FreeGave: 3D Physics Learning from Dynamic Videos by Gaussian Velocity

Supplementary Material

A. Proof of Divergence-free Property

The velocity field is defined as:

$$\boldsymbol{v}(\boldsymbol{p}_t, t) = \mathbb{V}_t \cdot \mathcal{B}(\boldsymbol{p}_t) = \sum_{k=1}^{6} \mathbb{V}_t^k \mathcal{B}^k(\boldsymbol{p}_t). \tag{10}$$

In order to prove the divergence-free property, we just need to show $\nabla_{\boldsymbol{p}_t} \cdot \boldsymbol{v}(\boldsymbol{p}_t, t) = 0$. Since $\mathbb{V}_t$ is totally irrelevant to $\boldsymbol{p}_t$, we only need to show each basis vector function follows $\nabla_{\boldsymbol{p}_t} \cdot \mathcal{B}^k(\boldsymbol{p}_t) = 0$, *i.e.*:

$$\nabla_{\boldsymbol{p}_t} \cdot \boldsymbol{v}(\boldsymbol{p}_t, t) = \nabla_{\boldsymbol{p}_t} \cdot (\mathbb{V}_t \cdot \mathcal{B}(\boldsymbol{p}_t)) \tag{11}$$

$$= \sum_{k=1}^{6} \mathbb{V}_t^k \nabla_{\boldsymbol{p}_t} \cdot \mathcal{B}^k(\boldsymbol{p}_t) = 0. \tag{12}$$

Next, we show each of the basis vector function is divergence-free:

$$\nabla_{\boldsymbol{p}_t} \cdot \mathcal{B}^1(\boldsymbol{p}_t) = \nabla_{\boldsymbol{p}_t} \cdot \begin{bmatrix}1 & 0 & 0\end{bmatrix} = \frac{\partial 1}{\partial p_t^x} = 0; \tag{13}$$

$$\nabla_{\boldsymbol{p}_t} \cdot \mathcal{B}^2(\boldsymbol{p}_t) = \nabla_{\boldsymbol{p}_t} \cdot \begin{bmatrix}0 & 1 & 0\end{bmatrix} = \frac{\partial 1}{\partial p_t^y} = 0; \tag{14}$$

$$\nabla_{\boldsymbol{p}_t} \cdot \mathcal{B}^3(\boldsymbol{p}_t) = \nabla_{\boldsymbol{p}_t} \cdot \begin{bmatrix}0 & 0 & 1\end{bmatrix} = \frac{\partial 1}{\partial p_t^z} = 0; \tag{15}$$

$$\nabla_{\boldsymbol{p}_t} \cdot \mathcal{B}^4(\boldsymbol{p}_t) = \nabla_{\boldsymbol{p}_t} \cdot \begin{bmatrix}-p_t^y & p_t^x & 0\end{bmatrix} \tag{16}$$

$$= \frac{\partial - p_t^y}{\partial p_t^x} + \frac{\partial p_t^x}{\partial p_t^y} = 0; \tag{17}$$

$$\nabla_{\boldsymbol{p}_t} \cdot \mathcal{B}^5(\boldsymbol{p}_t) = \nabla_{\boldsymbol{p}_t} \cdot \begin{bmatrix}p_t^z & 0 & -p_t^x\end{bmatrix} \tag{18}$$

$$= \frac{\partial p_t^z}{\partial p_t^x} + \frac{\partial - p_t^x}{\partial p_t^z} = 0; \tag{19}$$

$$\nabla_{\boldsymbol{p}_t} \cdot \mathcal{B}^6(\boldsymbol{p}_t) = \nabla_{\boldsymbol{p}_t} \cdot \begin{bmatrix}0 & -p_t^z & p_t^y\end{bmatrix} \tag{20}$$

$$= \frac{\partial - p_t^z}{\partial p_t^y} + \frac{\partial p_t^y}{\partial p_t^z} = 0. \tag{21}$$

B. Implementation Details

We implement our networks by MLPs, and the detailed configurations are as follows:

- f_{code}: This network includes 4 MLP layers and each hidden layer has a size of 128 neurons followed by ReLU. In addition, we make a degree of 8 positional encoding for input positions. We set the dimension of the output physics code L as 16.
- f_{neck}: This network has MLP layers as $L \mapsto 4L \mapsto 4L \mapsto K$. We choose the bottleneck vector dimension K as 32 for Dynamic Indoor Scene dataset and 16 for other two datasets.
- f_{weight}: This network includes 5 MLP layers and each hidden layer has a size of 128 neurons. In addition, we make a degree of 8 positional encoding for input positions, and we add a ResNet connection on the 3rd layer. The output vector has a dimension of $K * 6$ and is reshaped as a matrix of $K \times 6$.
- f_{deform}: For two synthetic datasets, this network includes 6 MLP layers and each hidden layer has a size of 128 neurons, while it includes 8 MLP layers and each hidden layer has a size of 256 neurons for the challenging FreeGave-GoPro dataset. In addition, we also make a degree of 8 positional encoding for input positions, and the physics code $\boldsymbol{z}$ is concatenated to the encoded position. We also add a ResNet connection on the third layer.

C. Details of Deformation-aided Optimization

We train our models on all datasets on a single NVIDIA 3090 GPU. We normalize the total time span in all datasets to be 1. Δt is set to be 1/60 in both Dynamic Object dataset and Dynamic Indoor Scene dataset, while it is set as 1/88 in the challenging FreeGave-GoPro dataset.

D. Details of All Datasets

Dynamic Object Dataset [30]: This dataset contains 6 moving objects with a white background, and the corresponding motions include: 1. part-wise rigid motions with accelerations, *i.e.*, rotating fan, freely falling basketball in a gravitational field, and rotating telescope; 2. self-propelling deformable objects, *i.e.*, a bat flapping wings, a swimming whale, and a swimming shark. Each scene contains 15 viewing angles, where the first 46 frames from 12 selected viewing angles are used as training split, *i.e.*, 552 frames in total, and the first 46 frames from the other 3 viewing angles are used for evaluating novel view interpolation within the training time period, *i.e.*, 138 frames in total. All the remaining 14 frames from 15 viewing angles are used to evaluate future frame extrapolation, *i.e.*, 210 frames.

Dynamic Indoor Scene Dataset [30]: This dataset contains 4 indoor scenes, each containing 3 to 5 moving objects, and each moving object is undergoing different rigid motions. Each scene contains 30 viewing angles, where the first 46 frames from 25 selected viewing angles are used as the training split, *i.e.*, 1150 frames in total, and the first 46 frames from the other 5 viewing angles are used for evaluating novel view interpolation within the training time period, *i.e.*, 230 frames in total. All the remaining 14 frames from 30 viewing angles are used to evaluate future frame extrapolation, *i.e.*, 450 frames.

ParticleNeRF Dataset [1]: This dataset includes 6 challenging dynamic objects.

- **Object #1: Robot**. This scene includes a robot arm waving from one side to another side.
- **Object #2: Robot Task**. This scene shows a robot arm putting a box onto a sliding platform.
- **Object #3: Wheel**. This scene includes a constant rotating wheel.
- **Object #4: Spring**. This scene shows a box tied on a spring, which undergoes a harmonic oscillation motion. The training and the test observation period in total form a whole oscillation period.
- **Object #5: Pendulums**. This scene includes two swing pendulums, each undergoing harmonic oscillation motion. The training and the test observation period in total form a whole oscillation period.
- **Object #6: Cloth**. This scene includes a flat cloth being folded.

Each scene contains 40 viewing angles. For Object #1 & #2, we choose the first 53 frames from 36 selected viewing angles as the training split, *i.e.* 1908 frames in total, and the first 53 frames from the other 4 viewing angles for evaluating novel view interpolation within the training time period, *i.e.*, 212 frames in total. All the remaining 17 frames from 40 viewing angles are used to evaluate future frame extrapolation, *i.e.*, 680 frames. For Object #3 & #4 & #5 & #6, the first 104 frames from 36 selected viewing angles are used as the training split, *i.e.*, 3744 frames in total, and the first 104 frames from the other 4 viewing angles are used for evaluating novel view interpolation within the training time period, *i.e.*, 416 frames in total. All the remaining 36 frames from 40 viewing angles are used to evaluate future frame extrapolation, *i.e.*, 1440 frames.

FreeGave-GoPro Dataset: This dataset includes 6 challenging real-world dynamic scenes.

- **Scenes #1/#2: Pen & Tape**. There is a person holding a pen and trying to pass it through the hole of a static tape. The difference between these two scenes is that there are more static objects in the second scene, introducing more visual occlusions and requiring more accurate separation between moving areas and static areas. The difficulty lies in the motion of one object which is going to penetrate through another in future.
- **Scene #3: Box**. This scene contains a drawer-like box, and a person is trying to close it. The difficulty lies in a tight combination of the moving part and the static part of the box, especially in future.
- **Scene #4: Hammer**. This scene contains a hammer moving on the topside of a box. The difficulty lies in the direct contact of moving objects and static objects, which requires sharp separation of diverse motion patterns in order to keep right static/moving states in future.
- **Scene #5: Collision**. This scene contains a cube and a cup moving towards each other. The difficulty is the different directions of two motions. It is hard to keep the shapes of these two objects in future.
- **Scene #6: Wrist Rest**. A person is trying to bend a wrist rest. The difficulty is that the object is deformable and the motion is thus not rigid or part-wise rigid.

E. Quantitative Results on NVIDIA Dynamic Scene Dataset

Though not primarily collected for physics learning, we also evaluate our model on two simple scenes from NVIDIA Dynamic Scene Dataset [79] selected by NVFi [30]. The quantitative results are shown in Table 5.

Table 5. Results on NVIDIA Dynamic Scene Dataset.

	Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
T-NeRF	23.078	0.684	0.355	21.120	0.707	0.358
D-NeRF	22.827	0.711	0.309	20.633	0.709	0.327
TiNeuVox	**28.304**	0.868	0.216	24.556	0.863	0.215
T-NeRF$_{PINN}$	18.443	0.597	0.439	17.975	0.605	0.428
HexPlane$_{PINN}$	24.971	0.818	0.281	24.473	0.818	0.279
NVFi	27.138	0.844	0.231	<u>28.462</u>	0.876	0.214
DefGS	26.662	<u>0.893</u>	<u>0.127</u>	24.240	0.895	0.140
DefGS$_{nvfi}$	26.972	0.890	0.128	27.529	<u>0.927</u>	<u>0.102</u>
FreeGave (Ours)	<u>27.345</u>	**0.896**	**0.097**	**29.005**	**0.933**	**0.072**

F. Quantitative Results on Collision Cases

We evaluate on two more scenes with collisions: We extend the *dining* scene of Dynamic Indoor Scene Dataset into two collision cases with different collision patterns. The first scene has 28 frames $\times$ 25 views for training without observing collision, 28 frames $\times$ 5 views for interpolation, and 8 $\times$ 30 views for extrapolation where the collision happens. The second scene has 46 frames $\times$ 25 views for training with the collision observed, 46 frames $\times$ 5 views for interpolation, and 14 $\times$ 30 views for extrapolation.

Table 6. Results on four scenes of oscillations or collistions.

	Collisions					
	Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TiNeuVox	23.429	0.794	0.277	20.794	0.807	0.250
NVFi	20.301	0.690	0.413	22.917	0.780	0.313
DefGS	29.411	0.894	0.117	23.129	0.867	0.122
DefGS$_{nvfi}$	29.424	0.894	0.118	28.017	0.907	0.081
FreeGave (Ours)	**29.971**	**0.916**	**0.074**	**28.426**	**0.912**	**0.058**

G. Analysis of Computational Costs

We calculate the average time and memory consumption in training, the average speed and memory consumption in test, and model sizes for most baselines in Table 7.

We can see that: 1) our method has a clear advantage over the strong baseline DefGS$_{nvfi}$ in terms of time and memory cost in training, thanks to our new and more efficient divergence-free velocity module over the PINN loss;

2) our method is generally better or on par with other baselines (TiNeuVox / NVFi / DefGS) in computation cost of training and test, but our method demonstrates significantly better extrapolation results (as shown in Tables 1&2).

H. Limitation of Our Model

The main limitation is that our method would fail to predict abrupt motions, such as an explosion, primarily because the underlying physics rules are unable to be observed or learned from visual frames.

I. Details of Segmenting Motion Patterns

For our models and the DefGS / DefGS$_{nvfi}$ baselines, we render the segmentation masks after grouping all learned Gaussian kernels. We follow the rendering module in Gaussian-Grouping [78] to obtain segmentation masks. Gaussian-Grouping renders hidden segmentation features in a size of 16. Therefore, we directly expand our one-hot object group into 16 channels and then render masks. More details are as follows.

I.1. More Details of Our FreeGave

We segment our well-trained Gaussian kernels by their bottleneck vectors $\boldsymbol{h} = f_{neck}(\boldsymbol{z})$. To be specific, we build a grouping feature vector for each Gaussian kernel as $\boldsymbol{h} \oplus \lambda \boldsymbol{p}_0$, where $\oplus$ means concatenation and λ is a hyperparameter, working as smoothing regularization. Then the Gaussian kernels are simply grouped by K-means algorithm with respect to this built features into C groups.

We choose λ as 0 for Genome House and Chessboard scene, and 0.5 for Factory and Dining Table scene. C is set as 13 for Dining Table scene and 8 for other three scenes.

I.2. More Details of Segmenting DefGS / DefGS$_{nvfi}$

Given a well-trained DefGS or DefGS$_{nvfi}$ model with N canonical Gaussian kernels, we first assign learnable per-Gaussian object codes $\boldsymbol{O} \in (0,1)^{N\times K}$ to all Gaussian kernels, where K is the maximum number of objects that is expected to appear in the scene.

After that, we query the position displacements for Gaussians kernels from the well-trained deformation field at time 0 and t respectively, thus obtaining the Gaussians $\boldsymbol{P}_0$ at time 0 and $\boldsymbol{P}_t$ at time t. Then the per-Gaussian scene flows $\boldsymbol{M}_t$ from time 0 to t is calculated as $\boldsymbol{M}_t = \boldsymbol{P}_t - \boldsymbol{P}_0$.

Lastly, two losses proposed in OGC [57] are computed on the learnable object codes. **1) Dynamic rigid consistency:** For the k^{th} object, we first retrieve its (soft) binary mask $\boldsymbol{O}^k$, and feed the tuple $\{\boldsymbol{P}_0, \boldsymbol{P}_t, \boldsymbol{O}^k\}$ into the weighted-Kabsch algorithm to estimate its transformation matrix $\boldsymbol{T}_k \in \mathbb{R}^{4\times 4}$ belonging to $SE(3)$ group. Then the dynamic loss is computed as:

$$\ell_{dynamic} = \frac{1}{N}\sum_{\boldsymbol{p}\in\boldsymbol{P}_0}\left\|\Big(\sum_{k=1}^{K} o_{\boldsymbol{p}}^{k}\cdot(\boldsymbol{T}_k \circ \boldsymbol{p})\Big) - (\boldsymbol{p}+\boldsymbol{m}_t)\right\|_2$$

where $o_{\boldsymbol{p}}^k$ represents the probability of being assigned to the k^{th} object for a specific point $\boldsymbol{p}$, and $\boldsymbol{m}_t \in \mathbb{R}^3$ represents the motion vector of $\boldsymbol{p}$ from time 0 to t. The operation $\circ$ applies the rigid transformation to the point. This loss aims to discriminate objects with different motions. **2) Spatial smoothness:** For each point $\boldsymbol{p}$ in $\boldsymbol{P}_0$, we first search H nearest neighboring points. Then the smoothness loss is defined as:

$$\ell_{smooth} = \frac{1}{N}\sum_{\boldsymbol{p}\in\boldsymbol{P}_0}\Big(\frac{1}{H}\sum_{h=1}^{H}\|\boldsymbol{o}_{\boldsymbol{p}} - \boldsymbol{o}_{\boldsymbol{p}_h}\|_1\Big) \tag{22}$$

where $\boldsymbol{o}_{\boldsymbol{p}} \in (0,1)^K$ represents the object assignment of center point $\boldsymbol{p}$, and $\boldsymbol{o}_{\boldsymbol{p}_h} \in (0,1)^K$ represents the object assignment of its h^{th} neighbouring point. This loss aims to avoid the over-segmentation issues. More details are provided in [57].

In our experiments for DefGS and DefGS$_{nvfi}$, the maximum number of predicted objects K is set to be 8. A softmax activation is applied on per-Gaussian object codes. During optimization, we adopt the Adam optimizer with a learning rate of 0.01 and optimize object codes for 1000 iterations until convergence.

All quantitative results for scene decomposition are in Table 8.

J. More Results of Ablation Study

We report all ablation results in Table 9. We conduct all ablations at a setting of $K = 16$ originally, while we find $K = 32$ is slightly better on Dynamic Indoor Scene dataset. Nevertheless, this does not influence the analysis to the influencing factors as shown in the main paper.

K. More Results on Dynamic Object Dataset

The quantitative results for each scene of Dynamic Object Dataset are in Table 10.

L. More Results on Dynamic Indoor Scene Dataset

The quantitative results for each scene of Dynamic Indoor Scene Dataset are in Table 11.

M. More Results on FreeGave-GoPro Dataset

The quantitative results for each scene of FreeGave-GoPro Dataset are in Table 12.

Table 7. The average time (hours) and GPU memory (GB) cost for training, the inference speed (fps), GPU memory (GB), and model size (MB) for test on all three datasets.

	Dynamic Object Dataset					Dynamic Indoor Scene Dataset					FreeGave-GoPro Dataset				
	Training		Test			Training		Test			Training		Test		
	Time↓	Mem↓	fps↑	Mem↓	Size↓	Time↓	Mem↓	fps↑	Mem↓	Size↓	Time↓	Mem↓	fps↑	Mem↓	Size↓
TiNeuVox	**0.5**	8.0	0.40	2.3	49.8	**0.6**	8.2	0.51	3.7	**49.9**	**0.6**	9.4	0.16	4.7	**49.6**
NVFi	2.2	22.6	0.11	16.5	114.7	2.3	21.5	0.34	16.8	107.9	2.3	23.3	0.03	23.1	121.4
DefGS	0.8	6.4	19.40	5.0	53.1	0.8	**5.9**	**21.9**	4.0	98.1	1.3	**8.4**	**3.22**	**3.4**	88.5
DefGS$_{nvfi}$	2.1	22.7	13.59	5.2	54.5	6.0	26.4	10.98	4.1	101.2	8.0	32.6	2.90	5.8	92.1
FreeGave (Ours)	0.8	**5.7**	**19.70**	**4.0**	**27.0**	1.5	10.4	13.60	**3.1**	55.8	1.9	16.1	3.03	8.1	115.7

Table 8. Quantitative results of scene decomposition on the Synthetic Indoor Scene dataset.

	Gnome House						Chessboard					
	AP↑	PQ↑	F1↑	Pre↑	Rec↑	mIoU↑	AP↑	PQ↑	F1↑	Pre↑	Rec↑	mIoU↑
Mask2Former [11]	60.89	73.05	77.32	85.32	70.69	66.94	82.68	81.35	90.81	97.54	84.94	76.17
D-NeRF [51]	80.54	62.24	85.28	85.28	85.28	54.82	57.12	48.11	60.22	56.20	64.85	48.97
NVFi [30]	**100.00**	85.01	**100.00**	**100.00**	**100.00**	68.01	67.97	57.95	76.96	76.96	76.96	56.79
DefGS [75]	86.67	86.04	91.44	85.91	97.74	74.21	42.90	49.48	60.75	56.53	65.64	50.29
DefGS$_{nvfi}$	99.12	96.02	99.17	98.36	**100.00**	77.45	31.27	47.55	54.87	56.87	53.01	44.41
FreeGave (Ours)	**100.00**	**97.59**	**100.00**	**100.00**	**100.00**	**78.07**	**100.00**	**92.83**	**100.00**	**100.00**	**100.00**	**79.57**
	Dining Table						Factory					
	AP↑	PQ↑	F1↑	Pre↑	Rec↑	mIoU↑	AP↑	PQ↑	F1↑	Pre↑	Rec↑	mIoU↑
Mask2Former [11]	77.65	84.61	87.42	97.44	79.28	76.80	40.25	53.54	57.60	99.01	40.61	37.76
D-NeRF [51]	74.05	57.15	69.3	59.35	83.27	61.82	17.33	17.08	21.29	25.35	18.35	20.72
NVFi [30]	98.01	91.81	98.95	98.99	98.92	76.68	98.86	80.17	99.09	99.09	99.09	69.07
DefGS [75]	57.66	62.92	70.51	69.12	71.95	55.73	19.69	31.96	43.02	41.27	44.94	37.59
DefGS$_{nvfi}$	67.02	72.12	78.37	64.85	99.01	76.19	23.64	35.30	46.90	57.49	39.60	29.23
FreeGave (Ours)	**98.99**	**98.36**	**99.97**	**99.98**	**99.97**	**81.89**	**100.00**	**96.31**	**100.00**	**100.00**	**100.00**	**82.55**

N. More Results on ParticleNeRF Dataset

The quantitative results for each scene of ParticleNeRF Dataset are in Table 13.

O. Additional Qualitative Results

We present additional qualitative results for future frame extrapolation in Figures 7, 8, 9, 10, 11, 12, 13, 14, and 15. We also present additional qualitative results for scene decomposition in Figures 16, 17, 18, and 19.

Table 9. Complete ablation study results on both Dynamic Object Dataset and Dynamic Indoor Scene Dataset

					Dynamic Object Dataset					
					Interpolation			Extrapolation		
	Code $\boldsymbol{z}$	f_{deform}	$\boldsymbol{v}(\boldsymbol{p}_t, t)$	K	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
(1)	learnable	full	full	16	38.722	0.995	0.005	25.961	0.975	0.025
(2)	field	full	w/o $\mathcal{B}(\boldsymbol{p}_t)$	16	39.126	0.995	0.005	29.400	0.986	0.010
(3)	field	full	w/o decomp	16	39.111	0.995	0.005	29.432	0.985	0.009
(4)	field	full	full	8	39.324	**0.996**	**0.004**	30.972	0.989	0.009
(4)	field	full	full	32	39.318	**0.996**	**0.004**	31.438	**0.990**	**0.007**
(5)	field	✗	full	16	20.974	0.945	0.068	17.927	0.922	0.088
(6)	field	w/o $\boldsymbol{z}$	full	16	39.151	0.995	0.005	31.217	0.988	0.009
(7)	field	w/o $\delta\boldsymbol{s}$	full	16	39.191	0.995	0.005	31.704	**0.990**	**0.007**
FreeGave	field	full	full	16	**39.393**	0.995	0.005	**31.987**	**0.990**	**0.007**
					Dynamic Indoor Scene Dataset					
					Interpolation			Extrapolation		
	Code $\boldsymbol{z}$	f_{deform}	$\boldsymbol{v}(\boldsymbol{p}_t, t)$	K	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
(1)	learnable	full	full	16	**32.343**	**0.930**	**0.091**	30.444	0.933	0.087
(2)	field	full	w/o $\mathcal{B}(\boldsymbol{p}_t)$	16	31.471	0.921	0.108	32.316	0.944	0.077
(3)	field	full	w/o decomp	16	31.707	0.921	0.106	31.204	0.943	0.071
(4)	field	full	full	8	32.005	0.929	0.093	34.159	0.962	0.053
(4)	field	full	full	16	31.996	0.929	0.092	34.716	0.965	**0.051**
(5)	field	✗	full	16	-	-	-	-	-	-
(6)	field	w/o $\boldsymbol{z}$	full	16	31.603	0.921	0.107	33.408	0.955	0.067
(7)	field	w/o $\delta\boldsymbol{s}$	full	16	32.094	0.929	0.092	34.504	0.964	0.052
FreeGave	field	full	full	32	32.287	**0.930**	0.092	**35.019**	**0.966**	**0.051**

Table 10. Per-scene quantitative results on Dynamic Object Dataset.

Methods	Falling Ball						Bat					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
T-NeRF [51]	14.921	0.782	0.326	15.418	0.793	0.308	13.070	0.836	0.234	13.897	0.834	0.230
D-NeRF [51]	15.548	0.665	0.435	15.116	0.644	0.427	14.087	0.845	0.212	15.406	0.887	0.175
TiNeuVox [15]	35.458	0.974	0.052	20.242	0.959	0.067	16.080	0.908	0.108	16.952	0.930	0.115
T-NeRF$_{PINN}$	17.687	0.775	0.368	17.857	0.829	0.265	16.412	0.903	0.197	18.983	0.930	0.132
HexPlane$_{PINN}$	32.144	0.965	0.065	20.762	0.951	0.081	23.399	0.958	0.057	21.144	0.951	0.064
NVFi [30]	35.826	0.978	0.041	31.369	0.978	0.041	23.325	0.964	0.046	25.015	0.968	0.042
DefGS [75]	37.535	0.995	0.009	20.442	0.976	0.033	38.750	**0.997**	0.004	17.063	0.936	0.072
DefGS$_{nvfi}$	38.606	0.996	0.010	24.873	0.985	0.015	38.075	**0.997**	0.004	**28.950**	0.980	**0.015**
FreeGave (Ours)	**42.369**	**0.998**	**0.003**	**38.321**	**0.997**	**0.003**	**39.662**	**0.997**	**0.002**	27.235	**0.982**	0.013

Methods	Fan						Telescope					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
T-NeRF [51]	8.001	0.308	0.646	8.494	0.392	0.593	13.031	0.615	0.472	13.892	0.670	0.417
D-NeRF [51]	7.915	0.262	0.690	8.624	0.370	0.623	13.295	0.609	0.469	14.967	0.700	0.385
TiNeuVox [15]	24.088	0.930	0.104	20.932	0.935	0.078	31.666	0.982	0.041	20.456	0.921	0.067
T-NeRF$_{PINN}$	9.233	0.541	0.508	9.828	0.606	0.443	14.293	0.739	0.366	15.752	0.804	0.298
HexPlane$_{PINN}$	22.822	0.921	0.079	19.724	0.919	0.080	25.381	0.948	0.066	23.165	0.932	0.074
NVFi [30]	25.213	0.948	0.049	27.172	0.963	0.037	26.487	0.959	0.048	27.101	0.963	0.046
DefGS [75]	**35.858**	**0.985**	0.017	20.932	0.948	0.038	37.502	0.996	0.003	20.684	0.927	0.048
DefGS$_{nvfi}$	35.217	0.984	0.019	26.648	0.972	0.023	37.568	0.996	0.003	34.096	0.994	0.005
FreeGave (Ours)	35.767	**0.985**	**0.013**	**32.393**	**0.986**	**0.009**	**40.332**	**0.998**	**0.002**	**40.401**	**0.998**	**0.002**

Methods	Shark						Whale					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
T-NeRF [51]	13.813	0.853	0.223	15.325	0.882	0.193	16.141	0.860	0.212	15.880	0.860	0.203
D-NeRF [51]	17.727	0.903	0.150	19.078	0.936	0.092	16.373	0.898	0.154	14.771	0.883	0.171
TiNeuVox [15]	23.178	0.971	0.059	19.463	0.950	0.050	37.455	0.994	0.016	19.624	0.943	0.063
T-NeRF$_{PINN}$	17.315	0.878	0.177	18.739	0.921	0.115	16.778	0.927	0.141	15.974	0.919	0.127
HexPlane$_{PINN}$	28.874	0.976	0.040	22.330	0.961	0.047	29.634	0.981	0.035	21.391	0.961	0.053
NVFi [30]	32.072	0.984	0.024	28.874	0.982	0.021	31.240	0.986	0.025	26.032	0.978	0.029
DefGS [75]	37.802	0.994	0.006	19.924	0.957	0.034	**39.740**	**0.997**	0.004	20.048	0.951	0.046
DefGS$_{nvfi}$	37.327	0.994	0.006	**29.240**	0.987	0.007	37.101	0.996	0.005	28.686	0.986	0.012
FreeGave (Ours)	**40.211**	**0.996**	**0.004**	29.236	**0.990**	**0.005**	38.015	**0.997**	**0.003**	**28.950**	**0.989**	**0.009**

Table 11. Per-scene quantitative results on Dynamic Indoor Scene Dataset.

Methods	Gnome House						Chessboard					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
T-NeRF [51]	26.094	0.716	0.383	23.485	0.643	0.419	25.517	0.796	0.294	20.228	0.708	0.365
D-NeRF [51]	27.000	0.745	0.319	21.714	0.641	0.367	24.852	0.774	0.308	19.455	0.675	0.384
TiNeuVox [15]	30.646	0.831	0.253	21.418	0.699	0.326	33.001	0.917	0.177	19.718	0.765	0.310
T-NeRF$_{PINN}$	15.008	0.375	0.668	16.200	0.409	0.651	16.549	0.457	0.621	17.197	0.472	0.618
HexPlane$_{PINN}$	23.764	0.658	0.510	22.867	0.658	0.510	24.605	0.778	0.412	21.518	0.748	0.428
NSFF [32]	31.418	0.821	0.294	25.892	0.750	0.327	32.514	0.810	0.201	21.501	0.805	0.282
NVFi [30]	30.667	0.824	0.277	30.408	0.826	0.273	30.394	0.888	0.215	27.840	0.872	0.219
DefGS [75]	32.041	0.918	0.132	21.703	0.775	0.207	27.355	0.912	0.147	20.032	0.808	0.218
DefGS$_{nvfi}$	**32.881**	0.919	0.132	33.630	0.953	0.077	26.200	0.907	0.156	26.730	0.917	0.110
FreeGave (Ours)	32.791	**0.923**	**0.103**	**36.458**	**0.963**	**0.062**	**35.388**	**0.962**	**0.061**	**35.016**	**0.970**	**0.044**
Methods	Factory						Dining Table					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
T-NeRF [51]	26.467	0.741	0.328	24.276	0.722	0.344	21.699	0.716	0.338	20.977	0.725	0.324
D-NeRF [51]	28.818	0.818	0.252	22.959	0.746	0.303	20.851	0.725	0.319	19.035	0.705	0.341
TiNeuVox [15]	32.684	0.909	0.148	22.622	0.810	0.229	23.596	0.798	0.274	20.357	0.804	0.258
T-NeRF$_{PINN}$	16.634	0.446	0.624	17.546	0.480	0.609	16.807	0.486	0.640	18.215	0.548	0.595
HexPlane$_{PINN}$	27.200	0.826	0.283	24.998	0.792	0.312	25.291	0.788	0.350	22.979	0.771	0.355
NSFF [32]	**33.975**	0.919	0.152	26.647	0.855	0.196	19.552	0.665	0.464	22.612	0.770	0.351
NVFi [30]	32.460	0.912	0.151	31.719	0.908	0.154	**29.179**	0.885	0.199	29.011	0.898	0.171
DefGS [75]	33.629	**0.943**	0.096	22.820	0.839	0.169	27.680	0.890	0.145	20.965	0.855	0.157
DefGS$_{nvfi}$	33.643	**0.943**	0.097	33.049	0.954	0.062	27.957	0.891	0.145	30.975	0.955	0.060
FreeGave (Ours)	33.316	**0.943**	**0.079**	**35.765**	**0.966**	**0.048**	27.652	**0.892**	**0.124**	**32.838**	**0.963**	**0.048**

Table 12. Per-scene quantitative results on FreeGave-GoPro Dataset.

Methods	Pen & Tape 1						Pen & Tape 2					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TiNeuVox [15]	19.368	0.758	0.304	20.127	0.795	0.289	19.594	0.732	0.334	20.514	0.779	0.296
NVFi [30]	21.397	0.816	0.243	23.869	0.824	0.258	22.31	0.813	0.245	23.574	0.806	0.269
DefGS [75]	**29.598**	**0.933**	**0.080**	20.284	0.865	0.163	**27.587**	**0.909**	**0.098**	20.674	0.861	0.169
DefGS$_{nvfi}$	29.571	0.932	0.081	26.289	0.922	0.108	27.456	**0.909**	0.099	27.124	0.915	0.120
FreeGave (Ours)	29.412	0.927	0.087	**29.001**	**0.936**	**0.090**	27.498	0.907	**0.098**	**28.842**	**0.925**	**0.107**
Methods	Box						Wrist Rest					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TiNeuVox [15]	19.464	0.726	0.318	23.958	0.807	0.247	19.133	0.751	0.315	18.204	0.727	0.342
NVFi [30]	19.391	0.777	0.282	24.867	0.806	0.263	13.235	0.570	0.490	19.222	0.683	0.431
DefGS [75]	28.448	0.918	0.087	25.656	0.904	0.117	28.178	0.923	0.100	18.834	0.809	0.220
DefGS$_{nvfi}$	**29.571**	**0.932**	**0.081**	26.289	0.922	0.108	27.938	0.921	0.102	22.741	0.856	0.170
FreeGave (Ours)	28.339	0.916	0.088	**30.964**	**0.935**	**0.084**	**28.708**	**0.925**	**0.098**	**24.093**	**0.867**	**0.159**
Methods	Hammer						Collision					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TiNeuVox [15]	18.75	0.733	0.324	22.638	0.710	0.251	17.848	0.741	0.316	18.158	0.743	0.329
NVFi [30]	22.817	0.817	0.235	25.526	0.830	0.241	14.530	0.638	0.438	19.422	0.717	0.391
DefGS [75]	28.141	0.916	**0.089**	23.995	0.899	0.123	**28.493**	0.923	0.089	18.619	0.808	0.228
DefGS$_{nvfi}$	**28.478**	**0.917**	0.088	29.392	0.928	0.095	28.125	0.923	0.092	23.512	0.871	0.157
FreeGave (Ours)	28.314	**0.917**	**0.089**	**30.090**	**0.932**	**0.091**	28.434	**0.925**	**0.088**	**25.571**	**0.886**	**0.139**

Table 13. Per-scene quantitative results on ParticleNeRF Dataset.

Methods	Robot						Robot Task					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TiNeuVox [15]	32.079	0.975	0.063	17.287	0.861	0.162	34.672	0.984	0.047	21.078	0.906	0.097
NVFi [30]	28.740	0.962	0.065	18.518	0.875	0.125	30.906	0.971	0.051	26.130	0.945	0.067
DefGS [75]	34.713	**0.989**	**0.015**	15.793	0.872	0.129	37.218	**0.994**	**0.006**	19.193	0.911	0.079
DefGS$_{nvfi}$	**33.924**	0.987	0.017	17.965	0.892	0.092	**37.640**	**0.994**	**0.006**	**26.566**	**0.962**	**0.023**
FreeGave (Ours)	33.298	0.986	0.017	**19.361**	**0.901**	**0.076**	37.538	**0.994**	**0.006**	25.526	0.954	0.029

Methods	Cloth						Wheel					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TiNeuVox [15]	32.406	0.981	0.052	18.476	0.885	0.117	28.544	0.946	0.058	22.599	0.880	0.079
NVFi [30]	27.309	0.951	0.075	18.904	0.894	0.116	26.225	0.935	0.056	12.990	0.790	0.153
DefGS [75]	**34.072**	**0.991**	**0.010**	16.687	0.880	0.115	30.290	**0.971**	**0.028**	25.840	0.945	0.034
DefGS$_{nvfi}$	32.547	0.986	0.012	26.655	0.964	**0.023**	28.537	0.968	0.029	22.393	0.914	0.063
FreeGave (Ours)	32.604	0.987	0.011	**27.934**	**0.966**	0.026	**30.350**	**0.971**	**0.028**	**30.926**	**0.972**	**0.022**

Methods	Spring						Pendulums					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
TiNeuVox [15]	32.731	0.990	0.022	20.448	0.891	0.073	36.093	0.991	0.028	22.551	0.905	0.084
NVFi [30]	30.315	0.982	0.020	15.575	0.853	0.107	29.691	0.970	0.046	16.922	0.844	0.146
DefGS [75]	35.684	0.995	0.004	19.286	0.905	0.060	**39.392**	**0.997**	**0.003**	18.428	0.889	0.082
DefGS$_{nvfi}$	34.606	0.995	0.004	23.648	0.953	0.024	37.973	0.996	0.004	19.154	0.903	0.075
FreeGave (Ours)	**38.465**	**0.997**	**0.003**	**25.501**	**0.959**	**0.015**	38.992	**0.997**	**0.003**	**30.696**	**0.985**	**0.009**

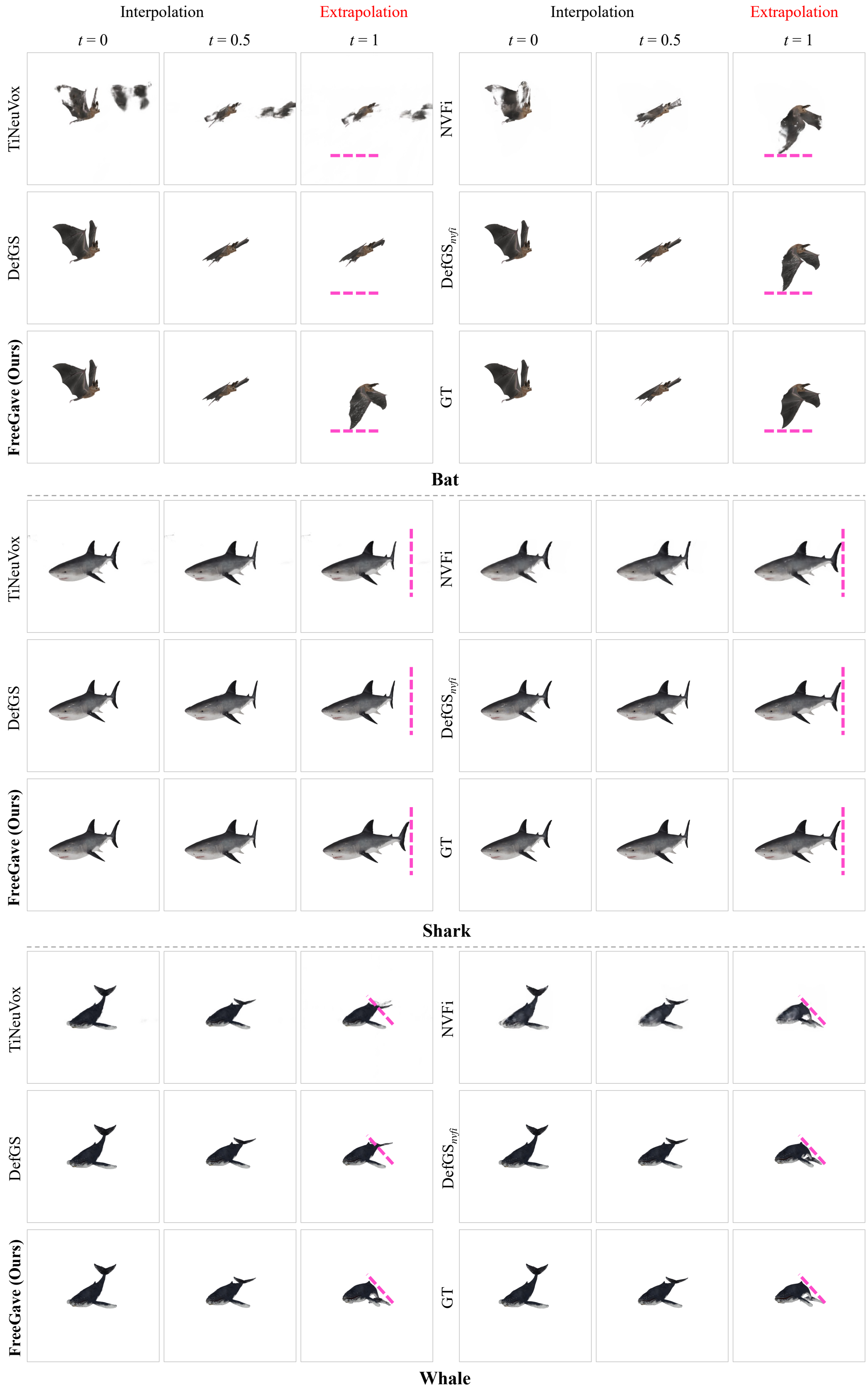

Figure 7. Qualitative results for future frame extrapolation on Dynamic Object Dataset.

Figure 8. Qualitative results for future frame extrapolation on Dynamic Object Dataset.

Figure 9. Qualitative results for future frame extrapolation on ParticleNeRF Dataset.

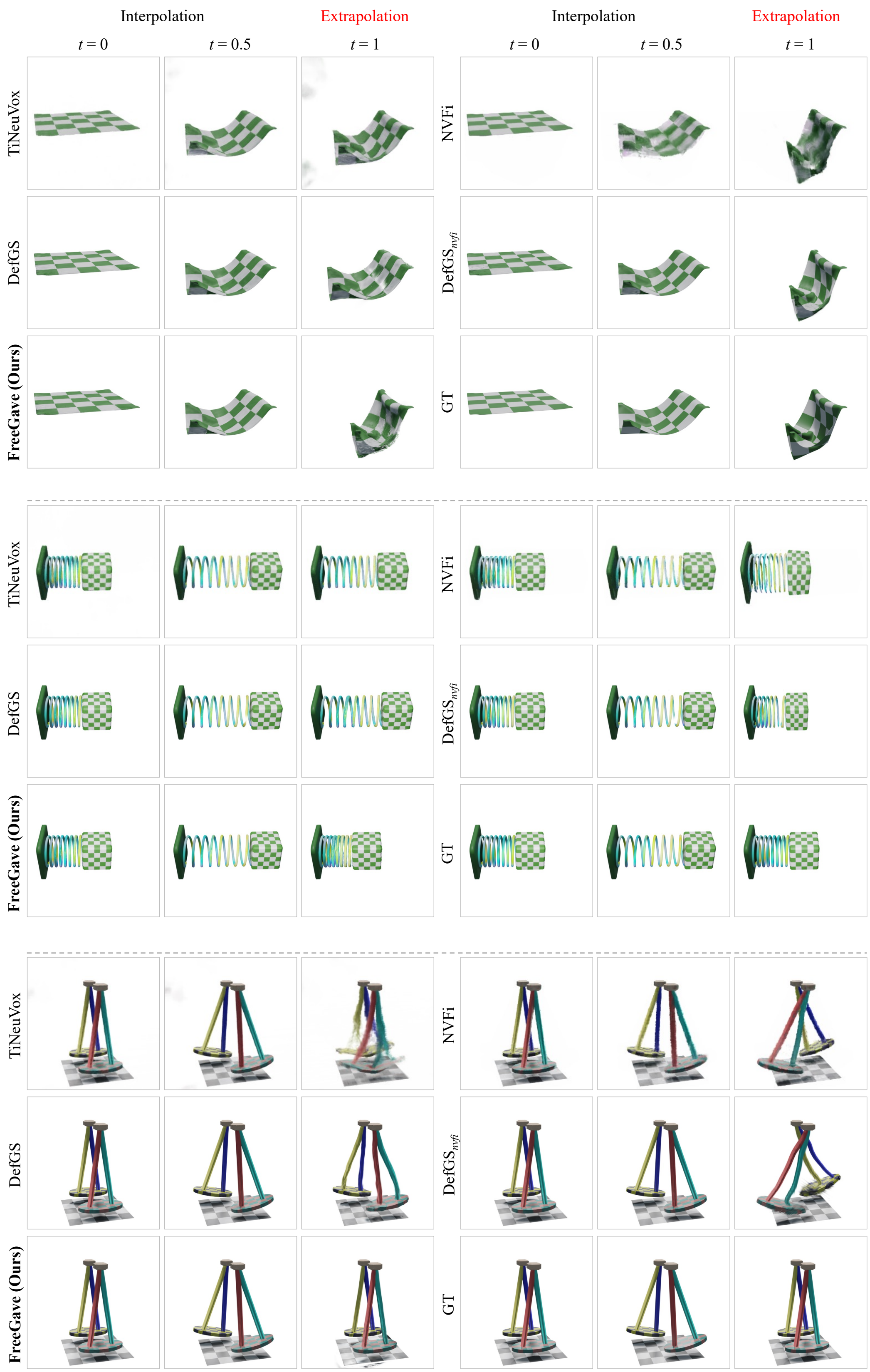

Figure 10. Qualitative results for future frame extrapolation on ParticleNeRF Dataset.

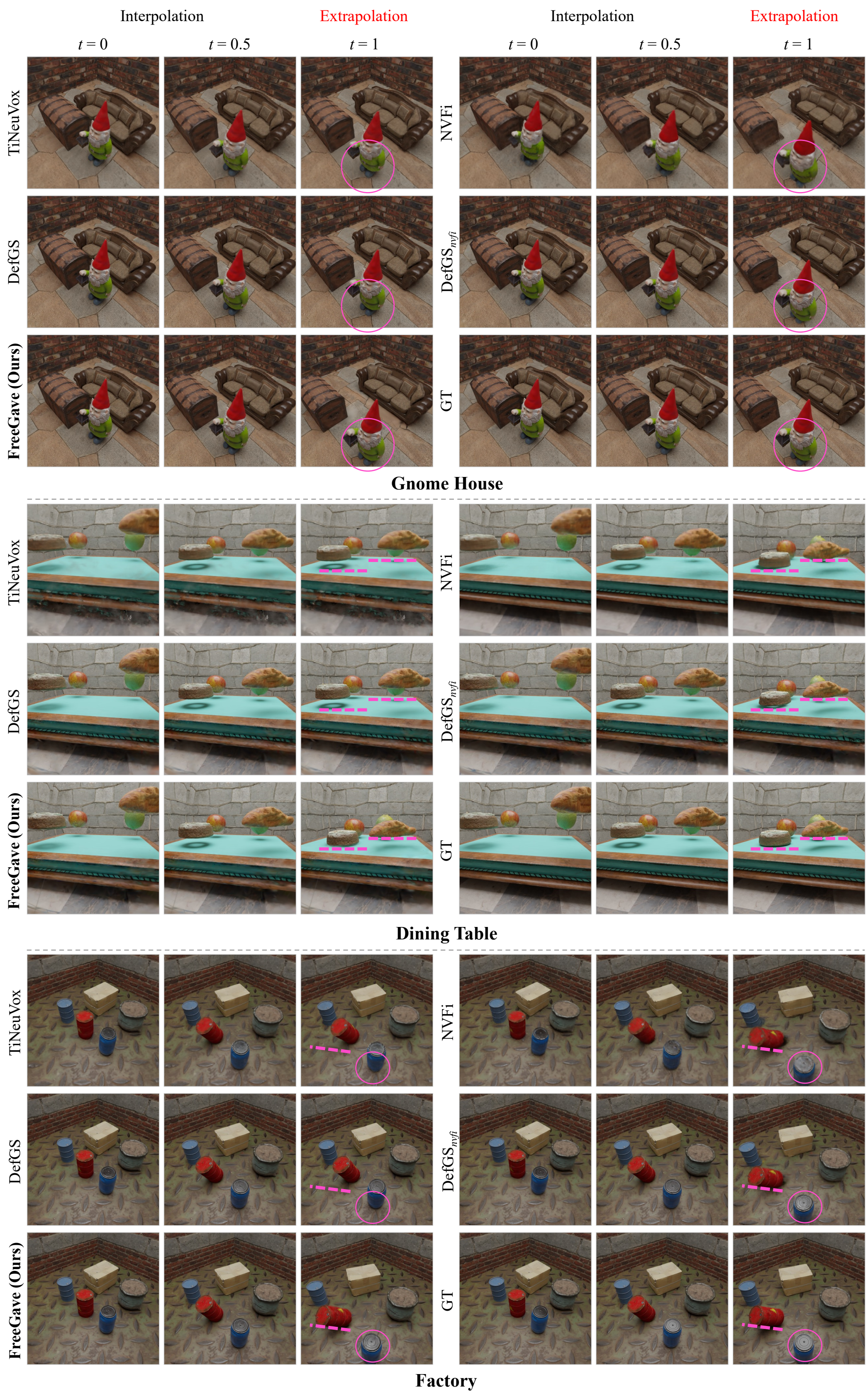

Figure 11. Qualitative results for future frame extrapolation on Dynamic Indoor Scene Dataset.

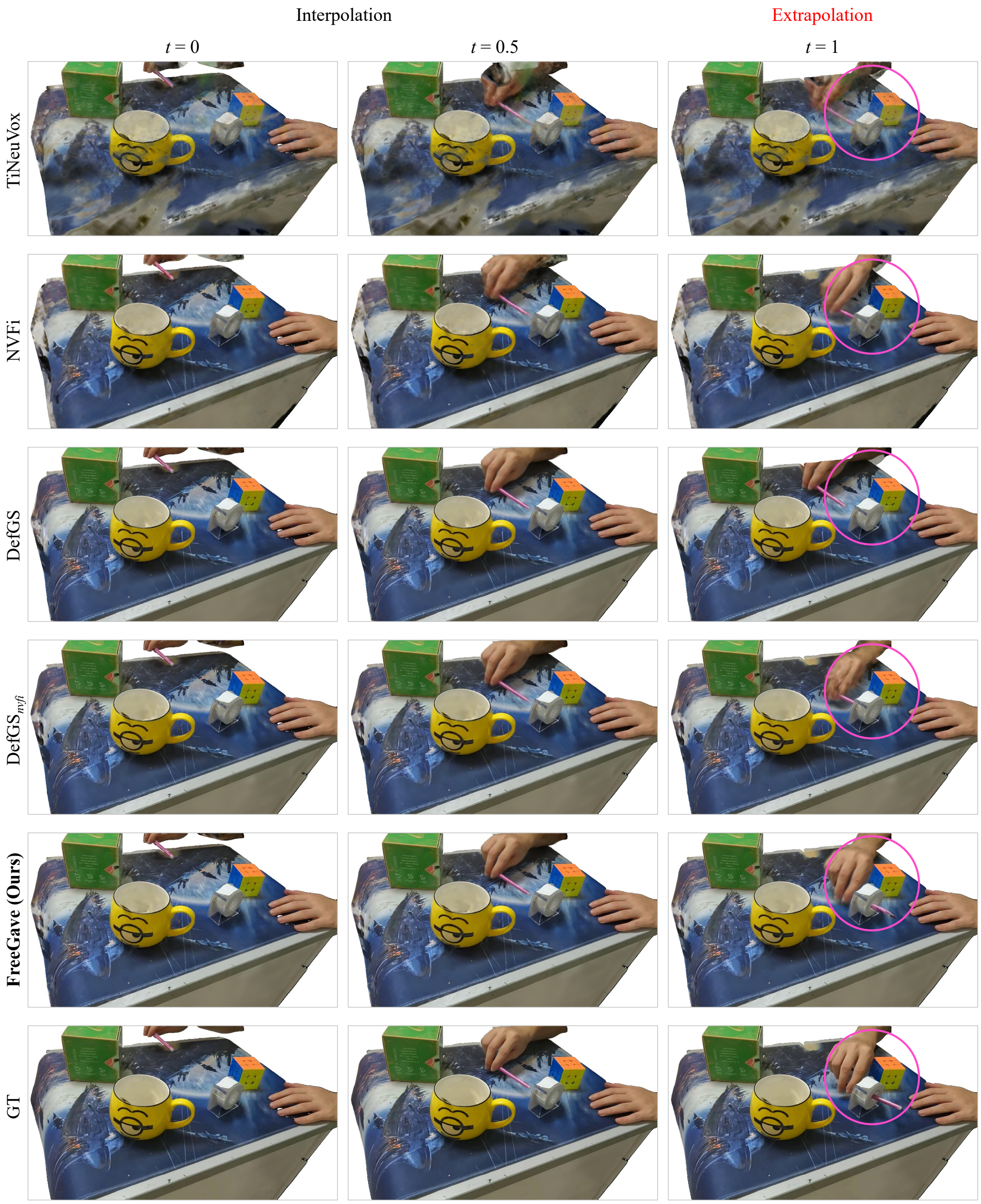

Figure 12. Qualitative results for future frame extrapolation on “Pen & Tape 2” of FreeGave-GoPro Dataset.

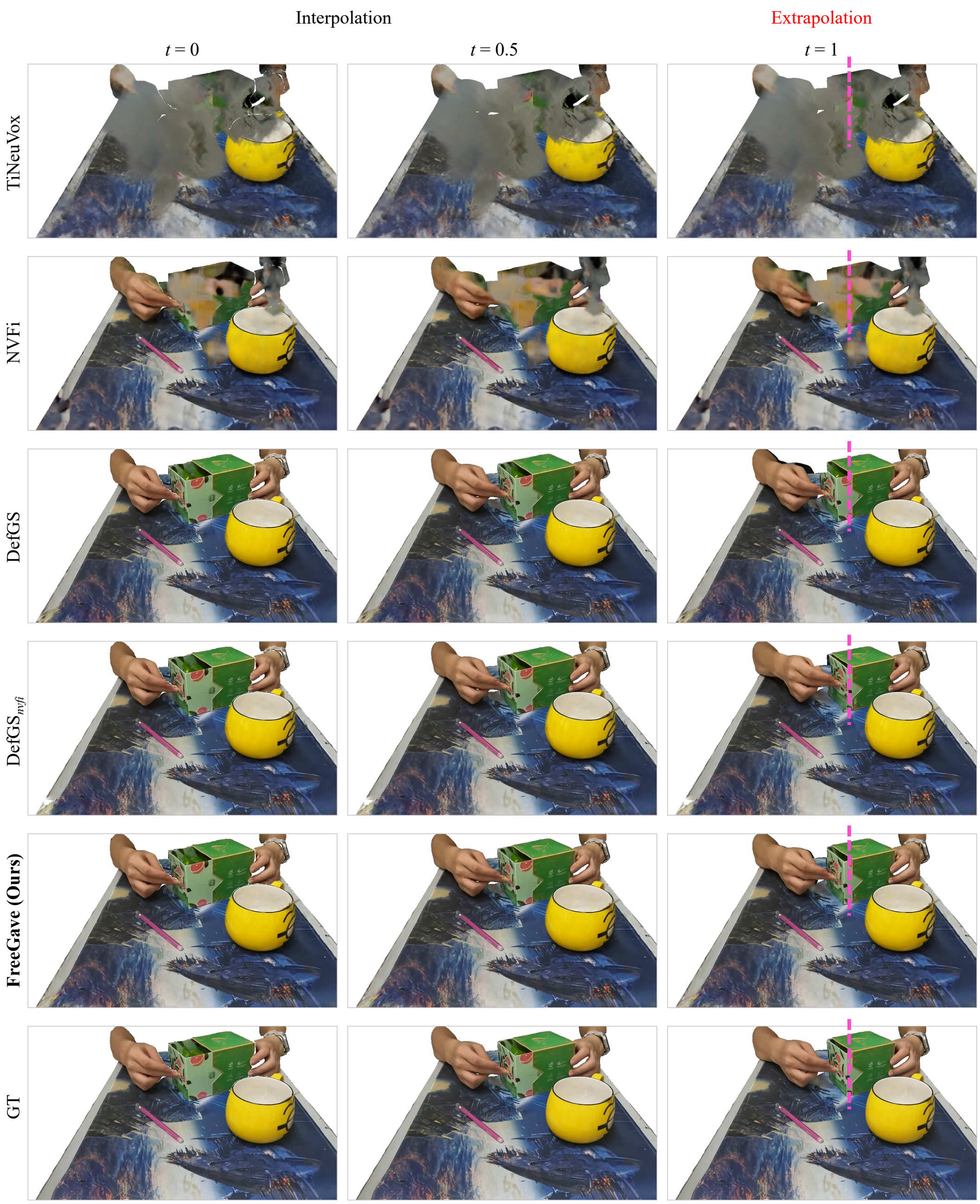

Figure 13. Qualitative results for future frame extrapolation on “Box” of FreeGave-GoPro Dataset.

Figure 14. Qualitative results for future frame extrapolation on “Hammer” of FreeGave-GoPro Dataset.

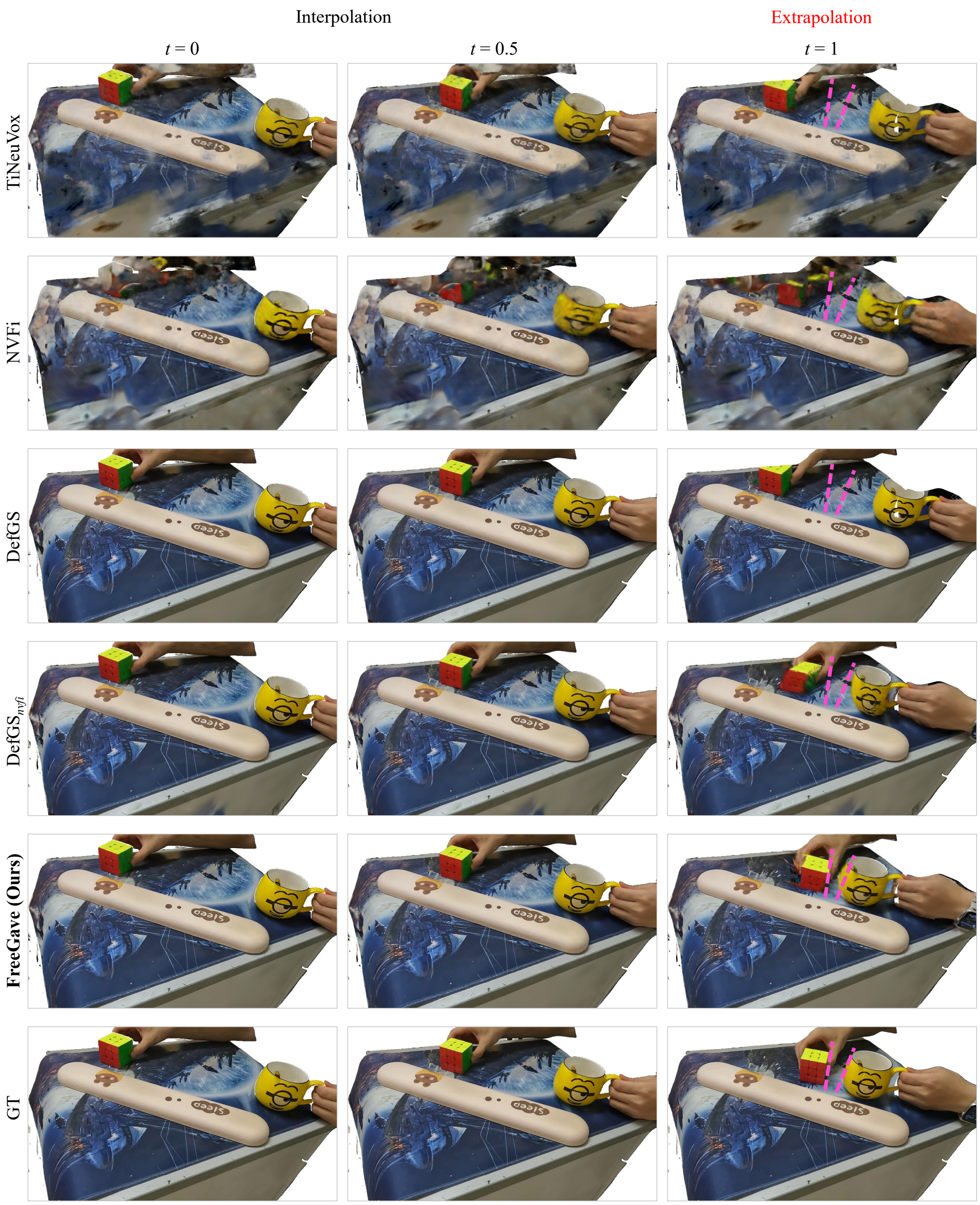

Figure 15. Qualitative results for future frame extrapolation on “Collision” of FreeGave-GoPro Dataset.

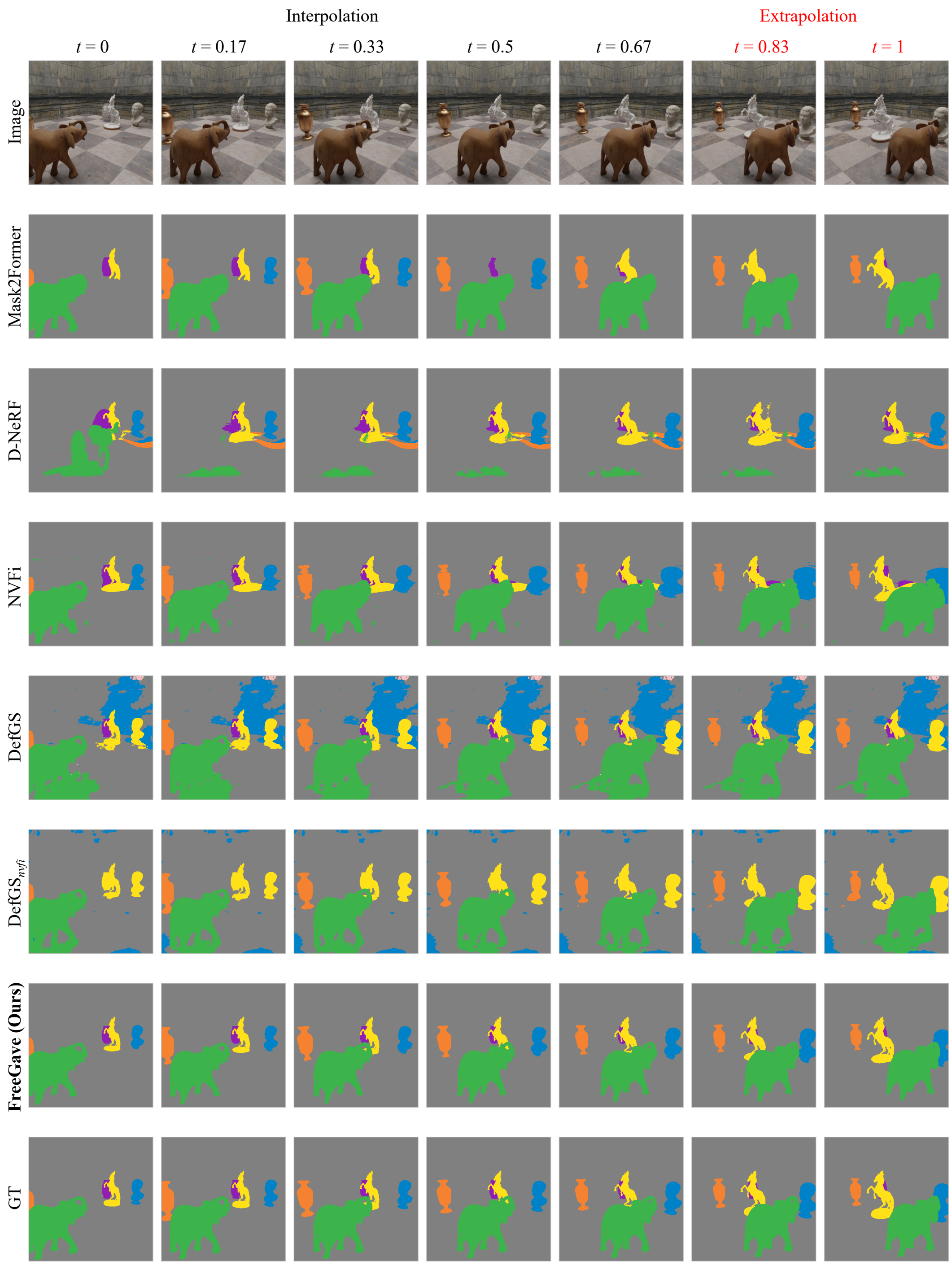

Figure 16. Qualitative results for unsupervised motion segmentation on “Chessboard” of Dynamic Indoor Scene Dataset.

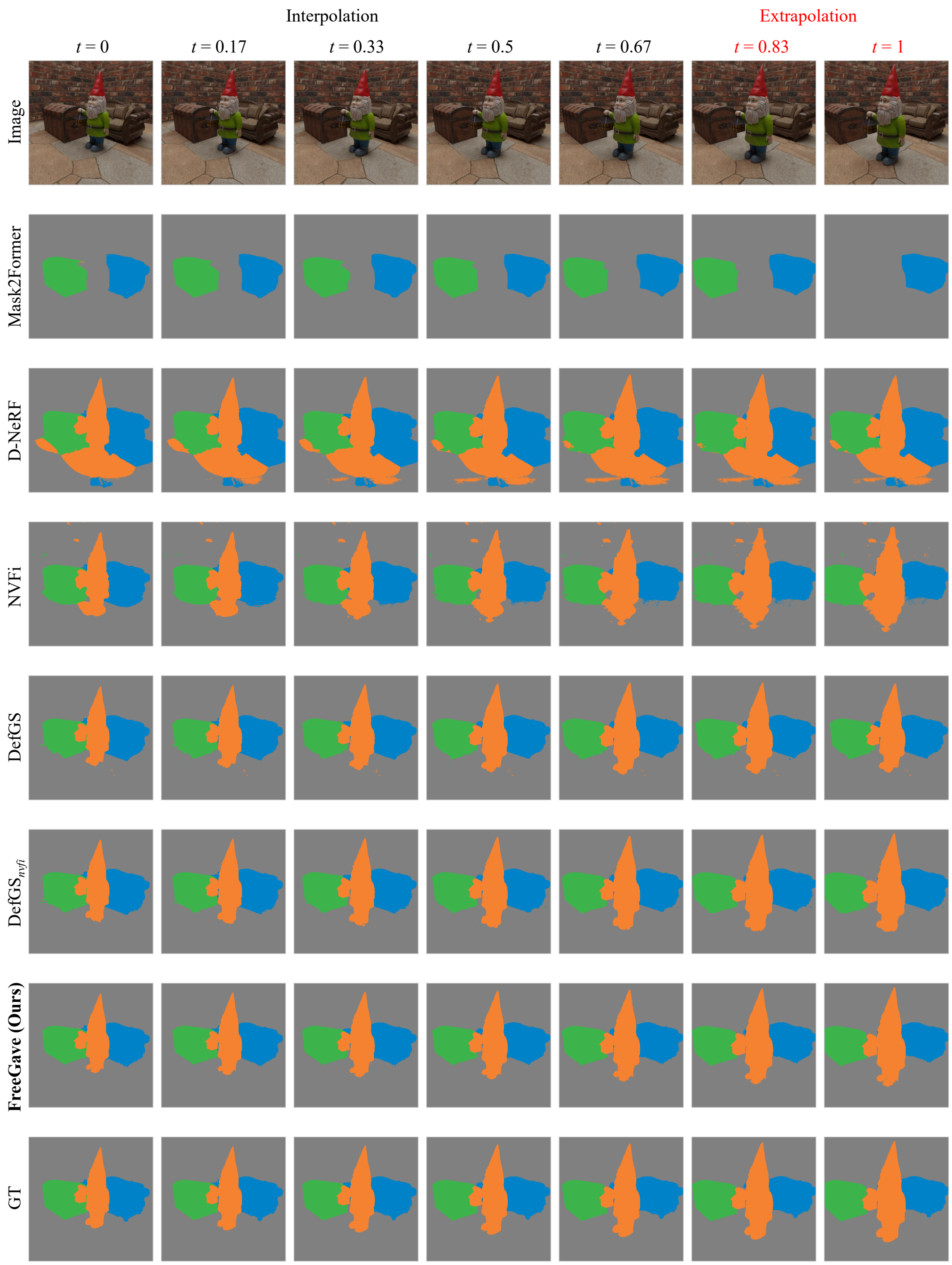

Figure 17. Qualitative results for unsupervised motion segmentation on “Gnome House” of Dynamic Indoor Scene Dataset.

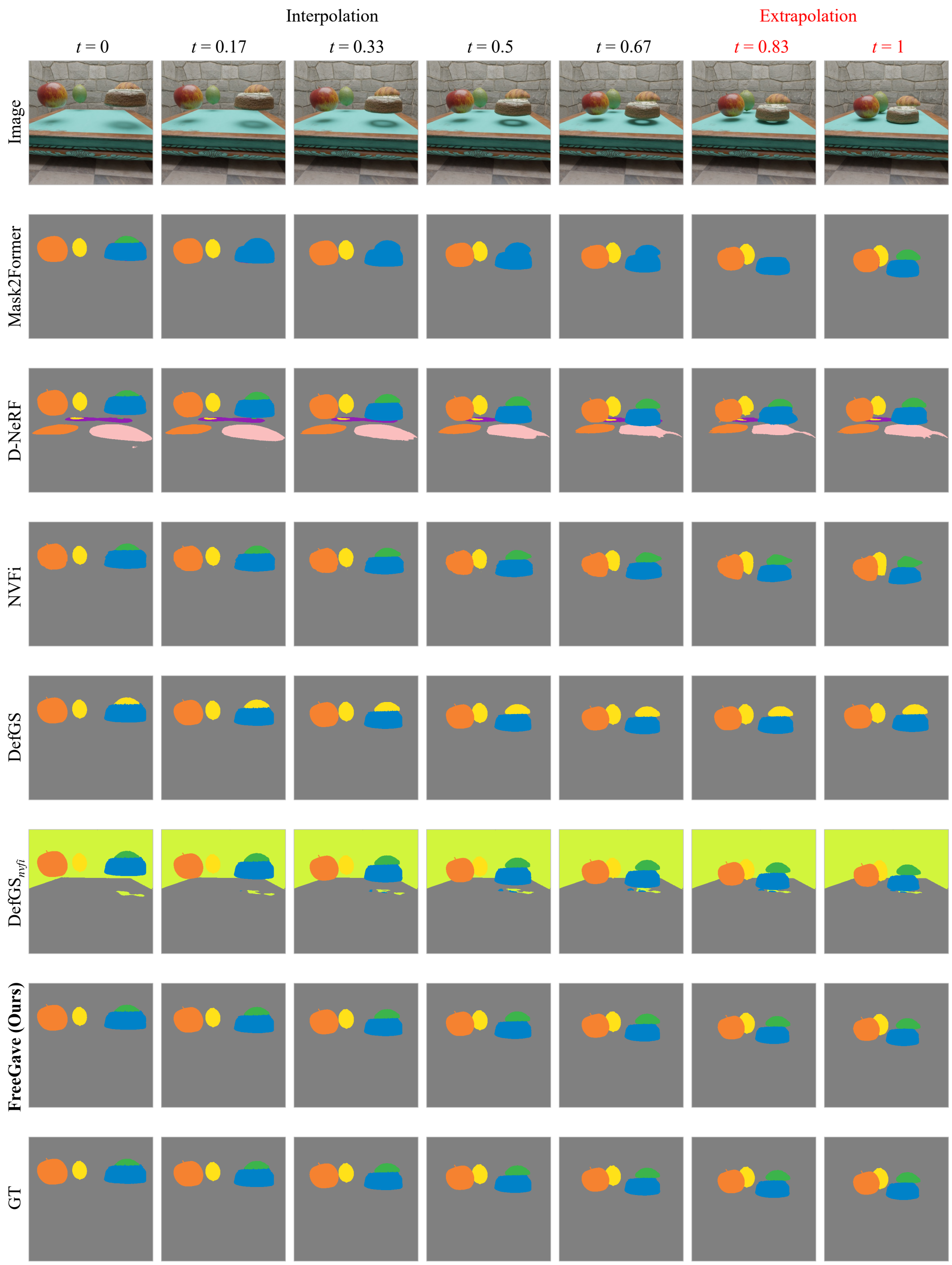

Figure 18. Qualitative results for unsupervised motion segmentation on “Dining Table” of Dynamic Indoor Scene Dataset.

Figure 19. Qualitative results for unsupervised motion segmentation on "Factory" of Dynamic Indoor Scene Dataset.