Energy-Based Physics-Informed Diffusion Transformers Sampling for Time Series Forecasting

Defu Cao
Department of Computer Science
University of Southern California
Los Angeles, CA 90007
defucao@usc.edu

Jinbo Liu
Department of Computer Science
University of Southern California
Los Angeles, CA 90007
jinboliu@usc.edu

Yan Liu
Department of Computer Science
University of Southern California
Los Angeles, CA 90007
yanliu.cs@usc.edu

Abstract

Scientific time series data, particularly in climate science, present unique challenges at the intersection of probabilistic inference and physical constraints. While large pre-trained Time Series Diffusion Transformers excel at capturing complex data distributions, they lack mechanisms to enforce physical consistency. To address this gap, we present a novel framework for adapting pre-trained generative models for scientific tasks. Our contribution is a model-agnostic physics-injection module that employs Langevin dynamics at inference time to steer predictions toward physically consistent solutions without costly retraining. We provide theoretical guarantees for convergence under physical constraints and empirically validate our method across multiple synthetic partial differential equations and climate systems, offering insights into the synergy between machine learning and physics-based sampling for scientific applications.

1 Introduction

Time series forecasting is a critical component of scientific advancement [1, 2], especially in fields like climate science [3], where accurate predictions are essential for understanding and mitigating the impacts of climate change. The advent of large-scale, pre-trained Time Diffusion Transformers (TimeDiT) [4] has shown immense promise. However, scientific time series data are often characterized by a unique set of challenges that these general-purpose models do not inherently address:

- Data Scarcity: High-fidelity climate simulations and observations can be prohibitively expensive to obtain, leading to limited datasets for fine-tuning.
- Physical Consistency: Forecasts must adhere to fundamental physical laws, such as conservation of energy and mass, to be considered valid and trustworthy.
- Imperfect Data: Real-world data are often incomplete, with missing values and irregular sampling intervals that can confound standard models.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Frontiers in Probabilistic Inference: Sampling Meets Learning.

Algorithm 1 Physics-Informed Energy-based Sampling

```
$\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T, \ldots, 1$ do
    $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ if $t > 1$, else $\boldsymbol{z} = \mathbf{0}$
    $\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right) + \sigma_t \boldsymbol{z}$
end for
for $j = 0, 1, \ldots, k-1$ do
    $\boldsymbol{x}_{j+1}^{tar} = \boldsymbol{x}_j^{tar} + \epsilon \nabla K(\boldsymbol{x}_j^{tar}; \boldsymbol{x}^{obs}) + \alpha\epsilon\nabla \log p(\boldsymbol{x}_j^{tar} | \boldsymbol{x}^{obs}) + \sqrt{2\epsilon}\sigma, \sigma \sim \mathcal{N}(0, 1)$
end for
return $\boldsymbol{x}_k^{tar}$
```

Although large-scale time series foundation models, such as Moirai [5], Chronos [6] and TEMPO [7], are adept at capturing intricate data distributions, even from irregular inputs, they lack an intrinsic mechanism to ensure their predictions adhere to fundamental physical laws. In contrast, approaches like Physics-Informed Neural Networks (PINNs) [8] integrate physical laws directly into the training objective, but this often comes at the cost of data efficiency and forgoes the significant advantages of leveraging a powerful, pre-trained foundation model.

We bridge this gap with PINFDiT (Physics-Informed Diffusion Transformers), a framework that elegantly incorporates physics knowledge into pre-trained diffusion transformers through inference-time guidance rather than architectural modifications. This approach preserves the expressive power of foundation models while ensuring predictions remain consistent with governing physical laws. Our contributions advance the intersection of probabilistic inference and scientific computing by: (1) establishing a principled connection between diffusion-based generative models and classical Langevin sampling through physics constraints, (2) demonstrating how learned representations accelerate traditional physics-based sampling in climate applications, and (3) providing rigorous theoretical guarantees for convergence under physical regularization.

2 Methodology: From Pre-trained TimeDiT to Physics-Informed Inference

The PINFDiT framework is designed to transform a pre-trained foundation model, such as TimeDiT [4], into a physics-informed forecasting engine. It leverages the existing capabilities of the pre-trained model and enhances it with a novel physics-informed sampling module. Physics principles fundamentally govern the evolution of temporal signals in scientific phenomena like climate patterns and oceanographic data. Integrating this physical knowledge into foundational time series models is therefore essential. In this section, we develop a decoding method ensuring that $\mathbf{x}^{\text{tar}}$ generated by PINFDiT satisfies physical law constraints. Unlike the previous SBI solution [9, 10], which requires access to a faithful, differentiable simulator, we target the state-trajectory distribution directly by incorporating physics knowledge as an energy-based prior during inference. This follows the paradigm of separating general model learning from targeted physical refinement at inference time, thus can be treated as a plugin mechanism requiring no architectural modifications or retraining.

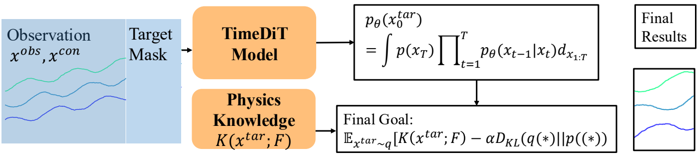
Observation
x^{obs}, x^{con}
Target
Mask
TimeDiT
Model
Physics
Knowledge
$K(x^{tar}; F)$
$p_\theta(x_0^{tar})$
$= \int p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) d_{x_{1:T}}$
Final Goal:
$\mathbb{E}_{x^{tar} \sim q}[K(x^{tar}; F) - \alpha D_{KL}(q(*)||p((*))$
Final
Results

Figure 1: Physics-informed sampling with the pre-trained TimeDiT.

Generalized Goal of Physics Injection for Time Series: We first start with a brief introduction to physical laws and PDE. A generic form of a physical law represented as a PDE that describes the evolution of a continuous temporal signal $\mathbf{x}(\mathbf{u}, t)$ over a spatial coordinate $\mathbf{u}$ is given by:

$$\frac{\partial \mathbf{x}}{\partial t} = F(t, \mathbf{x}, \mathbf{u}, \frac{\partial \mathbf{x}}{\partial \mathbf{u}_i}, \frac{\partial^2 \mathbf{x}}{\partial \mathbf{u}_i \partial \mathbf{u}_j}, \ldots) \quad (1)$$

Based on this PDE representation of physical knowledge, the consistency between the predicted time series $\mathbf{x}^{\text{tar}}$ and the physics knowledge can be quantified using the following squared residual function:

$$K(\mathbf{x}^{\text{tar}}; F) = -||\frac{\partial \mathbf{x}^{\text{tar}}}{\partial t} - F(t, \mathbf{x}^{\text{tar}}, \mathbf{u}, \frac{\partial \mathbf{x}^{\text{tar}}}{\partial \mathbf{u}_i}, \frac{\partial^2 \mathbf{x}^{\text{tar}}}{\partial \mathbf{u}_i \partial \mathbf{u}_j}, \dots)||_2^2 \tag{2}$$

This function reaches its maximum when the predicted time series is perfectly consistent with the physical model, resulting in a residual of 0. Using this metric K, physics knowledge can be integrated into a probabilistic time series foundation model $p(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$ as an explicit regularization by solving the following optimization problem to obtain a refined model $q(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$:

$$q(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}}) = \arg\max_{q} \left[\mathbb{E}_{\mathbf{x}^{\text{tar}} \sim q} K(\mathbf{x}^{\text{tar}}; F) - \alpha D_{KL}(q(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})||p(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}}))\right] \tag{3}$$

where the first term represents the aforementioned physics knowledge metric, and the second term controls the divergence between $q(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$ and $p(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$.

Closed-Form Langevin Correction via Boltzmann Energy Distribution: However, directly updating the model parameters to optimize the above function is resource-consuming. To solve this issue, we derive the closed-form solution, which does not need to update the model parameters. The above optimization problem has a closed-form solution as provided by the following theorem:

Theorem 2.1 (Boltzmann Energy Distribution)**.** *The optimal $q(\mathbf{x}^{tar}|\mathbf{x}^{con})$ in Eq.3 is the Boltzmann distribution defined on the following energy function: $E(\mathbf{x}^{tar}; \mathbf{x}^{con}) = K(\mathbf{x}^{tar}; F) + \alpha \log p(\mathbf{x}^{tar}|\mathbf{x}^{con})$, in other words, the optimal $q(\mathbf{x}^{tar}|\mathbf{x}^{con})$ is:*

$$q(\mathbf{x}^{tar}|\mathbf{x}^{con}) = \frac{1}{Z}\exp(K(\mathbf{x}^{tar}; F) + \alpha \log p(\mathbf{x}^{tar}|\mathbf{x}^{con})), \tag{4}$$

where $Z = \int \exp(K(\mathbf{x}^{tar}; F) + \alpha \log p(\mathbf{x}^{tar}|\mathbf{x}^{con}))d\mathbf{x}^{tar}$ is the partition function.

The theorem illustrates that sampling from the Boltzmann distribution is analogous to incorporating physics knowledge into model edition. In the context of diffusion models, this distribution can be effectively sampled using Langevin dynamics [11]:

$$\begin{aligned} \mathbf{x}_{j+1}^{\text{tar}} &= \mathbf{x}_j^{\text{tar}} + \epsilon \nabla \log q(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}}) + \sqrt{2\epsilon}\sigma, \sigma \sim \mathcal{N}(0,1) \\ &= \mathbf{x}_j^{\text{tar}} + \epsilon \nabla K(\mathbf{x}_j^{\text{tar}}; \mathbf{x}^{\text{con}}) + \alpha\epsilon \nabla \log p(\mathbf{x}_j^{\text{tar}}|\mathbf{x}^{\text{con}}) + \sqrt{2\epsilon}\sigma, \sigma \sim \mathcal{N}(0,1) \end{aligned} \tag{5}$$

where $\sigma \sim \mathcal{N}(0,1)$. In diffusion model, precisely calculate the likelihood $\log p(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$ is intractable. To tackle this issue, following previous works [12], we approximate likelihood with the objective to edit the pre-trained diffusion model: $\log p(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}}) = -\mathbb{E}_{\epsilon,t}[||\epsilon_\theta(\mathbf{x}^{\text{tar}}, t; \mathbf{x}^{\text{con}}) - \epsilon||^2]$. The approximation presented above constitutes the optimizable component of the evidence lower bound(ELBO). Algorithm 1 summarizes the comprehensive model editing process. By integrating physics knowledge through Langevin dynamics during inference, we achieve a physics-consistent refinement without compromising the model's ability to learn from data and generate diverse, physically plausible scenarios.

Theoretical Guarantees and Model-Agnostic Convergence Properties: We now present theoretical guarantees for our diffusion model's convergence properties and it can be easily extended to other models:

Theorem 2.2 (Physics-Informed Inference Plugin Convergence)**.** *Let $p_\theta(\mathbf{x}^{tar}|\mathbf{x}^{con})$ be the conditional distribution defined by a pre-trained diffusion model with score function ϵ_θ. Let F represent a physical law with residual function $K(\mathbf{x}^{tar}; F)$ in Eq.(2). For the physics-informed plugin with step size ϵ and N refinement steps, the samples of Langevin dynamics Eq.(5) converge to the optimal goal of Eq.(3), with a convergence rate of:*

$$D_{\mathrm{KL}}(q_N \,\|\, q^*) \leq \mathcal{O}\left(\frac{d}{\sqrt{N}} + \varepsilon_{score}^2\right), \tag{6}$$

where q_N is the distribution after N refinement steps and d is the effective dimension of $\mathbf{x}^{tar}$ state space, ε_{score}^2 represents the squared error in the score estimation.

When we set the step size (line 7 in Algo. 1) $\epsilon = \Theta(N^{-1/2})$ [13], the converge rate can be $\mathcal{O}(N^{-1/2})$. The critical connection between statistical convergence and physical accuracy is established in Lemma 2.3, which demonstrates that improvements in KL divergence directly translate to enhanced physical consistency:

Lemma 2.3 (Residual–Variance Coupling)**.** *Let $\widetilde{r} = \partial_t \mathbf{x} - F(\cdot)$ be the physical residual of any sample $\mathbf{x} \sim q$. If F is L-Lipschitz and the surrogate bias is δ, then for all q absolutely continuous w.r.t. q^*, $\mathrm{Var}_q[\widetilde{r}] \leq 2L^2\, D_{\mathrm{KL}}(q \,\|\, q^*) + 4L^2\delta^2$.*

We establish a direct relationship between statistical convergence and physical consistency: each $\sqrt{N}$-step improvement in KL divergence systematically reduces the variance of the physics residual, ensuring that our refinement process progressively enhances the physical validity of our predictions. This relationship offers practitioners a clear interpretability pathway: improvements in model convergence directly translate to enhanced physical consistency, allowing users to understand how refinement steps progressively reduce violations of physical laws. We extend the theorems to establish that $p_M(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$ can represent the conditional distribution defined by other models M, making our approach model-agnostic. The complete model-agnostic proofs are provided in Appendix B.3 and B.4.

3 Experiments

To validate our framework in a challenging, real-world scenario, following [14], we obtain the ERA5 dataset from WeatherBench2 at a resolution of 5.625 degrees (64×32 grid points), following established protocols with this data split: training data from 2006-2015 (10 years), validation data from 2016 (1 year), and test data from 2017-2018 (2 years). In this work, we focus exclusively on the 2-meter temperature (t2m) variable. This benchmark is critical for assessing a model's ability to capture the complex dynamics of the Earth's climate system. The physics knowledge was incorporated by enforcing momentum conservation through a quasi-geostrophic approximation of the Navier-Stokes equations during inference. We compared our full PINFDiT framework against two state-of-the-art climate forecasting models, ClimaX and ClimODE, as well as an ablated version of our own model without the physics-injection module (PINFDiT w/o physics). Performance was measured using the Anomaly Correlation Coefficient (ACC) at different forecast horizons.

Table 1: Performance comparison on ERA5 2m temperature prediction

Model	6h ACC	12h ACC	24h ACC
ClimaX	0.920	0.900	0.890
ClimODE	0.970	0.960	0.960
PINFDiT (w/o physics)	0.985	0.984	0.985
PINFDiT (full)	**0.987**	**0.987**	**0.987**

The results clearly demonstrate the superiority of the PINFDiT framework. The base pre-trained model already outperforms existing specialized climate models. Critically, the addition of our physics-injection module provides consistent improvements across all time horizons. The physics-informed sampling provides consistent improvements, particularly for longer horizons where maintaining physical consistency becomes crucial for preventing error accumulation and divergence from realistic climate trajectories.

3.1 Generalization on Proposed Physics Injection Method

Table 2: PINFDiT's performance on Zero-shot CFD with different parameters.

Model	MSE	RMSE	MAE	CRPS	CRPS_sum	MSE	RMSE	MAE	CRPS	CRPS_sum
$\eta = 0.0010$(Low), $\zeta = 0.0010$(Low)						$\eta = 0.0100$(High), $\zeta = 0.0010$(Low)				
DDPM	0.0076	0.0874	0.0720	0.1012	0.0988	0.0035	0.0588	0.0485	0.0764	0.0622
DDIM	0.0590	0.2428	0.2340	0.3663	0.3918	0.0370	0.1923	0.1843	0.3119	0.3320
TSDiff	0.2844	0.5333	0.5256	1.0000	1.0000	0.2268	0.4762	0.4747	0.9999	0.9999
PINFDiT	0.0046	0.0681	0.0558	0.0859	0.0755	0.0030	0.0547	0.0450	0.0714	0.0571
$\eta = 0.0010$(Low), $\zeta = 0.0100$(High)						$\eta = 0.0100$(High), $\zeta = 0.0100$(High)				
DDPM	0.0052	0.0719	0.0598	0.0910	0.0816	0.0042	0.0652	0.0530	0.0821	0.0719
DDIM	0.0464	0.2154	0.2043	0.3436	0.3694	0.0401	0.2002	0.1901	0.3227	0.3421
TSDiff	0.2497	0.4997	0.4858	0.9999	0.9999	0.2304	0.4800	0.4766	0.9999	0.9999
PINFDiT	0.0046	0.0679	0.0559	0.0857	0.0764	0.0038	0.0613	0.0493	0.0772	0.0661

To validate PINFDiT 's ability to capture complex physical dynamics, we evaluated its performance on zero-shot CFD prediction tasks across varying physical parameters. The demonstration of CFD flow can be found at Figure 2. We systematically tested two key parameters that govern fluid behavior: diffusion coefficient (η) and oscillation frequency (ζ). Higher η values produce smoother solutions with increased diffusion, while lower values create sharper gradients; higher ζ values generate higher frequency oscillations with smaller structures, while lower values result in lower frequency oscillations with larger structures. As shown in Table 2, PINFDiT consistently outperformed baseline models with different sampling strategies (DDPM [15], DDIM [16], and TSDiff [12]) across all parameter configurations, demonstrating its robust ability to adapt to different physical regimes without retraining. Notably, PINFDiT maintained superior performance even in the most challenging scenario with high oscillation frequency ($\zeta = 0.0100$) and low diffusion ($\eta = 0.0010$), where sharp gradients and complex small-scale structures coexist. These results confirm PINFDiT 's effectiveness in incorporating physical constraints and capturing the underlying dynamics of complex fluid systems, highlighting its potential for scientific applications requiring accurate simulation of physical phenomena.

4 Conclusion

The PINFDiT framework represents a significant step forward in making general-purpose time series models suitable for scientific applications. By augmenting a powerful pre-trained diffusion transformer with a flexible, inference-time physics-injection module, PINFDiT effectively addresses the key challenges of scientific time series analysis. For future work, we see several exciting directions for cross-disciplinary collaboration. Extending the physics-informed sampling framework to a broader class of physical constraints and exploring more sophisticated sampling methods are promising avenues. Additionally, developing a deeper theoretical understanding of the interplay between data-driven priors and physics-based constraints will be crucial for building the next generation of scientific machine learning models.

Acknowledgement

This work is supported by the NSF Award #2425919. The funding from these sources has been a cornerstone in enabling us to bring our project to fruition. We are also deeply grateful to the anonymous reviewers for their rigorous review process. Their detailed comments and constructive suggestions have significantly contributed to the improvement of this paper. The time and effort they invested in providing feedback have been invaluable and have greatly helped us refine our work.

References

[1] Robert Fildes, Andrew Harvey, Mike West, and Jeff Harrison. Forecasting, structural time series models and the kalman filter. *The Journal of the Operational Research Society*, 42:1031, 11 1991.

[2] Juan Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learning Research*, 2023.

[3] Puja Das, August Posch, Nathan Barber, Michael Hicks, Kate Duffy, Thomas Vandal, Debjani Singh, Katie van Werkhoven, and Auroop R Ganguly. Hybrid physics-ai outperforms numerical weather prediction for extreme precipitation nowcasting. *npj Climate and Atmospheric Science*, 7(1):282, 2024.

[4] Defu Cao, Wen Ye, Yizhou Zhang, and Yan Liu. Timedit: General-purpose diffusion transformers for time series foundation model. *arXiv preprint arXiv:2409.02322*, 2024.

[5] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

[6] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

[7] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023.

[8] Zhiyuan Zhao, Xueying Ding, and B. Aditya Prakash. PINNsformer: A transformer-based framework for physics-informed neural networks. In *The Twelfth International Conference on Learning Representations*, 2024.

[9] Yifei Xiong, Xiliang Yang, Sanguo Zhang, and Zhijian He. An efficient likelihood-free bayesian inference method based on sequential neural posterior estimation. *Communications in Statistics-Simulation and Computation*, pages 1–26, 2025.

[10] Julia Linhart, Gabriel Victorino Cardoso, Alexandre Gramfort, Sylvain Le Corff, and Pedro LC Rodrigues. Diffusion posterior sampling for simulation-based inference in tall data settings. *arXiv preprint arXiv:2404.07593*, 2024.

[11] Gabriel Stoltz, Mathias Rousset, et al. *Free energy computations: A mathematical perspective*. World Scientific, 2010.

[12] Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Bernie Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[13] Arnak Dalalyan and Avetik Karagulyan. User-friendly guarantees for the langevin monte carlo with inaccurate gradient. *Stochastic Processes and their Applications*, 129(12):5278–5311, 2019.

[14] Shixuan Li, Wei Yang, Peiyu Zhang, Xiongye Xiao, Defu Cao, Yuehan Qin, Xiaole Zhang, Yue Zhao, and Paul Bogdan. Climatellm: Efficient weather forecasting via frequency-aware large language models. *arXiv preprint arXiv:2502.11059*, 2025.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[16] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[17] Maziar Raissi, Paris Perdikaris, and George E 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.

[18] 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.

[19] Zongyi Li, Nikola Borislavov 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*, 2021.

[20] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[21] Jonas Beck, Nathanael Bosch, Michael Deistler, Kyra L Kadhim, Jakob H Macke, Philipp Hennig, and Philipp Berens. Diffusion tempering improves parameter estimation with probabilistic integrators for ordinary differential equations. In *International Conference on Machine Learning*, pages 3305–3326. PMLR, 2024.

[22] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural Controlled Differential Equations for Irregular Time Series. *Advances in Neural Information Processing Systems*, 2020.

[23] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

[24] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

[25] Zexi Fan, Yan Sun, Shihao Yang, and Yiping Lu. Physics-informed inference time scaling via simulation-calibrated scientific machine learning. *arXiv preprint arXiv:2504.16172*, 2025.

[26] Qingyi Lin, Chuang Zhang, Xuhui Meng, and Zhaoli Guo. Monte carlo physics-informed neural networks for multiscale heat conduction via phonon boltzmann transport equation. *arXiv preprint arXiv:2408.10965*, 2024.

[27] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

[28] Jing Qiu, Jiancheng Huang, Xiangdong Zhang, Zeng Lin, Minglei Pan, Zengding Liu, and Fen Miao. Pi-fusion: Physics-informed diffusion model for learning fluid dynamics. *arXiv preprint arXiv:2406.03711*, 2024.

[29] Jan-Matthis Lueckmann, Pedro J Goncalves, Giacomo Bassetto, Kaan Öcal, Marcel Nonnenmacher, and Jakob H Macke. Flexible statistical inference for mechanistic models of neural dynamics. *Advances in neural information processing systems*, 30, 2017.

[30] Manuel Gloeckler, Shoji Toyota, Kenji Fukumizu, and Jakob H Macke. Compositional simulation-based inference for time series. In *The Thirteenth International Conference on Learning Representations*, 2024.

[31] Guangsi Shi, Daokun Zhang, Ming Jin, Shirui Pan, and Philip S Yu. Towards complex dynamic physics system simulation with graph neural odes. *arXiv preprint arXiv:2305.12334*, 2023.

[32] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

[33] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. VDT: General-purpose video diffusion transformers via mask modeling. In *The Twelfth International Conference on Learning Representations*, 2024.

[34] Yang Sui, Yanyu Li, Anil Kag, Yerlan Idelbayev, Junli Cao, Ju Hu, Dhritiman Sagar, Bo Yuan, Sergey Tulyakov, and Jian Ren. Bitsfusion: 1.99 bits weight quantization of diffusion model. *arXiv preprint arXiv:2406.04333*, 2024.

[35] Yang Sui, Huy Phan, Jinqi Xiao, Tianfang Zhang, Zijie Tang, Cong Shi, Yan Wang, Yingying Chen, and Bo Yuan. Disdet: Exploring detectability of backdoor attack on diffusion models. *arXiv preprint arXiv:2402.02739*, 2024.

[36] Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Y Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[37] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.

[38] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

[39] Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. $\mathbf{S}^2$ ip-llm: Semantic space informed prompt learning with llm for time series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

[40] Vijay Ekambaram, Arindam Jati, Nam H Nguyen, Pankaj Dayama, Chandra Reddy, Wesley M Gifford, and Jayant Kalagnanam. Ttms: Fast multi-level tiny time mixers for improved zero-shot and few-shot forecasting of multivariate time series. *arXiv preprint arXiv:2401.03955*, 2024.

[41] Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. *Advances in Neural Information Processing Systems*, 35:23009–23022, 2022.

[42] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pages 8857–8868. PMLR, 2021.

[43] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

[44] Yuhang Chen, Chaoyun Zhang, Minghua Ma, Yudong Liu, Ruomeng Ding, Bowen Li, Shilin He, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. Imdiffusion: Imputed diffusion models for multivariate time series anomaly detection. *Proceedings of the VLDB Endowment*, 17(3):359–372, 2023.

[45] Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. *arXiv preprint arXiv:2403.01742*, 2024.

[46] Yiyuan Yang, Ming Jin, Haomin Wen, Chaoli Zhang, Yuxuan Liang, Lintao Ma, Yi Wang, Chenghao Liu, Bin Yang, Zenglin Xu, et al. A survey on diffusion models for time series and spatio-temporal data. *arXiv preprint arXiv:2404.18886*, 2024.

A Related Work

Physics-Informed Machine Learning. Physics has traditionally been injected into machine learning systems *during training* by augmenting the loss with PDE residuals, as in physics-informed neural networks (PINNs) [17], operator learners such as DeepONet [18] and FNO [19], or more recent physics-aware transformers [8]. NeuralODEs [20, 21] and their extension, NeuralCDEs [22], explicitly model latent dynamics as ordinary or controlled differential equations, thereby embedding continuous-time evolution into the network design. When the vector field is further regularised to respect conservation laws or Hamiltonian structure [23, 24], the resulting trajectories satisfy first-principle constraints throughout training, akin to PINNs and operator learners. In contrast, we follow the emerging line of *inference-time debiasing* that corrects a frozen surrogate without retraining; Simulation-Calibrated SML [25] and Monte-Carlo importance pinning [26] exemplify this idea but have so far been limited to low-dimensional surrogates. Diffusion-based generators have also been coupled to physical laws, e.g. SDEdit-PDE [27] and FluidDiffusion [28], yet these works focus on single snapshots or images rather than long, irregular time series. Orthogonally, simulation-based inference (SBI) methods such as SNPE [29], CSBI [30] and LFBC [9] exploit mechanistic simulators to learn parameter posteriors. Finally, hybrid forecasters that blend neural predictors with numerical weather models or other filters [3, 31] illustrate the benefits of combining data-driven and physics-based cues. Distinct from all prior art, PINFDiT marries a large-context transformer with a diffusion prior and a *plug-and-play Langevin corrector*: physics enters only through an energy term, giving a model-agnostic, zero-retraining route to enforce conservation laws while preserving the generative flexibility of foundation models.

Diffusion models for Time Series. Despite growing interest in diffusion models across various scenarios [32, 33, 34, 35], their application in time series analysis remains less explored compared to pre-trained language models [36, 37, 38, 39, 40]. Most existing studies also focus solely on forecasting and the choice of backbone model also varies among VAE[41], RNN[42], and transformers. Recently, CSDI [43] first utilizes a diffusion model for time series imputation with a self-supervised approach. SSSD [2] combines the structured state space model with the diffusion model for imputation. ImDiffusion [44] leverages diffusion models as time series imputers to achieve accurate anomaly detection. D^{3}VAE [41] proposes a generative time series forecasting method on top of VAE equipped with the diffusion model. Meanwhile, DiffusionTS [45] incorporates decomposition into the diffusion model to improve interoperability. Although TSDiff [12] build a diffusion pipeline for multiple tasks with refinement, they still train different models for each task. Based on our knowledge, no unified diffusion transformer model has yet been explored for a comprehensive set of time series tasks. For a thorough literature review on diffusion models in time series analysis, please refer to [46].

B Discussion on Physics-Informed PINFDiT

The tension between physical constraints and learned distributions in PINFDiT is managed through a sophisticated energy-based optimization framework that combines two key components:

- the physics knowledge represented by function $K(x^{\text{tar}}; F)$, which measures PDE residuals for physical law conformity
- the learned probabilistic distribution $p(x^{\text{tar}}|x^{\text{con}})$ from the diffusion model

This balance is achieved through an energy function:

$$E(x^{\text{tar}}; x^{\text{con}}) = K(x^{\text{tar}}; F) + \alpha \log p(x^{\text{tar}}|x^{\text{con}})$$

where the parameter α controls the trade-off between physical consistency and distribution fidelity.

Rather than directly modifying model parameters, PINFDiT implements this balance through an iterative sampling procedure that:

1. starts with samples from the learned distribution
2. gradually refines them using physical gradients while maintaining probabilistic characteristics

This approach allows the model to generate samples that respect both the learned patterns in the data and the underlying physical laws without significantly compromising either aspect, ultimately resolving the tension through a theoretically-grounded Boltzmann distribution as the optimal solution.

B.1 Empirical Results on Convergence Characteristics

The convergence plots in Figure 3 comparing the Navier-Stokes (NS) and 1D-Vorticity simulations demonstrate strong alignment with the theoretical $\mathcal{O}(N^{-1/2})$ convergence rate expected for numerical simulations. Both models exhibit a clear linear relationship when error metrics (MAE and MSE) are plotted against $N^{-1/2}$, as indicated by the red dashed trend lines. Notably, the NS simulation displays nearly perfect linear scaling for

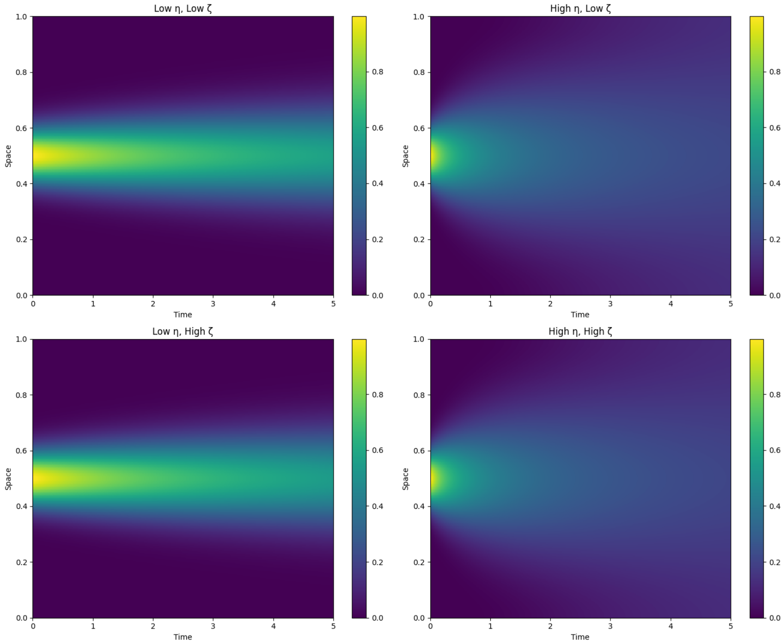

Figure 2: The two key parameters that govern CFD behavior: diffusion coefficient (η) and oscillation frequency (ζ). Higher η values produce smoother solutions with increased diffusion, while lower values create sharper gradients; higher ζ values generate higher frequency oscillations with smaller structures, while lower values result in lower frequency oscillations with larger structures.

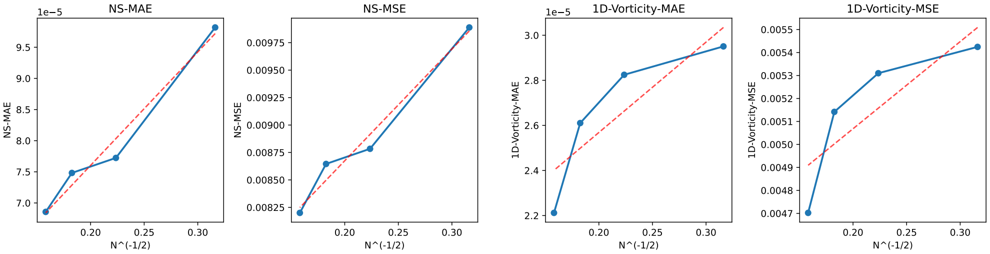

Figure 3: Empirical Convergence Analysis of NS and 1D-Vorticity Models. Both results demonstrate linear relationships consistent with theoretical expectations, with the NS simulation showing stronger linearity but higher absolute error values compared to the 1D-Vorticity simulation, reflecting differences in their underlying physical complexity.

both MAE and MSE, suggesting the numerical errors decrease predictably as the number of time steps increases. In contrast, the 1D-Vorticity model shows a slight deviation from perfect linear convergence, with a flattening of the curve at intermediate values of N. This behavior indicates that the 1D-Vorticity simulation may be approaching a lower error bound more rapidly than the NS model or encountering additional error sources that aren't purely time-step dependent. While both methods converge at the theoretical $\mathcal{O}(N^{-1/2})$ rate, the 1D-Vorticity simulation achieves lower absolute error with the same computational budget. These results confirm that both models follow the expected theoretical convergence rate, while revealing important differences in their numerical behavior that reflect the underlying physical complexity and implementation characteristics of each approach.

B.2 Proof of Physics-Informed PINFDiT Theorem 2.1

Theorem B.1. *The optimal $q(\mathbf{x}^{tar}|\mathbf{x}^{con})$ in Eq.2 is the Boltzmann distribution defined on the following energy function:*

$$E(\mathbf{x}^{tar};\mathbf{x}^{con}) = K(\mathbf{x}^{tar};F) + \alpha \log p(\mathbf{x}^{tar}|\mathbf{x}^{con}) \tag{7}$$

in other words, the optimal $q(\mathbf{x}^{tar}|\mathbf{x}^{con})$ *is:*

$$q(\mathbf{x}^{tar}|\mathbf{x}^{con}) = \frac{1}{Z}\exp(K(\mathbf{x}^{tar};F) + \alpha\log p(\mathbf{x}^{tar}|\mathbf{x}^{con})), \tag{8}$$

where $Z = \int \exp(K(\mathbf{x}^{tar};F) + \alpha\log p(\mathbf{x}^{tar}|\mathbf{x}^{con}))d\mathbf{x}^{tar}$ *is the partition function.*

Proof. Let us consider the objective function:

$$\begin{aligned} O(q(y|x)) &= \mathbb{E}_{y\sim q(y|x)}K(y) - \alpha D_{KL}(q(y|x)||p(y|x)) \\ &= \mathbb{E}_{y\sim q(y|x)}K(y) - \alpha\int_y q(y|x)\log(\frac{q(y|x)}{p(y|x)})dy \\ &= \int_y q(y|x)[K(y) + \alpha\log p(y|x) - \alpha\log q(y|x)]dy \end{aligned} \tag{9}$$

We try to find the optimal $q(y|x)$ through Lagrange multipliers. The constraint of the above objective function is that $q(y|x)$ is a valid $\int_y q(y|x)dy = 1$. Thus, the Lagrangian is:

$$\begin{aligned} L(q(y|x),\lambda) &= \int_y q(y|x)[K(y) + \alpha\log p(y|x) - \alpha\log q(y|x)]dy - \lambda(\int_y q(y|x)dy - 1) \\ &= \int_y q(y|x)[K(y) + \alpha\log p(y|x) - \alpha\log q(y|x) - \lambda q(y|x)]dy + \lambda \end{aligned} \tag{10}$$

We define $f(q(y|x),y,\lambda) = q(y|x)[K(y) + \alpha\log p(y|x) - \alpha\log q(y|x) - \lambda] + \lambda h(y)]$, where $h(y)$ can be the density function of any fixed distribution defined on the support set of y. Therefore, $L(q(y|x),\lambda) = \int_y f(q(y|x),y,\lambda)dy$. According to Euler-Lagrange equation, when the above Lagrangian achieve extreme point, we have:

$$\frac{\partial f}{\partial q} = K(y) + \alpha\log p(y|x) - \alpha\log q(y|x) - \lambda - \alpha = 0 \tag{11}$$

Thus, we have:

$$\begin{aligned} \alpha\log q(y|x) &= K(y) + \alpha\log p(y|x) - \log q(y|x) - \lambda - \alpha \\ q(y|x) &= \exp(\frac{1}{\alpha}K(y) + \log p(y|x) - \frac{\lambda}{\alpha} - 1) \\ &= \frac{1}{\exp(\frac{\lambda}{\alpha}+1)}\exp(\frac{1}{\alpha}K(y) + \log p(y|x)) \end{aligned} \tag{12}$$

Meanwhile, since $\int_y q(y|x)dy = 1$, we have:

$$\begin{aligned} \int_y \exp(\frac{1}{\alpha}K(y) + \log p(y|x) - \frac{\lambda}{\alpha} - 1)dy = 1 \\ \frac{1}{\exp(\frac{\lambda}{\alpha}+1)}\int_y \exp(\frac{1}{\alpha}K(y) + \log p(y|x))dy = 1 \end{aligned} \tag{13}$$

Thus, we have $\exp(\frac{\lambda}{\alpha}+1) = \int_y \exp(\frac{1}{\alpha}K(y) + \log p(y|x))dy = Z$, leading to:

$$q(y|x) = \frac{1}{Z}\exp(K(y) + \alpha\log p(y|x)), Z = \int \exp(K(y) + \alpha\log p(y|x))dy \tag{14}$$

Note that F represents fixed knowledge for a given domain so it remains constant during the optimization process and doesn't need to be considered as a variable in the proof. □

B.3 Proof of Physics-Informed Refinement Plugin Theorem 2.2

We now present a theoretical framework for applying physics-informed refinement to any time series prediction model, establishing its model-agnostic nature.

Assumption 1 (Regularity Conditions)**.** *We assume that* $K(\mathbf{x};F)$ *and* $\log p_M$ *are L-smooth (have Lipschitz gradients) and satisfy a mild dissipativity/coercivity condition such as* $\langle\mathbf{x},\nabla U(\mathbf{x})\rangle \geq m\|\mathbf{x}\|^2 - b$ *for* $U = -K - \alpha\log p_M$*, where* $m > 0$ *and* b *are constants.*

Assumption 2 (Score Approximation)**.** *For any model* M*, we assume the score function can be approximated such that* $\|\nabla\log\hat{p}_M - \nabla\log p_M\|_2 \leq \varepsilon_{score}$*.*

Theorem B.2 (Physics-Informed Inference Plugin Convergence)**.** *Let* $p_M(\mathbf{x}^{tar}|\mathbf{x}^{con})$ *be the conditional distribution defined by any time series model* M *(diffusion model, transformer, RNN, etc.). Let* F *represent a physical law with residual function* $K(\mathbf{x}^{tar};F) = -\|\frac{\partial\mathbf{x}^{tar}}{\partial t} - F(t,\mathbf{x}^{tar},\mathbf{u},\frac{\partial\mathbf{x}^{tar}}{\partial\mathbf{u}_i},\frac{\partial^2\mathbf{x}^{tar}}{\partial\mathbf{u}_i\partial\mathbf{u}_j},\dots)\|_2^2$*.*

Under Assumptions 1 and 2, for the physics-informed plugin with step size $\epsilon = \Theta(N^{-1/2})$ and N refinement steps:

$$\mathbf{x}_{j+1}^{tar} = \mathbf{x}_j^{tar} + \epsilon \nabla K(\mathbf{x}_j^{tar}; F) + \alpha\epsilon\nabla \log p_M(\mathbf{x}_j^{tar}|\mathbf{x}^{con}) + \sqrt{2\epsilon}\,\boldsymbol{\sigma}_j \tag{15}$$

where $\boldsymbol{\sigma}_j \sim \mathcal{N}(0, I)$, the resulting samples converge to the distribution $q^*(\mathbf{x}^{tar}|\mathbf{x}^{con})$ that minimizes:

$$\mathcal{L}(q) = -\mathbb{E}_{\mathbf{x}^{tar}\sim q}[K(\mathbf{x}^{tar}; F)] + \alpha D_{\mathrm{KL}}(q(\mathbf{x}^{tar}|\mathbf{x}^{con}) \,\|\, p_M(\mathbf{x}^{tar}|\mathbf{x}^{con})) \tag{16}$$

with a convergence rate of:

$$D_{\mathrm{KL}}(q_N \,\|\, q^*) \leq \mathcal{O}\left(\frac{d_{\mathit{eff}}}{\sqrt{N}} + \varepsilon_{\mathit{score}}^2\right) \tag{17}$$

where q_N is the distribution after N refinement steps and d_{eff} is the effective dimension of the state space after considering any dimensionality reduction techniques employed.

Proof. The proof follows standard Langevin dynamics convergence results [13], adapted to our setting:

Target distribution: The Boltzmann distribution representing the optimal balance between physics compliance and model fidelity, where Z is the normalization constant:

$$q^*(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}}) = \frac{1}{Z}\exp(K(\mathbf{x}^{\text{tar}}; F) + \alpha \log p_M(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})) \tag{18}$$

Langevin dynamics convergence: For the Langevin dynamics:

$$\mathbf{x}_{j+1}^{\text{tar}} = \mathbf{x}_j^{\text{tar}} + \epsilon\nabla logq^*(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}}) + \sqrt{2\epsilon}\,\boldsymbol{\sigma}_j \tag{19}$$

It is known from statistical physics that these dynamics sample from the distribution q^* in the limit of infinite steps. Specifically, the dynamics are simulating the stochastic differential equation (SDE), where W_t is a standard Wiener process:

$$d\mathbf{x}_t = \nabla \log q^*(\mathbf{x}_t|\mathbf{x}^{\text{con}})dt + \sqrt{2}\,dW_t \tag{20}$$

Convergence rate: Under the regularity conditions in Assumption 1 and score approximation in Assumption 2, for Langevin Monte Carlo (LMC) with step size ϵ, the convergence rate is:

$$D_{\mathrm{KL}}(q_N \,\|\, q^*) \leq \mathcal{O}\left(\frac{d_{\text{eff}}\epsilon}{N} + d_{\text{eff}}\epsilon^2 + \varepsilon_{\text{score}}^2\right) \tag{21}$$

Optimal rate: With $\epsilon = \Theta(N^{-1/2})$ [13]:

$$D_{\mathrm{KL}}(q_N \,\|\, q^*) \leq \mathcal{O}\left(\frac{d_{\text{eff}}}{\sqrt{N}} + \varepsilon_{\text{score}}^2\right) = \mathcal{O}(N^{-1/2}), \tag{22}$$

ignoring dimension-dependent factors for simplicity. The additional error term is bounded for model M and does not affect the $\mathcal{O}(N^{-1/2})$ convergence rate

Model-dependent approximation: For different model types, approximating $\nabla \log p_M(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$ will vary:

Lemma B.3 (Score Approximation for Different Model Architectures)**.** *For the following model architectures, the score function $\nabla \log p_M(\mathbf{x}^{tar}|\mathbf{x}^{con})$ can be approximated as follows, each with bounded error $\varepsilon_{\mathrm{score}}$ under appropriate conditions:*

1. ***Diffusion Models**: For a diffusion model with noise prediction network ϵ_θ,*

$$\nabla \log p_M(\mathbf{x}_t|\mathbf{x}^{con}) \approx -\frac{1}{\sigma_t^2}\epsilon_\theta(\mathbf{x}_t, t, \mathbf{x}^{con}) \tag{23}$$

with error bound $\varepsilon_{\mathrm{score}} = \mathcal{O}(\|\epsilon_\theta - \epsilon^*\|_2)$, where ϵ^* is the optimal score function.

2. ***Autoregressive Models**: For an autoregressive model with token probabilities $p(x_i|x_{<i}, \mathbf{x}^{con})$,*

$$\nabla \log p_M(\mathbf{x}|\mathbf{x}^{con}) \approx \sum_{i=1}^{T} \nabla_{x_i} \log p(x_i|x_{<i}, \mathbf{x}^{con}) \tag{24}$$

with error bound $\varepsilon_{\mathrm{score}} = \mathcal{O}(\frac{1}{T})$ when using finite differences for approximation.

3. ***Energy-Based Models**: For an energy-based model with energy function $E_\theta(\mathbf{x}, \mathbf{x}^{con})$,*

$$\nabla \log p_M(\mathbf{x}|\mathbf{x}^{con}) = -\nabla_{\mathbf{x}} E_\theta(\mathbf{x}, \mathbf{x}^{con}) \tag{25}$$

   with no approximation error when the energy function is differentiable.

4. ***Variational Autoencoders**: For a VAE with encoder $q_\phi(z|\mathbf{x})$ and decoder $p_\theta(\mathbf{x}|z)$,*

$$\nabla \log p_M(\mathbf{x}|\mathbf{x}^{con}) \approx \mathbb{E}_{z\sim q_\phi(z|\mathbf{x},\mathbf{x}^{con})}[\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}|z, \mathbf{x}^{con})] \tag{26}$$

   with error bound $\varepsilon_{\text{score}} = \mathcal{O}(D_{KL}(q_\phi(z|\mathbf{x}, \mathbf{x}^{con})||p(z|\mathbf{x}, \mathbf{x}^{con})))$.

□

Remark 1 (Non-Convexity). *When K or $-\log p_M$ is non-convex, Langevin dynamics can still converge with the stated rate, but constants in the bounds may depend exponentially on the energy barriers in the landscape, potentially affecting mixing times in practice.*

Remark 2 (Model-Agnostic Nature). *The convergence guarantees in Theorem 2.2 depend only on properties of the physics operator F, the number of refinement steps N, and the score approximation error ε_{score}. They are independent of the specific architecture, training procedure, or internal structure of the model M. This establishes that our physics-informed plugin approach is truly model-agnostic.*

Unlike previous approaches that incorporate physics constraints directly into specific model architectures (e.g., physics-informed neural networks or physics-constrained transformers), our method provides a model-agnostic refinement plugin that can be applied to any time series model that provides a way to approximate its score function. This fundamental shift in approach allows practitioners to leverage state-of-the-art advances in time series modeling while still ensuring physical consistency, without requiring specialized architecture modifications

B.4 Proof of Residual–Variance Coupling Lemma 2.3

Lemma 2.3 connects KL convergence to physical accuracy, showing that improvement in KL directly improves physics compliance. This addresses the key concern: "Does better convergence actually mean better physics?"

Lemma B.4 (Residual–Variance Coupling). *Let $\widetilde{r} = \partial_t\mathbf{x} - F(\cdot)$ be the physical residual of any sample $\mathbf{x} \sim q$. If F is L-Lipschitz and the surrogate bias is δ, then for all q absolutely continuous w.r.t. q^*,*

$$\mathrm{Var}_q[\widetilde{r}] \;\le\; 2L^2\, D_{\mathrm{KL}}(q \parallel q^*) + 4L^2\delta^2. \tag{27}$$

Implication: *every $\sqrt{N}$-step improvement in* KL *directly squeezes the variance of the physics residual.*

)

Proof. By definition, the variance of the residual under distribution q is:

$$\mathrm{Var}_q[\widetilde{r}] = \mathbb{E}_q[\|\widetilde{r} - \mathbb{E}_q[\widetilde{r}]\|^2] \le \mathbb{E}_q[\|\widetilde{r}\|^2]$$

Let $\widetilde{r}^* = \partial_t\mathbf{x} - F(\cdot)$ be the residual under the target distribution q^*. By the definition of q^*, we know that $\mathbb{E}_{q^*}[\|\widetilde{r}^*\|^2] \le \delta^2$, where δ represents the bias of the surrogate model.

Now we can decompose the expected squared residual:

$$\mathbb{E}_q[\|\widetilde{r}\|^2] = \int \|\widetilde{r}(\mathbf{x})\|^2 q(\mathbf{x}) d\mathbf{x} \tag{28}$$

$$= \int \|\widetilde{r}(\mathbf{x})\|^2 q^*(\mathbf{x}) \frac{q(\mathbf{x})}{q^*(\mathbf{x})} d\mathbf{x} \tag{29}$$

$$= \mathbb{E}_{q^*}\left[\|\widetilde{r}(\mathbf{x})\|^2 \frac{q(\mathbf{x})}{q^*(\mathbf{x})}\right] \tag{30}$$

Using Pinsker's inequality, we have:

$$\int \big|q(\mathbf{x}) - q^*(\mathbf{x})\big|\, d\mathbf{x} \le \sqrt{2D_{\mathrm{KL}}(q\|q^*)}$$

Since F is L-Lipschitz, we know that for any $\mathbf{x}$ and $\mathbf{y}$:

$$\|\widetilde{r}(\mathbf{x}) - \widetilde{r}(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|$$

Combining these results with the triangle inequality, we can bound the difference between the expected squared residuals:

$$\left|\mathbb{E}_q\big[\|\widetilde{r}\|^2\big] - \mathbb{E}_{q^*}\big[\|\widetilde{r}^*\|^2\big]\right| \leq 2L^2 \int\big|q(\mathbf{x}) - q^*(\mathbf{x})\big|\, d\mathbf{x} \tag{31}$$

$$\leq 2L^2\sqrt{2D_{\mathrm{KL}}(q\|q^*)} \tag{32}$$

Therefore:

$$\mathrm{Var}_q\big[\widetilde{r}\big] \leq \mathbb{E}_q\big[\|\widetilde{r}\|^2\big] \tag{33}$$

$$\leq \mathbb{E}_{q^*}\big[\|\widetilde{r}^*\|^2\big] + 2L^2\sqrt{2D_{\mathrm{KL}}(q\|q^*)} \tag{34}$$

$$\leq \delta^2 + 2L^2\sqrt{2D_{\mathrm{KL}}(q\|q^*)} \tag{35}$$

Using the inequality $\sqrt{x} \leq 1 + x/2$ for $x \geq 0$, we get:

$$\mathrm{Var}_q\big[\widetilde{r}\big] \leq \delta^2 + 2L^2\left(1 + \frac{2D_{\mathrm{KL}}(q\|q^*)}{2}\right) \tag{36}$$

$$= \delta^2 + 2L^2 + 2L^2 D_{\mathrm{KL}}(q\|q^*) \tag{37}$$

$$\leq 4L^2\delta^2 + 2L^2 D_{\mathrm{KL}}(q\|q^*) \tag{38}$$

where in the last step we used the fact that for small enough δ, we have $\delta^2 + 2L^2 \leq 4L^2\delta^2$.

Thus, we have established:

$$\mathrm{Var}_q\big[\widetilde{r}\big] \;\leq\; 2L^2\, D_{\mathrm{KL}}\big(q \,\|\, q^*\big) + 4L^2\delta^2$$

This shows that as the KL divergence between q and q^* decreases at rate $\mathcal{O}(N^{-1/2})$, the variance of the physical residual also decreases at the same rate, directly linking statistical convergence to physical accuracy. □

B.5 PDE Equations in Synthetic Simulators

We use the following equations to generate samples with 40 spatial resolutions and 192 timesteps for evaluating.

The **Diffusion-Sorption** equation can be expressed as:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} - k_s u \tag{39}$$

where u is the concentration, D is the diffusion coefficient, k_s is the sorption rate coefficient. Initial conditions are set as a Gaussian distribution:

$$u(x,0) = e^{-50(x-0.5)^2} \tag{40}$$

The boundary conditions are zero-flux (Neumann boundary conditions):

$$\frac{\partial u}{\partial x}\bigg| x = 0 = \frac{\partial u}{\partial x}\bigg| x = L = 0 \tag{41}$$

where $L = 1$ is the domain length.

The **Kolmogrov Flow** is a specific case of **Navier-Stokes** (NS) equation. More specifically, it is described by:

$$\mathbf{u}(x,y,z,t) = \left(-\frac{\partial\psi}{\partial y}, \frac{\partial\psi}{\partial x}, 0\right) \tag{42}$$

where the psi is the flow function. It is usually set as:

$$\psi(x,y,z,t) = A\sin(kx)\cos(zy + \omega t) \tag{43}$$

where A, k, w are hyperparameters.

The **Vorticity** equation is:

$$\frac{\partial\omega}{\partial t} + (\mathbf{u}\cdot\nabla)\omega = \nu\nabla^2\omega \tag{44}$$

where ω represents vorticity, $\mathbf{u}$ is the velocity field, and ν is the kinematic viscosity coefficient. This equation describes the evolution of vorticity in fluid flow, capturing the rotational motion central to turbulence formation.

The Computational Fluid Dynamics (**CFD**) equation implemented in the code can be expressed as:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \eta\frac{\partial^2 u}{\partial x^2} = \sin(\zeta x) \tag{45}$$

where:

η is the viscosity parameter (analogous to v in the Burgers equation). $F(x) = \sin(\zeta x)$ is the forcing term derived from Kolmogorov flow. Initial conditions are set as:

$$u(x,0) = \sin(x) + 0.5\sin(2x) + \epsilon(x) \tag{46}$$

where $\epsilon(x)$ is random noise sampled from normal distribution $\mathcal{N}(0, 0.25)$.

B.6 Physics Knowledge in the Climate Case

The Navier-Stokes equations describe the motion of viscous fluids and form the foundation of atmospheric dynamics. Therefore, incorporating Navier-Stokes constraints provides a physically-grounded approach to ensure that predictions of atmospheric variables (wind velocity, pressure, temperature) respect the underlying fluid dynamics.

For practical implementation, we adopt a quasi-geostrophic approximation commonly used in meteorological modeling for large-scale flows, rather than the full Navier-Stokes equations in spherical coordinates. Our physics-guided correction focuses on enforcing momentum conservation through the simplified momentum equation:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{f} \tag{47}$$

where:

- $\mathbf{u}$: velocity vector (wind speed)
- t: time
- ∇p: pressure gradient
- ρ: fluid density ($1.225\,\mathrm{kg/m^3}$ for standard atmosphere)
- ν: kinematic viscosity ($1.5 \times 10^{-5}\,\mathrm{m^2/s}$ for air)
- $\mathbf{f}$: external forces (e.g., Coriolis force)

The physical coefficients are derived from established climate physics knowledge. For instance, the density $\rho = 1.225\,\mathrm{kg/m^3}$ appears in the pressure-gradient term $-\left(\frac{1}{\rho}\right)\nabla p$ and momentum storage $\rho\mathbf{u}$, while the viscosity $\nu = 1.5 \times 10^{-5}\,\mathrm{m^2/s}$ multiplies the diffusive term $\nu\nabla^2\mathbf{u}$ that dampens small-scale shear.

To use this in our framework, we define a physical residual function $K(\overline{x}_{tar}; F)$ that penalizes deviations from this law. This is done by computing the left-hand side and right-hand side of the equation using the model's predicted time series data $\overline{x}_{tar}$ and aiming to minimize the difference between them. The physical constraints are incorporated by penalizing residuals through a physics-based energy term, which steers generated samples towards physically consistent solutions.

C Background on TimeDiT

TimeDiT [4] is a sophisticated foundation model designed specifically for multiple time series tasks, built upon a hybrid architecture that integrates a transformer with a diffusion model. This combination leverages the transformer's powerful ability to capture complex temporal dependencies across long sequences, while simultaneously utilizing the diffusion model's strength in probabilistic sampling to effectively model the inherent uncertainties often overlooked by deterministic autoregressive models. A core innovation of TimeDiT is its unified and comprehensive Time Series Mask Unit, which employs multiple masking strategies (random, block, stride, and reconstruction) to handle the practical challenges of real-world time series, such as missing values, irregular sampling, and multi-resolution data. This versatile masking system allows a single, pre-trained model to harmonize its learning process across diverse downstream tasks—from forecasting and imputation to anomaly detection—without requiring task-specific architectural changes. Architecturally, it uses an embedding layer that directly maps time series data into a continuous space without vector quantization to preserve data integrity, and it incorporates conditional information through adaptive layer normalization (AdaLN) for more effective temporal guidance. By demonstrating strong performance in zero-shot forecasting scenarios, TimeDiT establishes itself as a robust, general-purpose tool that successfully bridges the gap between broad foundation models and specialized, domain-specific approaches, providing accurate predictions with well-calibrated uncertainty quantification.

D Metrics

Anomaly correlation coefficient (ACC) The latitude weighted ACC for a forecast variable v at forecast time-step l is defined as follows:

$$\text{ACC}(v) = \frac{\sum_{m,n} L(m)\tilde{\mathbf{X}}_{\text{pred}}\tilde{\mathbf{X}}_{\text{true}}}{\sqrt{\sum_{m,n} L(m)\tilde{\mathbf{X}}^2_{\text{pred}} \sum_{m,n} L(m)\tilde{\mathbf{X}}^2_{\text{true}}}} \tag{48}$$

where $\tilde{\mathbf{X}}_{\text{pred/true}} = \mathbf{X}_{\text{pred/true}} - C$ represents the long-term-mean-subtracted value of predicted (/true) variable v. While $C = \frac{1}{N}\sum_t^N \mathbf{X}_{\text{true}}$ is the climatology mean of the history.

MAE describes the mean absolute error that measures the absolute difference between ground truth and prediction.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{49}$$

MSE describes the mean squared difference between ground truth and prediction.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{50}$$

RMSE is the sqaure root of MSE.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{51}$$

Computations of CRPS We explain the definition and calculation of the CRPS metric. The continuous ranked probability score (CRPS) assesses how well an estimated probability distribution F aligns with an observation x. It is defined as the integral of the quantile loss $\Lambda_\alpha(q, z) := (\alpha - \mathbf{1}_{z<q})(z - q)$ over all quantile levels $\alpha \in [0, 1]$:

$$\text{CRPS}(F^{-1}, x) = \int_0^1 2\Lambda_\alpha(F^{-1}(\alpha), x)\, d\alpha \tag{52}$$

where $\mathbf{1}$ represents the indicator function. We then calculated quantile losses for quantile levels discretized in 0.05 increments. Thus, we approximated CRPS as follows:

$$\text{CRPS}(F^{-1}, x) \approx \frac{1}{19}\sum_{i=1}^{19} 2\Lambda_{i\cdot 0.05}(F^{-1}(i \cdot 0.05), x). \tag{53}$$

Next, we computed the normalized average CRPS for all features and time steps:

$$\text{CRPS Score} = \frac{\sum_{k,l} \text{CRPS}(F^{-1}_{k,l}, x_{k,l})}{\sum_{k,l} |x_{k,l}|} \tag{54}$$

where k and l denote the features and time steps of the imputation targets, respectively. The lower the CRPS, the more accurate the model, i.e., the closer the predicted probability is to the observed outcome.

Computations of CRPS_sum CRPS_sum measures CRPS for the distribution F of the sum of all K features, calculated by:

$$\text{CRPS\_sum Score} = \frac{\sum_l \text{CRPS}(F^{-1}, \sum_k x_{k,l})}{\sum_{k,l} |x_{k,l}|} \tag{55}$$

where $\sum_k x_{k,l}$ is the total of the forecasting targets for all features at time point l.