Gaussian Variation Field Diffusion for High-fidelity Video-to-4D Synthesis
Supplementary Material

A. Additional Implementation Details

A.1. Model Architecture

We will detail the architecture of each model below, with the summary demonstrated in Table 1.

A.1.1. Gaussian Variation Field Encoder

Our encoder mainly comprises two parts: the canonical GS autoencoder $\mathcal{E}_{GS}$ and $\mathcal{D}_{GS}$ and a cross attention layer to create latent space for Gaussian Variation Fields.

For the canonical GS autoencoder, we adopt the model architecture from [13], which introduces a Structured Latent (SLAT) representation for static 3D assets. This representation defines a set of local latents on a 3D grid, where each latent is associated with an active voxel intersecting with the surface of the 3D asset. The SLAT representation effectively captures both the overall structure through active voxels and fine details through local latent codes. The canonical GS autoencoder is built using a transformer-based architecture. It first aggregates visual features from multiview images using a pre-trained DINOv2 [9] encoder to create voxelized features. These features are then processed through a sparse transformer encoder that handles variable-length tokens corresponding to active voxels. The transformer incorporates shifted window attention in 3D space to enhance local information interaction while maintaining computational efficiency. The encoder outputs structured latents that follow a regularized distribution through KL-divergence penalties, which are then decoded to various representations. For this work, we only leverage its Gaussian representation decoder for our canonical GS autoencoding. $\mathcal{D}_{GS}$ is set to resolution 64, and decode to 8 Gaussians per voxel. We finetune the decoder $\mathcal{D}_{GS}$ while keeping the encoder $\mathcal{E}_{GS}$ frozen.

For the cross attention layer, we adopt the vanilla full attention [12] implementation. we set the motion-aware $\Delta \boldsymbol{p}_t^{fps} \in \mathbb{R}^{512\times 3}$ using proposed ***mesh-guided interpolation*** mechanism as query and point displacement fields $\Delta P_t \in \mathbb{R}^{8192\times 3}$ from mesh as keys and values. Then the latent representation $\boldsymbol{z} \in \mathbb{R}^{512\times 16}$ is obtained after the cross attention layer.

A.1.2. Gaussian Variation Field Decoder

For the Gaussian Variation Field decoder, we first adopt 12 layers of vanilla self attention for thorough information exchange. For the last cross attention layer to decode Gaussian Variation Fields $\Delta G_t \in \mathbb{R}^{N_G\times 14}$ The output feature of last self attention layer is set to keys and values, and we adopt all parameters of $G_1 \in \mathbb{R}^{N_G\times 14}$ as query, where N_G is the total number of canonical GS.

A.1.3. Canonical GS Generation Model

We adopt the model architecture from [13] to generate structure latent representation for further decoding to canonical GS, which follows a two-stage process. First, a structure generator creates the sparse structure by denoising a low-resolution feature grid using transformer blocks with adaptive layer normalization and cross-attention for condition injection. Then, a latent generator $\boldsymbol{\mathcal{G}}_{\mathrm{L}}$ generates local latents for the given structure using a sparse transformer architecture with downsampling and upsampling blocks. These two generators both adopt RMSNorm [14] to the queries and keys (QK Norm.) in diffusion training. They are conditioned on image conditions through CLIP and DINOv2 features respectively, and are trained separately using a continuous flow matching objective. Since we freeze the $\mathcal{E}_{GS}$, we can directly leverage the pretrained image-to-3D model [13] to create canonical GS.

A.1.4. Gaussian Variation Field Diffusion Model

Our Gaussian Variation Field diffusion model builds upon the diffusion transformer architecture [10]. To enable temporal coherence in generation, we introduce a temporal self-attention layer that complements the existing cross-attention, spatial self-attention, and feedforward layers. For video sequence conditioning, we extract frame-wise features using DINOv2 [9] and incorporate the farthest-sampled canonical Gaussian Splatting to maintain awareness of the canonical 3D model. To enhance spatial consistency, we incorporate positional priors into the generation process. During training, we encode the Gaussian Variation Field latent along with their corresponding canonical GS positions to formulate positional embeddings. During inference, we directly utilize the positions from farthest-sampled Gaussian Splatting for positional embedding computation.

Table 1. **Detailed configuration of model architecture.** *SW* and *FFN* denotes "Shifted Window" and "FeedForward Net". *MSA*, *MSSA*, *MTSA*, *MCA* stand for "Multihead Self-Attention", "Multihead Spatial Self-Attention", "Multihead Temporal Self-Attention" and "Multihead Cross-Attention", respectively.

Network	#Layer	#Dim.	#Head	Block Arch.	Special Modules	#Param.
$\mathcal{E}_{GS}$	12	768	12	3D-SW-MSA + FFN	3D Swin Attn.	85.8M
$\mathcal{D}_{GS}$	12	768	12	3D-SW-MSA + FFN	3D Swin Attn.	85.1M
VAE Transformer	12	768	12	MSA / MCA + FFN	-	125.21M
Diffusion	12	512	16	MSSA + MTSA + MCA + FFN	QK Norm.	105.51M

A.2. Additional Training and Inference Details

In this paper, we designate the first frame of each video as the canonical frame. For our Direct 4DMesh-to-GS Variation Field VAE training, we set the loss weights as follows: $\lambda_{lpips} = 0.2$, $\lambda_{ssim} = 0.2$, $\lambda_{mg} = 1.0$, and $\lambda_{kl} = 1e-6$. Computationally, the VAE training requires one day on 32 Nvidia Tesla V100 GPUs (32GB) for the first stage and two days on 8 Nvidia Tesla A100 GPUs (40GB) for joint training, while the diffusion model training spans approximately one week on 8 Nvidia Tesla A100 GPUs (80GB). During inference, we adopt the adaptive mode of DPM-Solver [7] with order 2, requiring approximately 18 steps per instance.

During inference, we address potential orientation misalignment between the generated canonical GS and input images through an azimuth alignment process similar to [11]. Specifically, we render the canonical GS from multiple azimuth angles and compute image-level losses between these renders and the first video frame. We then transform the canonical GS according to the azimuth angle that yields the minimal loss, ensuring better alignment with the input video.

The in-the-wild conditional videos shown in the teaser (Figure 1 in main paper) are created by Kling [5]. The walking astronaut and boxing rat video frames in Figure 5 of the main paper are sourced from consistent4D and Emu video [3], respectively.

A.3. Additional Details of Creating Animation for Existing 3D Model

To animate existing 3D models using our approach, users follow a simple pipeline: First, their 3D assets are rendered as multiview images. These images are then processed to extract and aggregate DINOv2 features. Using these features, we construct a canonical Gaussian Splatting representation through our $\mathcal{E}_{GS}$ encoder and $\mathcal{D}_{GS}$ decoder. Finally, animations are generated by our diffusion model, which takes both the canonical GS and a conditional video as input. Users can create these conditional videos using state-of-the-art video diffusion models [3–5, 8] to specify their desired motion for the 3D model.

B. Data Preparation Details

Our training dataset consists of 34K 3D mesh animations sourced from Objaverse-V1 [2] and Objaverse-XL [1]. For Objaverse-V1, we utilize the curated set of 9K high-quality 3D animations from [6]. For Objaverse-XL, we apply two filtering criteria: first, following [13], we filter out samples whose average aesthetic score [1] across 4 rendered views of the first frame falls below 5.5; second, we remove sequences with minimal motion. This filtering process yields 25K additional animations from Objaverse-XL.

Table 2. Additional ablation of our proposed VAE.

Model	PSNR↑	LPIPS↓	SSIM↑
Ours w/o $\mathcal{D}_{GS}$ Finetuning	28.80	0.0460	0.962
Ours	**29.28**	**0.0439**	**0.964**

Table 3. Additional ablation of hyper-parameters in our mesh-guided interpolation.

K	β	PSNR↑	LPIPS↓	SSIM↑	K	β	PSNR↑	LPIPS↓	SSIM↑
16	7.0	28.38	0.0464	0.960	8	10.0	28.55	0.0462	0.961
8	**7.0**	**29.28**	**0.0439**	**0.964**	**8**	**7.0**	**29.28**	**0.0439**	**0.964**
4	7.0	28.94	0.0451	0.963	8	4.0	29.04	0.0446	0.963
1	7.0	28.22	0.0465	0.960	8	1.0	28.64	0.0457	0.962

C. Additional Ablation

Ablation of $\mathcal{D}_{GS}$ Joint Finetuning. We investigate the importance of jointly finetuning the canonical GS decoder during our Direct 4DMesh-to-GS Variation Field VAE training. Starting from a pretrained canonical 3D $\mathcal{D}_{GS}$ checkpoint, we compare two settings: freezing $\mathcal{D}_{GS}$ while training other modules, and jointly training all modules (our approach). As shown in Table 2, joint training allows $\mathcal{D}_{GS}$ to receive feedback from animation reconstruction rather than being limited to static data only. This ensures the canonical GS reconstruction coherent with its corresponding variation fields.

Ablation of hyper-parameters in mesh-guided interpolation. We ablate the hyper-parameters including nearest neighbors K, and distance decay rate β of interpolation in Table 3. Our setting ($K = 8, \beta = 7.0$) yields optimal results. Performance is relatively stable for other values, showing reasonable robustness.

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

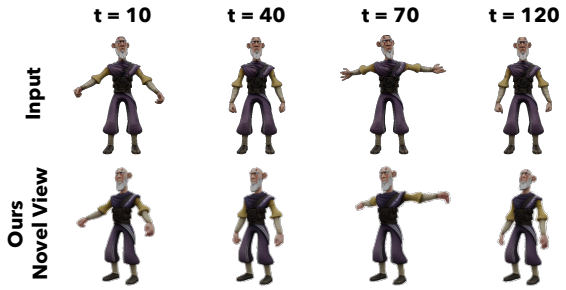

Figure 1. Sample of our autoregressive generation result.

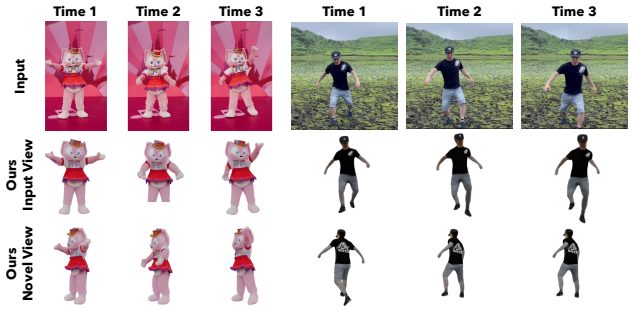

Figure 2. More generation results of real-world input videos.

D. More Results

D.1. Autoregressive Generation Results for Temporal Generalization

Temporal generalization is a known challenge in 2D/3D video generation. In our case, we can employ an autoregressive approach during inference for videos exceeding our training length: the GS from the last frame of a generated segment serves as the canonical GS for inferring the next segment's variation fields, which allows for coherent long animations. We show a 120-frame generated sample using such an approach in Figure 1.

D.2. VAE Reconstruction Results

As illustrated in Figure 4, we demonstrate the reconstruction capabilities of our proposed Direct 4DMesh-to-GS Variation Field VAE. Our method efficiently encodes both canonical GS and their temporal variations from 4D meshes in a single pass, eliminating the need for time-consuming per-instance fitting procedures. The results demonstrate our model's effectiveness in preserving both geometric fidelity and motion dynamics.

D.3. More Visual Comparison with SOTA Methods

As illustrated in Figure 5, we provide extensive visual comparisons with state-of-the-art methods. Our approach demonstrates consistent superiority across diverse test cases, achieving better results in terms of both visual fidelity and temporal motion coherence.

D.4. More Reults of Animating Existing 3D Models

As shown in Figure 6, we demonstrate additional results showcasing our method's capability to animate existing 3D models using conditional videos. Our approach successfully extracts and transfers motion patterns from the input videos, generating high-fidelity animations that faithfully preserve both geometric and temporal characteristics.

D.5. Additional Results on Real-World Video Inputs

Although our model is trained on synthetic data, it effectively generalizes to real-world video inputs. Figure 2 presents additional results, demonstrating the model's robust generalization capabilities.

D.6. Additional Video Results

In the supplementary material, we showcase additional videos demonstrating both our high-quality results and comparative analyses with baseline approaches.

E. Borader Impact

Like all generative models, our video-to-4D generation framework requires careful consideration of societal implications. While we mitigate certain ethical concerns by training exclusively on synthetic 3D animations from the Objaverse dataset, thus avoiding privacy and copyright issues associated with real-world data, we acknowledge potential risks. The ability to generate animated 3D content from videos could be misused for creating misleading content. We therefore emphasize the importance of establishing clear guidelines for the responsible deployment of video-to-4D generation technology.

F. Limitation Discussion and Future Work

While our model demonstrates impressive results in video-to-4D generation, it has certain limitations. Our two-stage generation process first employs a pretrained static 3D generative model to create canonical Gaussian Splatting representations, which then serve as conditions for our diffusion model to generate Gaussian Variation Fields. A notable limitation arises when the static 3D generative model [13] produces canonical GS that exhibits discrepancies with the conditional video, such as mismatched head pose, incorrect eyes or light effects in Figure 3, potentially creating inconsistencies in the final animation. To address this limitation, future work could explore either fine-tuning the static 3D model to ensure better image alignment or developing an end-to-end 4D diffusion framework that jointly generates both the canonical representation and its temporal variations.

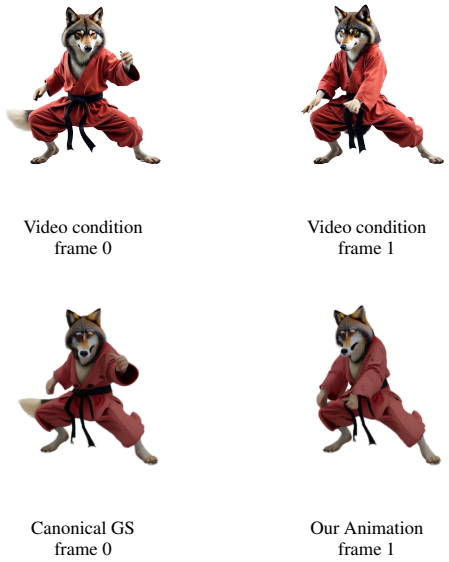

Figure 3. **Failure case.** When the pretrained static 3D generative model produces canonical GS that are not well-aligned with the conditional video frames, our Gaussian Variation Field diffusion model struggles to bridge this inconsistency, resulting in suboptimal animations.

References

[1] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023. 2

[2] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 2

[3] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Factorizing text-to-video generation by explicit image conditioning. In *European Conference on Computer Vision*, pages 205–224. Springer, 2024. 2

[4] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. Hunyuanvideo: A systematic framework for large video generative models, 2025.

[5] Kuaishou. Kling. *https://klingai.kuaishou.com*, 2024. 2

[6] Hanwen Liang, Yuyang Yin, Dejia Xu, Hanxue Liang, Zhangyang Wang, Konstantinos N Plataniotis, Yao Zhao, and Yunchao Wei. Diffusion4d: Fast spatial-temporal consistent 4d generation via video diffusion models. *arXiv preprint arXiv:2405.16645*, 2024. 2

[7] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 2

[8] OpenAI. Video generation models as world simulators. *https://openai.com/index/video-generation-models-as-world-simulators*, 2024. 2

[9] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1

[10] 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. 1

[11] Jiawei Ren, Cheng Xie, Ashkan Mirzaei, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, Huan Ling, et al. L4gm: Large 4d gaussian reconstruction model. *Advances in Neural Information Processing Systems*, 37:56828–56858, 2025. 2

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 1

[13] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024. 1, 2, 3

[14] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019. 1

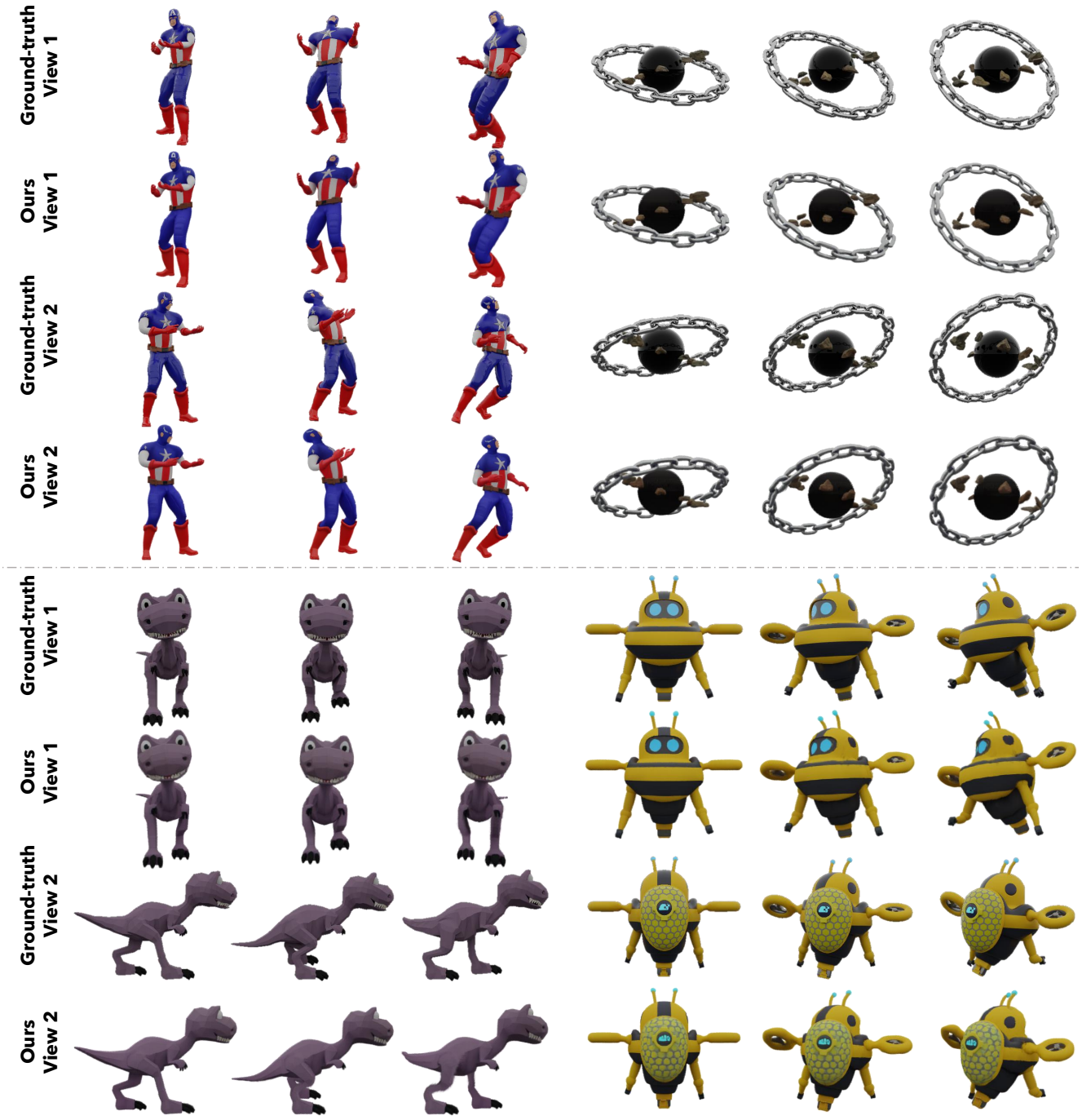

Figure 4. Additional visual results of VAE reconstruction.

Figure 5. More visual comparison with SOTA Methods.

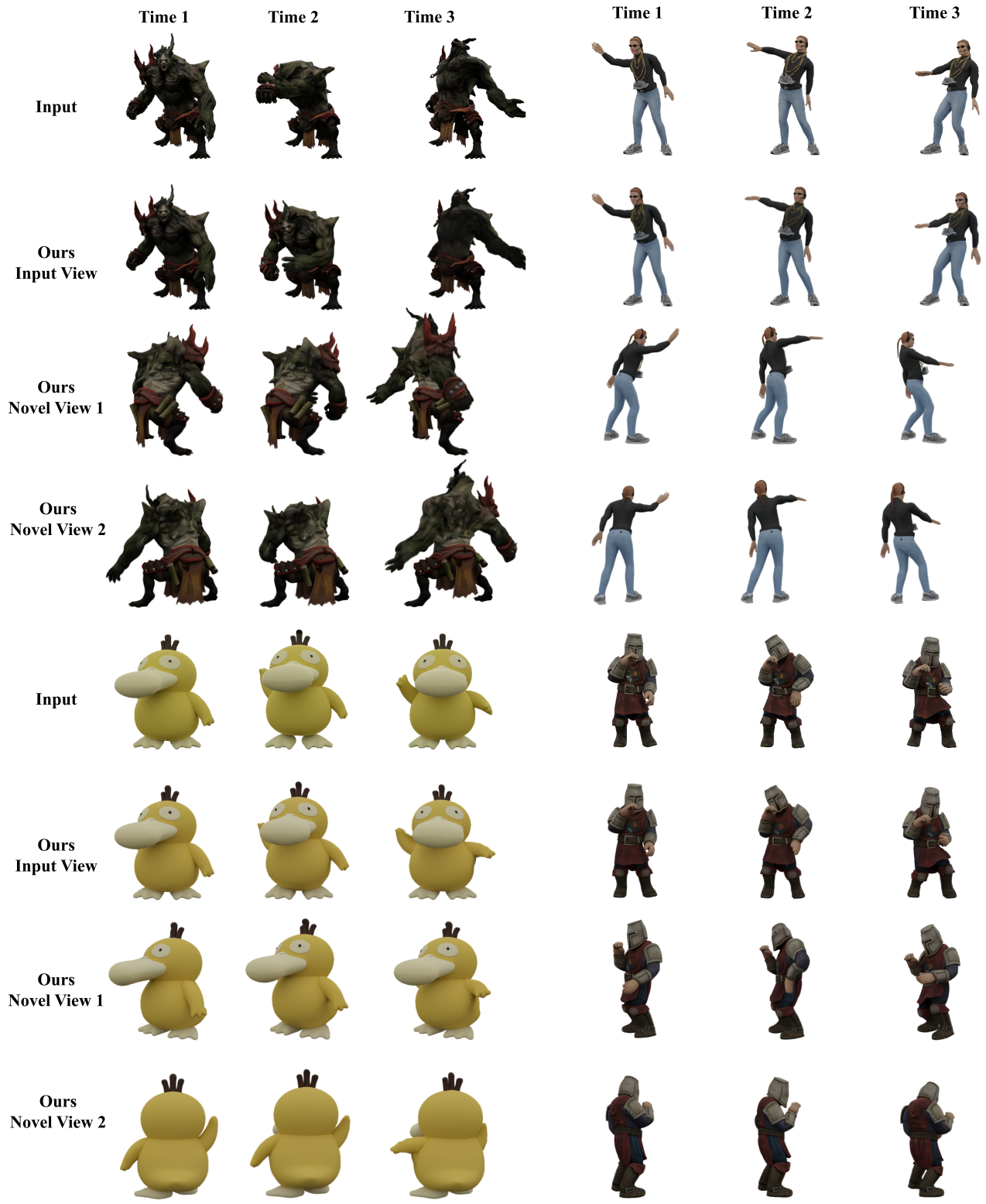

Figure 6. More results of animating existing 3D model input with conditional videos.