PhysTwin: Physics-Informed Reconstruction and Simulation of Deformable Objects from Videos

1st Hanxiao Jiang 2st Hao-Yu Hsu 3st Kaifeng Zhang 4st Hsin-Ni Yu 5st Shenlong Wang 6st Yunzhu Li

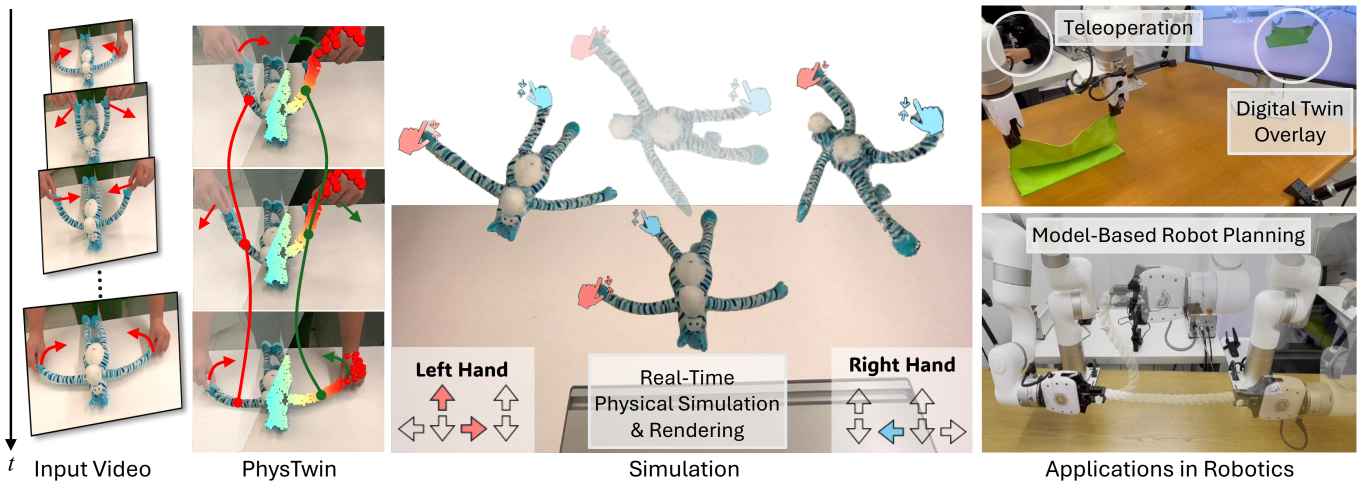

Fig. 1: **PhysTwin** takes sparse videos (three camera views) of deformable objects under interaction as input and reconstructs a simulatable digital twin with complete geometry, high-fidelity appearance, and accurate physical parameters. This enables multiple applications, such as real-time interactive simulation using keyboards and robotic teleoperation devices, as well as model-based robot planning.

***Abstract*—Creating a physical digital twin of a real-world object has immense potential in robotics, content creation, and XR. In this paper, we present PhysTwin, a novel framework that uses sparse videos of dynamic objects under interaction to produce a photo- and physically realistic, real-time interactive virtual replica. Our approach centers on two key components: (1) a physics-informed representation that combines spring-mass models for realistic physical simulation, generative shape models for geometry, and Gaussian splats for rendering; and (2) a novel multi-stage, optimization-based inverse modeling framework that reconstructs complete geometry, infers dense physical properties, and replicates realistic appearance from videos. Our method integrates an inverse physics framework with visual perception cues, enabling high-fidelity reconstruction even from partial, occluded, and limited viewpoints. PhysTwin supports modeling various deformable objects, including ropes, stuffed animals, cloth, and delivery packages. Experiments show that PhysTwin outperforms competing methods in reconstruction, rendering, future prediction, and simulation under novel interactions. We further demonstrate its applications in interactive real-time simulation and model-based robotic motion planning. Project Page: https://jianghanxiao.github.io/phystwin-web/**

I. Introduction

In this work, we aim to build an interactive PhysTwin from sparse-viewpoint RGB-D video sequences Fig. 2, capturing object geometry, non-rigid dynamic physics, and appearance for realistic physical simulation and rendering. We model deformable object dynamics with a spring-mass-based representation, enabling efficient physical simulation and handling a wide range of common objects, such as ropes, stuffed animals, cloth, and delivery packages. To address the challenges posed by sparse observations, we leverage shape priors and motion estimation from advanced 3D generative models [6] and vision foundation models [4, 2, 5] to estimate the topology, geometry, and physical parameters of our physical representation. Since some physical parameters (such as topology-related properties) are non-differentiable and optimizing them efficiently is non-trivial, we design a hierarchical sparse-to-dense optimization strategy. This strategy integrates zero-order optimization [1] for non-differentiable topology and sparse physical parameters (e.g., collision parameters and homogeneous spring stiffness), while employing first-order gradient-based optimization to refine dense spring stiffness and further optimize collision parameters. For appearance modeling, we adopt a Gaussian blending strategy, initializing static Gaussians from sparse observations in the first frame using shape priors and deforming them with a linear blending algorithm to generate realistic dynamic appearances.

Our inverse modeling framework effectively constructs interactive PhysTwin from videos of objects under interaction. We create a real-world deformable object interaction dataset and evaluate our method on three key tasks: reconstruction and resimulation, future prediction, and generalization to unseen interactions. Both quantitative and qualitative results demonstrate that our reconstructed PhysTwin aligns accurately with real-world observations, achieves precise future predictions, and generates realistic simulations under diverse unseen interactions. Furthermore, the high computational efficiency of our physics simulator enables real-time dynamics and rendering of our constructed PhysTwin, facilitating multiple applications, including real-time interactive simulation and model-based robotic motion planning.

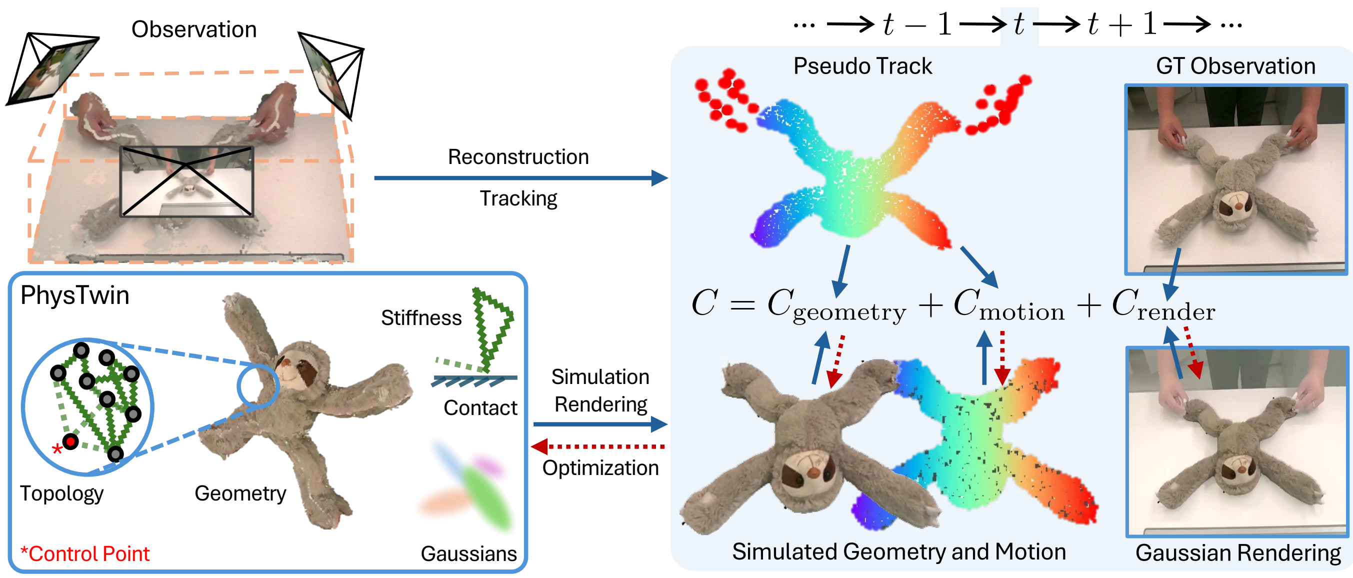

Fig. 2: **Overview of Our PhysTwin Framework.** We present an overview of our PhysTwin framework, where the core representation includes geometry, topology, physical parameters (associated with springs and contacts), and Gaussian kernels. To optimize PhysTwin, we minimize the rendering loss and the discrepancy between simulated and observed geometry/motion. The rendering loss optimizes the Gaussian kernels, while the geometry and motion losses refine the overall geometry, topology, and physical parameters in PhysTwin.

II. Experiments

In this section, we evaluate the performance of our PhysTwin framework across three distinct tasks involving different types of objects. Our primary objective is to address the following three questions: (1) How accurately does our framework reconstruct and resimulate deformable objects and predict their future states? (2) How well does the constructed PhysTwin generalize to unseen interactions? (3) What is the utility of PhysTwin in downstream tasks?

A. Experiment Settings

Tasks. To assess the effectiveness of our PhysTwin framework and the quality of our constructed PhysTwin, we formulate three tasks: (1) Reconstruction & Resimulation; (2) Future Prediction; and (3) Generalization to Unseen Actions.

For the Reconstruction & Resimulation task, the objective is to construct PhysTwin such that it can accurately reconstruct and resimulate the motion of deformable objects given the actions represented by the control point positions.

For the Future Prediction task, we aim to assess whether PhysTwin can perform well on unseen future frames during its construction.

For the Generalization to Unseen Interactions task, the goal is to assess whether PhysTwin can adapt to different interactions. To evaluate this, we construct a generalization dataset consisting of interaction pairs performed on the same object but with varying motions, including differences in hand configuration and interaction type.

B. Results

To assess the performance of our framework and the quality of our constructed PhysTwin, we compare with two augmented baselines across three task settings. Our quantitative analysis reveals that the PhysTwin framework consistently outperforms the baselines across various tasks.

Reconstruction & Resimulation. The quantitative results in **??** Reconstruction & Resimulation column demonstrate the superior performance of our PhysTwin method over baselines. Our approach significantly improves all evaluated metrics, including Chamfer Distance, tracking error, and 2D IoU, confirming that our reconstruction and resimulation align more closely with the original observations. This highlights the effectiveness of our model in learning a more accurate dynamics model under sparse observations. Additionally, rendering metrics show that our method produces more realistic 2D images, benefiting from the Gaussian blending strategy and enhanced dynamic modeling. Fig. 3 further provides qualitative visualizations across different objects, illustrating precise alignment with original observations. Notably, our physics-based representation inherently improves point tracking. After physics-constrained optimization, our tracking surpasses the original CoTracker3 [2] predictions used for training, achieving better alignment after global optimization (See supplement for more details).

Future Prediction. **??**, in the Future Prediction column, demonstrates that our method achieves superior performance in predicting unseen frames, excelling in both dynamics alignment and rendering quality. Fig. 3 further provides qualitative results, illustrating the accuracy of our predictions on unseen frames.

Generalization to Unseen Interactions. We also evaluate the generalization performance to unseen interactions. Our dataset includes transfers from one interaction (e.g., single lift) to significantly different interactions (e.g., double stretch). We directly use our constructed PhysTwin and leverage our registration pipeline to align it with the first frame of the target case. Fig. 4 shows that our method closely matches the ground truth observations in terms of dynamics. Quantitative results further demonstrate the robustness of our method across different actions. In contrast, the neural dynamics model struggles to adapt to environmental changes and diverse

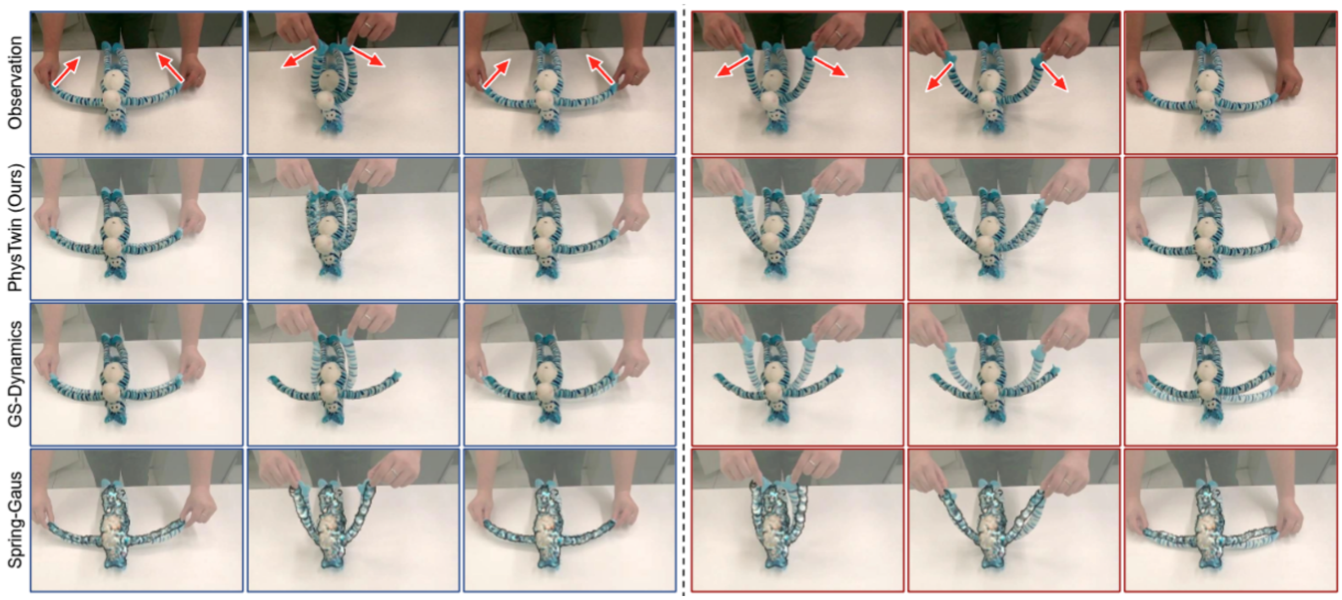

Fig. 3: **Qualitative Results on Reconstruction & Resimulation and Future Prediction.** We visualize the rendering results of different methods on two tasks. For the reconstruction & resimulation task, our method achieves a better match with the observations. For the future prediction task, our method accurately predicts the future state of the objects. In contrast, the baselines fail in most cases: GS-Dynamics [7] tends to remain static, while Spring-Gauss [8] frequently causes the physical model to crash.

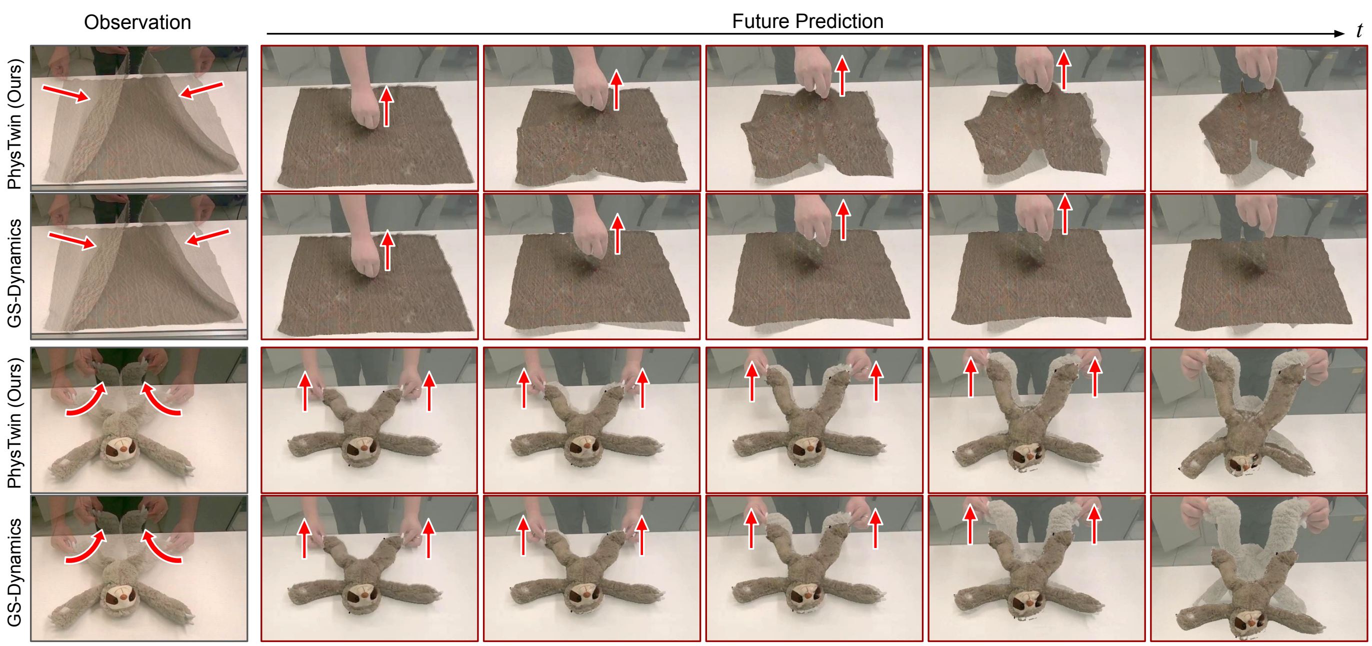

Fig. 4: **Qualitative Results on Generalization to Unseen Interactions.** We visualize the simulation of a deformable object under unseen interactions using our method and GS-Dynamics [7]. The leftmost image shows the interaction used to train the dynamics models, while the images on the right demonstrate their generalization to unseen interactions. Our PhysTwin significantly outperforms prior work.

interactions as effectively as our approach. Moreover, in unseen interaction scenarios, our method achieves performance comparable to that on the future prediction task, highlighting the robustness and generalization capability of our constructed PhysTwin.

C. Application

The efficient forward simulation capabilities of our Spring-Mass simulator, implemented using Warp [3], enable a variety of downstream applications. **??** showcases key applications enabled by our PhysTwin: (1) Interactive Simulation: Users can interact with objects in real time using keyboard controls, either with one or both hands. The system also supports real-time simulation of an object's future state during human teleoperation with robotic arms. This feature serves as a valuable tool for predicting object dynamics during manipulation. (2) Model-Based Robotic Planning: Owing to the high fidelity of our constructed PhysTwin, it can be used as a dynamic model in planning pipelines. By integrating it with model-based planning techniques, we can generate effective motion plans for robots to complete a variety of tasks.

REFERENCES

[1] Nikolaus Hansen. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*, pages 75–102, 2006.

[2] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

[3] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. https://github.com/nvidia/warp, 2022. NVIDIA GPU Technology Conference (GTC).

[4] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[6] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[7] Mingtong Zhang, Kaifeng Zhang, and Yunzhu Li. Dynamic 3d gaussian tracking for graph-based neural dynamics modeling. *arXiv preprint arXiv:2410.18912*, 2024.

[8] Licheng Zhong, Hong-Xing Yu, Jiajun Wu, and Yunzhu Li. Reconstruction and simulation of elastic objects with spring-mass 3d gaussians. In *European Conference on Computer Vision*, pages 407–423. Springer, 2024.