Improving Physics-Augmented Continuum Neural Radiance Field-Based Geometry-Agnostic System Identification with Lagrangian Particle Optimization

Supplementary Material

Takuhiro Kaneko
NTT Corporation

Contents

A. Details of PAC-NeRF-3v†

A.1. Method

In the experiments (Section 4), we used PAC-NeRF-3v† as the stronger baseline. PAC-NeRF-3v† is an improved variant of PAC-NeRF-3v for sparse-view settings in which Eulerian static voxel grid optimization (Figure 2(a)) is improved. We examined this model to determine whether the proposed LPO, a few-shot learning method for *dynamic* scenes, can be combined with other few-shot learning methods, such as those for *static* scenes.[11] This appendix explains the details of the model. PAC-NeRF-3v† adopts three modifications: scheduling a surface regularizer, introducing view-invariant pixel-wise loss, and adjusting the training length.

Scheduling of surface regularizer. In the original PAC-NeRF [6], a surface regularizer $\mathcal{L}_{surf}$ is applied to regularize the volume density field

$$\mathcal{L}_{surf} = \sum_p \mathrm{clamp}(\alpha_p, 10^{-4}, 10^{-1}) \left(\frac{\Delta x}{2}\right)^2, \quad (9)$$

where α_p indicates the alpha value of a particle with length 1 and is calculated as $\alpha_p = 1 - \exp(-\mathrm{softplus}(\sigma_p))$, where σ_p denotes the volume density of the particle. This regularizer minimizes the total surface area, making the spread of the particles more compact and tightening their shapes. Consequently, the quality of the reconstructed geometries improved [6].

In a preliminary experiment, we found that this regularizer was also effective in eliminating unexpected masses that tend to appear in places where there are few clues owing to the lack of views. However, we also found that this regularizer has a side effect: it removes necessary components when it is too strong. Based on these observations, we scheduled this regularizer.

1. At the beginning of the training, the weight of $\mathcal{L}_{surf}$ is initialized to a default value of PAC-NeRF [6].
2. From the beginning of the training, the weight of $\mathcal{L}_{surf}$ is gradually increased for a certain period of time.
3. After a certain period, the weight of $\mathcal{L}_{surf}$ is gradually decreased until it reaches the default value.

We do not impose a large weight on $\mathcal{L}_{surf}$ from the beginning of the training (Step 1) because it can eliminate all particles, leading to learning "none" object. We decreased the weight of $\mathcal{L}_{surf}$ in Step 3 to alleviate the negative effects caused by the introduction of strong regularization.

Implementation details. In the experiments, we doubled the weight of $\mathcal{L}_{surf}$ in Step 2 every 100 iterations until the weight reached eight times its default value. In Step 3, we halved the weight of $\mathcal{L}_{surf}$ each time the resolution of voxel grids was scaled. This halving process was conducted three

[11] As described in the footnote of the main text,[8] in preliminary experiments, we found that previous representative few-shot learning methods (for example, DietNeRF [4] and FreeNeRF [10]) were less stable than the standard PAC-NeRF-3v. This is possible because, in our experimental settings, the number of views was small (three) despite the wide range of views (upper hemisphere), and explicit voxel representations were more effective than the fully implicit representation in [4, 10]. Therefore, we used an improved variant that we developed. However, this study and previous studies are not competitive but complementary. Therefore, further investigation is important.

times; therefore, the weight of $\mathcal{L}_{surf}$ returned to the default value when all halving processes were finished.

View-invariant pixel-wise loss. In sparse view settings, it is challenging to distinguish view-dependent from view-independent factors because there are few clues. Specifically, in NeRF [7] (mainly, voxel-based NeRF [8]), the view-dependent and view-independent factors (i.e., colors) are represented by a multilayer perceptron (MLP), which additionally receives a view direction $\mathbf{d}$, and color fields $\mathbf{c}^G$ that do not receive $\mathbf{d}$, respectively (Equation (8)). In sparse-view settings, dividing the roles between the MLP and $\mathbf{c}^G$ is not trivial. In extreme cases, the MLP can overfit specific views in the training data (in such cases, $\mathbf{c}^G$ no longer plays an essential role in representing colors), making it difficult to represent colors in novel views. To alleviate this difficulty, we introduce a view-invariant (VI) pixel-wise loss $\mathcal{L}^{VI}_{pixel}$, which is a variant of the pixel-wise loss $\mathcal{L}_{pixel}$ (Equation (3)) where the color of a sample on a ray, i.e., $\mathbf{c}$, is calculated by the following equation instead of Equation (8)

$$\tilde{\mathbf{c}}(\mathbf{x}, \tilde{\mathbf{d}}, t) = \mathrm{MLP}(\mathrm{interp}(\mathbf{x}, \mathbf{c}^G), \tilde{\mathbf{d}}), \tag{10}$$

where $\tilde{\mathbf{d}} \in \mathbb{S}^2$ denotes the view direction randomly sampled from a unit sphere. Here, we use $\tilde{\mathbf{c}}$ to denote $\mathbf{c}$ and distinguish it from the original $\mathbf{c}$. This loss encourages MLP and $\mathbf{c}^G$ to capture the colors of the training images independent of the viewing direction. Consequently, the model makes it possible to avoid extreme cases (i.e., it mitigates the MLP to overfit specific views in the training data and prevents $\mathbf{c}^G$ from losing a role) and provides some colors even for novel views. The total pixel-wise loss $\mathcal{L}^{\dagger}_{pixel}$ is given by

$$\mathcal{L}^{\dagger}_{pixel} = \mathcal{L}_{pixel} + \lambda \mathcal{L}^{VI}_{pixel}, \tag{11}$$

where λ is a hyperparameter balancing the two losses. During training, $\mathcal{L}^{\dagger}_{pixel}$ was used instead of $\mathcal{L}_{pixel}$.

Implementation details. We set $\lambda = 0.1$ in the experiments.

Adjustment of training length. As discussed in previous studies [1, 10], one of the factors that make learning difficult in few-shot settings is overfitting to sparse views in the training data, causing a loss of generalization ability. In preliminary experiments, we observed a similar tendency in our settings. Based on this observation, we adjusted the training length (TL). In particular, the number of iterations was reduced. Despite its simplicity, we empirically found that this solution works well in severe view settings (e.g. when the number of views is three).

Implementation details. In the experiments, we reduced the number of iterations to one-third; the default number (6000) was reduced to 2000. Based on this change, we reduced the timing of scaling the resolution of voxel grids by one-third.

		(a) PAC-NeRF-3v	(b) PAC-NeRF-3v†	(c) w/o scheduling of $\mathcal{L}_{surf}$	(d) w/o $\mathcal{L}^{VI}_{pixel}$	(e) w/o adjustment of TL
Droplet	PSNR↑	24.42	26.83	26.51	26.41	24.88
	SSIM↑	0.975	0.981	0.981	0.979	0.977
	LPIPS↓	0.048	0.045	0.050	0.048	0.047
Letter	PSNR↑	28.56	29.37	28.52	27.75	27.62
	SSIM↑	0.979	0.981	0.978	0.977	0.977
	LPIPS↓	0.032	0.030	0.036	0.032	0.034
Cream	PSNR↑	26.10	27.09	26.51	25.73	25.67
	SSIM↑	0.982	0.983	0.981	0.981	0.980
	LPIPS↓	0.027	0.025	0.027	0.029	0.024
Toothpaste	PSNR↑	28.70	31.77	31.70	31.60	30.76
	SSIM↑	0.988	0.991	0.991	0.991	0.990
	LPIPS↓	0.013	0.012	0.011	0.011	0.012
Torus	PSNR↑	25.17	26.77	25.33	26.15	26.55
	SSIM↑	0.972	0.974	0.972	0.973	0.972
	LPIPS↓	0.047	0.043	0.047	0.043	0.044
Bird	PSNR↑	26.29	27.64	28.82	27.45	26.26
	SSIM↑	0.979	0.982	0.982	0.982	0.981
	LPIPS↓	0.034	0.029	0.033	0.031	0.030
Playdoh	PSNR↑	27.87	30.05	29.03	29.77	28.32
	SSIM↑	0.976	0.981	0.981	0.981	0.978
	LPIPS↓	0.046	0.042	0.046	0.043	0.043
Cat	PSNR↑	30.82	31.93	31.90	30.36	31.75
	SSIM↑	0.987	0.989	0.989	0.987	0.988
	LPIPS↓	0.035	0.035	0.035	0.038	0.036
Trophy	PSNR↑	28.93	29.05	29.23	29.17	28.85
	SSIM↑	0.963	0.963	0.964	0.964	0.964
	LPIPS↓	0.038	0.039	0.039	0.038	0.036
Average	PSNR↑	27.43	28.94	28.62	28.27	27.85
	SSIM↑	0.978	0.980	0.980	0.979	0.979
	LPIPS↓	0.036	0.033	0.036	0.035	0.034

Table 4. Comparison of PSNR↑, SSIM↑, and LPIPS↓ on **static voxel grid optimization**. The scores are calculated for the first frames of the video sequences in the test set. Here, PAC-NeRF-3V† (b), an improved variant of PAC-NeRF-3v, is compared with the original PAC-NeRF-3v (a) and the ablated models, including PAC-NeRF-3V† without scheduling of $\mathcal{L}_{surf}$ (c), that without $\mathcal{L}^{VI}_{pixel}$ (d), and that without adjustment of TL (e). PAC-NeRF-3V† (b) achieved the 20 best and five second-best scores among 27 evaluation items.

A.2. Experiments

Ablation studies were conducted to confirm the importance of each modification. Specifically, we compared *PAC-NeRF-3v*† with three ablated models: *PAC-NeRF-3v*† *without scheduling of* $\mathcal{L}_{surf}$, *PAC-NeRF-3v*† *without* $\mathcal{L}^{VI}_{pixel}$, and *PAC-NeRF-3v*† *without adjustment of TL*. We also examined the original *PAC-NeRF-3v*, that is, PAC-NeRF-3v† without all three modifications.

Results. Table 4 summarizes the results. We found that PAC-NeRF-3v† achieved the best or second-best scores in most cases (20 best scores and five second-best scores among the 27 evaluation items). Consequently, PAC-NeRF-3v† achieved the best average scores in terms of all metrics. These results indicate the importance of each modification.

		PAC-NeRF	PAC-NeRF-3v	+LPO	+LPO-F	+LPO-P	+GO	PAC-NeRF-3v†	+LPO	+LPO-F	+LPO-P	+GO
Droplet	PSNR↑	35.30	25.42	27.56	27.05	27.26	27.55	26.40	28.18	27.36	28.00	27.36
	SSIM↑	0.990	0.975	0.978	0.977	0.977	0.978	0.978	0.980	0.979	0.980	0.979
	LPIPS↓	0.029	0.047	0.043	0.045	0.044	0.044	0.046	0.044	0.045	0.044	0.046
Letter	PSNR↑	36.01	28.94	29.99	29.53	29.90	29.56	29.59	30.44	30.23	30.29	30.09
	SSIM↑	0.991	0.981	0.982	0.982	0.982	0.982	0.983	0.982	0.983	0.982	0.983
	LPIPS↓	0.012	0.028	0.029	0.028	0.029	0.027	0.028	0.029	0.027	0.030	0.026
Cream	PSNR↑	36.70	26.61	28.48	27.79	27.80	28.39	27.43	28.82	28.34	28.50	28.73
	SSIM↑	0.993	0.983	0.983	0.984	0.983	0.984	0.983	0.984	0.984	0.984	0.984
	LPIPS↓	0.014	0.026	0.023	0.024	0.025	0.023	0.024	0.023	0.023	0.024	0.023
Toothpaste	PSNR↑	39.46	29.23	31.27	30.90	30.66	31.63	31.72	33.54	33.15	33.15	33.77
	SSIM↑	0.996	0.991	0.991	0.991	0.991	0.991	0.993	0.993	0.993	0.993	0.993
	LPIPS↓	0.006	0.010	0.010	0.010	0.011	0.010	0.010	0.010	0.010	0.010	0.009
Torus	PSNR↑	34.54	23.99	28.91	25.22	27.82	24.92	26.65	30.05	29.18	29.03	29.27
	SSIM↑	0.988	0.970	0.978	0.971	0.977	0.970	0.974	0.980	0.978	0.978	0.978
	LPIPS↓	0.026	0.048	0.038	0.046	0.040	0.045	0.040	0.034	0.036	0.037	0.036
Bird	PSNR↑	35.55	24.82	27.20	25.47	27.18	25.30	24.91	28.51	25.73	28.43	26.18
	SSIM↑	0.991	0.978	0.980	0.978	0.980	0.978	0.979	0.983	0.980	0.983	0.980
	LPIPS↓	0.019	0.036	0.032	0.035	0.032	0.036	0.036	0.028	0.033	0.028	0.033
Playdoh	PSNR↑	36.43	27.70	28.39	27.85	28.41	27.80	29.66	30.07	29.71	30.13	29.63
	SSIM↑	0.991	0.978	0.978	0.977	0.978	0.977	0.982	0.982	0.982	0.982	0.982
	LPIPS↓	0.026	0.042	0.041	0.042	0.041	0.042	0.038	0.038	0.038	0.038	0.037
Cat	PSNR↑	37.53	30.45	31.00	30.53	30.99	30.19	31.11	31.41	30.98	31.48	30.55
	SSIM↑	0.993	0.987	0.987	0.987	0.987	0.986	0.988	0.988	0.988	0.988	0.988
	LPIPS↓	0.016	0.033	0.032	0.031	0.031	0.031	0.034	0.032	0.032	0.032	0.032
Trophy	PSNR↑	32.43	29.33	30.16	29.69	30.01	29.66	28.80	29.92	29.10	29.78	28.96
	SSIM↑	0.967	0.963	0.964	0.963	0.964	0.963	0.962	0.964	0.963	0.964	0.963
	LPIPS↓	0.034	0.039	0.037	0.039	0.038	0.039	0.039	0.037	0.039	0.038	0.039
Average	PSNR↑	35.99	27.39	29.22	28.22	28.89	28.33	28.47	30.11	29.31	29.87	29.39
	SSIM↑	0.989	0.978	0.980	0.979	0.980	0.979	0.980	0.982	0.981	0.981	0.981
	LPIPS↓	0.020	0.034	0.032	0.033	0.032	0.033	0.033	0.031	0.032	0.031	0.031

Table 5. Comparison of PSNR↑, SSIM↑, and LPIPS↓ for each scene on **geometric correction**. This table is an extended version of Table 1. The qualitative comparisons are provided in Figures 3–7.

B. Detailed analyses on main experiments

The main text focuses on representative results because of space limitations. This appendix provides the extended results that further clarify the effectiveness of the proposed method. In particular, we provide the scores for each scene (Appendix B.1), discuss the impact of the number of iterations in Algorithm 1 (Appendix B.3), and discuss the computation times (Appendix B.2).

B.1. Scores for each scene

In this appendix, we provide the scores for each scene. To evaluate the effectiveness of the proposed method from various perspectives, we describe three experiments in the main text. (I) Evaluation of the geometric correction (Table 1), where the image quality after LPO was applied once, was evaluated. (II) Evaluation of the physical identification (Table 2), in which the accuracy of the physical property estimation after Algorithm 1 was applied, was evaluated. (III) Evaluation of the geometric recorrection (Table 3), in which the image quality after Algorithm 1 was applied, was evaluated. In this appendix, we discuss these issues in detail.

I. Evaluation of geometric correction. First, we discuss the scores for each scene on geometric correction (image quality after LPO was applied once) in detail. Table 5 summarizes the results. This is an extension of the list in Table 1. We discuss the results from three perspectives.

(1) PAC-NeRF-3v/3v† vs. +LPO. When PAC-NeRF-3v was used as the baseline, +LPO outperformed the baseline in most cases (21 wins, 5 draws, and 1 loss). Similarly, when PAC-NeRF-3v† was used as the baseline, +LPO outperformed the baseline in most cases (20 wins, 5 draws, and 2 losses). Consequently, in both cases, +LPO yielded better average scores for all metrics. These results demonstrate that +LPO is effective for geometric correction independent of the baseline models. The qualitative comparisons are shown in Figures 3–7.

(2) +LPO vs. +LPO-F/P. +LPO-F and +LPO-P are ablated variants of +LPO. In +LPO-F, position (shape) optimization is ablated, and only feature (appearance) optimization is conducted. In +LPO-P, feature (i.e., appearance) optimization is ablated, and only position (i.e., shape) optimization is conducted. +LPO outperformed +LPO-F/P in most cases when both PAC-NeRF-3v and PAC-NeRF-3v† were used as the baselines. Specifically, when PAC-NeRF-3v was used as the baseline, +LPO outperformed +LPO-F with 20 wins, 4 draws, and 3 losses and outperformed +LPO-P with 15 wins, 10 draws, and 2 losses. When PAC-NeRF-3v†

		Ground truth	PAC-NeRF	PAC-NeRF-3v	+LPO^4	+LPO-F^4	+LPO-P^4	+GO^4	+None^4
Droplet	μ	2.00×10^2	2.19×10^2	2.76×10^2	2.59×10^2	2.59×10^2	2.63×10^2	2.58×10^2	2.70×10^2
	κ	1.00×10^5	9.62×10^4	5.19×10^3	2.36×10^3	2.30×10^3	3.42×10^3	2.11×10^3	2.07×10^3
Letter	μ	1.00×10^2	9.10×10^1	2.12×10^1	9.67×10^1	9.42×10^1	1.20×10^2	1.12×10^2	2.95×10^1
	κ	1.00×10^5	9.14×10^4	1.69×10^{-2}	6.69×10^4	3.88×10^4	9.12×10^3	1.37×10^4	8.26×10^{-2}
Cream	μ	1.00×10^4	1.23×10^4	2.05×10^4	1.51×10^4	1.46×10^4	1.72×10^4	1.52×10^4	1.78×10^4
	κ	1.00×10^6	1.35×10^6	1.64×10^6	1.44×10^6	1.78×10^6	9.39×10^5	9.37×10^5	1.75×10^6
	τ_Y	3.00×10^3	3.05×10^3	2.90×10^3	2.97×10^3	2.80×10^3	2.96×10^3	3.04×10^3	2.83×10^3
	η	10.00	10.36	19.10	15.22	18.02	12.11	8.69	17.85
Toothpaste	μ	5.00×10^3	5.31×10^3	3.89×10^5	6.42×10^3	7.16×10^3	3.80×10^3	3.43×10^3	3.05×10^3
	κ	1.00×10^5	5.66×10^4	2.63×10^3	2.51×10^4	4.41×10^3	2.53×10^4	2.25×10^4	1.87×10^4
	τ_Y	2.00×10^2	2.33×10^2	3.18×10^2	1.54×10^2	3.11×10^2	1.68×10^2	1.12×10^2	1.74×10^2
	η	10.00	9.71	4.23	9.93	5.41	10.31	10.16	8.97
Torus	E	1.00×10^6	1.05×10^6	1.89×10^6	1.15×10^6	1.02×10^6	1.21×10^6	1.02×10^6	8.83×10^5
	ν	0.300	0.323	0.215	0.299	0.420	0.331	0.374	-0.016
Bird	E	3.00×10^5	2.91×10^5	8.43×10^5	3.50×10^5	1.73×10^5	3.56×10^5	1.46×10^5	1.70×10^5
	ν	0.300	0.329	0.402	0.301	-0.072	0.225	-0.065	-0.281
Playdoh	E	2.00×10^6	3.87×10^6	3.90×10^6	2.61×10^6	7.61×10^6	2.64×10^6	4.03×10^6	2.96×10^6
	τ_Y	1.54×10^4	1.68×10^4	2.96×10^4	2.25×10^4	1.64×10^4	2.52×10^4	2.42×10^4	2.66×10^4
	ν	0.300	0.224	-0.195	0.189	0.127	0.191	0.052	0.172
Cat	E	1.00×10^6	1.39×10^5	5.00×10^4	1.06×10^5	8.55×10^4	8.49×10^4	7.46×10^4	8.58×10^4
	τ_Y	3.85×10^3	3.62×10^3	5.08×10^3	4.93×10^3	5.04×10^3	4.99×10^3	4.69×10^3	4.99×10^3
	ν	0.300	0.327	0.344	0.304	0.336	0.231	0.110	0.379
Trophy	θ_{fric}	40.00°	37.28°	36.97°	36.84°	37.07°	36.79°	36.73°	36.90°

Table 6. Comparison of the values of the physical properties for each scene on **physical identification** when PAC-NeRF-3v was used as a baseline. The absolute differences between the ground truth and the estimated physical properties are provided in Table 2. The qualitative comparisons are provided in Figures 3–9.

was used as the baseline, +LPO outperformed +LPO-F with 17 wins, 8 draws, and 2 losses, and outperformed +LPO-P with 12 wins, 13 draws, and 2 losses, respectively. Between +LPO-F and +LPO-P, +LPO-P tends to outperform +LPO-F. Specifically, when PAC-NeRF-3v was used as the baseline, +LPO-P outperformed +LPO-F with 17 wins, 5 draws, and 5 losses. When PAC-NeRF-3v† was used as the baseline, +LPO-P outperformed +LPO-F with 13 wins, 9 draws, and 5 losses. We consider this because shape correction by +LPO-P can correct the failure estimation of the geometry within the physical constraints of MPM. In contrast, appearance correction by +LPO-F cannot do so and can cause overcorrection beyond the physical constraints, as shown in Figure 8. These results indicate that the feature and position optimizations are complementary rather than competitive.

(3) +LPO vs. +GO. The difference between these two models is that +LPO conducts optimization in Lagrangian particle space and can optimize not only the features but also the positions of the particles, whereas +GO performs optimization in Eulerian grid space and can optimize the features of the grids, but cannot optimize their positions. Owing to these characteristics, the performance of +GO is close to that of +LPO-F, which also optimizes the features but not the positions.[12] Specifically, when PAC-NeRF-3v was used as the baseline, +LPO outperformed +GO with 18 wins, 5 draws, and 4 losses. When PAC-NeRF-3v† was used as the baseline, +LPO outperformed +GO with 16 wins, 6 draws, and 5 losses. These results confirm the importance of position optimization in the +LPO.

II. Evaluation of physical identification. Next, we discuss the scores for each scene for physical identification (the accuracy of the physical property estimation after Algorithm 1 was applied) in detail. Table 2 in the main text summarizes the absolute differences between the ground truth and the estimated physical properties. Tables 6 and 7 summarize the physical property values when PAC-NeRF-3v and PAC-NeRF-3v† were used as the baselines, respectively. These results are discussed from four perspectives.

(1) PAC-NeRF-3v/3v† vs. +LPO^4. +LPO^4 improved the physical identification of PAC-NeRF-3v/3v† in most cases. Specifically, when PAC-NeRF-3v was used as the baseline, +LPO^4 outperformed the baseline for 21 of the 23 evaluation items. When PAC-NeRF-3v† was used as the baseline, +LPO^4 outperformed the baseline for 21 of 23 evaluation items. These results indicate that +LPO is effective for physical identification independent of the baseline models. The qualitative comparisons are shown in Figures 3–7.

(2) +LPO^4 vs. +LPO-F^4/P^4. When comparing these models, we found that superiority depends on the physical properties. This is because the physical properties interact, and finding the optimal balance for performance is challenging. However, +LPO-F^4/P^4 sometimes encountered apparent difficulties. For example, the difference in $\log_{10}(E)$ and

[12] The main differences between these two models are that, in +LPO-F, the positions of particles are fixed during training, while in +GO, the positions of particles are changed in each iteration by random sampling (note that they are not trainable).

		Ground truth	PAC-NeRF	PAC-NeRF-3v†	+LPO^4	+LPO-F^4	+LPO-P^4	+GO^4	+None^4
Droplet	μ	2.00×10^2	2.19×10^2	2.73×10^2	2.41×10^2	2.34×10^2	2.33×10^2	2.42×10^2	2.46×10^2
	κ	1.00×10^5	9.62×10^4	5.45×10^3	1.28×10^5	7.26×10^3	3.02×10^4	3.02×10^3	2.21×10^3
Letter	μ	1.00×10^2	9.10×10^1	4.18×10^1	1.02×10^2	8.18×10^1	9.84×10^1	7.62×10^1	3.66×10^1
	κ	1.00×10^5	9.14×10^4	5.91×10^{-1}	8.70×10^4	1.06×10^5	1.34×10^5	7.28×10^4	8.72×10^{-1}
Cream	μ	1.00×10^4	1.23×10^4	1.51×10^4	1.26×10^4	1.18×10^4	1.16×10^4	1.24×10^4	1.19×10^4
	κ	1.00×10^6	1.35×10^6	2.17×10^6	1.32×10^6	1.57×10^6	1.39×10^6	1.56×10^6	2.18×10^6
	τ_Y	3.00×10^3	3.05×10^3	2.94×10^3	3.04×10^3	2.99×10^3	2.97×10^3	2.93×10^3	2.95×10^3
	η	10.00	10.36	15.67	10.80	11.44	12.01	12.47	11.25
Toothpaste	μ	5.00×10^3	5.31×10^3	2.80×10^3	4.66×10^3	3.15×10^3	4.94×10^3	2.80×10^3	3.64×10^3
	κ	1.00×10^5	5.66×10^4	3.67×10^3	2.12×10^4	2.34×10^4	1.26×10^4	2.57×10^4	2.53×10^4
	τ_Y	2.00×10^2	2.33×10^2	3.16×10^2	1.62×10^2	1.72×10^2	1.53×10^2	1.42×10^2	1.77×10^2
	η	10.00	9.71	6.13	10.20	9.65	10.12	10.16	9.07
Torus	E	1.00×10^6	1.05×10^6	8.43×10^5	1.09×10^6	1.06×10^6	1.09×10^6	9.66×10^5	1.18×10^6
	ν	0.300	0.323	0.431	0.307	0.340	0.267	0.350	0.429
Bird	E	3.00×10^5	2.91×10^5	3.99×10^5	3.19×10^5	4.61×10^5	3.28×10^5	5.10×10^5	5.93×10^5
	ν	0.300	0.329	0.441	0.347	0.372	0.308	0.432	0.445
Playdoh	E	2.00×10^6	3.87×10^6	6.63×10^6	2.72×10^6	5.95×10^6	1.51×10^4	3.41×10^6	1.42×10^4
	τ_Y	1.54×10^4	1.68×10^4	1.99×10^4	2.30×10^4	1.85×10^4	2.33×10^5	2.42×10^4	1.43×10^5
	ν	0.300	0.224	0.088	0.237	0.167	0.410	0.173	0.406
Cat	E	1.00×10^6	1.39×10^5	6.42×10^4	1.97×10^5	2.93×10^5	1.94×10^5	1.65×10^5	1.61×10^5
	τ_Y	3.85×10^3	3.62×10^3	4.67×10^3	4.50×10^3	3.97×10^3	4.59×10^3	4.52×10^3	4.54×10^3
	ν	0.300	0.327	0.418	0.303	0.327	0.293	0.272	0.363
Trophy	θ_{fric}	40.00°	37.28°	38.31°	37.75°	37.66°	37.51°	37.46°	37.79°

Table 7. Comparison of the values of the physical properties for each scene on **physical identification** when PAC-NeRF-3v† was used as a baseline. The absolute differences between the ground truth and the estimated physical properties are provided in Table 2. The qualitative comparisons are provided in Figures 3–9.

that in ν on Bird was large when +LPO-F^4 was used with PAC-NeRF-3v and the difference in $\log_{10}(E)$ on Playdoh was large when +LPO-P^4 was used with PAC-NeRF-3v†. In contrast, +LPO^4 exhibited stable performance. Owing to this stability, +LPO^4 outperformed +LPO-F^4 and +LPO-P^4 in most cases. Specifically, when PAC-NeRF-3v was used as the baseline, +LPO^4 outperformed +LPO-F^4 for 18 items and outperformed +LPO-P^4 for 17 of the 23 evaluation items. When PAC-NeRF-3v† was used as the baseline, +LPO^4 outperformed +LPO-F^4 for 13 items and outperformed +LPO-P^4 for 16 of the 23 evaluation items. These results indicate that the joint optimization of features and positions in Lagrangian space is useful for obtaining stability and tackling difficult situations.

(3) +LPO^4 vs. +GO^4. +GO^4 also sometimes suffers from critical difficulties. For example, the difference in $\log_{10}(E)$ and that in ν on Bird was large when +GO^4 was used with PAC-NeRF-3v. In contrast, +LPO^4 exhibited stable performance and outperformed +GO^4 in most cases. Specifically, when PAC-NeRF-3v was used as the baseline, +LPO^4 outperformed +GO^4 for 18 of the 23 evaluation items. When PAC-NeRF-3v† was used as the baseline, +LPO^4 outperformed +GO^4 for 18 of the 23 evaluation items.

(4) +LPO^4 vs. +None^4. +None^4 is outperformed by the dynamic optimization methods (i.e., +LPO^4, +LPO-F^4, +LPO-P^4, and +GO^4) in most cases. In particular, when PAC-NeRF-3v was used as the baseline, +LPO^4 outperformed +None^4 for 20 of the 23 evaluation items. When PAC-NeRF-3v† was used as the baseline, +LPO^4 outperformed +None^4 for 19 of the 23 evaluation items. These results indicate that simple iterative updates without geometric correction in Algorithm 1 are insufficient to improve physical identification and that it is crucial to correct geometric structures using a dynamic optimization method.

III. Evaluation of geometric recorrection. Finally, we discuss the scores for each scene in the geometry recorrection (image quality after Algorithm 1 was applied) in detail. Table 8 summarizes these results. This is an extension of the results in Table 3. We observed tendencies similar to those for geometry correction. Specifically, +LPO^4 outperformed not only the baselines (PAC-NeRF-3v and PAC-NeRF-3v†) but also the ablated and comparative models, including +LPO-F^4, +LPO-P^4, +GO^4, and +None^4, in most cases. These results indicate that joint optimization of the features and positions of particles in Lagrangian space is essential not only when geometric correction is conducted once but also when geometric correction is repeatedly conducted along with physical reidentification through Algorithm 1. Qualitative comparisons of PAC-NeRF-3v/3v†, +LPO, and +LPO^4 are shown in Figure 3–7. Qualitative comparisons of +LPO^4, +LPO-F^4, +LPO-P^4, +GO^4, and +None^4 are presented in Figures 8 and 9.

B.2. Computation times

Table 9 lists the computation times for executing Algorithm 1 for one iteration on Droplet with PAC-NeRF-

		PAC-NeRF	PAC-NeRF-3v	+LPO[4]	+LPO-F[4]	+LPO-P[4]	+GO[4]	+None[4]	PAC-NeRF-3v†	+LPO[4]	+LPO-F[4]	+LPO-P[4]	+GO[4]	+None[4]
Droplet	PSNR↑	35.30	25.42	28.21	26.52	28.21	26.03	25.42	26.40	28.91	27.85	29.03	26.23	27.52
	SSIM↑	0.990	0.975	0.981	0.978	0.980	0.978	0.977	0.978	0.983	0.982	0.983	0.980	0.981
	LPIPS↓	0.029	0.047	0.042	0.046	0.045	0.046	0.047	0.046	0.041	0.042	0.043	0.045	0.043
Letter	PSNR↑	36.01	28.94	30.07	30.07	29.83	30.00	29.25	29.59	30.26	31.09	30.50	29.70	29.80
	SSIM↑	0.991	0.981	0.983	0.983	0.982	0.983	0.983	0.983	0.983	0.984	0.982	0.982	0.984
	LPIPS↓	0.012	0.028	0.026	0.026	0.029	0.026	0.026	0.028	0.026	0.024	0.029	0.025	0.025
Cream	PSNR↑	36.70	26.61	29.65	29.02	28.83	28.95	26.57	27.43	30.23	29.87	29.65	29.79	27.88
	SSIM↑	0.993	0.983	0.985	0.985	0.984	0.985	0.983	0.983	0.985	0.985	0.985	0.985	0.984
	LPIPS↓	0.014	0.026	0.022	0.023	0.024	0.022	0.026	0.024	0.021	0.022	0.022	0.021	0.024
Toothpaste	PSNR↑	39.46	29.23	32.18	32.41	31.87	32.87	28.96	31.72	34.00	34.51	33.94	34.40	31.49
	SSIM↑	0.996	0.991	0.991	0.992	0.991	0.992	0.991	0.993	0.992	0.993	0.993	0.993	0.993
	LPIPS↓	0.006	0.010	0.011	0.010	0.011	0.010	0.010	0.010	0.010	0.009	0.010	0.009	0.009
Torus	PSNR↑	34.54	23.99	30.07	27.06	29.37	25.36	24.82	26.65	30.48	30.01	30.23	28.93	26.74
	SSIM↑	0.988	0.970	0.982	0.975	0.981	0.970	0.973	0.974	0.983	0.982	0.982	0.979	0.975
	LPIPS↓	0.026	0.048	0.032	0.039	0.035	0.041	0.044	0.040	0.031	0.032	0.033	0.033	0.040
Bird	PSNR↑	35.55	24.82	27.97	24.86	27.71	24.20	23.98	24.91	28.98	26.51	28.74	26.08	24.95
	SSIM↑	0.991	0.978	0.980	0.979	0.980	0.977	0.978	0.979	0.982	0.980	0.982	0.979	0.980
	LPIPS↓	0.019	0.036	0.034	0.039	0.034	0.041	0.040	0.036	0.031	0.033	0.032	0.034	0.036
Playdoh	PSNR↑	36.43	27.70	29.32	28.05	29.01	28.02	27.94	29.66	30.01	29.45	29.25	28.94	28.02
	SSIM↑	0.991	0.978	0.981	0.978	0.980	0.979	0.979	0.982	0.983	0.982	0.982	0.982	0.981
	LPIPS↓	0.026	0.042	0.040	0.043	0.043	0.042	0.042	0.038	0.038	0.039	0.045	0.041	0.047
Cat	PSNR↑	37.53	30.45	29.81	29.71	29.95	28.35	30.41	31.11	30.61	30.47	31.47	29.21	31.78
	SSIM↑	0.993	0.987	0.987	0.987	0.987	0.985	0.987	0.988	0.988	0.988	0.989	0.987	0.989
	LPIPS↓	0.016	0.033	0.028	0.028	0.028	0.028	0.028	0.034	0.025	0.026	0.025	0.026	0.025
Trophy	PSNR↑	32.43	29.33	29.57	29.13	29.89	28.31	29.44	28.80	29.58	29.23	30.01	27.87	29.18
	SSIM↑	0.967	0.963	0.964	0.963	0.964	0.963	0.963	0.962	0.964	0.963	0.964	0.963	0.963
	LPIPS↓	0.034	0.039	0.037	0.039	0.039	0.039	0.039	0.039	0.036	0.039	0.038	0.039	0.039
Average	PSNR↑	35.99	27.39	29.65	28.54	29.41	28.01	27.42	28.47	30.34	29.89	30.31	29.02	28.60
	SSIM↑	0.989	0.978	0.982	0.980	0.981	0.979	0.979	0.980	0.983	0.982	0.982	0.981	0.981
	LPIPS↓	0.020	0.034	0.030	0.032	0.032	0.033	0.034	0.033	0.029	0.030	0.031	0.030	0.032

Table 8. Comparison of PSNR↑, SSIM↑, and LPIPS↓ for each scene on **geometric recorrection**. This table is an extended version of Table 3. The qualitative comparisons are provided in Figures 3–9.

Process	# Frames	Time (s)
(1) Eulerian static voxel grid optimization	1	284.6
(2) Physical property optimization		
(a) Velocity optimization	4	101.9
(b) Warm-up optimization with partial frames	7	765.2
(c) Main optimization with entire frames	13	3153.3
(3) Lagrangian particle optimization	13	3225.4
(4) Color prediction	1	1.0

Table 9. Computation times of Algorithm 1 on NVIDIA A100 GPU. (1) and (2) are processes driven from the original PAC-NeRF. (3) and (4) are processes newly introduced.

3v+LPO.[13] The total computation time increases linearly when repeatedly running Algorithm 1. However, it is adjustable under a quality-and-time trade-off, as discussed in Appendix B.3. The computation time of LPO (3) is almost identical to that of the main process of physical property optimization (2-c) because the forward and backward processes are identical with different optimization targets, as shown in Figure 2(2) and (3). Similarly, the calculation times for +LPO-F, +LPO-P, and +GO were almost identical to those for +LPO, indicating that the performance improvement was attributable to the ingenuity of the algorithm and not to an increase in calculation cost.

[13]Computation times vary with scenes because of the difference in number of frames and object size (which affects the number of particles); however, we observed similar tendencies.

B.3. Impact of the number of iterations

In the main experiments, we fixed the number of iterations in Algorithm 1, that is, R, to four for simplicity and fair comparison. However, it is interesting and important to investigate how the number of iterations affects performance. To answer this question, we analyze the impact of the number of iterations in this appendix.

Results. Table 10 summarizes the performance changes with geometry (re)correction when the number of iterations was changed. This table calculates the scores using the images in the *test* set. Table 11 lists the scores calculated using images in the *training* set. We investigated these two cases to examine the trade-off between the feasible reconstruction of the training data and the generalization ability for the test data. Table 12 presents the performance changes with physical identification when the number of iterations is changed.

When PAC-NeRF-3v was used as the baseline, the performances for geometry (re)correction (for both the test and training sets) and physical identification were the best when the number of iterations was four in most cases. Specifically, as listed in Table 10, 6, 8, 13, and 18 itmes achieved the best scores among the 27 evaluation items when the number of iterations was one, two, three, and four, respectively, for geometry (re)correction for the test set. When

		PAC-NeRF-3v	+LPO[1]	+LPO[2]	+LPO[3]	+LPO[4]	PAC-NeRF-3v†	+LPO[1]	+LPO[2]	+LPO[3]	+LPO[4]
Droplet	PSNR↑	25.42	27.56	28.19	28.06	28.21	26.40	28.18	28.95	29.14	28.91
	SSIM↑	0.975	0.978	0.980	0.981	0.981	0.978	0.980	0.982	0.983	0.983
	LPIPS↓	0.047	0.043	0.043	0.042	0.042	0.046	0.044	0.042	0.041	0.041
Letter	PSNR↑	28.94	29.99	29.90	29.96	30.07	29.59	30.44	30.76	30.70	30.26
	SSIM↑	0.981	0.982	0.983	0.982	0.983	0.983	0.982	0.983	0.984	0.983
	LPIPS↓	0.028	0.029	0.027	0.031	0.026	0.028	0.029	0.026	0.026	0.026
Cream	PSNR↑	26.61	28.48	29.18	29.38	29.65	27.43	28.82	29.69	29.93	30.23
	SSIM↑	0.983	0.983	0.984	0.985	0.985	0.983	0.984	0.985	0.985	0.985
	LPIPS↓	0.026	0.023	0.022	0.023	0.022	0.024	0.023	0.022	0.022	0.021
Toothpaste	PSNR↑	29.23	31.27	31.78	31.93	32.18	31.72	33.54	34.16	34.04	34.00
	SSIM↑	0.991	0.991	0.991	0.991	0.991	0.993	0.993	0.993	0.992	0.992
	LPIPS↓	0.010	0.010	0.011	0.011	0.011	0.010	0.010	0.010	0.010	0.010
Torus	PSNR↑	23.99	28.91	29.87	30.20	30.07	26.65	30.05	30.90	30.97	30.48
	SSIM↑	0.970	0.978	0.981	0.982	0.982	0.974	0.980	0.983	0.984	0.983
	LPIPS↓	0.048	0.038	0.034	0.032	0.032	0.040	0.034	0.031	0.030	0.031
Bird	PSNR↑	24.82	27.20	26.55	27.89	27.97	24.91	28.51	28.88	28.85	28.98
	SSIM↑	0.978	0.980	0.979	0.981	0.980	0.979	0.983	0.983	0.982	0.982
	LPIPS↓	0.036	0.032	0.035	0.033	0.034	0.036	0.028	0.030	0.031	0.031
Playdoh	PSNR↑	27.70	28.39	29.14	29.32	29.32	29.66	30.07	30.16	30.26	30.01
	SSIM↑	0.978	0.978	0.981	0.981	0.981	0.982	0.982	0.983	0.983	0.983
	LPIPS↓	0.042	0.041	0.040	0.040	0.040	0.038	0.038	0.038	0.038	0.038
Cat	PSNR↑	30.45	31.00	30.64	30.21	29.81	31.11	31.41	31.25	31.29	30.61
	SSIM↑	0.987	0.987	0.988	0.987	0.987	0.988	0.988	0.989	0.989	0.988
	LPIPS↓	0.033	0.032	0.027	0.027	0.028	0.034	0.032	0.027	0.024	0.025
Trophy	PSNR↑	29.33	30.16	29.99	29.78	29.57	28.80	29.92	29.97	29.82	29.58
	SSIM↑	0.963	0.964	0.964	0.964	0.964	0.962	0.964	0.964	0.964	0.964
	LPIPS↓	0.039	0.037	0.037	0.037	0.037	0.039	0.037	0.037	0.036	0.036
Average	PSNR↑	27.39	29.22	29.47	29.64	29.65	28.47	30.11	30.52	30.56	30.34
	SSIM↑	0.978	0.980	0.981	0.981	0.982	0.980	0.982	0.983	0.983	0.983
	LPIPS↓	0.034	0.032	0.031	0.031	0.030	0.033	0.031	0.029	0.029	0.029

Table 10. Comparison of PSNR↑, SSIM↑, and LPIPS↓ on **geometric (re)correction** when the number of iterations in Algorithm 1 is changed. The scores were calculated using the images in the *test* set.

scores were the same, they were counted multiple times. As listed in Table 11, 1, 3, 11, and 25 items achieved the best scores among the 27 evaluation items when the number of iterations was one, two, three, and four, respectively, for geometry (re)correction for the training set. As listed in Table 12, 2, 3, 3, and 16 items achieved the best scores among the 23 evaluation items when the numbers of iterations were one, two, three, and four for physical identification.

In contrast, when PAC-NeRF-3v† was used as the baseline, the performance of the geometry (re)correction for the test set was the best when the number of iterations was three, whereas that for the training set was the best when the number of iterations was four. The performance of the physical identification was the best when the number of iterations was three or four. Specifically, as listed in Table 10, 5, 10, 15, and 10 items achieved the best scores among the 27 evaluation items when the number of iterations was one, two, three, and four, respectively, for geometry (re)correction for the test set. As listed in Table 11, 0, 4, 14, and 26 items achieved the best scores among the 27 evaluation items when the number of iterations was one, two, three, and four, respectively, for geometry (re)correction for the training set. As listed in Table 12, 1, 4, 7, and 11 items achieved the best scores among the 23 evaluation items when the numbers of iterations were one, two, three, and four for physical identification.

We consider that these differences arise from differences in the initial static voxel grids. PAC-NeRF-3v† had better static voxel grids; therefore, fewer updates were required to obtain the best performance (i.e., optimal geometric structures). Note that it is generally not trivial to determine when to stop iterative updates because there is an intractable trade-off between faithful reproduction of training data and overfitting. When PAC-NeRF-3v was used as the baseline, the performance improvement continued until four iterations because the geometric reconstruction was more influential than overfitting. In contrast, when PAC-NeRF-3v† was used as the baseline, the performance improvement (particularly for geometric (re)correction for the test set) was saturated at three iterations because the performance achieved the upper bound in an earlier phase. From this time on, the overfitting problem became non-negligible. Conducting further studies on this topic and exploring an improved method (for example, determining the number of iterations adaptively during training) will be interesting future research topics. Furthermore, the effect of the above-mentioned trade-off on the physical identification performance is an open issue. A detailed investigation of this will be an interesting topic for future research.

		PAC-NeRF-3v	+LPO[1]	+LPO[2]	+LPO[3]	+LPO[4]	PAC-NeRF-3v†	+LPO[1]	+LPO[2]	+LPO[3]	+LPO[4]
Droplet	PSNR↑	32.48	36.61	36.97	37.12	37.41	32.31	36.09	37.21	37.83	37.96
	SSIM↑	0.985	0.990	0.991	0.991	0.992	0.985	0.990	0.992	0.992	0.992
	LPIPS↓	0.034	0.029	0.029	0.029	0.028	0.036	0.033	0.029	0.027	0.027
Letter	PSNR↑	32.57	35.89	38.74	37.62	39.59	32.92	36.84	38.81	39.09	39.90
	SSIM↑	0.986	0.990	0.994	0.992	0.995	0.986	0.992	0.994	0.995	0.995
	LPIPS↓	0.020	0.020	0.015	0.019	0.013	0.021	0.019	0.015	0.013	0.012
Cream	PSNR↑	34.91	38.45	39.50	37.90	40.11	35.18	38.50	38.70	39.60	40.39
	SSIM↑	0.991	0.990	0.994	0.992	0.995	0.990	0.993	0.994	0.995	0.995
	LPIPS↓	0.016	0.014	0.013	0.014	0.013	0.017	0.014	0.014	0.013	0.012
Toothpaste	PSNR↑	37.41	41.49	42.10	42.54	42.91	37.36	41.43	42.56	43.07	43.41
	SSIM↑	0.996	0.997	0.997	0.997	0.997	0.996	0.997	0.997	0.997	0.998
	LPIPS↓	0.006	0.005	0.005	0.005	0.004	0.006	0.005	0.005	0.004	0.004
Torus	PSNR↑	28.45	34.29	35.96	36.75	37.20	31.70	35.30	36.66	37.22	37.56
	SSIM↑	0.977	0.986	0.989	0.990	0.990	0.981	0.987	0.990	0.990	0.990
	LPIPS↓	0.041	0.031	0.028	0.027	0.027	0.034	0.029	0.027	0.026	0.026
Bird	PSNR↑	30.72	33.22	32.06	34.91	34.88	30.97	33.84	34.73	35.47	35.91
	SSIM↑	0.985	0.987	0.986	0.990	0.990	0.987	0.989	0.990	0.990	0.991
	LPIPS↓	0.029	0.027	0.030	0.025	0.025	0.028	0.026	0.025	0.024	0.023
Playdoh	PSNR↑	34.50	40.38	41.47	41.23	41.80	35.51	40.92	41.22	41.93	41.74
	SSIM↑	0.988	0.993	0.994	0.994	0.994	0.989	0.993	0.994	0.994	0.994
	LPIPS↓	0.027	0.022	0.022	0.021	0.021	0.027	0.023	0.022	0.021	0.021
Cat	PSNR↑	37.25	41.34	41.77	42.37	42.94	36.66	40.43	42.17	43.32	43.49
	SSIM↑	0.993	0.996	0.996	0.997	0.997	0.993	0.995	0.996	0.997	0.997
	LPIPS↓	0.020	0.018	0.017	0.016	0.016	0.026	0.024	0.018	0.014	0.014
Trophy	PSNR↑	32.35	34.76	34.82	35.01	35.13	31.86	34.54	34.78	35.00	35.19
	SSIM↑	0.966	0.969	0.969	0.970	0.970	0.966	0.969	0.969	0.970	0.970
	LPIPS↓	0.033	0.031	0.031	0.031	0.031	0.035	0.032	0.032	0.031	0.031
Average	PSNR↑	33.41	37.38	38.16	38.38	39.11	33.83	37.54	38.54	39.17	39.51
	SSIM↑	0.985	0.989	0.990	0.990	0.991	0.986	0.990	0.991	0.991	0.991
	LPIPS↓	0.025	0.022	0.021	0.021	0.020	0.026	0.023	0.021	0.019	0.019

Table 11. Comparison of PSNR↑, SSIM↑, and LPIPS↓ on **geometric (re)correction** when the number of iterations in Algorithm 1 is changed. The scores were calculated using the images in the *training* set.

		PAC-NeRF-3v	+LPO[2]	+LPO[3]	+LPO[4]	PAC-NeRF-3v†	+LPO[2]	+LPO[3]	+LPO[4]
Droplet	$\log_{10}(\mu)$	0.140	0.136	0.111	0.112	0.136	0.129	0.084	0.082
	$\log_{10}(\kappa)$	1.285	1.447	1.784	1.628	1.263	1.053	0.097	0.106
Letter	$\log_{10}(\mu)$	0.674	0.023	1.264	0.015	0.379	0.013	0.053	0.010
	$\log_{10}(\kappa)$	6.772	0.325	1.424	0.174	5.229	0.507	0.589	0.060
Cream	$\log_{10}(\mu)$	0.311	0.200	0.115	0.178	0.179	0.031	0.066	0.100
	$\log_{10}(\kappa)$	0.215	0.384	0.392	0.158	0.336	0.157	0.060	0.121
	$\log_{10}(\tau_Y)$	0.014	0.032	0.031	0.004	0.009	0.007	0.005	0.006
	$\log_{10}(\eta)$	0.281	0.209	0.198	0.183	0.195	0.105	0.026	0.033
Toothpaste	$\log_{10}(\mu)$	1.891	0.264	0.246	0.109	0.252	0.259	0.026	0.031
	$\log_{10}(\kappa)$	1.580	1.439	1.434	0.601	1.436	1.382	0.888	0.673
	$\log_{10}(\tau_Y)$	0.201	0.118	0.116	0.114	0.199	0.168	0.137	0.093
	$\log_{10}(\eta)$	0.373	0.200	0.180	0.003	0.212	0.187	0.005	0.009
Torus	$\log_{10}(E)$	0.277	0.055	0.053	0.061	0.074	0.049	0.040	0.036
	ν	0.085	0.129	0.050	0.001	0.131	0.032	0.060	0.007
Bird	$\log_{10}(E)$	0.449	0.136	0.146	0.067	0.123	0.158	0.084	0.027
	ν	0.102	0.466	0.037	0.001	0.141	0.009	0.056	0.047
Playdoh	$\log_{10}(E)$	0.290	0.403	0.190	0.116	0.521	0.303	0.260	0.133
	$\log_{10}(\tau_Y)$	0.283	0.075	0.215	0.165	0.110	0.119	0.105	0.173
	ν	0.495	0.092	0.236	0.111	0.212	0.218	0.082	0.063
Cat	$\log_{10}(E)$	1.301	1.120	1.051	0.973	1.192	0.584	0.687	0.706
	$\log_{10}(\tau_Y)$	0.120	0.119	0.111	0.107	0.084	0.062	0.070	0.067
	ν	0.044	0.079	0.104	0.004	0.118	0.054	0.024	0.003
Trophy	θ_{fric} [rad]	0.053	0.058	0.053	0.055	0.030	0.036	0.046	0.039

Table 12. Comparison of the absolute differences between the ground-truth and estimated physical properties on **physical identification** when the number of iterations in Algorithm 1 is changed. The smaller the values, the better the performance.

	PAC-NeRF-3v†	+LPO	+LPO-F	+LPO-P	+GO	+LPO^4	+LPO-F^4	+LPO-P^4	+GO^4	+None^4
PSNR↑	28.18±.43	29.60±.59	28.89±.45	29.40±.55	28.91±.45	29.88±.53	29.34±.51	29.81±.54	28.79±.44	28.49±.45
SSIM↑	0.979±.001	0.981±.001	0.980±.001	0.981±.001	0.980±.001	0.982±.001	0.981±.001	0.982±.001	0.981±.001	0.981±.001
LPIPS↓	0.036±.002	0.034±.003	0.035±.002	0.035±.003	0.035±.002	0.031±.002	0.032±.002	0.033±.003	0.033±.002	0.034±.002

Table 13. Comparison of PSNR↑, SSIM↑, and LPIPS↓ averaged over five view settings on **geometric correction** and **recorrection**.

C. Experiments on other view settings

In the main text (Section 4), we investigate the performance when three specific views are selected as training data. In this appendix, experiments were conducted using other settings to investigate the versatility of the proposed method. In particular, we investigated the robustness of the selection of views (Appendix C.1) and the number of views (Appendix C.2).

C.1. Robustness of the selection of views

In the main text (Section 4), we investigate the performance when three specific views are selected as training data. We examined the performance when the training set was changed to investigate the robustness of the selection of views. Specifically, we investigated performance when five different view sets were used for training. The number of views used for training was fixed at three, and the remaining eight were used for testing.

Compared models. We used *PAC-NeRF-3v*† as the baseline and applied *+LPO* and *+LPO*^{4}. Furthermore, we examined the performances of the ablated and comparative models. Specifically, in the evaluation of the geometry correction, we examined the performance when *+LPO-F*, *+LPO-P*, and *+GO* were applied to the baseline. In the evaluation of the physical identification and geometric recorrection, we examined the performance when *+LPO-F*^{4}, *+LPO-P*^{4}, *+GO*^{4}, and *+None*^{4} were applied to the baseline.

Results. Table 13 summarizes the geometric correction and recorrection results. We observed tendencies similar to those for the results when three specific views were used for training (see Appendix B.1). Specifically, with respect to geometry correction, +LPO outperformed not only the baseline (PAC-NeRF-3v†) but was also superior to or comparable to the ablated and comparative models, including +LPO-F, +LPO-P, and +GO. Similarly, with respect to geometry recorrection, +LPO^4 outperformed not only the baseline (PAC-NeRF-3v†) but was also superior to or comparable to the ablated and comparative models, including +LPO-F^4, +LPO-P^4, +GO^4, and +None^4. In terms of physical property identification, +LPO^4 outperformed PAC-NeRF-3v†, +LPO-F^4, +LPO-P^4, +GO^4, and +None^4 in the 20.8±1.2, 15.2±2.0, 16.6±2.0, 17.8±1.2, and 18.8±1.9 cases, respectively, across the 23 properties for each view setting. These results indicate that the joint optimization of features and positions in Lagrangian space is effective for geometry-agnostic system identification, regardless of the selection of views used for training.

C.2. Robustness of the number of views

In the main experiment (Section 4) and the above experiments (Appendix C.1), we investigated the performance when the number of views in the training set was three. To investigate the robustness of the number of views, we examined the performance when the number of views in the training set was changed to six. Specifically, among the 11 views in the dataset, six were used for training, and the remaining five were used for testing.

Compared models. We used *PAC-NeRF-6v*, that is, PAC-NeRF [6] trained with six views, as the baseline. In preliminary experiments, we found that PAC-NeRF-6v demonstrated reasonable performance without the advanced techniques described in Appendix A owing to the increase in the number of views. Therefore, we only used this model as the baseline. We applied *+LPO* and *+LPO*^{4} to the baseline. Furthermore, we investigated the performances of the ablated and comparative models. Specifically, in the evaluation of the geometry correction, we investigated the performance when *+LPO-F*, *+LPO-P*, and *+GO* were applied to the baseline. In the evaluation of the physical identification and geometric recorrection, we investigated the performance when *+LPO-F*^{4}, *+LPO-P*^{4}, *+GO*^{4}, and *+None*^{4} were applied to the baseline.

Results. Tables 14, 15, and 16 summarize the results of geometry correction, physical identification, and geometry reconstruction, respectively. Similar tendencies were observed when the number of views in the training set was three (as discussed in Section 4 and Appendix C.1). Specifically, with respect to geometry correction, +LPO outperformed not only the baseline (PAC-NeRF-6v) but also the ablated and comparative models, including +LPO-F, +LPO-P, and +GO, in most cases. On physical identification and geometry recorrection, +LPO^4 outperformed not only the baseline (PAC-NeRF-6v) but also the ablated and comparative models, including +LPO-F^4, +LPO-P^4, +GO^4, and +None^4, in most cases. These results indicate that the joint optimization of features and positions in Lagrangian space is effective for geometry-agnostic system identification, regardless of the number of views used for training.

		PAC-NeRF	PAC-NeRF-6v	+LPO	+LPO-F	+LPO-P	+GO
Droplet	PSNR↑	35.29	28.92	30.58	29.82	30.47	29.13
	SSIM↑	0.989	0.982	0.984	0.983	0.984	0.983
	LPIPS↓	0.030	0.042	0.040	0.041	0.040	0.042
Letter	PSNR↑	36.99	31.42	32.85	32.36	32.62	32.32
	SSIM↑	0.992	0.985	0.987	0.987	0.986	0.986
	LPIPS↓	0.011	0.020	0.019	0.019	0.019	0.020
Cream	PSNR↑	36.46	30.36	31.31	31.13	30.89	30.80
	SSIM↑	0.993	0.986	0.987	0.987	0.986	0.987
	LPIPS↓	0.014	0.021	0.019	0.019	0.020	0.019
Toothpaste	PSNR↑	38.84	34.74	35.82	35.27	35.58	34.79
	SSIM↑	0.996	0.993	0.994	0.993	0.993	0.993
	LPIPS↓	0.006	0.009	0.009	0.009	0.009	0.009
Torus	PSNR↑	34.60	29.28	32.31	30.82	31.80	30.80
	SSIM↑	0.988	0.979	0.984	0.981	0.984	0.982
	LPIPS↓	0.026	0.035	0.030	0.033	0.030	0.034
Bird	PSNR↑	35.70	27.38	30.48	28.67	30.39	28.09
	SSIM↑	0.992	0.981	0.984	0.982	0.984	0.981
	LPIPS↓	0.019	0.029	0.025	0.027	0.025	0.028
Playdoh	PSNR↑	36.55	29.74	31.01	29.93	30.99	29.84
	SSIM↑	0.991	0.982	0.983	0.982	0.983	0.982
	LPIPS↓	0.026	0.039	0.037	0.039	0.037	0.039
Cat	PSNR↑	37.10	30.77	31.40	30.85	31.33	30.72
	SSIM↑	0.993	0.989	0.987	0.989	0.988	0.988
	LPIPS↓	0.016	0.024	0.026	0.025	0.026	0.025
Trophy	PSNR↑	32.23	30.22	30.97	30.38	30.85	30.23
	SSIM↑	0.965	0.962	0.964	0.963	0.963	0.962
	LPIPS↓	0.036	0.039	0.037	0.039	0.038	0.039
Average	PSNR↑	35.97	30.31	31.86	31.03	31.66	30.75
	SSIM↑	0.989	0.982	0.984	0.983	0.984	0.983
	LPIPS↓	0.020	0.029	0.027	0.028	0.027	0.028

Table 14. Comparison of PSNR↑, SSIM↑, and LPIPS↓ for each scene on **geometric correction** when the number of views in a training set was six.

		PAC-NeRF	PAC-NeRF-6v	+LPO[4]	+LPO-F[4]	+LPO-P[4]	+GO[4]	+None[4]
Droplet	$\log_{10}(\mu)$	0.039	0.026	0.016	0.039	0.025	0.042	0.039
	$\log_{10}(\kappa)$	0.017	0.386	0.045	0.128	0.076	0.404	0.144
Letter	$\log_{10}(\mu)$	0.041	0.229	0.175	0.220	0.199	0.222	0.224
	$\log_{10}(\kappa)$	0.039	0.166	0.006	0.021	0.060	0.064	0.089
Cream	$\log_{10}(\mu)$	0.090	0.131	0.056	0.047	0.065	0.057	0.065
	$\log_{10}(\kappa)$	0.132	0.835	0.146	0.020	0.143	0.072	0.132
	$\log_{10}(\tau_Y)$	0.007	0.011	0.001	0.003	0.001	0.003	0.012
	$\log_{10}(\eta)$	0.015	0.287	0.049	0.060	0.075	0.088	0.134
Toothpaste	$\log_{10}(\mu)$	0.026	0.084	0.045	0.179	0.144	0.208	0.166
	$\log_{10}(\kappa)$	0.247	0.629	0.580	0.370	0.655	0.654	0.573
	$\log_{10}(\tau_Y)$	0.066	0.088	0.036	0.090	0.062	0.101	0.097
	$\log_{10}(\eta)$	0.013	0.015	0.006	0.022	0.006	0.018	0.018
Torus	$\log_{10}(E)$	0.019	0.020	0.023	0.019	0.025	0.032	0.025
	ν	0.023	0.024	0.001	0.057	0.035	0.139	0.058
Bird	$\log_{10}(E)$	0.013	0.106	0.047	0.105	0.058	0.119	1.769
	ν	0.029	0.133	0.013	0.134	0.147	0.146	0.741
Playdoh	$\log_{10}(E)$	0.286	0.483	0.252	0.311	0.282	0.310	0.289
	$\log_{10}(\tau_Y)$	0.038	0.209	0.126	0.203	0.171	0.184	0.180
	ν	0.076	0.317	0.110	0.135	0.141	0.130	0.193
Cat	$\log_{10}(E)$	0.855	2.821	0.787	0.949	0.843	2.423	0.745
	$\log_{10}(\tau_Y)$	0.026	0.978	0.008	0.005	0.024	0.393	0.020
	ν	0.027	0.071	0.017	0.065	0.026	0.218	0.026
Trophy	θ_{fric} [rad]	0.048	0.052	0.035	0.040	0.041	0.044	0.044

Table 15. Comparison of the absolute differences between the ground-truth and estimated physical properties for each scene on **physical identification** when the number of views in a training set was six. The smaller the values, the better the performance.

		PAC-NeRF	PAC-NeRF-6v	+LPO[4]	+LPO-F[4]	+LPO-P[4]	+GO[4]	+None[4]
Droplet	PSNR↑	35.29	28.92	30.72	29.64	31.26	27.66	28.73
	SSIM↑	0.989	0.982	0.986	0.984	0.986	0.981	0.983
	LPIPS↓	0.030	0.042	0.037	0.041	0.039	0.044	0.042
Letter	PSNR↑	36.99	31.42	32.70	32.69	32.66	32.13	32.25
	SSIM↑	0.992	0.985	0.989	0.989	0.988	0.987	0.987
	LPIPS↓	0.011	0.020	0.017	0.016	0.018	0.018	0.018
Cream	PSNR↑	36.46	30.36	32.11	31.78	31.48	30.65	30.34
	SSIM↑	0.993	0.986	0.988	0.988	0.987	0.987	0.987
	LPIPS↓	0.014	0.021	0.018	0.019	0.020	0.019	0.020
Toothpaste	PSNR↑	38.84	34.74	34.91	34.44	35.35	33.52	34.84
	SSIM↑	0.996	0.993	0.994	0.994	0.994	0.993	0.994
	LPIPS↓	0.006	0.009	0.009	0.009	0.009	0.009	0.009
Torus	PSNR↑	34.60	29.28	32.33	31.57	31.94	30.64	29.58
	SSIM↑	0.988	0.979	0.986	0.984	0.986	0.982	0.982
	LPIPS↓	0.026	0.035	0.028	0.031	0.030	0.032	0.034
Bird	PSNR↑	35.70	27.38	30.57	28.27	29.67	27.84	25.96
	SSIM↑	0.992	0.981	0.985	0.981	0.983	0.981	0.983
	LPIPS↓	0.019	0.029	0.028	0.029	0.028	0.030	0.035
Playdoh	PSNR↑	36.55	29.74	30.26	28.83	30.90	29.03	29.65
	SSIM↑	0.991	0.982	0.984	0.981	0.984	0.982	0.984
	LPIPS↓	0.026	0.039	0.037	0.041	0.037	0.040	0.039
Cat	PSNR↑	37.10	30.77	31.45	31.14	31.11	29.95	31.79
	SSIM↑	0.993	0.989	0.988	0.989	0.988	0.987	0.989
	LPIPS↓	0.016	0.024	0.024	0.024	0.026	0.025	0.023
Trophy	PSNR↑	32.23	30.22	30.44	29.69	30.64	29.20	30.23
	SSIM↑	0.965	0.962	0.963	0.962	0.963	0.962	0.963
	LPIPS↓	0.036	0.039	0.037	0.039	0.038	0.040	0.039
Average	PSNR↑	35.97	30.31	31.72	30.89	31.67	30.07	30.37
	SSIM↑	0.989	0.982	0.985	0.983	0.984	0.982	0.983
	LPIPS↓	0.020	0.029	0.026	0.028	0.027	0.029	0.029

Table 16. Comparison of PSNR↑, SSIM↑, and LPIPS↓ for each scene on **geometric recorrection** when the number of views in a training set was six.

D. Qualitative results

This appendix discusses the qualitative results obtained using several frames selected from the video sequences. We provide video samples at https://www.kecl.ntt.co.jp/people/kaneko.takuhiro/projects/lpo/.

D.1. Qualitative comparisons among PAC-NeRF-3v/3v†, +LPO, and +LPO^{4}

- Figure 3:
  Qualitative comparisons among PAC-NeRF-3v/3v†, +LPO, and +LPO^{4} on Newtonian fluids (Droplet and Letter).

- Figure 4:
  Qualitative comparisons among PAC-NeRF-3v/3v†, +LPO, and +LPO^{4} on non-Newtonian fluids (Cream and Toothpaste).

- Figure 5:
  Qualitative comparisons among PAC-NeRF-3v/3v†, +LPO, and +LPO^{4} on elasticity materials (Torus and Bird).

- Figure 6:
  Qualitative comparisons among PAC-NeRF-3v/3v†, +LPO, and +LPO^{4} on plasticine materials (Playdoh and Cat).

- Figure 7:
  Qualitative comparisons among PAC-NeRF-3v/3v†, +LPO, and +LPO^{4} on granular media (Trophy).

D.2. Qualitative comparisons among +LPO^{4}, +LPO-F^{4}, +LPO-P^{4}, +GO^{4}, and +None^{4}

- Figure 8:
  Qualitative comparisons among +LPO^{4}, +LPO-F^{4}, +LPO-P^{4}, +GO^{4}, and +None^{4} in the scenes where +LPO-P^{4} outperformed +LPO-F^{4}.

- Figure 9:
  Qualitative comparisons among +LPO^{4}, +LPO-F^{4}, +LPO-P^{4}, +GO^{4}, and +None^{4} in the scenes where +LPO-F^{4} outperformed +LPO-P^{4}.

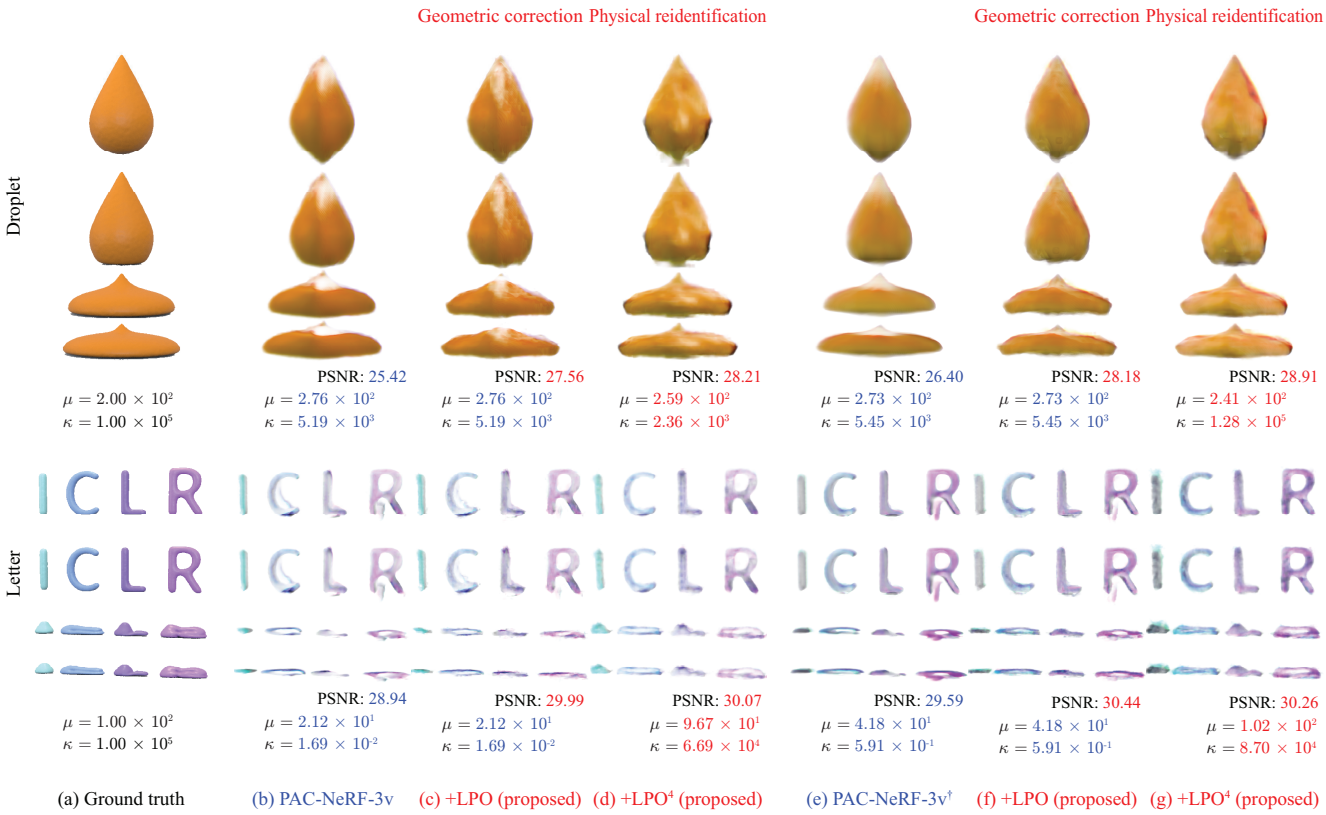

Figure 3. Qualitative comparisons among PAC-NeRF-3v/3v†, +LPO, and +LPO^4 on Newtonian fluids (Droplet and Letter). Blue fonts indicate the scores obtained by the baselines (PAC-NeRF-3v/3v†). Red fonts indicate the scores obtained by the proposed methods (+LPO and +LPO^4). Given the initial estimation by the baseline (b)(e), +LPO first corrects the geometric structures (including appearance and shape) (c)(f). By repeatedly conducting physical identification and geometric correction via Algorithm 1, +LPO^4 reidentifies physical properties and recorrects geometric structures (d)(g). In the Droplet scene, the bottom of the droplet is sharply pointed, and its tip is whitened in the baseline (b)(e). They are gradually mitigated by applying +LPO (c)(f) and +LPO^4 (d)(g). In the Letter scene, +LPO (c)(f) and +LPO^4 (d)(g) succeed in gradually eliminating artifacts existing in the vicinity of the left line of the letter "R.", which arise in the baselines (b)(e).

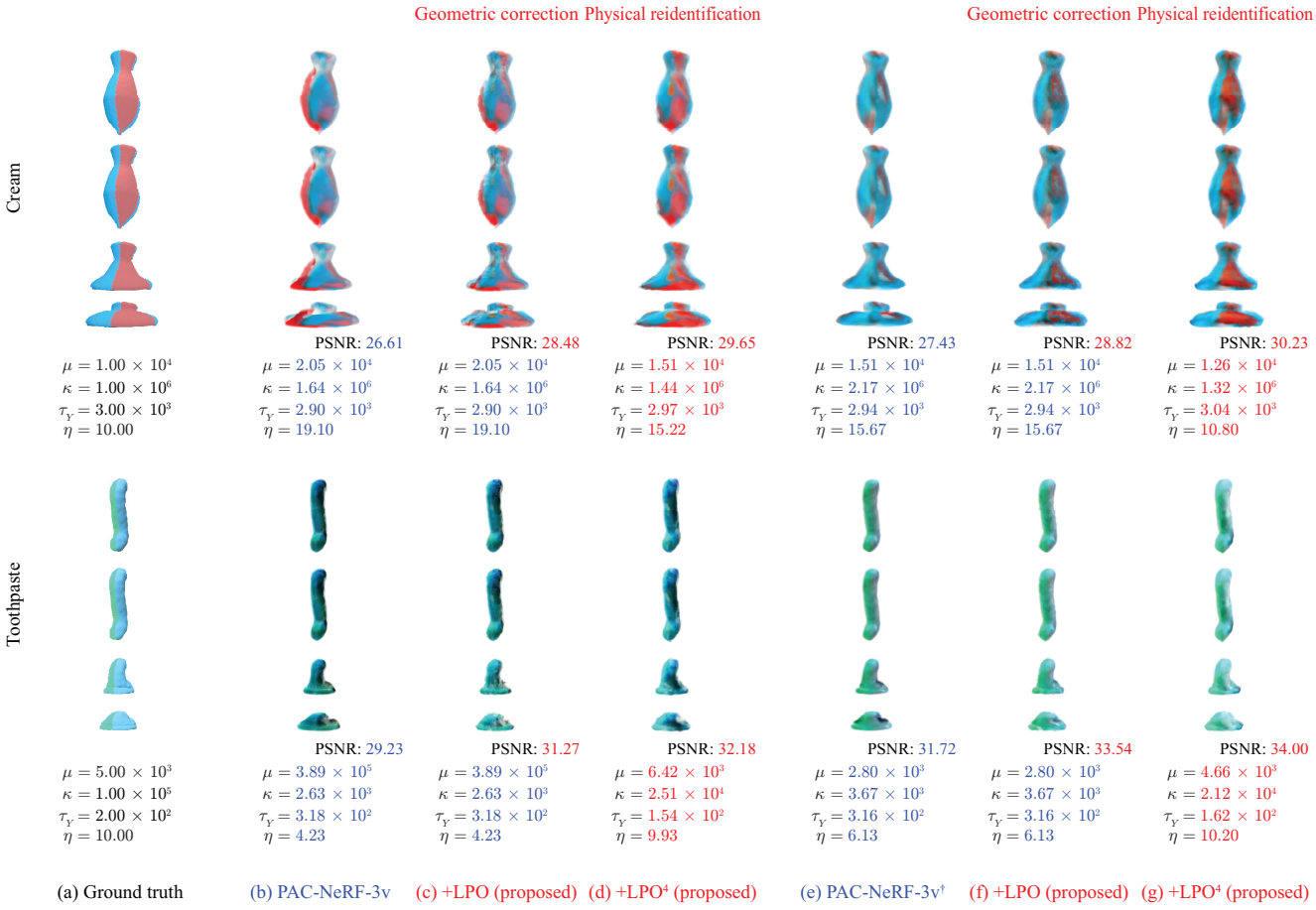

Figure 4. Qualitative comparisons among PAC-NeRF-3v/3v†, +LPO, and +LPO^4 on non-Newtonian fluids (Cream and Toothpaste). Blue fonts indicate the scores obtained by the baselines (PAC-NeRF-3v/3v†). Red fonts indicate the scores obtained by the proposed methods (+LPO and +LPO^4). Given the initial estimation by the baseline (b)(e), +LPO first corrects the geometric structures (including appearance and shape) (c)(f). By repeatedly conducting physical identification and geometric correction via Algorithm 1, +LPO^4 reidentifies physical properties and recorrects geometric structures (d)(g). In the Cream scene, the baselines (b)(e) fail to color the materials correctly. This failure is alleviated by +LPO (c)(f) and further mitigated by +LPO^4 (d)(g). In the Toothpaste scene, the baselines (b)(e) make the material darker color than that in the ground truth (a). +LPO (c)(f) makes the material brighter, and +LPO^4 (d)(g) obtains the color closer to the ground truth (a). This effect is pronounced when PAC-NeRF-3v† (e) is used as a baseline.

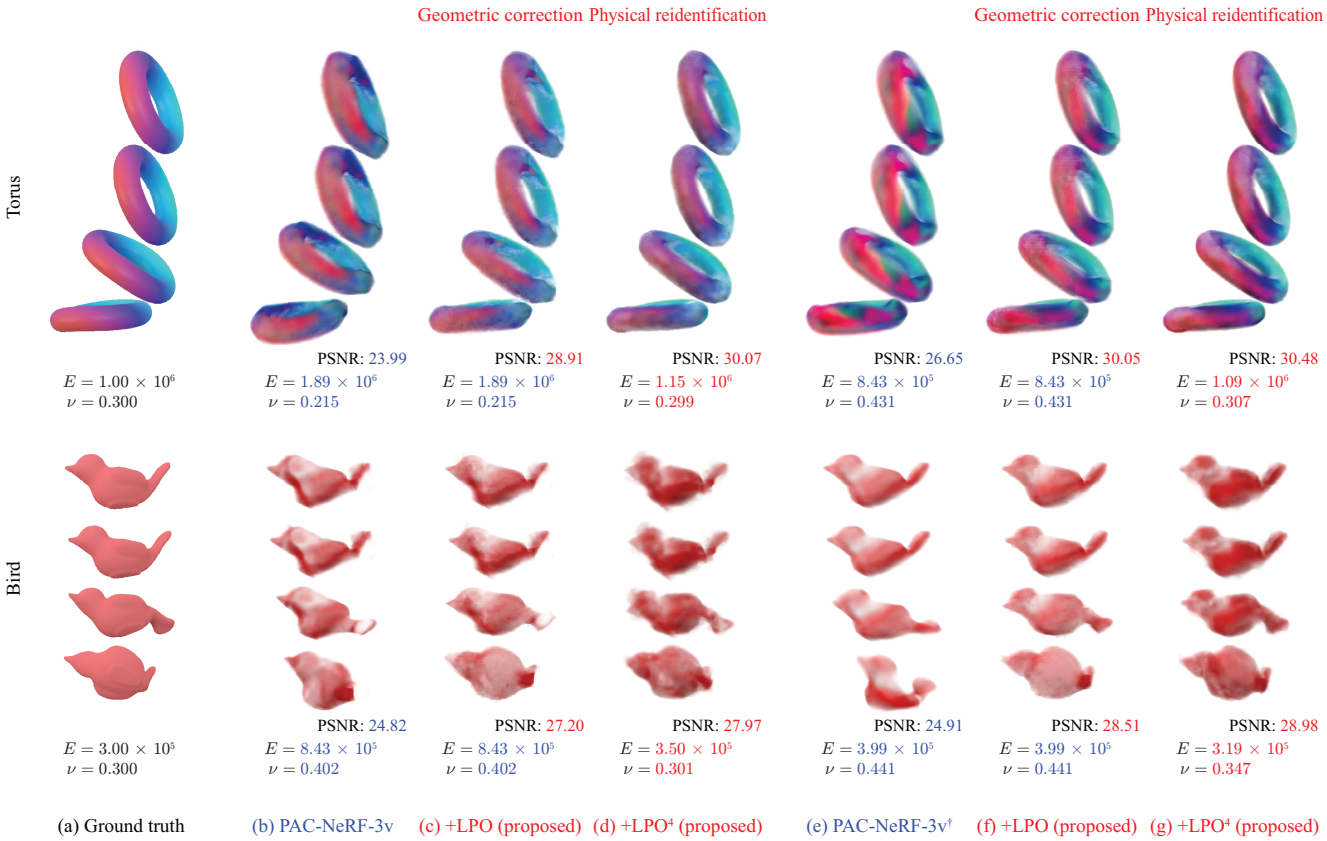

Figure 5. Qualitative comparisons among PAC-NeRF-3v/3v†, +LPO, and +LPO^4 on elasticity materials (Torus and Bird). Blue fonts indicate the scores obtained by the baselines (PAC-NeRF-3v/3v†). Red fonts indicate the scores obtained by the proposed methods (+LPO and +LPO^4). Given the initial estimation by the baseline (b)(e), +LPO first corrects the geometric structures (including appearance and shape) (c)(f). By repeatedly conducting physical identification and geometric correction via Algorithm 1, +LPO^4 reidentifies physical properties and recorrects geometric structures (d)(g). In the Torus scene, the baselines (b)(e) have difficulty correctly capturing color and shape. They are improved by applying +LPO (c)(f), and the fine details are also improved by utilizing +LPO^4 (d)(g). Also, in the Bird scene, the baselines (b)(e) fail to capture color and shape correctly. The shape (e.g., the directions of the tail) is first corrected by +LPO (c)(f), and then the color is corrected by +LPO^4 (d)(g).

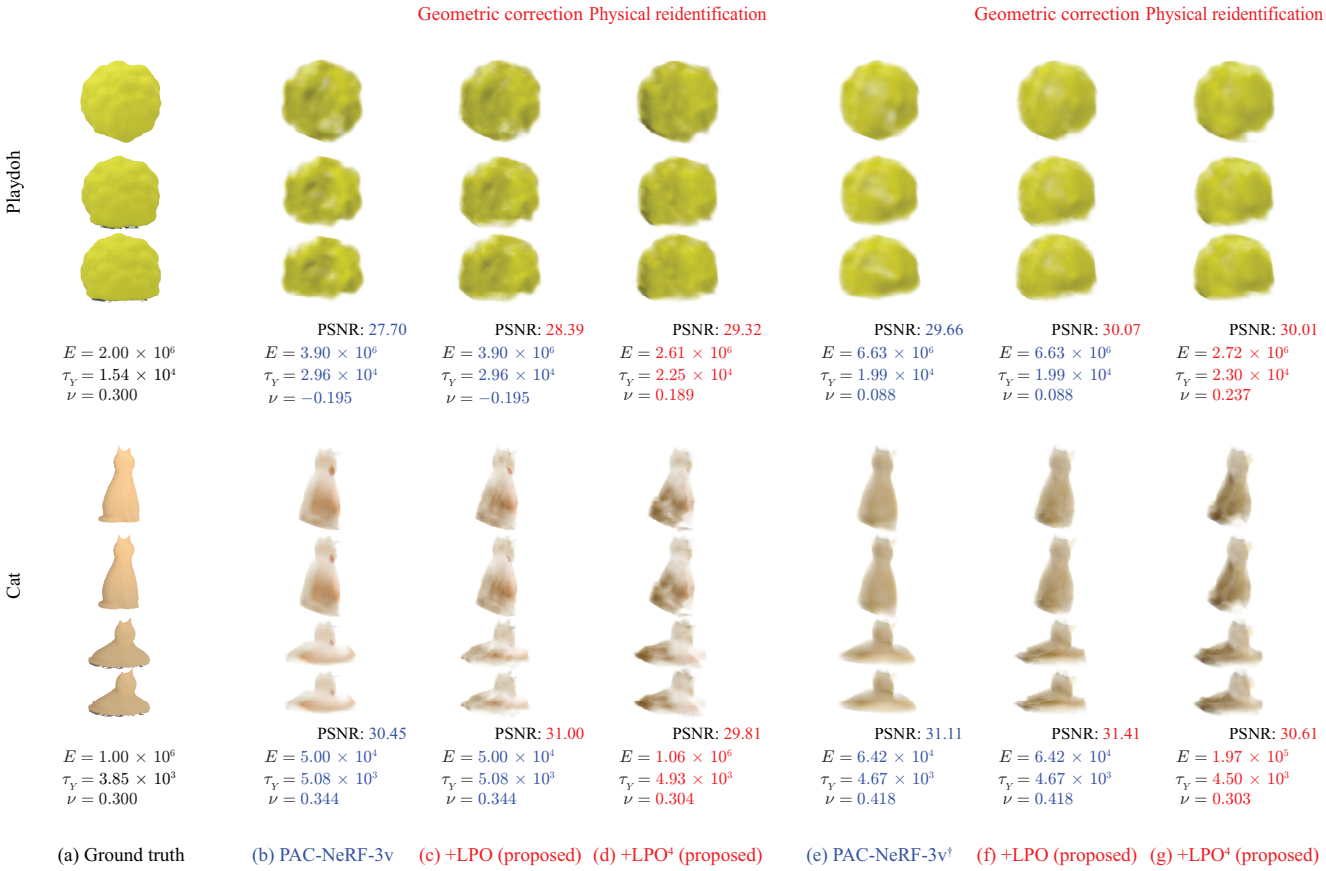

Figure 6. Qualitative comparisons among PAC-NeRF-3v/3v†, +LPO, and +LPO^4 on plasticine materials (Playdoh and Cat). Blue fonts indicate the scores obtained by the baselines (PAC-NeRF-3v/3v†). Red fonts indicate the scores obtained by the proposed methods (+LPO and +LPO^4). Given the initial estimation by the baseline (b)(e), +LPO first corrects the geometric structures (including appearance and shape) (c)(f). By repeatedly conducting physical identification and geometric correction via Algorithm 1, +LPO^4 reidentifies physical properties and recorrects geometric structures (d)(g). In the Playdoh scene, the baselines (b)(e) make the playdoh a little crushed compared to the ground truth (a). These geometric failures are gradually alleviated by applying +LPO (c)(f) and +LPO^4 (d)(g). In the Cat scene, the baselines (b)(e) fail to capture the tail of the cat in the lower left corner. +LPO (c)(f) and +LPO^4 (d)(g) struggle to recover it.

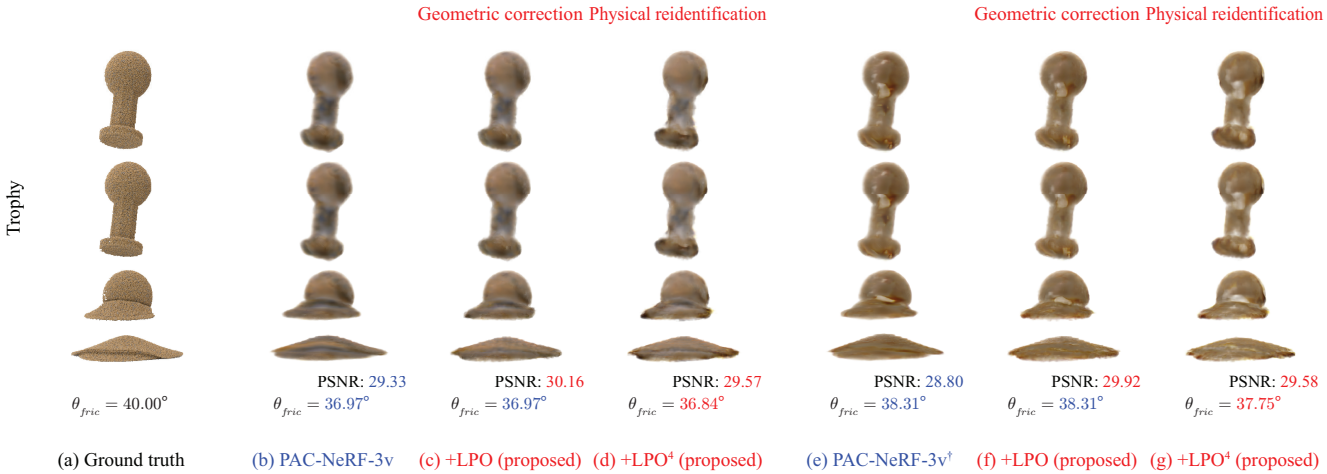

Figure 7. Qualitative comparisons among PAC-NeRF-3v/3v†, +LPO, and +LPO^4 on granular media (Trophy). Blue fonts indicate the scores obtained by the baselines (PAC-NeRF-3v/3v†). Red fonts indicate the scores obtained by the proposed methods (+LPO and +LPO^4). Given the initial estimation by the baseline (b)(e), +LPO first corrects the geometric structures (including appearance and shape) (c)(f). By repeatedly conducting physical identification and geometric correction via Algorithm 1, +LPO^4 reidentifies physical properties and recorrects geometric structures (d)(g). In the Trophy scene, the baselines (b)(e) achieve good performance in terms of physical identification (comparable with PAC-NeRF with full-view supervision). However, they tend to make the trophy darker than the ground truth (a). This misestimation is corrected by +LPO and +LPO^4.

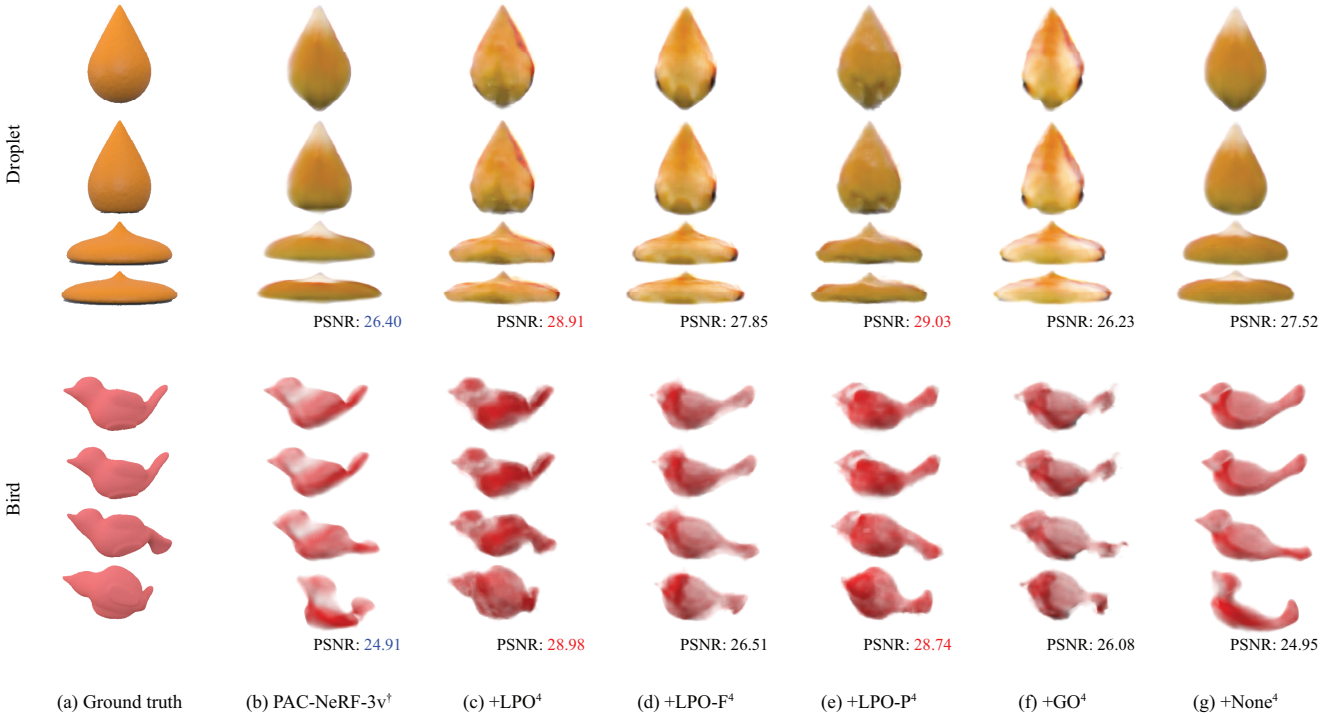

Figure 8. Qualitative comparisons among +LPO^4, +LPO-F^4, +LPO-P^4, +GO^4, and +None^4 in the scenes where +LPO-P^4 outperformed +LPO-F^4 in terms of PSNR. In the +LPO-F^4, position (that is, shape) optimization was ablated, and only feature (that is, appearance) optimization was conducted. In the +LPO-P^4, feature (that is, appearance) optimization was ablated, and only position (that is, shape) optimization was conducted. In the above scenes, large geometric gaps exist between the ground truth (a) and PAC-NeRF-3v† (b). For example, in the second row of the Droplet scene, the bottom of the droplet is flat in the ground truth (a), while that is bulging in PAC-NeRF-3v† (b). As shown in (d), only appearance correction by +LPO-F^4 is insufficient to correct this geometry failure estimation, and the bottom of the droplet is still bulging. Instead, +LPO-F^4 attempts to solve this problem by changing the appearance, making the colors overcorrected. In contrast, shape correction by +LPO-P^4 effectively addresses this failure, and the bottom of the droplet is flat in (e). The same correction was also conducted in +LPO^4 (c), a combination of +LPO-F^4 and +LPO-P^4. Similarly, in the Bird scene, the directions of the tail are adequately corrected in +LPO^4 (c) and +LPO-P^4 (e). In contrast, those are not sufficiently conducted in +LPO-F^4 (d). Also, in this case, +LPO-F^4 (d) attempts to solve this problem by overcorrecting the colors. +GO^4 (f), which also only corrects appearance, has the same difficulty as +LPO-F^4 (d). In the Droplet scene, the bottom of the droplet is bulging; in the Bird scene, the bird's tail is corrupted. +None^4 (g), which performs Algorithm 1 without geometry correction, does not have a sufficient correction ability. In the Droplet scene, shape and appearance are almost identical to those in PAC-NeRF-3v† (b). In the Bird scene, the pose of the bird is not corrected.

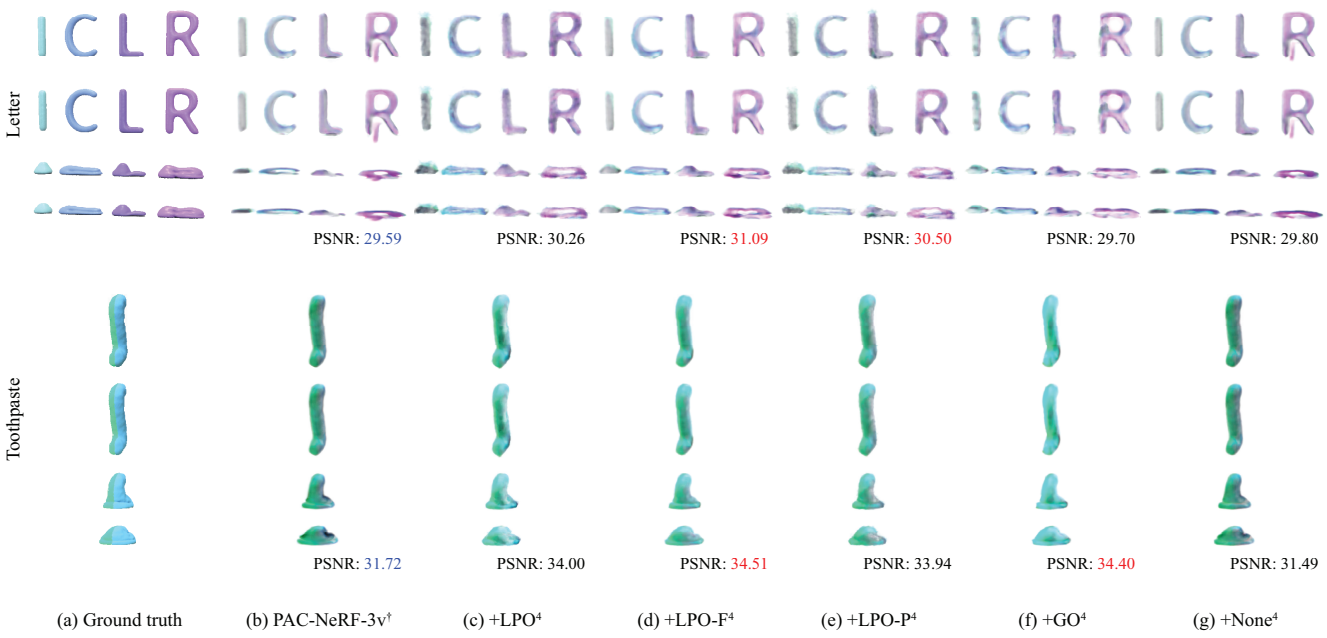

Figure 9. Qualitative comparisons among +LPO^4, +LPO-F^4, +LPO-P^4, +GO^4, and +None^4 in the scenes where +LPO-F^4 outperformed +LPO-P^4 in terms of PSNR. In the +LPO-F^4, position (that is, shape) optimization was ablated, and only feature (that is, appearance) optimization was conducted. In the +LPO-P^4, feature (that is, appearance) optimization was ablated, and only position (that is, shape) optimization was conducted. As discussed in Figure 8, +LPO-F^4 is unsuitable for significantly correcting the geometric shape because it focuses on correcting appearance. However, as shown in the above scenes, when PAC-NeRF-3v† (b) captures the geometric structure relatively well, +LPO-F^4 (d) also works well because appearance correction is more important than shape correction. For example, in the Letter scene, +LPO-F^4 (d) succeeds in eliminating artifacts existing in the vicinity of the left line of the letter "R." In the Toothpaste scene, +LPO-F^4 (d) succeeds in making the color of the material brighter and closer to the ground truth (a). Notably, in both scenes, +None^4 (g) fails to do so and produces almost the same results as those in PAC-NeRF-3v† (b). These results indicate that appearance correction by +LPO-F^4 is essential (simple iterative updates in Algorithm 1 are insufficient) to address these problems. Another interesting finding is that +LPO-P^4 (e), which focuses on correcting shape, also works well. For example, in the Letter scene, +LPO-P^4 (e) eliminates the artifacts existing in the vicinity of the left line of the letter "R," and in the Toothpaste scene, it makes the material brighter. This is possible because moving correctly colored particles from other places allows for appearance changes.

E. Implementation details

E.1. Dataset

We investigated the benchmark performance on the dataset provided by the original study on PAC-NeRF [6]. This dataset comprised nine scenes and various continuum materials, including the following:

- Newtonian fluids with fluid viscosity μ and bulk modulus κ:
  - *Droplet* with $\mu = 200$ and $\kappa = 10^5$
  - *Letter* with $\mu = 100$ and $\kappa = 10^5$
- Non-Newtonian fluids with shear modulus ν, bulk modulus κ, yield stress τ_Y, and plasticity viscosity η:
  - *Cream* with $\nu = 10^4$, $\kappa = 10^6$, $\tau_Y = 3 \times 10^3$, and $\eta = 10$
  - *Toothpaste* with $\nu = 5 \times 10^3$, $\kappa = 10^5$, $\tau_Y = 200$, and $\eta = 10$
- Deformable solid with Young's modulus E and Poisson's ratio ν:
  - *Torus* with $E = 10^6$ and $\nu = 0.3$
  - *Bird* with $E = 3 \times 10^5$ and $\nu = 0.3$
- Plasticine with Young's modulus E, Poisson's ratio ν, and yield stress τ_Y:
  - *Cat* with $E = 2 \times 10^6$, $\nu = 0.3$, and $\tau_Y = 1.54 \times 10^4$
  - *Playdoh* with $E = 10^6$, $\nu = 0.3$, and $\tau_Y = 3.85 \times 10^3$
- Granular media with friction angle θ_{fric}:
  - *Trophy* with $\theta_{fric} = 40°$

In each scene, the objects fall freely under the influence of gravity and undergo collisions. The ground-truth simulation data were generated using the MLS-MPM framework [2]. A photorealistic simulation engine rendered objects under diverse environmental lighting conditions and ground textures. Each scene was captured from 11 viewpoints with cameras evenly spaced on the upper hemisphere, including the object. To evaluate our method in sparse-view settings, three views were used for training, and the remaining eight views were used for testing in the main experiments (Section 4) and the experiments described in Appendix C.1. Six views were used for training, and the remaining five were used for testing in the experiments described in Appendix C.2. Data were downloaded from the website[14] provided by the authors of PAC-NeRF [6].

E.2. Model

The models were implemented based on the official PAC-NeRF source code.[14] For simplicity and fair comparison, we used the default model configurations provided in the source code for the experiments. Specifically, the architecture of a discretized voxel-based NeRF followed direct voxel grid optimization [8], in which a volume density field σ^G (Equation (7)) and color field $\mathbf{c}^G$ (Equation (8)) were represented by voxel grids, and a 2-layer MLP with a hidden dimension of 128 was applied to $\mathbf{c}^G$ with positional embedding for a view direction $\mathbf{d}$. Regarding a differentiable MPM, DiffTaichi [3] was used.

[14] https://github.com/xuan-li/PAC-NeRF

E.3. Training settings

For simplicity and fair comparison, we conducted Eulerian static voxel grid optimization (Figure 2(1)) and physical property optimization (Figure 2(2)) using the default training settings provided in the source code[14], in PAC-NeRF, PAC-NeRF-3v, and PAC-NeRF-6v, except that the number of views was changed. In PAC-NeRF-3v†, we applied the three modifications described in Appendix A (scheduling of $\mathcal{L}_{surf}$, introduction of $\mathcal{L}^{VI}_{pixel}$, and adjustment of training length) to the Eulerian static voxel grid optimization and adopted the default training settings for physical property optimization. For the LPO (Figure 2(3)), we trained the features and positions of the particles for 100 iterations using the Adam optimizer [5]. In particular, we optimized the features of the particles at a learning rate of 0.1, which is the default value used for training the feature grids in the Eulerian static voxel grid optimization. The positions of the particles were optimized at a learning rate of $\frac{\mathrm{d}x}{32}$, where $\mathrm{d}x$ indicates the voxel grid size and differs depending on the scene (set in the configuration files in the source code[14]). In a preliminary experiment, we found that careful setting of this learning rate is vital for stable training because the particles can diverge when the learning rate is exceptionally high. Based on this observation, we use a learning rate adjusted according to $\mathrm{d}x$. We set the momentum terms of the Adam optimizer, β_1 and β_2, to 0.9 and 0.999, respectively.

E.4. Evaluation metrics

Evaluation of geometric (re)correction. We evaluated the performance of the geometric (re)correction using metrics commonly used to assess the performance of novel view synthesis in NeRF studies: the peak signal-to-noise ratio (*PSNR*), structural similarity index (*SSIM*) [9], and learned perceptual image patch similarity (*LPIPS*) [11]. For PSNR and SSIM, the larger the values, the better the performance. For LPIPS, the smaller the values, the better the performance. In particular, we report the scores averaged over the video sequences in a test set.

Evaluation of physical identification. To evaluate the performance of the physical identification, we measured the absolute distance between the ground truth and the estimated physical properties. The values of the ground-truth physical properties are provided in Appendix E.1. For an easy comparison, we calculated these distances after adjusting the scale (i.e., either a logarithmic scale or a linear scale) following the study of PAC-NeRF [6]. The smaller the values, the better the performance.

References

[1] Mengyu Chu, Lingjie Liu, Quan Zheng, Erik Franz, Hans-Peter Seidel, Christian Theobalt, and Rhaleb Zayer. Physics informed neural fields for smoke reconstruction with sparse data. *ACM Trans. Graph.*, 41(4), 2022. 2

[2] Yuanming Hu, Yu Fang, Ziheng Ge, Ziyin Qu, Yixin Zhu, Andre Pradhana, and Chenfanfu Jiang. A moving least squares material point method with displacement discontinuity and two-way rigid body coupling. *ACM Trans. Graph.*, 37(4), 2018. 20

[3] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. DiffTaichi: Differentiable programming for physical simulation. In *ICLR*, 2020. 20

[4] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting NeRF on a diet: Semantically consistent few-shot view synthesis. In *ICCV*, 2021. 1

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 20

[6] Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. PAC-NeRF: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. In *ICLR*, 2023. 1, 9, 20

[7] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[8] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2, 20

[9] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4), 2004. 20

[10] Jiawei Yang, Marco Pavone, and Yue Wang. FreeNeRF: Improving few-shot neural rendering with free frequency regularization. In *CVPR*, 2023. 1, 2

[11] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 20