# Appendix

In this appendix, we first present additional related works on 3D refinement (Appendix A). Then we provide detailed network specifications (Appendix B). Next, To ensure reproducibility and facilitate fair perceptual studies, we describe the experimental settings in detail (Appendix C). Finally, we include extended ablation studies (Appendix D) and additional results (Appendix E) to demonstrate the robustness and superiority of our methods across various settings.

## A More Related Works

**3D refinement with generative priors.** To deal with view-inconsistency and low quality problems, many works (Wu et al., 2024c; Roessle et al., 2023; Chen et al., 2024; Wu et al., 2024b; Haque et al., 2023; Vachha & Haque, 2024; Zhou & Tulsiani, 2023) take advantages from generative priors, *e.g.*, adversarial training (Goodfellow et al., 2014) and score distillation sampling (SDS) (Poole et al., 2023) to optimize the 3D representation. GANeRF (Roessle et al., 2023) refines the rendered images with an image-conditional generator and leverages the re-rendered image constraints to guide the NeRF optimization in the adversarial formulation. InstructNeRF2NeRF (Haque et al., 2023) uses the text-conditioned image generator, InstructPix2pix (Brooks et al., 2023), to edit the image rendered by pre-trained NeRF in an iterative manner and updates the underlying 3D representation with the edited images. ReconFusion (Wu et al., 2024b) uses the diffusion priors, Zero-123 (Liu et al., 2023b), as a drop-in regularizer to enhance the 3D reconstruction performance, especially for sparse-view scenarios. In contrast to directly optimizing the implicit representation, another line of researches (Tang et al., 2024b; Ren et al., 2023) first extracts the explicit textured mesh, and then refine the texture in UV-space with diffusion prior and differentiable rendering. In particular, DreamGaussian4D leverages SVD as image-to-video prior to enhance the texture temporal consistency. In our paper, in consideration of the artifacts generated in video diffusion, we extend the refinement techniques to the 4D representation.

## B Network Details

In this section, we unpack the core network design in Figure 1.

**Attention injection.** In Sec. 4.1, we exploit the attention injection strategy to alleviate the temporal difference between multi-view diffusion models. Figure 9 illustrates its network details: in each spatial attention layer, we replace the self-attention by simultaneously considering the current $z_t^*$ and previous visual information with EMA.

**Deformation field with color transformation.** In Sec. 4.2, we use color affine transformation to model the temporal texture variation. Figure 10 shows the detailed architecture of that. We first query the time-specific feature $f_t$ from the learnable HexPlane (Cao & Johnson, 2023) with the canonical Gaussian positions $\bar{\mu}$. After that, the geometric deformations of Gaussian properties ($\mu$ location, $r$ rotation, and $s$ scale) are predicted with a lightweight decoder. Additionally, we use the affine color transformation to model the temporal texture variations. Finally, these deformed Gaussians are rendered into an image.

## C Additional Experimental Settings

### C.1 Optimization Details

We report the optimization of 4D Gaussian splatting for the purpose of reproduction. Basically, we follow the training recipe from 4DGS (Wu et al., 2024a) in the coarse 4D reconstruction stage. In the semantic refinement stage (Stage III), we fine-tune 4DGS for 5k steps with Adam optimizer. The initial learning rate is set to 1e-4 with exponential decay. The weight $\lambda$ in diffusion refinement loss is set to 0.5. Our implementation is primarily based on the PyTorch framework and tested on a single NVIDIA RTX 3090 GPU.

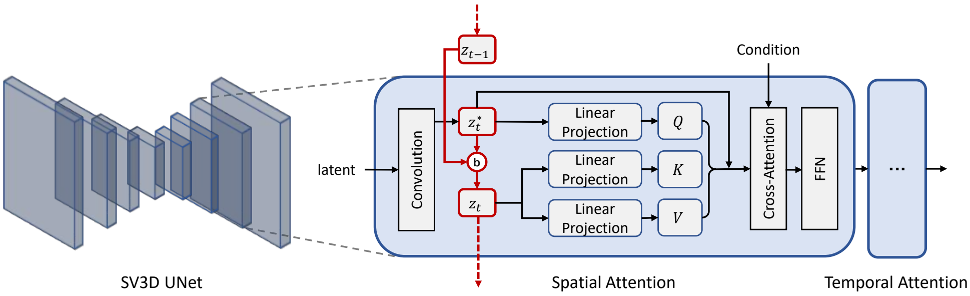


Figure 9: **Network details of Attention Injection.** ⓑ denotes the EMA blending operator mentioned in Sec. 4.1. $z_t^*$ is the multi-view latent at current timestamp $t$, and $z_t$ is the blended latent. Previous visual information is injected into the current latent by modifying the original spatial self-attention mechanism.

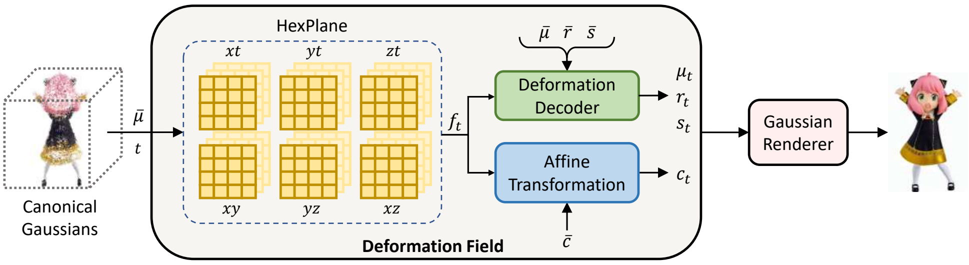


Figure 10: **Network architecture of our 4D representation.** $\bar{\mu}, \bar{r}, \bar{s}$ represents the canonical Gaussian properties: 3D location, rotation and scale from the coarse stage training in 4DGS (Wu et al., 2024a). The time-specific local feature $f_t$ is queried from the HexPlane (Cao & Johnson, 2023), where the subscribe $t$ means the time-specific property. Different from vanilla 4DGS, we employ additional color affine transformation to obtain the time-specific color $c_t$. The geometric deformations are predicted by a lightweight decoder. Finally, the time-specific Gaussians are rendered to produce an image (right).

### C.2 Reproduction, Data and Code

We reproduced our baselines, including Animate124 (Zhao et al., 2023), DreamGaussian4D (Ren et al., 2023), and Consistent4D (Jiang et al., 2024b), Efficient4D (Pan et al., 2024), using their official code. Additionally, we have included the input images and SVD-generated videos in the *supplementary materials*. Apart from the data provided by Animate124 and DreamGaussian4D, we have added three more examples: `android`, `chicken-basketball`, and `penguin`. Code is also available in the *supplementary materials*.

### C.3 User Study Details

We provide details of the user preference study with two screenshots. Figure 21 illustrates the guidelines: each participant is asked to evaluate images and videos rendered by four different methods across five metrics. Figure 22 shows the image and video samples presented to the participants. After comparing the images (Figure 22(a)) rendered by different models, participants select the method with the highest ”reference image consistency” and ”3D appearance”. After watching the videos (Figure 22(b)) rendered by different models, participants select the method with the highest ”motion realism” and ”motion range”. Finally, they choose the method with the best overall quality. We presented several cases to 47 participants and compiled the statistics. For statistical significance, we make the assumption of multinomial distribution, and report the 2-sigma error bar (95.6% CI). We use standard deviation for error bar calculation.

| Method | Independent | S-Res | S-Linear | T-EMA | S-EMA (Ours) |
|---|---|---|---|---|---|
| CLIP-I ↑ (Radford et al., 2021) | 0.9323 | 0.9136 | 0.9654 | **0.9962** | 0.9925 |
| Flow Intensity↑ (Teed & Deng, 2020) | - | - | **2.912** | 1.102 | 2.756 |

Table 4: **Quantitative ablation on attention injection.** We evaluate temporal consistency using CLIP-I score between the first and subsequent frames (↑ higher is better), and motion range using optical Flow Intensity (↑ larger indicates larger motion range when CLIP scores are comparable). '-' means no reasonable results predicted by the optical flow estimator on video with noisy background. Both 'T-EMA' and 'S-EMA' improve temporal consistency, but while 'T-EMA' results in nearly static output, 'S-EMA' maintains substantial motion range. Qualitative results are shown in Figure 5.

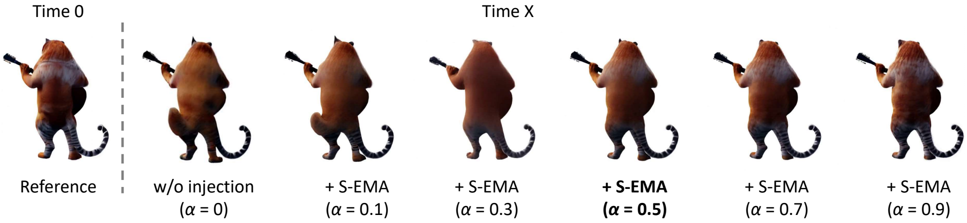


Figure 11: **Qualitative sensitivity analysis on EMA blending weight of attention injection**. As the blending weight increases, the temporal consistency is significantly improved (similar white textures and consistent leg geometry). However, the overly high ($> 0.5$) blending weight leads to a very small motion range. To balance the motion range and temporal consistency, we choose the EMA weight as $\alpha = 0.5$. Video demonstration can be found in our *supplementary materials*.

## D Extended Ablations

**Quantitative ablation on attention injection.** We conduct comprehensive experiments to evaluate different variants of attention injection. Please refer to Figure 5 in the main paper for qualitative results. Quantitative analysis is summarized in Table 4, demonstrating the impact of different attention injection variants on video temporal consistency and motion range. To quantify temporal consistency, we compute the CLIP-I score between the first frame and subsequent frames. Our results indicate that both 'T-EMA' and 'S-EMA' significantly improve temporal consistency (inter-frame similarity), achieving higher CLIP-I scores compared to other variants. For motion range assessment, we employ the flow intensity, calculated by average value of optical flow on adjacent frames. When CLIP scores are comparable, larger flow intensity indicates a larger range of motion. The optical flow is estimated with RAFT (Teed & Deng, 2020). '-' means no reasonable results predicted by the optical flow estimator on video with noisy background. Notably, 'T-EMA' yields a DINO score approaching 1, suggesting minimal object movement. Among all variants examined, our proposed 'S-EMA' uniquely achieves an optimal balance, maintaining high temporal consistency while preserving substantial motion range.

**Sensitivity analysis for attention injection weight.** Figure 11 analyzes different EMA blending weights $\alpha$ of attention injection in the spatial attention layers. It is obvious that the increasing blending weight benefits the temporal consistency in texture, *e.g.*, similar white texture in the back and consistent leg geometry. We also observe that overly high ($> 0.5$) blending weight significantly attenuates the object motion range. This trade-off can be better illustrated by the videos provided in the *supplementary materials*. Taking both motion range and temporal consistency into consideration, we choose $\alpha = 0.5$ as an appropriate blending weight without sacrificing the dynamics. Figure 12 demonstrates the impact of different blending weights. As the weight increases, CLIP-I score (image quality) improves while motion range becomes smaller, indicated by decreasing flow intensity. CD-FVD will not be better due to the diminishing motion.

**Number of Gaussians.** In Figure 13, we show the number of Gaussians before and after adding the multiscale renderer. Guo et al. (Guo et al., 2024b) observed that visual overfitting often leads to redundant Gaussian splats in dynamic scene reconstruction, which is hard to optimize and causes

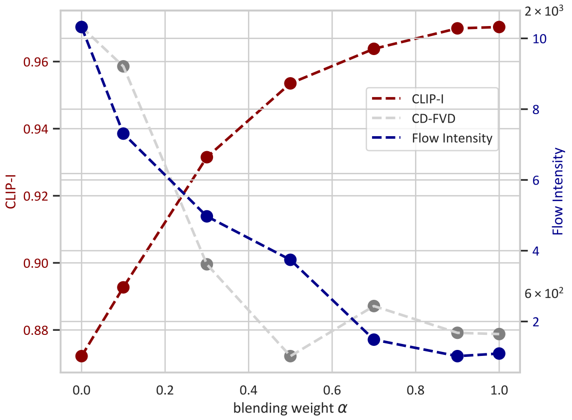


Figure 12: **Quantitative sensitivity analysis on EMA blending weight of attention injection.** Higher blending weights improve temporal consistency but excessive values restrict motion range indicated by lower flow intensity. We select $\alpha$=0.5, achieving well balance between motion range and temporal consistency, with the best CD-FVD score.

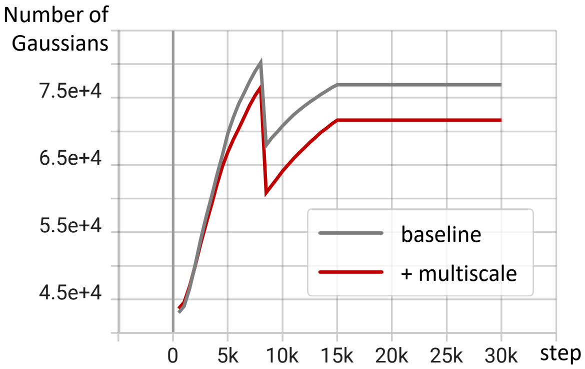


Figure 13: **Additional ablation on the multiscale renderer.** With the multiscale rendering augmentation in Stage II (darkred), the number of Gaussians declines significantly.

unsatisfying rendering results. With the multiscale renderer, we observe a significant decline of Gaussian points, in addition to the dropped training PSNR reported in Figure 7.

**Additional results for diffusion refinements.** In Figure 14, the effectiveness of our diffusion refinement is illustrated with zoomed-in details. It can be observed that the facial and hand details become finer and Gaussian noises are removed after the refinement stage.

## E Extended Results

**Dynamics of our results.** For the best demonstration of our 4D model dynamics, please refer to the *supplementary materials* where you can find videos generated by our 4D model. Figure 15 has illustrated more examples beyond SVD-generated videos, and show the scalability and generalizability of our framework. The panel (a) uses video rendered from Objaverse (Deitke et al., 2023) dataset, a large-scale 3D dataset that also contains some animation models. Figure 15 (b) shows the 4D generation results from in-the-wild videos from the Consistent4D benchmark; In panel (c), we leverage the pose-conditioned character video generation model, AnimateAnyone (Hu, 2024), as our video model in our framework.

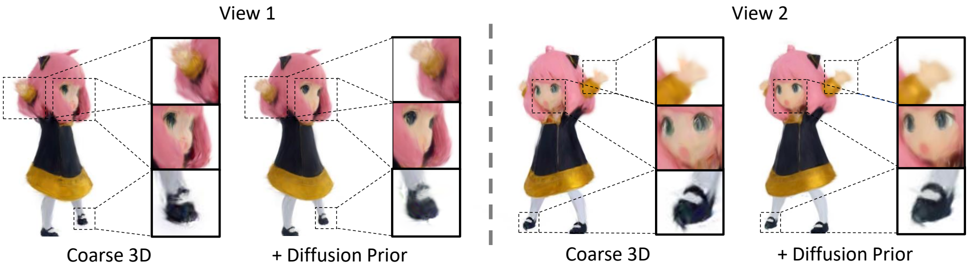


Figure 14: **Ablation on diffusion refinement.** The left and right panels depict two different view of renderings with the case `anya`. The results after adding the diffusion refinement show finer facial and hand details with less noisy Gaussians.

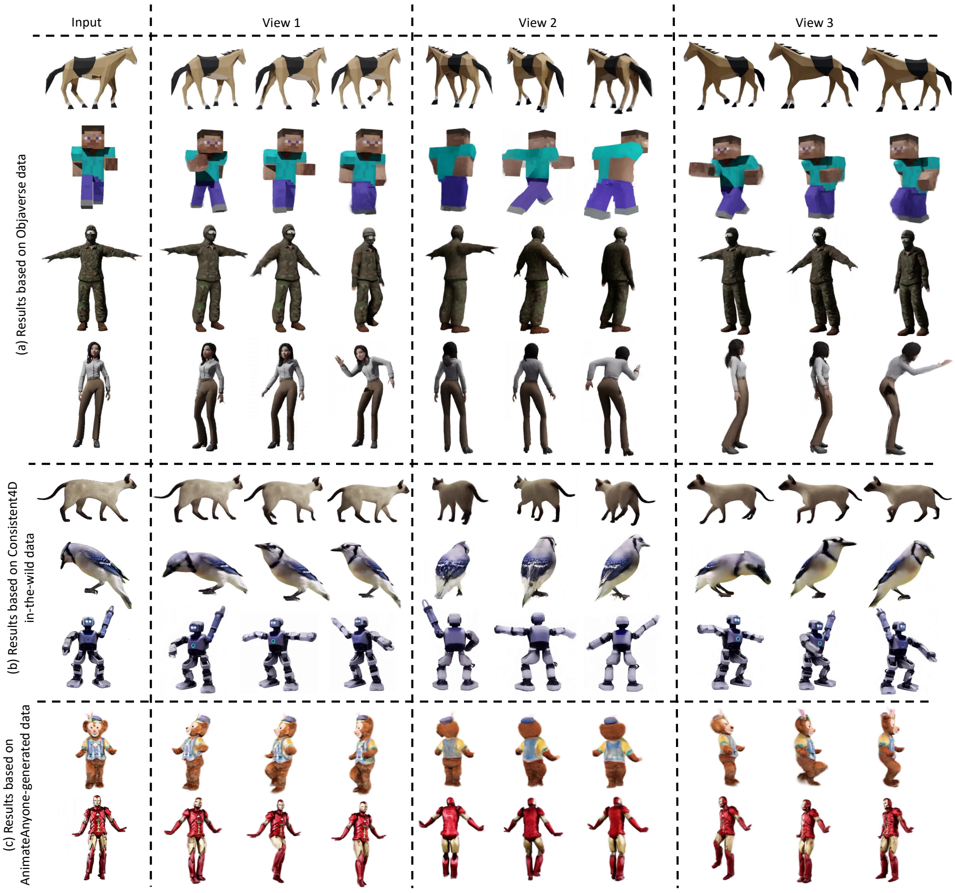

Input
View 1
View 2
View 3
(a) Results based on Objaverse data
(b) Results based on Consistent4D in-the-wild data
(c) Results based on AnimateAnyone-generated data

Figure 15: Extended results rendered by our EG4D based on semantic/large motion of synthetic/real-world objects. The input data includes **(a)** single-view rendering from Objaverse (Deitke et al., 2023) objects, **(b)** in-the-wild videos from Consistent4D (Jiang et al., 2024b), and **(c)** character motions generated by pose-conditioned video diffusion, AnimateAnyone (Hu, 2024).

**Qualitative comparison with Efficient4D.** We compare our results with another baseline Efficient4D (Pan et al., 2024), which uses 4DGS (Yang et al., 2024a) as reconstruction backbone. Consistent with quantitative results in main paper Figure 2, the 4D results generated by our method have higher view consistency and temporal consistency.

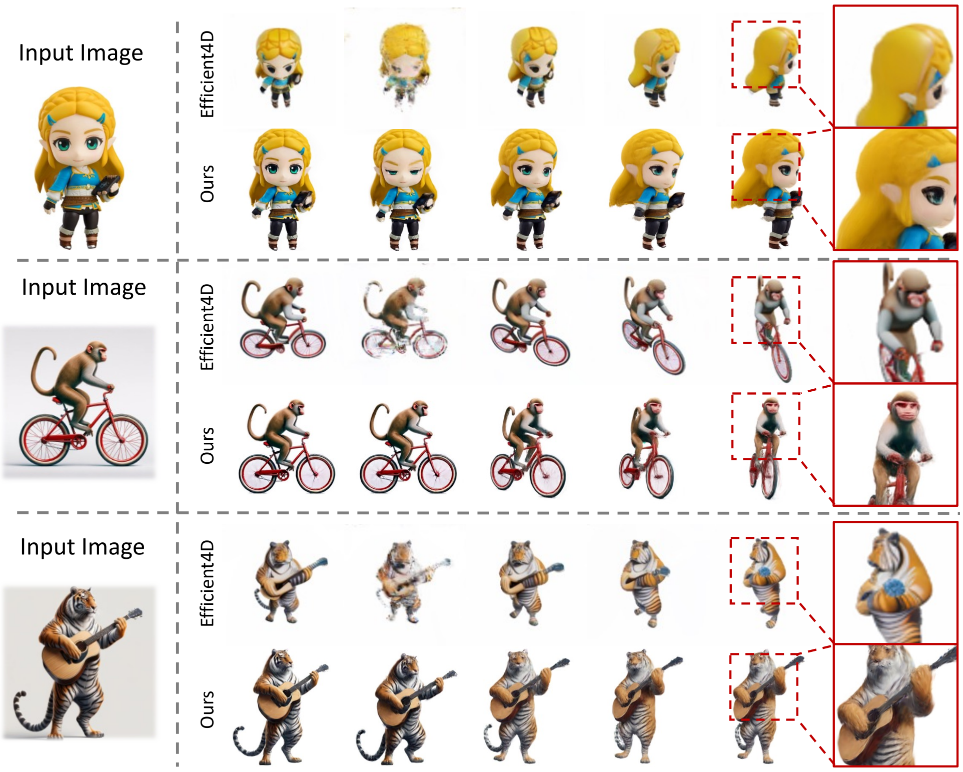


Figure 16: Qualitative comparison with another baseline, Efficient4D (Pan et al., 2024).

| Method | CLIP-I ↑ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FVD↓ | CD-FVD↓ |
|---|---|---|---|---|---|---|
| SV4D (Xie et al., 2024) | 0.9459 | 22.57 | 0.852 | 0.196 | **138.81** | **311.06** |
| EG4D (Ours) | **0.9535** | **23.28** | **0.904** | **0.173** | 142.34 | 459.10 |

Table 5: Quantitative comparison with a training-based method, SV4D (Xie et al., 2024).

**Comparison with training-based methods.** Recent works (Liang et al., 2024; Jiang et al., 2024a; Xie et al., 2024) have advanced multi-view video diffusion through training on large-scale 4D datasets, demonstrating significant improvements in 4D generation quality. Notably, Animate3D (Jiang et al., 2024a) extends AnimateDiff (Guo et al., 2024a) to generate spatiotemporally consistent multi-view videos of static 3D objects. We compare our method with SV4D (Xie et al., 2024), as shown in Figure 17 and Table 5. While SV4D achieves better temporal consistency, our approach exhibits superior image fidelity and view-consistency. This is evident in examples like `luigi` and `zelda`, where SV4D produces overly bright faces lacking detail and shading. This suggests that while SV4D performs well on its training set, it may have limited generalization capability on out-of-distribution (O.O.D.) samples.

**More discussion about training-based methods.** Our framework offers two key advantages over training-based methods like Diffusion4D and SV4D: First, our approach is *training-free*, leveraging off-the-shelf video and multi-view diffusion models without modifications. This allows rapid adoption of advances in either model type to generate 4D content efficiently, eliminating the need for expensive training on large-scale 4D datasets. Second, our method maintains *dataset independence* and directly benefits from improvements in video diffusion. Regarding motion of 4D object, advanced video diffusion models like CogVideoX (Yang et al., 2024b) would enable more dynamic and diverse animations. For 3D content, multi-view diffusion for 3D scene, *e.g.*ViewCrafter (Yu et al., 2024), would provide possibility for 4D scene generation.

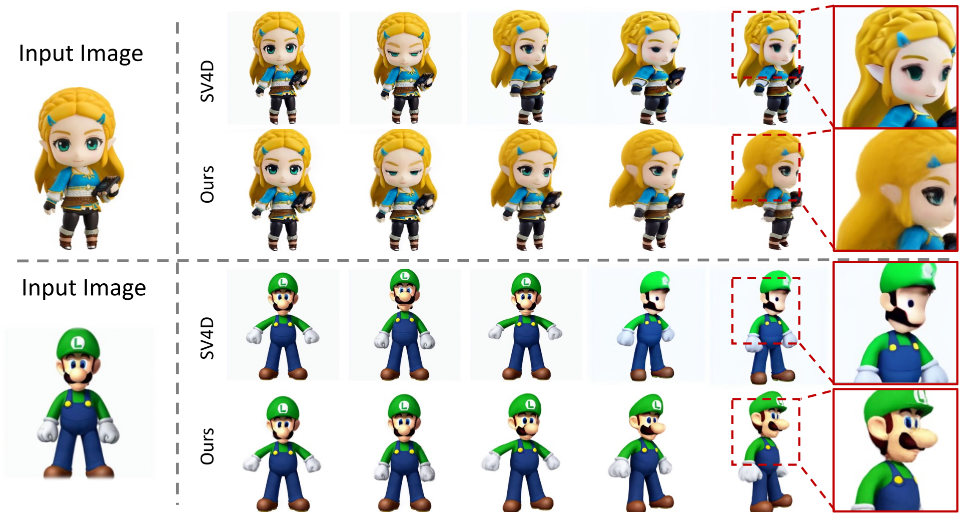


Figure 17: **Comparison with Training-based method, SV4D (Xie et al., 2024).** Benefited from the dataset-independency, we achieve higher view-consistency compared to SV4D in O.O.D data.

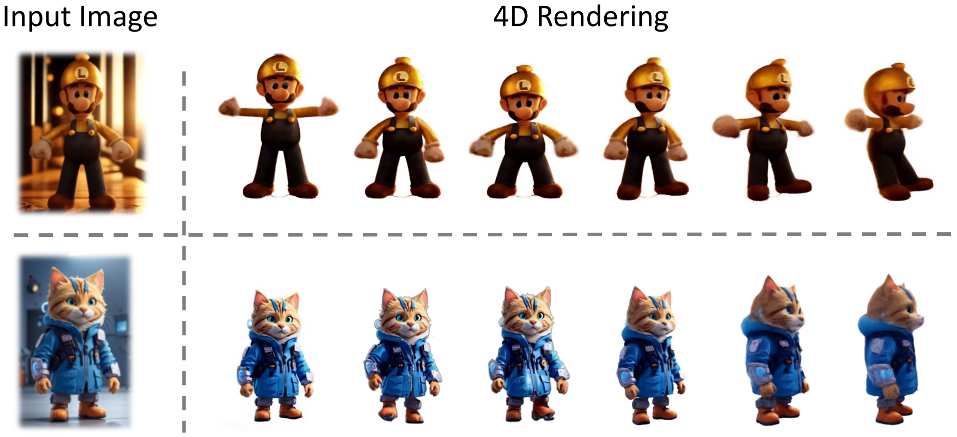


Figure 18: **4D generation results with complex image prompts.** Figure down is `luigi` relighted by IC-Light (Zhang et al., 2024), and Figure above is image prompt from DreamCraft3D (Sun et al., 2023).

**4D Generation with complex image prompt.** Apart from the images/videos from Animate124 and Consistent4D benchmark, we experiment our methods in more complex image prompt from DreamCraft3D (Sun et al., 2023). For images with complex background (Figure down) and with extreme lighting (Figure up), we find that our method can produce results with high view-consistency, image fidelity and substantial motion range. Two generated videos are included in the supplementary material.

**Efficiency.** Our framework takes approximately 1 hour and 15 minutes on average for each 4D object generation in a single NVIDIA RTX 3090. Specifically, Stage I requires about 20 minutes for video and multi-view generation; Stage II, involving 4D Gaussian Splatting optimization, takes around 25 minutes; and the refinement process takes about 30 minutes. In previous works, Consistent4D (Jiang et al., 2024b) and Animate124 (Zhao et al., 2023) take about 2.5 and 9 hours, respectively, for 4D generation. Notably, DreamGaussian4D (Ren et al., 2023) achieves extremely short optimization time of 7 minutes. Diffusion4D (Liang et al., 2024) and SV4D (Xie et al., 2024)

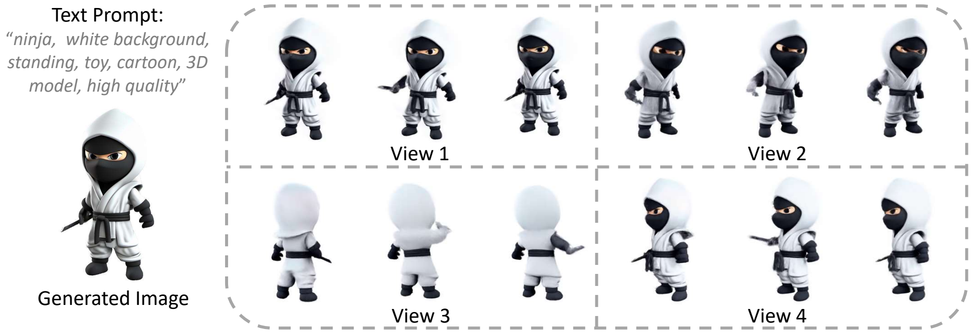

Text Prompt:
"ninja, white background, standing, toy, cartoon, 3D model, high quality"
Generated Image
View 1
View 2
View 3
View 4

Figure 19: **Text-to-4D results.** We feed a text prompt (left top) into SDXL Podell et al. (2024) to generate a ninja image (left bottom). This image can be transformed into 4D objects with our framework, presenting indirect text-to-4D application. The right panel shows multi-view renderings of the 4D model.

train a diffusion network to generate the multi-view multi-frame image matrix. We have comparable inference time since both methods share similar pipeline of multi-view video generation and 4D representation optimization. However, they need to take huge computational expense for training. In contrast to optimization-based approaches, L4GM (Ren et al., 2024) uses feed-forward network to direct predict the Gaussian sequences within several minutes. Our optimization time falls between these, but our framework offers superior view consistency, 3D appearance, and motion quality. Since Stage I appears to be one of the efficiency bottlenecks, future work should focus on incorporating efficient sampling for video diffusion models to boost speed.

**Multi-view results of our results.** Figure 23 shows the multi-view results of our 4D model, which is a supplement of Figure 4. Due to the page limit of the main paper, we only show two views of the 4D model there, which is not enough to illustrate the 3D appearance of our model. To this end, we render our model in more views: 0°, 90°, 135°, 180°, 225°, and 270°. The rendered multi-view images show that our method can produce images with high 3D consistency and satisfactory quality.

**More visual comparisons.** Figure 24 provides additional visual comparisons with our baselines, continuing from Figure 3 in the main paper. We use three additional cases: `luigi`, `anya`, and `chicken-basketball`. The first two columns show animation results from the same view, while column 3 to 5 display three different views. The last column presents a zoomed-in image of the final rendered view. Multi-view videos for visual comparison can be found in the *supplementary*.

**More applications.** Benefiting from our explicit generation, we can easily adapt EG4D to both text-to-4D and video-to-4D tasks. Figure 19 shows the generation results of the text-to-4D. We first feed an example text prompt into SDXL (Podell et al., 2024) to get the high-resolution image. Then this image is transformed into a 4D model with our framework. Figure 20 shows the results of the video-to-4D. We just skip the dynamic generation step and start with our view synthesis pipeline.

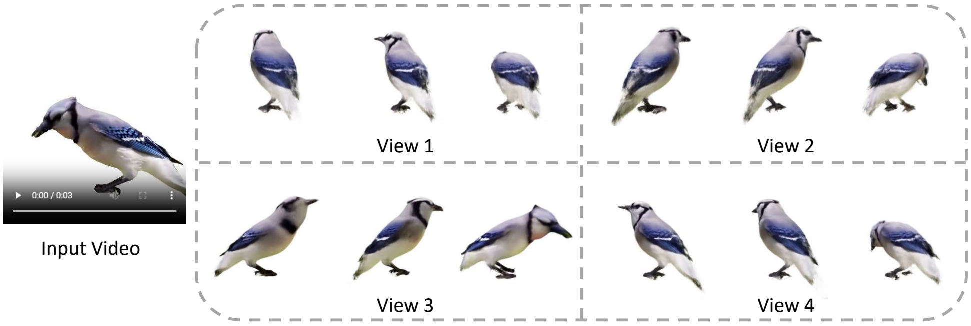
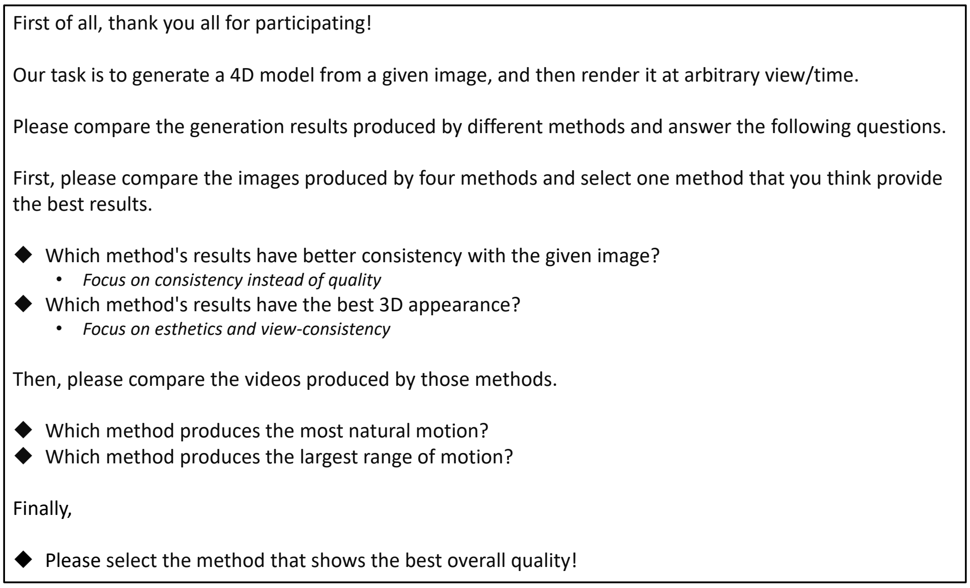


Figure 20: **Video-to-4D results.** Our framework can be seamlessly extended to video-to-4D generation. The right panel shows the renderings of our 4D model from four viewpoints. This `bird` video is taken from Consistent4D (Jiang et al., 2024b).



First of all, thank you all for participating!

Our task is to generate a 4D model from a given image, and then render it at arbitrary view/time.

Please compare the generation results produced by different methods and answer the following questions.

First, please compare the images produced by four methods and select one method that you think provide the best results.

- ◆ Which method's results have better consistency with the given image?
  - *Focus on consistency instead of quality*
- ◆ Which method's results have the best 3D appearance?
  - *Focus on esthetics and view-consistency*

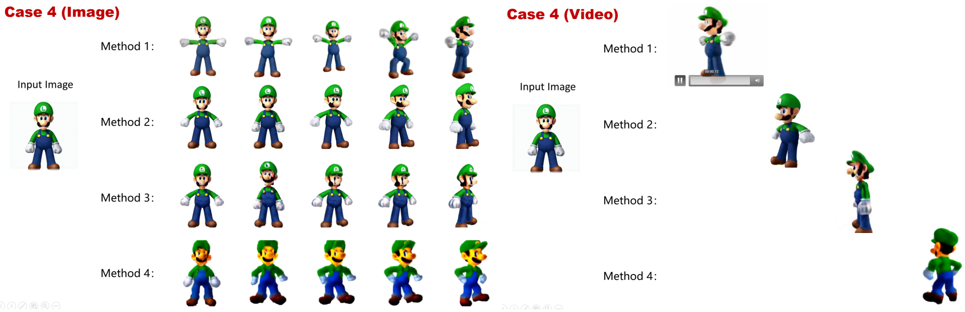
Then, please compare the videos produced by those methods.

- ◆ Which method produces the most natural motion?
- ◆ Which method produces the largest range of motion?

Finally,

- ◆ Please select the method that shows the best overall quality!

Figure 21: **Screenshot of our user study guidelines.** Each participant is asked to evaluate the images and videos rendered by 4 different methods with 5 metrics, *i.e.*, reference view consistency, 3D appearance, motion realism, motion range, and overall quality.



(a) Screenshot of the image evaluation.

(b) Screenshot of the video evaluation.

Figure 22: **Screenshot of our user study content.** Each participant is provided with several images and videos rendered by different methods.

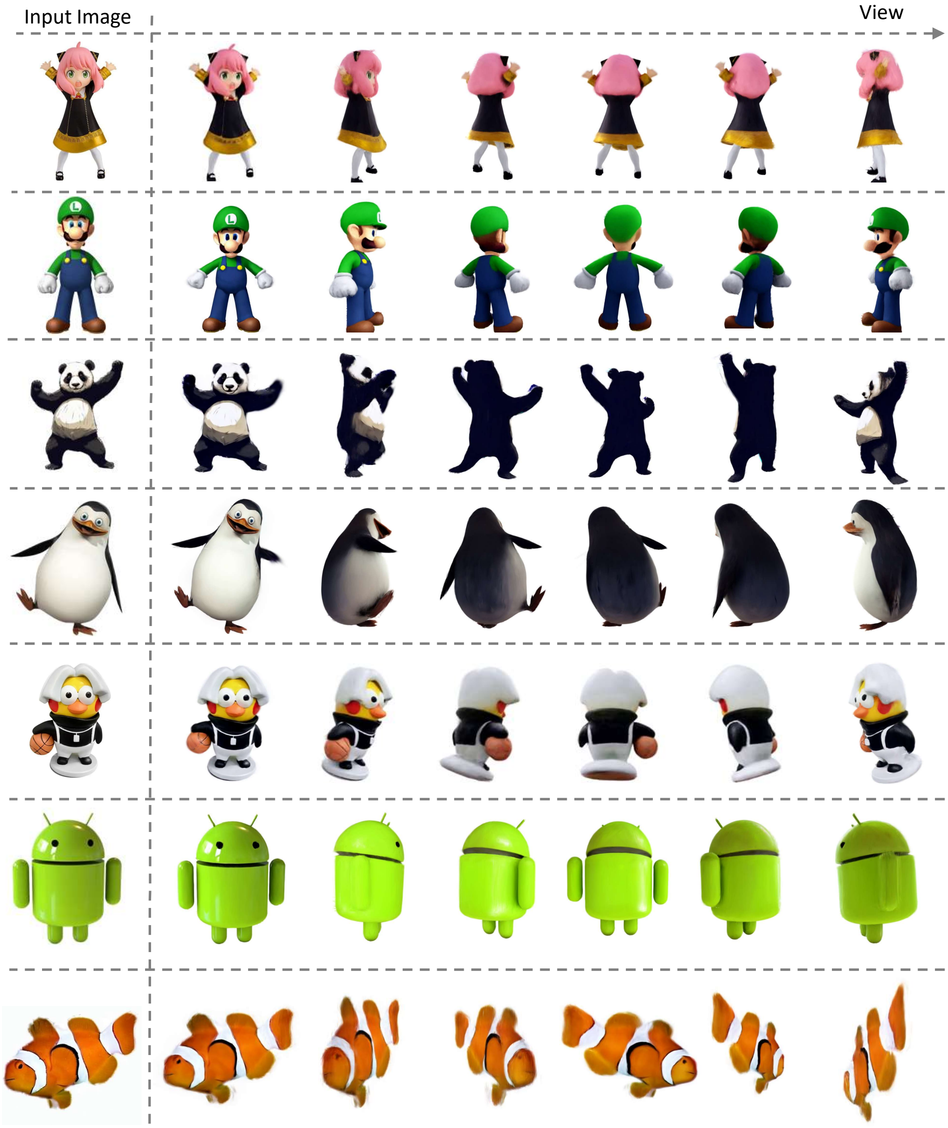


Figure 23: **Multi-view results of our models.** This figure is a supplement of Figure 4 in the main paper. The 6 columns show the images rendered by our model in different views: 0°, 90°, 135°, 180°, 225°, and 270°. Multi-view renderings demonstrate the geometry/texture consistency and promising quality of our 4D representation.

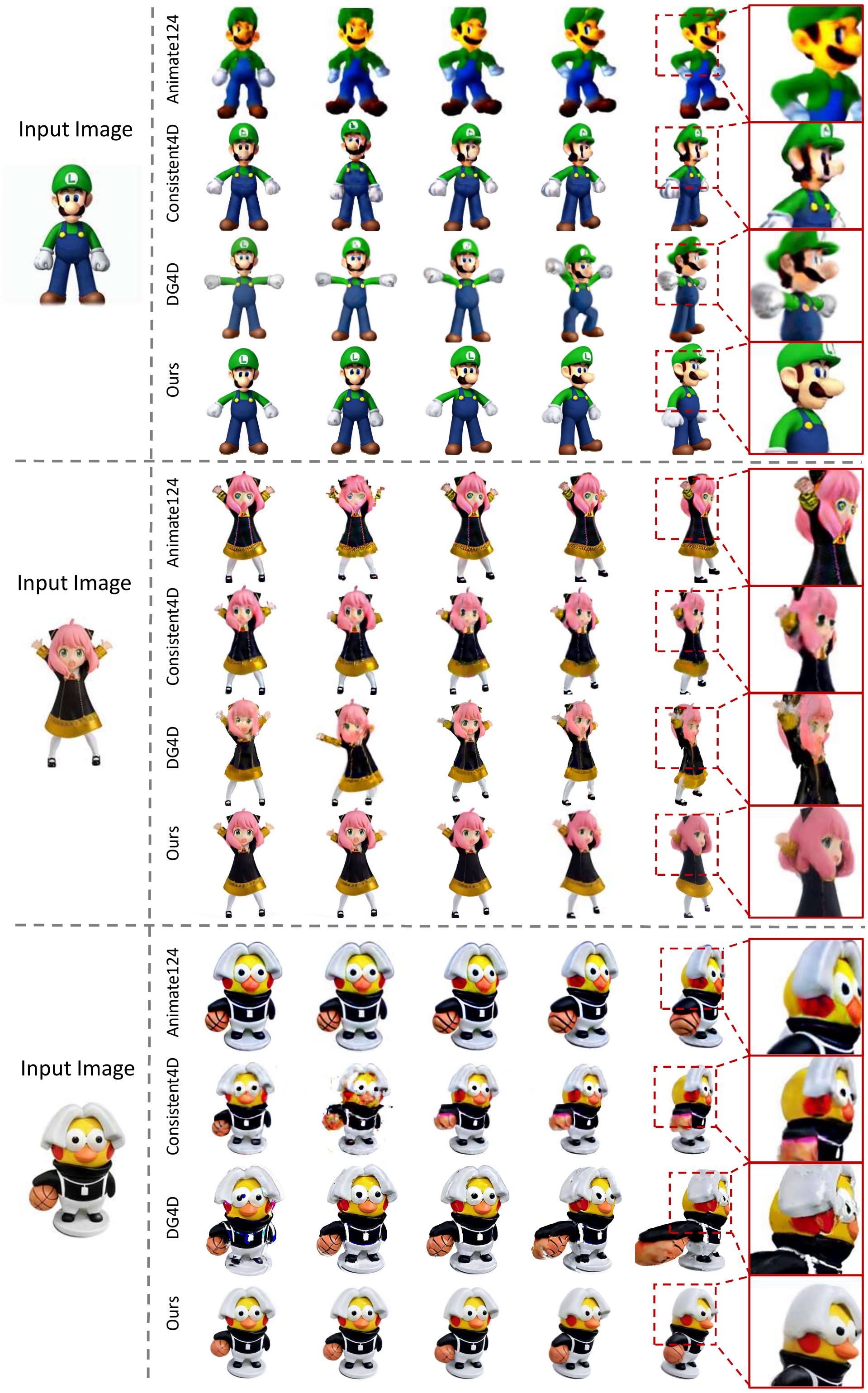


Figure 24: More comparison examples with the SOTA results in three cases `luigi`, `anya` and `chicken-basketball` (better zoom in).