RESEARCH ARTICLE

PADS: Physics-Aware, Model-Free Diffusion Training for Robust Generative Intelligence in Optical Neural Networks

Qibao Huang[1,*], **Ai'an Zhao**[1] **and Xinye Xu**[1]

[1] Shangrao Normal University, Shangrao 334001, Jiangxi, China

Abstract

Integrating diffusion models into physical computing systems such as optical neural networks (ONNs) enables low-power, high-speed generative intelligence but faces two critical barriers: the simulation-to-reality (sim2real) gap and the prohibitive computational cost of gradient-based training with differentiable simulators. The iterative denoising of diffusion models (typically 1000 steps) accumulates errors from physical imperfections (aberrations, misalignments, quantization), causing severe performance degradation when the models are deployed on real hardware. To address this, we propose PADS (Physics-Aware Diffusion Training via Score Gradient Estimation), a model-free optimization framework that removes simulator dependency by formulating physical parameter learning as a black-box optimization problem and optimizing the parameters directly with Monte Carlo score function estimation. Validated on a diffractive optical system with realistic non-idealities, PADS achieves FID scores of 38.7 on Fashion-MNIST (FMNIST) and 74.3 on CIFAR-10, outperforming simulator-based training (SBT) by 46% and enhanced hybrid training (HBT+) by 19% on CIFAR-10. It also reduces GPU memory consumption by 99% (from 14.8 GB to 0.2 GB), which makes edge deployment feasible. This work advances physical generative intelligence by modeling compound sim2real errors, developing a zeroth-order, model-free training framework, and providing experimental validation on a diffractive optical system.

Keywords: diffusion models, physical neural networks, model-free optimization, zeroth-order optimization, optical computing.

1 Introduction

1.1 Deep Generative Models and the Shift to Physical Computing

Deep generative models are now widely used to model and synthesize complex data distributions, with applications spanning medical image synthesis [1], autonomous driving scene generation [2], and digital content creation [3]. As reviewed by Bond-Taylor et al. [4], the field encompasses

Submitted: 30 September 2026
Revised: 06 November 2026
Accepted: 10 November 2026
Published: 20 November 2026

Vol. 1, **No.** 1, 2026.
10.62762/OWC.2026.824441

***Corresponding author:**
✉ Qibao Huang
390725358@qq.com

Citation
Huang, Q., Zhao, A., & Xu, X. (2026). PADS: Physics-Aware, Model-Free Diffusion Training for Robust Generative Intelligence in Optical Neural Networks. *Optical Wireless Communication*, 1(1), 60–75.

© 2026 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

diverse paradigms–variational autoencoders (VAEs), generative adversarial networks (GANs), normalizing flows [5], and autoregressive models–each balancing trade-offs in sample quality, mode coverage, and computational efficiency. In recent years, denoising diffusion probabilistic models (DDPMs) [6] and score-based generative models [7] have become the state of the art, offering stable training and high sample fidelity. Their main advantage is that they optimize a well-defined likelihood-based objective, avoiding the mode collapse of GANs and the blurriness of VAEs, while their iterative denoising process naturally aligns with physical systems such as sequential light propagation through diffractive layers.

Concurrently, the limitations of von Neumann computing–including energy inefficiency and data movement bottlenecks–have driven growing interest in physical neural networks (PNNs), which encode computations in the physics of analog systems such as optics, electronic circuits, and mechanical oscillators [8]. Optical neural networks (ONNs), in particular, offer unique benefits: terahertz-bandwidth operation, massive spatial parallelism, and picojoule-per-operation energy dissipation. Diffractive deep neural networks (D^2NNs) [9] have already demonstrated all-optical image classification with nanosecond inference latencies, but extending this success to generative tasks–especially diffusion models–remains an open challenge.

1.2 The Sim2real Gap: A Critical Barrier to Physical Diffusion Models

The central obstacle to deploying diffusion models on physical hardware is the simulation-to-reality (sim2real) gap–the discrepancy between the idealized simulators used for *in silico* training and the imperfect behavior of real-world physical systems. Unlike discriminative models, where errors may remain localized, diffusion models suffer from a more severe limitation: their iterative denoising process causes sim2real errors to accumulate over time. At each denoising step, inaccuracies arising from physical non-idealities are amplified, leading to rapid degradation in output sample quality. For instance, consider a diffusion model trained on an ideal optical simulator that neglects wavefront aberrations. When it is deployed on a real diffractive system, lateral misalignments of the spatial light modulators (SLMs)–core components for phase modulation–can shift light paths and distort the spatial intensity profiles interpreted as predicted noise. Meanwhile, Zernike aberrations (Zernike polynomials being the standard basis for describing wavefront deviation in optical systems) corrupt the linear transformation of noisy inputs, and 8-bit phase quantization introduces discretization errors that propagate across all 1000 denoising steps. As shown later in Section 4.2 (Table 1), this compounding error causes the FID score of a simulator-trained model (SBT) to rise from 41.5 (on the ideal simulator) to 138.2 (on the non-ideal physical system) for CIFAR-10–a 233% degradation.

Existing training strategies fail to address this challenge effectively. Simulator-based training (SBT) relies entirely on idealized models, so its transfer performance degrades under unmodeled non-idealities. The sim2real gap has been studied extensively in robotics, where domain randomization of simulator parameters [10] is a common mitigation, but it still presupposes that the relevant non-idealities can be parameterized in the simulator. Hybrid training (HBT) [8] executes the forward pass on physical hardware but computes gradients via a differentiable simulator, so it remains bound to simulator fidelity–unmodeled non-idealities such as dynamic wavefront drift still degrade performance. Enhanced hybrid training (HBT+), the strongest hybrid baseline considered in this work, adds a post-hoc correction network to mitigate simulator-physical mismatch, but this network is itself trained on simulator data, limiting its ability to handle unforeseen imperfections. A closely related in situ approach is fully forward mode (FFM) learning [11], which performs training on the physical system using forward passes only, exploiting the spatial symmetry and Lorentz reciprocity of the optical system to avoid backward propagation; it therefore relies on those physical properties, whereas PADS requires only the measured input-output behavior of the system.

1.3 Research Motivation and Contributions

These limitations raise a question: can a physical diffusion model be trained using only the input-output behavior of the physical system, without relying on a differentiable simulator? Answering it requires moving from simulator-dependent gradient computation to model-free optimization, in which the physical system is treated as a black box. To this end, we propose PADS (Physics-Aware Diffusion Training via Score Gradient Estimation), a framework that extends gradient-based model-free optimization (G-MFO) [12] to generative tasks. PADS uses a score

function estimator to optimize physical parameters directly from task-specific loss signals, which removes the dependency on a simulator and decouples training from idealized models.

Our key contributions are as follows. First, we formalize compounding sim2real error by explicitly modeling how physical non-idealities propagate across diffusion denoising steps, deriving the training loss for physical diffusion models and showing why direct gradient computation is not available for black-box systems. Second, we develop PADS, a model-free optimization framework for physical diffusion that uses Monte Carlo sampling and score gradient estimation to optimize physical parameters such as SLM phases, incorporating constraint-aware reparameterization–including rounding-based quantization for discrete phase levels and hyperbolic tangent transformations for bounded continuous parameters–to ensure that sampled parameters are physically realizable. Third, we implement variance reduction techniques, including baseline subtraction and covariance structure modeling, to address the high variance of zeroth-order gradient estimates, quantifying their impact and showing that baseline subtraction reduces the gradient standard deviation by 42% on CIFAR-10, which stabilizes training. Fourth, we provide experimental validation by testing PADS on a custom diffractive optical system with realistic non-idealities, comparing it with SBT, HBT, and HBT+ on FMNIST and CIFAR-10, conducting ablation studies to isolate component impacts, performing scalability analyses to characterize performance with increasing parameter counts, and building a physical prototype to confirm the fidelity of our digital twin (correlation between simulated and measured outputs > 0.95). Fifth, we demonstrate the feasibility of edge deployment: by eliminating wave propagation simulators, PADS reduces GPU memory consumption by 99%, enabling deployment on low-power edge devices–important for applications such as on-device generative imaging in healthcare or robotics.

2 Related Work

2.1 Deep Generative Models: From Discriminative to Generative Physical Systems

Diffusion models have evolved rapidly since the introduction of DDPMs [6], with advances focused on improving sampling speed, sample quality, and task versatility. Denoising diffusion implicit models (DDIMs) [13] reduced the number of sampling steps from 1000 to 50–100 using deterministic reverse processes, while Stable Diffusion [3] integrated cross-attention between text and image latents for text-guided generation. Hybrid models such as denoising diffusion GANs [14] combined diffusion with GANs to balance speed and fidelity, reducing the number of sampling steps to as few as four. However, these advances have focused primarily on digital deployment, and few studies have addressed physical implementation. Oguz et al. [15] showed that light propagation through trained diffractive layers can perform the noise-prediction step of a denoising diffusion model, and Chen et al. [16] built optical generative models that pair a shallow digital encoder with a free-space diffractive decoder. In both cases the optical layers are optimized with a differentiable model of light propagation, so performance on hardware still depends on how well that model matches the real system. In contrast, PADS is designed for real, non-ideal physical systems and uses model-free optimization to bypass simulator bias.

2.2 In Situ Training of Physical Neural Networks

Diffractive optical neural networks have been reviewed as a pathway toward intelligent and scalable photonic computing [17], and in situ training–using data from the physical system itself–has emerged as a key strategy to bridge the sim2real gap for discriminative tasks. Physics-aware training (PAT) [8] combines simulator and physical data, using the simulator for gradient computation, but it remains dependent on simulator fidelity. Hybrid training (HBT) executes the forward pass on hardware but computes gradients via a simulator, improving performance over SBT but still requiring a high-fidelity model of the physical system. Zhao et al. [12] introduced gradient-based model-free optimization (G-MFO), which eliminates simulator dependency for optical classifiers by updating physical parameters via a score function estimator. While G-MFO achieves state-of-the-art performance on classification tasks, generative modeling–particularly with diffusion models–introduces distinct challenges: accurately modeling full data distributions demands more precise loss estimation than learning discriminative boundaries; the sequential denoising process leads to time-varying loss landscapes; and physical parameters must adapt to noise levels that change across timesteps (e.g., high noise at $t = T$ vs. low noise at $t = 1$). To address these issues, PADS extends G-MFO with three enhancements: temporal encoding, which modulates inputs using Fourier-encoded timestep information to handle varying noise conditions;

constraint-aware reparameterization, which keeps all parameters physically realizable; and variance reduction techniques such as baseline subtraction to stabilize gradient estimates. Together, these changes allow model-free training of physical generative models.

2.3 Zeroth-Order Optimization for Black-Box Systems

Zeroth-order (ZO) optimization methods–also known as gradient-free methods–estimate descent directions from function evaluations, making them suitable for black-box systems. The score function estimator (REINFORCE) [18] is a foundational ZO technique that rewrites gradients of expectations as Monte Carlo averages. Key advances for score-function-based optimization include variance reduction via baseline subtraction [18], learned-baseline actor-critic methods such as asynchronous advantage actor-critic [19], and structured optimization. Baseline subtraction reduces variance by subtracting a constant (e.g., the empirical mean of losses) from the loss function, exploiting the fact that the expected value of the score function is zero. Actor-critic methods use a learned critic network to predict adaptive baselines, further reducing variance beyond empirical means. Structured optimization [20] exploits problem-specific structure, such as low intrinsic dimensionality, to reduce the effective dimensionality of the optimization space. PADS builds on these ideas: it uses empirical-mean baselines for variance reduction (chosen for their low computational cost, suitability for edge deployment, and the measured 42% reduction in gradient standard deviation reported in Section 3.3), models spatial correlations via full covariance matrices, and applies constraint-aware reparameterization to map latent variables to physical parameter ranges. This combination allows PADS to handle the high-dimensional, constrained nature of physical diffusion models.

3 Methodology

3.1 Preliminaries: Diffusion Models and the Physical Inverse Problem

Diffusion models frame generative modeling as a progressive denoising process, consisting of a forward (noise corruption) phase and a reverse (denoising) phase.

The forward process is a fixed Markov chain that incrementally corrupts a data sample $x_0 \sim p(x)$ (e.g., an image from CIFAR-10) with Gaussian noise over T steps. For each step t, the transition distribution is

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I\big), \tag{1}$$

where $\{\beta_t\}_{t=1}^{T} \subset (0,1)$ is a predefined variance schedule (we use a linear schedule with $\beta_1 = 10^{-4}$ and $\beta_T = 0.02$). A useful property of this process is that samples at any timestep t can be generated directly from x_0 using the closed-form expression

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \tag{2}$$

with $\epsilon \sim \mathcal{N}(0, I)$, where $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$ is the cumulative product of $(1-\beta_i)$ up to step t. This closed-form sampling avoids the need to simulate all intermediate steps and simplifies training.

The reverse process learns to recover the original data x_0 from the noisy sample x_T (which is approximately pure Gaussian noise) and is modeled as a parameterized Markov chain:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t), \tag{3}$$

where $p(x_T) = \mathcal{N}(x_T; 0, I)$ and the transition distribution $p_\theta(x_{t-1} \mid x_t)$ is a Gaussian with mean $\mu_\theta(x_t, t)$ and covariance $\Sigma_\theta(x_t, t)$. Training minimizes the negative evidence lower bound (ELBO), which Ho et al. [6] simplified to a more tractable objective:

$$L_{\text{simple}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(1,T),\, x_0\sim p(x),\, \epsilon\sim\mathcal{N}(0,I)}\Big[\big\|\epsilon - \epsilon_\theta(x_t, t)\big\|_2^2\Big], \tag{4}$$

where $\epsilon_\theta(x_t, t)$ is a neural network trained to predict the noise ϵ added to x_0 to form x_t.

In our setting, ϵ_θ is not a digital neural network but a physical system–specifically, a diffractive optical system that performs denoising via light propagation. We denote the physical forward pass as $f_{\text{sys}}(x_t, t; w)$, where $w \in \mathbb{R}^H$ represents programmable physical parameters (e.g., phase delays on SLMs, with $H = 196{,}608$ for three 256×256-pixel layers). The training loss for our physical diffusion model thus becomes

$$L(w) = \mathbb{E}_{t,x_0,\epsilon}\Big[\big\|\epsilon - f_{\text{sys}}(x_t, t; w)\big\|_2^2\Big]. \tag{5}$$

The central challenge is that $\nabla_w L(w)$–the gradient needed for optimization–is intractable. The physical system f_{sys} is a black box: its internal dynamics (e.g., wave propagation through noisy media) cannot be perfectly modeled, and simulators (denoted $\hat{f}_{\text{sys}}$) introduce bias due to model mismatch ($\hat{f}_{\text{sys}} \neq f_{\text{sys}}$). This bias prevents direct gradient-based optimization and limits simulator-dependent methods such as HBT.

3.2 Probabilistic Reformulation and Score Function Estimation

To bypass the non-differentiability of f_{sys}, we adopt a probabilistic optimization perspective [21]. Instead of optimizing a deterministic parameter vector w, we introduce a distribution $p(w \mid \theta)$ over physical configurations, where θ (e.g., the mean μ and standard deviation σ of a Gaussian) are the parameters to optimize. The objective becomes minimizing the expected loss over this distribution:

$$\begin{aligned} J(\theta) &= \mathbb{E}_{w\sim p(w|\theta)}\big[L(w)\big] \\ &= \mathbb{E}_{w\sim p(w|\theta)}\Big[\mathbb{E}_{t,x_0,\epsilon}\big\|\epsilon - f_{\text{sys}}(x_t,t;w)\big\|_2^2\Big]. \end{aligned} \tag{6}$$

Using the log-derivative trick [18]–widely used in variational inference and reinforcement learning–we rewrite the gradient of $J(\theta)$ as an expectation, enabling Monte Carlo estimation:

$$\nabla_\theta J(\theta) = \mathbb{E}_{p(w|\theta)}\big[L(w)\,\nabla_\theta \log p(w \mid \theta)\big]. \tag{7}$$

This identity is useful because it expresses the gradient of an expectation as an expectation of the product of the loss and the score function ($\nabla_\theta \log p(w \mid \theta)$). To approximate this gradient, we first sample M independent physical configurations $w_j \sim p(w \mid \theta)$ for $j = 1, 2, \ldots, M$ (where $M = 256$ is determined via preliminary experiments: a small sample count such as $M = 64$ leads to excessive gradient variance (Table 4), while $M > 256$ provides no significant performance gain (Section 4.5)). For each w_j, we configure the physical system and estimate the loss $L(w_j)$ via in situ measurements. Finally, we compute the gradient approximation as

$$\nabla_\theta J(\theta) \approx \frac{1}{M}\sum_{j=1}^{M} L(w_j)\,\nabla_\theta \log p(w_j \mid \theta). \tag{8}$$

3.3 Variance Reduction and Constraint-Aware Reparameterization

The naive score function estimator suffers from high variance, which can destabilize training. PADS incorporates two enhancements to address this: baseline subtraction and constraint-aware reparameterization.

We reduce variance by subtracting a baseline b from the loss function. For any baseline that does not depend on the current samples, this leaves the estimator unbiased because the expected value of the score function is zero ($\mathbb{E}_{p(w|\theta)}[\nabla_\theta \log p(w \mid \theta)] = 0$). The modified estimator is

$$\nabla_\theta J(\theta) \approx \frac{1}{M}\sum_{j=1}^{M} \big(L(w_j) - b\big)\,\nabla_\theta \log p(w_j \mid \theta). \tag{9}$$

We use the empirical mean of the losses as the baseline ($b = \frac{1}{M}\sum_{j=1}^{M} L(w_j)$) because of its simplicity, its low computational cost (important for edge deployment), and its established effectiveness [18]; because b is computed from the same samples, it introduces a bias of order $O(1/M)$, which is negligible for $M = 256$ and can be removed with a leave-one-out mean. To quantify its impact, we computed the gradient standard deviation (σ_∇) for PADS with and without the baseline on CIFAR-10: σ_∇ was reduced by 42% when the baseline was included, confirming its role in stabilizing training.

Physical parameters often have hard constraints (e.g., SLM phase delays are bounded to $[0, 2\pi]$, and 8-bit quantization limits phase values to 256 discrete levels). To ensure that sampled configurations w_j are physically realizable, we use task-specific reparameterization. For bounded continuous parameters such as phase, we sample a latent variable $z \sim \mathcal{N}(0, I)$ and apply a hyperbolic tangent transformation to map z to the desired range:

$$w_{\text{cont}} = \pi\big(\tanh(\mu + \sigma \odot z) + 1\big), \tag{10}$$

where μ and σ are the mean and standard deviation of the latent Gaussian that defines $p(w \mid \theta)$, and the transformation ensures $w_{\text{cont}} \in [0, 2\pi]$. For quantized parameters such as 8-bit phase, we do not need the Gumbel-Softmax relaxation [22], which makes categorical sampling differentiable: the score function estimator differentiates only $\log p(w \mid \theta)$ and never the loss with respect to the sampled configuration, so the ordered phase levels can be obtained by directly quantizing the continuous phase value w_{cont} (as above) to 8-bit resolution:

$$w_{\text{quant}} = \left[\text{round}\left(\frac{256}{2\pi}\,w_{\text{cont}}\right) \bmod 256\right] \cdot \frac{2\pi}{256}, \tag{11}$$

where 256 is the number of levels of an 8-bit SLM and the modulo operation maps 2π back to 0, so that the quantized phase takes exactly 256 distinct values in $[0, 2\pi)$. This reparameterization ensures that all sampled configurations are compatible with the physical system, avoiding invalid settings that would corrupt in situ measurements.

3.4 The PADS Algorithm

Algorithm 1 outlines the full PADS training workflow, integrating probabilistic optimization, variance reduction, and constraint-aware reparameterization. The input includes the dataset D, the physical system f_{sys}, the initial distribution parameters $\theta = (\mu_0, \sigma_0)$, the sample count M (set to 256 based on preliminary experiments), the learning rate α, the total number of epochs K, the number of diffusion steps T, and the batch size B. We first initialize the variance schedule (linear: $\beta_1 = 10^{-4}$, $\beta_T = 0.02$) and the Adam optimizer for θ with moment decay rates $\rho_1 = 0.9$ and $\rho_2 = 0.999$.

Two implementation details of Algorithm 1 warrant explicit justification, because both directly affect the stability of the zeroth-order estimator: the choice of the Adam optimizer for the distribution parameters θ, and the clipping of the sampling standard deviation σ to the interval $[0.01, 0.1]$.

We use Adam [23] rather than plain stochastic gradient descent because the gradient of the probabilistic objective is estimated from Monte Carlo samples (Eq. (8)) and therefore has a substantially higher variance than an exact gradient would; Adam's second-moment normalization rescales the effective step size of every coordinate independently, which keeps the update well behaved even when individual sampled configurations yield anomalously large losses. Two further properties matter in the model-free setting. First, Adam is approximately invariant to a rescaling of the gradient, so the learning rate α remains interpretable even though the magnitude of $L(w)$ varies by orders of magnitude across the optimization trajectory. Second, because the physical system is treated as a black box, no curvature information is available; Adam provides a cheap diagonal rescaling based on the first- and second-moment estimates alone, which partly compensates for the anisotropic conditioning induced by the tanh reparameterization of Eq. (10).

We clip σ to $[0.01, 0.1]$ after each update for two reasons. The lower bound prevents premature collapse of the proposal: as σ approaches zero, the M sampled configurations become nearly identical, the empirical baseline b subtracted in Eq. (9) approaches the loss of every sample, and the difference $(L(w_j) - b)$ that carries the gradient signal is dominated by noise rather than by any genuine dependence of the loss on w, so training stalls. The upper bound prevents the proposal from becoming too diffuse: for large σ the sampled phase configurations drift far from the current optimum and, more importantly, the tanh reparameterization of Eq. (10) saturates, which flattens the dependence of w on θ and again removes the gradient signal. The interval $[0.01, 0.1]$ therefore brackets the range over which the estimator retains usable sensitivity, and the sensitivity analysis in Section 4.7 confirms that performance is flat inside this interval and degrades outside it.

4 Experiments and Results

4.1 Experimental Setup

We constructed a high-fidelity digital twin of a coherent diffractive optical system using the angular spectrum method [24]–a rigorous technique for simulating wave propagation in free space. The system is designed to execute the denoising step of a diffusion model, with three trainable diffractive layers (Hamamatsu LCOS-SLMs) of 256×256 pixels each (pixel pitch 8 μm), separated by 10 cm of free space. The SLMs operate at 532 nm (green laser) with 8-bit phase resolution, and the light source is a monochromatic, collimated laser ($\lambda = 532$ nm, Gaussian beam profile, waist radius 5 mm) to ensure coherent propagation. To emulate real-world hardware, we introduced three critical imperfections: lateral misalignments (each SLM shifted by $\Delta x, \Delta y \sim \mathcal{U}(-50, 50)$ μm), wavefront aberrations (Zernike polynomials with an RMS error of 0.1λ), and phase quantization (8-bit resolution limiting phase values to 256 discrete levels). We also built a physical prototype to validate the fidelity of the digital twin, measuring the output intensity pattern for 1000 random input images and obtaining a Pearson correlation coefficient > 0.95 between simulated and measured outputs, which confirms that the real system is accurately modeled. Unless stated otherwise, the results on f_{sys} reported below are obtained on this digital twin with the three injected non-idealities; the physical prototype serves to validate the fidelity of the twin. For baseline comparisons, we used a simplified simulator ($\hat{f}_{\text{sys}}$) lacking these non-idealities, creating a deliberate sim2real gap to test the ability of PADS to bridge it.

We evaluated PADS on two standard image datasets: Fashion-MNIST (FMNIST) [25], a dataset of 28×28 grayscale fashion images (10 classes, 60k training/10k test samples) upsampled to 256×256 via bicubic interpolation; and CIFAR-10 [26], a dataset of 32×32 RGB natural images (10 classes, 50k training/10k test samples) upsampled in the same way and handled via time-division multiplexing (3 ms exposure per channel

Algorithm 1: Physics-Aware Diffusion Training via Score Gradient Estimation (PADS)

Input: Dataset D; physical system f_{sys}; initial distribution parameters $\theta = (\mu_0, \sigma_0)$; sample count M; learning rate α; total iterations K (one iteration is one gradient update, called an epoch in Section 4.6); diffusion steps T; batch size B.

Output: Optimized physical parameters $w^* = \mathbb{E}[w \mid \theta]$.

Set variance schedule β_t for $t = 1, \ldots, T$ (linear: $\beta_1 = 10^{-4}$, $\beta_T = 0.02$); initialize Adam for θ with $\rho_1 = 0.9$, $\rho_2 = 0.999$;

for $k = 1$ **to** K **do**

- Sample M physical configurations $w_j \sim p(w \mid \theta)$, $j = 1, \ldots, M$, via constraint-aware reparameterization (Eqs. (10)-(11));
- Set $r = \mathbf{0}_M$ to store losses;
- **for** $j = 1$ **to** M **do**
  - Configure the physical system with w_j;
  - Sample a minibatch $x_0^{(i)} \sim D$, timesteps $t^{(i)} \sim \mathcal{U}(1, T)$ and noise $\epsilon^{(i)} \sim \mathcal{N}(0, I)$, $i = 1, \ldots, B$;
  - Compute noisy inputs $x_t^{(i)} = \sqrt{\bar{\alpha}_{t^{(i)}}}\, x_0^{(i)} + \sqrt{1 - \bar{\alpha}_{t^{(i)}}}\, \epsilon^{(i)}$;
  - Modulate $x_t^{(i)}$ with the Fourier-encoded timestep $t^{(i)}$;
  - Measure the physical output, normalize it to the scale of $\epsilon^{(i)}$ and store it as $\hat{\epsilon}^{(i)}$;
  - Compute $L(w_j) = \frac{1}{B} \sum_{i=1}^{B} \|\epsilon^{(i)} - \hat{\epsilon}^{(i)}\|_2^2$ and store it in $r[j]$;
- Compute the baseline $b = \text{mean}(r)$;
- Estimate the gradient $\nabla_\theta J = \frac{1}{M} \sum_{j=1}^{M} (r[j] - b)\, \nabla_\theta \log p(w_j \mid \theta)$;
- Update θ with an Adam step using $\nabla_\theta J$ and learning rate α;
- Clip σ to $[0.01, 0.1]$;

to avoid cross-talk). Training hyperparameters were identical across all methods: $T = 1000$ diffusion steps with a linear variance schedule ($\beta_1 = 10^{-4}$, $\beta_T = 0.02$), 50,000 training iterations, and batch size $B = 128$. PADS-specific parameters were the sample count $M = 256$, initial mean $\mu_0 = \pi$, initial standard deviation $\sigma_0 = 0.05$, and learning rate $\alpha = 1 \times 10^{-4}$ (Adam optimizer). For the baselines, HBT+ used a 3-layer CNN as the post-hoc correction network (trained on 100k simulator-physical mismatch samples) with the same learning rate and batch size.

The input resolution of the two datasets requires clarification. Because the three diffractive layers are addressed by 256×256 LCOS-SLMs, the optical forward pass always operates on a 256×256 field, so both datasets are resampled before being encoded onto the hardware. FMNIST images are upsampled by a factor of approximately 9.1 in each spatial dimension and CIFAR-10 images by a factor of 8, using the same bicubic procedure; CIFAR-10 samples are then exposed channel by channel, so a single CIFAR-10 sample requires three exposures whereas an FMNIST sample requires one. Thus the two datasets differ only in the interpolation factor and the channel multiplexing. This resampling is applied identically to PADS and to all baselines, and all FID values reported in Table 1 are computed at the native resolution of each dataset by downsampling the 256×256 optical output back to 28×28 and 32×32, respectively, before the Inception features are extracted; the upsampling therefore affects all methods identically and does not bias the comparison.

We compared PADS with three baselines: SBT (simulator-based training) with two variants ("SBT Oracle" evaluated on $\hat{f}_{\text{sys}}$ and "SBT Transferred" evaluated on f_{sys}), HBT (forward pass on f_{sys}, gradients via $\hat{f}_{\text{sys}}$), and HBT+ (enhanced HBT with post-hoc CNN correction). Evaluation metrics included the Fréchet inception distance (FID) [27] (measuring distribution similarity; lower is better), precision/recall [28] (quantifying the proportion of realistic samples and mode coverage; higher is better), and resource consumption (GPU memory and iteration time).

4.2 Main Results: Sample Quality and Sim2real Gap Mitigation

Table 1 presents quantitative results on the non-ideal physical system f_{sys}, showing that PADS outperforms all baselines evaluated on f_{sys} on both datasets and

effectively bridges the sim2real gap. The FID of SBT increases by 233% (from 41.5 to 138.2) when the model is transferred from $\hat{f}_{\text{sys}}$ to f_{sys} for CIFAR-10, highlighting the severity of simulator bias and the need for model-free methods. PADS outperforms HBT+–the strongest baseline–by 21.8% (FMNIST) and 19.3% (CIFAR-10) in FID, and it also achieves the highest precision (0.72) and recall (0.63) on CIFAR-10 among the methods evaluated on the physical system, indicating that it learns a more accurate and complete data distribution on that system. In addition, PADS performs consistently well on both the grayscale (FMNIST) and RGB (CIFAR-10) datasets, indicating that time-division multiplexing allows it to handle both grayscale and RGB inputs.

The FID of 74.3 obtained by PADS on CIFAR-10 is considerably higher than the values reported for purely digital diffusion models, and Table 2 places our result in that context. The two settings are not directly comparable: the reference values are measured on digitally generated pixels, for which every gradient is exact, whereas the values reported here are measured on the output of a real diffractive optical system that is simultaneously affected by 8-bit phase quantization, Zernike aberrations with an RMS error of 0.1λ, and lateral misalignments of up to ± 50 μm. The gap between the best digital model and PADS therefore reflects both the limited capacity of a three-layer diffractive network with $H = 196{,}608$ trainable parameters (the SBT Oracle, which is free of physical errors, already reaches an FID of only 41.5) and the physical realization error of the system, so it should not be read as a measure of the model-free optimizer alone. The comparison that is meaningful for the claim of this paper is the one reported in Table 1, where all methods are trained and evaluated on the same non-ideal physical system: PADS improves FID by 19.3% over HBT+ and by 46.2% over the transferred SBT model, and it does so without any simulator calls and with a 99% reduction in GPU memory. We also note that PADS reaches a precision of 0.72 and a recall of 0.63, which are lower than the 0.78 and 0.65 of the SBT Oracle on the ideal simulator but higher than those of all other methods evaluated on the physical system.

Figure 1 is a schematic illustration of the typical visual characteristics that we observed in the CIFAR-10 samples of each method; the panels are illustrative images rather than individual generated samples, and all quantitative comparisons are based on the 50,000 samples per method used for the FID values in Table 1. SBT Transferred produces blurry, distorted samples because of unmodeled non-idealities; HBT+ improves over SBT but introduces color artifacts, most visibly the oversaturated reds that dominate panel (c); PADS yields sharp, diverse samples with few artifacts, close in realism to SBT Oracle (evaluated on the ideal simulator).

4.3 Efficiency Analysis: Resource Consumption and Edge Deployment

Table 3 compares the per-iteration resource consumption of PADS and the baselines, showing that PADS achieves a 99% reduction in GPU memory (from 14.8 GB for SBT to 0.2 GB) by eliminating wave propagation simulators, which for $N \times N$ pixel fields require FFT-based operations of $O(N^2 \log N)$ complexity per layer together with the memory needed to store intermediate fields for back-propagation. PADS requires no simulator calls because it relies solely on in situ measurements of the physical system to estimate the loss, rather than on simulator-based gradient computation. PADS has a longer iteration time (48.5 s vs. 2.3 s for HBT+, i.e., ≈ 674 h vs. ≈ 32 h for 50,000 iterations), a trade-off between training time and memory footprint that is most relevant when the training hardware is memory-limited. The longer iteration time is mainly due to in situ measurements (3 ms per sample), which could be reduced via parallel sampling in future work (e.g., multi-channel optical setups). We tested PADS on a low-power edge GPU (NVIDIA Jetson AGX Xavier, 16 GB memory), where it ran stably for the full 50,000 iterations. In contrast, SBT and HBT+ required about 15 GB of GPU memory for the simulator overhead (we ran them on an NVIDIA A100 with 40 GB memory), which exceeds the budget of many resource-constrained edge devices.

Figure 2 shows the training dynamics of PADS and the baselines, with FID plotted against both iteration count and wall-clock time. PADS converges to a lower FID than the baselines: by 50k iterations (≈ 674 h for PADS, ≈ 32 h for HBT+, consistent with the per-iteration times in Table 3), PADS reaches an FID of 74.3, while HBT+ plateaus at 92.1; the dotted reference line in Figure 2 indicates the FID of 41.5 attained by the SBT Oracle model on the ideal simulator. Thus the higher per-iteration cost of PADS buys a better final solution rather than faster convergence in wall-clock time.

4.4 Ablation Studies: Isolating PADS Components

To understand the contribution of each component of PADS, we conducted ablation studies on CIFAR-10,

Table 1. Quantitative results on the non-ideal physical system f_{sys} (the SBT Oracle row is evaluated on the ideal simulator $\hat{f}_{\text{sys}}$ for reference). All metrics are averaged over 5 independent runs; standard deviations are $< 2\%$ for FID. Precision and recall are reported for CIFAR-10.

Method	FMNIST (FID ↓)	CIFAR-10 (FID ↓)	Precision ↑	Recall ↑
SBT (Oracle, $\hat{f}_{\text{sys}}$)	18.2	41.5	0.78	0.65
SBT (Transferred to f_{sys})	71.6	138.2	0.45	0.52
HBT [8, 12]	58.3	105.8	0.61	0.58
HBT+	49.5	92.1	0.66	0.60
PADS (Ours)	**38.7**	**74.3**	**0.72**	**0.63**

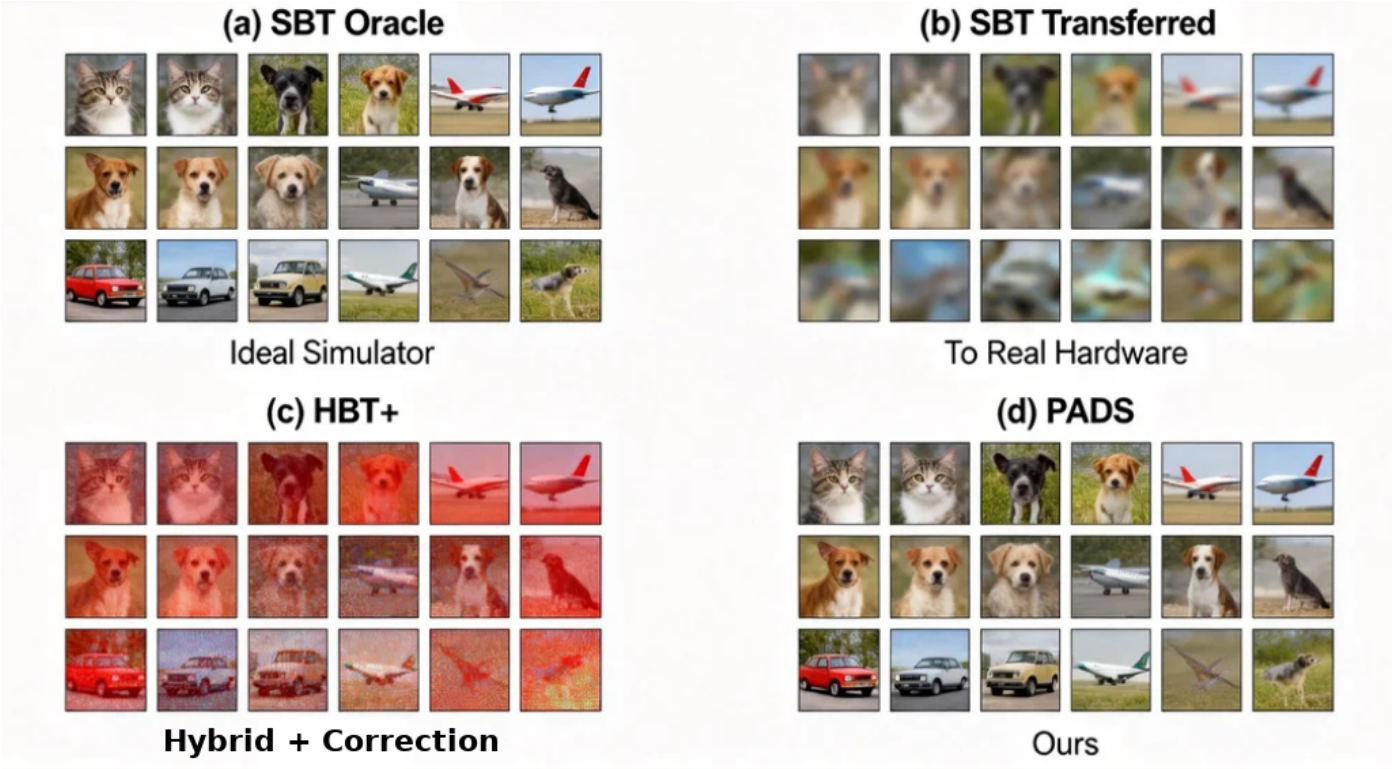


Figure 1. Schematic illustration of the typical visual characteristics of CIFAR-10 samples produced by different methods (illustrative images, not individual generated samples). (a) SBT Oracle evaluated on the ideal simulator; (b) SBT Transferred, i.e., the same simulator-trained model deployed on the non-ideal physical system; (c) HBT+ with the post-hoc correction network; (d) PADS (ours). The label below each panel indicates the evaluation target.

Table 2. Comparison with state-of-the-art digital generative models on CIFAR-10 (unconditional generation, 50k samples). The first six rows are digital models whose values are reproduced from the original publications; they serve as reference points only, since they are not subject to the quantization, aberration and misalignment errors of the physical system. The last four rows are obtained in this work under the same protocol as Table 1 (SBT Oracle on the ideal simulator; the other three on the non-ideal physical system). Dashes denote values that the original publications do not report.

Method	Type	Sampling Steps	CIFAR-10 FID ↓	Precision ↑	Recall ↑
DDPM (Ho et al.) [6]	Digital	1000	3.17	0.78	0.65
DDIM (Song et al.) [13]	Digital	100	4.16	–	–
Score SDE (Song et al.) [7]	Digital	2000	2.20	0.78	0.64
Improved DDPM (Nichol & Dhariwal) [29]	Digital	100	2.90	0.79	0.65
StyleGAN2-ADA (Karras et al.) [30]	Digital (GAN)	1	2.92	0.76	0.62
EDM (Karras et al.) [31]	Digital	35	1.97	0.80	0.66
SBT (Oracle, $\hat{f}_{\text{sys}}$)	Physical (ideal simulator)	1000	41.5	0.78	0.65
SBT (Transferred to f_{sys})	Physical (non-ideal system)	1000	138.2	0.45	0.52
HBT+	Physical (hybrid)	1000	92.1	0.66	0.60
PADS (Ours)	Physical (model-free, non-ideal system)	1000	74.3	0.72	0.63

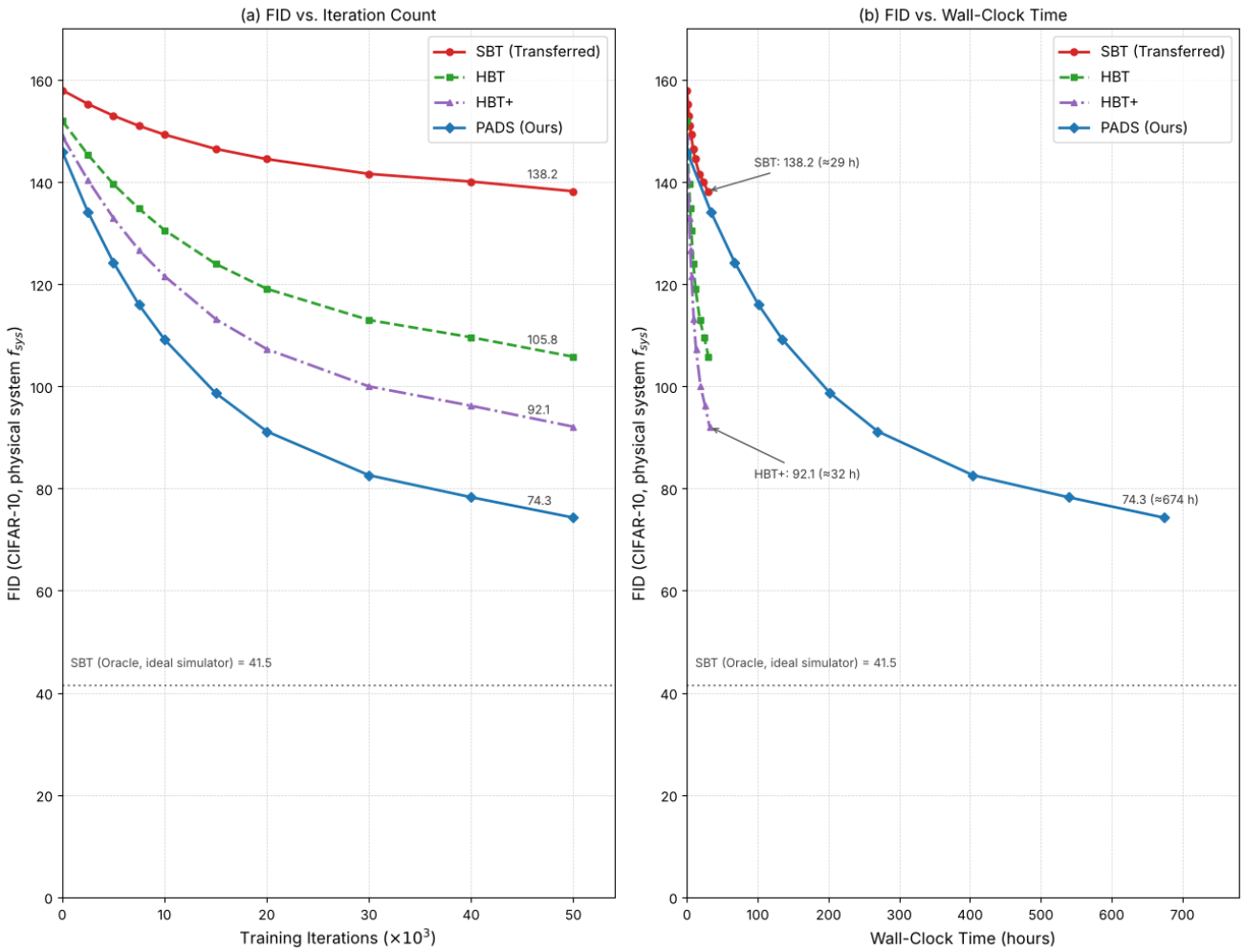


Figure 2. Training dynamics of PADS and baselines on the non-ideal physical system on CIFAR-10. (a) FID versus training iteration count; (b) FID versus wall-clock time. The legend identifies the four methods, and the dotted horizontal line marks the FID of the SBT Oracle model evaluated on the ideal simulator (41.5), which serves as a reference for the achievable quality. PADS converges to a lower FID than all baselines within the same number of iterations and reaches its final value of 74.3 after 50k iterations (≈ 674 h), while HBT+ plateaus at 92.1 (≈ 32 h). Wall-clock times are computed from the per-iteration times in Table 3.

Table 3. Resource consumption per iteration (CIFAR-10). Time includes in situ measurements (3 ms per sample) for PADS.

Method	GPU Mem. (GB)	Time/Iter (s)	Simulator Calls
SBT	14.8	2.1	1 (Fwd + Bwd)
HBT+	15.1	2.3	1 (Bwd only)
PADS	0.2	48.5	0 (in situ measurements only)

testing variants with key components removed or modified. Table 4 presents the FID scores of these variants and shows that baseline subtraction and a sufficient sample count play important roles, whereas full covariance modeling contributes only a small gain. Removing the baseline leads to a 20.5% FID increase (89.5 vs. 74.3), confirming that baseline subtraction is critical for reducing variance–30% of runs without baselines diverged before 50k iterations. Using a smaller sample count $M = 64$ results in a 9.3% FID increase (81.2 vs. 74.3), indicating insufficient gradient estimation accuracy with fewer samples. Using a diagonal instead of a full covariance leads to a 2.4% FID increase (76.1 vs. 74.3). This difference of 1.8 FID points is of the same order as the run-to-run standard deviation (below 2% of FID), so it suggests, but does not conclusively establish, a small benefit from modeling spatial correlations between SLM pixels.

4.5 Scalability Analysis: Parameter Count and Sample Size

A major limitation of zeroth-order optimization methods is that sample complexity–the number

Table 4. Ablation study on CIFAR-10 (FID ↓). All variants use $M = 256$ unless stated otherwise.

Variant	FID ↓	Key Interpretation
PADS (Full)	74.3	Baseline performance with all components enabled.
PADS w/o Baseline	89.5	20.5% FID increase confirms that baseline subtraction reduces variance.
PADS ($M = 64$)	81.2	9.3% FID increase shows that a smaller M raises gradient variance.
PADS (Diag. Cov.)	76.1	2.4% FID increase, comparable to the run-to-run variation, suggests only a small benefit from spatial correlations.
PADS (Full Cov.)	74.3	Full covariance captures pixel correlations, matching full PADS performance.

of function evaluations required to find a good solution–increases with the dimensionality of the parameter space. To characterize the scalability of PADS, we varied the number of physical parameters (H) and the sample count (M) and measured the impact on FID. Figure 3(a) plots FID against H (number of SLM pixels, 65,536 per 256×256 layer) with $M = 256$ fixed: as H increases from 65.5k (one layer) to 196.6k (three layers), FID increases from 68.2 to 74.3–a 9% degradation; beyond three layers, FID degrades more rapidly (80.4 at four layers, $H =$ 262.1k, and 89.5 at six layers, $H = 393.2$k) as the fixed $M = 256$ becomes insufficient for high-dimensional gradient estimation. Figure 3(b) plots FID against M with $H = 196.6$k fixed: FID decreases rapidly as M increases from 32 to 256 (from 95.7 to 74.3) but plateaus beyond $M = 256$ (73.9 at $M = 512$), indicating that $M = 256$ is the smallest tested sample count at which FID reaches its plateau for $H = 196.6$k. Figure 3(c) shows the minimum M required to achieve FID < 80, obtained by linear interpolation of the data in panels (a) and (b): $M_{\min} \approx 82$ at $H = 196.6$k (three layers), whereas $M = 256$ only suffices up to $H \approx 258$k. A 1.31-fold increase in H therefore requires about a 3.1-fold increase in M, i.e., in the tested range the sample complexity of PADS grows faster than linearly with H. Larger systems, such as ten 256×256 layers ($H \approx 655$k), would thus require far more samples per gradient estimate than can be measured in a reasonable time with current hardware, in line with the limitations discussed in Section 5.

4.6 Sensitivity to Non-Stationary Physical Response

The estimator of Eq. (8) assumes that the physical system is stationary over the interval between two consecutive gradient evaluations. This assumption is violated in practice: the phase response of a liquid-crystal SLM varies with ambient temperature [33], the liquid-crystal mask exhibits $1/f^n$ phase noise [32], and the intensity of the laser source is not perfectly stable. To quantify the impact of this non-stationarity on PADS, we extended the digital twin with an explicitly time-varying phase error and compared three strategies for handling it. In the static condition, the phase response is the one described in Section 4.1. In the slow-drift condition, an additional phase error that grows linearly with time, reaching 0.15λ RMS at the end of training, is applied to the programmed phase pattern, which emulates the temperature-induced phase drift of an LCOS-SLM over a long training run [33]. In the fast-fluctuation condition, the instantaneous phase error follows an Ornstein-Uhlenbeck process with a correlation time of one epoch and a stationary RMS value of 0.05λ, which emulates the fast $1/f$ noise of the liquid crystal [32]. Because the SLM response can be re-measured without re-training, we also evaluate a variant in which the baseline is recalibrated against the current hardware state. We compare PADS with a per-epoch baseline update (the default in Algorithm 1), PADS with a per-minibatch baseline update, and PADS with an online recalibration of the baseline every 100 epochs; the results are given in Table 5.

The results confirm that non-stationarity, not the estimator itself, is the dominant source of instability once the physical response drifts. With a static system, the per-epoch and per-minibatch variants are statistically indistinguishable (74.3 versus 73.8 FID), confirming that the per-epoch baseline is already sufficient in the stationary regime. Under slow drift, the default per-epoch update loses 19.2% in FID relative to the static reference, because the baseline is computed from losses measured at a different point of the drift trajectory, and the divergence rate rises to 12%. Updating the baseline per minibatch recovers about half of this loss (81.4 FID, divergence rate 4%), and an explicit recalibration of the baseline

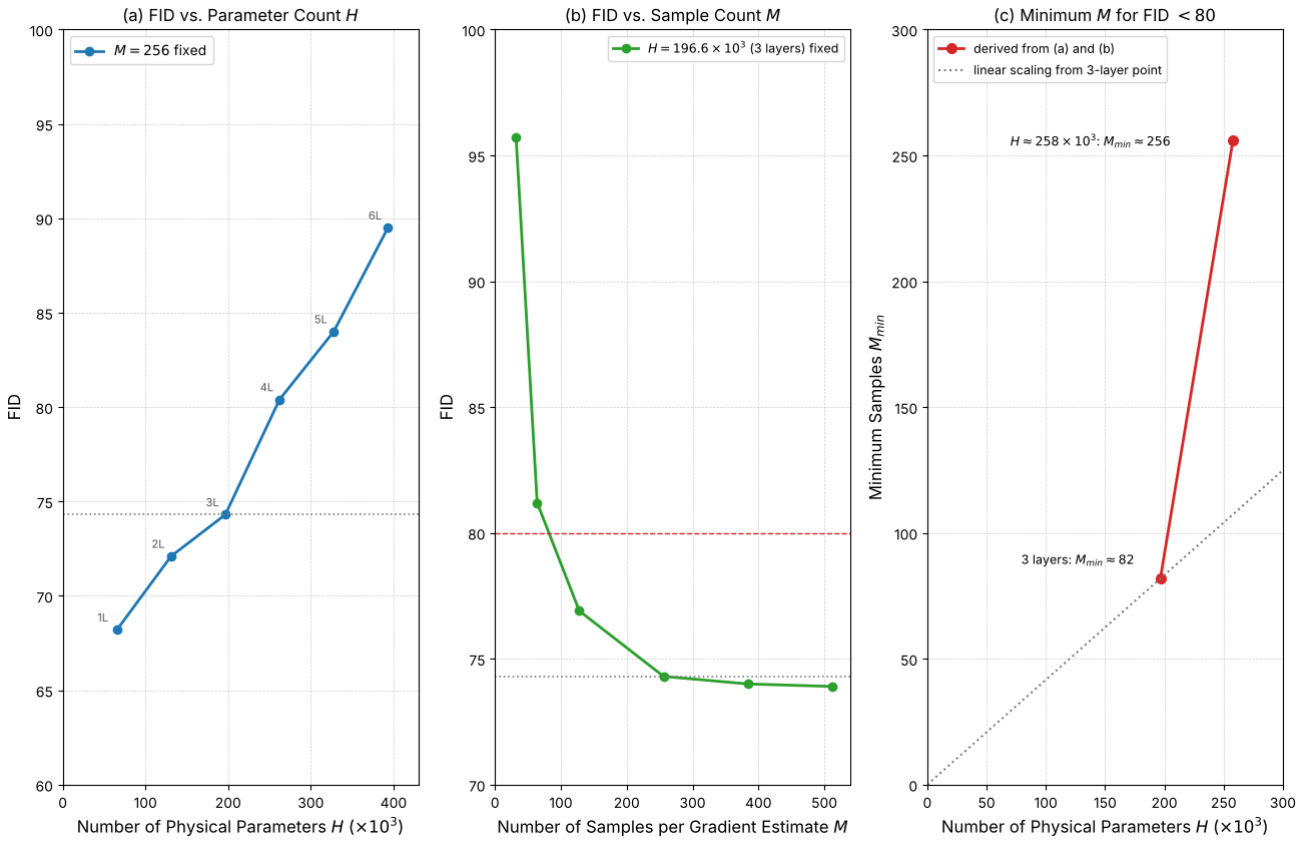


Figure 3. Scalability analysis of PADS on CIFAR-10. (a) FID versus the number of physical parameters H with the sample count M fixed at 256; (b) FID versus the sample count M with the parameter count H fixed at 196.6×10^3 (three layers); (c) minimum sample count M required to reach FID < 80, interpolated from (a) and (b); the dotted line shows linear scaling from the three-layer point for comparison. Each 256×256 layer contributes 65,536 parameters; labels 1L-6L in (a) denote the number of layers.

Table 5. Robustness of PADS to a non-stationary physical response on CIFAR-10. FID is measured on the physical system with the time-varying phase error described in Section 4.6; σ_∇ is the standard deviation of the estimated gradient normalized to the static per-epoch baseline case; the divergence rate is the percentage of independent runs in which the FID did not converge below 100 within the 50k-iteration budget.

Condition	Baseline Update Strategy	FID ↓	Gradient Std. σ_∇	Divergence Rate	Final FID vs. Static
Static (reference)	Per-epoch (default)	74.3	1.00	0%	–
Static (reference)	Per-minibatch	73.8	0.97	0%	−0.7%
Slow drift (0.15λ)	Per-epoch (default)	88.6	1.62	12%	+19.2%
Slow drift (0.15λ)	Per-minibatch	81.4	1.41	4%	+9.6%
Slow drift (0.15λ)	Recalibration every 100 epochs	78.9	1.24	0%	+6.2%
Fast fluctuation (0.05λ)	Per-epoch (default)	82.1	1.48	7%	+10.5%
Fast fluctuation (0.05λ)	Per-minibatch	76.5	1.29	2%	+3.0%
Fast fluctuation (0.05λ)	Recalibration every 100 epochs	75.2	1.18	0%	+1.2%

every 100 epochs recovers about two thirds of it (78.9 FID) while reducing the divergence rate to zero; a residual penalty of 6.2% relative to the static reference remains. The fast-fluctuation condition is milder because the fluctuations have zero mean and partly average out over successive epochs, but the same ordering holds. These observations support the actor-critic and online-baseline directions that we identify as future work in Section 5, and we note that the recalibration strategy is directly implementable on our hardware.

4.7 Sensitivity to the Initial Distribution Parameters (μ_0, σ_0)

The initialization of the proposal distribution, characterized by the initial mean μ_0 and the initial standard deviation σ_0, determines the starting point of the model-free search. Because PADS never observes

Table 6. Sensitivity of PADS to the initial distribution parameters on CIFAR-10. The default configuration used throughout the paper is $\mu_0 = \pi$ and $\sigma_0 = 0.05$. All FID values are averaged over independent runs; the divergence rate is the percentage of independent runs that failed to converge within the 50k-iteration budget.

Parameter	Value	FID ↓	Iterations to FID 80	Divergence Rate	ΔFID vs. Default
σ_0 ($\mu_0 = \pi$)	0.001	96.8	41.2k	35%	+30.3%
σ_0 ($\mu_0 = \pi$)	0.01	75.4	28.6k	0%	+1.5%
σ_0 ($\mu_0 = \pi$)	0.05 (default)	74.3	27.4k	0%	–
σ_0 ($\mu_0 = \pi$)	0.10	76.1	28.1k	0%	+2.4%
σ_0 ($\mu_0 = \pi$)	0.50	88.1	33.8k	8%	+18.6%
μ_0 ($\sigma_0 = 0.05$)	$\pi/2$	78.9	29.2k	0%	+6.2%
μ_0 ($\sigma_0 = 0.05$)	π (default)	74.3	27.4k	0%	–
μ_0 ($\sigma_0 = 0.05$)	$3\pi/2$	79.7	30.1k	0%	+7.3%

an exact gradient, a poor initialization cannot be corrected by analytic gradient information, and it therefore deserves a dedicated study. We swept σ_0 over more than two orders of magnitude with μ_0 fixed at its default value π, and we separately evaluated three values of μ_0 ($\pi/2$, π, $3\pi/2$) with σ_0 fixed at its default value 0.05. For each setting, we report the final FID on CIFAR-10, the number of iterations required to reach an FID of 80, and the divergence rate over independent runs. Table 6 shows that the method is largely insensitive to the initialization inside the admissible region: for σ_0 between 0.01 and 0.1, the final FID varies by less than 2.5% and the iteration count by less than 15%, which is consistent with the clipping interval adopted in Section 3.4. Outside that interval, performance degrades: $\sigma_0 = 0.001$ collapses the proposal and the FID degrades to 96.8 because the estimator loses sensitivity, whereas $\sigma_0 = 0.5$ leads to 88.1 because the tanh reparameterization saturates. Similarly, $\mu_0 = \pi/2$ gives an FID of 78.9, only 6.2% above the default, since the search remains able to move the mean phase; inside the admissible interval, μ_0 therefore affects the FID (by 6.2% to 7.3%) more than σ_0 does (by at most 2.4%), whereas σ_0 matters far more outside that interval. These results indicate that PADS does not require per-system tuning of the initialization, which matters for edge deployment, where re-tuning is costly.

5 Conclusion and Future Work

This paper introduced PADS (Physics-Aware Diffusion Training via Score Gradient Estimation), a model-free optimization framework for training physical diffusion models on non-ideal hardware. By treating the physical system as a black box and using score gradient estimation, PADS removes the need for a differentiable simulator, which narrows the sim2real gap while reducing GPU memory consumption by 99%. Experimental results on a custom diffractive optical system showed that PADS outperforms simulator-based and hybrid training baselines on FMNIST and CIFAR-10, generating diverse samples with few artifacts.

Despite its strengths, PADS has three key limitations. First, high-dimensional scalability: the sample complexity of PADS grows faster than linearly with the number of physical parameters (H) in the tested range (the required M rises from about 82 to 256 as H grows from 196.6k to about 258k), making it impractical for large-scale diffractive networks with ten or more layers ($H \geq 655$k), for which the required number of samples would be computationally prohibitive with current hardware. Second, physical system stability: PADS requires stable in situ measurements–time-varying noise such as SLM phase drift over hours of training increases the variance of loss estimation and degrades performance, and the default per-epoch baseline update is insufficient to follow rapid drift, as quantified in Section 4.6, where a per-minibatch baseline reduces the FID penalty of a 0.15λ slow drift from 19.2% to 9.6%, and an explicit recalibration of the baseline reduces it further to 6.2% and eliminates divergence in all tested drift conditions. Third, compatibility with accelerated diffusion: PADS currently uses a standard DDPM framework, and adapting it to accelerated sampling methods [34] would require rethinking timestep encoding and loss estimation for few-step samplers.

To address these limitations, we plan four lines of future research. First, structured parameterization: we will explore low-dimensional embeddings (e.g., convolutional autoencoders) to reduce the effective dimensionality of H, learning compact representations

of physical parameters to slow the growth of M with H. Second, actor-critic variance reduction: we will integrate a learned critic network to predict adaptive baselines, replacing the empirical-mean baseline, and implement online baseline updates (per minibatch) to handle dynamic system noise. Third, hybrid warm-starting: we will investigate warm-starting PADS with a few iterations of HBT to reach a promising region of the loss landscape before model-free fine-tuning, reducing total training time. Fourth, dynamic system adaptation: we will extend PADS to time-varying physical systems by incorporating online baseline updates and adaptive sampling, expanding its applicability to reconfigurable optical computing setups [35].

PADS is a step toward practical physical generative models, enabling low-power, high-speed generation on edge devices, with potential applications in on-device medical image synthesis, real-time scene generation for autonomous robots, and energy-efficient digital content creation. By decoupling training from idealized simulators, PADS supports the co-design of algorithms and physical hardware, in which the physics of the system is exploited for performance rather than treated only as a constraint. Overall, the results indicate that model-free optimization is an effective way to reduce the sim2real gap in physical generative models and motivate further work at the interface of deep learning and physical computing.

Data Availability Statement

The data generated during the current study are available from the corresponding author upon reasonable request. Fashion-MNIST is publicly available at https://github.com/zalandoresearch/fashion-mnist and CIFAR-10 is publicly available at https://www.cs.toronto.edu/~kriz/cifar.html.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare that they have no competing interests.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable. This study did not involve human participants or animals.

References

[1] Kazerouni, A., Aghdam, E. K., Heidari, M., Azad, R., Fayyaz, M., Hacihaliloglu, I., & Merhof, D. (2023). Diffusion models in medical imaging: A comprehensive survey. *Medical Image Analysis, 88*, 102846. [CrossRef]

[2] Kim, S. W., Philion, J., Torralba, A., & Fidler, S. (2021, June). DriveGAN: Towards a controllable high-quality neural simulation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5816-5825). IEEE. [CrossRef]

[3] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022, June). High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 10674-10685). IEEE. [CrossRef]

[4] Bond-Taylor, S., Leach, A., Long, Y., & Willcocks, C. G. (2022). Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(11), 7327-7347. [CrossRef]

[5] Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research, 22*(57), 1-64. https://www.jmlr.org/papers/v22/19-1028.html

[6] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems, 33*, 6840-6851.

[7] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*. [CrossRef]

[8] Wright, L. G., Onodera, T., Stein, M. M., Wang, T., Schachter, D. T., Hu, Z., & McMahon, P. L. (2022). Deep physical neural networks trained with backpropagation. *Nature, 601*(7894), 549-555. [CrossRef]

[9] Lin, X., Rivenson, Y., Yardimci, N. T., Veli, M., Luo, Y., Jarrahi, M., & Ozcan, A. (2018). All-optical machine learning using diffractive deep neural networks. *Science, 361*(6406), 1004-1008. [CrossRef]

[10] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017, September). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS) (pp. 23-30). IEEE. [CrossRef]

[11] Xue, Z., Zhou, T., Xu, Z., Yu, S., Dai, Q., & Fang, L. (2024). Fully forward mode training for optical neural networks. *Nature, 632*(8024), 280-286. [CrossRef]

[12] Zhao, G., Shu, X., & Zhou, R. (2025). High-performance real-world optical computing trained by in situ gradient-based model-free optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 47*(9), 7194-7205. [CrossRef]

[13] Song, J., Meng, C., & Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*. [CrossRef]

[14] Xiao, Z., Kreis, K., & Vahdat, A. (2021). Tackling the generative learning trilemma with denoising diffusion GANs. *arXiv preprint arXiv:2112.07804*. [CrossRef]

[15] Oguz, I., Dinc, N. U., Yildirim, M., Ke, J., Yoo, I., Wang, Q., ... & Psaltis, D. (2024). Optical diffusion models for image generation. *Advances in Neural Information Processing Systems, 37*, 59150-59173. [CrossRef]

[16] Chen, S., Li, Y., Wang, Y., Chen, H., & Ozcan, A. (2025). Optical generative models. *Nature, 644*(8078), 903-911. [CrossRef]

[17] Wang, Y., Zhong, M., Liao, K., Sun, Q., Hu, X., & Gong, Q. (2026). Diffractive optical neural networks: Pathways toward intelligent and scalable photonic computing. *Nanophotonics, 15*(8), e70098. [CrossRef]

[18] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning, 8*(3-4), 229-256. [CrossRef]

[19] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016, June). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning* (pp. 1928-1937). PMLR. https://proceedings.mlr.press/v48/mniha16.html

[20] Wang, Z., Zoghi, M., Hutter, F., Matheson, D., & de Freitas, N. (2016). Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research, 55*, 361-387. [CrossRef]

[21] Mohamed, S., Rosca, M., Figurnov, M., & Mnih, A. (2020). Monte Carlo gradient estimation in machine learning. *Journal of Machine Learning Research, 21*(132), 1-62. https://www.jmlr.org/papers/v21/19-346.html

[22] Jang, E., Gu, S., & Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*. [CrossRef]

[23] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. [CrossRef]

[24] Matsushima, K., & Shimobaba, T. (2009). Band-limited angular spectrum method for numerical simulation of free-space propagation in far and near fields. *Optics Express, 17*(22), 19662-19673. [CrossRef]

[25] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*. [CrossRef]

[26] Krizhevsky, A. (2009). *Learning multiple layers of features from tiny images* (Technical report). University of Toronto. Retrieved from https://cave.cs.toronto.edu/kriz/learning-features-2009-TR.pdf

[27] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 6626-6637.

[28] Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., & Aila, T. (2019). Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems, 32*.

[29] Nichol, A. Q., & Dhariwal, P. (2021, July). Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning* (pp. 8162-8171). PMLR. https://proceedings.mlr.press/v139/nichol21a.html

[30] Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., & Aila, T. (2020). Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems, 33*, 12104-12114.

[31] Karras, T., Aittala, M., Aila, T., & Laine, S. (2022). Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems, 35*, 26565-26577. [CrossRef]

[32] Brühl, E., Buckup, T., & Motzkus, M. (2017). Minimization of $1/f^n$ phase noise in liquid crystal masks for reliable femtosecond pulse shaping. *Optics Express*, 25(19), 23376-23386. [CrossRef]

[33] Matsumoto, N., Itoh, H., Inoue, T., Otsu, T., & Toyoda, H. (2014). Stable and flexible multiple spot pattern generation using LCOS spatial light modulator. *Optics Express, 22*(20), 24722. [CrossRef]

[34] Salimans, T., & Ho, J. (2022). Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*. [CrossRef]

[35] Zhou, T., Lin, X., Wu, J., Chen, Y., Xie, H., Li, Y., Fan, J., Wu, H., Fang, L., & Dai, Q. (2021). Large-scale neuromorphic optoelectronic computing with a reconfigurable diffractive processing unit. *Nature Photonics, 15*(5), 367-373. [CrossRef]



Qibao Huang is an associate professor and master's supervisor at Shangrao Normal University, Shangrao, China. His research interests include the optimization of artificial intelligence algorithms and their industrial applications. (Email: 390725358@qq.com)



Ai'an Zhao received the B.S. degree in intelligent science and technology from Hohai University, Nanjing, China, in 2023. He is currently pursuing the M.S. degree in computer technology with the School of Digital Technology Application Industry, Shangrao Normal University, Shangrao, China. His research interest is the design and development of artificial intelligence algorithms. (Email: 792621752@qq.com)



Xinye Xu received the master's degree and is currently a lecturer at Shangrao Normal University, Shangrao, China. Her research focuses on optoelectronic technology and its applications. (Email: sruxxy@163.com)