# M-PhyGs: Multi-Material Object Dynamics from Video —Supplementary Material—

Norika Wada Kohei Yamashita Ryo Kawahara Ko Nishino

https://vision.ist.i.kyoto-u.ac.jp/research/m-phygs/

{nwada,kyamashita,ryo}@vision.ist.i.kyoto-u.ac.jp, kon@i.kyoto-u.ac.jp

Graduate School of Informatics, Kyoto University, Kyoto, Japan

In this supplementary material, we provide additional details of M-PhyGs and more experimental results including cross-sequence dynamics prediction. Please also see the supplementary video for qualitative results.

## A. Object Boundary Estimation

Since the voxel density computed from the 3D Gaussians is continuous, it is non-trivial to accurately estimate the object boundary from the Gaussians. M-PhyGs resolves this by refining an estimate of the object boundary during the material parameter estimation. When M-PhyGs samples uniformly distributed 3D particles from the object volume defined by a density threshold, it also samples particles from a narrow band enclosing the object boundary. For each particle, we compute a signed distance $d_i$ from the tentative object boundary and determine the particle volume $V_i$ for the MPM simulation accordingly:

$$V_i = V_g \left\{ \text{sigmoid} \left( -\beta \left( d_i + o_{s_i} \right) \right) + \epsilon \right\} , \tag{1}$$

where $V_g$ is the particle volume computed from the particle spacing, $s_i$ is the material segment to which the $i$-th particle belongs, and $o_s$ is an optimizable boundary offset for each segment $s$. We set $\beta$ to $5 \times 10^3$ and $\epsilon$ to $10^{-4}$. This ensures that particles outside the estimated object boundary have almost zero volume and do not affect the MPM simulation.

## B. Incorporating Contact-Point Motion

M-PhyGs recovers the 6D motion of the contact point from 2D annotations and the particle 3D tracks $\bar{\mathbf{x}}_i^t$ used in the computation of the 3D loss $\mathcal{L}_{\text{3D}}$. It first recovers coarse estimates of the contact point location $\tilde{\mathbf{x}}_c^t$ for each frame $t$ by triangulating manually annotated 2D locations. It then recovers refined per-frame contact point locations $\hat{\mathbf{x}}_c^t$ and per-frame contact point rotations (quaternions) $\hat{\mathbf{q}}_c^t$ by minimizing

$$\mathcal{L}_{\text{c}} = \mathcal{L}_{\text{a}} + \mathcal{L}_{\text{t}} + \lambda_{\text{ps}} \mathcal{L}_{\text{ps}} + \lambda_{\text{rs}} \mathcal{L}_{\text{rs}} , \tag{2}$$

where $\mathcal{L}_{\text{a}}$ is an annotation loss

$$\mathcal{L}_{\text{a}} = \sum_t \left\| \hat{\mathbf{x}}_c^t - \tilde{\mathbf{x}}_c^t \right\|^2 , \tag{3}$$

$\mathcal{L}_{\text{t}}$ is a tracking loss

$$\mathcal{L}_{\text{t}} = \sum_t \sum_{i \in \mathcal{N}_c} \left\| \hat{\mathbf{x}}_c^t + \text{R}_c^t \left( \bar{\mathbf{x}}_i^1 - \hat{\mathbf{x}}_c^1 \right) - \bar{\mathbf{x}}_i^t \right\|^2 , \tag{4}$$

$\mathcal{L}_{\text{ps}}$ is a position smoothness loss

$$\mathcal{L}_{\text{ps}} = \sum_t \left\| \hat{\mathbf{x}}_c^{t-1} + \hat{\mathbf{x}}_c^{t+1} - 2\hat{\mathbf{x}}_c^t \right\|^2 , \tag{5}$$

and $\mathcal{L}_{\text{rs}}$ is a rotation smoothness loss

$$\mathcal{L}_{\text{rs}} = \sum_t \left\| \hat{\mathbf{q}}_c^{t-1} + \hat{\mathbf{q}}_c^{t+1} - 2\hat{\mathbf{q}}_c^t \right\|^2 . \tag{6}$$

$\mathcal{N}_c$ the a set of indices of particles near the initial contact point location $\tilde{\mathbf{x}}_c^1$ and $\hat{\text{R}}_c^t$ is a ratation matrix computed from $\hat{\mathbf{q}}_c^t$. In practice, we set $\lambda_{\text{ps}}$ to 0.1 and $\lambda_{\text{rs}}$ to $10^{-4}$.

M-PhyGs incorporates the estimated contact-point motion into the MPM simulation by adding boundary conditions. As directly imposing boundary conditions on particle positions is non-trivial, we instead impose boundary conditions on particle velocities $\mathbf{v}_{i,\text{bc}}^t$

$$\mathbf{v}_{i,\text{bc}}^t = \frac{\mathbf{x}_{i,\text{c}}^{t+1} - \mathbf{x}_{i,\text{c}}^t}{n_s T} + \frac{\mathbf{x}_{i,\text{c}}^t - \mathbf{x}_i^t}{\kappa T} , \tag{7}$$

where $\mathbf{x}_{i,\text{c}}^t$ is the position of the particle computed from the recovered contact point motion under the rigid-motion assumption, $T$ is the simulation substep size, $n_s$ is the number of substeps per frame (*i.e.*, $n_s T$ is the time interval between neighboring frames), and $\kappa$ is a hyperparameter. The first term is the particle velocity computed from the estimated contact point motion, and the second term encourages the particle location to become close to $\mathbf{x}_{i,\text{c}}^t$. For particles near the estimated contact point, at each simulation substep, we overwrite the particle velocity $\mathbf{v}_i^t$ with $\mathbf{v}_{i,\text{bc}}^t$ before the particle-to-grid transfer in the MLS-MPM simulation [S2].

## C. Implementation Details

**Initial Particle Registration** For video sequences capturing the object dynamics, we align the physical particles and the 3D Gaussians to the first frames of the sequences based on sparse correspondences and a rendering loss. We first manually annotate 2D-2D correspondences between the first frames and frames corresponding to the rest shape, and then recover 3D-3D correspondences by triangulation. From these 3D-3D correspondences, we estimate a rigid motion that roughly aligns the physical particles and the 3D Gaussians with the first frames. We further refine per-particle 3D positions by minimizing a rendering loss that is a weighted sum of the RGB loss $\mathcal{L}_{\text{rgb}}$ and the DINO feature loss $\mathcal{L}_{\text{DINO}}$. During the refinement, we impose a local rigidity loss [S6] with a large weight to preserve the overall structure of the particle distribution.

**Temporal Mini Batching** During the material parameter estimation, the temporal mini-batch size is gradually increased. We start with a mini-batch size of 5 frames and then increase it to 10, 15, and 25 frames. The number of mini-batches is set to 3 for batch sizes up to 15 and reduced to 2 when the batch size is 25. During this temporal mini-batch training, the 3D loss $\mathcal{L}_{3\text{D}}$ is minimized. At the end, the material parameters are refined using all 50 frames without temporal mini-batching. During the refinement, the 3D loss and then the 2D loss $\mathcal{L}_{2\text{D}}$ are minimized. For each stage, the number of iterations is 150.

**MLS-MPM Simulation** We implemented our differentiable MLS-MPM simulator based on the implementation originally implemented by Xie *et al*. [S7] and later modified by Zhang *et al*. [S9] with adaptations for integration with M-PhyGs. First, as described in Sec. B, the motion of the contact point is incorporated into the MPM simulation. The efficiency of the MPM simulator is also improved by constructing voxel grids for the particle-to-grid and grid-to-particle transfers only within the bounding box of the simulation particles. For further details of the MLS-MPM simulation, please refer to the original paper [S2] and the SIGGRAPH course notes [S3].

**Experimental Settings** We set the grid spacing for the particle-to-grid and grid-to-particle transfers in the MLS-MPM simulation to 11 mm. The spacing between uniformly distributed simulation particles is 2 mm, and the number of simulation particles ranges from 37k to 180k per object. Most experiments were run on NVIDIA A100 GPUs with 80 GB of memory. For objects with 135k particles or more, we used an NVIDIA H200 GPU with 141 GB of memory. The total training time per object ranges from 80 to 100 hours.

Table 1. Quantitative accuracy comparison of cross-sequence dynamics prediction using estimated material. M-PhyGs achieves state-of-the-art accuracy even in this cross-sequence dynamics prediction.

| | PSNR (↑) | IoU (↑) | CD (↓) |
|---|---|---|---|
| PhysDreamer [S9] | 15.66 | 25.24% | 16.7 px |
| OmniPhysGS [S5] | 15.39 | 15.18% | 129.3 px |
| Pixie [S4] | 16.45 | 31.48% | 58.55 px |
| Spring-Gaus [S10] | 15.54 | 9.03% | 169.8 px |
| gs-dynamics [S8] | 16.31 | 29.69% | 65.6 px |
| GIC [S1] | 15.78 | 19.04% | 138.6 px |
| **M-PhyGs (Ours)** | **18.17** | **67.87%** | **4.4 px** |

Inference (dynamics prediction) takes on the order of a few seconds per frame.

## D. Modifications to Existing Methods

As most existing methods do not incorporate human-object interaction into their simulators, we adapt them for experimental comparison.

For PhysDreamer [S9], OmniPhysGS [S5], and Pixie [S4], similar to Sec. B, we incorporate the contact point motion into their MPM simulators. Note that this modification does not affect their training procedures as they do not require real videos as inputs. For PhysDreamer [S9] and OmniPhysGS [S5], we use an image from the dense view capture as the input image. For Pixie [S4], we use the dense view capture to recover the input voxel representation.

For Spring-Gaus [S10], we incorporate the contact point motion into its simulator, by replacing the positions of simulation particles near the contact point with those computed from the contact point motion. This replacement is performed just before the velocity update of each simulation substep. As the registration method of Spring-Gaus struggles with the complex geometry of flowers in the Phlowers dataset, we instead use our recovered Gaussians and registration results as inputs to its material parameter estimation stage.

For gs-dynamics [S8], it is difficult to recover 3D Gaussians from initial frames of the videos of Phlowers dataset, as they suffer from occlusions by a person. We instead recover 3D Gaussians for gs-dynamics from the dense view capture and align them with the first frame, similar to M-PhyGs. For each flower, we use a single 50-frame sequence for training, as in the training of M-PhyGs.

For GIC [S1], it is difficult to apply the original implementation to Phlowers dataset as it does not incorporate the contact point motion and requires a large amount of GPU memory for training for large objects (large flowers). We instead implement GIC on top of our M-PhyGs framework. We implement the Chamfer distance loss and the mask L1

loss of GIC and optimize a single set of material parameters, rather than our multi-material representation.

# E. Experimental Results

## E.1. Additional Qualitative Results

**Comparison with Existing Methods** Figures 1 to 3 and the supplementary video show additional qualitative results of dynamic prediction for unseen (in-sequence) interactions. The results show that M-PhyGs successfully predicts the complex motion of each flower which aligns with the actual held out observations.

**Ablation Studies** In the supplementary video, we show qualitative results of ablation studies. The results show that the every proposed components contributes to accurate dynamics prediction for unseen interactions.

## E.2. Cross-Sequence Dynamics Prediction

We conduct cross-sequence dynamics prediction, *i.e.*, estimate physical material parameters from one sequence of an object and use it to predict dynamics in another sequence of the same object. Table 1, Figs. 4 to 6, and the supplementary video show quantitative and qualitative results. The results show that M-PhyGs successfully predicts dynamics even in this cross-sequence setting, which further shows the accuracy of their physical material parameter estimates.

# References


[S1] Junhao Cai, Yuji Yang, Weihao Yuan, Yisheng He, Zilong Dong, Liefeng Bo, Hui Cheng, and Qifeng Chen. GIC: Gaussian-Informed Continuum for Physical Property Identification and Simulation. In *NeurIPS*, 2024. 2

[S2] 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 TOG*, 37 (4):150, 2018. 1, 2

[S3] Chenfanfu Jiang, Craig A. Schroeder, Joseph Teran, Alexey Stomakhin, and Andrew Selle. The material point method for simulating continuum materials. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference, SIGGRAPH '16, Anaheim, CA, USA, July 24-28, 2016, Courses*, pages 24:1–24:52. ACM, 2016. 2

[S4] Long Le, Ryan Lucas, Chen Wang, Chuhao Chen, Dinesh Jayaraman, Eric Eaton, and Lingjie Liu. Pixie: Fast and generalizable supervised learning of 3d physics from pixels. *arXiv preprint arXiv:2508.17437*, 2025. 2

[S5] Yuchen Lin, Chenguo Lin, Jianjin Xu, and Yadong MU. OmniPhysGS: 3D Constitutive Gaussians for General Physics-Based Dynamics Generation. In *ICLR*, 2025. 2

[S6] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis. In *3DV*, 2024. 2

[S7] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. PhysGaussian: Physics-Integrated 3D Gaussians for Generative Dynamics. In *CVPR*, pages 4389–4398. IEEE, 2024. 2

[S8] Mingtong Zhang, Kaifeng Zhang, and Yunzhu Li. Dynamic 3D Gaussian Tracking for Graph-Based Neural Dynamics Modeling. In *8th Annual Conference on Robot Learning*, 2024. 2

[S9] Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y. Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T. Freeman. PhysDreamer: Physics-Based Interaction with 3D Objects via Video Generation. In *ECCV*, pages 388–406. Springer, 2024. 2

[S10] Licheng Zhong, Hong-Xing Yu, Jiajun Wu, and Yunzhu Li. Reconstruction and Simulation of Elastic Objects with Spring-Mass 3D Gaussians. In *ECCV*, pages 407–423. Springer, 2024. 2

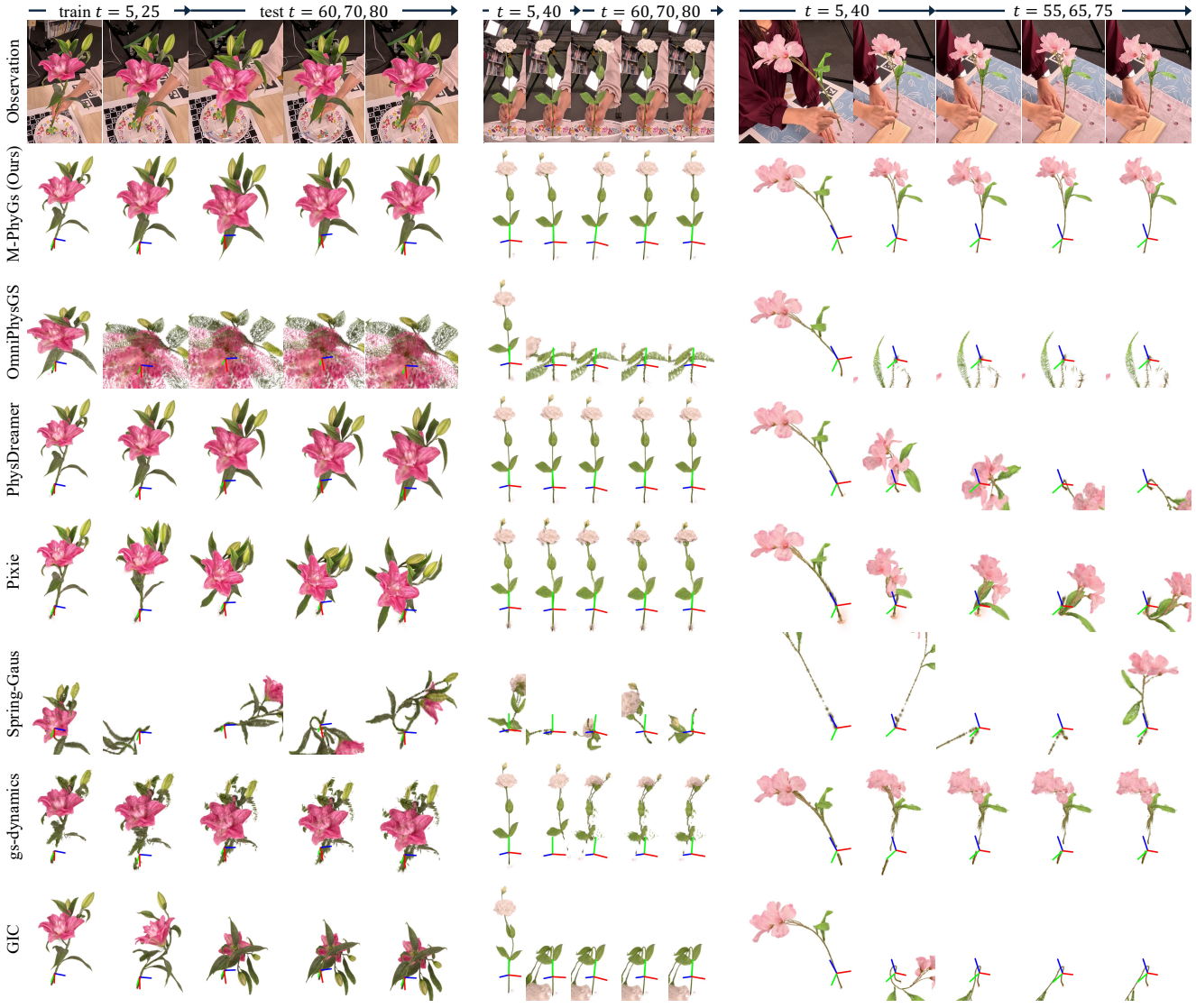


Figure 1. Unseen dynamics predicted with estimated physical material parameters. M-PhyGs successfully predicts the complex motion of each flower which aligns with the actual held out observations.

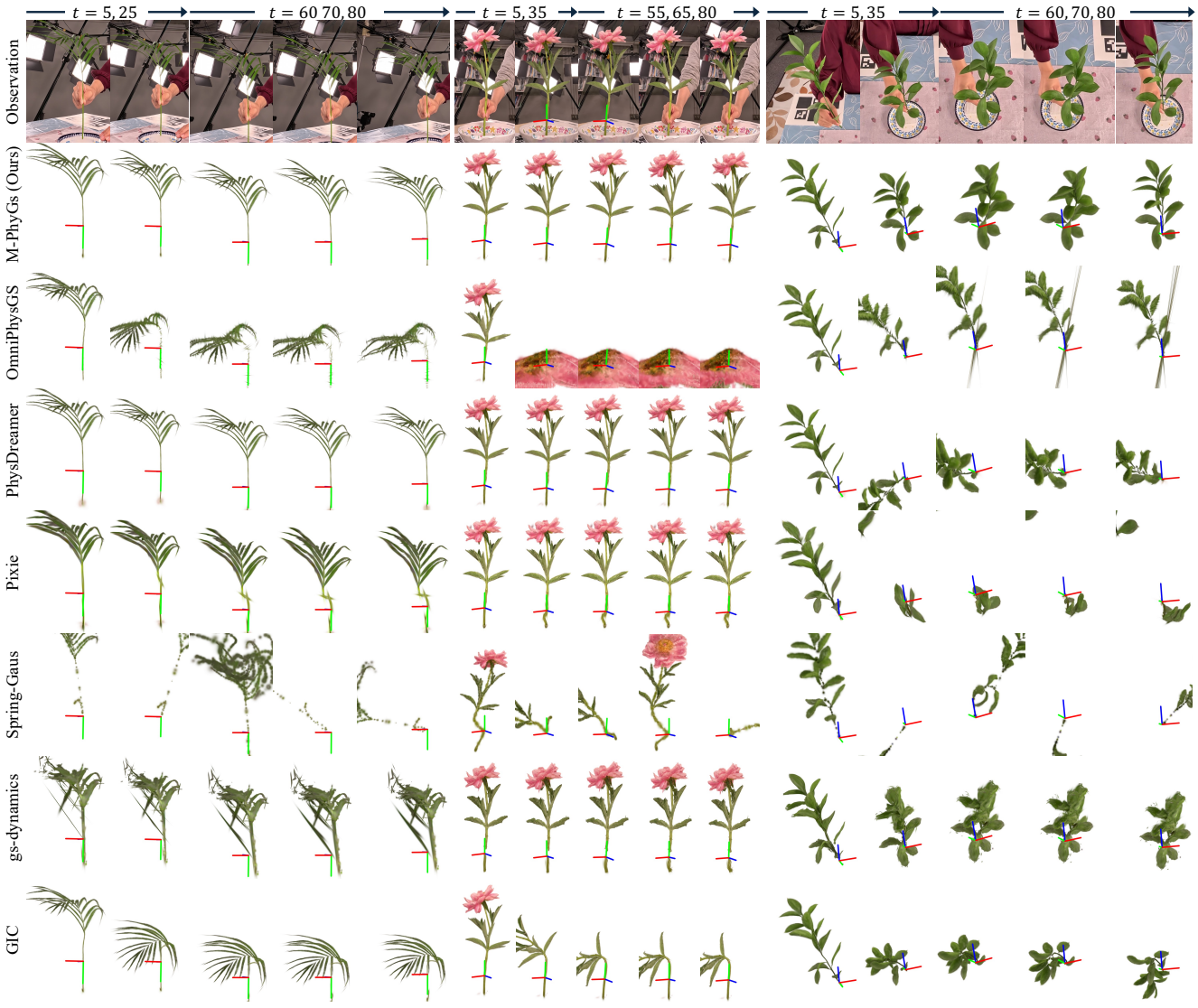


Figure 2. Unseen dynamics predicted with estimated physical material parameters. M-PhyGs successfully predicts the complex motion of each flower which aligns with the actual held out observations.


Figure 3. Unseen dynamics predicted with estimated physical material parameters. M-PhyGs successfully predicts the complex motion of each flower which aligns with the actual held out observations.


Figure 4. Qualitative results of cross-sequence dynamics prediction using estimated material. M-PhyGs successfully predicts the complex motion of each flower even in this cross-sequence setting.


Figure 5. Qualitative results of cross-sequence dynamics prediction using estimated material. M-PhyGs successfully predicts the complex motion of each flower even in this cross-sequence setting.

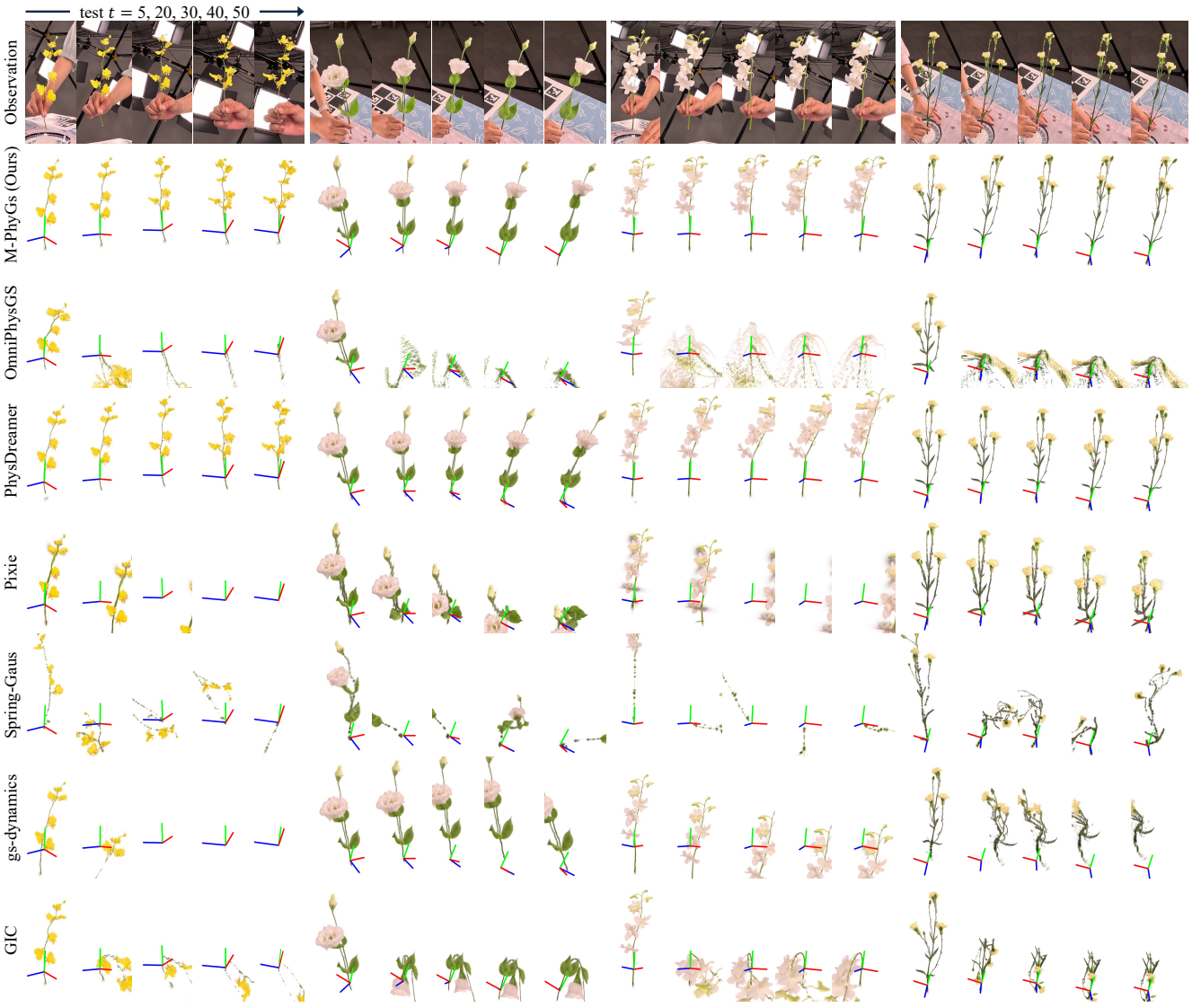


Figure 6. Qualitative results of cross-sequence dynamics prediction using estimated material. M-PhyGs successfully predicts the complex motion of each flower even in this cross-sequence setting.