Not All Frame Features Are Equal: Video-to-4D Generation via Decoupling Dynamic-Static Features

Supplementary Material

1. The Principle of Dynamic Saliency

Our use of feature norm as a saliency indicator is motivated by the observation that regions with larger semantic magnitudes across time tend to exhibit higher semantic consistency, while dynamic regions often yield smaller projection values due to variations in object pose and position. This principle aligns with prior studies [6, 9, 17], where feature norms are used to estimate attention, objectness, or semantic relevance. Accordingly, in Eq.(1) in main article, we adopt the feature norm to approximate each region's semantic response strength, assuming that consistently high responses indicate static or salient areas.

2. Training Objectives

In this section, we present more details on training objectives. Please note that each loss in our loss function has a clear derivation from previous works. The theoretical derivation can be found in corresponding previous works, e.g., [13, 19, 21] etc. Hence, we will not provide a detailed derivation process in our article.

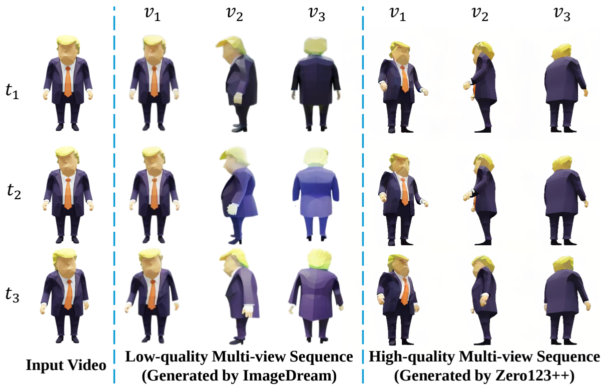
Score distillation sampling loss. Following [19], we employ multi-view score distillation sampling (SDS) loss using rendered images under camera poses of pseudo multi-view images and input video.In detail, there are rendered multi-view images $I = \{I^{(i,1)}, \dots, I^{(i,j)}\}$ under 6 camera poses, where i denotes the timestamps and j denotes the number of views. Formally, SDS loss can be defined as:

$$\begin{aligned} \mathcal{L}_{SDS} &= \alpha_1 \mathcal{L}_{SDS}^{pseudo} + \alpha_2 \mathcal{L}_{SDS}^{real} \\ &= \alpha_1 \mathcal{L}_{SDS}(\phi, I^{(i,j)}) + \alpha_2 \mathcal{L}_{SDS}(\phi, I_{real}^{i}) \end{aligned} \tag{1}$$

where α_1 and α_2 are hyperparameters, I_{real}^{i} is the rendered image under camera pose of input video at time i.

Photometric loss. Following [13, 19], we compute the reconstruction loss $\mathcal{L}_{rec}$ between rendered images and pseudo multi-view images, and the foreground mask $\mathcal{L}_{mask}$.

LPIPS loss. We introduce the LPIPS loss $\mathcal{L}_{lpips}$ [21] to compute the feature similarity between pseudo multi-view images and corresponding rendered images. We leverage VGG [12] as backbone to extract image features.



v_1
v_2
v_3
v_1
v_2
v_3
t_1
t_2
t_3
Input Video
Low-quality Multi-view Sequence (Generated by ImageDream)
High-quality Multi-view Sequence (Generated by Zero123++)

Figure 1. Example on low-quality images generated by ImageDream and high-quality images generated by Zero123++, both using the same input video.

Overall loss. Based on above loss functions, we obtain the final overall loss $\mathcal{L}$:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{SDS} + \lambda_2 \mathcal{L}_{rec} + \lambda_3 \mathcal{L}_{mask} + \lambda_4 \mathcal{L}_{lpips} \tag{2}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are hyperparameters, we set $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1.0$.

3. Training Details of DS4D-GA and DS4D-DA

We train our two models DS4D-GA (using GA in TSSF) and DS4D-DA (using DA in TSSF) under the same training setting. During the initial 1,000 iterations, we train our models except TSSF and deformation MLP. Subsequently, the models with TSSF and deformation MLP are optimized over 6,000 additional iterations. For the deformation MLP, we employ each MLP with 64 hidden layers and 32 hidden features. The learning rate of TSSF and deformation MLP is set to 1.6×10^{-4} and is decayed to 1.6×10^{-6}. Following [19], the top 2.5% of points are densified with the most accumulated gradient. The overall training process costs approximately 3 hours on a V100 GPU.

Methods	High-quality Input Images				Low-quality Input Images			
	CLIP ↑	LPIPS ↓	FVD ↓	FID-VID ↓	CLIP ↑	LPIPS ↓	FVD ↓	FID-VID ↓
STAG4D [19]	0.9078	0.1354	986.8271	26.3705	0.9026	0.1437	1311.8770	40.8664
DS4D-GA (Ours)	0.9206	0.1311	799.9367	26.1794	0.9195	0.1380	849.8154	25.8726
DS4D-DA (Ours)	**0.9225**	**0.1309**	**784.0235**	**24.0492**	**0.9221**	**0.1339**	**805.4721**	**24.0623**

Table 1. Evaluation and comparison of the performance when facing low-quality input images and high-quality input images. The best score is highlighted in bold.

4. Training Details on Discussion

In this section, we present the training details of experiments about Discussion (Section 5) in our manuscript. For a fair comparison, the training settings are the same as 4D-GS [15].

Network Architecture. We add our proposed DSFD in 4D-GS. Meanwhile, we directly replace the spatial-temporal structure encoder and multi-head Gaussian deformation decoder of 4D-GS with our TSSF. Since the multi-view sequences from Neu3D's dataset are authentic, we use TSSF-GA instead of TSSF-DA. Additionally, the hyperparameter settings of HexPlane are the same as 4D-GS. In detail, the basic resolution of HexPlane is 64, and is unsampled by 2 and 4.

Training Settings. We use the same dense point cloud generated by SFM as 4D-GS. Then we also downsample point clouds lower than 100k. The learning rate of TSSF and deformation MLP are set to 1.6×10^{-3} which decreases to 1.6×10^{-5}. The batch size is 1. Following 4D-GS, we do not use opacity reset operation. The overall training iterations spend 14000 for each scene.

Note that: a) Our DS4D includes point initialization (init.) via LRM. However, in Neu3D's data, common point cloud init. (e.g., 4D-GS) uses colmap rather than LRM. Thus, it is unfair to compare DS4D and 4D-GS due to the different init.. b) DSFD and TSSF are core contributions, thus inserting them into 4D-GS can directly demonstrate the effectiveness of our contributions in real-world scenes.

5. More details on Datasets

In this section, we present more details on **four datasets**.

Consistent4D Dataset. Following [19], we use seven 30-frame video in front view as input video, and their ground-truth with four novel views (azimuth angles of $-75°$, $15°$, $105°$, and $195°$, respectively) as evaluation.

Objaverse Dataset. We random sample seven dynamic objects from [2, 8]. The 24-frame ground truth under 360° cameras (the range of azimuth angles is $[0°, 360°]$) rendered from each object is used as evaluation. Meanwhile, we use seven 24-frame videos in front view as input video. Compared to Consistent4D Dataset, the object in Objaverse Dataset has more complex motion, e.g., suddenly waving at some time.

Neu3D's Dataset. We use three real-world scenarios from Neu3D's dataset [7]. Each scene has 300 frames with 20 cameras, a total of 6000 high-quality images. Following [15], we use 300 frames under the front camera pose as evaluation, others are used as training videos.

Data from Online Sources. Following [19], we introduce some challenging videos from online sources for qualitative evaluation. Moreover, we also generate input videos by Stable Video Diffusion [1]. Each input video has 14 or 30 frames.

6. Experiments

In this section, we conduct more experiments to evaluate our method.

6.1. Robustness of Our Methods

In this section, we explore whether the quality of input images has a huge influence on the generation results of our methods.

Specifically, we construct a dataset with low-quality input multi-view images. Using the same input video as Tab.1 of the main manuscript, we leverage ImageDream [14] to produce a series of multi-view sequences. Then, we select multi-view images with inconsistency or shape deformation from the generation multi-view sequences. These low-quality multi-view images are grouped as the low-quality input images. The example data of low-quality inputs can be seen in Fig. 1. The low-quality inputs has serious inconsistency between different timestamps and has color fading and texture blurry compared with the high-quality inputs we used in Tab.1 of the main manuscript (e.g., the shape and color of the suit).

Based on the low-quality inputs, we compare our methods with STAG4D. The quantitative and qualitative results are shown in Tab. 1 and Fig. 2. Our methods when using low-quality inputs maintain a similar performance compared to our results when using high-quality inputs. However, in image metrics (LPIPS) and video metrics (FID and FID-VID), STAG4D when using low-quality inputs has a significantly worse performance compared to STAG4D using high-quality inputs. Furthermore, in Fig. 2, STAG4D generates the blurry textures in the back view. The reason is that back's texture details are blurry in the low-quality input at some timestamps (e.g., at t_1 of v_3). In contrast, our

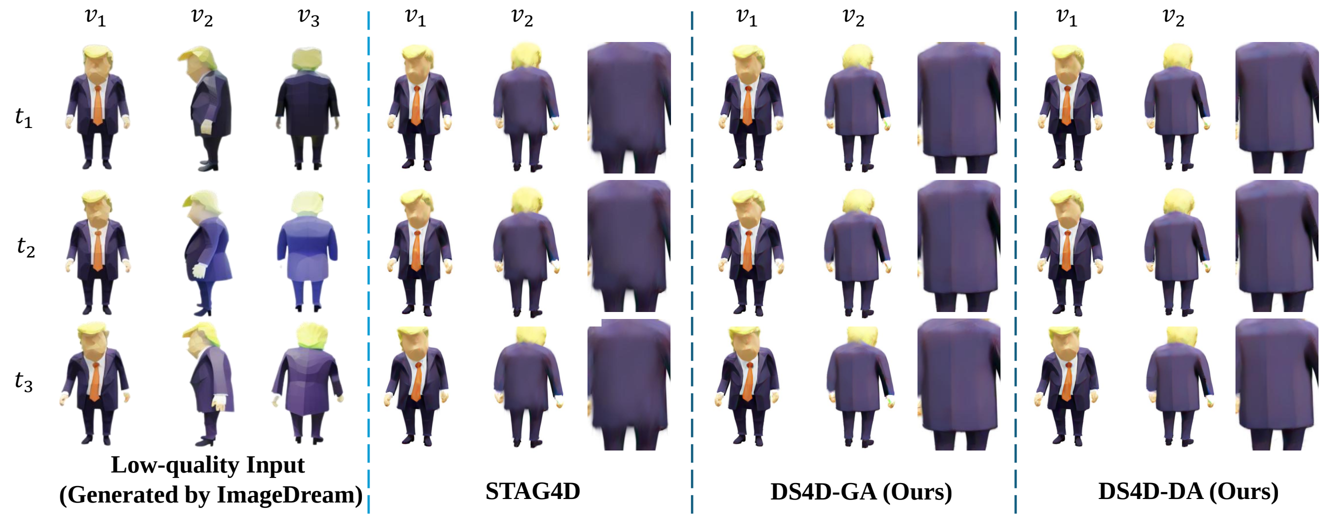


Figure 2. Qualitative comparison on 4D generation results with low-quality inputs. For each method, we render results under two novel views at three timestamps.

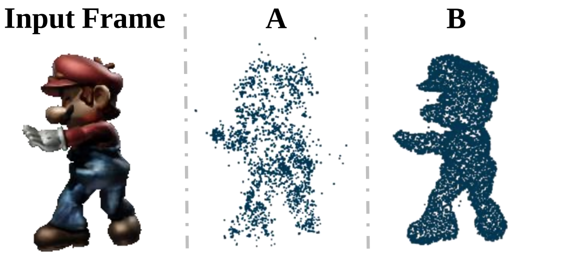


Figure 3. Comparison regarding whether using point clouds generated by a large reconstruction model as initialization. A: The model without point initialization. B: The model with point initialization.

methods can generate results with clear textures. This indicates that our methods can handle the low-quality inputs better than STAG4D. Since our methods decouple dynamic-static information at the feature-level. Even though input low-quality data, thanks to the robustness feature extraction ability of DINOv2, we can still leverage the inherent differences between features to decouple. The differences include the change of motion, shape and textures between the corresponding two frames.

In summary, the above experiments demonstrate that the quality of input images has few influence on the generation results of our methods and verify the robustness of our methods when input low-quality data.

6.2. Analysis on Different Point Initializations

We evaluate the effect of our method using LRM (OpenLRM) [3, 4] as point clouds initialization on Tab. 2. The results verify that our method using LRM achieves comparable performance.

6.3. Analysis on Frame choosing

We evaluate the effect of choosing first/last/middle frame for initialization and reference on Tab. 3. The results verify that our method exhibits comparable performance regardless of first/last/middle (CLIP variance $\approx 1.4\times10^{-7}$). Because in our method: Deformation MLP uses fused Gaussian features (FGS) to predict deformation of Gaussian properties between current frame and initialization frame, where FGS includes dynamic features. The initialization frames are the same as one of the reference frames, which ensures the dynamic features indicate the motion trend of current frame relative to reference frames.

6.4. More Analysis on Ablation Experiments

The effect on points initialization. To validate the effect of point initialization, we add the point initialization to the baseline model, which is labeled as B (as shown in Table.2 of the manuscript). After performing point initialization, the performance of B model has improved on all metrics. Additionally, we visualize the Gaussian points from baseline model (labeled as A, as shown in Table.2 of the manuscript) and B in Fig.3. Obviously, the Gaussian points of B are relatively uniform and denser in distribution than those of A, which ensures the stability of optimization and fidelity of the motion and shape fully learned by the model.

The effect on LPIPS Loss. To validate the effect of LPIPS Loss, we add the LPIPS Loss based on model B, labeled as C (as shown in Table.2 of the manuscript). The performance of C model has improved on all metrics. It indicates the effectiveness of LPIPS loss.

Methods	Point Initialization	Consistent4D dataset				Objaverse dataset			
		CLIP ↑	LPIPS ↓	FVD ↓	FID-VID ↓	CLIP ↑	LPIPS ↓	FVD ↓	FID-VID ↓
DS4D-GA	InstantMesh	0.9206	0.1311	799.9367	26.1794	0.8868	0.1761	890.2646	26.6717
DS4D-DA	InstantMesh	**0.9225**	**0.1309**	**784.0235**	**24.0492**	**0.8881**	**0.1759**	**870.9489**	**25.3836**
DS4D-GA	LRM	0.9200	0.1319	804.4437	26.1927	0.8862	0.1768	897.2301	26.7553
DS4D-DA	LRM	0.9220	0.1312	793.0164	24.1553	0.8875	0.1764	874.9876	25.4868

Table 2. Evaluation and comparison of the performance on Consistent4D dataset and Objaverse dataset when using different point initializations.

Methods	Consistent4D dataset				Objaverse datase			
	CLIP ↑	LPIPS ↓	FVD ↓	FID-VID ↓	CLIP ↑	LPIPS ↓	FVD ↓	FID-VID ↓
First Frame	0.9219	0.1306	788.8806	24.0211	0.8885	0.1759	871.0908	25.3978
Last Frame	0.9216	0.1307	785.1663	24.0728	0.8882	0.1760	873.8067	25.3407
Middle Frame	0.9225	0.1309	784.0235	24.0492	0.9221	0.1339	805.4721	24.0623

Table 3. Evaluation and comparison of the performance when choosing first/last/middle frame for initialization and reference.

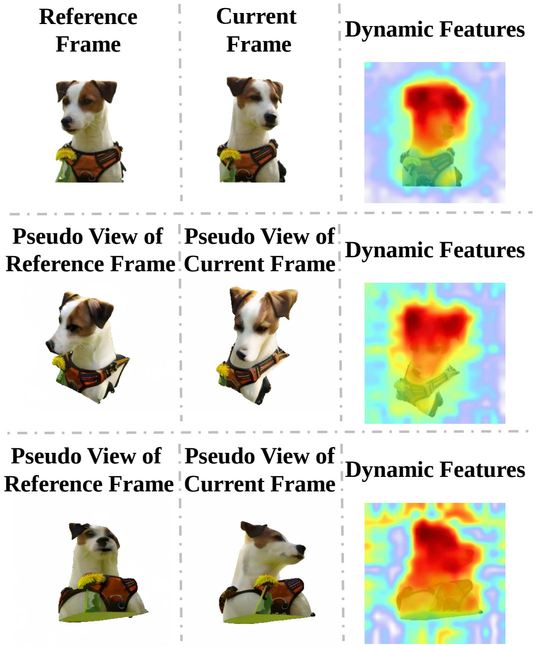

Reference Frame
Current Frame
Dynamic Features
Pseudo View of Reference Frame
Pseudo View of Current Frame
Dynamic Features
Pseudo View of Reference Frame
Pseudo View of Current Frame
Dynamic Features

Figure 4. Visualization on the heatmap of dynamic features in DSFD. The red region highlights the primary zone of interest in dynamic features.

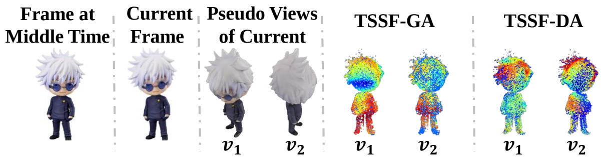


Figure 5. Visualization on the score map of point features in TSSF-GA and TSSF-DA. The red area indicates model's high attention on dynamic information in point features of a specific view.

6.5. More Visualization

In this section, we provide more visualizations of features in DSFD and TSSF, respectively.

DSFD. In Fig.4, we supplement more heatmaps of dynamic features obtained by DSFD decoupling features from the current and reference frame features. The red area indicates the primary region of interest in the features. The dog's head movements indicate the main motion trends between the reference frame and the current frame. No matter what kind of novel views, dynamic features decoupled by our DSFD can accurately represent the dynamic zones. It once again demonstrates our method can acquire accurate dynamic features using DSFD.

TSSF. In Fig.5, we supplement more score maps of selecting similarity dynamic information from point features of different views by TSSF-GA and TSSF-DA. Moreover, Fig.6 shows the corresponding generation results at the current timestamp, including the same example as Figure.8 (b) in the manuscript and the same example in Fig.5.

Specifically, the head movements of the person indicate the primary trend in motion between the middle and current time. The red area indicates the model's high attention on dynamic information in point features of a specific view. According to score maps based on different views, two approaches can capture a certain degree of similar dynamic information from different viewpoints. TSSF-GA is interested in both body and head in v_1, but TSSF-DA pays more concentration to the head. This is because TSSF-DA reduces the impact of novel views, resulting in TSSF-DA focusing more on capturing regions with a motion trend that is more similar to the front frame in other novel views. Thus, compared to TSSF-GA, TSSF-DA can produce results with clear details in the corresponding regions. For example, the hair texture of person at the top of Fig.6, and the leg texture of triceratops at the down of Fig.6. It once again indicates TSSF-DA can alleviate issues caused by novel views.

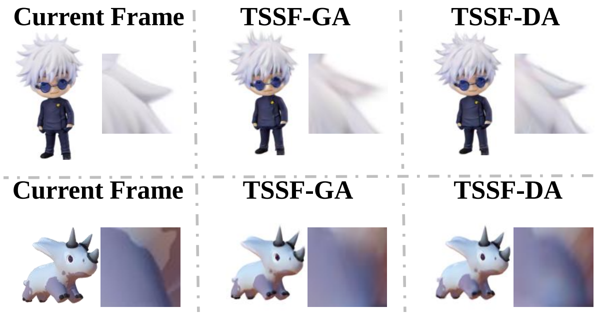


Figure 6. Qualitative comparison between model using TSSF-GA and model using TSSF-DA at current timestamp. Their corresponding visualization on the score map of point features is as shown in Fig.5 and Figure.8 (b) in our manuscript.

Direct Decoupling	Our Decoupling Approach
118.326 (ms)	8.241 (ms)

Table 4. Comparison of running time with different decoupling approaches. All approaches are tested on a NVIDIA 3090 GPU.

6.6. More Qualitative Comparisons

We supplement more qualitative comparisons in Fig.8. We compare our methods DS4D-GA and DS4D-DA with other SOTA methods, including Consistent4D [5], Dreamgaussian4D [10], STAG4D [19], SC4D [16], 4Diffusion [20], and L4GM [11]. All the experiments of the methods are carried out using the code from their official GitHub repository.

6.7. More Examples on 4D Content Generation

We supplement more 4D content generation examples produced by DS4D-GA and DS4D-DA in Fig.7.

6.8. More Examples on Real-World Scenario

We supplement more real-world 4D scene generation examples using our method and baseline 4D-GS [18] in Fig.9. The experiments of 4D-GS are carried out using the code from their official GitHub repository.

6.9. Time Consuming on Decoupling

As mentioned in our manuscript, direct decoupling always costs considerable computation time. To intuitively evaluate the time-consuming advantage of our decoupling approach compared to direct decoupling, we provide the running time of each approach on decoupling dynamic-static features from a frame features with a 30-frame video, as shown in Tab.4. Undoubtedly, our decoupling approach is about 14 times faster than direct decoupling.

7. Limitations

Limited to the resolution of input video, it is challenge for our method to produce high-resolution 4D contents , e.g., over 2K resolution.

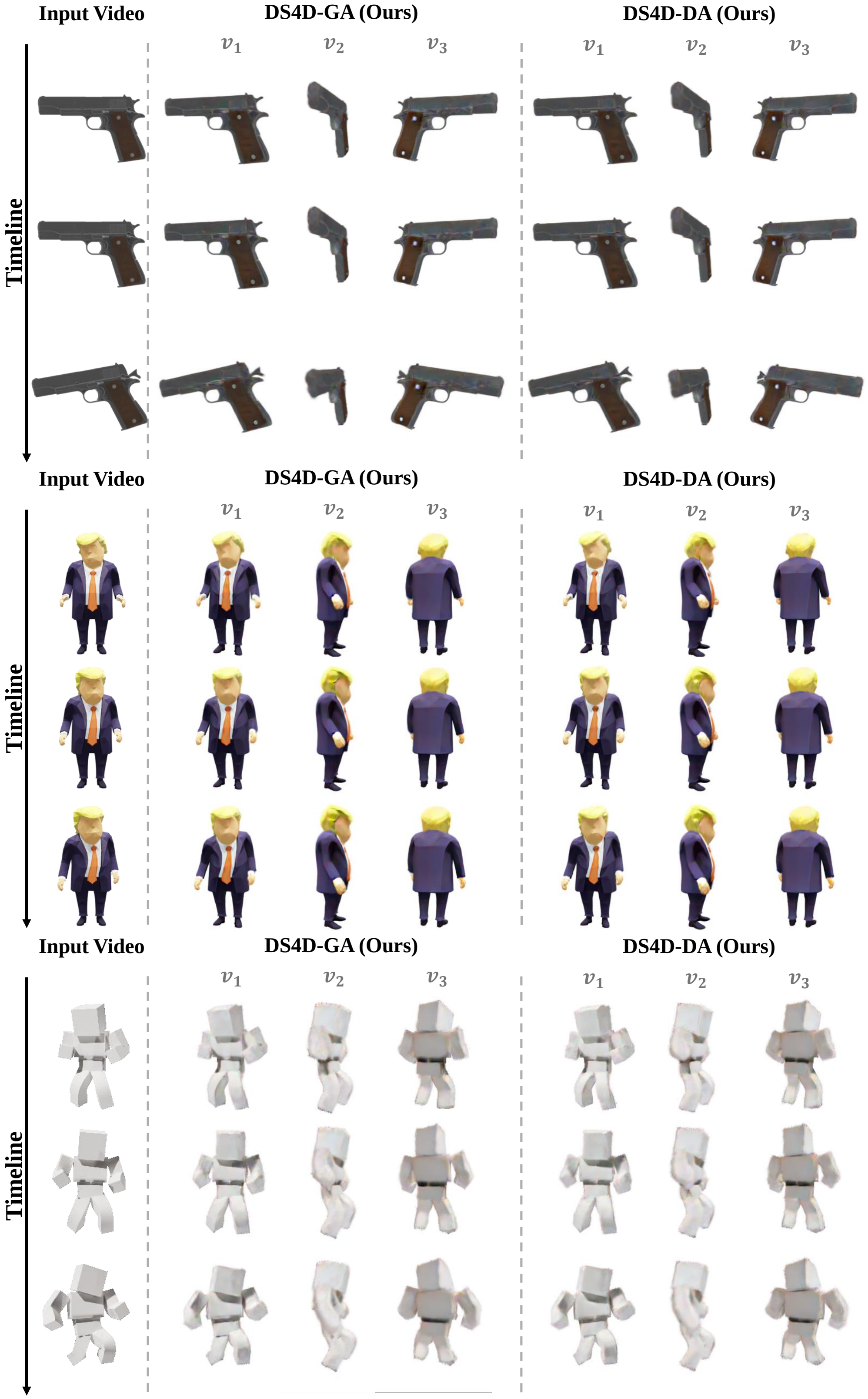


Figure 7. More Results for 4D Generation using DS4D-GA and DS4D-DA.

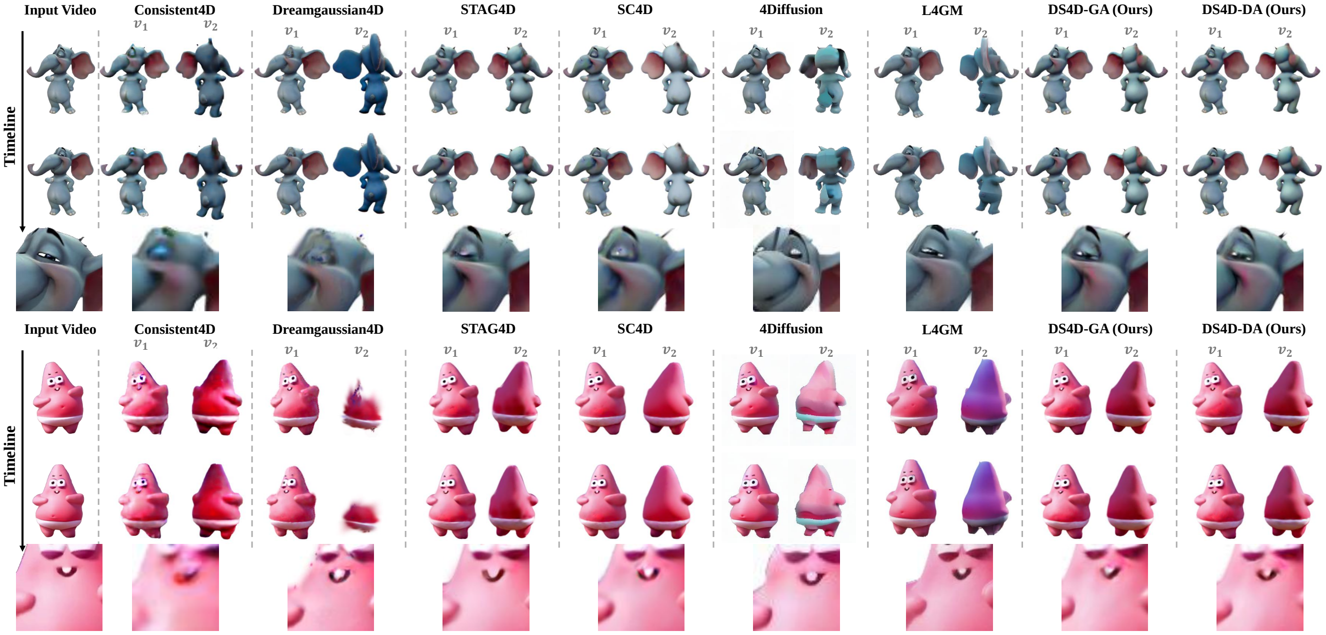
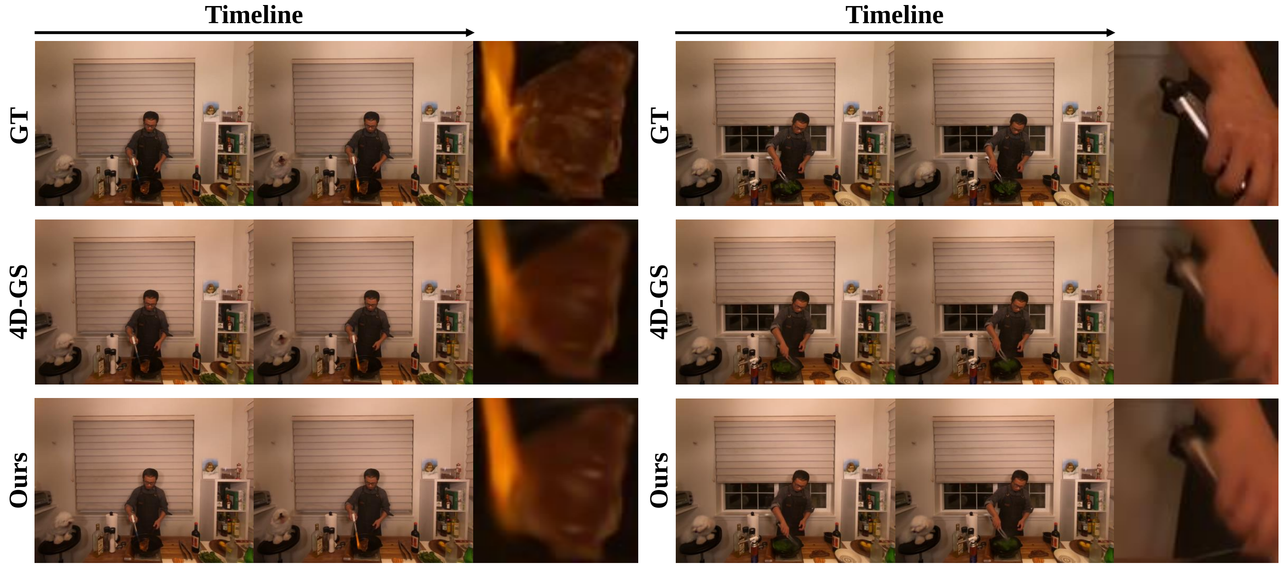


Figure 8. Qualitative comparison on video-to-4D generation. For each method, we render results under two novel views at two timestamps.



(a) Sear steak

(b) Cook spinach

Figure 9. Visualization of real-world 4D scene generation compared with 4D-GS.

References

[1] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2

[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] Zexin He and Tengfei Wang. Openlrm: Open-source large reconstruction models. `https://github.com/3DTopia/OpenLRM`, 2023. 3

[4] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to

3d. *arXiv preprint arXiv:2311.04400*, 2023. 3

[5] Yanqin Jiang, Li Zhang, Jin Gao, Weiming Hu, and Yao Yao. Consistent4d: Consistent 360° dynamic object generation from monocular video. In *The Twelfth International Conference on Learning Representations*, 2024. 5

[6] José Lezama, Qiang Qiu, Pablo Musé, and Guillermo Sapiro. Ole: Orthogonal low-rank embedding-a plug and play geometric loss for deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8109–8118, 2018. 1

[7] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 2

[8] 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

[9] Yang Liu, Lei Zhou, Pengcheng Zhang, Xiao Bai, Lin Gu, Xiaohan Yu, Jun Zhou, and Edwin R Hancock. Where to focus: Investigating hierarchical attention relationship for fine-grained visual classification. In *European Conference on Computer Vision*, pages 57–73. Springer, 2022. 1

[10] Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. Dreamgaussian4d: Generative 4d gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023. 5

[11] Jiawei Ren, Kevin Xie, Ashkan Mirzaei, Hanxue Liang, Xiaohui Zeng, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, et al. L4gm: Large 4d gaussian reconstruction model. *arXiv preprint arXiv:2406.10324*, 2024. 5

[12] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[13] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 1

[14] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023. 2

[15] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20310–20320, 2024. 2

[16] Zijie Wu, Chaohui Yu, Yanqin Jiang, Chenjie Cao, Fan Wang, and Xiang Bai. Sc4d: Sparse-controlled video-to-4d generation and motion transfer. *arXiv preprint arXiv:2404.03736*, 2024. 5

[17] Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuetong Xue, Fu Li, Errui Ding, and Jizhou Huang. Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features. In *Proceedings of the IEEE/CVF International conference on Computer Vision*, pages 11772–11781, 2021. 1

[18] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 5

[19] Yifei Zeng, Yanqin Jiang, Siyu Zhu, Yuanxun Lu, Youtian Lin, Hao Zhu, Weiming Hu, Xun Cao, and Yao Yao. Stag4d: Spatial-temporal anchored generative 4d gaussians. *arXiv preprint arXiv:2403.14939*, 2024. 1, 2, 5

[20] Haiyu Zhang, Xinyuan Chen, Yaohui Wang, Xihui Liu, Yunhong Wang, and Yu Qiao. 4diffusion: Multi-view video diffusion model for 4d generation. *arXiv preprint arXiv:2405.20674*, 2024. 5

[21] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1