Toward Material-Agnostic System Identification from Videos

Supplementary Material

A. Background

A.1. 3D Gaussian Splatting (3DGS)

3DGS [34] represents a scene using a set of 3D Gaussian kernels $\mathcal{G} = \{\boldsymbol{\mu}(i), \boldsymbol{\Sigma}(i), \mathbf{c}(i), \sigma(i)\}$, parameterized with center coordinates $\boldsymbol{\mu}(i) \in \mathbb{R}^3$, covariance matrix $\boldsymbol{\Sigma}(i) \in \mathbb{R}^{3\times3}$ defining the shape and orientation of the Gaussian, color attribute $\mathbf{c}(i)$, and the opacity $\sigma(i)$. Each point is denoted as

$$\mathcal{G}(\mathbf{x}) = \exp(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}(i))^T\boldsymbol{\Sigma}(i)^{-1}(\mathbf{x}-\boldsymbol{\mu}(i))), \quad (13)$$

where the covariance matrix can be factorized into rotation and scaling components as $\boldsymbol{\Sigma}(i) = \mathbf{R}(i)\mathbf{S}(i)\mathbf{S}(i)^T\mathbf{R}(i)^T$. A rotation matrix $\mathbf{R}(i)$ is represented by a quaternion vector $\mathbf{q}(i) \in \mathbb{R}^4$, and a diagonal scaling matrix $\mathbf{S}(i)$ is characterized by $\mathbf{s}(i) \in \mathbb{R}^3$ which defines the anisotropic spread of the Gaussian. During rendering, each 3D Gaussian is projected onto the 2D image plane, and an image is rendered via point-based alpha blending, where the rendered color $\mathbf{I}$ and the corresponding foreground mask $\mathbf{M}$ at pixel $\mathbf{u}$ are

$$\mathbf{I}(\mathbf{u}) = \sum_{i=1}^{G} T(i)\sigma(i)\mathbf{c}(i), \quad (14)$$

$$\mathbf{M}(\mathbf{u}) = \sum_{i=1}^{G} T(i)\sigma(i). \quad (15)$$

Here, $T(i) = \prod_{j=1}^{i-1}(1-\sigma(j))$ is the accumulated transmittance at point i, and G is the number of ordered Gaussians.

A.2. Material Point Method (MPM)

MPM [31, 77] is a hybrid Eulerian-Lagrangian approach used for simulating materials with complex dynamics, including elastic and plastic deformations. MPM represents the simulated object using a set of discrete particles, which store material properties such as mass, velocity, and deformation gradients. These particles interact with a background grid where forces are computed, and the governing equations of motion are solved. The method consists of four key steps: particle-to-grid transfer, solving the governing equations on the grid to do the grid update, grid-to-particle transfer, and updating particle states.

1. **Initialization.**

   Each material point $i = 1, 2, ..., Q$, where Q is the total number of material points, at every time step t is initialized by defining its position $\mathbf{x}_i^t$, velocity $\mathbf{v}_i^t$, deformation gradient $\mathbf{F}_i^t$, and affine velocity field $\mathbf{C}_i^t$. The initial values are set as $\mathbf{x}_i^0 = \mathbf{x}_i, \mathbf{v}_i^0 = \mathbf{0}, \mathbf{F}_i^0 = \mathbf{I}, \mathbf{C}_i^0 = \mathbf{0}$. And for each grid node $g \in \mathcal{G}$ at every time step t, we define its grid mass m_g^t and grid velocity $\mathbf{v}_g^t$. The initial values are set as $m_g^0 = 0, \mathbf{v}_g^0 = \mathbf{0}$.

2. **Stress update.**

   The stress state of each material point is determined by its constitutive model. The first Piola-Kirchhoff stress tensor is computed as:

   $$\mathbf{P}_i^t = \frac{\partial \Psi_i}{\partial \mathbf{F}_i^t}, \quad \forall i \in Q. \quad (16)$$

   where $\mathbf{P}_i^t$ represents the stress tensor, and Ψ_i is the strain energy function describing the material's response to deformation.

3. **Particle-to-grid transfer.**

   Mass and momentum are transferred between particles and grid nodes, according to the interpolation functions. We denote the interpolation weight of particle i at grid node g as $\omega_{ig}^t \in \mathbb{R}$. The mass contribution at a grid node is computed as:

   $$m_g^t = \sum_{i\in Q} \omega_{ig}^t m_i, \quad \forall g \in \mathcal{G}. \quad (17)$$

   The corresponding momentum at grid nodes is computed as:

   $$m_g^t \mathbf{v}_g^t = \sum_{i\in Q} \omega_{ig}^t m_i \left(\mathbf{v}_i^t + \mathbf{C}_i^t(\mathbf{x}_g - \mathbf{x}_i^t)\right). \quad (18)$$

   where $\mathbf{C}_i^t = \mathbf{B}_i^t(\mathbf{D}_i^t)^{-1}$, $\mathbf{B}_i^t$ is a matrix quantity stored at each particle (similar to mass, position, and velocity), and $\mathbf{D}_i^t$ is an inertia-like tensor.
   The inertia-like tensor $\mathbf{D}_i^t$ is defined as:

   $$\mathbf{D}_i^t = \sum_{g\in\mathcal{G}} \omega_{ig}^t(\mathbf{x}_g - \mathbf{x}_i^t)(\mathbf{x}_g - \mathbf{x}_i^t)^T, \quad (19)$$

   which accounts for the distribution of particles' positions relative to the particle i. The matrix $\mathbf{B}_i^t$, is given by:

   $$\mathbf{B}_i^t = \sum_{g\in\mathcal{G}} \omega_{ig}^t \mathbf{v}_g^t(\mathbf{x}_g - \mathbf{x}_i^t)^T. \quad (20)$$

4. **Grid update.**

After transferring mass and momentum, the governing equations of motion are solved to update grid velocities. The velocity at each grid node is updated as:

$$\mathbf{v}_g^{t+1} = \mathbf{v}_g^t - \frac{\Delta t}{m_g} \sum_{i \in Q} \mathbf{P}_i \nabla \omega_{ig}^t V_i + \Delta t \mathbf{f}. \tag{21}$$

where V_i represents the material point volume, and $\mathbf{f}$ accounts for external forces such as gravity.

5. **Grid-to-particle transfer.**

After computing the grid velocities, the updated values are mapped back to the material points. The velocity of a material point is obtained as:

$$\mathbf{v}_i^{t+1} = \sum_{g \in \mathcal{G}} \omega_{ig}^t \mathbf{v}_g^{t+1}. \tag{22}$$

The new position of each material point is updated using:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \Delta t \mathbf{v}_i^{t+1}. \tag{23}$$

To ensure a consistent velocity field, the affine velocity field is updated using:

$$\mathbf{C}_i^{t+1} = \frac{12}{(b+1)\Delta x^2} \sum_{g \in \mathcal{G}} \omega_{ig}^t \mathbf{v}_g^{t+1} (\mathbf{x}_g - \mathbf{x}_i^t)^\top. \tag{24}$$

where b is the B-spline degree, and Δx is the grid spacing.

The velocity gradient is computed as:

$$\nabla \mathbf{v}_i^{t+1} = \sum_{g \in \mathcal{G}} \mathbf{v}_g^{t+1} \nabla \omega_{ig}^t{}^\top. \tag{25}$$

The trial deformation gradient is then updated as:

$$\mathbf{F}_{i,\text{trial}}^{t+1} = (\mathbf{I} + \Delta t \nabla \mathbf{v}_i^{t+1}) \mathbf{F}_i^t. \tag{26}$$

6. **Plasticity correction.**

The trial deformation gradient is modified using a return mapping function to enforce material constraints:

$$\mathbf{F}_i^{t+1} = \psi_i(\mathbf{F}_{i,\text{trial}}^{t+1}), \quad \forall i \in Q. \tag{27}$$

where ψ_i ensures that the material satisfies the yield condition.

7. **Update particle positions.**

Finally, the updated positions of the material points are determined using:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \Delta t \mathbf{v}_i^{t+1}. \tag{28}$$

Pre-train	Fine-tune	Newtonian	Non-Newtonian	Elasticity	Plasticine	Sand	Mean
Jelly	✗	27.271	16.079	5.469	12.519	26.119	17.491
	✓	0.233	0.198	0.192	0.201	0.229	0.210
Plasticine	✗	2.826	1.118	24.430	1.419	2.883	6.535
	✓	0.202	0.189	4.408	0.191	0.208	1.040
Sand	✗	62.012	0.795	2.133	6.701	0.338	14.396
	✓	43.176	0.192	0.335	1.815	0.234	9.151

Table 4. Ablation study of pre-trained models on PAC-NeRF dataset. We report Chamfer Distance (CD) errors for models trained on full state data from [59] and our fine-tuned version. Fine-tuning substantially improves performance, with Jelly pre-training offering the best tradeoffs between different materials.

B. More Ablation Study

Impact of pre-trained models. In Tab. 4, we initialize the constitutive model parameters, θ^e and θ^p, of MASIV using pre-trained checkpoints trained on data with complete state observations, including position, velocity, deformation gradient, and affine momentum at each simulation time step. The results show that pre-trained models exhibit varying performance across different materials. Fine-tuning with MASIV consistently enhances performance across all scenes, demonstrating its robustness to different initializations. Unlike system identification with explicit physical parameters, which typically requires scene-specific initial guesses [6, 38], MASIV adapts effectively to diverse material behaviors. However, for challenging adaptations like Sand $\rightarrow$ Newtonian, large errors persist even after 1000 training steps. This limitation, however, opens a door for pre-training neural constitutive models capable of generalizing more effectively across diverse downstream materials. In our comparison experiments, we adopt the Jelly pre-training for the PAC-NeRF Synthetic dataset and Plasticine for both Spring-Gaus subsets, as the Jelly pre-training can be unstable in long-term roll-outs. We also observe an interesting phenomenon that the Sand pre-training performs worse than the Plasticine pre-training on Sand objects. By looking into the ground truth physical parameters of Sand objects, we conjecture that the Plasticine prior provides a smoother initialization due to its visco-plastic nature, which aligns better with the dynamic behavior of the Sand objects used for fine-tuning. In contrast, the Sand pre-training includes more brittle behaviors not featured in fine-tuning instances. This suggests that the effectiveness of pre-training may depend on the similarity of dynamic behavior, which cannot be simply measured with Chamfer Distance that focuses on global alignments rather than local motions.

Impact of silhouette loss. Tab. 5 presents an ablation study evaluating the impact of silhouette loss ($\mathcal{L}_{\text{sil}}$) on observable state simulation in the PAC-NeRF dataset. The results show that, when silhouette loss is applied, CD values significantly decrease across all materials. This improvement

$\mathcal{L}_{sil}$	Newtonian	Non-Newtonian	Elasticity	Plasticine	Sand	Mean
✗	0.410	0.412	0.238	0.500	3.071	0.926
✓	0.233	0.198	0.192	0.201	0.229	0.210

Table 5. Ablation study of silhouette loss for observable state simulation on PAC-NeRF dataset in terms of Chamfer Distance.

$\mathcal{L}_{geo}$	Newtonian	Non-Newtonian	Elasticity	Plasticine	Sand	Mean
✗	0.346	0.315	0.769	8776.226	0.519	1755.635
$\mathcal{L}_{surf}$	0.297	0.213	11500.481	6466.605	0.443	3593.608
$\mathcal{L}_{cont}$	0.259	0.226	0.418	0.225	0.411	0.308
$\mathcal{L}_{traj}$	0.282	0.219	0.281	0.210	0.477	0.294

Table 6. Ablation study of geometrical objectives for future state simulation on PAC-NeRF dataset in terms of Chamfer Distance.

(a) Elastic Egg (b) Plastic Bread (c) Liquid Cream (c) Sandy Pawn

Figure 8. **The multi-sequence dataset synthesized with Genesis [2].** It includes 4 objects in 4 multi-view sequences simulated under randomized initial locations, poses, and velocities.

demonstrates that silhouette loss effectively constrains object boundaries, leading to more accurate shape reconstructions and enhanced simulation fidelity.

Impact of different geometrical objectives. Tab. 6 evaluates the impact of geometrical objectives ($\mathcal{L}_{geo}$) on future state simulation in the PAC-NeRF dataset. Without geometrical supervision, the Plasticine category exhibits an extremely high error, indicating unstable and inaccurate predictions. Similarly, when using surface-based supervision ($\mathcal{L}_{surf}$), Elasticity and Plasticine show catastrophic errors, suggesting that relying solely on surface alignment fails to provide robust constraints for highly deformable materials. In contrast, continuum-based ($\mathcal{L}_{cont}$) and trajectory-based ($\mathcal{L}_{traj}$) supervision significantly reduce errors, with $\mathcal{L}_{traj}$ achieving the lowest mean CD. This highlights the crucial role of spatially and temporally dense geometric supervision in training neural constitutive models.

C. More Quantitative Analysis

In Tab. 7, we examine the advantages of using a data-driven constitutive model instead of predefined ones. Specifically,

	Method	# Train	Elastic Egg	Plastic Bread	Liquid Cream	Sandy Pawn	Mean
CD ↓	GIC [6]	1	2.10	0.66	0.81	0.20	0.94
		2	2.33	1.40	0.12	0.20	1.01
		3	2.25	2.98	**0.11**	0.20	1.39
	Ours	1	11.36	89.42	15.65	0.55	29.24
		2	4.68	5.42	1.81	0.15	3.02
		3	**1.78**	**0.28**	0.12	**0.11**	**0.57**
PSNR ↑	GIC [6]	1	35.24	32.29	38.38	32.34	34.56
		2	34.87	30.94	41.31	32.37	34.87
		3	**34.99**	30.45	41.52	32.38	34.83
	Ours	1	28.54	27.58	36.64	31.69	31.11
		2	30.25	28.67	38.66	33.65	32.81
		3	30.48	**33.73**	**41.62**	**35.22**	**35.26**
SSIM ↑	GIC [6]	1	**0.992**	0.992	0.992	0.991	**0.992**
		2	0.991	0.990	0.994	0.991	**0.992**
		3	**0.992**	0.988	**0.995**	0.991	0.991
	Ours	1	0.980	0.982	0.987	0.992	0.985
		2	0.985	0.985	0.992	0.993	0.989
		3	0.987	**0.993**	**0.995**	**0.994**	**0.992**

Table 7. **Inter-sequence generalization on multi-sequence dataset.** MASIV improves performance with more training data.

we generate multi-view videos for 4 objects of different materials with Genesis [2], each including 4 multi-view sequences simulated under randomized initial conditions varying in location, pose, and velocity. The multi-view cameras are set to be the same as those in the PAC-NeRF dataset. Some samples of this synthetic dataset are shown in Fig. 8. We progressively add the number of training sequences for system identification and test the estimated parameters on an unseen sequence. The results show that MASIV improves the estimation accuracy with more observations of a certain object, highlighting its data-driven advantages. In contrast, GIC [6] exhibits relatively similar performance regardless of data quantity, likely due to its reliance on fixed constitutive priors.

D. More Qualitative Comparison

We conduct extra qualitative comparison on multi-material cross-shaped subsets of the PAC-NeRF dataset, showing the results from NCLaw [59] with full-state data pre-training and our MASIV in Figs. 9 to 13. Each figure compares the results of multiple approaches, including NCLaw pre-trained on Jelly, Plasticine, or Sand, alongside our MASIV fine-tuned over NCLaw-Jelly, against the ground truth over time. From the qualitative comparisons, it is evident that MASIV consistently demonstrates improved fidelity to the ground truth across all material types. For elastic objects (Fig. 9), MASIV captures finer deformations with greater accuracy. For sandy objects (Fig. 11), MASIV better preserves granular motion. In plasticine (Fig. 10), Newtonian (Fig. 12), and non-Newtonian (Fig. 13) objects, MASIV maintains structure and flow consistency, demonstrating adaptability to diverse material properties. These results emphasize the ability of MASIV to ground intrinsic dynamics with visual clues, without the need for an initialization sufficiently close to the material of interest.

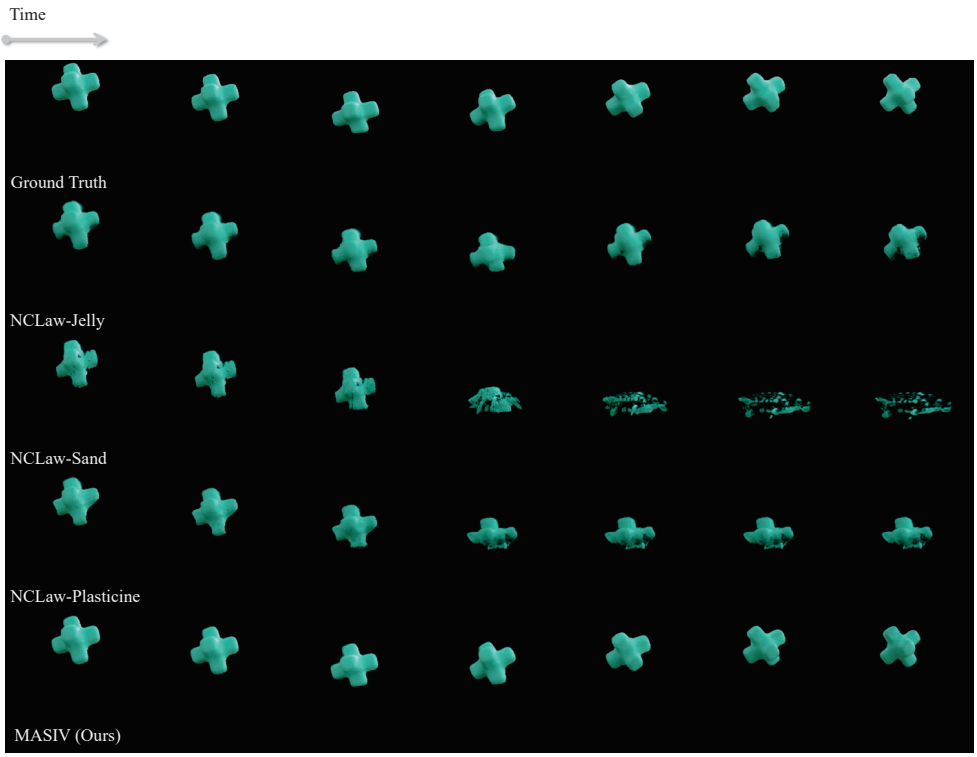

Figure 9. **Qualitative comparisons on the cross-shaped Elastic subset of the PAC-NeRF Dataset.**

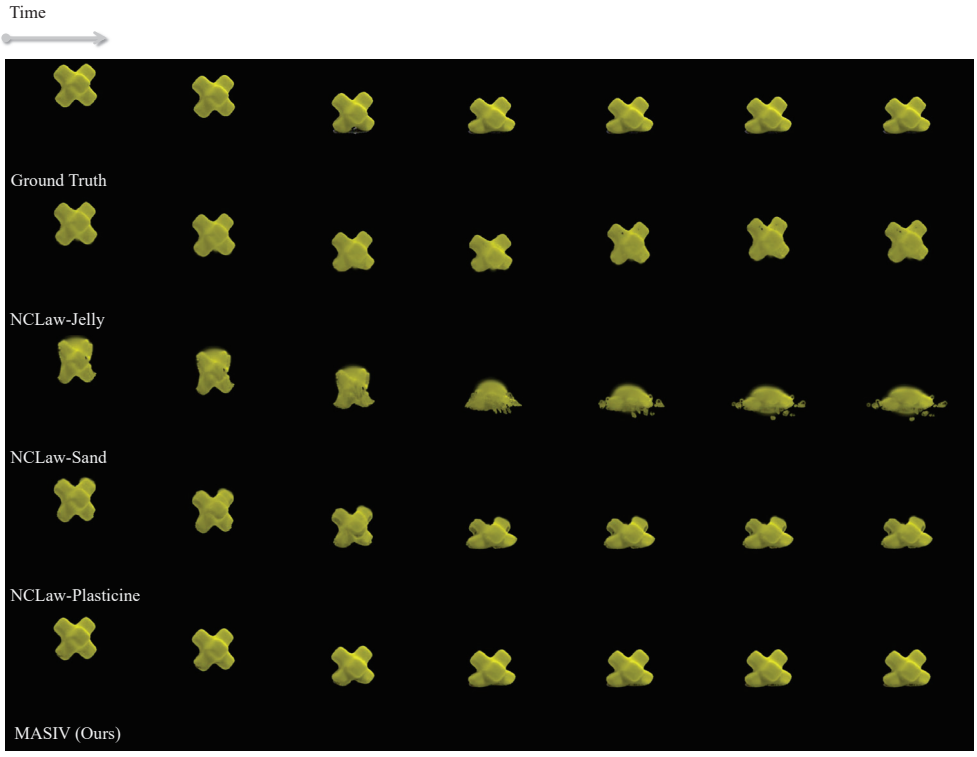

Figure 10. **Qualitative comparisons on the cross-shaped Plasticine subset of the PAC-NeRF Dataset.**

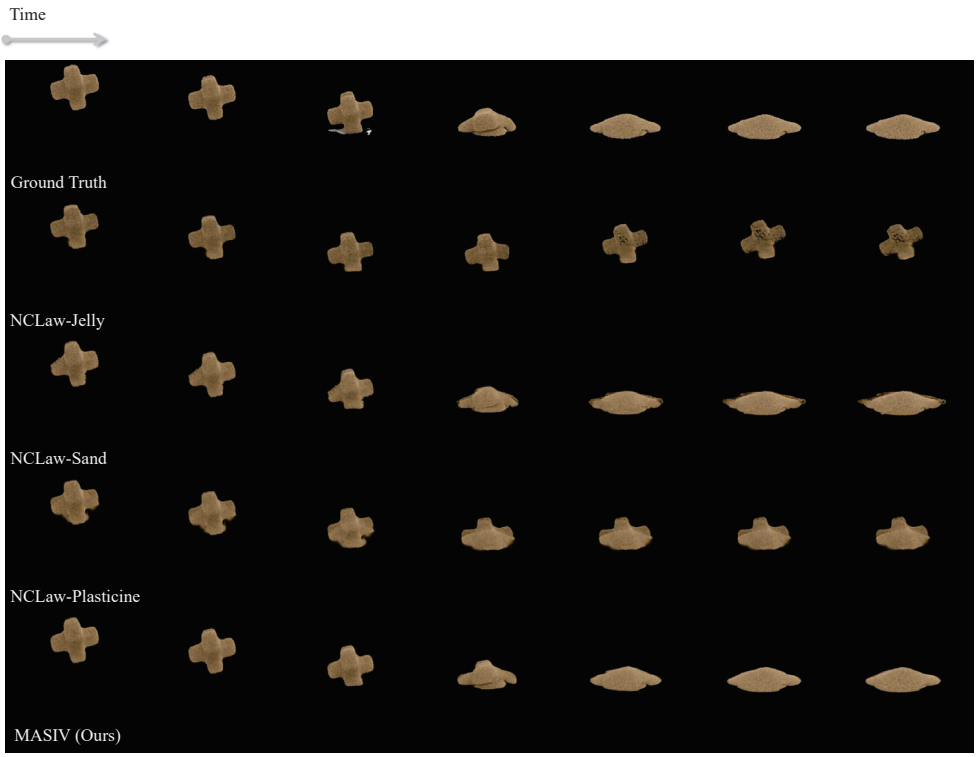

Figure 11. **Qualitative comparisons on the cross-shaped Sand subset of the PAC-NeRF Dataset.**

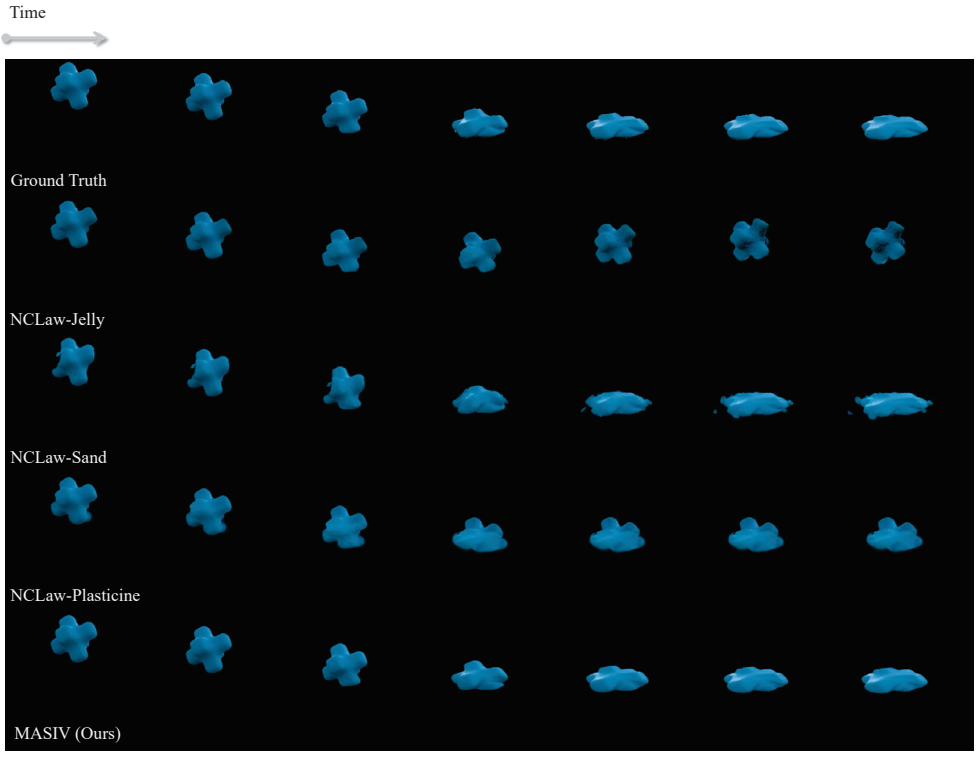

Figure 12. **Qualitative comparisons on the cross-shaped Newtonian subset of the PAC-NeRF Dataset.**

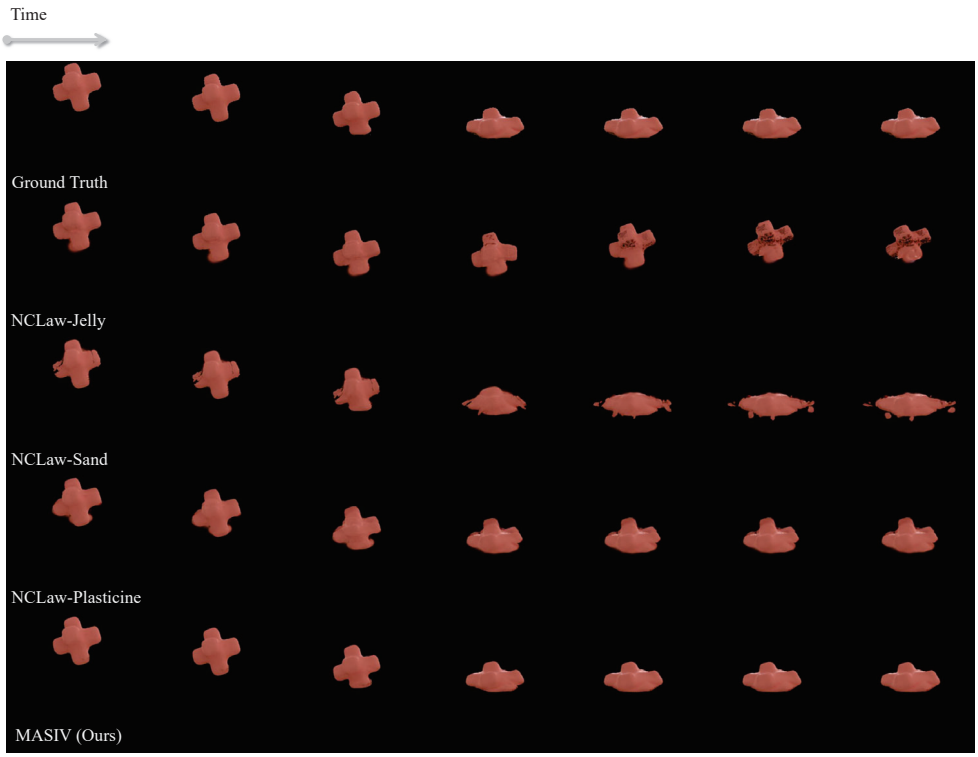

Figure 13. **Qualitative comparisons on the cross-shaped Non-Newtonian subset of the PAC-NeRF Dataset.**