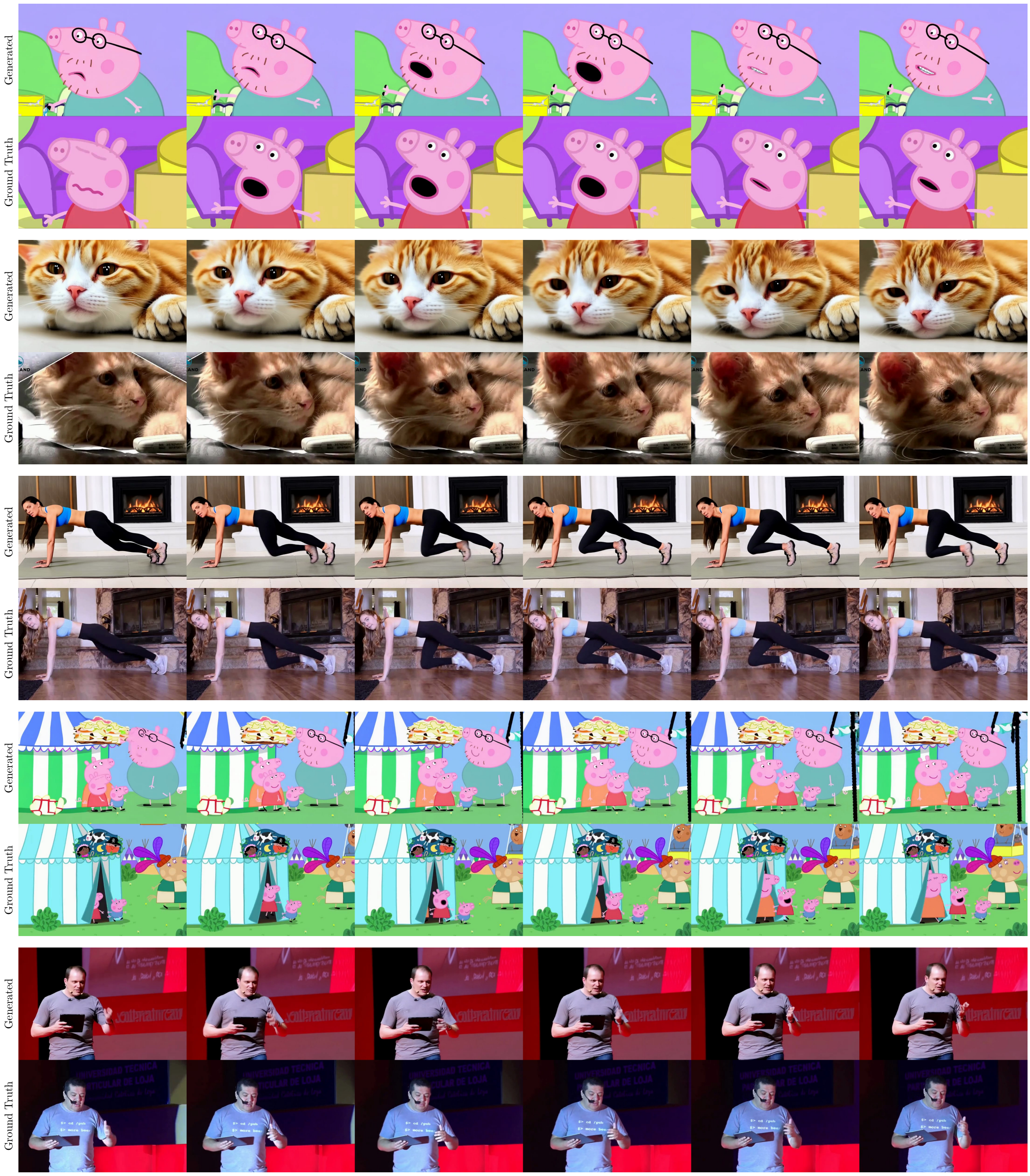

Figure 10 | The generated and driving videos of DiT-based video diffusion models.

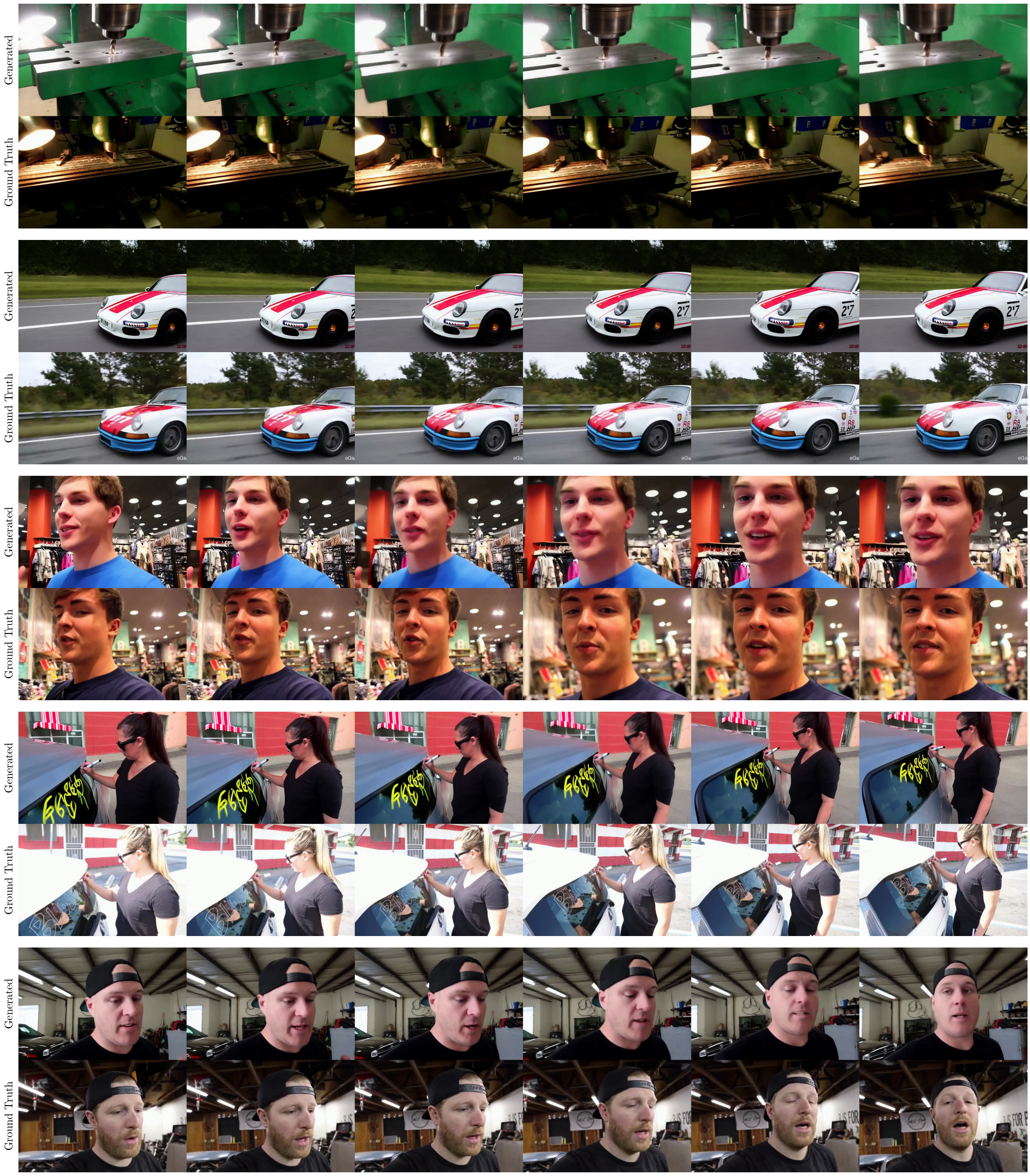

Figure 11 | The generated and driving videos of DiT-based video diffusion models.

Appendix C: Additional Results for Comparsions with other Methods

In Figure 12-17, we provide additional qualitative results for the comparison in Table 2 in Section 5.2.

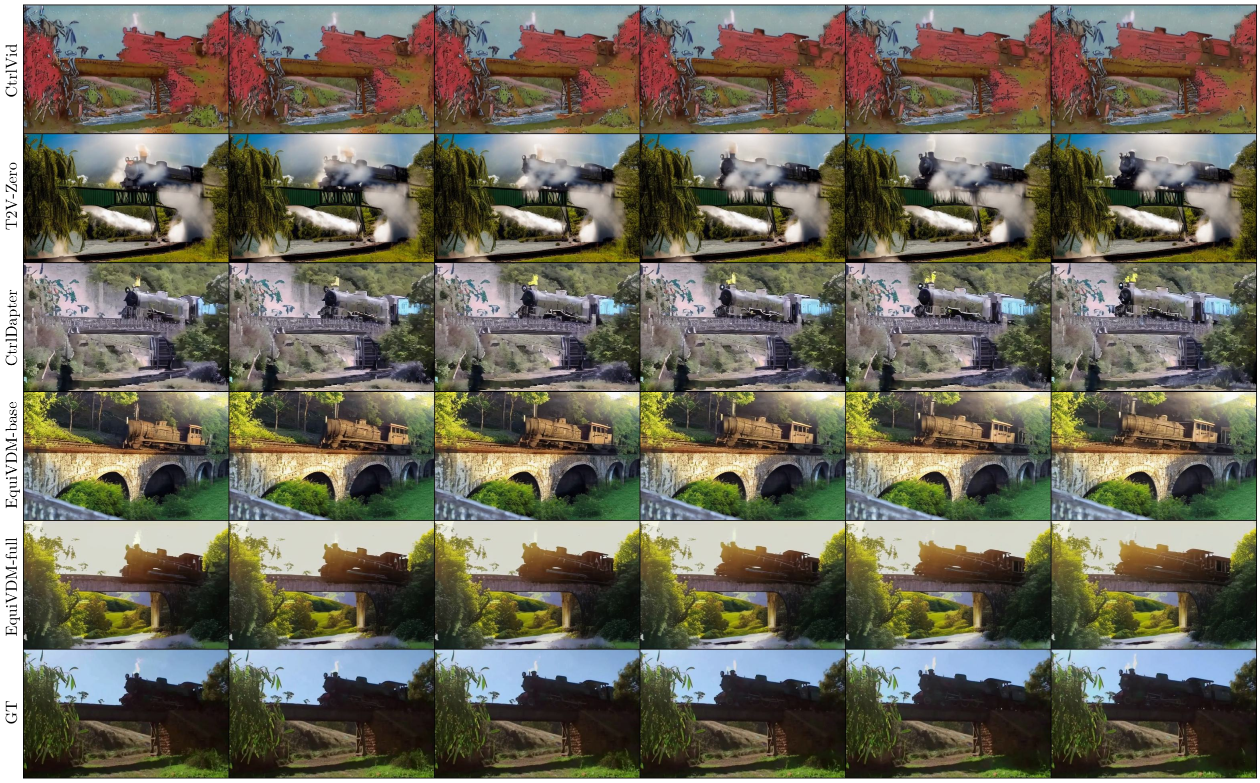

Figure 12 | Comparison of EquiVDM with other methods. CtrlVid [23], T2V-Zero [16], CtrlAdapter [19] and EquiVDM-full used soft-edge map as control signal for each frame along with the text prompt. EquiVDM-base only used text prompt.

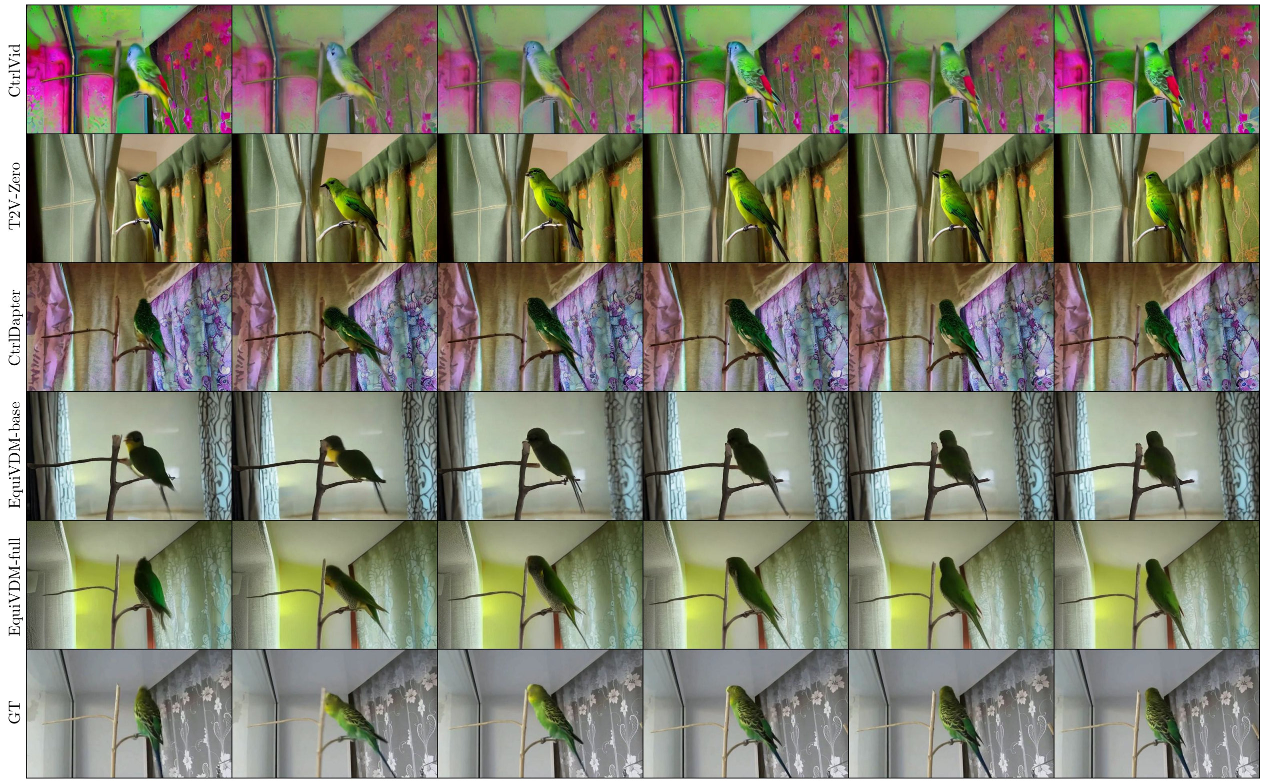

Figure 13 | Comparison of EquiVDM with other methods. CtrlVid [23], T2V-Zero [16], CtrlAdapter [19] and EquiVDM-full used soft-edge map as control signal for each frame along with the text prompt. EquiVDM-base only used text prompt.

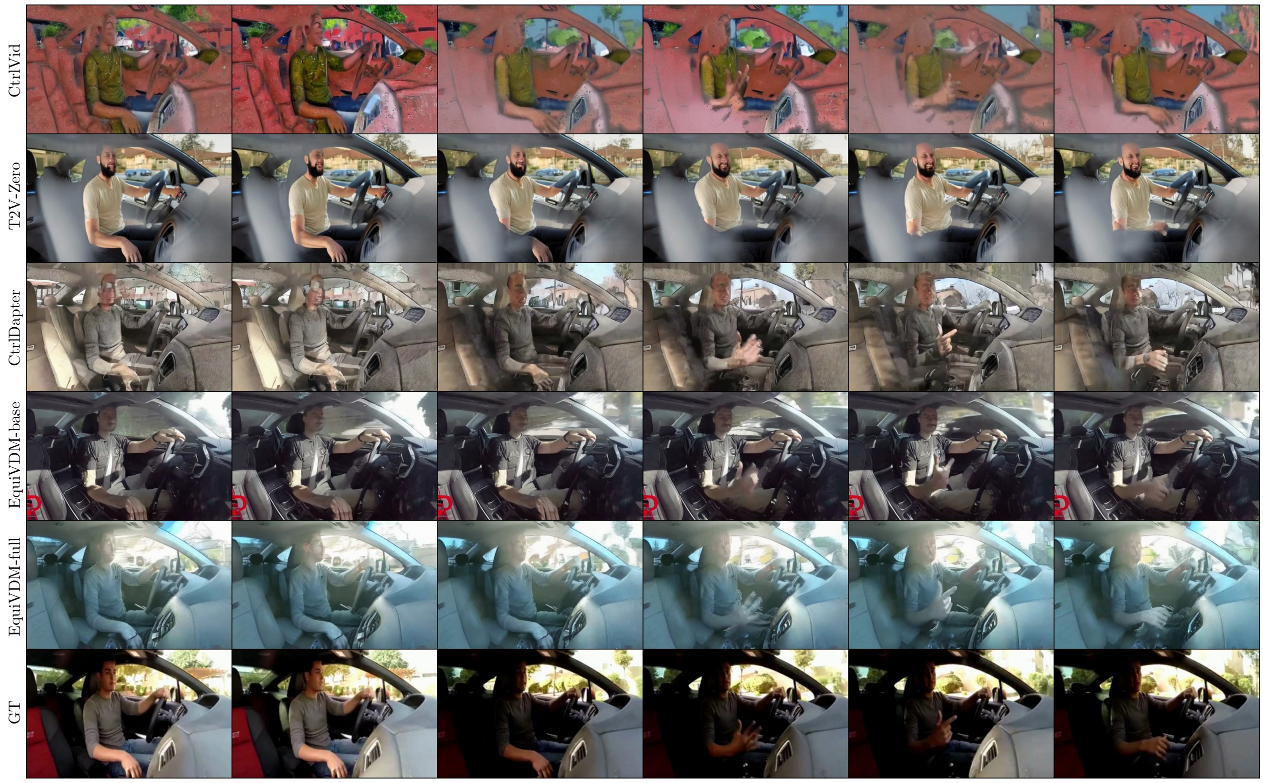

Figure 14 | Comparison of EquiVDM with other methods. CtrlVid [23], T2V-Zero [16], CtrlAdapter [19] and EquiVDM-full used soft-edge map as control signal for each frame along with the text prompt. EquiVDM-base only used text prompt.

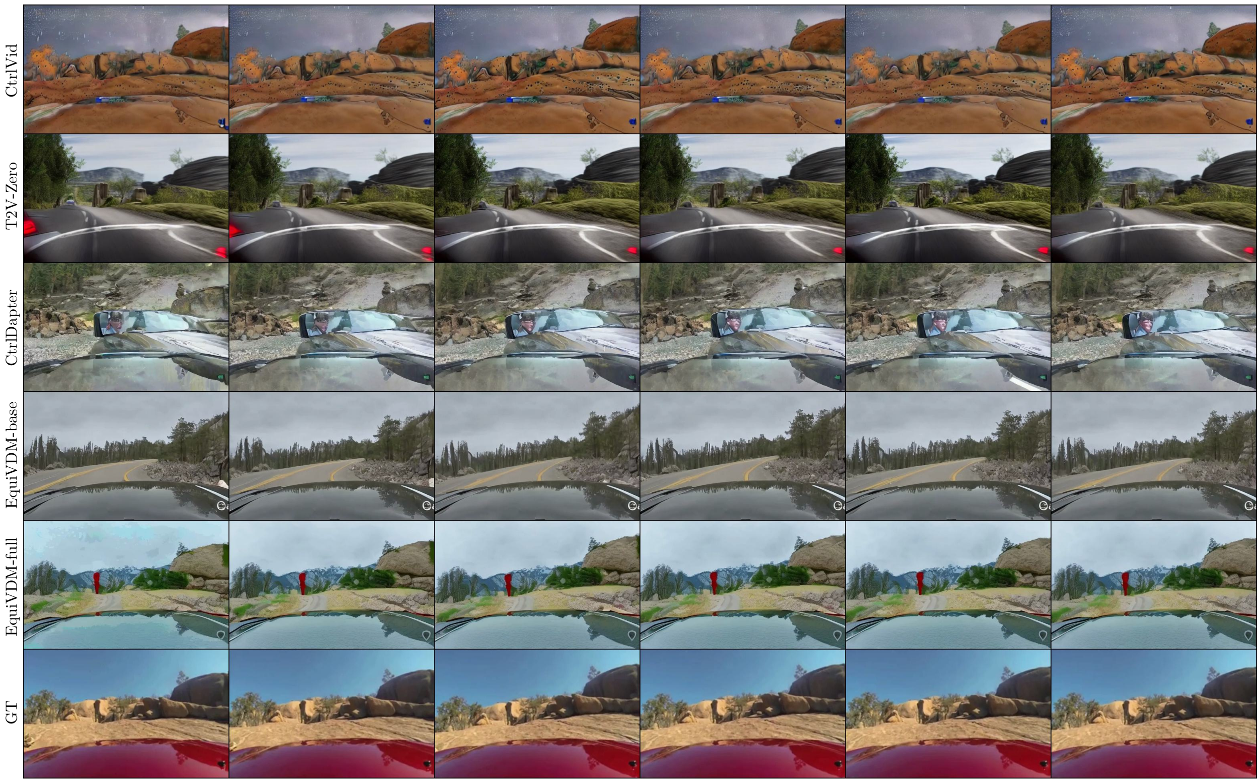

Figure 15 | Comparison of EquiVDM with other methods. CtrlVid [23], T2V-Zero [16], CtrlAdapter [19] and EquiVDM-full used soft-edge map as control signal for each frame along with the text prompt. EquiVDM-base only used text prompt.

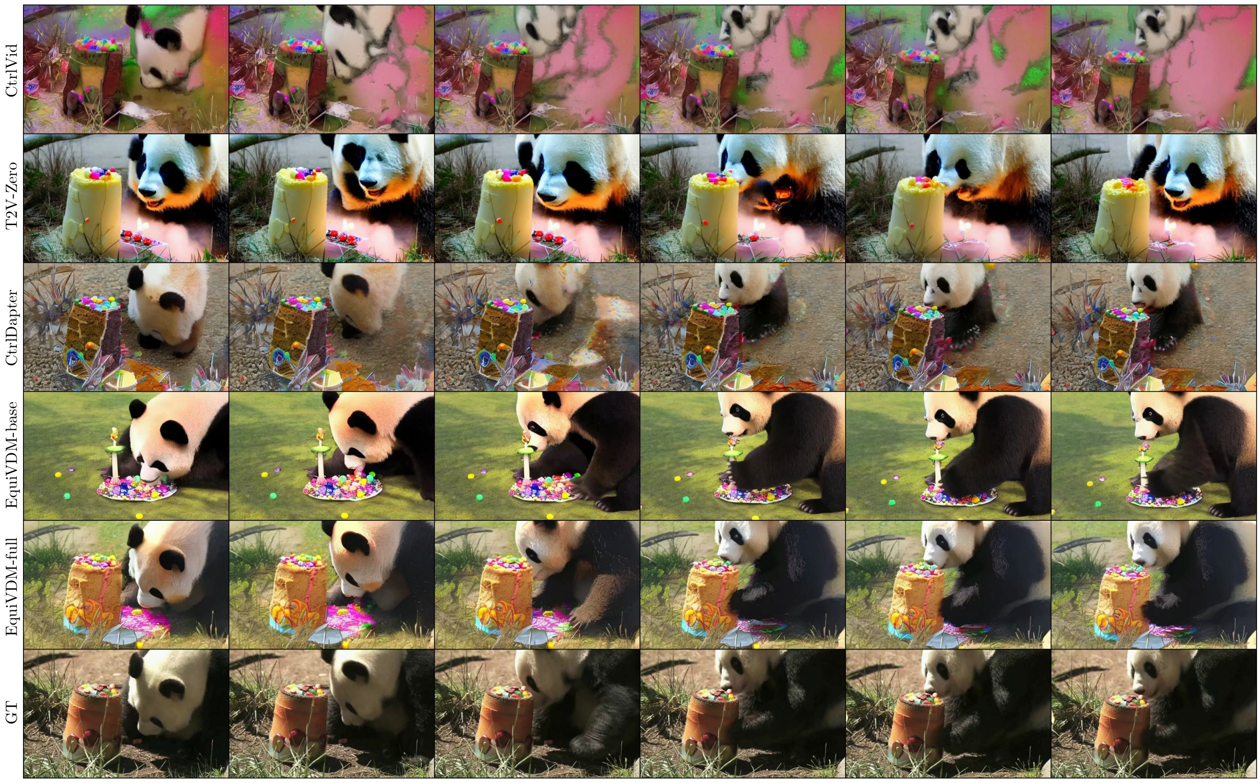

Figure 16 | Comparison of EquiVDM with other methods. CtrlVid [23], T2V-Zero [16], CtrlAdapter [19] and EquiVDM-full used soft-edge map as control signal for each frame along with the text prompt. EquiVDM-base only used text prompt.

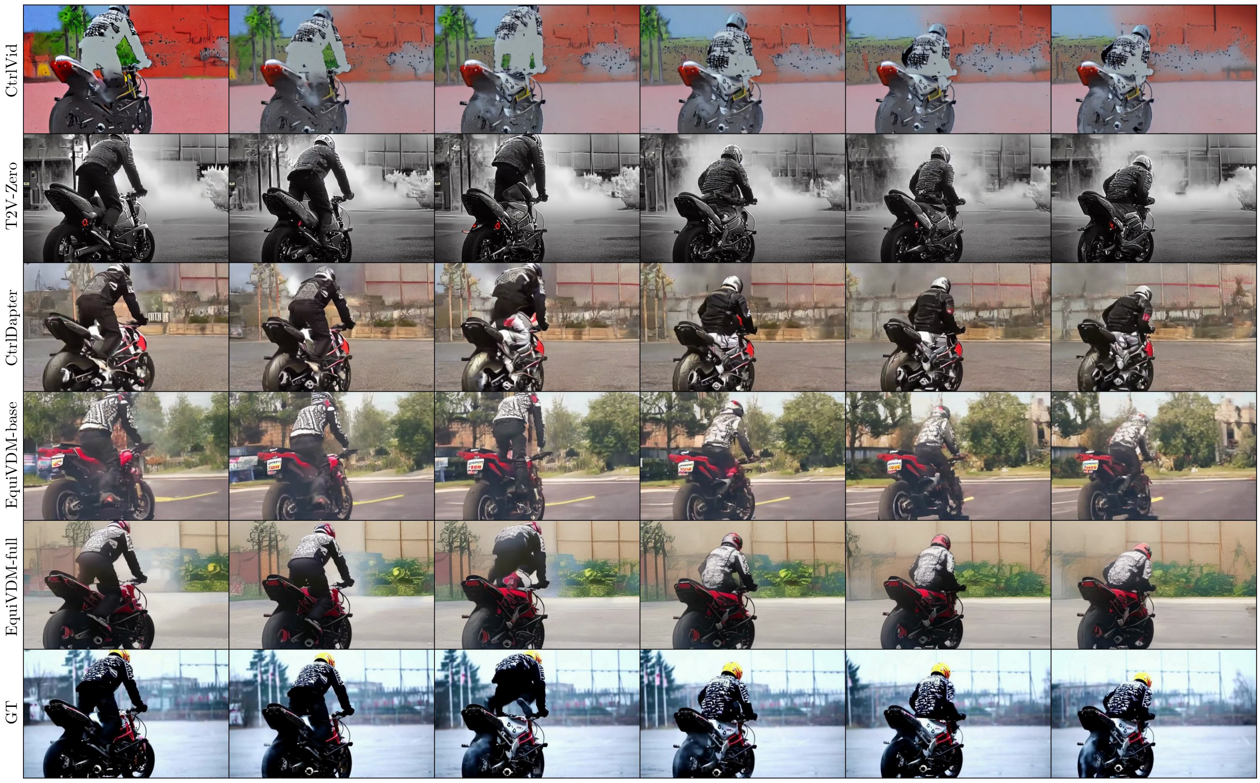

Figure 17 | Comparison of EquiVDM with other methods. CtrlVid [23], T2V-Zero [16], CtrlAdapter [19] and EquiVDM-full used soft-edge map as control signal for each frame along with the text prompt. EquiVDM-base only used text prompt.