CONSISTENCY TRAINING WITH PHYSICAL CONSTRAINTS

Che-Chia Chang[1], **Chen-Yang Dai**[2], **Te-Sheng Lin**[2,3], **Ming-Chih Lai**[2], **Chieh-Hsin Lai**[2]
[1]Institute of Artificial Intelligence Innovation, National Yang Ming Chiao Tung University, Taiwan
[2]Department of Applied Mathematics, National Yang Ming Chiao Tung University, Taiwan
[3]National Center for Theoretical Sciences, National Taiwan University, Taiwan

ABSTRACT

We propose a physics-aware Consistency Training (CT) (Song et al., 2023) method that accelerates sampling in Diffusion Models with physical constraints. Our approach leverages a two-stage strategy: (1) learning the noise-to-data mapping via CT, and (2) incorporating physics constraints as a regularizer. Experiments on toy examples show that our method generates samples in a single step while adhering to the imposed constraints. This approach has the potential to efficiently solve partial differential equations (PDEs) using deep generative modeling.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021b) have achieved significant success in high-dimensional data generation. Recent efforts have focused on adapting diffusion models to generate samples that satisfy physical constraints (Yuan et al., 2023; Mazé & Ahmed, 2023; Shu et al., 2023; Jacobsen et al., 2024; Bastek et al., 2024). In physics-informed diffusion models (PIDM) (Bastek et al., 2024), the authors combine physics-informed neural networks (PINNs) (Raissi et al., 2019) with diffusion models to sample data distributions while adhering to partial differential equation (PDE) constraints. However, PIDM still suffers from the inherent slow sampling issue of diffusion models. Inspired by PIDM and the recent development of Consistency Training (CT) (Song et al., 2023), we propose *CT-Physics*, a method that trains a consistency model with physical constraints from scratch. Unlike PIDM, which requires iterative denoising of samples, CT-Physics generates high-quality samples in one or two steps while ensuring the satisfaction of the system's physical constraints. CT-Physics presents a promising research direction, bridging deep generative models with efficient PDE solving.

2 BACKGROUND

2.1 PHYSICS-INFORMED DIFFUSION MODELS

Let $q(\boldsymbol{x}_0)$ be the target distribution, where samples $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$ satisfy the physical constraints $\boldsymbol{\mathcal{R}}(\mathbf{x}_0) = \mathbf{0}$, defined by a mapping $\boldsymbol{\mathcal{R}}$. Our goal is to train a generative model to approximate $q(\mathbf{x}_0)$. The model is trained by the denoising loss $\ell(\boldsymbol{\theta})$ and enforce constraints using the residual loss $\mathcal{R}(\boldsymbol{\theta})$ defined as:

$$\ell(\boldsymbol{\theta}) := \lambda_t \|\mathbf{x}_0 - \hat{\mathbf{x}}_0(\mathbf{x}_t, t; \boldsymbol{\theta})\|^2; \quad \mathcal{R}(\boldsymbol{\theta}) := \eta_t \|\boldsymbol{\mathcal{R}}(\mathbf{x}_0^{\text{DDIM}}(\mathbf{x}_t, t))\|^2,$$

where λ_t and η_t are weights, $\hat{\mathbf{x}}_0$ is the model estimate of the clean data, and $\mathbf{x}_0^{\text{DDIM}}$ is the DDIM estimate (Song et al., 2021a) with N steps. The total loss is:

$$\mathcal{L}_{\text{PIDM}}(\boldsymbol{\theta}) := \mathbb{E}_{t,\mathbf{x}_0,\mathbf{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\ell(\boldsymbol{\theta}) + \mathcal{R}(\boldsymbol{\theta})\right]. \tag{1}$$

2.2 CONSISTENCY MODELS

Consistency Training (CT) enables single-step generation by learning a mapping $\mathbf{f} : (\mathbf{x}_t, t) \mapsto \mathbf{x}_\epsilon$. The function satisfies the self-consistency property $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t'}, t'), \quad \forall t, t' \in [\epsilon, T],$

with boundary condition $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_\epsilon, \epsilon) = \mathbf{x}_\epsilon$. The parameterization ensures the boundary condition: $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}, t) = c_{\text{skip}}(t)\mathbf{x} + c_{\text{out}}(t)\mathbf{F}_{\boldsymbol{\theta}}(\mathbf{x})$, where $\mathbf{F}_{\boldsymbol{\theta}}$ is a neural network and $c_{\text{skip}}(t), c_{\text{out}}(t)$ are differentiable functions with $c_{\text{skip}}(0) = 1$, $c_{\text{out}}(0) = 0$. Self-consistency is enforced via the loss function

$$\mathcal{L}_{\text{CT}}(\boldsymbol{\theta}) := \mathbb{E}_{t_n,\mathbf{x}_0,\mathbf{z}}[\ell_{\text{CT}}(\boldsymbol{\theta})], \quad \ell_{\text{CT}}(\boldsymbol{\theta}) := \lambda(t_n) d\left(\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_0 + t_{n+1}\mathbf{z}, t_{n+1}), \mathbf{f}_{\text{sg}(\boldsymbol{\theta})}(\mathbf{x}_0 + t_n\mathbf{z}, t_n)\right),$$

where $\lambda(t_n)$ is a weight, d a distance function, and $\text{sg}(\cdot)$ the stop-gradient operation.

3 Method and Experiments

To achieve fast generation while maintaining high sample quality, we propose substituting the diffusion model's less accurate clean prediction, $\hat{\mathbf{x}}_0$, with the consistency model's one-step denoiser, $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$, within the PIDM framework. Consequently, the loss function $\ell(\boldsymbol{\theta})$ in Eq. 1 is replaced by the consistency training loss $\ell_{\text{CT}}(\boldsymbol{\theta})$. To ensure that the samples generated by the consistency model satisfy the physical constraints of the system, we introduce a new loss function given by:

$$\mathcal{R}_{\text{CT}}(\boldsymbol{\theta}) := \|\boldsymbol{\mathcal{R}}(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_T, T))\|^2.$$

Empirically, we found that defining the residual loss with prediction at time T leads to better enforcement of physical constraints. The final loss function is defined as:

$$\mathcal{L}_{\text{CT-physics}}(\boldsymbol{\theta}) := \mathbb{E}_{t_n,\mathbf{x}_0,\mathbf{z}}\left[\ell_{\text{CT}}(\boldsymbol{\theta}) + \mathcal{R}_{\text{CT}}(\boldsymbol{\theta})\right].$$

One might expect that replacing PIDM's clean estimation via the diffusion model with CT would work. However, directly training $\mathbf{f}_{\boldsymbol{\theta}}$ using $\mathcal{L}_{\text{CT-physics}}(\boldsymbol{\theta})$ from scratch leads to poor results. We hypothesize that the model overfits the physical constraints, leading to a failure in accurately capturing the original data distribution. See an example in A.2. To address this issue, we propose a two-stage training algorithm:

1. Stage 1 (Consistency Training): Train the consistency model only using the consistency loss $\mathcal{L}_{\text{CT}}(\boldsymbol{\theta})$. This stage acts as a warm-up phase, allowing the model to learn the global structure of the data distribution.
2. Stage 2 (Physics-informed Training): Train the consistency model using the loss function $\mathcal{L}_{\text{CT-physics}}(\boldsymbol{\theta})$, ensuring that the generated samples not only follow the data distribution but also satisfy the physical constraints.

We validate our method using toy examples, with results presented in Figure 1. Additional details and discussions are provided in the appendix.

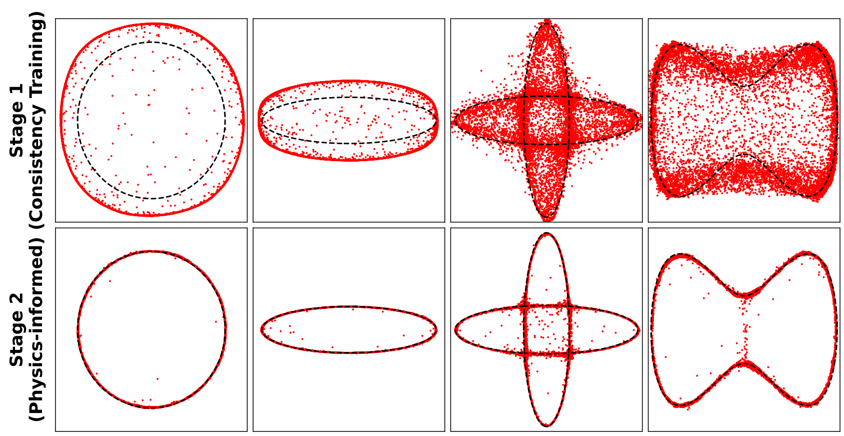

Figure 1: Results of CT-Physics on the toy examples. Red dots: model samples, black dashed line: $\boldsymbol{\mathcal{R}}(\mathbf{x}_0) = 0$.

4 Conclusion and Future Work

We proposed CT-Physics to train consistency models with physical constraints, enabling one-step sampling while ensuring physics constraints are satisfied. Future work includes integrating PDE constraints into training for data generation and efficient PDE solving.

REFERENCES

Jan-Hendrik Bastek, WaiChing Sun, and Dennis M. Kochmann. Physics-informed diffusion models, 2024. URL https://arxiv.org/abs/2403.14404.

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

Christian Jacobsen, Yilin Zhuang, and Karthik Duraisamy. Cocogen: Physically-consistent and conditioned score-based generative models for forward and inverse problems, 2024. URL https://arxiv.org/abs/2312.10527.

François Mazé and Faez Ahmed. Diffusion models beat gans on topology optimization. Proceedings of the AAAI Conference on Artificial Intelligence, 37(8):9108–9116, Jun. 2023. doi: 10.1609/aaai.v37i8.26093. URL https://ojs.aaai.org/index.php/AAAI/article/view/26093.

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. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. Journal of Computational Physics, 478:111972, 2023. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2023.111972. URL https://www.sciencedirect.com/science/article/pii/S0021999123000670.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, pp. 2256–2265. JMLR.org, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In International Conference on Learning Representations, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=WNzy9bRDvG.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3001ef257407d5a371a96dcd947c7d93-Paper.pdf.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In International Conference on Learning Representations, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In Proceedings of the 40th International Conference on Machine Learning, ICML'23. JMLR.org, 2023.

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model, 2023. URL https://arxiv.org/abs/2212.02500.

A APPENDIX

A.1 EXPERIMENTAL DETAILS

This section details the experimental setup. For all examples, we will adopt the training settings from improved consistency training (iCT) (Song & Dhariwal, 2024) except for the maximum number of discretization steps, which is chosen differently for each toy example. For all examples, we sample 10^4 data points from the target distributions $q(\mathbf{x}_0)$, each satisfying the constraints given by individual $\boldsymbol{\mathcal{R}}$.

Example 1: Unit Circle Let $\mathbf{x} = (x, y)$. This first example is given by the equation

$$\boldsymbol{\mathcal{R}}(\mathbf{x}) = x^2 + y^2 - 1 = 0.$$

We set the maximum number of discretization steps to 15. The neural network architecture is a 4-layer MLP with 128 hidden units and Sigmoid activation functions. The time step variable t is transformed into the Fourier feature and then concatenated with the input. We train stage 1 and stage 2 for 1000 epochs, each with a batch size of 128, using the Adam optimizer with a learning rate of 5×10^{-5}.

Example 2: Ellipse The second example is given by the equation

$$\boldsymbol{\mathcal{R}}(\mathbf{x}) = \frac{x^2}{2^2} + \frac{y^2}{0.5^2} - 1 = 0.$$

The training settings are the same as in the first example.

Example 3: Double Ellipse The third example is given by the equation

$$\boldsymbol{\mathcal{R}}(\mathbf{x}) = \left(\frac{x^2}{2^2} + \frac{y^2}{0.5^2} - 1\right)\left(\frac{x^2}{0.5^2} + \frac{y^2}{2^2} - 1\right) = 0.$$

We set the maximum number of discretization steps to 512. The neural network architecture is a 16-layer MLP with 128 hidden units and ReLU activation functions. The time step variable t is concatenated with the input after sinusoidal embedding. We train stage 1 for 20000 epochs using the RAdam optimizer with a batch size of 4096. The learning rate is set to 10^{-3}. We decay the learning rate by half every 1000 iterations to improve numerical stability. For stage 2, we train the model for 10000 epochs using the Adam optimizer with a learning rate of 5×10^{-5} with a batch size of 4096.

Example 4: Saddle Shape The fourth example is given by the equation

$$\boldsymbol{\mathcal{R}}(\mathbf{x}) = x^4 - 2x^2 + y^2 - \frac{1}{4} = 0.$$

We set the maximum number of discretization steps to 256. The neural network architecture is a 4-layer MLP with 128 hidden units and ReLU activation function, with the same time embedding method as in Example 3. We train stage 1 for 10000 epochs using the RAdam optimizer with a batch size of 512. The learning rate is set to 10^{-3}. We decay the learning rate by a factor of 0.9 every 1000 iterations for improved numerical stability. For stage 2, we train the model for 10000 epochs using the Adam optimizer with a learning rate of 5×10^{-5} with a batch size of 512.

A.2 STAGE 2 TRAINING WITHOUT STAGE 1 WARM-UP

We will use the unit circle example from A.1 to demonstrate the importance of the warm-up phase. Using the same training settings as in Example 1, we train the model directly from Stage 2 without the warm-up phase. The results are shown in Figure 2. The model fails to capture the original data distribution, indicating the importance of the warm-up phase in learning the global structure of the data distribution.

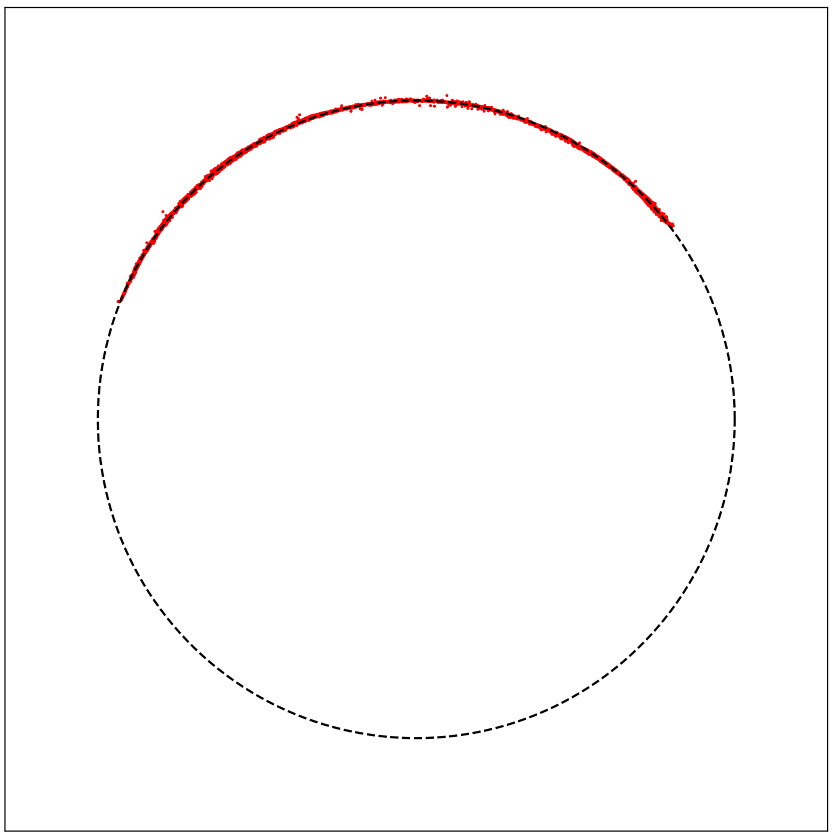

Figure 2: Sampling results of only using Stage 2 training. Red dots: model samples, black dashed line: unit circle. The model fails to capture the original data distribution.