CONSTRAINED DIFFUSION MODELS WITH PRIMAL-DUAL INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper develops constrained diffusion models with primal-dual inference (PDI) to sample from optimal distributions of entropy-regularized optimization problems with *average* constraints. We formalize constrained sampling in the Lagrangian dual domain, where the optimal distribution takes the form of a Gibbs distribution indexed by the optimal dual variable. Rather than estimating this dual multiplier before sampling and freezing it throughout generation, PDI jointly infers the optimal primal distribution and its parametrizing dual variable. Each reverse diffusion step denoises using the score field associated with the current multiplier and then updates the multiplier through dual ascent using the estimated constraint violation of the denoised samples. To enable this conditional score field, we train a single dual-conditioned score network over the family of Gibbs distributions induced by the dual variables encountered during inference. We prove that the dual iterates converge to a neighborhood of the dual optimum, and that the generated distribution approaches that of an oracle sampler that knows the optimal multiplier, up to a bias floor. We evaluate PDI on constrained sampling from a mixture of Gaussians, wireless resource allocation, and portfolio management. Our code is available at: https://anonymous.4open.science/r/PDI-Diffusion-7173/README.md.

1 INTRODUCTION

Diffusion models have become a dominant framework for sampling from complex, high-dimensional distributions by learning to reverse a noising process (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021b). While their most visible successes lie in unconstrained perceptual tasks such as image (Rombach et al., 2022), audio (Kong et al., 2021), and video synthesis (Ho et al., 2022), the same modeling capabilities are valuable for constrained optimization problems whose optimal solutions are distributions rather than single points. This is because when the constraints need only hold on average, randomized solutions can do strictly better. In wireless resource allocation, for instance, interference can prevent any single power allocation from meeting every user's rate requirement, whereas a time-sharing policy alternating between allocations meet them on average while maximizing the network utility (Uslu et al., 2025; Babazadeh Darabi and Coleri, 2025). Similarly, portfolio management seeks diverse allocations that maximize expected returns under expected risk constraints (Tiwari, 2026; Gao et al., 2025). Standard diffusion samplers, however, lack effective mechanisms to enforce such average constraints. Guidance and projection methods correct each sample individually (Chung et al., 2023; Christopher et al., 2024), which imposes a stricter per-sample requirement and can exclude solutions that are feasible only on average. This paper bridges that gap by developing *primal–dual inference* (PDI), a diffusion sampler that enforces average constraints during sampling.

Entropy-regularized problems over distributions with average constraints admit a clean form in the Lagrangian dual domain (Chamon et al., 2024; Khalafi et al., 2024). For a fixed dual variable, the distributional minimizer of the Lagrangian is a Gibbs distribution whose energy is linear in the dual variable. This connects constrained sampling to the vast literature on diffusion-based samplers for unnormalized Gibbs targets (Zhang and Chen, 2022; Vargas et al., 2023; Berner et al., 2024; Richter and Berner, 2024; Akhound-Sadegh et al., 2024). In principle, one could first solve for the optimal dual variable and then train a diffusion model to sample from the corresponding Gibbs distribution. In our setting, however, the sampler must serve a family of problem instances, each with its own optimal multiplier, and each multiplier is defined through expectations under the very Gibbs distribution the sampler is meant to learn. Dual-training (DT) approaches (Khalafi et al., 2024; 2025) resolve

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

this circularity by alternating dual ascent with score-model training. Retraining the score model to convergence after every dual update is expensive, while truncating it leaves a score model that blends past targets rather than matching the Gibbs distribution at the final multiplier. Either way, once training finalizes, the sampler is frozen and cannot respond to constraint violations or adapt to shifted constraint levels during generation. Primal–dual Langevin Monte Carlo (Chamon et al., 2024) instead updates the multiplier at sampling time, but it relies on a Markov chain that must mix, which is slow for multi-modal targets, and it does not learn a generative model that transfers across instances.

PDI proposes a different view by bringing the sampling-time dual updates of primal–dual Langevin into the reverse diffusion itself. We make the multiplier part of the sampling state. At each denoising step, PDI updates the samples with the score field of the current multiplier, evaluates the constraints at Tweedie estimates of the denoised samples (Efron, 2011), and takes a dual ascent step. Because this step uses the ensemble average with a shared multiplier, the constraints are enforced on average rather than on each sample. Different from Langevin dynamics, PDI does not rely on mixing. The reverse process transports noise to the target in a fixed number of steps, following a path-dependent sequence of Gibbs score fields while the multiplier approaches its optimum. These scores are supplied by a single score network conditioned on the multiplier and the problem instance, so that one model represents the entire family of Gibbs targets. Since no samples from these targets are available directly, the network regresses onto Monte Carlo score estimates computed from the energy alone (Akhound-Sadegh et al., 2024).

Our theoretical analysis makes the benefits of this coupling precise. The PDI multipliers converge to a neighborhood of its optimum within a single reverse pass, while the generated distribution approaches that of an oracle sampler that knows the optimal multiplier, up to a floor set by the Tweedie, reverse-sampling and score-approximation errors. Thus not knowing the optimal multiplier in advance costs little beyond this bias, and the generated samples are approximately feasible in expectation. Empirically, we show that PDI matches the feasibility of projection- and Langevin-based baselines with a better objective and better mode coverage in a constrained Gaussian mixture setup. In wireless resource allocation, PDI attains better feasibility and tail rates than DT and degrades more gracefully under shifted constraint levels. DT becomes competitive only after an additional stage that freezes its final multiplier and continues score training for that single Gibbs target, which requires more than ten times as many outer training iterations as PDI. This suggests that DT's gap lies in fitting the score at the right multiplier, which the conditional network provides without retraining. Freezing the multiplier within PDI at its own final or time-averaged value increases the total constraint violation, indicating that the inference-time coupling itself also contributes.

We make the following contributions.

- **(C1)** We cast constrained sampling with average constraints as a saddle-point problem in the Lagrangian dual domain, where the optimal distribution is a Gibbs distribution indexed by an optimal dual variable.
- **(C2)** We propose PDI, a reverse-diffusion sampler that treats the dual multiplier as part of its state and solves the dual problem within a single diffusion pass by interleaving each denoising diffusion step with a dual ascent step.
- **(C3)** We train a single score network that represents the entire family of Gibbs targets parametrized by arbitrary dual multipliers, without requiring the optimal multiplier. The network adapts to new problem instances and shifted constraint levels at inference time without retraining.
- **(C4)** We prove that the dual iterates converge to a neighborhood of the optimal multiplier despite using biased gradient estimates, and that the output distribution of PDI is within $\mathcal{O}(T^{-1/2})$ in KL divergence of an oracle sampler that knows the optimal multiplier, up to a bias floor.
- **(C5)** We validate PDI empirically on a constrained Gaussian mixture, wireless power control, and portfolio construction. PDI achieves a better objective than per-sample baselines at matched feasibility, matches strengthened dual training with over ten times fewer outer training iterations, and degrades more gracefully under shifted constraint levels.

Figure 1 illustrates the effect of inference-time dual updates on wireless power allocation, where users share a channel and each requires a minimum average rate. An unconstrained sampler leaves many users underserved. DT improves feasibility, but its multipliers are set during training, so it cannot correct violations that arise during generation. PDI updates the multiplier as it denoises, steering samples toward feasibility on average and lifting the worst-served users closer to their targets.

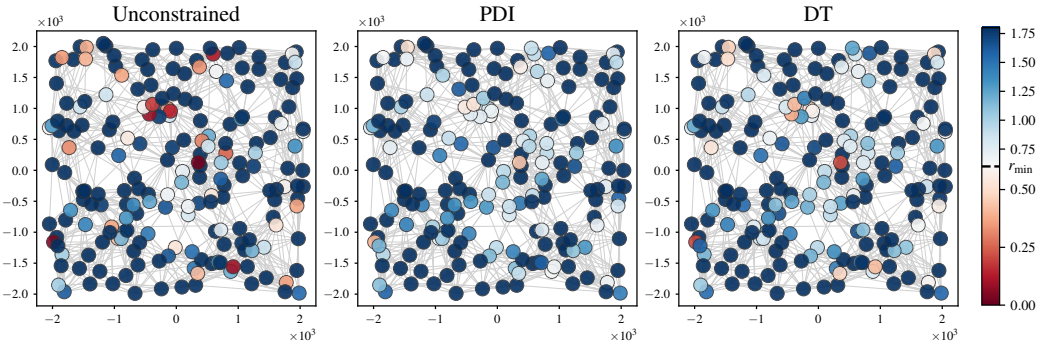

Figure 1: **Wireless power allocation under ergodic minimum-rate $r_{\min}$ constraints.** Nodes represent users and colors indicate how well the rate requirement of each user is met, with red marking those that fall short. Here, average constraints are meaningful because mutual interference prevents any single solution from satisfying every requirement at once. Optimal solutions must therefore alternate samples (i.e., power allocations and/or channel access) over time, so that some compensate for others. The unconstrained sampler leaves many users below the target rate (red nodes), while DT partially improves feasibility through training-time dual updates. PDI further improves the lower-rate users by updating the dual variables during denoising.

2 Related Work

Diffusion model sampling with constraints. Guidance techniques steer reverse diffusions toward reward or constraint alignment by modifying the score in Tweedie's denoising formula (Efron, 2011), as in classifier guidance (Dhariwal and Nichol, 2021), classifier-free guidance (Ho and Salimans, 2022), and CLIP-based variants (Nichol et al., 2022). A separate line of work enforces domain-feasibility constraints directly on the diffusion process through boundary reflections (Lou and Ermon, 2023), mirror maps (Liu et al., 2023), Riemannian formulations (De Bortoli et al., 2022), log-barrier constructions (Fishman et al., 2023), feasibility projections (Christopher et al., 2024), and constrained reverse steps (Huang et al., 2024a). Most importantly, these methods address pointwise constraints rather than distributional (average) ones.

Constrained diffusion modeling with Lagrangian dual formulations. Closely related to our work are (Khalafi et al., 2024; 2025; Chamon et al., 2024; Gutjahr et al., 2026), which adopt primal–dual formulations for average constraints. Specifically, (Khalafi et al., 2024; 2025) pair outer-loop dual ascent with inner-loop score matching but retrain the score network for each dual iterate, while primal–dual Langevin Monte Carlo (Chamon et al., 2024) jointly evolves samples and multipliers at sampling time but forgoes a learned generative model. Our method shares the Lagrangian saddle-point structure of these works but differs in two key aspects. First, the dual variables evolve jointly with the denoising trajectory rather than in a separate training loop or ergodic Langevin chain. Second, a single score network, conditioned on the dual variable, is trained across a distribution of multipliers rather than refitted per iterate. An extended related work is provided in Appendix A.

3 Constrained Diffusion Models

Let $\mathcal{X} \subset \mathbb{R}^p$ be a compact domain of decision variables $\mathbf{x}$ and $\mathcal{P}(\mathcal{X})$ the space of all probability distributions supported on $\mathcal{X}$. A constraint function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^d$ encodes the set of requirements that need only hold *on average*. Given also an objective cost $f_0 : \mathcal{X} \to \mathbb{R}$, and a temperature $\beta > 0$, the target distribution μ^* solves the entropy-regularized problem,

$$\begin{aligned} P^* \;=\; \min_{\mu \in \mathcal{P}(\mathcal{X})} \quad & \mathbb{E}_\mu\big[f_0(\mathbf{x})\big] - \beta\, \mathcal{H}(\mu), \\ \text{subject to} \quad & \mathbb{E}_\mu\big[\mathbf{f}(\mathbf{x})\big] \;\preceq\; \mathbf{0}, \end{aligned} \tag{1}$$

where $\mathcal{H}(\mu) = -\int_{\mathcal{X}} \mu \log \mu \, \mathrm{d}\mathbf{x}$ is the differential entropy if μ admits a density, and $\mathcal{H}(\mu) = -\infty$ otherwise. Throughout, f_0 and $\mathbf{f}$ are bounded and measurable. Problem (1) is equivalently KL-regularized as $D_{\mathrm{KL}}(\mu\|\nu) = \int \log(\mathrm{d}\mu/\mathrm{d}\nu)\, \mathrm{d}\mu$ toward a uniform distribution $\nu = \mathrm{Uniform}(\mathcal{X})$.

Constraining averages rather than individual samples is valuable in settings such as multi-player cooperative games and wireless resource optimization problems, where a single deterministic policy may not satisfy all competing requirements simultaneously. For instance, when interfering users

cannot all be served at once, a μ^* can alternate between decisions that compensate for another via time-sharing to improve the trade-off between objective utility and feasibility. In our applications, f_0 and $\mathbf{f}$ depend on a problem instance $\mathcal{G}$, e.g., the channel gains of a wireless network. The solution μ^* is therefore instance-specific, but we suppress this dependence for simplicity of presentation.

The problem in (1) is (β-strongly) convex in μ with a zero duality gap and can be handled in the dual domain. Given Lagrangian multipliers $\boldsymbol{\lambda} \succeq \mathbf{0}$, we form the Lagrangian function,

$$\mathcal{L}\big(\mu, \boldsymbol{\lambda}\big) \;=\; \mathbb{E}_\mu\Big[\, f_0(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x}) \,\Big] - \beta\, \mathcal{H}(\mu), \tag{2}$$

The dual function is $g(\boldsymbol{\lambda}) = \min_{\mu \in \mathcal{P}(\mathcal{X})} \mathcal{L}(\mu, \boldsymbol{\lambda})$, and the dual problem is given by

$$D^* \;=\; \max_{\boldsymbol{\lambda} \succeq \mathbf{0}} \; g(\boldsymbol{\lambda}) \;=\; \max_{\boldsymbol{\lambda} \succeq \mathbf{0}} \; \min_{\mu \in \mathcal{P}(\mathcal{X})} \mathbb{E}_\mu\Big[\, f_0(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x}) \,\Big] - \beta\, \mathcal{H}(\mu). \tag{3}$$

For a fixed $\boldsymbol{\lambda}$, the inner minimizer of the Lagrangian is the Gibbs distribution (Dupuis and Ellis, 1997, p. 34),

$$\mu^\dagger_{\boldsymbol{\lambda}}(\mathbf{x}) = \frac{\exp\big(-E(\mathbf{x}, \boldsymbol{\lambda})\big)}{Z(\lambda)}, \qquad E(\mathbf{x}, \boldsymbol{\lambda}) \;=\; \frac{1}{\beta}\Big(f_0(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x})\Big), \tag{4}$$

where $E(\mathbf{x}, \boldsymbol{\lambda})$ is the energy, and $Z(\boldsymbol{\lambda}) = \int_{\mathcal{X}} \exp(-E(\mathbf{x}, \boldsymbol{\lambda}))\, \mathrm{d}\mathbf{x}$ is the partition function for normalization $\int_{\mathcal{X}} \mu^\dagger_{\boldsymbol{\lambda}}\, \mathrm{d}\mathbf{x} = 1$. The Lagrangian minimizer $\mu^\dagger_{\boldsymbol{\lambda}}$ is thus an exponential tilt of the implicit uniform prior distribution ν, and the dual function is given by $g(\boldsymbol{\lambda}) = -\beta \log Z(\boldsymbol{\lambda})$.

The next proposition shows that the primal and dual problems are equivalent under Slater, and the optimal distribution μ^* is also a Gibbs distribution parametrized by an optimal dual multiplier $\boldsymbol{\lambda}^*$.

Proposition 1 (Strong duality (Chamon et al., 2024))**.** *Under Slater's condition (stated as Assumption 1 in the Appendix C.1), and for $\beta > 0$, we have $P^* = D^*$, and the dual problem* (3) *is attained at some finite $\boldsymbol{\lambda}^* \in \boldsymbol{\Lambda}$. The optimal pair $(\mu^*, \boldsymbol{\lambda}^*)$ is a saddle-point of the Lagrangian, i.e., $\mathcal{L}(\mu^*, \boldsymbol{\lambda}) \;\le\; \mathcal{L}(\mu^*, \boldsymbol{\lambda}^*) \;\le\; \mathcal{L}(\mu, \boldsymbol{\lambda}^*)$, for any $\mu \in \mathcal{P}(\mathcal{X})$ and $\boldsymbol{\lambda} \in \mathbb{R}^d_+$, and $\mu^* = \mu^\dagger_{\boldsymbol{\lambda}^*}$.*

The dual function is smooth and concave, with

$$\nabla g(\boldsymbol{\lambda}) \;=\; \mathbb{E}_{\mu^\dagger_{\boldsymbol{\lambda}}}\big[\mathbf{f}(\mathbf{x})\big], \qquad \nabla^2 g(\boldsymbol{\lambda}) \;=\; -\tfrac{1}{\beta}\, \mathrm{Cov}_{\mu^\dagger_{\boldsymbol{\lambda}}}\big(\mathbf{f}(\mathbf{x})\big). \tag{5}$$

Dual ascent on (3) only requires the expected constraint values under the Gibbs distribution indexed by the current multiplier. The energy is known in closed form for every $\boldsymbol{\lambda}$; however, this does not reveal the optimal distribution, since $\boldsymbol{\lambda}^*$ is defined implicitly through the distribution it indexes. Nor does it provide samples; $\mu^\dagger_{\boldsymbol{\lambda}}$ is known only up to its normalizing constant, and importance-reweighting or rejecting proposals from ν become inefficient in high dimensions (Chatterjee and Diaconis, 2018).

Solving (3) iteratively thus requires samples from $\mu^\dagger_{\boldsymbol{\lambda}}$ at every multiplier visited by dual ascent, and existing approaches differ in where this dual search takes place. Primal–dual Langevin Monte Carlo (Chamon et al., 2024) performs it at sampling time where a Langevin chain sidesteps the normalizing constant by following the energy gradient, and its samples supply the constraint estimates for each dual update. Prior constrained diffusion models (Khalafi et al., 2024; Gutjahr et al., 2026) perform it at training time, alternating dual updates with score-model training, which leaves a frozen sampler once training is complete. Diffusion-based samplers for unnormalized densities (Zhang and Chen, 2022; Vargas et al., 2023; Berner et al., 2024; Richter and Berner, 2024) avoid the mixing time of Langevin chains (Menz and Schlichting, 2014; Dalalyan, 2017) by transporting Gaussian noise through a sequence of smoothed marginals.

We take a different route and propose *primal–dual inference (PDI)*, which moves the dual search to sampling time within a diffusion process by making the multiplier a state of the reverse process.

3.1 Primal–Dual Inference

We want to generate samples from the Gibbs family $\{\mu^\dagger_{\boldsymbol{\lambda}} \,|\, \boldsymbol{\lambda} \in \boldsymbol{\Lambda}\}$, which contains $\mu^* = \mu^\dagger_{\boldsymbol{\lambda}^*}$ by Proposition 1. We build the sampler from the reverse process of a score-based diffusion model (Song et al., 2021b). To construct it, consider the variance-preserving (VP) forward process that would noise samples of $\mu^\dagger_{\boldsymbol{\lambda}}$,

$$\mathbf{y}_\tau(\boldsymbol{\lambda}) \;=\; \sqrt{a_\tau}\, \mathbf{y}_{\tau-1}(\boldsymbol{\lambda}) + \sqrt{b_\tau}\, \boldsymbol{\epsilon}_\tau, \qquad \mathbf{y}_0(\boldsymbol{\lambda}) \sim \mu^\dagger_{\boldsymbol{\lambda}}, \tag{6}$$

with $a_\tau = 1 - b_\tau$ and $\boldsymbol{\epsilon}_\tau \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \perp \mathbf{y}_0$. Its marginals $q_\tau^{\boldsymbol{\lambda}} := q_\tau(\cdot|\boldsymbol{\lambda})$ are the Gaussian smoothings $q_\tau(\mathbf{x}|\boldsymbol{\lambda}) = \int \mathcal{N}(\mathbf{x}; \alpha_\tau \mathbf{y}, \sigma_\tau^2 \mathbf{I})\, \mu_{\boldsymbol{\lambda}}^\dagger(\mathrm{d}\mathbf{y})$, where $\alpha_\tau^2 = \prod_{s \le \tau} a_s$ and $\sigma_\tau^2 = 1 - \alpha_\tau^2$, so that $q_T^{\boldsymbol{\lambda}} \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$ hold for every $\boldsymbol{\lambda}$. We write $\mathrm{SNR}_\tau := \alpha_\tau^2 / \sigma_\tau^2$ for the signal-to-noise ratio (SNR) which decays monotonically in τ. Since samples of $\mu_{\boldsymbol{\lambda}}^\dagger$ are unavailable, we never simulate this process and thus cannot train on its noised samples as standard diffusion models do. The process only serves to construct the reverse process, and we explain shortly how its scores are learned without such samples.

We index the reverse (sampling) process by inference time $t = T - \tau$. It starts from $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ at $t = 0$ and runs to $t = T$, producing a sample $\mathbf{x}_T$. For each fixed $\boldsymbol{\lambda}$, the reverse process is

$$\mathbf{x}_{t+1} \;=\; \frac{1}{\sqrt{a_{T-t}}} \Big(\mathbf{x}_t + b_{T-t}\, \nabla \log q_{T-t}(\mathbf{x}_t|\boldsymbol{\lambda}) \Big) + \sqrt{b_{T-t}}\, \boldsymbol{\epsilon}_t, \tag{7}$$

where the $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ are independent of each other and of $\mathbf{x}_0$. This is a discretization of the time reversal of the forward diffusion (Anderson, 1982; Song et al., 2021b), and its final iterate $\mathbf{x}_T$ is approximately distributed as $\mu_{\boldsymbol{\lambda}}^\dagger$ for large enough T and suitable noise schedules (Chen et al., 2023). In particular, it would approximately sample μ^* if $\boldsymbol{\lambda}^*$ were known.

PDI updates the multiplier along the reverse process instead. At each step, a dual-conditioned score model $s_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}_t)$ drives both a primal and a dual update:

$$\mathbf{x}_{t+1} \;=\; \frac{1}{\sqrt{a_{T-t}}} \Big(\mathbf{x}_t + b_{T-t}\, s_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}_t) \Big) + \sqrt{b_{T-t}}\, \boldsymbol{\epsilon}_t, \tag{PDI-P}$$

$$\boldsymbol{\lambda}_{t+1} \;=\; \Big[\boldsymbol{\lambda}_t + \eta_t\, \mathbb{E}_{p_t} \big[\mathbf{f}\big(\widehat{\mathbf{y}}_0(\mathbf{x}_t, \boldsymbol{\lambda}_t)\big) \big] \Big]_+ . \tag{PDI-D}$$

Here, p_t is the law of $\mathbf{x}_t$ under PDI, $\eta_t > 0$ is the dual step size, and $[\cdot]_+$ is the projection onto the nonnegative orthant. The clean-sample estimate $\widehat{\mathbf{y}}_0(\mathbf{x}, \boldsymbol{\lambda}) = \big(\mathbf{x} + \sigma_{T-t}^2\, s_{\boldsymbol{\theta}}(\mathbf{x}, t, \boldsymbol{\lambda})\big) / \alpha_{T-t}$ is Tweedie's estimate of the posterior mean $\mathbb{E}[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \mathbf{x}, \boldsymbol{\lambda}]$, and the expectation in (PDI-D) is estimated over an ensemble of samples that share one multiplier. Since $\widehat{\mathbf{y}}_0(\mathbf{x}_t, \boldsymbol{\lambda}_t)$ reuses the score evaluation $s_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}_t)$ of (PDI-P), each PDI step requires a single score evaluation, as in fixed-multiplier sampling, plus the evaluation of the constraints $\mathbf{f}$. In practice, we run several chains in parallel, each with its own multiplier, and this preserves the average-constraint interpretation while adding robustness to perturbations in the final multiplier (see Algorithm 2).

A well-trained score model $s_{\boldsymbol{\theta}}$ represents the entire Gibbs family by approximating the $\boldsymbol{\lambda}$-conditional Gibbs scores, $s_{\boldsymbol{\theta}}(\mathbf{x}, t, \boldsymbol{\lambda}) \approx \nabla \log q_{T-t}(\mathbf{x}|\boldsymbol{\lambda})$. It is also conditioned on the problem instance $\mathcal{G}$, which we suppress from the notation. Conditioning diffusion models on side information is standard practice (Dhariwal and Nichol, 2021; Ho and Salimans, 2022; Rombach et al., 2022). Conditioning on $\boldsymbol{\lambda}$ is particularly benign, as $\boldsymbol{\lambda}$ enters the energy $E(\cdot, \boldsymbol{\lambda})$ linearly, and the score field varies smoothly with $\boldsymbol{\lambda}$ at every noise level (Lemma 8). Unlike standard diffusion models, however, the network cannot be trained on noisy data samples because the samples from the Gibbs targets are unavailable. Instead, its regression targets are estimated from the energy alone by reweighting Gaussian candidates for the clean sample using the Gibbs density (Akhound-Sadegh et al., 2024; Huang et al., 2024b). These targets are evaluated at the noisy samples and the relevant multipliers visited by PDI rollouts; Appendix B.1 details the training pipeline. In our experiments, we refer to PDI with the trained network as PDI-Net, whereas the training-free variant, which uses these Monte Carlo estimates directly in place of $s_{\boldsymbol{\theta}}$, is called PDI-MC.

4 Convergence Analysis

Two aspects of PDI set it apart from standard dual ascent and diffusion sampling. Firstly, the dual step in (PDI-D) is not an exact, unbiased dual ascent step. By (5), the dual gradient is the expected constraint value under $\mu_{\boldsymbol{\lambda}_t}^\dagger$. In contrast, (PDI-D) evaluates $\mathbf{f}$ at Tweedie estimates, which introduces bias from the nonlinearity of $\mathbf{f}$, from sampling p_t rather than $q_{T-t}^{\boldsymbol{\lambda}_t}$, and from score errors. Theorem 1 shows that this bias does not prevent convergence, but it determines how close the dual iterates get to $\boldsymbol{\lambda}^*$. In particular, the dual mismatch bias of the early steps, where the Tweedie estimates are coarsest, decays geometrically.

Secondly, because the multiplier changes along the trajectory, the marginal law p_t generated by PDI is generally not equal to $q_{T-t}(\cdot|\boldsymbol{\lambda}_t)$. The latter only defines the local fixed-multiplier score field used

at that step. In other words, PDI is a path-dependent reverse process rather than the exact reversal of any single forward process in (6). Theorem 2 nevertheless compares p_T with the output of an oracle sampler, which knows $\boldsymbol{\lambda}^*$ and keeps it fixed throughout, and bounds their terminal KL-divergence by the score error and the dual mismatch accumulated along the trajectory.

The PDI dual update is biased relative to exact dual ascent because it is evaluated along the evolving reverse process using Tweedie estimates and an approximate conditional score. This bias combines four effects analyzed separately in Appendix C.2: the Tweedie posterior-mean error, the marginal mismatch induced by the evolving multiplier, the reverse-sampling error of the exact-score sampler, and the accumulated score approximation error. We denote their aggregate effect at step s by $\mathbf{b}_s$. In Theorem 1, we establish convergence of the last dual iterate to a neighborhood of the optimal multiplier in the presence of this bias.

Theorem 1 (Dual Convergence)**.** *For a sequence of $\{\boldsymbol{\lambda}_t\}$ generated by PDI, it holds under Assumptions 2–6 that, for $\eta_t = \eta < \frac{2}{\vartheta}, \forall t$, and $r := 1 - \frac{\eta\vartheta}{2}$, the last-iterate multiplier satisfies*

$$\|\boldsymbol{\lambda}_T - \boldsymbol{\lambda}^*\|^2 \;\le\; r^T\|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^*\|^2 + \frac{2\eta}{\vartheta}R^2 + \frac{2\eta}{\vartheta}\sum_{s<T} r^{T-1-s}\|\mathbf{b}_s\|^2. \tag{8}$$

The proof is provided in Appendix C.3.1. The bound reveals a favorable interaction between the diffusion trajectory and the dual recursion. Errors incurred early in denoising, when the Tweedie estimates are coarsest, are strongly discounted by the geometric factor r^{T-1-s}. Conversely, the later updates receive larger weights precisely as the Tweedie posterior-mean and score-parameterization errors decrease toward the data end. Consequently, large early errors are progressively forgotten, and the convergence region is governed primarily by the dual step size and the residual bias of the later updates. We additionally derive an $O(T^{-1/2})$ time-average dual-gap bound for $\eta = \kappa/\sqrt{T}$ in Appendix C.3.2, which is used subsequently to establish the primal convergence result.

Theorem 2 (Primal Convergence)**.** *Let $\widetilde{p}_T^{\boldsymbol{\lambda}}$ be the output law of* (7) *with exact scores, fixed multiplier $\boldsymbol{\lambda}$, and the initialization of PDI. Under Assumptions 2– 7, and for $\eta = \kappa/\sqrt{T}$, the KL divergence between the terminal distribution of PDI and the oracle's output distribution satisfies*

$$D_{\mathrm{KL}}\Big(p_T\|\widetilde{p}_T^{\boldsymbol{\lambda}^*}\Big) \le O\bigg(\frac{1}{\sqrt{T}}\bigg) + O\bigg(\frac{1}{T}\sum_{t=0}^{T-1}\chi_t^2 + \delta_t^2\bigg) + O\big(\varepsilon^2\big)\,, \tag{9}$$

where χ_t^2 and δ_t^2 are the Tweedie and reverse-sampling errors (Lemmas 2 and 3), and ε is the score approximation error.

The proof is in Appendix C.4. The theorem shows that primal accuracy depends on the *trajectory-wide* quality of the dual updates, rather than only on the final multiplier. The $O(T^{-1/2})$ term reflects the average dual suboptimality accumulated along inference, while the remaining terms quantify how imperfect Tweedie estimates and score approximation limit the attainable accuracy. In particular, errors that are large only over a small portion of the reverse trajectory are averaged out rather than determining the terminal distribution by themselves. The oracle's own error with respect to μ^* stems only from the discretization and initialization, and Corollary 1 turns both into an approximate feasibility in expectation statement.

5 Numerical Results

We provide numerical evidence in three stochastic optimization problems: constrained MoG, wireless resource allocation, and portfolio management. More details are provided in Appendices E, F, and G.

5.1 Mixture of Gaussians

We first evaluate our approach on sampling from a weighted mixture of Gaussians in $\mathbb{R}^d$, $f_0(\mathbf{x}) = -\log\sum_{k=1}^K w_k\,\mathcal{N}\big(\mathbf{x};\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\big)$, truncated to a polytope $\{\mathbf{x}:\mathbf{A}^\top\mathbf{x}\preceq\mathbf{b}\}$. The optimization problem is

$$P^*_{\text{MoG}} \;=\; \min_{\mu(\mathbf{x})}\; \mathbb{E}_{\mathbf{x}\sim\mu}\big[f_0(\mathbf{x})\big] - \beta\mathcal{H}(\mu) \quad \text{s.t.} \quad \mathbb{E}_{\mathbf{x}\sim\mu}\big[\mathbf{A}^\top\mathbf{x}-\mathbf{b}\big] \;\preceq\; \mathbf{0}, \tag{10}$$

i.e., the sampler must produce a distribution that concentrates on high-density regions of the mixture while satisfying M linear inequality constraints in expectation.

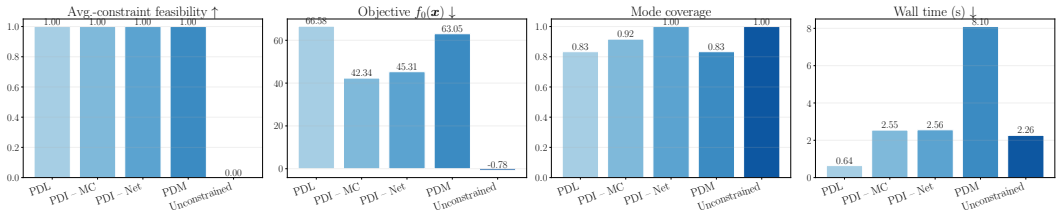

Figure 2: **Mixture of Gaussians** ($K = 12$ modes with $M = 10$ constraints in a 30-dimensional space). While PDI, PDL and PDM maintain full feasibility, PDI exhibits the best objective and mode diversity.

Table 1: **Baseline comparisons.** PDI achieves a near-feasible performance and provides the best balance between mean rates and constraint violations in the wireless power allocation problem ($r_{\min} = 0.6$).

Method	Mean rates (↑)	5th%ile rate (↑)	1st%ile rate (↑)	Mean viol. (↓)	Sum viol. (↓)	Feas. % (↑)
Unconstrained	2.957±0.123	0.239±0.052	0.081±0.037	0.0421±0.0078	8.43±1.57	85.6±2.4
PDI–Net	**2.803**±0.132	**0.613**±0.042	0.424±0.082	**0.0062**±0.0027	**1.25**±0.54	95.5±1.5
PDI–Net (warm-up)	2.578±0.132	**0.670**±0.055	0.429±0.102	**0.0056**±0.0026	**1.12**±0.51	96.7±1.4
PDI–MC	**2.839**±0.130	**0.614**±0.052	0.355±0.120	0.0085±0.0037	1.71±0.75	95.5±1.5
PDL	2.635±0.141	**0.641**±0.057	0.405±0.106	0.0068±0.0031	1.36±0.63	96.2±1.5
PDM	2.591±0.157	0.510±0.082	0.165±0.107	0.0180±0.0065	3.61±1.29	91.9±2.7
DPS	2.731±0.138	0.461±0.071	0.184±0.082	0.0198±0.0060	3.96±1.19	90.8±2.5
ST	2.482±0.134	**0.825**±0.060	**0.666**±0.071	**0.0007**±0.0008	**0.14**±0.16	99.3±0.6

Baselines. We compare our approach to two baselines: i) a projected diffusion model (PDM) (Christopher et al., 2024), and ii) primal-dual Langevin (PDL) dynamics (Chamon et al., 2024). For PDM, we reuse our trained model with $\boldsymbol{\lambda} = \mathbf{0}$ as the base model that samples from the unconstrained MoG and performs a Cimmino parallel projection (Sloboda, 1991) onto the polytope at each step.

Results. As shown in Figure 2, all constrained methods achieve the same constraint feasibility. However, our method (PDI-Net and PDI-MC) obtains a better objective value than the other constrained baselines. This is because the two baselines enforce the constraints on every generated sample, whereas our formulation only requires the constraints to hold in expectation. Pointwise enforcement is therefore more conservative and can discard useful samples for optimality that violate the constraints individually but could be offset by other samples in the distribution. We also include an ablation on the dual variable through the unconstrained baseline, which runs our model with $\boldsymbol{\lambda}$ fixed to zero at all time steps. As the figure shows, the unconstrained model ignores the constraints entirely and achieves the lowest objective. This performance confirms that the constrained behavior is not merely a consequence of the training algorithm.

5.2 Wireless Resource Allocation

For a given network state $\mathbf{H}$ with $N = 200$ users, the goal is to maximize the ergodic sum-rate while satisfying a minimum ergodic rate constraint for each user. That is,

$$P^*_{\text{pc}} \;=\; \max_{\mu(\mathbf{x})} \; \mathbf{1}^\top \mathbf{r}(\mu, \mathbf{H}) + \beta \mathcal{H}(\mu), \quad \text{s.t.} \quad \mathbf{r}(\mu, \mathbf{H}) \geq \mathbf{1} \cdot r_{\min}, \tag{11}$$

where μ denotes the distribution of power allocations, and $r_{\min}$ is the minimum ergodic rate requirement. The optimal solution is rarely a degenerate policy. Since users compete for the communication channels and cause interference, activating all users simultaneously is typically suboptimal and may prevent the rate constraints from being satisfied. Instead, the optimal distribution often corresponds to a multimodal switching policy, in which different subsets of users transmit at different times while meeting their rate requirements in expectation.

Baselines. In addition to PDL and PDM, we include two other diffusion-based baselines: i) diffusion posterior sampling (DPS) (Chung et al., 2023), and ii) supervised training (ST) with expert data (Uslu et al., 2026a). The expert data are generated by a primal-dual method run for the deterministic version of (11) and its resulting trajectories (after a transient period) are collected to train the diffusion model.

Results. Table 1 shows that PDI achieves the best overall balance between mean rates (objective) and feasibility represented by the 5th-percentile rate and mean and sum violations. The 1st-percentile users, however, are harder to correct by the dual variables and suffer under all methods, except for ST.

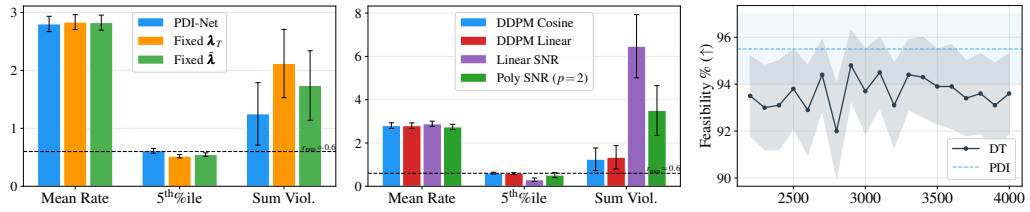

Figure 3: **Ablation.** Comparisons between our base PDI-Net model, plotted in blue, and (left) a PDI-Net while setting $\boldsymbol{\lambda}_t$ to a fixed value along the denoising trajectory, (middle) a PDI-Net with different noise schedules, and (right) DT checkpoints corresponding to different training epochs and different estimates of the dual variable.

Similarly to the MoG experiment, PDL and PDM achieve lower mean rates (worse objectives) with no substantial gains in feasibility because they sacrifice optimality for forcing pointwise constraints. Moreover, ST generates feasible solutions at the expense of optimality. Because ST is trained only to imitate expert trajectories, it is agnostic to the dual structure of the constrained distribution and cannot explicitly balance utility against feasibility during sampling. Lastly, instead of using the untrained score model to generate rollouts during the early epochs of training, we initialize these rollouts with the ST model. We observe that the trained PDI-Net with ST warm-up inherits the same bias toward feasibility, achieving a higher, feasible 1st-percentile rate. However, this comes at the expense of the objective value, which decreases from 2.80 to 2.58.

Ablation. Figure 3 (left) evaluates the effect of keeping the dual variable fixed during inference. We run the score model with two fixed values: the final dual iterate $\boldsymbol{\lambda}_T$ and the time-average dual variable $\bar{\boldsymbol{\lambda}}$, both computed from our base experiment. The results suggest that the distribution generated by the full PDI dynamics is not equivalent to the Gibbs distribution associated with either fixed value. Instead, it behaves like a mixture induced by the changing dual variable, and this mixture exhibits better feasibility. Among the two values, the Gibbs distribution with $\bar{\boldsymbol{\lambda}}$ provides a better approximation to the PDI-generated distribution than the one associated with the final iterate. To evaluate the role of the noise schedule, we report the performance metrics of four schedules in Figure 3 (middle): the DDPM cosine used in our base experiment, the DDPM linear schedule and the linear and polynomial SNR schedules. The latter two schedules fail to generate feasible samples.

Dual training. To ablate the role of inference-time dual updates, we train a score model $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t)$ that does not take the dual variable as input. During training, we still update the dual variables using a primal-dual algorithm, so the score model is trained against targets induced by the evolving dual variables. At inference, there are no more dual updates and the trained model therefore acts as a fixed sampling policy that is intended to sample from the optimal distributions. We refer to this approach as dual training (DT) and the training algorithm is provided in Appendix D.

Figures 3 (right) and 8 compare DT checkpoints along the training trajectory with PDI. Although some DT checkpoints reach dual variables close to those produced by PDI, their generated samples exhibit lower tail rates and lower feasibility. This indicates matching the dual variables alone is not sufficient. Since DT does not condition the score model on the current multiplier, a checkpoint trained over a changing dual trajectory need not represent the score field associated with its current dual value. Instead, it reflects the accumulated effect of the preceding training trajectory. We further verify this in Appendix F.3 by freezing DT multipliers and continuing score training, which substantially improves performance but requires a much larger training budget. PDI avoids this limitation by keeping the dual variable as an explicit state during inference. Thus, when the dual update increases the penalty on violated constraints, the penalty is reflected directly in the denoising dynamics.

Out-of-distribution constraints. The dual state conditioning in our PDI score model also benefits OOD performance. In Figure 4, we plot the sum violation and feasibility percentage under different constraint

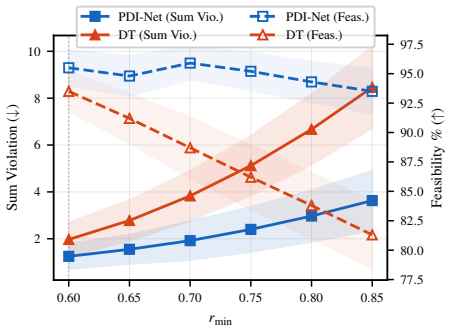

Figure 4: **Out-of-distribution constraints.** The models are trained under $r_{\min} = 0.6$ and tested under different values of constraint levels. PDI degrades more gracefully than DT, showing more robustness to distribution shifts.

Table 2: **Constrained portfolio sampling**, $N = 500$ assets, 521 constraints, 40 instances $\times$ 1024 samples. Reported are 95th%ile and max violations over constraints within each family; $\pm$ std across instances. **Bold** = best, underlined = second best among constrained samplers. Time is seconds per instance (RTX 3090).

	Variance Viol. ↓		Sector upper Viol. ↓		Sector lower Viol. ↓		N_{eff} viol. ↓	Return ↑	Time (s)
Method	95th%ile	max	95th%ile	max	95th%ile	max			
Unconstrained	0.0013 $\pm$0.0006	0.0233 $\pm$0.0247	0.0012 $\pm$0.0038	0.0021 $\pm$0.0069	0.0002 $\pm$0.0008	0.0004 $\pm$0.0015	0.0000 $\pm$0.0000	0.1993 $\pm$0.0228	37.9
PDI-MC (ours)	**0.0000 $\pm$0.0000**	**0.0000 $\pm$0.0000**	0.0046 $\pm$0.0067	0.0074 $\pm$0.0105	**0.0069 $\pm$0.0069**	**0.0098 $\pm$0.0095**	1.8662 $\pm$5.7070	**0.1773 $\pm$0.0188**	29.2
PDI-Net (ours)	**0.0000 $\pm$0.0000**	**0.0000 $\pm$0.0000**	0.0041 $\pm$0.0060	0.0060 $\pm$0.0082	0.0281 $\pm$0.0095	0.0350 $\pm$0.0097	**0.0000 $\pm$0.0000**	0.1537 $\pm$0.0180	37.7
PDL	**0.0000 $\pm$0.0000**	0.0008 $\pm$0.0008	0.0006 $\pm$0.0014	0.0011 $\pm$0.0024	0.0415 $\pm$0.0028	0.0451 $\pm$0.0029	0.1666 $\pm$0.8680	0.1334 $\pm$0.0162	0.9
PDM	**0.0000 $\pm$0.0000**	**0.0000 $\pm$0.0000**	**0.0004 $\pm$0.0011**	**0.0006 $\pm$0.0017**	0.0480 $\pm$0.0035	0.0510 $\pm$0.0037	**0.0000 $\pm$0.0000**	0.1259 $\pm$0.0160	88.3
DPS	0.0016 $\pm$0.0004	0.0268 $\pm$0.0337	0.0006 $\pm$0.0019	0.0011 $\pm$0.0034	0.0176 $\pm$0.0060	0.0209 $\pm$0.0065	**0.0000 $\pm$0.0000**	0.1625 $\pm$0.0172	38.2
Equal weight	0.0001 $\pm$0.0001	0.0007 $\pm$0.0007	0.0000 $\pm$0.0000	0.0000 $\pm$0.0000	0.0000 $\pm$0.0000	0.0000 $\pm$0.0000	440.0000 $\pm$0.0000	0.2207 $\pm$0.0235	<0.1

levels $r_{\min}$, with $r_{\min} = 0.6$ corresponding to the in-distribution setting. We observe that PDI is more robust to the shift in $r_{\min}$. This is because the score function $\nabla \log q_{T-t}(\mathbf{x}_t \mid \boldsymbol{\lambda}_t)$ does not depend on $r_{\min}$ directly. Instead, changing $r_{\min}$ affects the constraint residual and, therefore, induces a different trajectory of dual variables. Since PDI is trained with both exploitation and exploration over $\boldsymbol{\lambda}$, the score model learns a family of score fields over a range of in-distribution dual values. Thus, as long as the dual trajectory induced by the new $r_{\min}$ remains close to this learned range, PDI can adapt to the new OOD constraints.

5.3 Constrained Portfolio Sampling

We consider constrained portfolio optimization over $N = 500$ assets organized into 10 sectors, with returns generated from a latent factor model. Each portfolio is represented by weights on the probability simplex, and the objective is to maximize expected return over $1,000$ return scenarios. We impose a large and heterogeneous set of constraints covering per-asset variance contributions, sector exposures, and portfolio diversification. In total, the problem contains 521 inequality constraints. The average constraints allow the sampler to trade risk across different portfolio samples, admitting occasional high-risk, high-return allocations while ensuring that the overall exposure of each asset remains within its prescribed budget in expectation.

Results. Table 2 shows that PDI achieves a strong return–constraint tradeoff across the different constraint families. Both PDI variants satisfy the variance constraints essentially exactly, while maintaining small violations of the sector constraints. PDI-MC achieves the highest return among the constrained diffusion-based methods, whereas PDI-Net provides a more balanced solution across the different constraint families. In particular, PDI-Net eliminates the diversification violation observed with PDI-MC while retaining a higher return than PDL and PDM. The baselines exhibit different forms of conservatism. PDL and PDM control the variance and upper-sector constraints, but incur larger violations of the lower-sector constraints and obtain lower returns. DPS preserves a competitive return, but does so at the cost of noticeably larger variance violations. These results show that the constraints cannot be treated independently. Enforcing one family more aggressively can shift the violation to another family or reduce the achievable return, while PDI learns a distribution that balances these competing requirements jointly. Additional ablations are provided in Appendix G.

6 Conclusions

We introduced a primal–dual inference (PDI) framework that enforces distributional constraints by coupling a reverse diffusion process with dual ascent dynamics. We trained a single dual-variable-conditioned score network that serves the entire family of Gibbs targets encountered along the dual trajectory. At inference, PDI generates samples through a path-dependent sequence of Gibbs score fields indexed by the evolving multipliers, avoiding the need to first estimate and freeze an optimal dual variable. We showed that the last dual iterate converges to a neighborhood of the optimal dual multiplier and that PDI tracks an oracle sampler that knows it. We validated our framework and methodology on constrained Gaussian sampling, wireless power control, and portfolio management. Extending PDI to adaptive dual step sizes, chance-constrained problems, and larger-scale setups are promising directions for future work. Our work has some limitations worth noting. PDI requires problem-specific tuning of the initial dual multiplier distribution used during training and the entropy regularization strength β, for which principled selection rules remain an open question.

ACKNOWLEDGMENT

Generative AI tools were used to assist with proofreading and improving the grammar and clarity of the manuscript. They were also used to support experimental implementation and to assist in drafting portions of the experimental appendices. All technical content, experimental results, and final text were reviewed and edited by the authors.

REFERENCES

Tara Akhound-Sadegh, Jarrid Rector-Brooks, Avishek Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Alexander Tong. Iterated denoising energy matching for sampling from Boltzmann densities. In *International Conference on Machine Learning (ICML)*, 2024.

Brian D. O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Sravan Kumar Ankireddy, Heasung Kim, and Hyeji Kim. Residual diffusion models for joint source channel coding of MIMO CSI. In *2025 59th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2025.

Marius Arvinte and Jonathan I. Tamir. Mimo channel estimation using score-based generative models. *IEEE Transactions on Wireless Communications*, 22(6):3698–3713, 2023. doi: 10.1109/TWC.2022.3220784.

Amirhassan Babazadeh Darabi and Sinem Coleri. Diffusion model based resource allocation strategy in ultra-reliable wireless networked control systems. *IEEE Communications Letters*, 29(1):85–89, 2025. doi: 10.1109/LCOMM.2024.3499745.

Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-Linear Convergence Bounds for Diffusion Models via Stochastic Localization. In *International Conference on Learning Representations*, 2024.

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Luiz F Chamon, Mohammad R Karimi, and Anna Korba. Constrained sampling with primal-dual langevin monte carlo. *Advances in Neural Information Processing Systems*, 37:29285–29323, 2024.

Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099 – 1135, 2018. doi: 10.1214/17-AAP1326. URL https://doi.org/10.1214/17-AAP1326.

Junhua Chen, Lorenz Richter, Julius Berner, Denis Blessing, Gerhard Neumann, and Anima Anandkumar. Sequential controlled langevin diffusions, 2024. URL https://arxiv.org/abs/2412.07081.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R. Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *International Conference on Learning Representations*, 2023.

Himanshu Choudhary, Arishi Orra, and Manoj Thakur. Diffusion-augmented reinforcement learning for robust portfolio optimization under stress scenarios, 2025.

Jacob K Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. *Advances in Neural Information Processing Systems*, 37:89307–89333, 2024.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Arnak S. Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3): 651–676, 2017. doi: 10.1111/rssb.12183.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. *Advances in neural information processing systems*, 35:2406–2422, 2022.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.

Paul Dupuis and Richard S. Ellis. *A Weak Convergence Approach to the Theory of Large Deviations*. Wiley Series in Probability and Statistics. John Wiley & Sons, New York, 1997.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Nic Fishman, Leo Klarner, Emile Mathieu, Michael Hutchinson, and Valentin De Bortoli. Metropolis sampling for constrained diffusion models. *Advances in Neural Information Processing Systems*, 36:62296–62331, 2023.

Xuefeng Gao, Mengying He, and Xuedong He. Factor-based conditional diffusion model for portfolio optimization. In *NeurIPS 2025 Workshop on Generative AI in Finance*, 2025. arXiv:2509.22088.

Jia Guo, Xiaoxia Xu, Yuanwei Liu, and Arumugam Nallanathan. Diffusion model for multiple antenna communication. *IEEE Communications Magazine*, 63(10):44–50, 2025. doi: 10.1109/MCOM.001.2400766.

Yingqing Guo, Hui Yuan, Yukang Yang, Minshuo Chen, and Mengdi Wang. Gradient guidance for diffusion models: An optimization perspective. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 90736–90770. Curran Associates, Inc., 2024. doi: 10.52202/079017-2881. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a5059a9a389ccc76da85760ea79490d8-Paper-Conference.pdf.

Zhengyi Guo, Wenpin Tang, and Renyuan Xu. Conditional diffusion guidance under hard constraint: A stochastic analysis approach. *arXiv preprint arXiv:2602.05533*, 2026.

Sven Gutjahr, Riccardo De Santi, Luca Schaufelberger, Kjell Jorner, and Andreas Krause. Constrained flow optimization via sequential fine tuning for molecular design. *arXiv preprint arXiv:2605.30610*, 2026.

Zhizhou He, Fabien Héliot, and Yi Ma. Deterministic score-based diffusion model for channel estimation in ris-assisted mimo systems. In *2025 IEEE 26th International Workshop on Signal Processing and Artificial Intelligence for Wireless Communications (SPAWC)*, pages 1–5, 2025. doi: 10.1109/SPAWC66079.2025.11143296.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David Fleet. Video diffusion models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 8633–8646. Curran Associates, Inc., 2022.

William Huang, Yifeng Jiang, Tom Van Wouwe, and C. Karen Liu. Constrained diffusion with trust sampling. *Advances in Neural Information Processing Systems*, 37, 2024a.

Xunpeng Huang, Hanze Dong, Yifan Hao, Yi-An Ma, and Tong Zhang. Reverse diffusion Monte Carlo. In *International Conference on Learning Representations*, 2024b.

Tong Hui, Xiao Tang, Yichen Wang, Qinghe Du, Dusit Niyato, and Zhu Han. Channel-aware conditional diffusion model for secure mu-miso communications. *IEEE Transactions on Vehicular Technology*, 75(7):15287–15292, 2026. doi: 10.1109/TVT.2026.3662745.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

Chen Jin and Ankush Agarwal. Forecasting implied volatility surface with generative diffusion models, 2025. URL `https://arxiv.org/abs/2511.07571`.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

Shervin Khalafi, Dongsheng Ding, and Alejandro Ribeiro. Constrained diffusion models via dual training. *Advances in Neural Information Processing Systems*, 37:26543–26576, 2024.

Shervin Khalafi, Ignacio Hounie, Dongsheng Ding, and Alejandro Ribeiro. Composition and alignment of diffusion models using constrained learning. In *Advances in Neural Information Processing Systems*, volume 38, pages 18629–18676, 2025. doi: 10.52202/085713-0628.

Heasung Kim, Taekyun Lee, Hyeji Kim, Gustavo De Veciana, Mohamed Amine Arfaoui, Asil Koc, Phil Pietraski, Guodong Zhang, and John Kaewell. Generative diffusion model-based compression of mimo csi. In *ICC 2025 - IEEE International Conference on Communications*, pages 6323–6328, 2025. doi: 10.1109/ICC52391.2025.11161629.

Lingkai Kong, Yuanqi Du, Wenhao Mu, Kirill Neklyudov, Valentin De Bortoli, Dongxia Wu, Haorui Wang, Aaron M. Ferber, Yi-An Ma, Carla P. Gomes, and Chao Zhang. Diffusion models as constrained samplers for optimization with unknown constraints. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 258 of *Proceedings of Machine Learning Research*, pages 4582–4590. PMLR, 2025a.

Lingkai Kong, Haichuan Wang, Yuqi Pan, Cheol Woo Kim, Mingxiao Song, Alayna Nguyen, Tonghan Wang, Haifeng Xu, and Milind Tambe. Robust optimization with diffusion models for green security. In Silvia Chiappa and Sara Magliacane, editors, *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pages 2325–2344. PMLR, 21–25 Jul 2025b.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

Taekyun Lee, Juseong Park, Hyeji Kim, and Jeffrey G. Andrews. Generating high dimensional user-specific wireless channels using diffusion models. *IEEE Transactions on Wireless Communications*, 25:2907–2921, 2026. doi: 10.1109/TWC.2025.3600286.

Guan-Horng Liu, Tianrong Chen, Evangelos Theodorou, and Molei Tao. Mirror diffusion models for constrained and watermarked generation. *Advances in Neural Information Processing Systems*, 36: 42898–42917, 2023.

Aaron Lou and Stefano Ermon. Reflected diffusion models. In *International Conference on Machine Learning*, pages 22675–22701. PMLR, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022.

Georg Menz and André Schlichting. Poincaré and logarithmic Sobolev inequalities by decomposition of the energy landscape. *The Annals of Probability*, 42(5):1809–1884, 2014. doi: 10.1214/14-AOP908.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models, 2022. URL https://arxiv.org/abs/2112.10741.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021.

Lorenz Richter and Julius Berner. Improved sampling via learned diffusions. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=h4pNROsO06.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01042.

Marcin Sendera, Minsu Kim, Sarthak Mittal, Pablo Lemos, Luca Scimeca, Jarrid Rector-Brooks, Alexandre Adam, Yoshua Bengio, and Nikolay Malkin. Improved off-policy training of diffusion samplers. *Advances in Neural Information Processing Systems*, 37:81016–81045, 2024.

Fridrich Sloboda. A projection method of the cimmino type for linear algebraic systems. *Parallel Computing*, 17(4):435–442, 1991. ISSN 0167-8191. doi: 10.1016/S0167-8191(05)80146-2.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

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

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

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.

Zhiqing Sun and Yiming Yang. DIFUSCO: Graph-based diffusion solvers for combinatorial optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=JV8Ff0lgVV.

Tomonori Takahashi and Takayuki Mizuno. Generation of synthetic financial time series by diffusion models. *Quantitative Finance*, 25(10):1507–1516, 2025.

Nilay Tiwari. Generative diffusion model for risk-neutral derivative pricing, 2026. URL https://arxiv.org/abs/2603.20582.

Yiğit Berkay Uslu, Samar Hadou, Shirin Saeedi Bidokhti, and Alejandro Ribeiro. Generative diffusion models for resource allocation in wireless networks. In *2025 IEEE 10th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 201–205. IEEE, 2025.

Yiğit Berkay Uslu, Samar Hadou, Shirin Saeedi Bidokhti, and Alejandro Ribeiro. Graph signal diffusion models for wireless resource allocation, 2026a. URL https://arxiv.org/abs/2604.05175.

Yiğit Berkay Uslu, Navid NaderiAlizadeh, Mark Eisen, and Alejandro Ribeiro. Fast state-augmented learning for wireless resource allocation with dual variable regression. *IEEE Transactions on Signal Processing*, 2026b.

Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=8pvnfTAbu1f.

Jiaqi Wen and Jianyi Yang. Distributionally robust optimization via generative ambiguity modeling. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=q67t0gFrdY.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: a stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022.

Xinren Zhang and Jiadong Yu. Improve the training efficiency of drl for wireless communication resource allocation: The role of generative diffusion models. *IEEE Transactions on Wireless Communications*, 25:11593–11608, 2026. doi: 10.1109/TWC.2026.3660250.

Nicolas Zilberstein, Ananthram Swami, and Santiago Segarra. Joint channel estimation and data detection in massive Mimo systems based on diffusion models. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13291–13295, 2024. doi: 10.1109/ICASSP48485.2024.10446413.

A Extended Related Work

Diffusion models and score-based generative modeling. Score-based diffusion models learn a stochastic reversal of a progressive noising process by matching the gradients of the log-density $\nabla_x \log p(x)$, a.k.a. the score function, for noisy observations (Hyvärinen, 2005). Noise-conditional score networks (Song and Ermon, 2019) train a single network across noise levels and sample via annealed Langevin dynamics, while denoising diffusion probabilistic models (Ho et al., 2020) frame generation as iterative Gaussian denoising guided by a simplified ELBO. These formulations are unified in (Song et al., 2021b) as instances of a common SDE, whose time reversal yields a generative process; and admits an equivalent deterministic probability-flow ODE that enables exact likelihood computation and has motivated accelerated sampling schemes such as denoising diffusion implicit models (DDIMs) (Song et al., 2021a; Nichol and Dhariwal, 2021; Lu et al., 2022; Karras et al., 2022).

Energy-based models and sampling from Gibbs targets. Energy-based models define distributions of the form $p(x) \propto \exp(-E(x))$, whose normalizing constant is generally intractable. Score-based methods sidestep this because the score $\nabla_x \log p(x) = -\nabla_x E(x)$ is independent of this constant. This motivated the use of diffusion as a sampler from unnormalized Gibbs targets by learning a controlled SDE, whose terminal law matches the target, e.g., (Zhang and Chen, 2022; Vargas et al., 2023; Berner et al., 2024; Richter and Berner, 2024; Chen et al., 2024; Domingo-Enrich et al., 2024; Sendera et al., 2024). In our setting, the Lagrangian minimizers are also Gibbs distributions whose energies are linear in the dual variables. Crucially, the energies are not fixed as they evolve during inference through dual ascent. Furthermore, the score network is trained over a distribution of dual variables, or equivalently, a distribution of target energies, rather than for a single one.

Guided and constrained diffusion sampling. Guidance techniques steer reverse diffusion processes toward improved fidelity or reward/constraint alignment more generally. They do so by modifying the score entering Tweedie's denoising formula (Efron, 2011), which expresses the MMSE-estimate of a clean signal (posterior mean) given a noisy signal observation in terms of the score function. Notably, classifier guidance (Dhariwal and Nichol, 2021) perturbs the predicted score with the gradient of a noise-aware classifier, trading diversity for fidelity. Classifier-free guidance (Ho and Salimans, 2022) removes the external classifier, instead interpolating between conditional and unconditional modes of a shared model during inference. Subsequent variants, such as CLIP-based guidance (Nichol et al., 2022), underpin most state-of-the-art diffusion systems.

Beyond guidance, an extensive body of work enforces domain-feasibility constraints on the diffusion process directly, e.g., boundary reflections (Lou and Ermon, 2023), mirror diffusion (Liu et al., 2023), Riemannian score-based models (De Bortoli et al., 2022), and log-barrier constructions (Fishman et al., 2023). A separate line of works target constraint-aware inference. In (Christopher et al., 2024), projected diffusions that apply feasibility projections after each denoising step are proposed, while a trust sampling approach (Huang et al., 2024a) reformulates each reverse step as a separate constrained optimization problem.

Closest to this paper are (Khalafi et al., 2024; 2025; Chamon et al., 2024), which adopt a constrained learning formulation for average (expectation) constraints. Specifically, (Khalafi et al., 2024) shows that the optimal KL-constrained distribution is an entropy-tilted mixture of reference distributions, enforced through a primal–dual training scheme that pairs an outer-loop dual ascent with inner-loop Lagrangian score matching, with (Khalafi et al., 2025) extending this to reward-based alignment and product/mixture compositions. Primal–dual Langevin Monte Carlo (PD-LMC) (Chamon et al., 2024) operates instead at sampling time and enforces equality and average constraints via gradient descent–ascent in Wasserstein space, without a learned generative model.

Our method shares the Lagrangian saddle-point structure of these works but differs in three aspects. (i) The dual search takes place within the reverse diffusion itself. Each denoising step is followed by a dual ascent step that evaluates the constraints at Tweedie estimates computed from the same score evaluation. In contrast, dual training performs this search in a separate outer loop that leaves a frozen sampler (Khalafi et al., 2024; 2025), and PD-LMC performs it along a Langevin chain whose accuracy depends on mixing (Chamon et al., 2024). (ii) A single score network conditioned on the multiplier and the problem instance represents the entire Gibbs family. It is trained on Monte Carlo score targets computed from the energy alone, at the multipliers visited by PDI rollouts. An unconditional score model trained against the evolving dual iterates instead reflects the history of those iterates rather

than the Gibbs target at its current multiplier. (iii) The target is the Lagrangian minimizer at the optimal multiplier, i.e., the Gibbs distribution $\mu^* = \mu^\dagger_{\boldsymbol{\lambda}^*}$ that tilts the reference (uniform) distribution as in the characterization of (Khalafi et al., 2024) and in the energy-based samplers discussed above. PDI reaches this target without estimating $\boldsymbol{\lambda}^*$ beforehand. Theorem 1 shows that the dual iterates converge to a neighborhood of $\boldsymbol{\lambda}^*$ despite the bias of the Tweedie-based dual updates. Theorem 2 shows that the output distribution tracks that of an oracle sampler that knows $\boldsymbol{\lambda}^*$, up to a floor set by the Tweedie, reverse-sampling, and score-approximation errors. Empirically, the path-dependent dynamics also improve feasibility over freezing the multiplier at its final or time-averaged value (Section 5.2).

Diffusion-enabled Stochastic Optimization. Pretrained diffusion models constrained on the data manifold can be steered toward downstream objectives through gradient-based guidance or fine-tuning (Guo et al., 2024), making them viable tools for learning-enabled optimization. Several works have exploited this connection between guidance with optimization gradients and regularized optimization. (Kong et al., 2025a) combined guided diffusion models and Langevin dynamics in a two-stage scheme that learns constrained samplers for optimization with unknown constraints. (Black et al., 2024) reinterpreted the denoising chain as a multi-step Markov Decision Process (MDP) and applied policy-gradient methods to fine-tune diffusion models on non-differentiable rewards. Beyond continuous settings, diffusion models have also advanced combinatorial optimization (Sun and Yang, 2023), and distributionally robust optimization (Wen and Yang, 2026; Kong et al., 2025b; Guo et al., 2026).

Diffusion models in wireless optimization and portfolio management. Diffusion models recast stochastic optimization as conditional sampling from learned solution distributions. This perspective is especially relevant to wireless communication systems, whose non-convex, constrained design problems typically admit optimal solutions that are probability distributions over optimization variables. E.g., (Arvinte and Tamir, 2023; Zilberstein et al., 2024; He et al., 2025) leveraged diffusion priors for score-based MIMO channel estimation, with subsequent works extending their use to CSI compression (Lee et al., 2026; Ankireddy et al., 2025; Kim et al., 2025), beamforming design (Guo et al., 2025; Hui et al., 2026), and resource allocation (Zhang and Yu, 2026; Babazadeh Darabi and Coleri, 2025; Uslu et al., 2025; 2026a). Specifically for power control, (Babazadeh Darabi and Coleri, 2025) trains DDPMs to generate optimal allocations conditioned on the channel state, while (Uslu et al., 2025; 2026a) proposes graph-signal diffusion models for matching primal-dual expert policies with near-optimal ergodic rates and cross-topology transferability.

Diffusion models have found great use in quantitative finance applications, where non-convex constraints and high-dimensional, heavy-tailed return distributions present analogous challenges to those encountered in wireless system design. E.g., (Takahashi and Mizuno, 2025) proposed wavelet-DDPMs for scenario and stylized facts generation. (Choudhary et al., 2025; Gao et al., 2025) tackled robust portfolio management with conditional diffusion models, with (Jin and Agarwal, 2025; Tiwari, 2026) enforcing arbitrage-free and risk-neutral diffusion dynamics for volatility forecasting.

B Proposed Algorithms

B.1 Score Network Training

The dual-conditioned score model $s_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}, \mathcal{G})$ is shared across problem instances $\mathcal{G}$, multipliers $\boldsymbol{\lambda}$, and diffusion times t. It is trained with the regression objective

$$\boldsymbol{\theta}^* \in \operatorname*{argmin}_{\boldsymbol{\theta}} \quad \mathbb{E}_{(\mathcal{G},\boldsymbol{\lambda},t,\mathbf{x}_t)\sim\rho}\Big[\omega(t)\,\big\|s_{\boldsymbol{\theta}}\big(\mathbf{x}_t, t, \boldsymbol{\lambda}, \mathcal{G}\big) - \nabla_{\mathbf{x}_t} \log q_{T-t}(\mathbf{x}_t|\boldsymbol{\lambda}, \mathcal{G})\big\|_2^2\Big], \tag{12}$$

where $\omega(t)$ is a time-dependent weighting function and ρ is a training distribution over instances, multipliers, diffusion times, and noisy samples. This objective differs from standard score matching in two ways.

Targets. Standard diffusion models regress onto the score of the Gaussian noising kernel around noised data samples, which requires samples of the target distribution. Here, samples of $\mu^\dagger_{\boldsymbol{\lambda}}$ are unavailable, and the regression target is the score of the noised Gibbs marginal itself. We estimate it from the energy with the Monte Carlo estimator of Lemma 1. Given $\mathbf{x}_t$, the estimator draws Gaussian

candidates for the clean sample and reweights them by the Gibbs density $e^{-E(\cdot,\boldsymbol{\lambda})}$, and the unknown partition function cancels in the normalization.

Inputs. Because each target is computed directly at its input, rather than through noised data samples, any training distribution ρ is admissible, and ρ only determines where the model is accurate. We draw $\mathbf{x}_t$ and $\boldsymbol{\lambda}$ from rollouts of PDI with the current model, which places accuracy on the states and multipliers encountered at inference. We complement them with multipliers drawn from an initial distribution and with small perturbations for coverage. The distribution of multipliers is not known in advance, since the dual trajectory itself induces it, so training alternates between collecting rollouts and fitting the model (Algorithm 1; see also the practical considerations below). Using the estimator of Lemma 1 directly in place of $s_{\boldsymbol{\theta}}$ in (PDI-P) yields the training-free variant PDI-MC.

Lemma 1 (MC estimator of the Gibbs score Akhound-Sadegh et al. (2024))**.** *For any $\mathbf{y}_\tau$, the score function of the Gibbs distribution can be estimated by a* biased *K_{MC} Monte Carlo estimator:*

$$\nabla_{\mathbf{y}_\tau} \log q_\tau(\mathbf{y}_\tau|\boldsymbol{\lambda}) \;\approx\; \nabla_{\mathbf{y}_\tau} \log \frac{1}{K_{MC}} \sum_{k=1}^{K_{MC}} \exp\left(- E\Big(\frac{\mathbf{y}_\tau + \sigma_\tau \boldsymbol{\epsilon}_k}{\alpha_\tau}, \boldsymbol{\lambda} \Big) \right), \tag{13}$$

where $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

We refer the reader to Akhound-Sadegh et al. (2024) for the proof.

Relation to guidance. Training a dual-conditioned model, rather than guiding a model of the unconstrained optimum $\mu^\dagger_{\mathbf{0}}$, is motivated by the following identity. By Proposition 1, $\mathrm{d}\mu^\dagger_{\boldsymbol{\lambda}}/\mathrm{d}\mu^\dagger_{\mathbf{0}} = e^{-\boldsymbol{\lambda}^\top \mathbf{f}/\beta}\, Z(\mathbf{0})/Z(\boldsymbol{\lambda})$. Hence,

$$q_\tau(\mathbf{x}|\boldsymbol{\lambda}) \;=\; \int \mathcal{N}\big(\mathbf{x}; \alpha_\tau \mathbf{y}, \sigma_\tau^2 \mathbf{I}\big)\, \mu^\dagger_{\boldsymbol{\lambda}}(\mathrm{d}\mathbf{y}) \;=\; \frac{Z(\mathbf{0})}{Z(\boldsymbol{\lambda})}\, q_\tau(\mathbf{x}|\mathbf{0})\, \mathbb{E}_{\mu^\dagger_{\mathbf{0}}}\Big[e^{-\boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{y}_0)/\beta} \,\Big|\, \mathbf{y}_\tau = \mathbf{x} \Big], \tag{14}$$

and taking $\nabla_{\mathbf{x}} \log$ of both sides yields the exact guidance identity

$$\nabla \log q_\tau(\mathbf{x}|\boldsymbol{\lambda}) \;=\; \nabla \log q_\tau(\mathbf{x}|\mathbf{0}) + \nabla_{\mathbf{x}} \log \mathbb{E}_{\mu^\dagger_{\mathbf{0}}}\Big[e^{-\boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{y}_0)/\beta} \,\Big|\, \mathbf{y}_\tau = \mathbf{x} \Big]. \tag{15}$$

The correction is a Doob h-transform of the unconstrained reverse process. Guidance methods such as DPS (Chung et al., 2023) replace the conditional expectation of the tilt by the tilt evaluated at the Tweedie mean $\widehat{\mathbf{y}}_0^{(\mathbf{0})}(\mathbf{x})$ of $\mu^\dagger_{\mathbf{0}}$. This yields the correction $-\beta^{-1}\nabla_{\mathbf{x}}\big[\boldsymbol{\lambda}^\top \mathbf{f}\big(\widehat{\mathbf{y}}_0^{(\mathbf{0})}(\mathbf{x})\big)\big]$. The approximation is exact when the posterior is a point mass, but its error grows with the posterior spread, i.e., at low SNR, and with $1/\beta$, since the tilt is then sharply nonlinear. In practice, guidance also replaces $\boldsymbol{\lambda}^*/\beta$ by a hand-tuned step size and penalizes each sample separately, as in our DPS baseline (128). This drives every sample toward feasibility rather than enforcing the constraints on average. PDI avoids both approximations: the dual-conditioned score represents (15) for every $\boldsymbol{\lambda}$, and dual ascent determines the weight.

B.2 Inference Algorithm

At inference, we run multiple chains in parallel, as described in Algorithm 2. Each chain carries one multiplier and an ensemble of samples, and the expectation in (PDI-D) is replaced by the ensemble average. A single score evaluation per step drives both the primal and the dual update. In addition to the projection $[\cdot]_+$, the implementation clips each multiplier at $\lambda_{\max}$, i.e., it projects onto the box $\boldsymbol{\Lambda} = [0, \lambda_{\max}]^d$ of Proposition 1, with $\lambda_{\max}$ reported with the hyperparameters of each experiment (Appendices E, F, and G).

Sensitivity to the multiplier. By Lemma 8, the noised Gibbs score is Lipschitz in the multiplier, $\|\nabla \log q_\tau(\mathbf{x}|\boldsymbol{\lambda}) - \nabla \log q_\tau(\mathbf{x}|\boldsymbol{\lambda}')\| \le \gamma_\tau \|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|$, with $\gamma_\tau = \frac{R\,\alpha_\tau}{\beta\,\sigma_\tau^2} \sup_{\mathbf{x},\boldsymbol{\lambda}} \|\mathrm{Cov}(\mathbf{y}_0 \mid \mathbf{y}_\tau = \mathbf{x}, \boldsymbol{\lambda})\|^{1/2}$. The posterior is supported on $\mathcal{X}$, so its covariance is at most $\mathrm{diam}(\mathcal{X})^2$ in operator norm, and

$$\gamma_\tau \;\le\; \frac{R\,\mathrm{diam}(\mathcal{X})}{\beta}\, \frac{\alpha_\tau}{\sigma_\tau^2} \;=\; \frac{R\,\mathrm{diam}(\mathcal{X})}{\beta}\, \frac{\sqrt{\mathrm{SNR}_\tau}}{\sigma_\tau}. \tag{16}$$

The bound vanishes as $\mathrm{SNR}_\tau \to 0$ and grows with the SNR. In the reverse step with exact scores, a multiplier error $\boldsymbol{\delta}$ at step t perturbs the drift by at most $b_{T-t}\gamma_{T-t}\|\boldsymbol{\delta}\|/\sqrt{a_{T-t}}$. Early steps are

Algorithm 1 Score Network Training

Require: Decay and noise schedule $a_{0:T}, b_{0:T}$, Perturbation params $(\rho_{\text{pert}}, \epsilon_{\mathbf{x}}, \epsilon_{\boldsymbol{\lambda}})$, exploitation fraction $\rho_{\text{exp}}(n)$ for all n

Initialize replay buffer $\mathcal{B} \leftarrow \emptyset$, score network $\mathbf{s}_{\boldsymbol{\theta}}$
for $n = 1, \ldots, N_{\text{outer}}$ **do**
  for each problem $\mathcal{G}$ in a batch $\mathcal{B}_{\text{train}}$ **do** ▷ Rollouts
    Sample $\{\mathbf{x}_0^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\}_i$, Initialize $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda}_0$
    Run inference (Algorithm 2) with $\mathbf{s}_{\boldsymbol{\theta}}$ to obtain $\{(\mathbf{x}_t^{(i)}, t, \boldsymbol{\lambda}, \mathcal{G})\}_{t=0}^{T-1},\ \forall i \in [I]$
    Push $\{(\mathbf{x}_t^{(i)}, t, \boldsymbol{\lambda}, \mathcal{G})\}_i^I$ to $\mathcal{B}$
  end for
  for inner $= 1, \ldots, N_{\text{inner}}$ **do**
    Sample a minibatch $\{(\mathbf{x}_t^{(j)}, t^{(j)}, \boldsymbol{\lambda}^{(j)}, \mathcal{G}^{(j)})\}_{j=1}^B$ from $\mathcal{B}$
    for $j = 1, \ldots, B$ **do**
      if $\text{Bernoulli}(\rho_{\text{pert}}) = 1$ **then** ▷ Perturb training pairs
        $\mathbf{x}_t^{(j)} \leftarrow \mathbf{x}_t^{(j)} + \epsilon_{\mathbf{x}} \cdot \boldsymbol{z}_x, \quad \boldsymbol{z}_x \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
        $\boldsymbol{\lambda}^{(j)} \leftarrow [\boldsymbol{\lambda}^{(j)} + \epsilon_{\boldsymbol{\lambda}} \cdot \boldsymbol{z}_\lambda]_+, \quad \boldsymbol{z}_\lambda \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
      end if
      if $\text{Bernoulli}(1 - \rho_{\text{exp}}(n)) = 1$ **then**
        $\boldsymbol{\lambda}^{(j)} \leftarrow \mathbf{v}, \quad \mathbf{v} \sim \pi_{\text{prior}}(\cdot)$ ▷ Sample from a prior
      end if
    end for
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla \frac{1}{B} \sum_j w(t^{(j)}) \left\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t^{(j)}, t^{(j)}, \boldsymbol{\lambda}^{(j)}, \mathcal{G}^{(j)}) - \nabla \log q_{T-t^{(j)}}(\mathbf{x}_t^{(j)} \mid \boldsymbol{\lambda}^{(j)}; \mathcal{G}^{(j)}) \right\|_2^2$
  end for
end for
return $\boldsymbol{\theta}$

Algorithm 2 PDI Dynamics for a given problem instance $\mathcal{G}$

Require: Trained score network $\mathbf{s}_{\theta^*}$, dual step size η_t, schedules $a_{1:T}, b_{1:T}$, initial multiplier $\boldsymbol{\lambda}_0$

Initialize output set $\mathcal{S} \leftarrow \emptyset$
for chain $= 1, \ldots, C$ **do**
  Sample $\mathbf{x}_0^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad i = 1, \ldots, I$ ▷ ensemble sharing one multiplier
  Initialize $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda}_0$
  for $t = 0, \ldots, T-1$ **do**
    $\mathbf{s}^{(i)} \leftarrow \mathbf{s}_{\theta^*}(\mathbf{x}_t^{(i)}, t, \boldsymbol{\lambda}, \mathcal{G})$ ▷ one score evaluation per step
    $\widehat{\mathbf{y}}_0^{(i)} \leftarrow \Pi_{\mathcal{X}}\left[(\mathbf{x}_t^{(i)} + \sigma_{T-t}^2 \mathbf{s}^{(i)})/\overline{\alpha}_{T-t}\right]$ ▷ Tweedie estimate
    $\mathbf{x}_{t+1}^{(i)} \leftarrow \frac{1}{\sqrt{a_{T-t}}}\left(\mathbf{x}_t^{(i)} + b_{T-t}\, \mathbf{s}^{(i)}\right) + \sqrt{b_{T-t}}\, \mathbf{z}^{(i)}, \quad \mathbf{z}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ▷ primal step
    $\boldsymbol{\lambda} \leftarrow \min\left\{\left[\boldsymbol{\lambda} + \eta_t \sum_i \mathbf{f}(\widehat{\mathbf{y}}_0^{(i)})/I\right]_+,\ \lambda_{\max}\right\}$ ▷ dual step, clipped at $\lambda_{\max}$
  end for
  Add $\{\mathbf{x}_T^{(i)}\}_i$ to $\mathcal{S}$
end for
return $\mathcal{S}$

therefore insensitive to the multiplier, while the nonlinearity bias of the dual step is largest there. Schedules that keep the SNR low for longer thus give the dual iterates more steps before the samples become sensitive to the multiplier, consistent with the noise-schedule ablations in Figure 3 and Appendix G.

B.3 Practical Considerations

In training the score model, we combine exploitation and exploration to generate training pairs $(\mathbf{x}, \boldsymbol{\lambda})$. Exploitation is performed by rolling out the sampling process using the most recent score model and storing the resulting primal-dual trajectories in a replay buffer. These rollouts expose the model to the states and dual variables it is likely to encounter during inference. Exploration is

introduced in two ways. First, we sample dual variables from an initial distribution to cover regions that may not be visited by the current rollouts. Second, we perturb both the primal and dual variables to improve robustness to local deviations from the replayed trajectories. The exploitation fraction $\rho_{\exp}(n)$ controls the balance between replayed dual variables and reference-sampled dual variables throughout training.

Although we refer to the learned network as a score model, our implementation uses the noise-prediction parameterization. For each training pair $(\mathbf{x}_t, \boldsymbol{\lambda})$, the MC estimator computes the score of the Gibbs distribution indexed by $\boldsymbol{\lambda}$ directly. Rather than regressing the network directly to this score, we reparameterize the target into the equivalent noise via $\boldsymbol{\epsilon}_t = -\sigma_{T-t}\mathbf{s}_t$, where $\mathbf{s}$ is the MC score estimate. The network is therefore trained to predict this noise target. During sampling, the predicted noise is converted back into score units using the same relation, and the resulting score is used in the PDI reverse update. Thus, throughout the paper, training a score model should be understood as training a noise-prediction network whose target is derived from the same MC score estimate.

In some of our numerical examples, the samples must satisfy physical domain constraints, such as box constraints on transmit powers or simplex constraints on portfolio weights. For convex $\mathcal{X}$, the exact posterior mean lies in $\mathcal{X}$, since it averages a posterior supported on $\mathcal{X}$. With an estimated score, projecting the Tweedie estimate $\widehat{\mathbf{y}}_0$ onto $\mathcal{X}$, as in Step 7 of Algorithm 2, can therefore only reduce its distance to the exact mean. We also divide by $\overline{\alpha}_{T-t} = \max\{\alpha_{\min}, \alpha_{T-t}\}$ when computing the Tweedie estimate to avoid numerical instabilities in the low-SNR regime. Beyond this projection, the implementation also enforces $\mathcal{X}$ inside the reverse process. In the Gaussian-mixture and wireless experiments, it clips the clean-sample estimates and the Monte Carlo candidates of Lemma 1 to the box $\mathcal{X}$. In the portfolio experiment, it runs the diffusion on logits and maps them to the simplex by a softmax. These devices keep every sample in $\mathcal{X}$ but alter the effective reference measure, most strongly near the boundary of $\mathcal{X}$, and our analysis does not cover them.

Finally, we implement (PDI-P) through the posterior mean of the forward process given $\mathbf{x}_t$ and the clean-sample estimate $\widehat{\mathbf{y}}_0$, with the posterior variance in place of b_{T-t} (Ho et al., 2020). Without clipping, this posterior-mean form has the same drift as (PDI-P).

C Analytical Proofs

The dual function of (1) is defined as

$$g(\boldsymbol{\lambda}) \;=\; \mathcal{L}(\mu^{\dagger}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}) \;=\; \min_{\mu \in \mathcal{P}(\mathcal{X})} \; \mathcal{L}(\mu, \boldsymbol{\lambda}), \tag{17}$$

where $\mu^{\dagger}_{\boldsymbol{\lambda}}$ is the Gibbs distribution induced by the dual variable $\boldsymbol{\lambda}$. We can then write the dual function in a closed form as

$$g(\boldsymbol{\lambda}) \;=\; \mathbb{E}_{\mu^{\dagger}_{\boldsymbol{\lambda}}}\Big[f_0(\mathbf{x}) + \boldsymbol{\lambda}^{\top}\mathbf{f}(\mathbf{x})\Big] - \beta\mathcal{H}(\mu^{\dagger}_{\boldsymbol{\lambda}}) = -\beta \log Z(\boldsymbol{\lambda}). \tag{18}$$

The true gradient of the dual function is

$$\nabla g(\boldsymbol{\lambda}_t) = \mathbb{E}_{\mu^{\dagger}_{\boldsymbol{\lambda}_t}}[\mathbf{f}(\mathbf{x}_T)], \tag{19}$$

which is the violation estimated by clean samples of the distribution $\mu^{\dagger}_{\boldsymbol{\lambda}_t}$. Since these clean samples are not available during the denoising process, PDI replaces this gradient with the biased gradient

$$\widehat{\nabla} g(\boldsymbol{\lambda}_t) = \mathbb{E}_{\mathbf{x}_t \sim p_t}\left[\mathbf{f}\left(\Pi_{\mathcal{X}}\left[\widehat{\mathbf{y}}_0(\mathbf{x}_t, \boldsymbol{\lambda}_t)\right]\right)\right], \tag{20}$$

where p_t is the marginal distribution at time t and $\widehat{\mathbf{y}}_0(\mathbf{x}_t, \boldsymbol{\lambda}_t)$ is the Tweedie posterior-mean estimate at $\mathbf{x}_t$ under the score model $\mathbf{s}_{\boldsymbol{\theta}}$. It is computed from the same score evaluation as the primal step. Here, $\Pi_{\mathcal{X}}$ denotes the Euclidean projection onto a convex set $\mathcal{X}$, as in Algorithm 2, and guarantees $\|\widehat{\nabla} g(\boldsymbol{\lambda}_t)\| \le R$ under Assumption 2. The exact posterior means lie in $\mathcal{X}$ and are left unchanged by the projection. The projection being nonexpansive, the bias bounds below hold as stated.

Additionally, the dual updates we write are of the population form, while the implementation averages over an ensemble instead, which we discuss after introducing Lemma 7. This formula replaces

clean samples with the Tweedie posterior means under the forward process started with $\mu^{\dagger}_{\boldsymbol{\lambda}_t}$. The bias in this gradient comes from four sources: i) replacing the score field with a parameterized network $\mathbf{s}_{\boldsymbol{\theta}}$, ii) propagating the dual mismatch and score-approximation error across diffusion steps (marginal mismatch), iii) the sampling (discretization) error of the exact-score reverse process, and iv) Tweedie's error, i.e., replacing clean samples $\mathbf{y}_0$ with a posterior mean $\mathbb{E}\big[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \mathbf{x}_t, \boldsymbol{\lambda}_t\,\big]$.

More concretely, we write the difference between the estimated gradient $\widehat{\nabla} g$ and true gradient ∇g as

$$
\begin{aligned}
\mathbf{b}_t(\boldsymbol{\lambda}_t) &\coloneqq \widehat{\nabla} g(\boldsymbol{\lambda}_t) - \nabla g(\boldsymbol{\lambda}_t) && (21)\\
&= \mathbb{E}_{p_t}\Big[\mathbf{f}\Big(\,\Pi_{\mathcal{X}}\big[\widehat{\mathbf{y}}_0(\mathbf{x}_t, \boldsymbol{\lambda}_t)\big]\,\Big)\Big] - \mathbb{E}_{\mu^{\dagger}_{\boldsymbol{\lambda}_t}}\big[\mathbf{f}(\mathbf{y}_0)\big] \\
&= \mathbb{E}_{p_t}\Big[\mathbf{f}\Big(\,\Pi_{\mathcal{X}}\big[\widehat{\mathbf{y}}_0(\mathbf{x}_t, \boldsymbol{\lambda}_t)\big]\,\Big)\Big] - \mathbb{E}_{p_t}\Big[\mathbf{f}\Big(\,\mathbb{E}\big[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \mathbf{x}_t, \boldsymbol{\lambda}_t\,\big]\,\Big)\Big] && (22)\\
&\quad + \mathbb{E}_{p_t}\Big[\mathbf{f}\Big(\,\mathbb{E}\big[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \mathbf{x}_t, \boldsymbol{\lambda}_t\,\big]\,\Big)\Big] - \mathbb{E}_{\widetilde{p}_t}\Big[\mathbf{f}\Big(\,\mathbb{E}\big[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \widetilde{\mathbf{x}}_t, \boldsymbol{\lambda}_t\,\big]\,\Big)\Big] && (23)\\
&\quad + \mathbb{E}_{\widetilde{p}_t}\Big[\mathbf{f}\Big(\,\mathbb{E}\big[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \widetilde{\mathbf{x}}_t, \boldsymbol{\lambda}_t\,\big]\,\Big)\Big] - \mathbb{E}_{q^{\boldsymbol{\lambda}_t}_{T-t}}\Big[\mathbf{f}\Big(\,\mathbb{E}[\mathbf{y}_0 \,|\, \mathbf{y}_{T-t}, \boldsymbol{\lambda}_t]\,\Big)\Big] && (24)\\
&\quad + \mathbb{E}_{q^{\boldsymbol{\lambda}_t}_{T-t}}\Big[\mathbf{f}\Big(\,\mathbb{E}[\mathbf{y}_0 \,|\, \mathbf{y}_{T-t}, \boldsymbol{\lambda}_t]\,\Big)\Big] - \mathbb{E}_{\mu^{\dagger}_{\boldsymbol{\lambda}_t}}\big[\mathbf{f}(\mathbf{y}_0)\big]. && (25)
\end{aligned}
$$

Here, we write $\widetilde{p}^{\boldsymbol{\lambda}}_t$ for the law at step t of (7) with exact scores, fixed multiplier $\boldsymbol{\lambda}$, and initialization $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and abbreviate $\widetilde{p}_t = \widetilde{p}^{\boldsymbol{\lambda}}_t$. The measure $q^{\boldsymbol{\lambda}_t}_{T-t}$ is the corresponding forward marginal. All posterior means are taken under the forward process (6) started at $\mu^{\dagger}_{\boldsymbol{\lambda}_t}$. The four terms are the score approximation error, the marginal mismatch caused by the changing multiplier, the sampling error of the exact-score reverse process with a fixed multiplier, and Tweedie's error. We bound them in Lemmas 5, 4, 3, and 2 (Section C.2).

The bias in the gradient results in dual convergence with a small but irreducible floor error. Theorem 1 characterizes this floor error and shows that it is controlled by the expected Tweedie posterior-mean error, the reverse-sampling error, and the approximation error of the parameterization. Theorem 2 then compares the terminal law p_T of PDI with that of an oracle sampler that knows $\boldsymbol{\lambda}^*$. In Sections C.3.1 and C.4, we provide the analytical proofs of these theorems. The assumptions under which our analysis holds are listed in Section C.1.

Throughout the analysis, we denote by $K^{\boldsymbol{\lambda}}_t$ the exact-score reverse kernel associated with a fixed multiplier $\boldsymbol{\lambda}$,

$$
K^{\boldsymbol{\lambda}}_t(\mathbf{x}, \cdot) \coloneqq \mathcal{N}\left(\frac{1}{\sqrt{a_{T-t}}}\Big(\mathbf{x} + b_{T-t}\mathbf{s}_t(\mathbf{x} \mid \boldsymbol{\lambda})\Big), b_{T-t}\mathbf{I}\right),
$$

where $\mathbf{s}_t(\mathbf{x} \mid \boldsymbol{\lambda}) = \nabla_{\mathbf{x}} \log q_{T-t}(\mathbf{x} \mid \boldsymbol{\lambda})$. We denote by $K^{\boldsymbol{\lambda}}_{\boldsymbol{\theta},t}$ the corresponding parameterized reverse kernel obtained by replacing $\mathbf{s}_t$ with $\mathbf{s}_{\boldsymbol{\theta}}$,

$$
K^{\boldsymbol{\lambda}}_{\boldsymbol{\theta},t}(\mathbf{x}, \cdot) \coloneqq \mathcal{N}\left(\frac{1}{\sqrt{a_{T-t}}}\Big(\mathbf{x} + b_{T-t}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t, \boldsymbol{\lambda})\Big), b_{T-t}\mathbf{I}\right).
$$

For any distribution p, we denote by $pK^{\boldsymbol{\lambda}}_t$ and $pK^{\boldsymbol{\lambda}}_{\boldsymbol{\theta},t}$ the distributions obtained after applying the corresponding one-step kernels to $\mathbf{x}_t \sim p$.

C.1 Assumptions

Our theoretical results hold under the following set of assumptions:

Assumption 1 (Existence of a strictly feasible solution)**.** *There exists $\mu' \in \mathcal{P}(\mathcal{X})$ such that $\mathbb{E}_{\mu'}[f_0(\mathbf{x})] - \beta\,\mathcal{H}(\mu') < C < \infty$, and $\mathbb{E}_{\mu'}[\mathbf{f}(\mathbf{x})] \preceq -\xi\mathbf{1} \prec \mathbf{0}$.*

Assumption 2 (Constraint regularity)**.** *The constraint function $\mathbf{f}$ is L_f-Lipschitz, i.e., $\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \le L_f\,\|\mathbf{x} - \mathbf{y}\|$. Also, the constraint function is bounded on the compact set $\mathcal{X}$, i.e., $\sup_{\mathbf{x}\in\mathcal{X}} \|\mathbf{f}(\mathbf{x})\| \le R$.*

Assumption 3 (Strong dual concavity)**.** *The dual function g is ϑ-strongly concave on $\boldsymbol{\Lambda}$, i.e., $\nabla^2_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}) \preceq -\vartheta\mathbf{I}$ for all $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda} = [0, \lambda_{\max}]^d$ is the box of Proposition 1, which contains $\boldsymbol{\lambda}^\star$ and onto which the dual iterates are projected (Appendix B.2).*

Since $\nabla^2 g(\boldsymbol{\lambda}) = -\frac{1}{\beta}\mathbf{Cov}_{\mu^\dagger_{\boldsymbol{\lambda}}}(\mathbf{f}(\mathbf{x}))$, the modulus is $\vartheta = \frac{1}{\beta}\min_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \sigma_{\min}\big(\mathbf{Cov}_{\mu^\dagger_{\boldsymbol{\lambda}}}(\mathbf{f}(\mathbf{x}))\big)$, where $\sigma_{\min}$ denotes the smallest eigenvalue. The Gibbs distributions $\mu^\dagger_{\boldsymbol{\lambda}}$ all share the null sets of the uniform distribution on $\mathcal{X}$. The covariance is therefore nonsingular for every $\boldsymbol{\lambda}$ as soon as no nonzero linear combination of the constraints is almost everywhere constant on $\mathcal{X}$, i.e., as soon as no constraint is redundant. Since the covariance is continuous in $\boldsymbol{\lambda}$ and $\boldsymbol{\Lambda}$ is compact, $\vartheta > 0$ then follows. Strong concavity must hold along the segments joining the iterates to $\boldsymbol{\lambda}^\star$, which is why we require it on the convex set $\boldsymbol{\Lambda}$ that contains all iterates rather than only near $\boldsymbol{\lambda}^\star$. A uniform modulus generally does not exist on all of $\mathbb{R}^d_+$, because the constraint covariance can vanish as the multipliers grow.

Assumption 4 (Rich parameterization)**.** *For any score function $\mathbf{s}_t = \nabla \log q_t$, there exists a parameterization $\boldsymbol{\theta}$ such that, uniformly over $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$ and t, it holds that*

$$\sup_t \sup_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \mathbb{E}_{p_t,\mathcal{G}}\Big[\big\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}, \mathcal{G}) - \mathbf{s}_t(\mathbf{x}_t|\boldsymbol{\lambda}; \mathcal{G})\big\|_2^2\Big] \le \varepsilon^2. \tag{26}$$

Assumption 4 is an approximation condition on the score network. We require it uniformly over the compact set $\boldsymbol{\Lambda}$ in which the dual iterates evolve and over time. We emphasize that this is an expected squared-error condition under the inference distribution, rather than a pointwise uniform bound over $\mathbf{x}$. The error ε aggregates three sources: i) the finite expressivity of the parameterization, ii) the variance of the Monte Carlo score estimator (Lemma 1) used to form the regression targets, and iii) the mismatch between the rollout-induced training distribution of $(\mathbf{x}_t, \boldsymbol{\lambda})$ pairs and the distribution encountered at inference. The expressivity of the parameterization controls the first source, and our training procedure is designed to control the latter two. That is, the exploration/exploitation rollout scheme with a replay buffer reduces the distribution shift, and K_{MC} candidates reduce the estimator variance. Our numerical experiments show that the difference between PDI with a trained score model (PDI–Net) and PDI with an MC score field (PDI–MC) is negligible. In some cases, PDI–Net improves PDI–MC on feasibility and tail-rate metrics.

Assumption 5. *Assume the noised optimal score is spatially Lipschitz, i.e. $\nabla^2_{\mathbf{x}} \log q_{T-t}(\mathbf{x}|\boldsymbol{\lambda})$ exists and is bounded in operator norm, $\sup_{\mathbf{x}\in\mathbb{R}^p} \|\nabla^2_{\mathbf{x}} \log q_{T-t}(\mathbf{x}|\boldsymbol{\lambda})\|_{\text{op}} \le M_t$, for each t and $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$.*

Assumption 6 (VP noise schedule)**.** *The diffusion process follows a variance-preserving schedule with $a_\tau = 1 - b_\tau$, $0 < b_\tau \le \frac{c_b}{T}$, for all $\tau = 1, \dots, T$, where $c_b > 0$ is independent of T. For the exact reverse kernel at inference step s, define the worst-case one-step stability factor*

$$\rho_s := \sup_{\boldsymbol{\lambda}\in\Lambda} \frac{1}{\sqrt{a_{T-s}}} \sup_{\mathbf{x}} \big\|\mathbf{I} + b_{T-s}\nabla^2_{\mathbf{x}} \log q_{T-s}(\mathbf{x} \mid \boldsymbol{\lambda})\big\|_{\text{op}}, \tag{27}$$

and let $\Psi_{j,t} := \prod_{s=j+1}^{t} \rho_s$, $0 \le j \le t$, with $\Psi_{t,t} = 1$. For the chosen T-step reverse process, define

$$C_\Psi(T) := \sup_{t\le T} \sum_{j=0}^{t-1} \Psi_{j,t-1}\,(t-j) < \infty, \tag{28}$$

and assume $C_\Psi(T)$ is bounded.

The two parts of Assumption 6 play different roles. The condition $b_\tau \le c_b/T$ is a refinement condition on the VP discretization. As the number of reverse steps increases, the perturbation introduced by each individual transition decreases proportionally to $1/T$. The second condition controls the accumulation of errors along the reverse trajectory. Here $\Psi_{j,t}$ is the product of the one-step stability factors between steps j and t, so $C_\Psi < \infty$ requires the total propagated influence of past perturbations to remain bounded. The weights $t - j$ account for the age of a perturbation, since the multipliers of steps j and t differ by at most $(t-j)\eta R$.

Assumption 7 (Dual sensitivity of the score model)**.** *The score model is L-Lipschitz in the multiplier, i.e., $\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t, \boldsymbol{\lambda}) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t, \boldsymbol{\lambda}')\| \le L\,\|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|$ for all $\mathbf{x} \in \mathbb{R}^p$, all t, and all $\boldsymbol{\lambda}, \boldsymbol{\lambda}' \in \boldsymbol{\Lambda}$.*

Assumption 7 concerns the trained network, which PDI evaluates at $\boldsymbol{\lambda}_t$ rather than $\boldsymbol{\lambda}^\star$. Note that it does not follow from Assumption 4, which controls the score error in value only.

C.2 Gradient Bias

In this subsection, we analyze the four sources of bias in the dual gradients, in the following order: the Tweedie error (25), the reverse-sampling error (24), marginal mismatch error (23), and the score approximation error (22).

C.2.1 TWEEDIE'S ERROR

Lemma 2. *Under Assumption 2, it holds that*

$$\left\| \mathbb{E}_{q^{\boldsymbol{\lambda}_t}_{T-t}}\Big[\mathbf{f}\Big(\mathbb{E}[\mathbf{y}_0 \,|\, \mathbf{y}_{T-t}, \boldsymbol{\lambda}_t]\Big)\Big] - \mathbb{E}_{\mu^{\dagger}_{\boldsymbol{\lambda}_t}}\big[\mathbf{f}(\mathbf{y}_0)\big] \right\|_2^2 \;\le\; L_f^2\, \chi_t^2, \tag{29}$$

where $\chi_t^2 := \sup_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \mathbb{E}\big[\|\mathbb{E}[\mathbf{y}_0 \mid \mathbf{y}_{T-t}, \boldsymbol{\lambda}] - \mathbf{y}_0\|_2^2\big]$ *is the uniform expected Tweedie posterior-mean error under the forward process* (6) *started at* $\mu^{\dagger}_{\boldsymbol{\lambda}}$.

Proof. Let $\Pi_t^{\boldsymbol{\lambda}_t}$ be the joint law of $(\mathbf{y}_0, \mathbf{y}_{T-t})$ under the forward process started at $\mu^{\dagger}_{\boldsymbol{\lambda}_t}$. Its marginals are $\mu^{\dagger}_{\boldsymbol{\lambda}_t}$ and $q^{\boldsymbol{\lambda}_t}_{T-t}$, and the conditional law of $\mathbf{y}_0$ given $\mathbf{y}_{T-t} = \mathbf{x}$ is the posterior $\pi_t(\cdot \,|\, \mathbf{x}, \boldsymbol{\lambda}_t)$. We can then write the bias as

$$b_1(\boldsymbol{\lambda}_t) \;:=\; \mathbb{E}_{q^{\boldsymbol{\lambda}_t}_{T-t}}\Big[\mathbf{f}\Big(\mathbb{E}[\mathbf{y}_0 \,|\, \mathbf{y}_{T-t}, \boldsymbol{\lambda}_t]\Big)\Big] - \mathbb{E}_{\mu^{\dagger}_{\boldsymbol{\lambda}_t}}\big[\mathbf{f}(\mathbf{y}_0)\big] \tag{30}$$

$$=\; \mathbb{E}_{\Pi_t^{\boldsymbol{\lambda}_t}}\Big[\mathbf{f}\Big(\mathbb{E}[\mathbf{y}_0 \,|\, \mathbf{y}_{T-t}, \boldsymbol{\lambda}_t]\Big) - \mathbf{f}(\mathbf{y}_0)\Big]. \tag{31}$$

Taking the norm and applying Jensen's inequality and Assumption 2, we get

$$\|b_1(\boldsymbol{\lambda}_t)\|_2^2 \;\le\; \mathbb{E}_{\Pi_t^{\boldsymbol{\lambda}_t}}\Big[\big\|\mathbf{f}\Big(\mathbb{E}[\mathbf{y}_0 \,|\, \mathbf{y}_{T-t}, \boldsymbol{\lambda}_t]\Big) - \mathbf{f}(\mathbf{y}_0)\big\|_2^2\Big] \tag{32}$$

$$\le\; L_f^2\, \mathbb{E}_{\Pi_t^{\boldsymbol{\lambda}_t}}\Big[\big\|\mathbb{E}[\mathbf{y}_0 \,|\, \mathbf{y}_{T-t}, \boldsymbol{\lambda}_t] - \mathbf{y}_0\big\|_2^2\Big] \;\le\; L_f^2\, \chi_t^2. \tag{33}$$

For convex $\mathcal{X}$, the posterior mean lies in $\mathcal{X}$, so $\mathbf{f}$ is evaluated on its domain. □

C.2.2 REVERSE-SAMPLING ERROR

Lemma 3. *Under Assumption 2, it holds that*

$$\left\| \mathbb{E}_{\widetilde{p}_t}\Big[\mathbf{f}\Big(\mathbb{E}\big[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \widetilde{\mathbf{x}}_t, \boldsymbol{\lambda}_t\,\big]\Big)\Big] - \mathbb{E}_{q^{\boldsymbol{\lambda}_t}_{T-t}}\Big[\mathbf{f}\Big(\mathbb{E}[\mathbf{y}_0 \,|\, \mathbf{y}_{T-t}, \boldsymbol{\lambda}_t]\Big)\Big] \right\|_2^2 \;\le\; 4R^2\, \delta_t^2, \tag{34}$$

where $\delta_t \coloneqq \sup_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \mathrm{TV}\big(\widetilde{p}_t^{\boldsymbol{\lambda}}, q^{\boldsymbol{\lambda}}_{T-t}\big)$ *and* $\widetilde{p}_t^{\boldsymbol{\lambda}}$ *is the law at step* t *of the reverse process* (7) *with exact scores and multiplier* $\boldsymbol{\lambda}$.

Proof. Both expectations integrate the function $\mathbf{h}(\mathbf{x}) = \mathbf{f}\big(\mathbb{E}[\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \mathbf{x}, \boldsymbol{\lambda}_t]\big)$. For convex $\mathcal{X}$, the posterior mean lies in $\mathcal{X}$, so $\|\mathbf{h}\| \le R$ by Assumption 2. Hence, the difference of the two expectations has norm at most $2R\,\mathrm{TV}(\widetilde{p}_t, q^{\boldsymbol{\lambda}_t}_{T-t}) \le 2R\,\delta_t$. □

The quantity δ_t is the reverse-sampling error, i.e., the error of an exact-score diffusion sampler at an intermediate step. With exact scores, it arises from both the discrete Gaussian reverse steps and the Gaussian initialization. It vanishes for an exact time reversal, and bounds of the type in (Chen et al., 2023; Benton et al., 2024) control it under schedule conditions. It is the intermediate-step counterpart of the oracle error in Appendix C.4.

C.2.3 MARGINAL MISMATCH

Lemma 4. *Under Assumptions 2– 6, it holds that*

$$\left\| \mathbb{E}_{\widetilde{p}_t}\Big[\mathbf{f}\Big(\mathbb{E}\big[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \widetilde{\mathbf{x}}_t, \boldsymbol{\lambda}_t\,\big]\Big)\Big] - \mathbb{E}_{p_t}\Big[\mathbf{f}\Big(\mathbb{E}\big[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \mathbf{x}_t, \boldsymbol{\lambda}_t\,\big]\Big)\Big] \right\|_2^2$$

$$\le\; 2L_h^2 C_{\mathrm{hist}}^2 \eta^2 R^2 + \mathcal{O}\left(\frac{\varepsilon^2}{T^2}\right). \tag{35}$$

where $L_h := L_f \max_{0\le t<T}(1 + \sigma_{T-t}^2 M_t)/\alpha_{T-t}$.

Proof. We use synchronous coupling to compare the Tweedie posterior mean at step t under two processes: 1) PDI, and 2) a reference reverse process (7) under $\boldsymbol{\lambda}_t$.

Under the PDI dynamics, one reverse step is given by

$$\mathbf{x}_{s+1} \;=\; \frac{1}{\sqrt{a_{T-s}}}\Big(\mathbf{x}_s + b_{T-s}\,\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, \boldsymbol{\lambda}_s)\Big) + \sqrt{b_{T-s}}\boldsymbol{\epsilon}_s, \tag{36}$$

where $\mathbf{x}_s \sim p_s$ while the corresponding step under (7) with fixed $\boldsymbol{\lambda}_t$ is

$$\widetilde{\mathbf{x}}_{s+1} \;=\; \frac{1}{\sqrt{a_{T-s}}}\Big(\widetilde{\mathbf{x}}_s + b_{T-s}\,\mathbf{s}_s(\widetilde{\mathbf{x}}_s|\boldsymbol{\lambda}_t)\Big) + \sqrt{b_{T-s}}\boldsymbol{\epsilon}_s. \tag{37}$$

Under the synchronous coupling, the two processes share the same noise $\boldsymbol{\epsilon}_s$ and the same initialization, i.e., $\mathbf{x}_0 = \widetilde{\mathbf{x}}_0$. Thus, the difference between the two noisy samples $\mathbf{x}_{s+1}$ and $\widetilde{\mathbf{x}}_{s+1}$ is

$$\begin{aligned}\mathbf{x}_{s+1} - \widetilde{\mathbf{x}}_{s+1} \;=\; &\frac{1}{\sqrt{a_{T-s}}}\Big(\mathbf{x}_s - \widetilde{\mathbf{x}}_s\Big) + \frac{b_{T-s}}{\sqrt{a_{T-s}}}\Big(\mathbf{s}_s(\mathbf{x}_s|\boldsymbol{\lambda}_s) - \mathbf{s}_s(\widetilde{\mathbf{x}}_s|\boldsymbol{\lambda}_s)\Big)\\ &+ \frac{b_{T-s}}{\sqrt{a_{T-s}}}\Big(\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, \boldsymbol{\lambda}_s) - \mathbf{s}_s(\mathbf{x}_s|\boldsymbol{\lambda}_s)\Big)\\ &+ \frac{b_{T-s}}{\sqrt{a_{T-s}}}\Big(\mathbf{s}_s(\widetilde{\mathbf{x}}_s|\boldsymbol{\lambda}_s) - \mathbf{s}_s(\widetilde{\mathbf{x}}_s|\boldsymbol{\lambda}_t)\Big),\end{aligned} \tag{38}$$

where we added and subtracted $\mathbf{s}_s(\widetilde{\mathbf{x}}_s|\boldsymbol{\lambda}_s)$ and $\mathbf{s}_s(\mathbf{x}_s|\boldsymbol{\lambda}_s)$. These quantities have been analyzed in Lemmas 7 and 8 for the optimal kernel $K_t^{\boldsymbol{\lambda}^*}$. Here, we use generalized formulas for any kernel associated with (7). Similarly to (99)–(103), we can bound the first term with

$$\Big\|\frac{1}{\sqrt{a_{T-s}}}\Big(\mathbf{x}_s - \widetilde{\mathbf{x}}_s\Big) + \frac{b_{T-s}}{\sqrt{a_{T-s}}}\Big(\mathbf{s}_s(\mathbf{x}_s|\boldsymbol{\lambda}_s) - \mathbf{s}_s(\widetilde{\mathbf{x}}_s|\boldsymbol{\lambda}_s)\Big)\Big\|_2 \;\le\; \rho_s(\boldsymbol{\lambda}_s)\cdot\|\mathbf{x}_s - \widetilde{\mathbf{x}}_s\|_2, \tag{39}$$

where $\rho_s(\boldsymbol{\lambda}_s) = \frac{1}{\sqrt{a_{T-s}}}\,\sup_{\mathbf{x}}\,\big\|\mathbf{I} + b_{T-s}\nabla^2_{\mathbf{x}}\log q_{T-s}(\mathbf{x}|\boldsymbol{\lambda}_s)\big\|_{\text{op}}$ is the stability factor of the one-step kernel $K_s^{\boldsymbol{\lambda}_s}$. For the last term, we follow (105)–(109) to get

$$\|\mathbf{s}_s(\widetilde{\mathbf{x}}_s|\boldsymbol{\lambda}_s) - \mathbf{s}_s(\widetilde{\mathbf{x}}_s|\boldsymbol{\lambda}_t)\|_2 \;\le\; \gamma_s\cdot\|\boldsymbol{\lambda}_s - \boldsymbol{\lambda}_t\|_2, \tag{40}$$

where $\gamma_s = \frac{R\alpha_{T-s}}{\beta\sigma^2_{T-s}}\sqrt{\sup_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}}\|\mathbf{Cov}_{\pi_s}(\mathbf{y}_0|\widetilde{\mathbf{x}}_s,\boldsymbol{\lambda})\|_{\text{op}}}$.

Combining the two results, along with Assumption 4, we obtain

$$\begin{aligned}\|\mathbf{x}_{s+1} - \widetilde{\mathbf{x}}_{s+1}\|_2 \;&\le \rho_s(\boldsymbol{\lambda}_s)\cdot\|\mathbf{x}_s - \widetilde{\mathbf{x}}_s\|_2\\ &\quad + \frac{b_{T-s}}{\sqrt{a_{T-s}}}\Big(\gamma_s\|\boldsymbol{\lambda}_s - \boldsymbol{\lambda}_t\|_2 + \|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, \boldsymbol{\lambda}_s) - \mathbf{s}_s(\mathbf{x}_s|\boldsymbol{\lambda}_s)\|_2\Big).\end{aligned} \tag{41}$$

Then, we take expectation with respect to the coupling Π_{s+1} to get

$$\begin{aligned}\mathbb{E}_{\Pi_{s+1}}\Big[\|\mathbf{x}_{s+1} - \widetilde{\mathbf{x}}_{s+1}\|_2\Big] \;&\le \rho_s(\boldsymbol{\lambda}_s)\cdot\mathbb{E}_{\Pi_s}\Big[\|\mathbf{x}_s - \widetilde{\mathbf{x}}_s\|_2\Big]\\ &\quad + \frac{b_{T-s}}{\sqrt{a_{T-s}}}\Big(\gamma_s\cdot\|\boldsymbol{\lambda}_s - \boldsymbol{\lambda}_t\|_2 + \mathbb{E}_{p_s}\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_s, s, \boldsymbol{\lambda}_s) - \mathbf{s}_s(\mathbf{x}_s|\boldsymbol{\lambda}_s)\|_2\Big)\\ &\le \rho_s(\boldsymbol{\lambda}_s)\cdot\mathbb{E}_{\Pi_s}\Big[\|\mathbf{x}_s - \widetilde{\mathbf{x}}_s\|_2\Big] + \frac{b_{T-s}}{\sqrt{a_{T-s}}}\Big(\gamma_s\cdot\|\boldsymbol{\lambda}_s - \boldsymbol{\lambda}_t\|_2 + \varepsilon\Big).\end{aligned} \tag{42}$$

The last inequality follows from Assumption 4 and Jensen's inequality. With this inequality in place, let $\mathbf{h}(\mathbf{x}_t) \;\coloneqq\; \mathbf{f}\Big(\mathbb{E}\big[\,\mathbf{y}_0\,|\,\mathbf{y}_{T-t} = \mathbf{x}_t, \boldsymbol{\lambda}_t\,\big]\Big)$, and bound the bias with

$$\begin{aligned}\|b_2(\boldsymbol{\lambda}_t)\|_2 \;&\coloneqq\; \Big\|\,\mathbb{E}_{\widetilde{p}_t}\big[\mathbf{h}(\widetilde{\mathbf{x}}_t)\big] - \mathbb{E}_{p_t}\big[\mathbf{h}(\mathbf{x}_t)\big]\,\Big\|_2\\ &=\; \Big\|\,\mathbb{E}_{\Pi_t}\big[\mathbf{h}(\widetilde{\mathbf{x}}_t) - \mathbf{h}(\mathbf{x}_t)\big]\,\Big\|_2 &(43)\\ &\le\; L_h\mathbb{E}_{\Pi_t}\|\widetilde{\mathbf{x}}_t - \mathbf{x}_t\|_2 &(44)\\ &\le\; L_h\Big(\rho_{t-1}(\boldsymbol{\lambda}_{t-1})\cdot\mathbb{E}_{\Pi_{t-1}}\big[\|\mathbf{x}_{t-1} - \widetilde{\mathbf{x}}_{t-1}\|_2\big] + \frac{b_{T-t+1}}{\sqrt{a_{T-t+1}}}\big(\gamma_{t-1}\cdot\|\boldsymbol{\lambda}_{t-1} - \boldsymbol{\lambda}_t\|_2 + \varepsilon\big)\Big), &(45)\end{aligned}$$

where L_h is the Lipschitz constant of $\mathbf{h}(\mathbb{E}[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \mathbf{x}_t, \boldsymbol{\lambda}_t\,])$ and depends on L_f and the score function.

Recursively, we get the bound:

$$\begin{aligned}\|b_2(\boldsymbol{\lambda}_t)\|_2 \;&\le\; L_h \sum_{j=0}^{t-1} \left(\prod_{s=j+1}^{t-1} \rho_s(\boldsymbol{\lambda}_s) \right) \cdot \frac{b_{T-j}}{\sqrt{a_{T-j}}} \Big(\gamma_j \|\boldsymbol{\lambda}_j - \boldsymbol{\lambda}_t\|_2 + \varepsilon \Big) \\ &=\; L_h \sum_{j=0}^{t-1} \Psi_{j,t-1} \frac{b_{T-j}}{\sqrt{a_{T-j}}} \Big(\gamma_j \|\boldsymbol{\lambda}_j - \boldsymbol{\lambda}_t\|_2 + \varepsilon \Big), \end{aligned} \tag{46}$$

where $\Psi_{j,t}$ is the stability product of Assumption 6 and $\rho_s = \sup_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} \rho_s(\boldsymbol{\lambda})$.

For a variance-preserving noise schedule (Assumption 6), we further have

$$\frac{b_{T-j}}{\sqrt{a_{T-j}}} \gamma_j = \frac{R}{\beta} \frac{b_{T-j} \alpha_{T-j-1}}{\sigma_{T-j}^2} \sqrt{\sup_{\mathbf{x}, \boldsymbol{\lambda} \in \Lambda} \left\| \mathrm{Cov}_{\pi_j}(\mathbf{y}_0 \mid \mathbf{x}, \boldsymbol{\lambda}) \right\|_{\mathrm{op}}} \tag{47}$$

$$\le \frac{R}{\beta} \sqrt{\sup_{\mathbf{x}, \boldsymbol{\lambda} \in \Lambda} \left\| \mathrm{Cov}_{\pi_j}(\mathbf{y}_0 \mid \mathbf{x}, \boldsymbol{\lambda}) \right\|_{\mathrm{op}}}. \tag{48}$$

We also have

$$\|\boldsymbol{\lambda}_j - \boldsymbol{\lambda}_t\|_2 \le (t-j) \eta R, \tag{49}$$

Hence,

$$C_{\text{hist}} \;:=\; \sum_{j=0}^{t-1} \Psi_{j,t-1} \frac{b_{T-j}}{\sqrt{a_{T-j}}} \gamma_j (t-j) \;\le\; \frac{R \,\mathrm{diam}(\mathcal{X})}{2\beta} \, C_\Psi < \infty, \tag{50}$$

since the posterior is supported on $\mathcal{X}$, so that its covariance has operator norm at most $\mathrm{diam}(\mathcal{X})^2/4$, and by Assumption 6. Therefore, we have $\|b_2(\boldsymbol{\lambda}_t)\|_2^2 \le 2L_h^2 C_{\text{hist}}^2 \eta^2 R^2 + 2L_h^2 H_t^2$, using Young's inequality. Under the VP noise schedule we also have $\frac{b_{T-j}}{\sqrt{a_{T-j}}} \le \frac{c_b}{T\sqrt{1-c_b/T}}$. Therefore,

$$H_t \;:=\; \varepsilon \sum_{j=0}^{t-1} \Psi_{j,t-1} \frac{b_{T-j}}{\sqrt{a_{T-j}}} \le \frac{c_b \varepsilon}{T\sqrt{1-c_b/T}} \sum_{j=0}^{t-1} \Psi_{j,t-1} \;\le\; \frac{c_b \, C_\Psi \, \varepsilon}{T\sqrt{1-c_b/T}} = \mathcal{O}\left(\frac{\varepsilon}{T}\right). \tag{51}$$

Moreover, the marginal-mismatch bias satisfies

$$\|b_2(\boldsymbol{\lambda}_t)\|_2^2 \le 2L_h^2 C_{\text{hist}}^2 \eta^2 R^2 + \mathcal{O}\left(\frac{\epsilon^2}{T^2}\right), \tag{52}$$

which completes the proof. □

C.2.4 Parameterization Error

Lemma 5. *Under Assumptions 2, 4 and 6, it holds that*

$$\left\| \mathbb{E}_{p_t} \Big[\mathbf{f}\big(\Pi_{\mathcal{X}} \big[\widehat{\mathbf{y}}_0(\mathbf{x}_t, \boldsymbol{\lambda}_t) \big] \big) - \mathbf{f}\big(\mathbb{E}[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \mathbf{x}_t, \boldsymbol{\lambda}_t\,] \big) \Big] \right\|_2^2 \;\le\; L_f^2 \frac{\sigma_{T-t}^4}{\alpha_{T-t}^2} \varepsilon^2. \tag{53}$$

Proof. The Tweedie posterior means are given as

$$\widehat{\mathbf{y}}_0(\mathbf{x}_t, \boldsymbol{\lambda}_t) \;=\; \frac{1}{\alpha_{T-t}} \Big(\mathbf{x}_t + \sigma_{T-t}^2 \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}_t) \Big), \tag{54}$$

$$\mathbb{E}[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t} = \mathbf{x}_t, \boldsymbol{\lambda}_t\,] \;=\; \frac{1}{\alpha_{T-t}} \Big(\mathbf{x}_t + \sigma_{T-t}^2 \mathbf{s}_t(\mathbf{x}_t | \boldsymbol{\lambda}_t) \Big), \tag{55}$$

where the second expression is the exact posterior mean by Tweedie's formula. The implementation divides by $\overline{\alpha}_{T-t} = \max\{\alpha_{\min}, \alpha_{T-t}\}$ instead, which only affects the lowest-SNR steps (Appendix B.3). Therefore, we have

$$\begin{aligned}
&\Big\|\mathbb{E}_{p_t}\Big[\mathbf{f}\big(\,\Pi_{\mathcal{X}}\big[\widehat{\mathbf{y}}_0(\mathbf{x}_t,\boldsymbol{\lambda}_t)\big]\,\big) - \mathbf{f}\big(\,\mathbb{E}\big[\,\mathbf{y}_0 \,|\, \mathbf{y}_{T-t}=\mathbf{x}_t,\boldsymbol{\lambda}_t\,\big]\,\big)\Big]\Big\|_2 \\
&\qquad \le\; L_f \frac{\sigma_{T-t}^2}{\alpha_{T-t}} \mathbb{E}_{p_t}\Big\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t,\boldsymbol{\lambda}_t) - \mathbf{s}_t(\mathbf{x}_t|\boldsymbol{\lambda}_t)\Big\|_2 \\
&\qquad \le\; L_f \frac{\sigma_{T-t}^2}{\alpha_{T-t}} \varepsilon, \qquad (56)
\end{aligned}$$

by the nonexpansiveness of $\Pi_{\mathcal{X}}$ (the exact posterior mean lies in $\mathcal{X}$), Jensen's inequality, and Assumptions 2 and 4, which completes the proof. □

C.3 Dual Convergence

We analyze how the bias in the PDI dual update affects convergence. Lemma 6 establishes a one-step recursion for the distance to the optimal multiplier in terms of the gradient bias $\mathbf{b}_t(\boldsymbol{\lambda}_t)$, which we then bound to obtain the last-iterate guarantee in Theorem 1. We additionally derive a time-average dual bound, which is used in the proof of Theorem 2.

Lemma 6. *Under Assumptions 2 and 3, and with $\eta_t = \eta$, the distance between $\boldsymbol{\lambda}_t$ and the optimum satisfies*

$$\begin{aligned}
&\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^\star\|^2 + \eta\big[D^\star - g(\boldsymbol{\lambda}_t)\big] + 2\eta\beta D_{KL}(\mu^\dagger_{\boldsymbol{\lambda}_t}\|\mu^\star) \\
&\qquad\qquad \le\; \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 + \frac{2\eta}{\vartheta}\|\mathbf{b}_t(\boldsymbol{\lambda}_t)\|^2 + \eta^2 R^2, \qquad (57)
\end{aligned}$$

or equivalently, by

$$\begin{aligned}
&\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^\star\|^2 + 2\eta\beta D_{KL}(\mu^\dagger_{\boldsymbol{\lambda}_t}\|\mu^\star) \\
&\qquad\qquad \le\; \left(1 - \frac{\eta\vartheta}{2}\right)\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 + \frac{2\eta}{\vartheta}\|\mathbf{b}_t(\boldsymbol{\lambda}_t)\|^2 + \eta^2 R^2. \qquad (58)
\end{aligned}$$

Proof. We analyze the update as implemented, with the multipliers projected onto the box $\boldsymbol{\Lambda} \ni \boldsymbol{\lambda}^\star$ of Assumption 3, i.e., $\boldsymbol{\lambda}_{t+1} = \Pi_{\boldsymbol{\Lambda}}[\boldsymbol{\lambda}_t + \eta\widehat{\nabla} g(\boldsymbol{\lambda}_t)]$. This is the projection $[\cdot]_+$ followed by clipping at $\lambda_{\max}$. By the non-expansiveness of $\Pi_{\boldsymbol{\Lambda}}$, we have $\|\Pi_{\boldsymbol{\Lambda}}[\boldsymbol{\lambda}] - \boldsymbol{\lambda}^\star\| \le \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^\star\|$, and therefore,

$$\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^\star\|^2 \;\le\; \|\boldsymbol{\lambda}_t + \eta\widehat{\nabla} g(\boldsymbol{\lambda}_t) - \boldsymbol{\lambda}^\star\|^2, \qquad (59)$$

where $\widehat{\nabla} g(\boldsymbol{\lambda}_t) := \mathbb{E}_{\mathbf{x}_t\sim p_t}\Big[\mathbf{f}\Big(\Pi_{\mathcal{X}}\big[\widehat{\mathbf{y}}_0(\mathbf{x}_t,\boldsymbol{\lambda}_t)\big]\Big)\Big]$ is the biased gradient in (PDI-D).

Define the true gradient of the dual function as

$$\nabla g(\boldsymbol{\lambda}_t) = \mathbb{E}_{\mu^\dagger_{\boldsymbol{\lambda}_t}}[\mathbf{f}(\mathbf{y}_0)],$$

which is the constraint violations evaluated on clean samples under the current Gibbs distribution $\mu^\dagger_{\boldsymbol{\lambda}_t}$. PDI replaces this true gradient with the biased gradient $\widehat{\nabla} g(\boldsymbol{\lambda}_t)$, Thus, we can write

$$\begin{aligned}
\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^\star\|^2 \;&\le\; \|\boldsymbol{\lambda}_t + \eta\widehat{\nabla} g(\boldsymbol{\lambda}_t) - \boldsymbol{\lambda}^\star\|^2 \qquad (60)\\
&\le\; \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 + 2\eta\langle\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star, \widehat{\nabla} g(\boldsymbol{\lambda}_t)\rangle + \eta^2\|\widehat{\nabla} g(\boldsymbol{\lambda}_t)\|^2 \qquad (61)\\
&=\; \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 + 2\eta\langle\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star, \nabla g(\boldsymbol{\lambda}_t)\rangle + 2\eta\langle\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star, \mathbf{b}_t(\boldsymbol{\lambda}_t)\rangle + \eta^2\|\widehat{\nabla} g(\boldsymbol{\lambda}_t)\|^2, \qquad (62)
\end{aligned}$$

where $\widehat{\nabla} g(\boldsymbol{\lambda}_t) = \nabla g(\boldsymbol{\lambda}_t) + \mathbf{b}_t(\boldsymbol{\lambda}_t)$ and the bias $\mathbf{b}_t(\boldsymbol{\lambda}_t)$ is split into three terms, as characterized in (21).

By Lemma 9, the Lagrangian function can be written as $\mathcal{L}(\mu, \boldsymbol{\lambda}) = \beta D_{\mathrm{KL}}(\mu \| \mu^\dagger_{\boldsymbol{\lambda}}) + g(\boldsymbol{\lambda})$, which leads to

$$\begin{aligned}\langle \boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star, \nabla g(\boldsymbol{\lambda}_t) \rangle &= \mathcal{L}(\mu^\dagger_{\boldsymbol{\lambda}_t}, \boldsymbol{\lambda}_t) - \mathcal{L}(\mu^\dagger_{\boldsymbol{\lambda}_t}, \boldsymbol{\lambda}^\star) \\ &= \underbrace{g(\boldsymbol{\lambda}_t) - g(\boldsymbol{\lambda}^\star)}_{\text{duality gap}} - \beta D_{\mathrm{KL}}(\mu^\dagger_{\boldsymbol{\lambda}_t} \| \mu^\star). \end{aligned} \tag{63}$$

The first equality follows from the definition in (2) and the second follows from the definition in the lemma. Using Young's inequality, for any $c > 0$, we can bound the inner product between $\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star$ and the bias by

$$2\langle \boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star, \mathbf{b}_t \rangle \le c\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 + \frac{1}{c}\|\mathbf{b}_t\|^2. \tag{64}$$

Choose $c = \vartheta/2$. By substituting (63) and (64) in (60), we get

$$\begin{aligned}&\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^\star\|^2 + 2\eta\Big(D^\star - g(\boldsymbol{\lambda}_t)\Big) + 2\eta\beta D_{\mathrm{KL}}(\mu^\dagger_{\boldsymbol{\lambda}_t} \| \mu^\star) \\ &\qquad \le \left(1 + \frac{\eta\vartheta}{2}\right)\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 + \frac{2\eta}{\vartheta}\|\mathbf{b}_t(\boldsymbol{\lambda}_t)\|^2 + \eta^2\|\widehat{\nabla} g(\boldsymbol{\lambda}_t)\|^2, \end{aligned} \tag{65}$$

with $D^\star := g(\boldsymbol{\lambda}^\star)$. The segment between $\boldsymbol{\lambda}_t$ and $\boldsymbol{\lambda}^\star$ lies in the convex set $\boldsymbol{\Lambda}$, where g is ϑ-strongly concave (Assumption 3), and $\boldsymbol{\lambda}^\star$ maximizes g over $\mathbb{R}^d_+ \supseteq \boldsymbol{\Lambda}$. Hence, $D^\star - g(\boldsymbol{\lambda}_t) \ge \frac{\vartheta}{2}\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2$.

(i) The time-average bound:

We split the dual gap in the LHS into two halves and lower bound one of them with $\frac{\eta\vartheta}{2}\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2$. Thus, we have

$$\begin{aligned}&\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^\star\|^2 + \eta\big(D^\star - g(\boldsymbol{\lambda}_t)\big) + 2\eta\beta D_{\mathrm{KL}}(\mu^\dagger_{\boldsymbol{\lambda}_t} \| \mu^\star) \\ &\qquad \le \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 + \frac{2\eta}{\vartheta}\|\mathbf{b}_t(\boldsymbol{\lambda}_t)\|^2 + \eta^2\|\widehat{\nabla} g(\boldsymbol{\lambda}_t)\|^2. \end{aligned} \tag{66}$$

(ii) The final-iterate bound:

For the final-iterate result, we upper-bound the duality gap term with

$$\begin{aligned}&\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^\star\|^2 + 2\eta\beta D_{\mathrm{KL}}(\mu^\dagger_{\boldsymbol{\lambda}_t} \| \mu^\star) \\ &\qquad \le \left(1 + \frac{\eta\vartheta}{2} - \eta\vartheta\right)\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 + \frac{2\eta}{\vartheta}\|\mathbf{b}_t(\boldsymbol{\lambda}_t)\|^2 + \eta^2\|\widehat{\nabla} g(\boldsymbol{\lambda}_t)\|^2. \end{aligned} \tag{67}$$

Since the projected Tweedie estimates lie in $\mathcal{X}$, Assumption 2 gives $\|\widehat{\nabla} g(\boldsymbol{\lambda}_t)\| \le R$, which completes the proof. □

Ensemble update. The recursion is deterministic given the update direction. For the implemented ensemble update, it therefore holds pathwise, with $\widehat{\nabla} g(\boldsymbol{\lambda}_t)$ replaced by the ensemble average of the projected Tweedie estimates and $\mathbf{b}_t$ by its realized error. Taking expectations yields the bounds below with $\|\mathbf{b}_t\|^2$ replaced by $\mathbb{E}\|\mathbf{b}_t\|^2$. The lemmas of Appendix C.2 bound the population part of this error. The realized error additionally contains ensemble fluctuations.

C.3.1 Convergence of Last-iterate Multiplier: Proof of Theorem 1

First, we bound the bias in the gradients. The bias comprises four components as described in (21). The Tweedie posterior mean error is bounded by Lemma 2:

$$\|b_1(\boldsymbol{\lambda}_t)\|^2 \le L_f^2\, \chi_t^2. \tag{68}$$

The marginal mismatch error is bounded by Lemma 4:

$$\|b_2(\boldsymbol{\lambda}_t)\|^2 \le 2L_h^2 C_{\mathrm{hist}}^2 \eta^2 R^2 + \mathcal{O}\left(\frac{\epsilon^2}{T^2}\right). \tag{69}$$

Finally, the parameterization error is bounded by Lemma 5:

$$\|b_3(\boldsymbol{\lambda}_t)\|^2 \;\le\; L_f^2 \,\frac{\sigma_{T-t}^4}{\alpha_{T-t}^2}\varepsilon^2 \;\le\; \mathcal{O}\left(\varepsilon^2 \frac{(T-t)^2}{T^2}\right). \tag{70}$$

The reverse-sampling error is bounded by Lemma 3:

$$\|b_4(\boldsymbol{\lambda}_t)\|^2 \;\le\; 4R^2\,\delta_t^2. \tag{71}$$

The total bias can then be bounded as

$$\|\mathbf{b}_t(\boldsymbol{\lambda}_t)\|^2 \;\le\; 4\|b_1(\boldsymbol{\lambda}_t)\|^2 + 4\|b_2(\boldsymbol{\lambda}_t)\|^2 + 4\|b_3(\boldsymbol{\lambda}_t)\|^2 + 4\|b_4(\boldsymbol{\lambda}_t)\|^2. \tag{72}$$

We then utilize this bound to prove the two statements of the theorem.

Choose $\eta \in (0, \frac{2}{\vartheta})$ and let $r := 1 - \frac{\eta\vartheta}{2}$. Thus, we have $0 < r < 1$. Starting from the dual recursion in (67), which is

$$\begin{aligned}\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^\star\|^2 + 2\eta\beta D_{\mathrm{KL}}(\mu^\dagger_{\boldsymbol{\lambda}_t}\|\mu^\star) \\ \le\; \left(1 - \frac{\eta\vartheta}{2}\right)\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 + \frac{2\eta}{\vartheta}\|\mathbf{b}_t(\boldsymbol{\lambda}_t)\|^2 + \eta^2 R^2.\end{aligned} \tag{73}$$

We then drop the KL term because it is non-negative, and apply recursion to get

$$\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 \;\le\; r^t\|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^\star\|^2 + \sum_{s<t} r^{t-1-s}\Big(\frac{2\eta}{\vartheta}\|\mathbf{b}_s(\boldsymbol{\lambda}_s)\|^2 + \eta^2 R^2\Big). \tag{74}$$

Additionally, we have

$$\sum_{s<t} r^{t-1-s}\eta^2 R^2 \le \frac{\eta^2 R^2}{1-r} \le \frac{2}{\vartheta}\eta R^2, \tag{75}$$

and therefore

$$\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 \;\le\; r^t\|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^\star\|^2 + \frac{2\eta}{\vartheta}R^2 + \frac{2\eta}{\vartheta}\sum_{s<t} r^{t-1-s}\|\mathbf{b}_s(\boldsymbol{\lambda}_s)\|^2. \tag{76}$$

This bound holds at every step t, and letting $t = T$ completes the proof.

C.3.2 Convergence of time-average multipliers

The proof of Theorem 2 relies on the average dual gap, which we bound here. For this bound, we assume that $\eta_t = \eta = \kappa/\sqrt{T}$ for all t.

From Lemma 6, the dual recursion is

$$\begin{aligned}\|\boldsymbol{\lambda}_{t+1} - \boldsymbol{\lambda}^\star\|^2 + \eta\big[D^\star - g(\boldsymbol{\lambda}_t)\big] + 2\eta\beta D_{\mathrm{KL}}(\mu^\dagger_{\boldsymbol{\lambda}_t}\|\mu^\star) \\ \le\; \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 + \frac{2\eta}{\vartheta}\|\mathbf{b}_t(\boldsymbol{\lambda}_t)\|^2 + \eta^2 R^2.\end{aligned} \tag{77}$$

Using the telescope sum, we get

$$\begin{aligned}\|\boldsymbol{\lambda}_T - \boldsymbol{\lambda}^\star\|^2 + \eta\sum_{t=0}^{T-1}\big[D^\star - g(\boldsymbol{\lambda}_t)\big] + 2\eta\beta\sum_{t=0}^{T-1} D_{\mathrm{KL}}(\mu^\dagger_{\boldsymbol{\lambda}_t}\|\mu^\star) \\ \le\; \|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^\star\|^2 + \frac{2\eta}{\vartheta}\sum_{t=0}^{T-1}\|\mathbf{b}_t(\boldsymbol{\lambda}_t)\|^2 + T\eta^2 R^2.\end{aligned} \tag{78}$$

Choose $\eta = \kappa/\sqrt{T}$. The bias can be bounded by

$$\|\mathbf{b}_t(\boldsymbol{\lambda}_t)\|^2 \;\le\; O\bigg(\chi_t^2 + \delta_t^2 + \frac{1}{T} + \varepsilon^2\frac{(T-t)^2}{T^2} + \frac{\varepsilon^2}{T^2}\bigg). \tag{79}$$

Since the left-side terms in (78) are non-negative, each one of them is bounded above by the right-hand side of (78). Thus, we get a uniform average dual gap of

$$\frac{1}{T}\sum_{t=0}^{T-1}\big[D^\star - g(\boldsymbol{\lambda}_t)\big] \;\le\; \underbrace{\frac{\|\boldsymbol{\lambda}_0-\boldsymbol{\lambda}^\star\|^2}{\eta T}}_{\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)} + \underbrace{\mathcal{O}\left(\frac{1}{T}\sum_{t=0}^{T-1}\left(\chi_t^2+\delta_t^2\right)+\varepsilon^2+\frac{1}{T}\right)}_{\text{floor error}} + \underbrace{\eta R^2}_{\mathcal{O}\left(\frac{1}{\sqrt{T}}\right)}. \tag{80}$$

The rate of convergence of both quantities is $1/\sqrt{T}$ and they converge to a neighborhood determined by the bias.

Define the time-average dual variable $\bar{\boldsymbol{\lambda}} = \frac{1}{T}\sum_t \boldsymbol{\lambda}_t$. By concavity of g, we have

$$D^\star - g\big(\bar{\boldsymbol{\lambda}}\big) \;\le\; D^\star - \frac{1}{T}\sum_{t=0}^{T-1} g(\boldsymbol{\lambda}_t) \;=\; \frac{1}{T}\sum_{t=0}^{T-1}\big[D^\star - g(\boldsymbol{\lambda}_t)\big], \tag{81}$$

which is bounded by (80). Furthermore, by the strong concavity of g, we can also write

$$\|\bar{\boldsymbol{\lambda}}-\boldsymbol{\lambda}^\star\|^2 \;\le\; \frac{2}{\vartheta}\big[D^\star - g(\bar{\boldsymbol{\lambda}})\big]. \tag{82}$$

C.4 Primal Convergence: Proof of Theorem 2

We bound the discrepancy between the terminal distribution of PDI and that of an oracle sampler that knows $\boldsymbol{\lambda}^\star$. The oracle runs the reverse process (7) with the exact scores and the multiplier fixed at $\boldsymbol{\lambda}^\star$, started from the same initialization as PDI. Per the notation introduced after (25), its marginals are $\widetilde{p}_t^{\boldsymbol{\lambda}}$, so that $\widetilde{p}_0^{\boldsymbol{\lambda}^*} = p_0 = \mathcal{N}(\mathbf{0},\mathbf{I})$ and

$$\widetilde{p}_{t+1}^{\boldsymbol{\lambda}^\star} \;=\; \widetilde{p}_t^{\boldsymbol{\lambda}^\star} K_t^{\boldsymbol{\lambda}^\star}. \tag{83}$$

This identity holds by construction, for every step size and noise schedule, because $\widetilde{p}_t^{\boldsymbol{\lambda}^\star}$ is defined as the law of the discrete chain itself rather than as a forward marginal. Let

$$D_t := D_{\mathrm{KL}}\left(p_t \,\middle\|\, \widetilde{p}_t^{\boldsymbol{\lambda}^\star}\right), \tag{84}$$

so that $D_0 = 0$. Under the population update (PDI-D), $\boldsymbol{\lambda}_t$ is fixed at step t. Hence,

$$p_{t+1} \;=\; p_t K_{\boldsymbol{\theta},t}^{\boldsymbol{\lambda}_t}. \tag{85}$$

Define the joint measures

$$P_t(d\mathbf{x}, d\mathbf{y}) = p_t(d\mathbf{x}) K_{\boldsymbol{\theta},t}^{\boldsymbol{\lambda}_t}(d\mathbf{y} \mid \mathbf{x}), \tag{86}$$

$$Q_t(d\mathbf{x}, d\mathbf{y}) = \widetilde{p}_t^{\boldsymbol{\lambda}^\star}(d\mathbf{x}) K_t^{\boldsymbol{\lambda}^\star}(d\mathbf{y} \mid \mathbf{x}). \tag{87}$$

Their $\mathbf{y}$-marginals are p_{t+1} and $\widetilde{p}_{t+1}^{\boldsymbol{\lambda}^\star}$. By data processing and the chain rule for relative entropy,

$$D_{t+1} \le D_t + \mathbb{E}_{\mathbf{x}_t\sim p_t}\left[D_{\mathrm{KL}}\left(K_{\boldsymbol{\theta},t}^{\boldsymbol{\lambda}_t}(\cdot \mid \mathbf{x}_t) \,\middle\|\, K_t^{\boldsymbol{\lambda}^\star}(\cdot \mid \mathbf{x}_t)\right)\right]. \tag{88}$$

The two kernels are Gaussian with covariance $b_{T-t}\mathbf{I}$. Their mean difference is

$$\frac{b_{T-t}}{\sqrt{a_{T-t}}}\left[\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}_t) - \mathbf{s}_t(\mathbf{x}_t \mid \boldsymbol{\lambda}^\star)\right]. \tag{89}$$

Hence,

$$D_{\mathrm{KL}}\left(K_{\boldsymbol{\theta},t}^{\boldsymbol{\lambda}_t}(\cdot \mid \mathbf{x}_t) \,\middle\|\, K_t^{\boldsymbol{\lambda}^\star}(\cdot \mid \mathbf{x}_t)\right) \;=\; \frac{b_{T-t}}{2a_{T-t}}\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}_t) - \mathbf{s}_t(\mathbf{x}_t \mid \boldsymbol{\lambda}^\star)\right\|^2. \tag{90}$$

We decompose

$$\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}_t) - \mathbf{s}_t(\mathbf{x}_t \mid \boldsymbol{\lambda}^\star) = \left[\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}_t) - \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}^\star)\right] + \left[\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t, \boldsymbol{\lambda}^\star) - \mathbf{s}_t(\mathbf{x}_t \mid \boldsymbol{\lambda}^\star)\right]. \tag{91}$$

Using $\|\mathbf{u}+\mathbf{v}\|^2 \le 2\|\mathbf{u}\|^2 + 2\|\mathbf{v}\|^2$, Assumptions 4 and 7, we obtain

$$D_{t+1} \le D_t + \frac{b_{T-t}}{a_{T-t}} \left[\varepsilon^2 + L^2 \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2\right]. \tag{92}$$

Summing (92) from $D_0 = 0$ gives

$$D_{\mathrm{KL}}(p_T \| \widetilde{p}_T^{\boldsymbol{\lambda}^\star}) \le \sum_{t=0}^{T-1} \frac{b_{T-t}}{a_{T-t}} \left[\varepsilon^2 + L^2 \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2\right]. \tag{93}$$

Under Assumption 6,

$$\frac{b_{T-t}}{a_{T-t}} \le \frac{c_b}{T - c_b}, \tag{94}$$

and therefore

$$D_{\mathrm{KL}}(p_T \| \widetilde{p}_T^{\boldsymbol{\lambda}^\star}) \le \frac{c_b T}{T - c_b}\varepsilon^2 + \frac{c_b L^2}{T - c_b} \sum_{t=0}^{T-1} \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2. \tag{95}$$

Since the segment between each iterate and $\boldsymbol{\lambda}^\star$ lies in $\boldsymbol{\Lambda}$, where g is ϑ-strongly concave (Assumption 3), and $\boldsymbol{\lambda}^\star$ maximizes g over $\mathbb{R}^d_+$, we have $\|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|^2 \le \frac{2}{\vartheta}\big[D^\star - g(\boldsymbol{\lambda}_t)\big]$. The average dual-gap bound in (80) then gives

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star\|_2^2 \le O\!\left(\frac{1}{\sqrt{T}}\right) + O\!\left(\frac{1}{T}\sum_{t=0}^{T-1}\left(\chi_t^2 + \delta_t^2\right) + \varepsilon^2 + \frac{1}{T}\right). \tag{96}$$

Substituting the above bound into (95) gives

$$D_{\mathrm{KL}}(p_T \| \widetilde{p}_T^{\boldsymbol{\lambda}^\star}) \le O\!\left(\frac{1}{\sqrt{T}}\right) + O\!\left(\frac{1}{T}\sum_{t=0}^{T-1}\left(\chi_t^2 + \delta_t^2\right) + \varepsilon^2\right), \tag{97}$$

which completes the proof.

Remarks. The comparison with the oracle avoids two issues that a comparison with $\mu^\star$ would face. A discrete Gaussian reverse step does not map $q_\tau(\cdot|\boldsymbol{\lambda}^\star)$ exactly onto $q_{\tau-1}(\cdot|\boldsymbol{\lambda}^\star)$, and $D_{\mathrm{KL}}(p_T\|\mu^\star)$ is infinite because p_T has full support while $\mu^\star$ is supported on $\mathcal{X}$. Both effects are part of the oracle's own error below. Under Assumption 6, $\sum_\tau b_\tau \le c_b$, so $\alpha_T^2 \ge e^{-2c_b}$ for $T \ge 2c_b$. The initialization part of this error therefore does not vanish as T grows with c_b fixed, and it is controlled by the noising horizon c_b.

C.5 Feasibility and Optimality Guarantees

Let $\varepsilon_\star \coloneqq \mathrm{TV}(\Pi_{\mathcal{X}\#}\,\widetilde{p}_T^{\boldsymbol{\lambda}^\star}, \mu^*)$ be the total-variation (TV) distance error of the oracle after projecting its outputs onto $\mathcal{X}$. It stems only from discretization and initialization, is common to every diffusion sampler for μ^* with exact scores, and is not bounded here (Chen et al., 2023; Benton et al., 2024).

Corollary 1 (Expected cost and approximate feasibility)**.** *Let $\widehat{p}_T \coloneqq \Pi_{\mathcal{X}\#}\,p_T$ be the law of the PDI outputs after projection onto $\mathcal{X}$, and let $\delta_T \coloneqq \varepsilon_\star + \sqrt{D_{KL}(p_T\|p_T^\star)/2}$. Then $\mathrm{TV}(\widehat{p}_T, \mu^*) \le \delta_T$, $\mathbb{E}_{\widehat{p}_T}[f_j(\mathbf{x})] \le 2R\,\delta_T$ for every constraint j, and $|\mathbb{E}_{\widehat{p}_T}[f_0(\mathbf{x})] - \mathbb{E}_{\mu^*}[f_0(\mathbf{x})]| \le 2\|f_0\|_\infty\,\delta_T$.*

Proof of Corollary 1. Since μ^* is supported on $\mathcal{X}$ and $\Pi_{\mathcal{X}}$ is the identity on $\mathcal{X}$, $\Pi_{\mathcal{X}\#}\,\mu^* = \mu^*$. The triangle inequality, the data-processing inequality for the total variation distance, and Pinsker's inequality give

$$\mathrm{TV}(\widehat{p}_T, \mu^*) \;\le\; \mathrm{TV}\big(\Pi_{\mathcal{X}\#}\,p_T, \Pi_{\mathcal{X}\#}\,\widetilde{p}_T^{\boldsymbol{\lambda}^\star}\big) + \mathrm{TV}\big(\Pi_{\mathcal{X}\#}\,\widetilde{p}_T^{\boldsymbol{\lambda}^\star}, \mu^*\big) \;\le\; \sqrt{D_{\mathrm{KL}}(p_T\|\widetilde{p}_T^{\boldsymbol{\lambda}^\star})/2} + \varepsilon_\star \;=\; \delta_T. \tag{98}$$

For any bounded measurable h on $\mathcal{X}$, $|\mathbb{E}_{\widehat{p}_T}[h] - \mathbb{E}_{\mu^*}[h]| \le 2\|h\|_\infty\,\mathrm{TV}(\widehat{p}_T, \mu^*)$. Taking $h = f_j$, with $\|f_j\|_\infty \le R$ by Assumption 2 and $\mathbb{E}_{\mu^*}[f_j] \le 0$ by feasibility of μ^*, gives the constraint bound. Taking $h = f_0$ gives the cost bound. □

Note that the corollary controls the expected cost and the expected constraint values, but not the entropy term of the regularized objective, which total variation does not control.

C.6 Additional Lemmas

Lemma 7 (One-step contraction modulus)**.** *Under Assumption 5, for any* $\mathbf{x}$, $\mathbf{y}$ *and under the synchronous coupling (common noise* $\boldsymbol{\epsilon}_t$*), the reverse-step outputs satisfy* $\|\mathbf{X}^+ - \mathbf{Y}^+\|^2 \le \rho_t^2 \|\mathbf{x}-\mathbf{y}\|^2$*, almost surely, with*

$$\rho_t(\boldsymbol{\lambda}^\star) = \frac{1}{\sqrt{a_{T-t}}} \sup_{\mathbf{x}} \left\| \mathbf{I} + b_{T-t} \nabla_{\mathbf{x}}^2 \log q_{T-t}(\mathbf{x}|\boldsymbol{\lambda}^\star) \right\|_{\text{op}}.$$

Proof. Under synchronous coupling, we have

$$\mathbf{X}^+ - \mathbf{Y}^+ = \frac{1}{\sqrt{a_{T-t}}} \Big(\mathbf{x} - \mathbf{y} + b_{T-t} \big(\nabla \log q_{T-t}(\mathbf{x}|\boldsymbol{\lambda}^\star) - \nabla \log q_{T-t}(\mathbf{y}|\boldsymbol{\lambda}^\star) \big) \Big). \tag{99}$$

The noise cancels exactly since the two processes share the same noise. Thus, this is deterministic given $\mathbf{x}$ and $\mathbf{y}$. Then by the mean value theorem along the segment from $\mathbf{y}$ to $\mathbf{x}$, we get

$$\nabla \log q_{T-t}(\mathbf{x}|\boldsymbol{\lambda}^\star) - \nabla \log q_{T-t}(\mathbf{y}|\boldsymbol{\lambda}^\star) \;=\; \Big(\int_0^1 \nabla^2 \log q_{T-t}(\mathbf{y} + s(\mathbf{x}-\mathbf{y})|\boldsymbol{\lambda}^\star) ds \Big) \cdot (\mathbf{x}-\mathbf{y}). \tag{100}$$

Substitute back into (99), we get

$$\mathbf{X}^+ - \mathbf{Y}^+ = \frac{1}{\sqrt{a_{T-t}}} (\mathbf{I} + b_{T-t} \widehat{\mathbf{H}}) \cdot (\mathbf{x}-\mathbf{y}), \tag{101}$$

where $\widehat{\mathbf{H}}$ is the averaged Hessian along the segment. Taking norms,

$$\|\mathbf{X}^+ - \mathbf{Y}^+\| \le \frac{1}{\sqrt{a_{T-t}}} \left\| \mathbf{I} + b_{T-t} \widehat{\mathbf{H}} \right\|_{\text{op}} \cdot \|\mathbf{x}-\mathbf{y}\| \tag{102}$$

$$\le \frac{1}{\sqrt{a_{T-t}}} \sup_{\mathbf{x}} \left\| \mathbf{I} + b_{T-t} \nabla_{\mathbf{x}}^2 \log q_{T-t}(\mathbf{x}|\boldsymbol{\lambda}^\star) \right\|_{\text{op}} \|\mathbf{x}-\mathbf{y}\| =: \rho_t \|\mathbf{x}-\mathbf{y}\|. \tag{103}$$

The last inequality upper bounded the averaged Hessian with the supremum over the compact domain $\mathcal{X}$. This completes the proof. □

Lemma 8. *Under Assumption 2, the score function* $\mathbf{s}_t(\cdot \mid \boldsymbol{\lambda})$ *is Lipschitz in* $\boldsymbol{\lambda}$*, and*

$$\left\| \mathbf{s}_t(\mathbf{x}|\boldsymbol{\lambda}_t) - \mathbf{s}_t(\mathbf{x}|\boldsymbol{\lambda}^\star) \right\| \;\le\; \gamma_t \left\| \boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star \right\|_2, \tag{104}$$

and $\gamma_t \;=\; \frac{R\alpha_{T-t}}{\beta\sigma_{T-t}^2} \sqrt{\sup_{\boldsymbol{\lambda},\mathbf{x}} \|\boldsymbol{Cov}_{\pi_t}(\mathbf{y}_0|\mathbf{x},\boldsymbol{\lambda})\|_{op}} \ge 0.$

Proof. Using the mean value theorem, we can then bound the difference by

$$\left\| \mathbf{s}_t(\mathbf{x}|\boldsymbol{\lambda}_t) - \mathbf{s}_t(\mathbf{x}|\boldsymbol{\lambda}^\star) \right\| \;\le\; \Big(\sup_{\boldsymbol{\lambda}} \left\| \nabla_{\boldsymbol{\lambda}} \mathbf{s}_t(\mathbf{x}|\boldsymbol{\lambda}) \right\|_{\text{op}} \Big) \left\| \boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star \right\|_2, \tag{105}$$

where $\|\cdot\|_{\text{op}}$ is the operator norm. The score function is defined through the Tweedie as

$$\mathbf{s}_t(\mathbf{x}|\boldsymbol{\lambda}) = \frac{1}{\sigma_{T-t}^2} \Big(\alpha_{T-t} \mathbb{E}_{\pi_t}[\mathbf{y}_0|\mathbf{x},\boldsymbol{\lambda}] - \mathbf{x} \Big), \tag{106}$$

where π_t is the posterior distribution:

$$\pi_t(\mathbf{y}_0|\mathbf{x},\boldsymbol{\lambda}) \propto \frac{1}{Z(\boldsymbol{\lambda})} \exp\Big(-E(\mathbf{y}_0,\boldsymbol{\lambda}) - \frac{1}{2\sigma_{T-t}^2} \|\mathbf{x} - \alpha_{T-t}\mathbf{y}_0\|^2 \Big). \tag{107}$$

The posterior depends on $\boldsymbol{\lambda}$ only through the prior. Thus, we have

$$\nabla_{\boldsymbol{\lambda}} \mathbb{E}_{\pi_t}\big[\mathbf{y}_0|\mathbf{x},\boldsymbol{\lambda}\big] \;=\; \frac{-1}{\beta} \mathbf{Cov}_{\pi_t}\Big(\mathbf{y}_0, \mathbf{f}(\mathbf{y}_0) \big| \mathbf{x}, \boldsymbol{\lambda} \Big). \tag{108}$$

Taking the operator norm, we get

$$\frac{\beta\sigma_{T-t}^2}{\alpha_{T-t}} \left\| \nabla_{\boldsymbol{\lambda}} \mathbf{s}_t(\mathbf{x}|\boldsymbol{\lambda}) \right\|_{\text{op}} \;\le\; \Big(\|\mathbf{Cov}_{\pi_t}(\mathbf{y}_0|\mathbf{x},\boldsymbol{\lambda})\|_{\text{op}} \Big)^{1/2} \cdot \Big(\|\mathbf{Cov}_{\pi_t}(\mathbf{f}(\mathbf{y}_0)|\mathbf{x},\boldsymbol{\lambda})\|_{\text{op}} \Big)^{1/2} \tag{109}$$

$$\le\; R \sqrt{\|\mathbf{Cov}_{\pi_t}(\mathbf{y}_0|\mathbf{x},\boldsymbol{\lambda})\|_{\text{op}}}, \tag{110}$$

using Assumption 2. Let $\gamma_t = \frac{R\alpha_{T-t}}{\beta\sigma_{T-t}^2} \sqrt{\sup_{\boldsymbol{\lambda},\mathbf{x}} \|\mathbf{Cov}_{\pi_t}(\mathbf{y}_0|\mathbf{x},\boldsymbol{\lambda})\|_{\text{op}}}$ to complete the proof. □

Lemma 9. *The Lagrangian function in* (2) *is also equivalent to*

$$\mathcal{L}(\mu, \boldsymbol{\lambda}) \;=\; \beta D_{KL}(\mu \| \mu^\dagger_{\boldsymbol{\lambda}}) + g(\boldsymbol{\lambda}).$$

Proof. Start with the definition of KL divergence:

$$\begin{aligned} D_{\mathrm{KL}}(\mu \| \mu^\dagger_{\boldsymbol{\lambda}_t}) \;&=\; \mathbb{E}_\mu\Big[\log \mu(\mathbf{x}) - \log \mu^\dagger_{\boldsymbol{\lambda}_t}(\mathbf{x})\Big] && (111)\\ &=\; \mathbb{E}_\mu[\log \mu(\mathbf{x})] + \frac{1}{\beta}\,\mathbb{E}_\mu\Big[f_0(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{f}(\mathbf{x})\Big] + \log Z(\boldsymbol{\lambda}) && (112)\\ &=\; \frac{1}{\beta}\,\mathcal{L}(\mu, \boldsymbol{\lambda}) + \log Z(\boldsymbol{\lambda}). && (113) \end{aligned}$$

The second equality follows from the definition of $\mu^\dagger_{\boldsymbol{\lambda}_t}$ while the last one uses the definition of the Lagrangian. With the fact that $g(\boldsymbol{\lambda}) = -\beta \log Z(\boldsymbol{\lambda})$, we complete the proof. □

D Dual Training

PDI updates the dual variables during inference, allowing the sampler to react instantaneously to constraint violations. Another alternative is to estimate the optimal dual variables during training while training a score model to sample directly from the optimal distribution. We refer to this approach as dual training (DT). The idea behind DT is related to prior work on constrained diffusion models (Khalafi et al., 2024; 2025). However, this prior work typically considers a single optimization problem, such as fine-tuning a pre-trained model to satisfy a fixed set of constraints, where there is only one optimal dual variable to estimate. In contrast, our formulation considers a family of problem instances, each with its own optimal distribution and corresponding optimal dual variable. In this section, we extend DT to this setting by estimating the dual variables separately for each problem instance during training. The resulting method then serves as an ablation baseline to test the effect of inference-time dual updates.

The dual problem in (3), for a given problem instance $\mathcal{G}$, can be re-written as a bi-level problem,

$$\begin{aligned} D^*(\mathcal{G}) \;&=\; \max_{\boldsymbol{\lambda} \succeq \mathbf{0}}\; \mathcal{L}(\mu^\dagger_{\boldsymbol{\lambda}}, \boldsymbol{\lambda}; \mathcal{G}) \\ &\qquad \text{with}\;\; \mu^\dagger_{\boldsymbol{\lambda}} = \operatorname*{argmin}_{\mu \in \mathcal{P}_2}\; \mathcal{L}(\mu, \boldsymbol{\lambda}; \mathcal{G}). \end{aligned} \tag{114}$$

The inner problem is equivalent to an unconstrained sampling problem and the diffusion process defined in (6)–(7) provides a mechanism to sample from the target distribution $\mu^\dagger_{\boldsymbol{\lambda}}$. Therefore, we replace the inner problem with the problem of training the score model of the reverse process (7). Let $\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}, t, \mathcal{G})$ be the score model, parameterized by $\boldsymbol{\phi}$, and $p_T^{\boldsymbol{\phi}}$ be the terminal distribution of the reverse process under a given dual variable $\boldsymbol{\lambda}$. The parameterized bi-level problem can then be cast as

$$\begin{aligned} D^*_{\boldsymbol{\phi}}(\mathcal{G}) = \max_{\boldsymbol{\lambda} \succeq \mathbf{0}}\; &\mathcal{L}(p_T^{\boldsymbol{\phi}_{\boldsymbol{\lambda}}}, \boldsymbol{\lambda}; \mathcal{G}) \\ &\text{with}\;\; \boldsymbol{\phi}_{\boldsymbol{\lambda}}(\mathcal{G}) = \operatorname*{argmin}_{\boldsymbol{\phi}}\; \mathbb{E}_{t, \mathbf{x}_t}\Big[\omega(t)\big\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t, \mathcal{G}) - \nabla \log q_{T-t}(\mathbf{x}_t | \boldsymbol{\lambda}; \mathcal{G})\big\|_2^2\Big],\; \forall \boldsymbol{\lambda}. \end{aligned} \tag{115}$$

The expectation is with respect to the noisy samples and the diffusion time. The inner problem in its current form assigns a different parametrization $\boldsymbol{\phi}$ for each dual variable $\boldsymbol{\lambda}$ and problem instance $\mathcal{G}$. This is not a training objective and is generally infeasible. To overcome this challenge, we alternate between the two problems so that, for an outer iteration k, the score model is trained only against the most recent dual variable, which is then updated based on the recent constraint violations. More concretely, the training algorithm follows the following dual-ascent iterations:

$$\boldsymbol{\phi}_k \;=\; \operatorname*{argmin}_{\boldsymbol{\phi}}\; \mathbb{E}_{\mathcal{G}, t, \mathbf{x}_t}\Big[\omega(t)\big\|\mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t, t, \mathcal{G}) - \nabla \log q_{T-t}\big(\mathbf{x}_t | \boldsymbol{\lambda}_k(\mathcal{G}); \mathcal{G}\big)\big\|_2^2\Big], \tag{116}$$

$$\boldsymbol{\lambda}_{k+1}(\mathcal{G}) \;=\; \Big[\boldsymbol{\lambda}_k(\mathcal{G}) + \eta_{\mathrm{DT}}\,\mathbb{E}_{p_T^{\boldsymbol{\phi}_k}}\big[\mathbf{f}(\mathbf{x}_T; \mathcal{G})\big]\Big]_+,\; \forall\, \mathcal{G}, \tag{117}$$

Algorithm 3 Dual Training

Initialize $\boldsymbol{\lambda}(\mathcal{G}) \leftarrow \boldsymbol{\lambda}_0$ for all $\mathcal{G}$
Initialize score network $\mathbf{s}_{\boldsymbol{\phi}}$
for $d = 1, \ldots, D$ **do** ▷ Outer: dual iterations
  Initialize replay buffer $\mathcal{B} \leftarrow \varnothing$
  for each problem $\mathcal{G}$ in a batch $\mathcal{B}_{\text{train}}$ **do**
    Rollout: sample $\mathbf{x}_0^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; run inference with $\mathbf{s}_{\boldsymbol{\phi}}$ to obtain $\{(\mathbf{x}_t^{(i)}, t)\}_{t=0}^{T-1},\ \forall i \in [I]$
    Push $\{(\mathbf{x}_t^{(i)}, t, \mathcal{G})\}_{i,t}$ to $\mathcal{B}$
  end for
  for $e = 1, \ldots, E$ **do** ▷ Inner: train at fixed $\boldsymbol{\lambda}$
    Sample a minibatch $\{(\mathbf{x}_t^{(j)}, t^{(j)}, \mathcal{G}^{(j)})\}_{j=1}^{B}$ from $\mathcal{B}$
    for $j = 1, \ldots, B$ **do**
      if $\mathrm{Bernoulli}(\rho_{\text{pert}}) = 1$ **then** ▷ Perturb the sample
        $\mathbf{x}_t^{(j)} \leftarrow \mathbf{x}_t^{(j)} + \epsilon_{\mathbf{x}} \cdot \boldsymbol{z}_x, \quad \boldsymbol{z}_x \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
      end if
    end for
    $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \alpha \nabla \frac{1}{B} \sum_j w(t^{(j)}) \left\| \mathbf{s}_{\boldsymbol{\phi}}(\mathbf{x}_t^{(j)}, t^{(j)}, \mathcal{G}^{(j)}) - \nabla \log q_{T-t^{(j)}}(\mathbf{x}_t^{(j)} \mid \boldsymbol{\lambda}(\mathcal{G}^{(j)}); \mathcal{G}^{(j)}) \right\|_2^2$
  end for
  for each problem $\mathcal{G}$ in a batch $\mathcal{B}_{\boldsymbol{\lambda}}$ **do** ▷ Dual ascent
    Generate samples $\{\mathbf{x}_T^{(i)} \sim p_T^{\boldsymbol{\phi}}\}_{i=1}^{I}$ via full reverse diffusion using the most recent $\boldsymbol{\phi}$
    $\boldsymbol{\lambda}(\mathcal{G}) \leftarrow \left[\boldsymbol{\lambda}(\mathcal{G}) + \eta_{\text{DT}} \sum_i \mathbf{f}(\mathbf{x}_T^{(i)}; \mathcal{G})/I\right]_+$
  end for
end for
return $\boldsymbol{\phi}$

where η_{DT} is the dual step size. The expectation in the training objective becomes jointly over diffusion time, noisy samples, and the problem distribution so that a single $\mathcal{G}$-conditional parameterization learns the score field for a family of problems. The dual variables, however, remain instance-specific. Each problem instance $\mathcal{G}$ maintains its own multiplier $\boldsymbol{\lambda}_k(\mathcal{G})$, updated only from the constraint violations observed for that instance.

Updating the dual variables requires estimating the constraint violations under the current sampler, which in turn requires running the full reverse process to obtain clean samples. Implementing (116)–(117) to completion would therefore be computationally prohibitive. In practice, we use an alternating implementation. We perform a few gradient steps on the score-matching objective in (116), then update the dual variables for a subset of problem instances. Specifically, at each outer iteration, Algorithm 3 samples a batch of instances $\mathcal{B}_{\boldsymbol{\lambda}}$ and updates only their corresponding multipliers, while the remaining dual variables are kept unchanged. As in Algorithm 1, the score target is computed using the MC estimator in (13) and the noisy samples $\mathbf{x}_t$ are obtained from rollouts of the inference dynamics at the beginning of each outer iteration.

At inference time, DT starts from Gaussian noise and runs the reverse diffusion process using the trained score model $\mathbf{s}_{\boldsymbol{\phi}^*}(\mathbf{x}_t, t, \mathcal{G})$. For a test instance $\mathcal{G}$, the reverse dynamics are

$$\mathbf{x}_{t+1} \;=\; \frac{1}{\sqrt{a_{T-t}}}\Big(\mathbf{x}_t + b_{T-t}\mathbf{s}_{\boldsymbol{\phi}^*}\big(\mathbf{x}_t, t, \mathcal{G}\big)\Big) + \sqrt{b_{T-t}}\boldsymbol{\epsilon}_t. \tag{118}$$

No dual variable is initialized, provided to the score model, or updated during this process. The constraint information enters only through the learned parameters $\boldsymbol{\phi}^*$, which were trained using the instance-dependent dual estimates in (116)–(117). Thus, DT produces samples in a single reverse pass and does not adapt the trajectory to constraint violations observed during inference.

E Extended Numerical Results: Mixture of Gaussians

We first validate PDI on the problem of sampling from a weighted mixture of Gaussians in $\mathbb{R}^d$,

$$f_0(\mathbf{x}) \;=\; -\log \sum_{k=1}^{K} w_k \, \mathcal{N}\big(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\big), \tag{119}$$

truncated to a polytope $\{\mathbf{x} : \mathbf{A}\mathbf{x} \le \mathbf{b}\}$. The optimization problem is

$$\min_{\mu(\mathbf{x})} \; \mathbb{E}_{\mathbf{x}\sim\mu}\big[f_0(\mathbf{x})\big] \qquad \text{s.t.} \qquad \mathbb{E}_{\mathbf{x}\sim\mu}\big[\mathbf{A}^\top\mathbf{x} - \mathbf{b}\big] \;\preceq\; \mathbf{0}, \tag{120}$$

i.e., the sampler must produce a distribution that concentrates on high-density regions of the mixture while satisfying M linear inequality constraints in expectation. This is not an everywhere constrained sampling problem, where each sample $\mathbf{x} \in \mathbb{R}^d$ is required to satisfy the constraints. In (10), the constraints are imposed on average over the target distribution. This allows some samples to violate the constraints while others should be strictly feasible to compensate.

E.1 Baseline configuration

Each baseline shares the evaluation seed, batch size, and diffusion discretisation $T = 500$ where applicable.

Unconstrained DM. We run our trained model with $\boldsymbol{\lambda}_t = \mathbf{0}$ for all time steps. This eliminates the constraint term from the energy function, and lets the model sample from the unconstrained Boltzmann density $\exp(-f_0(\mathbf{x})/\beta)$.

PDM. Similarly, we run a reverse process with $\boldsymbol{\lambda}_t = \mathbf{0}, \forall t$. At each step, the Tweedie prediction $\hat{\mathbf{x}}_0$ is projected onto the feasible set before being used in the DDPM posterior,

$$\tilde{\mathbf{x}}_0 \;=\; \Pi_{\mathcal{F}}\big(\hat{\mathbf{x}}_0\big), \qquad \mathcal{F} = \big\{\mathbf{x} : \mathbf{f}(\mathbf{x}) \preceq \mathbf{0},\; j = 1, \dots, M\big\}. \tag{121}$$

PDM then forces feasibility by construction and is therefore a hard-constraint method, unlike PDI.

PDL. Direct Langevin dynamics in the $\mathbf{x}$-space with per-sample dual variables:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_{\mathrm{p}} \, \nabla_{\mathbf{x}_t}\Big(f_0(\mathbf{x}_t) + \boldsymbol{\lambda}_t^\top \mathbf{f}(\mathbf{x}_t)\Big) + \sqrt{2\eta_{\mathrm{P}}}\, \boldsymbol{\epsilon}_t, \tag{122}$$

$$\boldsymbol{\lambda}_{t+1} = \big[\boldsymbol{\lambda}_t + \eta_{\mathrm{d}} \, \mathbf{f}(\mathbf{x}_{t+1})\big]_+, \tag{123}$$

where $\boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$. The main difference between PDL and PDI is that the former operates in the clean sample space. It therefore uses the gradient of the energy function to update $\mathbf{x}$. In PDI, the dynamics evolve under a decreasing noise level, and at each time step, we follow the score of the corresponding denoised distribution.

A distinction arises between these baselines and our method PDI. In all the aforementioned updates, the correction is made to each sample individually, forcing all samples to abide by the constraints. Their solutions are therefore feasible in our problem formulation, but suboptimal because they are more conservative. Figure 5 shows a toy example of the problem with $d = 2, K = 3$, and $M = 2$. Per-sample enforcement sacrifices the opportunity to sample from the high-density regions of the MoG while still satisfying the constraints on average. Instead, it concentrates samples near the boundary of the feasible set, shown as the white region.

In the following table, we provide all the hyperparameters of the baselines:

Method	η_{d}	η_{p}	T	Diffusion Noise Schedule
PDL	10	10^{-3}	500	–
PDM	–	–	500	cosine
Unconstrained	–	–	500	cosine

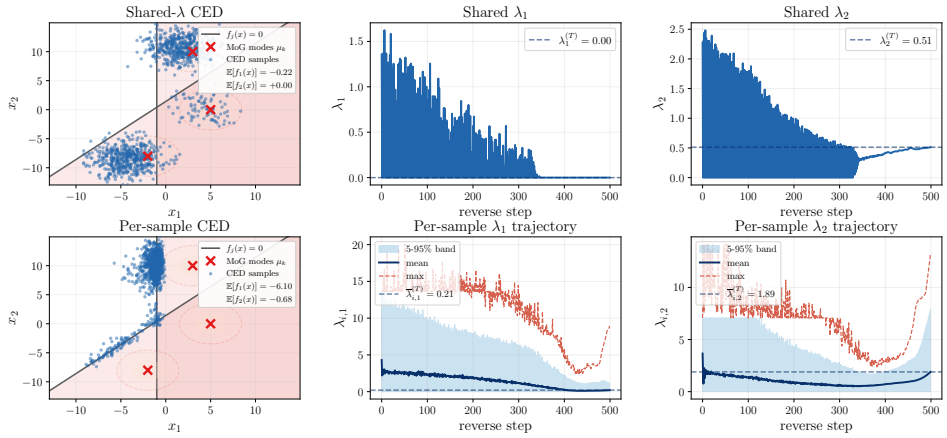

Figure 5: Comparison of **C**onstrained **E**nergy-based **D**iffusion-sampler (CED) in enforcing the constraints in expectation (top) vs per sample (bottom). Per-sample enforcement produces conservative samples that satisfy the constraints individually but fall short in providing a competitive objective.

E.2 Data and Architecture

The problem consists of a K-component Gaussian mixture, truncated by M halfplane constraints. Mode centers are placed uniformly at random in a ball of radius proportional to a spread parameter, with isotropic covariances drawn from a specified range. The constraint normals are chosen so that some halfplanes separate pairs of modes while others cut through mode basins, ensuring that constraints are active near high-density regions. Constraint levels $\mathbf{b}$ are calibrated so that a target fraction of modes have infeasible centers and the remaining feasible modes sit close to constraint boundaries. Since this example is used as a proof of concept, the dataset has only one problem instance.

The score network is an MLP with FiLM conditioning. The input $(\mathbf{x}_t, \boldsymbol{\lambda})$ is projected into a hidden dimension, added to the sinusoidal timestep embedding, and passed through a stack of residual blocks. Each block applies a linear layer, layer normalization with FiLM modulation from the timestep embedding (scale and shift), and a SiLU activation. The output is projected back to $\mathbb{R}^d$ to predict the noise $\boldsymbol{\epsilon}_t$.

E.3 Hyperparameter Choices

We ran a sweep over the hyperparameters: $\beta, T, \eta, \rho, K_{\text{MC}}, \nu_{\max}, \epsilon_{\mathbf{x}}$, and $\epsilon_{\boldsymbol{\lambda}}$, and Table 3 shows the values we use in our experiments. All experiments were run on a single NVIDIA GeForce RTX 3090 card.

E.4 Additional Results

We also record the input-to-output transition matrix and report it in Figure 6. The figure shows that PDL divides the mass between one mode (#10) and the diagonal of the transition matrix. The mass on the diagonal indicates that each input sample typically converges to its nearest mode. In contrast, PDI does not exhibit this behavior. This is partly due to the noise injected along the reverse trajectory, which acts as an annealing mechanism and allows samples to move across modes. It is also due to the entropy regularization in the PDI formulation, controlled by the temperature parameter, which encourages a distributional solution rather than deterministic convergence to the nearest mode. This temperature-driven randomization is absent in PDL.

F Extended Numerical Results: Wireless Resource Allocation

We study stochastic constrained optimization for optimal wireless resource allocation (Uslu et al., 2025; 2026a;b).

Table 3: PDI training hyperparameters for the MoG experiment.

Parameter	Value	Parameter	Value
Problem		*Training*	
Dimension d	30	Optimizer	AdamW
Mixture components K	12	Learning rate	10^{-3}
Constraints M	10	SGD steps	8000 (800×10)
		Batch size B	256
		MC samples K_{MC}	256
		Rollout batch $\lvert\mathcal{B}_{\text{train}}\rvert$	4 graphs
		Exploitation fraction ρ_{exp}	$0 \to 0.7$
Diffusion		*Training – dual ascent rollouts*	
Noise schedule	cosine	Training dual step size η_{train}	0.01
Timesteps T	500	Dual variable cap $\boldsymbol{\lambda}_{\max}$	50
Inverse temperature β^{-1}	50		
Score network		*Training – $\boldsymbol{\lambda}$ prior*	
Architecture	MLP	$\boldsymbol{\lambda}$-prior family	$\exp(\nu)$
Hidden dimension	512	$\nu_{\min}$	0.1
Number of layers	8	$\nu_{\max}$	20.0
Parameters	8.73M		
Conditioning inputs	$(\mathbf{x}_t, t, \boldsymbol{\lambda})$		
Training – perturbation		*Inference*	
Perturbation fraction ρ_{pert}	1.0	Inference dual step size η	1.0
$\mathbf{x}_t$ perturbation std $\epsilon_{\mathbf{x}}$	2.0	Initial $\boldsymbol{\lambda}_0$	$\mathbf{0}$
$\boldsymbol{\lambda}$ perturbation std $\epsilon_{\boldsymbol{\lambda}}$	5.0	Feasibility tolerance	0.02
		Inverse temperature β^{-1}	50

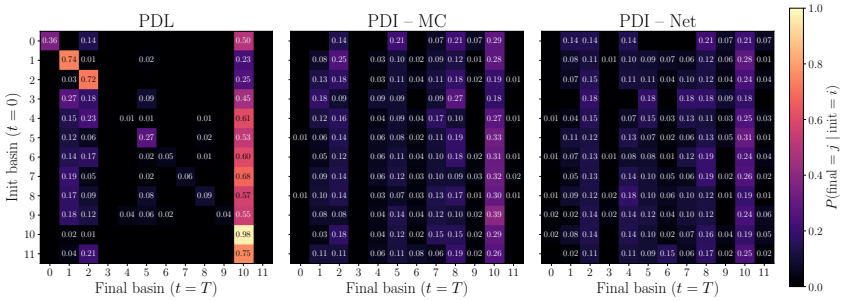

Figure 6: **Initial to final transition matrix.** The matrix is almost diagonal in the PD Langevin case, reflecting the fact that samples converge to their nearest modes.

Consider an ad-hoc wireless network comprised of $N = 200$ transmitter-receiver (tx-rx) pairs deployed uniformly over a square area of side length $R = 5800$ meters, yielding an average density of $\nu = 5.9$ tx-rx pairs/km^2. Each transmitter communicates with a single designated receiver, and each receiver treats incoming signals from all other transmitters as interference. The resource allocation (optimization) variable is the transmit power vector $\mathbf{x} \in [0, P_{\max}]^N$, where $P_{\max} = 10$ mW.

Channel model. The channel model includes both large-scale and small-scale fading effects. We denote by h_{ij} the large-scale channel gain from transmitter i to receiver j, governed by a dual-slope path-loss model with log-normal shadowing of standard deviation $\sigma = 7$ dB. A *network configuration* $\mathbf{H} \in \mathbb{R}_+^{N \times N}$ is the matrix of all large-scale gains $[\mathbf{H}]_{ij} = h_{ij}$, and it is jointly determined by the random geometric deployment of the tx-rx pairs, which fixes the pairwise distances and hence the path-loss components, and the realization of the log-normal shadowing. Each network configuration $\mathbf{H}$ remains fixed over the operating horizon, while the small-scale fading, which follows a Rayleigh distribution with a pedestrian velocity of 1 m/s, produces time-varying instantaneous channel gains $h_{ij,t}$ at each time slot t of duration 50 milliseconds. We set the channel bandwidth to $W = 40$ MHz and the noise power spectral density to $N_0 = -174$ dBm/Hz.

Given an allocation $\mathbf{x}_t$ and fading realization $\mathbf{H}_t$ at time slot t, the instantaneous rate of receiver j is

$$\tilde{r}_j(\mathbf{x}_t, \mathbf{H}_t) = \log_2\left(1 + \frac{[\mathbf{x}_t]_j \cdot |h_{jj,t}|^2}{WN_0 + \sum_{i \neq j}[\mathbf{x}_t]_i \cdot |h_{ij,t}|^2}\right). \tag{124}$$

The ergodic rate of receiver j is obtained by averaging the instantaneous rates over T time slots, i.e., $r_j \approx (1/T)\sum_{t=1}^{T} \tilde{r}_j(\mathbf{x}_t, \mathbf{H}_t)$.

Optimization objective. For a given network state $\mathbf{H}$, the goal is to maximize the ergodic sum-rate subject to per-user minimum ergodic rate constraints. That is,

$$P^*_{\text{pc}} \;=\; \max_{\mu(\mathbf{x})} \; \mathbf{1}^\top \mathbf{r}(\mu, \mathbf{H}) + \beta \mathcal{H}(\mu), \quad \text{s.t.} \quad \mathbf{r}(\mu, \mathbf{H}) \geq \mathbf{1} \cdot r_{\min}, \tag{125}$$

where μ denotes a power allocation policy, and $r_{\min}$ is the minimum ergodic rate requirement. The solution to this problem is rarely a degenerate policy. Because users compete for the same communication resources and interfere with one another, it is generally impossible for all users to transmit at high power simultaneously while satisfying their rate requirements. The optimal policy is therefore often a switching policy, where different users are activated at different times so that each receives enough transmission opportunities to meet its rate constraint on average. This naturally leads to a multimodal optimal distribution, with different modes corresponding to different active-user patterns.

F.1 Data and Architecture

Dataset Generation. We generate a dataset of 256 network configurations in total, viewed as independent samples from the underlying distribution of network configurations induced by the random geometry of tx-rx deployments and random shadowing for the given value of R. A 5:1:2 train-validation-test split results in a training dataset of $|\mathcal{D}| = 160$ networks, $|\mathcal{V}_{\text{al}}| = 32$ validation networks, and $|\mathcal{T}| = 64$ test networks.

Graph representation. We represent each network configuration $\mathbf{H}$ as a directed graph $\mathcal{G}_{\mathbf{H}} = (\mathcal{V}, \mathcal{E}, \mathbf{w})$, where the node set $\mathcal{V} = \{1, \ldots, N\}$ corresponds to the tx-rx pairs. We assign each directed edge $(i, j) \in \mathcal{E}$ a weight proportional to the log-normalized channel gain given by

$$e_{ij} \propto \log_2\left(1 + \frac{P_{\max}|h_{ij}|^2}{WN_0}\right), \tag{126}$$

and sparsify the graph by thresholding the weak edges so that we retain only the top-10 strongest incoming edges per receiver node.

Graph Neural Networks (GNNs). The power allocations $\mathbf{x}$ and ergodic rate vectors $\mathbf{r}$ are naturally interpreted as node-level signals supported on $\mathcal{G}_{\mathbf{H}}$. We therefore use GNNs as the base parameterization for the score model, which takes the problem representation $\mathcal{G}_{\mathbf{H}}$ as input. We implement graph convolutions using Torch Geometric's TAGConv (with K=2). The model consists of L=8 TAGConv residual blocks with hidden dimension d_h=256.

The timestep information is injected through Feature-wise Linear Modulation (FiLM). The dual variable $\boldsymbol{\lambda} \in \mathbb{R}^N$ is concatenated to the input features $\mathbf{x}$.

F.2 Baselines

PDM. We run our trained score model with $\boldsymbol{\lambda}_t = 0$ for all the denoising steps to eliminate the dual ascent. After each step, we perform a projection into the feasible set by solving the augmented-Lagrangian problem:

$$\mathbf{x}_t \leftarrow \arg\min_{\mathbf{w}} \; \|\mathbf{w} - \mathbf{x}_t\|^2 + \tfrac{\zeta}{2}\|[\mathbf{f}(\mathbf{w})]_+\|^2, \tag{127}$$

where $[\mathbf{x}]_+ = \max\{\mathbf{0}, \mathbf{x}\}$. We solve the problem via a few steps of gradient descent.

DPS. We run a reverse process with our trained score model while forcing $\boldsymbol{\lambda}_t = \mathbf{0}, \forall t$, to eliminate the dual ascent. At each step in the inference process, the prediction $\mathbf{x}_t$ is corrected by a gradient step that penalizes constraint violation before it is fed to the next iteration,

$$\mathbf{x}_t \leftarrow \mathbf{x}_t \; - \; s_t \, \nabla_{\mathbf{x}_t} \left\| \mathbf{f}(\mathbf{x}_t) \right\|^2. \tag{128}$$

The gradient is applied directly to the denoised signal at the end of each step, and weighted by a scalar step size s_t. The step size is set to the inverse of the noise variance at each denoising step.

ST & PD Expert Policy. Following the methodology in (Uslu et al., 2026a), for each training network $\{\mathbf{H}^{(b)}\}_{b \in \mathcal{D}}$, we solve the power control problem using an expert algorithm, specifically a dual gradient descent algorithm. The expert algorithm generates trajectories of primal iterates that are near-optimal and feasible in the ergodic sense. We view these iterates as draws from a stochastic, optimal policy, collecting $M = 200$ samples from the convergence regime of the algorithm to obtain an expert dataset of resource allocation vectors $\{\mathbf{x}^{(m)}\left(\mathbf{H}^{(b)}\right)\}_{m=1}^{M}$.

We train a diffusion model to imitate the expert data generated by the PD Expert. The neural network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ is trained via the standard denoising score matching objective and sampled using the deterministic DDIM update:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\, \hat{\mathbf{x}}_0 + \sqrt{1 - \bar{\alpha}_{t-1}}\, \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t). \tag{129}$$

This method has no explicit constraint enforcement mechanism; feasibility depends entirely on the training data distribution and the expressivity of the parameterization.

In order to replicate the performance of (Uslu et al., 2026a), we use the U-Graph Neural Network (U-GNN) parametrization proposed therein, which was designed for graph signal diffusion. This ensures that the DDIM imitation baseline follows the original architectural setup. We do not use U-GNN as the base parametrization for our method because it is computationally heavy. Instead, we use a vanilla GNN, since PDI enforces the constraints through dual ascent during sampling and therefore does not need to rely solely on a highly expressive score parametrization for feasibility.

DT. The training procedure is described in Appendix D. The score model shares the same architecture as PDI-Net, except that it uses three conditioning channels instead of four.

All hyperparameters of these baselines are given in Table 4.

F.3 Additional Results

Dual training (continued). We investigate the difference between PDI and DT more closely by comparing the dual trajectories. Figure 7 plots the cosine similarity and the L_2 distance between the dual variables generated along the DT trajectory and the last dual iterate obtained by the PDI dynamics. These quantities are averaged over the training examples. We use the training dataset because DT does not infer dual variables during inference.

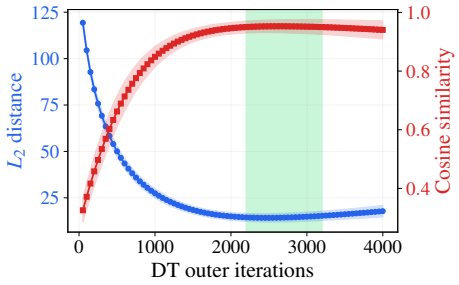

Figure 7: **Dual Training.** Cosine similarity and L_2 distance between the dual variables across the DT training iterations and the last-iterate PDI dual variables. The shaded region indicates the training iterations where the cosine similarity peaks above 0.95. The metrics then drift slightly afterwards.

The cosine similarity reaches its maximum between the epochs 2200 and 3200, after which the DT dual variables begin to drift away from the PDI dual variables. We therefore save all DT checkpoints after epoch 2200, evaluate them on the test set and report the resulting metrics in Figure 8. The figure shows that all DT checkpoints achieve higher mean rates (objective) than PDI, shown by the dashed blue line. This suggests that the models offer higher mean rates by reducing interference through more conservative power allocations. This reduction in interference benefits some users but sacrifices tail performance. This is reflected in the lower fifth and first percentile rates and feasibility percentages across all checkpoints.

These observations suggest that the dual variables generated by DT do not induce the same primal solutions obtained by PDI dynamics. This can be attributed to the fact that the PDI score field is

Table 4: Baseline hyperparameters for the wireless allocation experiment. The DT settings differ from PDI in Table 6; all other problem, diffusion, architecture, and evaluation settings are identical.

Parameter	Value	Parameter	Value
ST		*PDL*	
Backbone	UGNN (Uslu et al., 2026a)	Iterations	500
Base channels / levels	64 / 4	Primal learning rate η_p	10^{-4}
Graph filter	$K = 2$ hops	Dual learning rate η_d	1
Layers per level	2	Initial value $\boldsymbol{\lambda}_0$	0
Timesteps T	500		
Noise schedule	Linear		
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.98$)		
Learning rate / weight decay	10^{-4} / 10^{-4}		
LR schedule	Cosine		
Batch size	960		
Conditioning	3 channels $(\mathbf{x}, t, \mathcal{G})$		
Epochs	5 000		
PDM		*DPS*	
Base model	PDI-Net with $\boldsymbol{\lambda} = 0$	Base model	PDI-Net with $\boldsymbol{\lambda} = 0$
Projection iterations	100		
Projection step size	0.05		
Regularization ζ	30		
DT			
Score model	Same PDI-Net architecture	Outer iterations D	4 000
Conditioning channels	3 $(\mathbf{x}, t, \mathcal{G})$	Inner SGD steps E	10
Dual input	none	Minibatch size $\lvert\mathcal{B}_{\text{train}}\rvert$	4 problems
Samples per network I	20	Initial $\boldsymbol{\lambda}_0$	10
$\boldsymbol{\lambda}$ update batch size $\lvert\mathcal{B}_{\boldsymbol{\lambda}}\rvert$	16 problems	Dual learning rate η_{DT}	0.1

Table 5: **DT after extra training at fixed dual variables (DT+).** We continue training several DT checkpoints with no dual updates and report the best two results (corresponding to iterations 2200 and 2900). PDI achieves comparable performance with much lighter training.

Method	Mean rates (↑)	5th%ile rate (↑)	1st%ile rate (↑)	Mean viol. (↓)	Sum viol. (↓)
PDI–Net	$2.803_{\pm 0.132}$	$0.613_{\pm 0.042}$	$0.424_{\pm 0.082}$	$0.0062_{\pm 0.0027}$	$1.25_{\pm 0.54}$
DT+ (2200)	$2.814_{\pm 0.128}$	$0.656_{\pm 0.051}$	$0.426_{\pm 0.112}$	$0.0060_{\pm 0.0029}$	$1.19_{\pm 0.59}$
DT+ (2900)	$2.822_{\pm 0.128}$	$0.615_{\pm 0.045}$	$0.422_{\pm 0.090}$	$0.0063_{\pm 0.0028}$	$1.25_{\pm 0.57}$

conditioned on the current dual variable $\boldsymbol{\lambda}_t$ while that of DT is an unconditional score field obtained after training over a history of changing dual variables. Therefore, even when the DT dual variable is close to the PDI dual variable, the checkpoint at that epoch does not need to approximate the score field associated with that particular dual variable. The reason is that the checkpoint represents the accumulated effect of the preceding training trajectory, not a score model explicitly indexed by the current dual variable.

To further separate the quality of the learned dual variables from the quality of the corresponding score model, we continue training several DT checkpoints for extra 1000 outer iterations while freezing their dual variables. We refer to the resulting models as DT+. This experiment asks whether the dual variables produced by DT are useful once the score model is given enough optimization time to adapt to them. Table 5 reports the two best DT+ models and shows a substantial improvement over the original DT checkpoints. This indicates that DT can reach informative dual variables, but the score model at the corresponding checkpoint has not necessarily learned the score field induced by that dual value. Rather, because the model is updated only partially while the dual variables evolve, the checkpoint reflects the accumulated effect of the preceding training trajectory. In this sense, practical DT behaves as a score model shaped by a history of dual variables, and not as a fully optimized score model for the current dual variable. The improvement of DT+ also comes with a much larger training cost (4000 original outer iterations in addition to extra 1000 iterations for tuning the model), whereas PDI reaches comparable performance using only 400 training outer iterations.

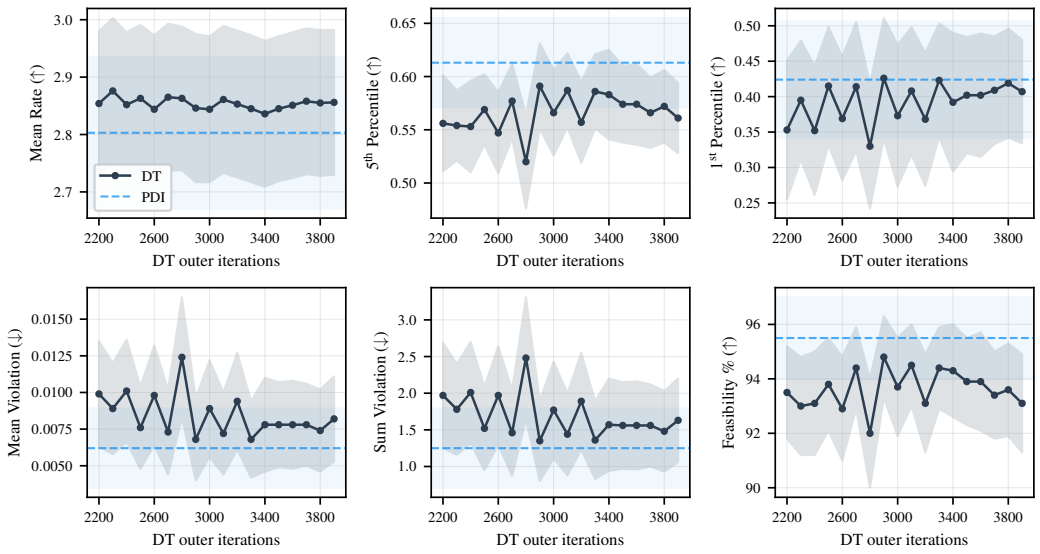

Figure 8: **Performance of DT checkpoints along the training trajectory.** The DT models fail to attain the same 5th percentile rates, mean violations and feasibility percentage as PDI. All the metrics are evaluated over the test set.

The improvement of DT+ shows that the dual variables produced by DT are informative. However, it does not remove the limitation of using an unconditional score model. Since DT does not take the dual variable as input, the correction induced by the dual trajectory is absorbed into the score-model parameters rather than represented explicitly. This can work well for problem instances whose required correction is close to the one encoded by the trained model. At inference, however, new problem instances may induce different optimal multipliers or different constraint residuals along the denoising trajectory. The DT score field cannot adapt to these instance-dependent corrections during sampling, because the dual variable is not provided to the model.

PDI reduces this limitation by learning a $\boldsymbol{\lambda}$-conditioned family of score fields. During inference, the current dual iterate is passed directly to the score model, so each reverse step uses a score field matched to the dual correction used at that step. This does not mean that PDI samples from the Gibbs distribution of one fixed dual variable, since the final sample is produced by a sequence of kernels conditioned on different dual iterates. Rather, the advantage is that the constraint correction and denoising dynamics remain coupled along the trajectory, allowing constraint violations to influence subsequent denoising steps more directly.

Temperature effect. In Figure 9, we analyze the effect of the temperature parameter β. Large β, equivalently small β^{-1}, generate more uniform samples, which cause severe interference and in turn the lowest mean rates and feasibility. Lower temperatures produce more degenerate distributions that concentrate the samples at the corners. This prevents severe interference and provides better mean rate, but does not fully benefit from time sharing (i.e., low feasibility). We find that $\beta = 1$ in this experiment balances the mean rates with feasibility.

Noise annealing effect. PDI differs from PDL because it follows the score fields of a sequence of progressively denoised distributions, rather than the score field of the clean target distribution. This mechanism is reminiscent of simulated annealing. Early noisy diffusion steps smooth the target landscape and promote exploration across modes, while later low-noise steps gradually sharpen the distribution and refine the samples toward the constrained target. Unlike classical simulated annealing, however, the annealing in PDI is induced by the diffusion noise schedule rather than by the temperature of the energy function. Figure 10 shows this effect, represented in the diverse samples generated by PDI compared with the more concentrated samples produced by PDL.

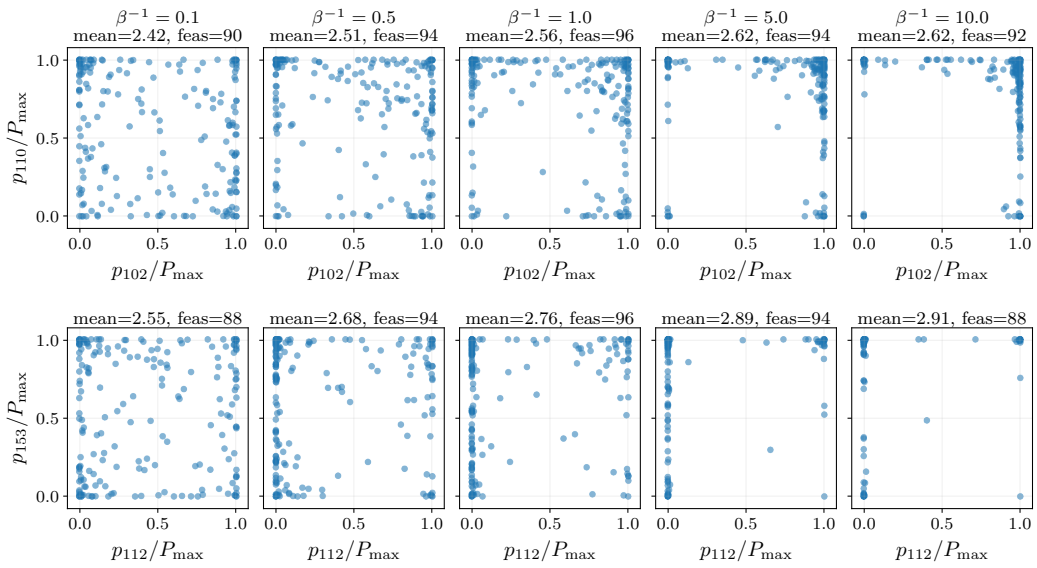

Figure 9: **Temperature effect.** Two-dimensional scatter plots of generated samples of two test networks. The title shows the mean rates and feasibility of the network. Higher temperature (left) promotes more diverse allocations while lower temperature (right) generates more degenerate samples. The value $\beta = 1$ provides a good balance between mean rates and feasibility.

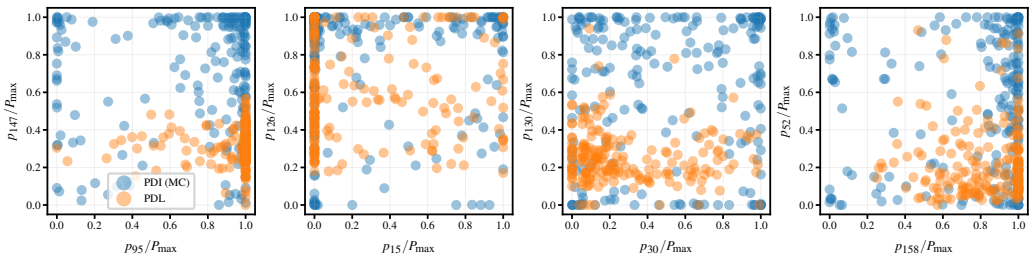

Figure 10: **Annealing effect of PDI.** A scatter plot of the predicted power allocation of four pairs of agents. PDI discovers more transmission modes, allowing more agents to transmit with higher power to achieve higher rates on average. PDL concentrates the samples in the same modes.

Table 6: PDI hyperparameters for the wireless power allocation experiment.

Parameter	Value	Parameter	Value
Problem setup		*Training*	
Number of users N	200	Optimizer	AdamW
Max transmit power $P_{\max}$	10 dBm	Learning rate	10^{-3} ($\to 10^{-5}$)
Bandwidth	40 MHz	Weight decay	10^{-4}
Minimum rate $r_{\min}$	0.6 bps/Hz	Outer iterations N_{outer}	400
Number of networks	256		
Score network		*Training*	
Hidden dimension	256	Inner SGD steps N_{inner}	$10 \to 30$
GNN layers	8	Batch size $\lvert\mathcal{B}_{\text{train}}\rvert$	4 problems
Filter order K	2	Replay buffer capacity $\lvert\mathcal{B}\rvert$	4 096
Time embedding dimension	128	Minibatch size B	128
Conditioning channels	4 $(\mathbf{x}, t, \boldsymbol{\lambda}, \mathcal{G})$		
Activation / normalization	SiLU / LayerNorm		
Diffusion process		*Training – perturbation and prior*	
Timesteps T	500	Exploitation fraction ρ_{exp}	0→0.7
Noise schedule	Cosine	$\boldsymbol{\lambda}$ prior range $[\nu_{\min}, \nu_{\max}]$	$[2, 10]$
MC candidates K_{MC}	8	Perturbation std $(\epsilon_{\mathbf{x}}/\epsilon_{\boldsymbol{\lambda}})$	$0.1/0.5$
		Perturbation fraction ρ_{pert}	0.5
Dual variable (inference)		*Evaluation*	
Inverse temperature β^{-1}	1.0	Policies per network K	200
Step size η_t	0.05	Time-sharing timeslots R_{eval}	500
Initialization $\boldsymbol{\lambda}_0$	10.0	Test networks	64
Clamp $\boldsymbol{\lambda}_{\max}$	50	Evolution trials	50
Sub-batch groups C	10		

F.4 Hyperparameters Choices

Table 6 collects the hyperparameters of the problem and PDI training. All experiments were run on a single NVIDIA GeForce RTX 3090 card.

G Extended Numerical Results: Portfolio Management

We consider constrained portfolio optimization over $N = 500$ assets organized into 10 sectors, with returns generated from a latent factor model. The goal is to learn a distribution $\mu(\mathbf{x})$ over portfolio weights $\mathbf{x} \in \Delta^{N-1}$ that maximizes expected return while satisfying per-asset risk, sector-exposure, and diversification constraints on average:

$$\begin{aligned} \max_{\mu(\mathbf{x})} \quad & \mathbb{E}_{\mathbf{x}\sim\mu}\left[\frac{1}{R}\sum_{k=1}^{R}\mathbf{r}^{(k)\top}\mathbf{x}\right] + \beta H(\mu) \\ \text{s.t.} \quad & \mathbb{E}_{\mathbf{x}\sim\mu}\left[x_j(\boldsymbol{\Sigma}\mathbf{x})_j\right] \le b_j, && j = 1,\dots,N, \\ & \underline{e}_s \le \mathbb{E}_{\mathbf{x}\sim\mu}\left[\sum_{j\in\mathcal{S}_s} x_j\right] \le \overline{e}_s, && s = 1,\dots,S, \\ & \mathbb{E}_{\mathbf{x}\sim\mu}\left[N_{\text{eff}}(\mathbf{x})\right] \ge N_{\text{eff}}^{\min}, \end{aligned} \tag{130}$$

where

$$N_{\text{eff}}(\mathbf{x}) = \exp\left(-\sum_{j=1}^{N} x_j \log x_j\right). \tag{131}$$

The first set of constraints limits the average variance contribution of each asset. The second bounds the average exposure to each sector, while the last promotes diversification by imposing a minimum effective number of assets on average. Altogether, the problem contains 521 inequality constraints.

Table 7: Baseline hyperparameters in the portfolio management problem.

Parameter	Value	Parameter	Value
PD-Langevin			
Primal step size η_{p}	10^{-2}	Iterations	500
Dual step size η_{d}	100		
PDM			
Projection iterations	50	$\boldsymbol{\lambda}$ (net input)	0
PDI–MC			
Inverse temperature β^{-1}	2 000	dual step size η_t	300
MC candidates K_{MC}	512	$\boldsymbol{\lambda}_{\text{decay}}$	0.001
Evaluation			
Generated samples	1024	Test instances	40

In practice, portfolio construction is rarely a single-solution problem. Even under common market assumptions, different clients or rebalancing instances may require different allocations due to heterogeneous preferences, liquidity needs, tax considerations, and model uncertainty. This motivates learning a distribution over portfolios rather than a single deterministic allocation. The average risk constraints allow the sampler to trade risk across different portfolio samples, admitting occasional high-risk, high-return allocations while ensuring that the overall exposure of each asset remains within its prescribed budget in expectation.

We note that $\mathcal{X}$ is the probability simplex in $\mathbb{R}^N$, and volume, uniform distribution, and entropy are taken with respect to its $(N-1)$-dimensional volume.

G.1 Data & Architecture

Data. We generate synthetic portfolio instances using a latent factor model with sector structure. Each instance consists of $N = 500$ assets organized into $S = 10$ sectors. Each asset's log-returns are drawn from a multivariate normal with a sector-structured covariance matrix $\boldsymbol{\Sigma}$, where assets within the same sector are more correlated with each other than with assets in other sectors. We draw 1000 return scenarios and convert to simple returns via exponentiation.

Graph construction. To construct the graph, we compute the absolute correlation matrix from $\boldsymbol{\Sigma}$, remove self-loops, and for each asset retain edges to its k-nearest neighbors (we use k=20) by absolute correlation magnitude. Edge weights are normalized by the largest eigenvalue of the resulting adjacency matrix.

Conditioning and architecture. We use the same GNN architecture used in the wireless allocation problem. The timestep information is injected through Feature-wise Linear Modulation (FiLM). The shared dual variable $\boldsymbol{\lambda} \in \mathbb{R}^N$ is concatenated to the input features. The network comprises L=6 residual blocks with hidden dimension d_h=128.

G.2 Baselines

The Hyperparameters of the baselines are summarized in Table 7.

G.3 Additional Results

We ran additional experiments to evaluate different aspects of our algorithm, and the results match our previous findings.

First, we compare the full PDI algorithm with variants that freeze the dual variable after an initial primal-dual run and report the results in Figure 11. In this experiment, we consider the last dual iterate and the time-averaged dual variable. The results show that the time-averaged multiplier performs better than the last iterate. However, both fixed-dual variants remain below the full PDI performance. We also notice that the initial $\boldsymbol{\lambda}_0$ does not affect the PDI trajectory. This suggests that PDI is not

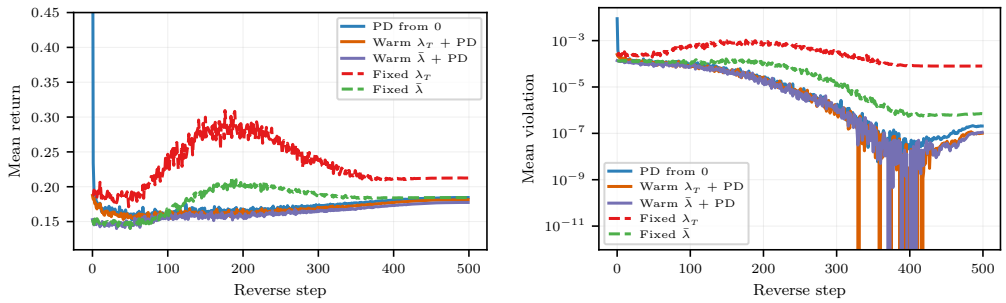

Figure 11: **Performance of PDI while fixing λ along the diffusion steps.** The time-average dual variable gives better objective and constraint satisfaction compared to the last iterate. However, running the dual updates beats both of them and does not depend on the initial value.

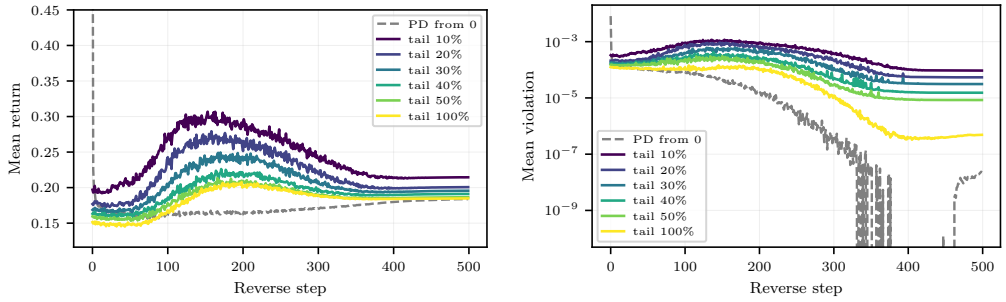

Figure 12: **Frozen $\bar{\lambda}_{\text{tail}}$ during sampling.** The effect of running the sampler with a fixed dual variable equal to the time-average of the tail dual iterates.

simply estimating one good multiplier and then sampling from it. Its advantage comes from updating the multiplier throughout the denoising trajectory.

Second, we test in Figure 12 whether using only a late-stage average of the dual trajectory is enough. The answer is negative. The performance degrades when the average is computed only over a suffix of the trajectory. This suggests that the early dual iterates are not merely transient noise; they contribute to the effective multiplier that governs the sampled distribution.

Our third experiment in Figure 13 investigates the role of the entropy regularization on the final solutions and trajectories. The figure shows that when the temperature β is high, the sampler produces more diverse samples, but the objective can be weaker. When the temperature is lower, the distribution becomes more concentrated around high-quality solutions. This improves the objective, but it can also make the dynamics sharper and more sensitive.

Finally, the noise-schedule comparisons of Figure 14 show that the schedule is important for convergence. Schedules that keep the process noisy for longer, such as the standard cosine and DDPM linear schedules, give more stable behavior and better feasibility. In contrast, schedules that reconstruct the clean sample too early make the sampler sensitive before the dual variable has stabilized. This leads to larger violations and weaker convergence.

G.4 Hyperparameter Choices

We ran a sweep over β and η_t. Table 8 displays all hyperparameters used in experiments. All experiments were run on a single NVIDIA GeForce RTX 3090 card.

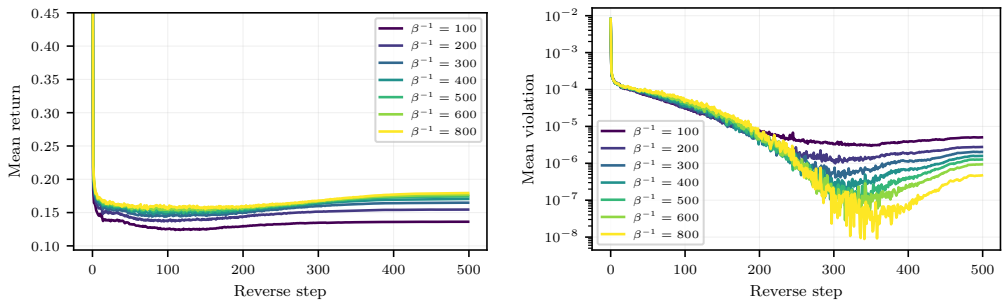

Figure 13: **Temperature effect.** Lower temperature β, equivalently larger β^{-1} (yellow), helps improve returns and constraint satisfaction.

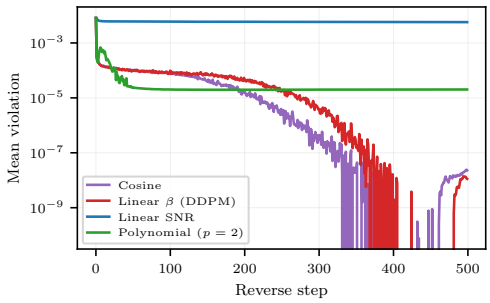

Figure 14: **Effect of noise schedules.** Standard DDPM cosine and linear schedules give better feasibility. Schedules that reconstruct the clean sample too early make the sampler sensitive to perturbations in the dual variable before the dual variable has stabilized.

Table 8: PDI hyperparameters in the portfolio management problem (training and inference).

Parameter	**Value**	**Parameter**	**Value**
Diffusion process		*Training*	
Timesteps T	500	Optimizer	AdamW
Noise schedule	Cosine	Learning rate	3×10^{-4}
Inverse temperature β^{-1}	1	λ prior $[\nu_{\min}, \nu_{\max}]$	$[0.001, 60]$
MC candidates K_{MC}	512	λ perturbation (mult.) ϵ_λ	0.5
		Exploitation ratio $\rho_{\exp}$	$0 \to 0.7$
		Minibatch size B	128
		Training instances	200
Dual variable (inference)		*Training*	
Step size η_t	0.01	Validation instances	5
Initialization $\boldsymbol{\lambda}_0$	7	Inner SGD steps N_{inner}	$10 \to 80$
Clamp $\lambda_{\max}$	500	Outer iterations N_{outer}	800
Weight decay	0.0	Replay buffer capacity	8 192
Constraint normalization	None	Rollout minibatch $\lvert\mathcal{B}_{\mathrm{train}}\rvert$	4 problems
		Rollout batch size	1 024
Score network			
Backbone	GNN		
Hidden dimension d_h	128		
Residual blocks L	6		
Graph filter order K	1		
Graph sparsity (top-k)	20		
Conditioning	FiLM + $\boldsymbol{\lambda}$ concat		