PhysiX: A Foundation Model for Physics Simulations

Anonymous Author(s)
Affiliation
Address
email

Abstract

Foundation models have achieved remarkable success across image and language domains. By scaling up the parameter count and data, these models acquire generalizable world knowledge and often surpass task-specific approaches. However, such progress has yet to extend to the domain of physics simulation. A primary bottleneck is data scarcity: while millions of images, videos, and textual resources are available on the internet, the largest physics simulation datasets contain only tens of thousands of samples. This data limitation hinders the use of large models, as overfitting becomes a major concern. As a result, physics applications typically rely on small models, which struggle with long-range prediction due to limited context understanding. Additionally, unlike other modalities that often exhibit fixed granularity, physics datasets vary drastically in scale, amplifying the challenges of scaling up multitask training. We introduce **PhysiX**, one of the first large-scale foundation models for physics simulation. PhysiX is a 4.5B parameter autoregressive generative model. It uses a discrete tokenizer to encode physical processes at different scales into a sequence of discrete tokens, and employs an autoregressive next-token prediction objective to model such processes in the token space. To mitigate the rounding error in the discretization process, PhysiX incorporates a specialized refinement module. Through extensive experiments, we show that PhysiX effectively addresses the data bottleneck, outperforming task-specific baselines under comparable settings as well as the previous absolute state-of-the-art approaches on The Well benchmark. Our results indicate that knowledge learned from natural videos can be successfully transferred to physics simulation, and that joint training across diverse simulation tasks enables synergistic learning.

1 Introduction

Simulating physical systems using partial differential equations (PDEs) is a fundamental aspect of science and engineering, traditionally tackled by computationally expensive numerical solvers [11, 5, 6, 31, 10, 23]. To address this high cost, machine learning (ML)-based surrogates have emerged, offering faster inference times by approximating simulation outputs [47, 42, 9, 43, 13]. However, most existing ML surrogates are task-specific, struggling to adapt to changes in simulation parameters or to capture shared patterns across different physical domains.

In this work, we introduce **PhysiX**, one of the first large-scale autoregressive foundation models for physical simulations. PhysiX utilizes a universal discrete tokenizer to represent heterogeneous spatiotemporal data in a unified token space, allowing for joint training on a diverse corpus of physics datasets. PhysiX consists of a 4.5B parameter autoregressive transformer, initialized with a pretrained video generation checkpoint to leverage strong spatiotemporal priors, and a refinement module to enhance output fidelity. Figure 1 shows the superior performance of PhysiX compared to task-specific baselines on The Well benchmark [33], demonstrating accurate long-range prediction and better generalization across diverse tasks.

Submitted to the Machine Learning and the Physical Sciences workshop (NeurIPS 2025).

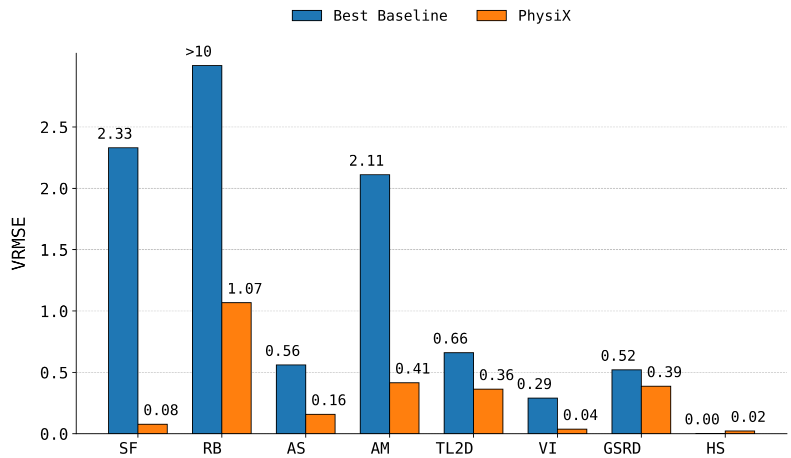

Figure 1: PhysiX versus the baselines in 8 tasks of the Well benchmark. We report VRMSE (lower is better) averaged across different physical properties and lead time between 9-26 frames for each task.

2 Method

PhysiX consists of three components: a discrete tokenizer, an autoregressive (AR) generation model, and a refinement module. Given k input frames $x_1, \ldots, x_k$, we convert them into $\hat{k}$ latent frames $z_1, \ldots, z_{\hat{k}}$, where each $z_i = [z_i^1, \ldots, z_i^L]$ contains L discrete tokens. These sequences are concatenated into a 1D input for the AR model, which predicts tokens $z_{\hat{k}+1}, \ldots, z_{\hat{k}+\hat{T}}$ corresponding to T pixel frames. These tokens are decoded back to pixel space to obtain the coarse AR prediction $\hat{x}_{k+1}, \hat{x}_{k+2}, \hat{x}_{k+T}$. We employ a refinement module to further improve the prediction by correcting rounding errors from the discretization process. Figure 2 illustrates the architecture of PhysiX.

2.1 Universal Tokenizer

We adopt the Cosmos tokenizer [2], an encoder-decoder model that maps video frames into discrete tokens while preserving spatiotemporal structure. The encoder applies causal convolution and attention to generate latent representations, which are quantized using Finite-Scalar Quantization (FSQ) [30]. The decoder then reconstructs frames from these quantized tokens.

To enable cross-task generalization, we train a universal tokenizer across all simulation datasets. We propose two changes to address dataset heterogeneity in channel dimensionality, spatial resolution, and physical semantics. First, we allow the encoder to accept the union of all channels observed across datasets, replacing missing channels with per-channel learnable 2D tensors. Second, while the encoder is shared to enforce a common embedding space, we employ dataset-specific decoders to improve reconstruction quality and capture output distributions unique to each dataset. To ensure balanced representation across datasets during training, we replicate smaller datasets so that each dataset contributes an equal number of sequences to the training process. We initialize the universal tokenizer from a pre-trained Cosmos checkpoint, which we found significantly accelerates convergence and improves reconstruction quality compared to training from scratch. This pre-trained initialization facilitates better transfer to the physics domain by leveraging learned priors from natural video data.

2.2 Autoregressive Generative Models

Given the tokenizer, we train a large-scale autoregressive model to simulate physics in the discrete latent space. PhysiX follows the autoregressive architecture introduced in Cosmos [2]. Given a sequence of discrete tokens from the past k input frames, the transformer is trained with a next-token prediction objective to generate tokens for the subsequent T frames. Formally, the objective is:

$$\mathcal{L}_{\mathcal{AR}} = -\sum_{i=1}^{\hat{M}} \sum_{j=1}^{L} \mathbb{E}_z \left[\log p(z_i^j | \{z_m^n | m < i \text{ or } m = i, n < j\}) \right], \tag{1}$$

where $L = \frac{HW}{8^2}$ is the length of each latent frame z_i, and $\hat{M} = \frac{k+T}{4}$ is the number of latent frames.

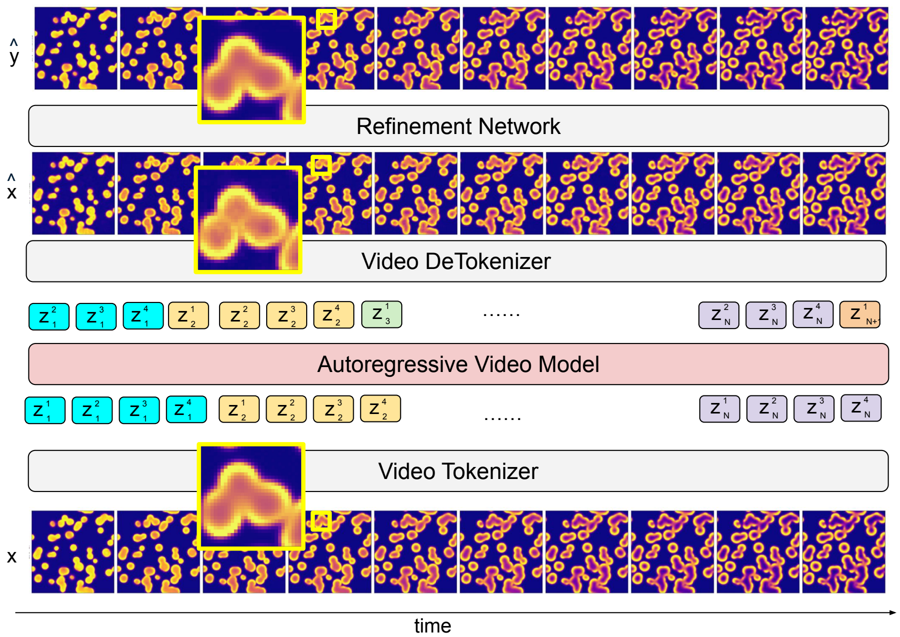

Figure 2: Given input frames $x_1, \ldots, x_N$, the tokenizer discretizes each frame into a sequence of discrete tokens, where the jth token of frame i is denoted as $\{z_i^j\}$. The autoregressive model generates predictions in this token space, before being de-tokenized into pixel-level predictions $\hat{x}$. A refinement module helps mitigate artifacts caused by the discretization error, such as blocky, pixelated outputs (visualized in yellow boxes), and produces the final sharper and more detailed output $\hat{y}$.

The autoregressive model incorporates 3D rotary position embeddings (RoPE) to capture relative spatiotemporal relationships across the token sequence. A key distinction from prior work is our support for variable spatial resolutions during training. Since simulation datasets differ in shape, we adjust the positional encodings dynamically, where we truncate the 3D RoPE frequencies along the height and width dimensions to match the size of the current input. This approach, implemented with minimal modification to the original RoPE module, allows seamless handling of mixed-resolution data without sacrificing performance. We found this simple strategy worked equally well as more advanced interpolation techniques [36, 55]. We initialize the autoregressive model from the 4.5B parameter Cosmos checkpoint (NVIDIA/COSMOS-1.0-AUTOREGRESSIVE-4B), enabling it to inherit strong spatiotemporal priors learned from large-scale natural video datasets. Similar to tokenizer training, we oversample smaller datasets to match the size of the largest one.

2.3 Refinement Module

The refinement module is a convolutional neural network designed to mitigate artifacts introduced by the discretization process of AR models. We show one such example in Figure 2: the AR output (middle) $\hat{x}$ displays a quantization-like noise pattern in the center, while the ground truth (bottom) x is noise-free. The refinement output (top) $\hat{y}$ successfully removes this noise. Such artifacts arise from the inherent limitations of discrete tokenization, originally developed for natural videos. While negligible in character or scenery generation, these artifacts can severely degrade performance in physical simulation tasks, where precision is critical.

We train the refinement module as a post-processing step after AR model training. Specifically, we autoregressively generate predictions on the training split and pair them with ground truth frames as refinement targets. Before feeding AR outputs into the module, we decode them into pixel space, allowing the model to directly improve visual fidelity. Our architecture follows the ConvNeXt-U-Net baseline from the Well benchmark, trained with MSE loss. The key distinction lies in the learning objective: instead of predicting new frames, the refinement model learns to enhance AR outputs. As with the decoder in the universal tokenizer, we train separate refinement modules for each dataset. Further details are provided in the appendix.

3 Experiments

We train and evaluate PhysiX across eight simulation tasks from the Well benchmark [33], as shown in Tables 1 and 2. Following the benchmark protocol, we report the Variance-Weighted Root Mean Squared Error (VRMSE), averaged over all physical channels for each dataset. For datasets such as `helmholtz_staircase` and `acoustic_scattering (maze)`, we exclude channels that remain constant across time steps from the evaluation. We compare PhysiX against four baselines provided by the Well benchmark: Fourier Neural Operator (FNO), Tucker-Factorized FNO (TFNO), U-Net, and U-Net with ConvNeXt blocks (C-U-Net), considering both next-frame and long-horizon rollout settings. In addition, we conduct extensive ablation studies to assess the impact of various architectural and training design choices in PhysiX. We also study the ability of PhysiX to adapt to unseen simulations, the impact of using video-pretrained models, scaling results, and qualitative results in Appendix H.

3.1 Next-frame Prediction

In the next-frame prediction benchmark, PhysiX outperforms the baselines on 5 out of 8 datasets, demonstrating strong generalization across diverse physical systems. In addition, PhysiX achieves the best average rank across the 8 tasks, with a score of 1.62 compared to 2.38 for the best-performing baseline. Importantly, PhysiX achieves this performance using a single model checkpoint shared across all tasks, whereas the baseline results are obtained from separate models trained specifically for each dataset. This highlights the ability of PhysiX to act as a general-purpose simulator. The performance gain is especially significant on the `shear_flow` and `rayleigh_benard` datasets, where PhysiX reduces the VRMSE by 91% and 78% respectively relative to the best baseline.

Table 1: **Next-frame prediction performance across 8 datasets on the Well benchmark**. We report VRMSE (lower is better) averaged across different fields for each dataset.

Dataset	Baseline				Ours
	FNO	TFNO	U-Net	C-U-Net	PhysiX
`shear_flow`	1.189	1.472	3.447	0.8080	**0.0700**
`rayleigh_benard`	0.8395	0.6566	1.4860	0.6699	**0.1470**
`acoustic_scattering (maze)`	0.5062	0.5057	0.0351	**0.0153**	0.0960
`active_matter`	0.3691	0.3598	0.2489	0.1034	**0.0904**
`turbulent_radiative_layer_2D`	0.5001	0.5016	0.2418	**0.1956**	0.2098
`viscoelastic_instability`	0.7212	0.7102	0.4185	0.2499	**0.2370**
`gray_scott_reaction_diffusion`	0.1365	0.3633	0.2252	0.1761	**0.0210**
`helmholtz_staircase`	**0.00046**	0.00346	0.01931	0.02758	0.0180
Average Rank (↓)	3.62	3.75	3.62	2.38	**1.62**

3.2 Long-horizon Prediction

While PhysiX already performs competitively in next-frame prediction, its true strength lies in long-horizon simulation. As shown in Table 2, PhysiX achieves state-of-the-art performance on $18/21$ evaluation points across different forecasting windows. The improvements are not only consistent but also significant in various tasks. For example, on `shear_flow`, PhysiX reduces VRMSE by over 97% at the 6:12 horizon compared to the best-performing baseline (from 2.33 to 0.077). On `rayleigh_benard`, PhysiX achieves more than 90% lower error across all rollout windows. Similar results are observed in `active_matter`, where PhysiX consistently achieves better performance at every forecast horizon, underscoring its robustness and adaptability across domains.

Table 2: **Long-horizon prediction performance across 8 datasets on the Well benchmark.** We report VRMSE (lower is better) averaged across different fields for each dataset. We report averaged results over different ranges of lead time: 2-8, 9-26 and 27-56 frames.

Dataset	Δt =2:8		Δt =9:26		Δt =27:56	
	Baseline	PhysiX	Baseline	PhysiX	Baseline	PhysiX
`shear_flow`	2.330	**0.077**	>10	**0.153**	>10	**0.236**
`rayleigh_benard`	>10	**1.067**	>10	**0.741**	>10	**0.847**
`acoustic_scattering (maze)`	0.560	**0.158**	**0.920**	1.246	**1.341**	2.189
`active_matter`	2.110	**0.415**	2.710	**0.974**	1.635	**1.320**
`turbulent_radiative_layer_2D`	0.660	**0.363**	1.040	**0.693**	1.331	**0.953**
`gray_scott_reaction_diffusion`	0.290	**0.037**	7.620	**1.984**	12.714	**12.643**
`viscoelastic_instability`	0.520	**0.387**	—	—	—	—
`helmholtz_staircase`	**0.002**	0.022	**0.003**	0.071	—	—

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[3] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021.

[4] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[5] Marsha J Berger and Randall J LeVeque. Implicit adaptive mesh refinement for dispersive tsunami propagation. *SIAM Journal on Scientific Computing*, 46(4):B554–B578, 2024.

[6] Lorenz T Biegler, Omar Ghattas, Matthias Heinkenschloss, and Bart van Bloemen Waanders. Large-scale pde-constrained optimization: an introduction. In *Large-scale PDE-constrained optimization*, pages 3–13. Springer, 2003.

[7] Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[9] Kamal Choudhary, Brian DeCost, Chi Chen, Anubhav Jain, Francesca Tavazza, Ryan Cohn, Cheol Woo Park, Alok Choudhary, Ankit Agrawal, Simon JL Billinge, et al. Recent advances and applications of deep learning methods in materials science. *npj Computational Materials*, 8(1):59, 2022.

[10] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

[11] Veronika Eyring, Sandrine Bony, Gerald A Meehl, Catherine A Senior, Bjorn Stevens, Ronald J Stouffer, and Karl E Taylor. Overview of the coupled model intercomparison project phase 6 (cmip6) experimental design and organization. *Geoscientific Model Development*, 9(5):1937–1958, 2016.

[12] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022.

[13] Vignesh Gopakumar, Stanislas Pamela, Lorenzo Zanisi, Zongyi Li, Ander Gray, Daniel Brennand, Nitesh Bhatia, Gregory Stathopoulos, Matt Kusner, Marc Peter Deisenroth, et al. Plasma surrogate modelling using fourier neural operators. *Nuclear Fusion*, 64(5):056025, 2024.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[15] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

[16] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[17] Hee E Kim, Alejandro Cosa-Linan, Nandhini Santhanam, Mahboubeh Jannesari, Mate E Maros, and Thomas Ganslandt. Transfer learning for medical image classification: a literature review. *BMC medical imaging*, 22(1):69, 2022.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Georgios Kissas, Jacob H Seidman, Leonardo Ferreira Guilhoto, Victor M Preciado, George J Pappas, and Paris Perdikaris. Learning operators with coupled attention. *Journal of Machine Learning Research*, 23(215):1–63, 2022.

[20] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

[21] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[22] Nikola B. Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *Journal of Machine Learning Research*, 24:1–97, 2023. Article 89.

[23] Pablo Lemos, Liam Parker, ChangHoon Hahn, Shirley Ho, Michael Eickenberg, Jiamin Hou, Elena Massara, Chirag Modi, Azadeh Moradinezhad Dizgah, Bruno Regaldo-Saint Blancard, et al. Simbig: Field-level simulation-based inference of galaxy clustering. *arXiv preprint arXiv:2310.15256*, 2023.

[24] Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *arXiv preprint arXiv:2205.13671*, 2022.

[25] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.

[26] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*. ICLR, 2021.

[27] Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. *Advances in Neural Information Processing Systems*, 36:67398–67433, 2023.

[28] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

[29] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

[30] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. In *The Twelfth International Conference on Learning Representations*.

[31] Bijan Mohammadi and Olivier Pironneau. Shape optimization in fluid mechanics. *Annu. Rev. Fluid Mech.*, 36(1):255–279, 2004.

[32] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*, 2023.

[33] Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina Agocs, Miguel Beneitez, Marsha Berger, Blakesly Burkhart, Stuart Dalziel, Drummond Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

[34] OpenAI. Sora: A video generation model. `https://openai.com/sora`, 2024. Accessed: 2025-05-13.

[35] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[36] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*.

[37] Zhiwu Qing, Shiwei Zhang, Jiayu Wang, Xiang Wang, Yujie Wei, Yingya Zhang, Changxin Gao, and Nong Sang. Hierarchical spatio-temporal decoupling for text-to-video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6635–6645, 2024.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[39] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[40] Colorado J Reed, Ritwik Gupta, Shufan Li, Sarah Brockman, Christopher Funk, Brian Clipp, Kurt Keutzer, Salvatore Candido, Matt Uyttendaele, and Trevor Darrell. Scale-mae: A scale-aware masked autoencoder for multiscale geospatial representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4088–4099, 2023.

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[42] Kevin Ryczko, David A Strubbe, and Isaac Tamblyn. Deep learning and density-functional theory. *Physical Review A*, 100(2):022512, 2019.

[43] Ali Siahkoohi, Rudy Morel, Randall Balestriero, Erwan Allys, Grégory Sainton, Taichi Kawamura, and Maarten V de Hoop. Martian time-series unraveled: A multi-scale nested approach with factorial variational autoencoders. *arXiv preprint arXiv:2305.16189*, 2023.

[44] Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W. Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[45] Mingzhen Sun, Weining Wang, Gen Li, Jiawei Liu, Jiahui Sun, Wanquan Feng, Shanshan Lao, SiYu Zhou, Qian He, and Jing Liu. Ar-diffusion: Asynchronous video generation with auto-regressive diffusion. *arXiv preprint arXiv:2503.07418*, 2025.

[46] Seiji Takeda, Indra Priyadarsini, Akihiro Kishimoto, Hajime Shinohara, Lisa Hamada, Hirose Masataka, Junta Fuchiwaki, and Daiju Nakano. Multi-modal foundation model for material design. In *AI for Accelerated Materials Design-NeurIPS 2023 Workshop*, 2023.

[47] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature physics*, 14(5):447–450, 2018.

[48] Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[49] Junke Wang, Yi Jiang, Zehuan Yuan, Bingyue Peng, Zuxuan Wu, and Yu-Gang Jiang. Omnitokenizer: A joint image-video tokenizer for visual generation. *Advances in Neural Information Processing Systems*, 37:28281–28295, 2024.

[50] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[51] Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *International Journal of Computer Vision*, 133(5):3059–3078, 2025.

[52] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[53] Yinhao Zhu and Nicholas Zabaras. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 2018.

[54] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[55] Le Zhuo, Ruoyi Du, Han Xiao, Yangguang Li, Dongyang Liu, Rongjie Huang, Wenze Liu, Xiangyang Zhu, Fu-Yun Wang, Zhanyu Ma, et al. Lumina-next: Making lumina-t2x stronger and faster with next-dit. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

A Related Works

Physics Simulation Traditional simulation modeling typically relies on numerical methods, such as finite element methods, finite difference methods, and finite volume methods, to approximate solutions to differential equations governing physical laws. While effective, these approaches often require significant computational resources, especially for high-resolution simulations or long-term predictions, limiting their scalability and real-time applicability.

Advances in machine learning have offered promising alternatives to accelerate or supplement traditional PDE solvers [44, 16]. Physics-informed neural networks (PINNs) incorporate prior knowledge of governing equations into the loss function [39]. These methods require little observational data, as physical constraints guide the learning process. This provides the benefit of interpretable and improved physical plausibility, but makes PINNs an unsuitable choice when the underlying physical laws are unknown or only partially understood.

Concurrently, data-driven surrogate modeling methods have also seen success in this area, shifting from explicitly modeling physical laws towards implicitly learning system dynamics through observed data [28]. Early work utilized CNNs, particularly U-Net architectures [41, 53], to model spatiotemporal relationships between physical fields. More recently, neural operator frameworks have emerged, which aim to learn mappings between infinite-dimensional function spaces [22, 29]. These include Fourier Neural Operators (FNOs) [26], which leverage Fast Fourier Transforms for efficient global convolution, and various Transformer-based architectures [24, 19] that utilize attention mechanisms to capture long-range dependencies. To handle complex geometries where methods like FNOs may struggle, Graph Neural Network (GNN) based operators have also been developed, capable of operating directly on unstructured meshes [25, 7]. These operator learning frameworks enable generalization to different initial conditions, boundary conditions, and spatial resolutions without explicit retraining.

Despite these advancements, current neural network-based physics simulators face limitations. They often struggle with long-range predictions [27], and many models are typically trained and optimized for a specific physical system, a narrow range of parameters, or a particular set of governing equations. Current neural network approaches can generalize within a physical domain, but perform poorly across distinct physical domains without substantial retraining or architectural modifications.

Video Generation Video generation models have achieved considerable progress in recent years. [48, 21, 34, 2]. These models achieve high-fidelity video generation by pre-training on web-scale video data [1, 3]. The most common approach for video generation employs diffusion models [15, 4, 51], which model videos in a continuous latent space. Several works also explored autoregressive video modeling [20, 12], which convert videos into sequences of discrete tokens using a discrete tokenizer and apply the next-token prediction objective. Most notably, Emu3 [50] demonstrated that autoregressive models can achieve competitive performance with diffusion models at scale. There are several dedicated lines of work focusing on specific design choices of video generative models, including video tokenizer [52, 49], model architecture [37], and learning objective [45].

Foundation Models The concept of foundation models first emerged in the context of transfer learning [54], where a model trained on large-scale data in one domain can be easily fine-tuned to perform many tasks in adjacent domains. Notable early examples include self-supervised learning on ImageNet-1K, a dataset of natural images [8, 14, 35]. These pre-trained vision models proved to be versatile for a wide range of downstream applications such as medical imaging [17]. More recent works shifted the training paradigm to vision-language alignment. Models like CLIP [38] are pre-trained on large amounts of image-text pairs and have demonstrated strong zero-shot generalization capabilities to a wide range of downstream tasks across multiple domains. Most recently, several works have focused on building foundation models for domain-specific use cases such as remote sensing [40], weather forecasting [32], and material design [46]. Most notably, Cosmos [2] builds a foundation world model for physical AI by pre-training on large amounts of video documenting physical applications using the video modeling objective. Its training data covers a wide range of physical applications such as robotic manipulation and self-driving. In this work, we investigate if similar approaches can be adapted to build a foundation model for physics simulations.

B Limitations

Despite the promising success of PhysiX, we acknowledge that it has several key limitations.

Generalization. Existing foundation models typically have zero-shot generalization capabilities. For example, CLIP [38], which was pretrained on a large set of vision-language data, can perform zero-shot classification on images for domain-specific applications. While PhysiX is trained on multiple datasets, generalizing to novel physical processes requires fine-tuning, as they may have unseen input channels or represent a drastically different dynamic system from those seen during training. We leave this to future work.

Discretization Error. The tokenization process introduces quantization errors, and while the refinement module helps mitigate this, residual errors can still affect the precision of long-term simulations. This is especially significant for datasets with low spatial or temporal variance which are much more sensitive to small perturbations. Exploring alternative tokenization schemes or end-to-end training of the tokenizer and autoregressive model could help minimize this error.

Data Diversity. PhysiX was only trained on 2D datasets, due to the architecture of the video tokenizer. This limits its direct applicability to 3D physical systems or systems with significantly different spatial structures. Future work could explore more flexible tokenization architectures that enable the compression of higher spatial dimensions, and include data from outside The Well.

C Experimental settings

Refinement Module For each trajectory in the raw training data, we randomly sample a starting timestamp and run autoregressive generation to obtain the training data for the refinement module. We adopted MSE loss. We use a global batch size of 64 frames, a learning rate of $5e-3$ and a cosine decay learning rate scheduler. We trained each refinement model for 500 epochs on its respective data. Unlike the base model, which is trained in bfloat16 precision, we observe that using float32 precision is crucial to achieve high-quality outputs, especially for datasets with low spatial variance.

Tokenizer We trained the universal tokenizer on the 8 datasets in Table 1 for 1000 epochs with an effective batch size of 32. We optimize the models using AdamW [18] with a base learning rate of $1e-3$, using a 10-epoch linear warmup, followed by a cosine decay schedule for the remaining 990 epochs. For model selection, we average the validation loss across all datasets after each training epoch and use the model with the lowest validation loss as the final tokenizer checkpoint.

AR Model For the autoregressive (AR) model, we trained for 10000 steps with an effective batch size of 32. We used Adam as the optimizer with a learning rate schedule similar to the tokenizer, where the number of warmup steps is set to 1000. We validated the model after every 100 training steps and used the best checkpoint for testing. For both tokenizer and AR training, we upsampled the smaller datasets to match the size of the largest one, ensuring the model learns from each dataset uniformly.

Evaluation After training, we tested the model on the held-out test set provided by the Well [33]. For the one-step setting, we evaluated the model on random sliding windows sampled from the test simulations. For the long-horizon setting, we always initiated the model from the beginning of each simulation. This adheres to the standard practice in the Well.

Finetuning To adapt PhysiX to an unseen task, we finetune both the tokenizer and the autoregressive model. Specifically, we finetune the tokenizer for 100 epochs and the autoregressive model for 1000 iterations, with similar learning rates and schedulers to pretraining. This means the compute requirement for each finetuning task is about 10% of that of pretraining. Section H.2 shows that PhysiX was able to achieve strong performance even with this limited compute, demonstrating its usefulness for the broad research community.

D Compute resources

We trained the tokenizer and PhysiX on $8\times$ 40GB A100 devices, and evaluated using $1\times$ 40GB A100 device for each task. We trained PhysiX for 24 hours on $8\times$A100s for 8 datasets. This is approximately equal to the combined cost of training the best baseline model for each dataset at current market rate cloud compute costs [1]. Each model in The Well required 12 hours on $1\times$H100 [33], for a total time of 96 H100 hours when only considering the best model for each dataset, or about half the A100 hours used by PhysiX.

E Reproducibility statement

We will release the training and evaluation code, as well as the model checkpoints. We also note that the Well's authors [2] reported some reproducibility issues with the baseline models at the moment and are planning to update the codebase and the paper. We cite the currently reported numbers in our main experiments. For numbers not reported (e.g. longer rollouts), we use the latest version of the official codebase at the time of writing.

F Licenses

G Statistical significance

While the Well does not publish variance of the baselines for test sampling, Table 3 shows that our 95% confidence interval for 1 frame prediction with PhysiX is outside the range of the baseline mean assuming a normal distribution. For rollout predictions, we start from the beginning of each sequence and evaluate on the entire test dataset, just as the baseline was evaluated.

Table 3: **PhysiX 1 frame prediction with 95% confidence intervals**.

Dataset	Interval	Dataset	Interval
`shear_flow`	0.070 ± 0.011	`turbulent_radiative_layer`	$0.210 \pm .0344$
`rayleigh_benard`	$0.147 \pm .029$	`gray_scott_reaction`	0.021 ± 0.005
`acoustic_scattering (maze)`	$0.096 \pm .002$	`viscoelastic_instability`	0.212 ± 0.029
`active_matter`	0.090 ± 0.011	`helmholtz_staircase`	0.018 ± 0.004

H Additional experiments

H.1 Ablation Studies

To study the effectiveness of our design, we conducted a series of thorough ablation studies. In the main paper, we explored the performance of universal (multi-task) models versus single-task models, and the effectiveness of the refinement module. We provide additional ablation studies, such as training the model from scratch versus initializing the model with weights pre-trained on natural videos in the appendix.

General Model vs Task Specific Models We compare the performance of our multi-task model and single-task models on both one-frame prediction and long-horizon prediction tasks. For the task-specific model, we followed the same setup as the universal model, including the model size, model architecture, and training hyperparameters. The only difference is the training data. We report VRMSE across 8 datasets and different lead times in Table 4. Experiment results show that the universal model outperforms task-specific models, achieving lower VRMSE on the majority of datasets across different lead times. Our results show that joint multi-task training improves the performance of individual tasks, as the model may learn some common patterns and mechanisms across different physical processes.

[1] Using pricing from Lambda Labs

[2] https://github.com/PolymathicAI/the_well/issues/49

Table 4: **Comparison of multi- and single-task models.** We report next-frame and long-horizon prediction results on the Well benchmark for the multi-task and single-task models.

Dataset	Δt =1		Δt =2:8		Δt =9:26		Δt =27:56	
	Spec.	Univ.	Spec.	Univ.	Spec.	Univ.	Spec.	Univ.
`shear_flow`	**0.0689**	0.070	0.236	**0.118**	0.378	**0.281**	0.452	**0.397**
`rayleigh_benard`	**0.137**	0.147	**0.436**	1.090	**0.522**	0.704	0.724	**0.646**
`turbulent_radiative_layer`	0.359	**0.343**	0.565	**0.357**	0.792	**0.710**	1.014	**0.998**
`active_matter`	0.150	**0.090**	0.844	**0.477**	**1.177**	1.396	**1.352**	1.381
`gray_scott_reaction`	0.0418	**0.0210**	1.487	**0.0375**	15.965	**0.390**	62.484	**0.895**
`viscoelastic_instability`	0.251	**0.237**	0.764	**0.406**	—	—	—	—

Effectiveness of Refinement Module We compare PhysiX with and without the refinement module. We show such differences for both the multi-task AR model and the single-task AR model at different prediction windows in Figure 3. The refinement model reduces MSE and VRMSE metrics for both models on all prediction windows of the `gray_scott_reaction_diffusion` dataset, highlighting the effectiveness of the proposed refinement process. Most notably, with the help of refinement model, the 8-frame prediction error (0.07) of our multi-task model, measured by VRMSE, is lower than the 1-frame prediction error of the best performing baseline on the Well benchmark (0.14).

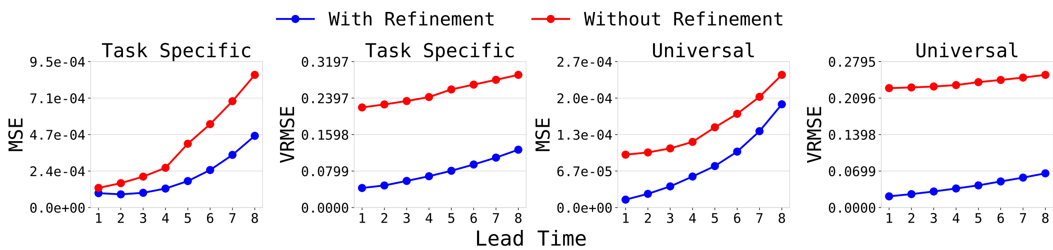

Figure 3: **Effect of refinement module.** We apply refinement module to both the multi-task and single-task AR model and study its effect on predication errors. We report VRMSE and MSE (lower is better) over prediction windows ranging from 1 frame to 8 frames on the `gray_scott_reaction_diffusion` dataset.

H.2 Adaptation to Unseen Simulations

We evaluate the adaptability of PhysiX on two unseen simulations: `euler_multi_quadrants (periodic b.c.)` and `acoustic_scattering (discontinuous)`. These tasks involve novel input channels and physical dynamics not encountered during training. To handle this distribution shift, we fully finetune the tokenizer for each task. We consider two variants of the autoregressive model: PhysiX$_f$, which finetunes the pretrained model, and PhysiX$_s$, which trains from scratch using the Cosmos checkpoint as initialization. Further finetuning details are provided in Appendix C.

Table 5 shows that PhysiX$_f$ achieves the best performance on nearly all tasks and prediction horizons, only losing to C-U-Net on one-step prediction for one task, and the performance gap widens significantly as the horizon increases. Notably, PhysiX$_f$ consistently outperforms PhysiX$_s$ across all settings, highlighting its ability to effectively transfer knowledge to previously unseen simulations.

Table 5: **Performance on two simulation tasks unseen during training.** We compare both the finetuning version (PhysiX$_f$) and the scratch version (PhysiX$_s$) with the baselines.

Models	euler_multi_quadrants (periodic b.c.)				acoustic_scattering (discontinuous)			
	$\Delta t = 1$	Δt =2:8	Δt =9:26	Δt =27:56	$\Delta t = 1$	Δt =2:8	Δt =9:26	Δt =27:56
PhysiX$_f$	**0.105**	**0.188**	**0.358**	**0.642**	0.038	**0.057**	**0.443**	**1.168**
PhysiX$_s$	**0.105**	**0.188**	0.366	0.658	0.039	0.062	0.455	1.192
FNO	0.408	1.130	1.370	–	0.127	2.146	2.752	3.135
TFNO	0.416	1.230	1.520	–	0.130	2.963	3.713	4.081
U-Net	0.183	1.020	1.630	–	0.045	2.855	6.259	8.074
C-U-Net	0.153	4.980	>10	–	**0.006**	5.160	>10	>10

H.3 Pretrained vs scratch

Figure 4 compares the performance of PhysiX when initialized from a Cosmos pretrained checkpoint (Pre-trained) vs when initialized from scratch (Random). Using the pretrained checkpoint outperforms training from scratch across almost all tasks and evaluation settings, which shows the effectiveness of PhysiX in transferring prior knowledge from natural videos to physical simulations. Table 6 details the performance of the two models.

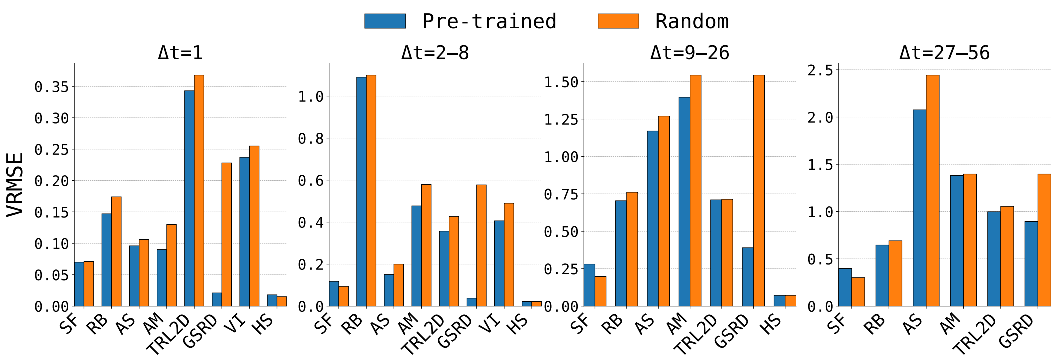

Figure 4: **Comparison of pretrained and randomly initialized weights**

Table 6: **Comparison of pre-trained and randomly initialized models.** Next-frame and long-horizon prediction results on the Well benchmark for Cosmos weights pre-trained on natural video and with randomly initialized weights.

Dataset	$\Delta t = 1$		$\Delta t = 2{:}8$		$\Delta t = 9{:}26$		$\Delta t = 27{:}56$	
	Pre.	Rand.	Pre.	Rand.	Pre.	Rand.	Pre.	Rand.
shear_flow	**0.070**	0.071	0.118	**0.094**	0.281	**0.198**	0.397	**0.301**
rayleigh_benard	**0.147**	0.174	**1.090**	1.100	**0.704**	0.761	**0.646**	0.691
acoustic_scattering (maze)	**0.096**	0.106	**0.150**	0.200	**1.170**	1.270	**2.076**	2.444
active_matter	**0.090**	0.130	**0.477**	0.579	**1.396**	1.544	**1.381**	1.397
turbulent_radiative_layer_2D	**0.343**	0.368	**0.357**	0.427	**0.710**	0.714	**0.998**	1.055
gray_scott_reaction_diffusion	**0.021**	0.228	**0.038**	0.577	**0.390**	1.544	**0.895**	1.397
viscoelastic_instability	**0.237**	0.255	**0.406**	0.490	—	—	—	—
helmholtz_staircase	0.018	**0.015**	0.022	0.022	0.072	0.072	—	—

H.4 Scaling results

We study the scalability of PhysiX by training and evaluating autoregressive models with 3 different sizes: 700M, 2B, and 4B. Since Cosmos only provides the 4B model checkpoint, we initialized all 3 models in this experiment from scratch for a fair comparison. Table 7 shows that 4B is the best performing model, followed by 700M, while 2B performed the worst. We observed that both the 4B and the 2B models overfit whereas the 700M model did not, and the 2B model converged to a worse point compared to the 700M and 4B models, leading to overall poorer performances.

Table 7: **Prediction errors for Scratch models at various time horizons.** We report next-frame and long-horizon prediction errors for Scratch 4B, Scratch 2B, and Scratch 700M across different datasets, highlighting the best (lowest) error in each horizon.

Dataset	$t+1$			$t+2{:}8$			$t+9{:}26$			$t+27{:}56$		
	4B	2B	700M	4B	2B	700M	4B	2B	700M	4B	2B	700M
`shear_flow`	**0.071**	0.075	0.073	**0.094**	0.112	0.096	0.198	0.216	**0.166**	0.301	0.303	**0.257**
`rayleigh_benard`	**0.174**	0.181	0.194	**1.10**	1.201	1.113	**0.761**	0.855	0.827	**0.691**	0.823	0.999
`acoustic_scattering (maze)`	**0.106**	0.110	0.120	**0.20**	0.211	0.237	1.270	1.284	**1.242**	2.444	2.497	**2.287**
`turbulent_radiative_layer`	0.368	0.421	**0.312**	**0.427**	0.443	0.450	**0.714**	0.758	0.730	1.055	1.099	**0.942**
`active_matter`	0.130	**0.102**	0.105	**0.579**	0.592	0.623	1.544	1.626	**1.394**	**1.397**	1.415	1.417
`gray_scott_reaction`	**0.228**	0.230	0.231	0.577	**0.509**	0.526	1.544	1.126	**1.051**	1.397	2.290	**1.300**
`viscoelastic_instability`	0.255	0.319	**0.246**	**0.490**	0.494	0.590	—	—	—	—	—	—
`helmholtz_staircase`	0.015	0.015	**0.014**	0.0224	0.019	**0.017**	0.0718	**0.056**	0.061	—	—	—

H.5 Qualitative Comparison

Figure 5 presents a qualitative comparison between PhysiX and the best-performing baseline models on two representative simulation tasks: `shear_flow` and `rayleigh_benard`. At rollout horizons of 24 and 15 steps respectively, PhysiX produces predictions that remain visually consistent with the ground truth across all physical fields, including tracer, pressure, buoyancy, and velocity components. In contrast, baseline models exhibit noticeable distortions, blurring, and loss of fine-grained structures, particularly evident in the vortex structures of `shear_flow` and the convective plumes of `rayleigh_benard`. These qualitative results highlight superior fidelity and stability of PhysiX over extended prediction windows.

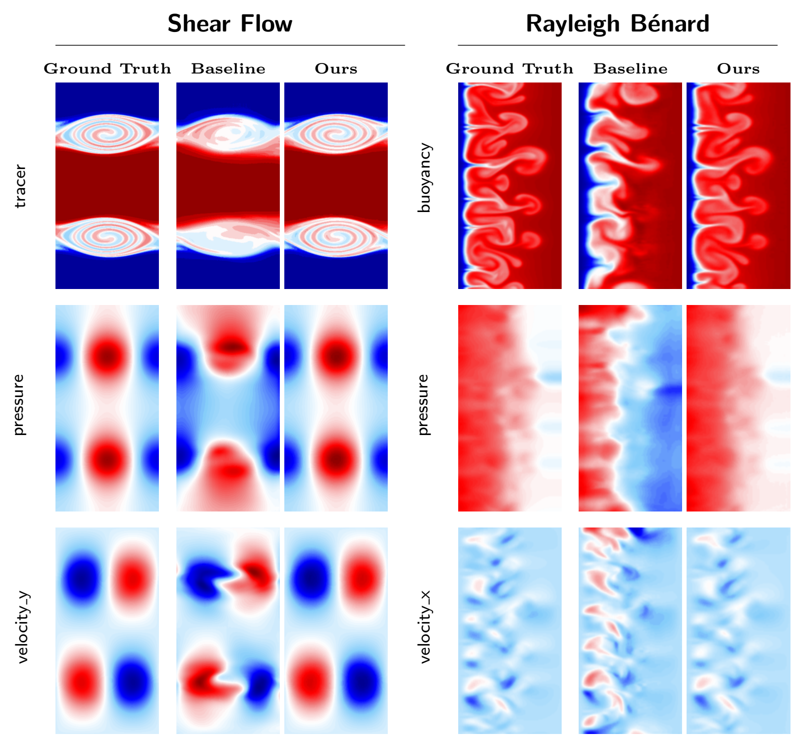

Figure 5: **Side-by-side qualitative comparison of PhysiX and baseline models.** PhysiX demonstrates superior performance in long horizon rollouts than the leading baseline model. At lead times of 24 and 15 steps for shear flow and Rayleigh–Bénard convection respectively, PhysiX maintains high-fidelity predictions across all physical fields, while baseline models ConvNeXt-UNet and TFNO exhibit visible distortions and loss of detail.

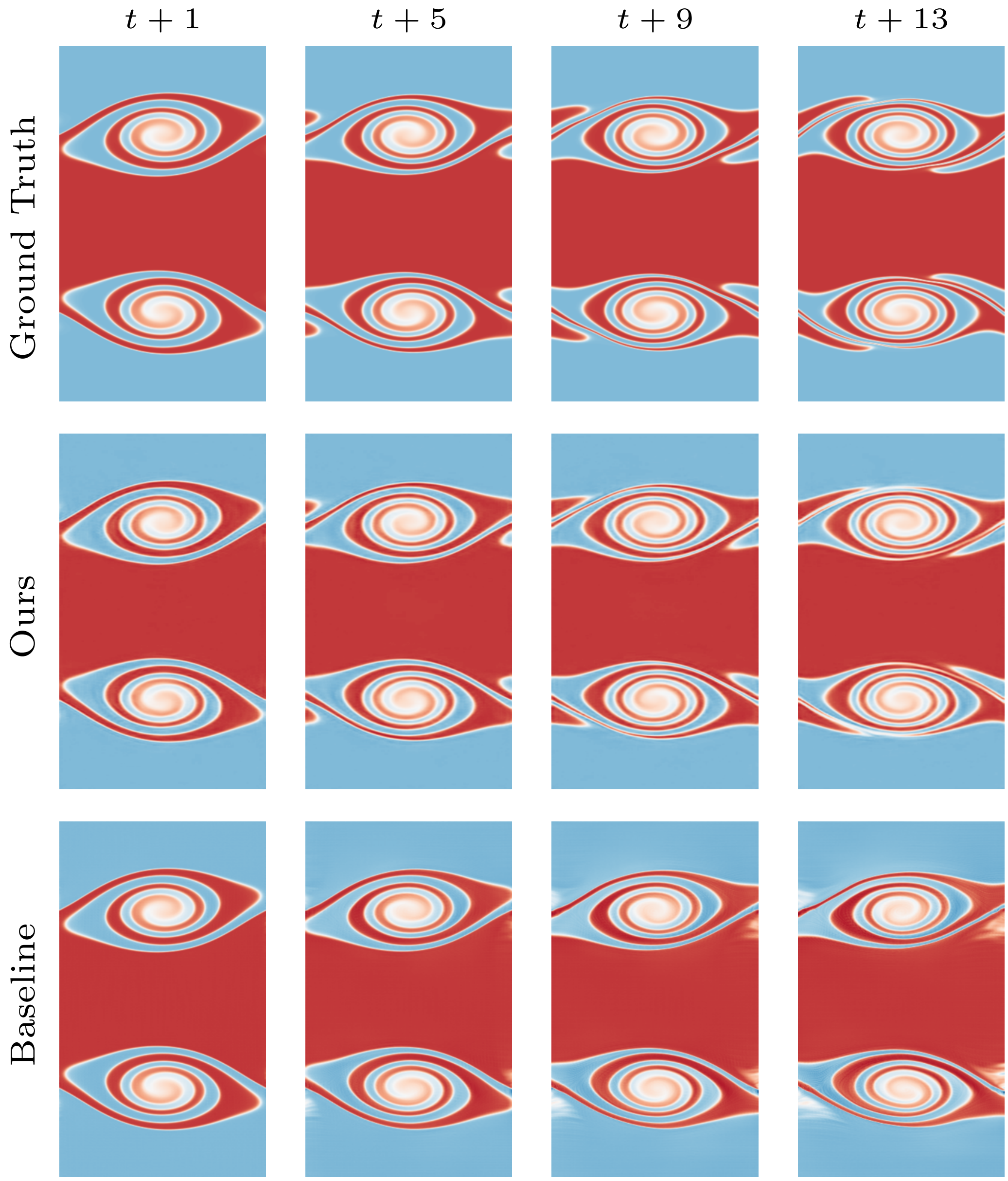

Figure 6: **Qualitative Comparisons on** `shear_flow` **Dataset.** We compare the prediction of PhysiX with the ground truth and the prediction of the best baseline model at lead times of 1,5,9,13 frames.

H.6 More qualitative results

We provide additional visualizations of the PhysiX's prediction results on `shear_flow` (Figure 6), `viscoelastic_instability` (Figure 7), `rayleigh_benard` (Figure 8) and `gray_scott_reaction_diffusion` (Figure 9). We compare the prediction of PhysiX with the ground truth and the prediction of baseline models at various lead times. PhysiX shows consistent improvement over baselines across all lead times. The improvements on longer lead times are more pronounced.

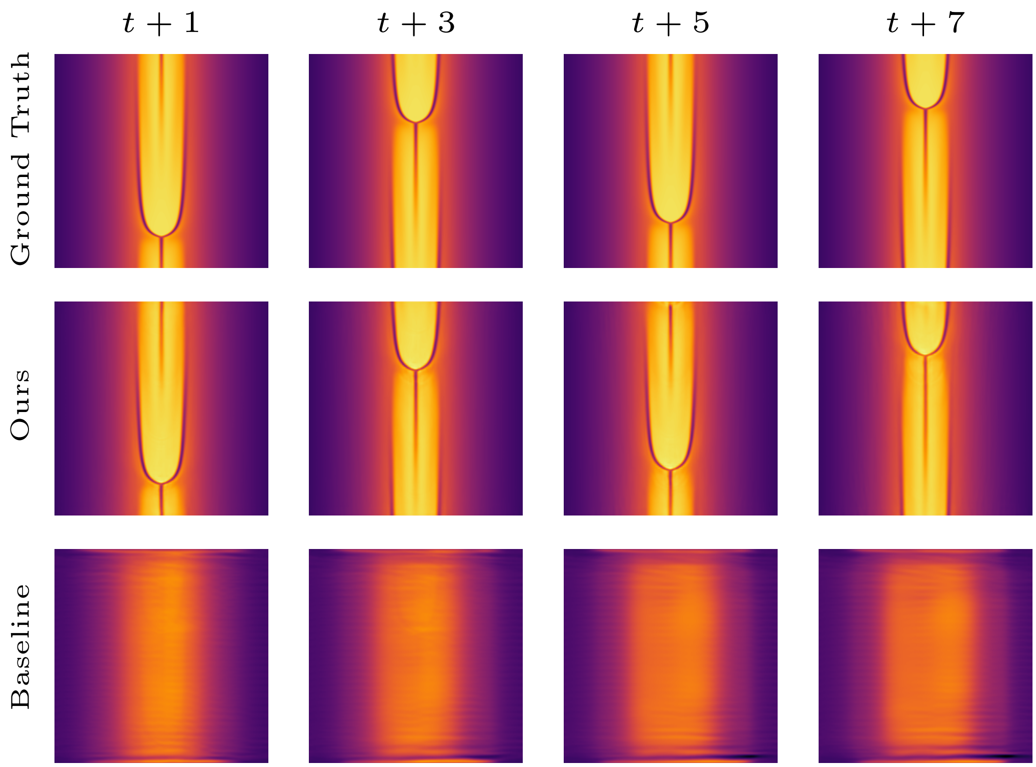

Figure 7: **Qualitative Comparisons on `viscoelastic_instability` Dataset.** We compare the prediction of PhysiX with the ground truth and the prediction of the best baseline model at lead times of 1,3,5,7 frames.

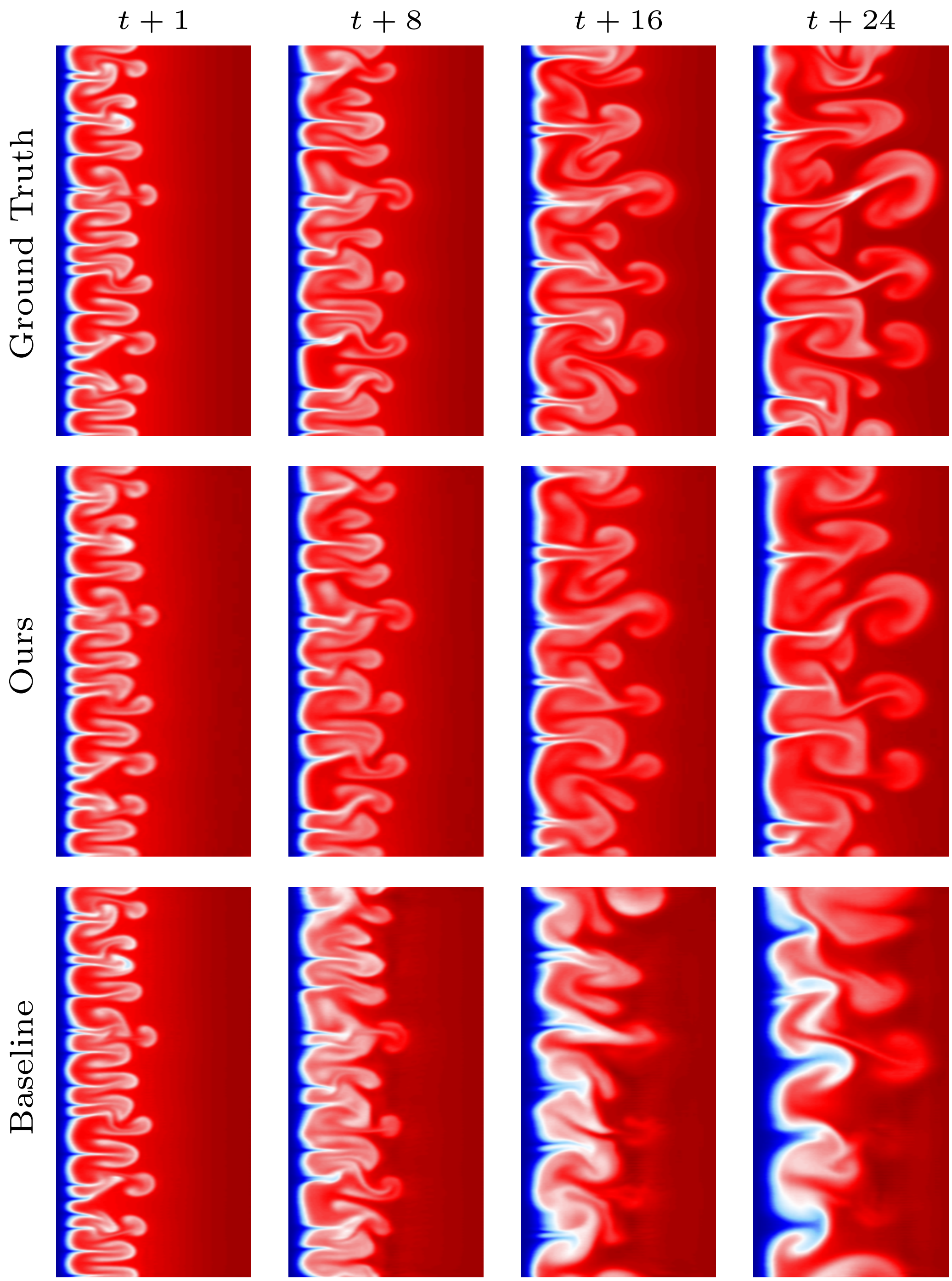

Figure 8: **Qualitative Comparisons on `rayleigh_benard` Dataset.** We compare the prediction of PhysiX with the ground truth and the prediction of the best baseline model at lead times of 1,8,16,24 frames.

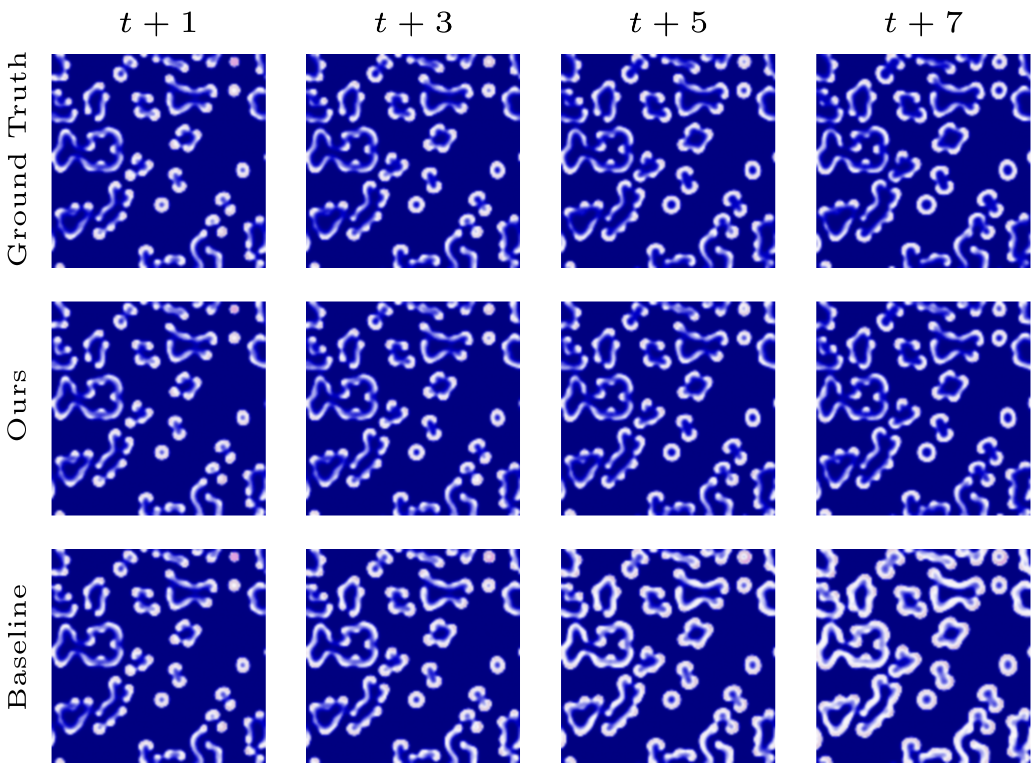

Figure 9: **Qualitative Comparisons on `gray_scott_reaction_diffusion` Dataset.** We compare the prediction of PhysiX with the ground truth and the prediction of the best baseline model at lead times of 1,3,5,7 frames.