Constrained Reward-Directed Diffusion via Primal–Dual Continuous-Time q-Learning with Score Signals

Liz Lemma Future Detective

January 19, 2026

Abstract

Diffusion models are increasingly deployed under hard constraints: safety, policy compliance, distribution shift limits, and domain-validity checks. Building on recent work that formulates reward-directed diffusion as continuous-time reinforcement learning by treating the unknown score as the control action, we introduce a constrained version of this framework. Our objective maximizes a terminal reward while regularizing deviation from the unknown true score (a KL/path-divergence proxy) and enforcing explicit constraints on the generated samples, such as bounded unsafe content probability or bounded feature-space shift. Since the running reward depends on the unknown score, we retain the data-driven score-signal construction via a ratio estimator, producing a reinforcement signal without learning the score function. We then develop a primal–dual actor–critic little q-learning algorithm: the critic learns a Lagrangian value via martingale residual minimization, the actor learns the Gaussian policy mean, and dual variables adaptively tune constraint penalties. For linear function approximation, we provide (i) almost-sure convergence to a projected saddle point under standard stochastic approximation conditions and (ii) finite-episode bounds on suboptimality and constraint violation, with an explicit decomposition into (a) function approximation error, (b) score-signal noise, and (c) discretization error. In an LQG diffusion special case we also show matching lower bounds of order $\Omega(\varepsilon^{-2})$ episodes under noisy terminal feedback. Experiments (recommended) on conditional image generation with modern safety reward models would validate constraint satisfaction and the reward–fidelity trade-off without relying on pretrained reference diffusion models.

Table of Contents

2. 2. Background: score-based diffusion SDEs/ODEs; entropy-regularized continuous-time RL; score-as-action formulation; martingale characterization of optimality; ratio score-signal estimator.

3. 3. Constrained Problem Formulation: terminal constraints (expectation constraints), optional path constraints; Lagrangian and dual; discussion of fidelity regularization vs constraint set; examples (unsafe probability, coverage, feature shift).

4. 4. Score-Signal and Constraint-Signal Oracles: construction of running reward signal from ratio estimator; assumptions on noise/bias; terminal constraint observations; practical surrogate choices (FID proxy, classifier risk, CLIP-based unsafe score).

5. 5. Primal–Dual Martingale q-Learning Algorithm: actor/critic parameterization; Gaussian policy structure; projected dual updates; pseudocode; stabilization tricks (dual clipping, entropy annealing, critic regularization).

6. 6. Theory in the Linear Setting: (i) saddle-point characterization; (ii) almost sure convergence under SA; (iii) finite-episode bounds; (iv) discretization and score-signal error decomposition; (v) discussion of extension to NTK regime.

7. 7. Lower Bounds and Hardness: reduction from constrained stochastic bandits to show $\Omega(\varepsilon^{-2})$ episodes needed for feasibility/optimality under noisy terminal feedback; implications for any black-box reward/constraint oracle.

8. 8. Implementation Notes and Recommended Experiments: constrained Swiss-roll and synthetic LQG; image generation with safety classifier constraints; ablations on dual stepsize/projection radius; evaluation metrics (FID, constraint violation, reward).

9. 9. Extensions: conditional diffusion with context C; ODE samplers; multiple constraints and Pareto fronts; robust constraints under distribution shift; privacy-preserving score signals.

10. 10. Limitations and Open Problems: deep-net theory beyond linear/NTK; better constraint surrogates; off-policy/replay; adaptive discretization; non-Gaussian policy families.

1 Introduction and Motivation

By 2026, the dominant failure modes of generative models are no longer primarily those of distributional mismatch or insufficient capacity, but rather those of misalignment between what a user or downstream system intends and what the model produces under deployment constraints. In many domains, we do not merely seek samples from a learned data distribution; we seek samples that optimize a task-dependent terminal objective while satisfying explicit safety, compliance, and resource constraints. Typical examples include: meeting specification thresholds (e.g. toxicity, privacy leakage, watermark detectability), respecting fairness or demographic parity limits, enforcing physical feasibility in design problems, and adhering to strict failure-probability budgets in planning. These constraints are naturally expressed as expectation constraints of the form $\mathbb{E}[c_i(Y)] \leq \tau_i$ over the generated output Y, rather than as hard per-sample filters, since both the constraints and their evaluations are noisy and may only be accessible through expensive black-box tests.

Existing alignment mechanisms for diffusion models tend to fall into two broad categories, each with structural limitations in the constrained setting. The first category is *guidance* (e.g. classifier guidance or reward-model guidance), which modifies sampling by adding a gradient term derived from an auxiliary model. While guidance can be effective when the auxiliary gradient is accurate and well-calibrated, it is intrinsically an unconstrained method: it trades off reward and sample quality through a scalar guidance weight without offering certificates of feasibility for multiple constraints. Moreover, when the terminal feedback is obtained from black-box evaluators (human preference, simulator-based scoring, proprietary APIs), guidance is unavailable because gradients are not accessible. Even when a differentiable surrogate is trained, distribution shift arises because the guided sampler visits states far from those seen during surrogate training, and the resulting feedback gradients can be systematically biased.

The second category is *pretrain-then-finetune*. Here one first trains an unconditional diffusion model to approximate the data distribution, and then finetunes either the score network or an auxiliary policy using preference optimization or reinforcement learning objectives. This paradigm inherits the classical difficulties of offline-to-online transfer: finetuning typically relies on samples from an evolving policy and thus departs from the pretraining distribution, but offline objectives do not directly control the induced state distribution. For constrained problems, one additionally requires a principled mechanism for trading off reward against multiple constraints. In practice, scalarization heuristics are used (weighted sums of penalty terms), which are brittle: the correct penalty weights depend on unknown sensitivity of constraint satisfaction to policy changes, and there is no reason to expect a fixed set of weights to simultaneously ensure feasibility and near-optimality.

Finally, both guidance and finetuning approaches commonly assume access to global surrogates (a global reward model, a global constraint model, or a global score model), yet in many applications the only reliable signal is a local, query-based evaluation on generated samples.

We therefore adopt a viewpoint in which reward-directed diffusion sampling is treated as an instance of constrained, entropy-regularized continuous-time reinforcement learning with partial observability of the true running reward. The essential modeling choice is to interpret the *substitute score* in the reverse-time drift as the control action. This yields a controlled diffusion whose terminal state y_T is the generated sample, with an objective that combines (i) a terminal reward $h(y_T)$ accessible only through noisy black-box queries and (ii) a running penalty that measures deviation from the (unknown) optimal score. The constraints are imposed on terminal costs $c_i(y_T)$, each observed with noise at the end of an episode. The resulting problem is formally a constrained stochastic control problem in continuous time, but its structure is not generic: the control enters linearly in the drift, the exploration distribution is naturally Gaussian due to entropy regularization, and the running reward is quadratic in the control with a time-dependent weight induced by the diffusion coefficients.

This structure suggests that we should not attempt to learn a global score function $\nabla \log p_t$ as an intermediate object. Instead, we exploit the fact that the diffusion forward process admits tractable conditionals $p_{t|0}(\cdot \mid x_0)$ and that we have sample access to p_0. From these, one can construct local *ratio estimators* that provide a noisy proxy for the score value at the visited states, thereby producing a per-step reward signal $\hat{r}$ without fitting a global score network. The algorithmic consequence is that training can proceed on-policy, with data access only through minibatch-based ratio computations at the encountered states, and without requiring gradients of the terminal reward or constraints.

Our first contribution is a precise constrained optimization formulation for reward-directed diffusion sampling under this information pattern. We use a Lagrangian relaxation with dual variables $\lambda \in \mathbb{R}^m_+$ and incorporate the terminal constraints into a shaped terminal reward. The resulting entropy-regularized control problem admits a particularly simple optimal policy class: for each fixed λ, the optimal exploratory policy is Gaussian with a covariance determined by the diffusion coefficient and the entropy temperature, and with a mean equal to the object that must be learned. This reduction converts policy optimization into learning a mean function $\mu(t, y)$, rather than learning an arbitrary action distribution, and it clarifies the role of entropy as fixing exploration at a scale compatible with the diffusion dynamics.

Our second contribution is an on-policy primal–dual algorithm, PD-CTQL, which combines (i) a martingale characterization of optimality for continuous-time q-learning with (ii) projected stochastic dual ascent. The critic is trained to satisfy an orthogonality condition derived from the Doob–

Meyer decomposition of the controlled process, implemented through a discretized residual loss over rollouts. The actor update is coupled to the critic through the entropy-regularized relation between the q-function and the policy, which yields a tractable gradient update for the Gaussian mean. Constraints are handled by dual variables updated using terminal observations only; projection ensures dual feasibility and provides stability under noise.

Our third contribution is a set of guarantees that match the operational requirements of constrained alignment: we aim simultaneously for near-optimality and approximate feasibility under noisy terminal feedback and noisy running signals. In the linear function approximation setting (linear actor mean and linear critic), and under standard bounded-moment and bounded-noise conditions, we show that the coupled primal–dual stochastic approximation converges almost surely to a stationary point of the projected Lagrangian saddle system. The proof follows the two-timescale method: on the fast timescale, the primal iterates track the solution of the martingale-based fixed-point conditions for a quasi-static dual vector; on the slow timescale, the dual variables perform projected ascent on the empirical constraint violations. The projection radius plays the usual role of enforcing compactness and enabling uniform bounds required by stochastic approximation theory.

Beyond asymptotic convergence, we provide finite-episode performance bounds in the same linear setting. With constant stepsizes tuned as $\Theta(N^{-1/2})$ over N episodes, the averaged iterates achieve a primal–dual gap and an expected constraint violation that decay as $\tilde{O}(N^{-1/2})$, up to additive terms that isolate three sources of unavoidable error: (i) approximation error due to function class restriction, (ii) signal error due to ratio-estimator noise and terminal oracle noise, and (iii) discretization error due to the time grid used for simulation (e.g. Euler–Maruyama). This decomposition is essential for practice: it separates what can be improved by more training episodes from what requires better estimators, richer function classes, or smaller time steps.

Finally, we address the question of whether these rates are intrinsic. In a special linear–quadratic–Gaussian diffusion instance with a single noisy terminal constraint, we prove a minimax lower bound: any algorithm that only observes terminal reward and constraint samples must use at least $\Omega(\varepsilon^{-2})$ episodes to reach ε-optimality and ε-feasibility with constant probability. The reduction embeds a constrained mean-estimation problem into the terminal-time decision, and standard information-theoretic arguments show that one cannot beat the ε^{-2} scaling under sub-Gaussian noise. This lower bound indicates that, even before accounting for score-signal estimation or discretization, the episodic sample complexity of constrained terminal feedback imposes a fundamental limit; consequently, the $\tilde{O}(N^{-1/2})$ guarantees obtained by PD-CTQL are rate-optimal in the relevant regime.

In summary, we propose a framework in which constrained alignment for diffusion sampling is treated as a principled constrained control problem with a clear information model, an on-policy algorithm that does not require global surrogate training, and provable convergence and finite-time behavior under standard assumptions. The remainder of the paper develops the necessary background on score-based diffusions and entropy-regularized continuous-time reinforcement learning, establishes the martingale characterization that underlies our critic update, introduces the ratio-based score-signal estimator used to form running rewards, and then presents the primal–dual analysis culminating in the aforementioned upper and lower bounds.

2 Background: diffusions, entropy-regularized control, and local score signals

Forward diffusion and tractable conditionals. We consider a forward diffusion $(x_t)_{t\in[0,T]}$ on $\mathbb{R}^d$ defined by the linear SDE

$$dx_t \;=\; f(t)\, x_t\, dt \;+\; g(t)\, dW_t, \qquad x_0 \sim p_0, \tag{1}$$

where f, g are known scalar functions (the standard VP/OU setting; extensions to matrix-valued coefficients are routine). Let p_t denote the marginal law of x_t. A key structural feature is that the transition density $p_{t|0}(\cdot \mid x_0)$ is Gaussian and known in closed form. Writing the (scalar) fundamental solution as $\alpha(t) := \exp(\int_0^t f(s)\, ds)$ and the accumulated variance as $\sigma^2(t) := \int_0^t \alpha(t)^2 \alpha(s)^{-2} g(s)^2\, ds$, we have

$$x_t \,\big|\, x_0 \;\sim\; \mathcal{N}\big(\alpha(t)\, x_0,\; \sigma^2(t)\, I\big), \qquad p_t(x) \;=\; \int_{\mathbb{R}^d} p_{t|0}(x \mid x_0)\, p_0(dx_0). \tag{2}$$

We emphasize that p_0 is unknown and is accessed only through i.i.d. samples; nonetheless, (26) gives an explicit likelihood model $x_0 \mapsto p_{t|0}(x \mid x_0)$ which will later be exploited to construct local score signals.

Reverse-time dynamics and the score. The time-reversed process associated with (1) admits the classical (Haussmann–Pardoux) reverse SDE

$$dy_t \;=\; \Big(f(T{-}t)\, y_t - g(T{-}t)^2\, \nabla \log p_{T-t}(y_t)\Big)\, dt + g(T{-}t)\, d\bar{W}_t, \qquad y_0 \sim p_T, \tag{3}$$

where $\bar{W}_t$ is a Brownian motion in reverse time. The unknown object is the score $\nabla \log p_t(\cdot)$, which is typically approximated by a neural network trained by denoising score matching. In contrast, our development treats the score value as an *implicit* quantity that may be queried locally (with noise) rather than globally approximated.

For completeness, we record the corresponding probability flow ODE, obtained by removing diffusion while preserving marginals:

$$\frac{d}{dt} y_t \;=\; f(T-t)\, y_t \;-\; \frac{1}{2} g(T-t)^2\, \nabla \log p_{T-t}(y_t), \tag{4}$$

which motivates deterministic samplers. Our subsequent algorithmic and analytical statements are phrased for SDE sampling, but discretized ODE samplers may be handled analogously, with modified discretization error terms.

Score-as-action: controlled reverse dynamics. We now introduce the central modeling step: we interpret a *substitute score* a_t as a control input, and we consider the controlled reverse-time diffusion

$$dy_t \;=\; \big(f(T-t)\,y_t \;+\; g(T-t)^2\,a_t\big)\,dt \;+\; g(T-t)\,dW_t, \qquad y_0 \sim \nu, \tag{5}$$

where ν is a Gaussian reference distribution approximating p_T (or otherwise chosen to initialize sampling). Comparing (5) with (3), we see that the choice

$$a_t^\star(y) \;=\; -\nabla \log p_{T-t}(y) \tag{6}$$

recovers the true reverse drift. Hence, learning to sample from the data distribution can be framed as learning a feedback control law $a_t \approx a_t^\star$, without committing to a global parametric estimator of $\nabla \log p_t$.

This viewpoint becomes particularly convenient once we introduce reward shaping: we penalize deviations from the optimal action (6) through a quadratic running reward

$$r(t,y,a) \;:=\; -g(T-t)^2\,\big\|\nabla \log p_{T-t}(y) - a\big\|^2, \tag{7}$$

which is maximal (equal to 0) when a matches the true score. In later sections, this running term is combined with an application-dependent terminal reward; for the present background discussion, (7) merely provides the canonical continuous-time signal that ties control to diffusion fidelity.

Entropy-regularized continuous-time RL and Gaussian policies. We adopt an entropy-regularized control formulation in which exploration is encouraged through a temperature parameter $\theta > 0$. For a Markov policy $\pi(\cdot \mid t,y)$ over actions $a \in \mathbb{R}^d$, we may view the objective as maximizing expected cumulative reward augmented by (negative) relative entropy at each time. In the present linear-in-drift, quadratic-in-action setting, the entropy-regularized instantaneous optimization induces a Gaussian form. Concretely, suppose we seek policies that maximize, at each state (t,y), an entropy-regularized Hamiltonian of the schematic form

$$\sup_{\pi(\cdot|t,y)} \Big\{ \mathbb{E}_{a\sim\pi}\big[\langle g(T-t)^2 a,\; \nabla_y J(t,y)\rangle \;+\; \tilde{r}(t,y,a)\big] \;+\; \theta\,\mathcal{H}(\pi(\cdot \mid t,y)) \Big\}, \tag{8}$$

where J is a value function, $\mathcal{H}$ is differential entropy, and $\tilde{r}$ contains terms quadratic in a. Completing the square shows that the optimizer is Gaussian with mean proportional to the gradient term and covariance proportional to θ divided by the quadratic weight. In our case, this yields the parametrization

$$\pi_\psi(a \mid t, y) \;=\; \mathcal{N}\Big(\mu_\psi(t, y),\; \frac{\theta}{2g(T-t)^2} I\Big), \tag{9}$$

which is the policy class used throughout: the diffusion coefficients dictate the exploration scale, while learning is reduced to estimating the mean function μ_ψ.

Martingale characterization and the q-function. A distinctive aspect of continuous-time RL is that Bellman equations are naturally expressed via martingale properties. Let $J(t, y)$ denote a candidate value function for (5) under a given policy, and let $q(t, y, a)$ denote the corresponding (entropy-regularized) q-function, normalized so that $\pi = \exp(q/\theta)$ (up to the log-partition function). For sufficiently smooth J, Itô's formula implies that along a controlled trajectory (y_t, a_t),

$$dJ(t, y_t) \;=\; \Big(\partial_t J(t, y_t) + \langle f(T-t)y_t + g(T-t)^2 a_t,\; \nabla J(t, y_t)\rangle + \tfrac{1}{2} g(T-t)^2 \Delta J(t, y_t)\Big)\, dt + g(T-t)\langle\nabla J(\tag{10}$$

The martingale characterization used by continuous-time q-learning (in the spirit of Doob–Meyer decompositions) asserts that, at optimality and with an appropriate definition of q, the drift part of the compensated process vanishes. Operationally, this yields an orthogonality (zero-mean residual) condition of the form

$$\mathbb{E}\Big[J(t+\Delta t, y_{t+\Delta t}) - J(t, y_t) \;+\; r(t, y_t, a_t)\,\Delta t \;-\; q(t, y_t, a_t)\,\Delta t \;\Big|\; \mathcal{F}_t\Big] \;\approx\; 0, \tag{11}$$

with an additional terminal term when t is the final grid point. This is precisely the type of condition that can be enforced by a squared-residual loss over rollouts, avoiding the need to estimate the infinitesimal generator terms in (10) explicitly. In discretized implementations, we simulate (5) by Euler–Maruyama,

$$y_{k+1} \;=\; y_k + \big(f(T-t_k)y_k + g(T-t_k)^2 a_k\big)\Delta t + g(T-t_k)\sqrt{\Delta t}\,\xi_k, \qquad \xi_k \sim \mathcal{N}(0, I), \tag{12}$$

and we regress parameters so that the empirical counterpart of (11) is small.

Local ratio estimators for score values. The running reward (7) depends on $\nabla \log p_{T-t}(y)$, which is unknown. We therefore require a mechanism that, given a queried point (t, y), returns a noisy but informative proxy for the score *value at that point*, using only samples from p_0 and knowledge of $p_{t|0}$. From (26), we can differentiate under the integral sign to obtain the identity

$$\nabla \log p_t(y) \;=\; \frac{\nabla p_t(y)}{p_t(y)} \;=\; \frac{\int \nabla_y p_{t|0}(y \mid x_0)\, p_0(dx_0)}{\int p_{t|0}(y \mid x_0)\, p_0(dx_0)} \;=\; \frac{\int p_{t|0}(y \mid x_0)\, \nabla_y \log p_{t|0}(y \mid x_0)\, p_0(dx_0)}{\int p_{t|0}(y \mid x_0)\, p_0(dx_0)}. \tag{13}$$

Thus $\nabla \log p_t(y)$ is an expectation of $\nabla_y \log p_{t|0}(y \mid X_0)$ under the posterior density proportional to $p_{t|0}(y \mid x_0)p_0(x_0)$. Given a minibatch $\{x_0^{(j)}\}_{j=1}^M$ of i.i.d. samples from p_0, we form self-normalized weights

$$w_j(t,y) \;:=\; \frac{p_{t|0}(y \mid x_0^{(j)})}{\sum_{\ell=1}^M p_{t|0}(y \mid x_0^{(\ell)})}, \qquad \sum_{j=1}^M w_j(t,y) = 1, \tag{14}$$

and define the ratio (self-normalized importance sampling) estimator

$$\widehat{\nabla \log p}_t(y) \;:=\; \sum_{j=1}^M w_j(t,y)\, \nabla_y \log p_{t|0}(y \mid x_0^{(j)}). \tag{15}$$

In the OU/VP case, $\nabla_y \log p_{t|0}(y \mid x_0)$ is explicit, since $p_{t|0}(\cdot \mid x_0)$ is Gaussian with mean $\alpha(t)x_0$ and covariance $\sigma^2(t)I$:

$$\nabla_y \log p_{t|0}(y \mid x_0) \;=\; -\frac{1}{\sigma^2(t)}\big(y - \alpha(t)x_0\big). \tag{16}$$

Substituting (15) into (7) yields a computable running signal $\hat{r}(t,y,a) = -g(T-t)^2\|\widehat{\nabla \log p}_{T-t}(y) - a\|^2$. The estimator (15) is generally biased due to self-normalization, but its bias and conditional second moments can be controlled under standard effective-sample-size conditions; our later convergence and finite-time analyses assume precisely such bounded-bias / bounded-variance properties for $\hat{r}$ along the on-policy state distribution.

The overarching point is that (15) provides *local* score information only at states that the current policy visits. This is aligned with the on-policy nature of the control problem: rather than learning $\nabla \log p_t$ on the entire space-time domain, we obtain an online reward signal sufficient to drive actor–critic updates through the martingale residual condition (11).

3 Constrained problem formulation: terminal and pathwise requirements

We now formalize the reward-directed sampling problem as a constrained, entropy-regularized control task posed on the controlled reverse dynamics (5). The control variable a_t plays the role of a substitute score, and the running reward (7) enforces diffusion fidelity by penalizing deviation from the (unknown) score. On top of this fidelity term, we introduce application-dependent terminal objectives and constraints that encode semantic or safety requirements on the generated sample y_T.

Terminal reward and expectation constraints. Let $h : \mathbb{R}^d \to \mathbb{R}$ be a measurable terminal reward function, which may be non-smooth and is accessed only through noisy evaluations at sampled terminal states.[1] For $m \geq 1$, let $c_i : \mathbb{R}^d \to \mathbb{R}$ be measurable terminal cost functions with thresholds $\tau_i \in \mathbb{R}$. We seek a policy π (equivalently, a mean function μ within the Gaussian class (9)) that solves

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{\pi}\left[\beta\, h(y_T) \;+\; \int_0^T r(t, y_t, a_t)\, dt\right] \\ \text{s.t.} \quad & \mathbb{E}_{\pi}[c_i(y_T)] \;\leq\; \tau_i, \qquad i = 1, \ldots, m, \end{aligned} \tag{17}$$

where $y_0 \sim \nu$ and the expectation is taken over trajectories induced by (5) and $a_t \sim \pi(\cdot \mid t, y_t)$. The constraints in (17) are *expectation constraints.* This choice is deliberate: it accommodates stochastic terminal observations, admits a tractable Lagrangian relaxation, and aligns with the fact that many generation constraints are naturally expressed in terms of population averages (e.g., average risk, average shift in a feature statistic, average rate of unsafe content).

Optional path constraints. In some applications, terminal feasibility alone is insufficient: one may wish to regulate intermediate states of the reverse diffusion, or penalize excessive control magnitude, or enforce temporal safety requirements. We therefore record an optional extension in which we introduce measurable running constraint costs $\ell_i : [0, T] \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and require

$$\mathbb{E}_{\pi}\left[\int_0^T \ell_i(t, y_t, a_t)\, dt\right] \;\leq\; \tau_i, \qquad i = 1, \ldots, m_{\text{path}}. \tag{18}$$

The subsequent algorithmic development is simplest in the purely terminal case (17), because constraint feedback arrives once per episode and therefore couples naturally to a dual ascent update. Nonetheless, (18) fits the same primal–dual template (one merely replaces $c_i(y_T)$ by $\int_0^T \ell_i\, dt$ in the dual update), and the distinction is primarily notational at the level of the saddle-point formulation.

Fidelity regularization versus explicit constraints. It is important to separate the role of the diffusion-fidelity term $r(t, y, a) = -g(T-t)^2 \|\nabla \log p_{T-t}(y) - a\|^2$ from the role of the constraint set. The running reward r is not an application constraint; rather, it regularizes the control so that the induced terminal law remains close to the data-generating reverse dynamics. In particular, since $r \leq 0$ with maximum 0, maximizing $\int_0^T r\, dt$ discourages policies that

[1] We allow h to be a black-box metric such as a proxy for perceptual quality, a classifier score, or a downstream utility; smoothness is not assumed since our updates will be driven by martingale residuals rather than direct differentiation of h.

"cheat" by steering y_t to satisfy h or the constraints while departing drastically from the reverse-time flow implied by p_0. The constraints, by contrast, encode *what* we want at terminal time and may in general conflict with strict fidelity to p_0 (e.g., filtering out unsafe generations necessarily alters the distribution). The parameter β and the constraint thresholds τ_i govern this trade-off: large β pushes toward terminal utility; tight τ enforces feasibility; the fidelity term resists degenerate solutions by imposing a control-theoretic notion of proximity to the score-driven dynamics.

Lagrangian relaxation and projected dual variables. We handle the constraints in (17) via a Lagrangian saddle-point formulation. For $\lambda \in \mathbb{R}_+^m$, define the (terminal-shaped) Lagrangian objective

$$\mathcal{L}(\pi, \lambda) \;:=\; \mathbb{E}_\pi\left[\int_0^T r(t, y_t, a_t)\, dt \;+\; \beta\, h(y_T) \;-\; \sum_{i=1}^m \lambda_i \big(c_i(y_T) - \tau_i\big)\right]. \quad (19)$$

The constrained problem (17) may be viewed (formally) as

$$\sup_\pi \inf_{\lambda \in \mathbb{R}_+^m} \mathcal{L}(\pi, \lambda), \quad (20)$$

with the understanding that strong duality may require additional regularity beyond our black-box setting; in the algorithmic development we instead target a stationary point of the projected saddle dynamics in the function-approximation class. In implementation and analysis, we maintain a projection $\lambda \in [0, \Lambda]^m$ for some $\Lambda < \infty$. This compactness plays two roles: it enforces stability of the stochastic approximation iterates, and it provides a meaningful certificate of constraint tightness (large λ_i indicates persistent difficulty in satisfying constraint i). We emphasize that the dual shaping affects only the *terminal* signal in (19); the diffusion-fidelity term remains unchanged, hence the quadratic-in-action structure that yields Gaussian optimal policies is preserved (cf. Theorem 1).

Constraint examples in generative modeling. We briefly list canonical constraints that fit (17).

(i) Unsafe content probability (chance-type constraint). Suppose a black-box detector outputs an unsafe indicator $u(y) \in \{0, 1\}$ (or a calibrated probability in $[0, 1]$). A natural requirement is that the rate of unsafe generations is at most $\delta \in (0, 1)$:

$$\mathbb{E}_\pi[u(y_T)] \;\leq\; \delta. \quad (21)$$

This is an expectation constraint on an indicator, hence a form of chance constraint. When u is discontinuous or noisy, we may replace it by a measurable surrogate score $\tilde{u}(y) \in [0, 1]$ (e.g., a sigmoid-transformed classifier logit), yielding the same mathematical form while improving statistical efficiency.

(ii) Coverage or diversity constraints. To prevent mode collapse toward high-reward but narrow regions, one may enforce a coverage constraint relative to a reference distribution ρ on a feature space. For a feature map $\varphi : \mathbb{R}^d \to \mathbb{R}^p$, consider constraints on feature moments such as

$$\left\| \mathbb{E}_\pi[\varphi(y_T)] - \mathbb{E}_{Z\sim\rho}[\varphi(Z)] \right\| \;\leq\; \epsilon, \tag{22}$$

which can be encoded as two-sided expectation constraints by introducing costs $c_{j,+}(y) = \varphi_j(y)$ and $c_{j,-}(y) = -\varphi_j(y)$ with appropriate thresholds. Alternatively, coverage over a finite set of prompts or categories can be enforced by defining $c_i(y)$ as the negative indicator of membership in a desired subset, or as a distance-to-set penalty. The common feature is that coverage constraints are naturally statistical and are therefore well aligned with the expectation form.

(iii) Feature shift / distribution matching constraints. In conditional generation or dataset editing, one may wish to constrain the deviation of certain attributes from a baseline (e.g., demographic or style statistics). Let $\phi : \mathbb{R}^d \to \mathbb{R}$ measure an attribute. One may enforce bounds on mean shift,

$$\mathbb{E}_\pi[\phi(y_T)] \;\leq\; \tau, \tag{23}$$

or on second moments $\mathbb{E}_\pi[\phi(y_T)^2] \leq \tau'$, or on a collection of linear functionals of an embedding representation. These constraints are particularly compatible with the primal–dual scheme because the dual update reduces to tracking empirical averages of $c_i(y_T)$.

Entropy regularization and the constrained max-ent objective. Although our policy class is fixed to the Gaussian family (9), it is conceptually helpful to view the control problem as entropy-regularized: exploration is controlled by the temperature θ, and the induced q-function normalization $\pi = \exp(q/\theta)$ provides an analytic bridge between the actor mean μ_ψ and critic quantities. In the constrained setting, entropy plays an additional stabilizing role: by preventing overly concentrated policies early in training, it reduces variance in terminal constraint estimation and mitigates premature collapse to a potentially infeasible deterministic control.

Information pattern and oracle access. Problem (17) is defined under a restricted information pattern: the score $\nabla \log p_t$ is unknown, and both $h(y_T)$ and $c_i(y_T)$ may be observed only through noise. Consequently, the Lagrangian (19) cannot be evaluated exactly, and the running reward term cannot be computed without an estimator. Our algorithm will therefore rely on two kinds of signals: (a) a *local* score-based proxy used to construct a running reward estimate $\hat{r}(t, y, a)$ along visited states, and (b) episodic terminal observations of $h(y_T)$ and $c_i(y_T)$ used to form unbiased (or bounded-bias) estimates of the Lagrangian terminal term and the constraint violations.

We make these oracle models explicit next, since they are the only interface through which the learning dynamics interacts with the unknown data distribution and the black-box objectives.

4 Score-signal and constraint-signal oracles

Our learning dynamics interacts with the unknown data distribution and with the black-box terminal objectives only through stochastic *signals*. In this section we specify (i) a score-signal oracle used to construct a running reward estimate along visited states, and (ii) terminal reward/constraint oracles returning noisy episode-end observations. We state these oracles at a level of generality sufficient for the stochastic-approximation analysis in later sections.

A ratio-based score signal from the dataset. Fix $t \in [0, T]$. Under the forward SDE, the marginal at time t satisfies the mixture representation

$$p_t(y) \;=\; \int_{\mathbb{R}^d} p_{t|0}(y \mid x)\, p_0(x)\, dx, \tag{24}$$

where $p_{t|0}(\cdot \mid x)$ is known and Gaussian. Differentiating (24) under the integral and using Bayes' rule yields the classical conditional-score identity

$$\nabla_y \log p_t(y) \;=\; \mathbb{E}\big[\nabla_y \log p_{t|0}(y \mid X_0) \,\big|\, X_t = y\big]\,. \tag{25}$$

Since $p_{t|0}(\cdot \mid x)$ is Gaussian, $\nabla_y \log p_{t|0}(y \mid x)$ is affine in (y, x) and is available in closed form. Concretely, for each t there exist a matrix $M_t \in \mathbb{R}^{d\times d}$ and a positive definite covariance $\Sigma_t \in \mathbb{R}^{d\times d}$ (both known from f, g) such that

$$p_{t|0}(y \mid x) \;=\; \mathcal{N}(M_t x, \Sigma_t), \qquad \nabla_y \log p_{t|0}(y \mid x) \;=\; -\Sigma_t^{-1}(y - M_t x). \tag{26}$$

Substituting (26) into (25) shows that the unknown score $\nabla \log p_t(y)$ is determined by the unknown posterior mean $\mathbb{E}[X_0 \mid X_t = y]$:

$$\nabla \log p_t(y) \;=\; -\Sigma_t^{-1}\Big(y - M_t\, \mathbb{E}[X_0 \mid X_t = y]\Big). \tag{27}$$

We approximate the posterior expectation in (27) using only i.i.d. samples $\{x^{(j)}\}_{j=1}^n \sim p_0$. For a given query point y and time t, define unnormalized importance weights

$$w_j(t, y) \;:=\; p_{t|0}(y \mid x^{(j)}), \qquad j = 1, \ldots, n, \tag{28}$$

and normalized weights $\bar{w}_j(t, y) := w_j(t, y)/\sum_{\ell=1}^n w_\ell(t, y)$ (with the convention that we add a small numerical $\varepsilon_{\text{den}} > 0$ to the denominator if needed). Then the self-normalized ratio estimator of the posterior mean is

$$\widehat{m}_t(y) \;:=\; \sum_{j=1}^n \bar{w}_j(t, y)\, x^{(j)} \;\approx\; \mathbb{E}[X_0 \mid X_t = y]. \tag{29}$$

Plugging (29) into (27) produces our score signal

$$\widehat{s}_t(y) \;:=\; -\Sigma_t^{-1}\big(y - M_t\,\widehat{m}_t(y)\big). \tag{30}$$

Equivalently, using (25) and linearity of expectation, one may write $\widehat{s}_t(y) = \sum_{j=1}^{n} \bar{w}_j(t,y)\,\nabla_y \log p_{t|0}(y \mid x^{(j)})$, emphasizing that we are estimating the conditional expectation in (25) by a weighted empirical average.

In practice we do not take n to be the full dataset size; instead we evaluate (30) on a fresh minibatch $\mathcal{B} = \{x^{(j)}\}_{j=1}^{n_{\mathrm{mb}}}$ drawn i.i.d. from the dataset each time a score signal is required. We denote the resulting estimator by $\widehat{s}_t(y;\mathcal{B})$ to make the randomness explicit. This “local” estimator is only queried at states (t, y) visited by the current policy and is not intended as a global score model.

Running reward signal. Recall that the ideal running reward at reverse time t depends on $\nabla \log p_{T-t}$:

$$r(t, y, a) \;=\; -g(T-t)^2 \big\|\nabla \log p_{T-t}(y) - a\big\|^2.$$

We define the on-trajectory reward signal by replacing the unknown score with (30):

$$\widehat{r}(t, y, a; \mathcal{B}) \;:=\; -g(T-t)^2 \big\|\widehat{s}_{T-t}(y;\mathcal{B}) - a\big\|^2. \tag{31}$$

Two remarks are important.

First, even if $\widehat{s}_{T-t}(y;\mathcal{B})$ were unbiased for $\nabla \log p_{T-t}(y)$, the quadratic map $u \mapsto -g^2\|u-a\|^2$ would introduce Jensen-type bias in $\widehat{r}$. Our subsequent algorithm and analysis therefore treat $\widehat{r}$ as a generic noisy signal with controlled moments rather than as an unbiased reward.

Second, because $p_{t|0}$ is Gaussian, $w_j(t, y)$ can underflow in high dimension or at small noise levels. Numerically stable evaluation uses log-weights $\log w_j(t, y)$ and the log-sum-exp trick. To avoid rare but extreme weight concentration (which inflates variance in (29)), one may apply standard stabilizers such as weight clipping $\bar{w}_j \leftarrow \min\{\bar{w}_j, w_{\max}\}$ followed by renormalization, or a tempered likelihood w_j^{κ} with $\kappa \in (0, 1]$. These modifications alter the bias/variance trade-off but fit our signal model below as long as the induced second moments remain bounded.

Signal model and bounded-moment assumptions. We now state the oracle properties required later. Along a policy-induced trajectory (y_t, a_t), let $\mathcal{F}_t$ denote the filtration generated by the rollout up to time t together with the internal randomness used to form previous minibatches. We postulate:

- *Score-signal error control:* there exist constants $b_s \ge 0$ and $\sigma_s^2 < \infty$ such that for all $t \in [0, T]$ and all states y encountered under all iterates,

$$\big\|\mathbb{E}[\widehat{s}_t(y;\mathcal{B}) - \nabla \log p_t(y) \mid y]\big\| \;\le\; b_s, \qquad \mathbb{E}\big[\|\widehat{s}_t(y;\mathcal{B}) - \mathbb{E}[\widehat{s}_t(y;\mathcal{B}) \mid y]\|^2 \mid y\big] \;\le\; \sigma_s^2. \tag{32}$$

This accommodates both unbiased estimation ($b_s = 0$) and controlled bias (e.g., due to minibatching, clipping, or tempering).

- *Running reward signal control:* there exist $b_r \geq 0$ and $\sigma_r^2 < \infty$ such that, for all visited (t, y, a),

  $$\left|\mathbb{E}[\widehat{r}(t,y,a;\mathcal{B}) - r(t,y,a) \mid y,a]\right| \;\leq\; b_r, \qquad \mathbb{E}\left[(\widehat{r}(t,y,a;\mathcal{B}) - \mathbb{E}[\widehat{r}(t,y,a;\mathcal{B}) \mid y,a])^2 \mid y,a\right] \;\leq\; \sigma_r^2. \tag{33}$$

  A sufficient condition for (33) is (32) together with uniform second-moment bounds on the action and the score (ensured by the fixed policy covariance and the moment-control assumption on trajectories).

We emphasize that (32)–(33) are *local* conditions: they are required only along the state distribution generated by the iterates, not uniformly over all $y \in \mathbb{R}^d$.

Terminal reward and terminal constraint oracles. At the end of each episode, upon observing y_T, we obtain noisy samples of the terminal reward and constraint costs. We model this as

$$\widehat{h} \;=\; h(y_T) + \xi_h, \qquad \widehat{c}_i \;=\; c_i(y_T) + \xi_{c,i}, \quad i = 1, \ldots, m, \tag{34}$$

where $(\xi_h, \xi_{c,1}, \ldots, \xi_{c,m})$ are conditionally zero-mean given y_T (or, more generally, have bounded conditional bias) and have bounded conditional second moments:

$$\left|\mathbb{E}[\xi_h \mid y_T]\right| \leq b_h, \quad \mathbb{E}[\xi_h^2 \mid y_T] \leq \sigma_h^2, \qquad \left|\mathbb{E}[\xi_{c,i} \mid y_T]\right| \leq b_{c,i}, \quad \mathbb{E}[\xi_{c,i}^2 \mid y_T] \leq \sigma_{c,i}^2. \tag{35}$$

This oracle captures measurement noise (e.g., stochastic downstream evaluation), Monte Carlo estimation error internal to h or c_i, or randomized detectors. It also permits deterministic but unknown h, c_i by setting $\xi \equiv 0$.

Surrogate choices for practical objectives and constraints. The framework requires only that h and c_i be measurable and observable through (34). In applications, one typically chooses surrogates that trade semantic relevance against statistical efficiency.

(i) FID-like or distributional-quality surrogates. While the Fréchet Inception Distance is not naturally a single-sample terminal function (it is a two-sample functional of distributions), one can introduce episodic surrogates by comparing a generated sample y_T to a running estimate of reference feature statistics. Let $\phi : \mathbb{R}^d \to \mathbb{R}^p$ be an embedding (e.g., an Inception feature). Maintaining reference mean μ_{ref} and covariance Σ_{ref}, one may define

$$h(y) \;=\; -\|\phi(y) - \mu_{\text{ref}}\|^2 \quad \text{or} \quad c(y) \;=\; \|\phi(y) - \mu_{\text{ref}}\|^2,$$

or use mini-batch episodes in which the terminal feedback aggregates across the B trajectories to estimate a distributional discrepancy. Such choices fit our oracle model because the resulting terminal signal is still a noisy scalar evaluation per episode (possibly computed from the episode batch).

(ii) Classifier risk or attribute constraints. Given a classifier producing a logit $\ell(y)$ for an undesired attribute, a hard indicator cost $c(y) = \mathbf{1}\{\ell(y) \geq 0\}$ yields a chance-type expectation constraint, but it can be statistically noisy. A common alternative is a bounded surrogate, e.g.,

$$c(y) \;=\; \sigma(\ell(y)) \in (0,1),$$

with σ a sigmoid, or a hinge-type penalty. The expectation constraint $\mathbb{E}[c(y_T)] \leq \tau$ then controls average risk while producing lower-variance feedback.

(iii) CLIP-based safety or alignment scores. If one has a similarity model producing a scalar score $s_{\text{clip}}(y, \text{prompt})$, then a terminal reward $h(y) = s_{\text{clip}}(y, \text{prompt})$ encourages alignment with the prompt, while safety constraints may take the form $c(y) = \max\{0,\, s_{\text{unsafe}}(y) - \kappa\}$ for a detector score s_{unsafe} and threshold κ. Again, these are measurable terminal functions with noisy evaluations due to model stochasticity or prompt variability.

Summary of the interface. To emphasize the information pattern: during rollout we have access to (t_k, y_k, a_k) and can call the score-signal oracle (30) (hence the running reward signal (31)) using only a minibatch from the dataset and the known kernel $p_{t|0}$. At the end of the episode we receive terminal samples (34). The next section shows how to combine these ingredients into a primal–dual martingale q-learning procedure whose updates remain well-defined despite the absence of exact scores and despite noisy terminal constraint feedback.

5 Primal–Dual Martingale q-Learning

We now combine the signal interface of Section 4 with an entropy-regularized control formulation to obtain an on-policy primal–dual actor–critic procedure. The algorithm is a continuous-time analogue of q-learning based on martingale orthogonality conditions, augmented with a projected dual ascent for terminal expectation constraints.

Lagrangian shaping for terminal constraints. Let $\lambda \in \mathbb{R}_+^m$ be dual variables. For a fixed λ, we consider the Lagrangian-shaped objective

$$\mathcal{J}(\pi; \lambda) \;:=\; \mathbb{E}_\pi\left[\int_0^T r(t, y_t, a_t)\, dt \;+\; \beta h(y_T) \;-\; \sum_{i=1}^m \lambda_i\big(c_i(y_T) - \tau_i\big)\right], \quad (36)$$

where the expectation is over trajectories of the controlled reverse SDE under π. Since the constraints enter only through the terminal term, the running control structure is unchanged; only the terminal condition is modified. At the level of episode-end samples, we form the noisy Lagrangian terminal signal

$$\widehat{h}_L \;:=\; \beta\widehat{h} \;-\; \sum_{i=1}^{m} \lambda_i\big(\widehat{c}_i - \tau_i\big), \tag{37}$$

using the noisy terminal oracles $\widehat{h}, \widehat{c}_i$ from (34).

Gaussian policy structure and actor parameterization. We work with entropy-regularized exploratory control with temperature $\theta > 0$, and we restrict attention to Gaussian policies with fixed covariance

$$\pi_\psi(\cdot \mid t, y) \;=\; \mathcal{N}\bigg(\mu_\psi(t,y),\; \frac{\theta}{2g(T-t)^2} I\bigg)\,. \tag{38}$$

This family is natural for two reasons. First, the controlled reverse drift is affine in the action a_t, so Gaussian exploration yields tractable dynamics. Second, the running reward is (up to an additive term independent of a) quadratic in a:

$$r(t,y,a) = -g(T-t)^2\|a\|^2 + 2g(T-t)^2\langle \nabla \log p_{T-t}(y), a\rangle - g(T-t)^2\|\nabla \log p_{T-t}(y)\|^2,$$

and the entropy regularizer contributes a concave term in $\pi(\cdot \mid t, y)$, so the pointwise maximization over π yields a Gibbs distribution which, under a quadratic q-structure, is Gaussian. Concretely, for any parameter ψ and any (t, y), the log-density of (38) is

$$\log \pi_\psi(a \mid t, y) = -\frac{g(T-t)^2}{\theta}\|a - \mu_\psi(t,y)\|^2 + \mathrm{const}(t), \tag{39}$$

so learning reduces to learning the mean map μ_ψ. In the linear setting used later, we take $\mu_\psi(t,y) = \Phi(t,y)^\top\psi$ for a feature map Φ; however, the algorithmic construction below does not require linearity.

Critic parameterization and the martingale residual. Let $J_\Theta(t, y)$ denote a parametric approximation to the value function associated with the shaped objective (36) at the current dual iterate, and let $q_\psi(t, y, a)$ denote the policy-induced q-function (the “soft” advantage) satisfying the standard entropy-consistency identity

$$\pi_\psi(a \mid t, y) \;\propto\; \exp\bigg(\frac{1}{\theta} q_\psi(t,y,a)\bigg)\,. \tag{40}$$

For the Gaussian family (38), a convenient compatible choice is

$$q_\psi(t,y,a) \;=\; -\,g(T-t)^2\|a - \mu_\psi(t,y)\|^2 \;+\; b_\psi(t,y), \tag{41}$$

where $b_\psi(t,y)$ is an (optional) scalar baseline absorbed by normalization; one may take $b_\psi \equiv 0$ without changing the policy.

The martingale approach proceeds as follows. In continuous time, applying Itô's formula to $J(t,y_t)$ and rearranging terms yields an identity of the form

$$J(T,y_T) - J(0,y_0) + \int_0^T \big(r(t,y_t,a_t) - q(t,y_t,a_t)\big)\,dt + (\text{terminal term})$$

is a martingale under the policy when J, q satisfy the appropriate soft HJB relations. Discretizing on a grid $t_k = k\Delta t$ and using Euler–Maruyama to simulate y_{k+1} from (t_k, y_k, a_k) suggests the per-step residual

$$G_k := J_\Theta(t_{k+1},y_{k+1}) - J_\Theta(t_k,y_k) + \widehat{r}(t_k,y_k,a_k;\mathcal{B}_k)\Delta t - q_\psi(t_k,y_k,a_k)\Delta t + \mathbf{1}_{\{k=K-1\}}\widehat{h}_L. \tag{42}$$

Heuristically, if (Θ,ψ) were exact and $\widehat{r}$ were replaced by r, then $(G_k)_{k=0}^{K-1}$ would have conditional mean approximately zero given the past. We therefore fit (Θ,ψ) by driving the empirical residuals toward zero.

Primal updates (actor–critic via residual minimization). Given a batch of trajectories $\{(y_k^{(b)}, a_k^{(b)})\}_{k,b}$ generated on-policy from (38) and simulated dynamics, we minimize the squared residual loss

$$\mathcal{L}(\Theta,\psi) \;:=\; \frac{1}{B}\sum_{b=1}^{B}\sum_{k=0}^{K-1}\big(G_k^{(b)}\big)^2, \tag{43}$$

where $G_k^{(b)}$ is defined by (42) on trajectory b. We then perform stochastic gradient steps

$$\Theta \leftarrow \Theta - \alpha_\Theta \nabla_\Theta \mathcal{L}(\Theta,\psi), \qquad \psi \leftarrow \psi - \alpha_\psi \nabla_\psi \mathcal{L}(\Theta,\psi), \tag{44}$$

using automatic differentiation through J_Θ and μ_ψ (hence through q_ψ via (41)). Since the rollouts are on-policy, no importance weights are required. In the linear regime, (44) becomes a stochastic approximation scheme for a system of orthogonality conditions; this is the viewpoint we adopt in the next section.

Dual update (projected ascent). The dual variables are updated by projected stochastic ascent on the constraint violations. With batch terminal observations $\widehat{c}_i^{(b)}$, we set

$$\lambda_i \leftarrow \Pi_{[0,\Lambda]}\left(\lambda_i + \eta_\lambda\left(\frac{1}{B}\sum_{b=1}^{B}\widehat{c}_i^{(b)} - \tau_i\right)\right), \qquad i = 1,\dots,m, \tag{45}$$

where $\Pi_{[0,\Lambda]}$ denotes projection onto $[0,\Lambda]$ and $\Lambda < \infty$ is a chosen radius. Projection is both a stability device and an analytical requirement for compactness in the stochastic approximation arguments.

Algorithmic summary. We summarize the procedure.

> **PD-CTQL (Primal–Dual Continuous-Time q-Learning).** Fix $(T, \Delta t, K)$, temperature θ, reward weight β, projection radius Λ, and stepsizes $(\alpha_\Theta, \alpha_\psi, \eta_\lambda)$. Initialize (Θ, ψ) and $\lambda = 0$. For episodes $n = 1, \dots, N$:
>
> 1. Roll out B trajectories on the grid $t_k = k\Delta t$: sample $y_0 \sim \nu$, then for $k = 0, \dots, K-1$ sample $a_k \sim \mathcal{N}(\mu_\psi(t_k, y_k), \frac{\theta}{2g(T-t_k)^2} I)$, simulate y_{k+1}, and compute $\widehat{r}(t_k, y_k, a_k; \mathcal{B}_k)$ using a fresh minibatch $\mathcal{B}_k$ from the dataset.
> 2. Observe terminal signals $(\widehat{h}, \widehat{c}_1, \dots, \widehat{c}_m)$ and form $\widehat{h}_L$ via (37).
> 3. Compute residuals G_k via (42) and update (Θ, ψ) by (44).
> 4. Update λ by (45).
>
> Return π_ψ (and λ as a dual certificate).

Stabilization and practical modifications. Although the core algorithm is conceptually simple, several stabilizers improve numerical behavior without changing the information pattern.

(i) Dual clipping and conservative ascent. Projection (45) already clips the dual variables, but in addition one may (a) use a smaller dual stepsize η_λ than primal stepsizes (a two-timescale choice), and (b) apply Polyak averaging to the empirical constraint violations before the ascent step. Both reduce oscillations when terminal feedback is noisy.

(ii) Entropy annealing. The fixed-covariance policy (38) enforces exploration level θ. In early training, larger θ can improve coverage of visited states and hence reduce the risk of weight degeneracy in the score-signal oracle; later, decreasing θ sharpens the policy around the learned mean. A simple schedule is $\theta_n = \max\{\theta_{\min}, \theta_0 \rho^n\}$ with $\rho \in (0, 1)$, while keeping the covariance consistent with θ_n.

(iii) Critic regularization. The residual loss (43) can admit near-degenerate solutions when function classes are rich. Standard remedies include weight decay on Θ, spectral normalization (for neural critics), and gradient clipping on $\nabla_\Theta \mathcal{L}$. In the linear regime, an explicit ridge term $\frac{\kappa}{2}\|\Theta\|^2$ yields strong monotonicity of the expected update map, which simplifies finite-time analysis.

(iv) Truncated backpropagation through time. Because G_k depends on $J_\Theta(t_{k+1}, y_{k+1})$, full-trajectory differentiation is possible but can be memory-intensive for large K. One may treat (y_k) as a stop-gradient in the critic update (semi-gradient) and still retain a valid stochastic approximation interpretation in the linear setting.

Transition to theory. The next section specializes to linear actor/critic parameterizations and imposes bounded-moment conditions on the signal noise and rollouts. In that setting, the primal updates (44) and the projected dual ascent (45) form a two-timescale stochastic approximation scheme for a projected saddle-point system, enabling almost sure convergence and finite-episode bounds with explicit decompositions into approximation, score-signal, terminal-noise, and discretization errors.

6 Lower Bounds and Hardness

We complement the upper bounds of the linear theory with an information-theoretic obstruction: with only noisy *terminal* feedback, no algorithm can, in general, reach simultaneous ε-optimality and ε-feasibility in $o(\varepsilon^{-2})$ episodes. The point is not that diffusion control is intrinsically hard, but that black-box terminal oracles reduce the problem to constrained mean estimation, for which ε^{-2} is minimax-optimal.

Terminal-feedback-only protocols. We consider an episodic protocol in which, in each episode, an algorithm selects a (possibly history-dependent) policy $\pi^{(n)}$ to generate a trajectory $(y_t)_{t\in[0,T]}$, and then observes only terminal noisy samples

$$\widehat{h}^{(n)} = h(y_T^{(n)}) + \xi_h^{(n)}, \qquad \widehat{c}_i^{(n)} = c_i(y_T^{(n)}) + \xi_{c,i}^{(n)},$$

with ξ sub-Gaussian (or merely bounded second moment) noise. The algorithm may, of course, have full access to the simulated state y_t during rollout; the restriction is that the *reward/constraint* feedback is available only through the terminal oracles. This is precisely the regime in which terminal constraints are most challenging, and it is the regime addressed by the lower bound below. (In our full framework, one additionally has access to a running score-signal $\widehat{r}$; the lower bound isolates the irreducible difficulty contributed by terminal feedback, and therefore applies a fortiori to any method that does not extract extra information about h, c_i beyond their terminal noisy evaluations.)

Embedding a constrained bandit into a controlled diffusion. We exhibit a reduction from a two-armed constrained stochastic bandit. Fix $d = 1$, choose any $T > 0$, and take a constant-diffusion controlled reverse dynamics of the form

$$dy_t = g^2 a_t \, dt + g \, dW_t, \qquad y_0 = 0, \tag{46}$$

with known $g > 0$. Consider the restricted policy class of *open-loop constant controls* $a_t \equiv a$ with action set $\mathcal{A} = \{-A, +A\}$ for some constant $A > 0$.

Under (46), the terminal state satisfies

$$y_T \sim \mathcal{N}(g^2 AT,\, g^2 T) \quad \text{or} \quad y_T \sim \mathcal{N}(-g^2 AT,\, g^2 T),$$

depending on the chosen arm. Choosing A sufficiently large (as a constant, independent of ε) ensures that the two terminal distributions are well-separated, so that the event $\{y_T > 0\}$ acts as a near-deterministic indicator of the chosen arm.[2]

We now define terminal reward/constraint functions that are essentially arm-dependent constants. Let $\varphi : \mathbb{R} \to [0, 1]$ be a smooth step function such that $\varphi(y) = 1$ for $y \geq 1$ and $\varphi(y) = 0$ for $y \leq -1$, and define an "arm-indicator" $\chi(y) := \varphi(y)$. For two scalars $(r_-, r_+) \in [0, 1]^2$ and $(\kappa_-, \kappa_+) \in \mathbb{R}^2$, set

$$h(y) := r_- + (r_+ - r_-)\chi(y), \qquad c(y) := \kappa_- + (\kappa_+ - \kappa_-)\chi(y). \tag{47}$$

When A is large, choosing $a = +A$ makes $\chi(y_T) \approx 1$, whereas choosing $a = -A$ makes $\chi(y_T) \approx 0$, so the episode produces (up to an A-controlled approximation error) a noisy sample from one of two mean pairs (r_+, κ_+) or (r_-, κ_-). In other words, the diffusion control problem restricted to $\mathcal{A}$ simulates a two-armed bandit with reward mean $r_\pm$ and constraint mean $\kappa_\pm$.

A pair of hard instances. Fix a feasibility threshold $\tau \in \mathbb{R}$ and noise level σ^2 for the sub-Gaussian terminal noises. For a target accuracy $\varepsilon \in (0, 1/8)$, consider two instances $\mathsf{I}^{(1)}$ and $\mathsf{I}^{(2)}$ defined by (46)–(47) with the same dynamics but different terminal oracle parameters:

$$\mathsf{I}^{(1)}: \quad (r_+, r_-) = \Big(\tfrac{1}{2} + \varepsilon,\ \tfrac{1}{2}\Big), \quad (\kappa_+, \kappa_-) = \Big(\tau + \varepsilon,\ \tau - \varepsilon\Big);$$

$$\mathsf{I}^{(2)}: \quad (r_+, r_-) = \Big(\tfrac{1}{2},\ \tfrac{1}{2} + \varepsilon\Big), \quad (\kappa_+, \kappa_-) = \Big(\tau - \varepsilon,\ \tau + \varepsilon\Big).$$

In $\mathsf{I}^{(1)}$, arm "+" is (slightly) higher reward but violates the constraint in expectation, while arm "−" is feasible and only ε worse in reward. In $\mathsf{I}^{(2)}$ the roles are swapped. Thus, in either instance, an ε-optimal *and* ε-feasible algorithm must identify which arm is feasible (up to the ε slack), which is exactly a constrained identification/mean-estimation task.

Theorem 6.1 (Minimax lower bound via constrained bandit reduction)**.** *Consider the one-dimensional controlled diffusion* (46) *with action set* $\mathcal{A} = \{-A, +A\}$ *and terminal reward/constraint oracles* (47)*, with additive independent sub-Gaussian terminal noises of variance proxy at most* σ^2*. There*

[2] If one prefers an exact embedding, one may instead define the oracles to depend on the chosen action as well as y_T; however, the near-deterministic separation suffices for the minimax argument and keeps the embedding within our terminal-oracle model.

exists a constant $c > 0$ (depending only on fixed separation parameters such as A, g, T and the step function φ) such that the following holds.

For any (possibly adaptive) algorithm that, after N episodes, outputs a policy $\widehat{\pi}$ which with probability at least $2/3$ is simultaneously ε-feasible and ε-optimal for both *instances $\mathsf{I}^{(1)}$ and $\mathsf{I}^{(2)}$, we must have*

$$N \ \geq \ c \, \frac{\sigma^2}{\varepsilon^2}.$$

Equivalently, the minimax episode complexity for achieving (ε-optimality, ε-feasibility) under noisy terminal feedback is $\Omega(\varepsilon^{-2})$.

Proof sketch. We reduce to Le Cam's two-point method. Under the restriction $\mathcal{A} = \{-A, +A\}$, each episode consists of choosing an arm and observing a noisy scalar reward and a noisy scalar constraint value. Because the oracles (47) are (approximately) constant on the high-probability terminal regions induced by each arm, the conditional distribution of the observed terminal pair $(\widehat{h}, \widehat{c})$ given the chosen arm is (up to a fixed, ε-independent approximation error that can be absorbed into constants) a product of two sub-Gaussian observations with means $(r_\pm, \kappa_\pm)$.

For any fixed adaptive algorithm, let $\mathbb{P}_1$ and $\mathbb{P}_2$ denote the laws of the full interaction transcript (actions and observations) under $\mathsf{I}^{(1)}$ and $\mathsf{I}^{(2)}$. The per-episode KL divergence between the conditional observation distributions under the same chosen arm is $O(\varepsilon^2/\sigma^2)$, since the means differ by $\Theta(\varepsilon)$ and the noise is sub-Gaussian with proxy σ^2. By the chain rule for KL and adaptivity, we obtain

$$\mathrm{KL}(\mathbb{P}_1 \,\|\, \mathbb{P}_2) \ \leq \ C \, N \, \frac{\varepsilon^2}{\sigma^2}$$

for a constant C independent of N, ε. If $N \leq c\,\sigma^2/\varepsilon^2$ with c small, then $\mathrm{KL}(\mathbb{P}_1 \,\|\, \mathbb{P}_2)$ is bounded by an absolute constant, implying that the total variation distance between $\mathbb{P}_1$ and $\mathbb{P}_2$ is bounded away from 1. Consequently, no decision rule based on the transcript can distinguish $\mathsf{I}^{(1)}$ from $\mathsf{I}^{(2)}$ with probability exceeding (say) $2/3$.

However, the identity of the instance determines which arm is (approximately) feasible and hence which arm any ε-optimal and ε-feasible solution must favor. Therefore, any algorithm that cannot reliably distinguish the instances must, with probability at least $1/3$ on one of them, either choose the infeasible arm often enough to incur constraint violation exceeding ε, or choose the safe arm when it is suboptimal by more than ε. This yields the stated lower bound. □

Implications for upper bounds. Theorem 6.1 shows that the $\tilde{O}(N^{-1/2})$ rates obtained by stochastic approximation analyses are unimprovable in the episode count, even in a highly benign LQG diffusion with a two-action policy class and with perfect knowledge of the dynamics. In particular, any improvement beyond $N^{-1/2}$ would contradict the $\Omega(\varepsilon^{-2})$ minimax episode complexity for terminal-feedback constrained mean estimation.

Multiple constraints. The above construction extends to $m > 1$ constraints by taking m independent constraint oracles whose means differ by $\Theta(\varepsilon)$ across instances, yielding a worst-case lower bound $\Omega(m\varepsilon^{-2})$ when one requires uniform ε-feasibility across all constraints. Formally, one constructs 2^m instances indexed by a sign vector in $\{\pm 1\}^m$, applies a standard multi-hypothesis testing reduction (Fano), and obtains that at least one constraint coordinate remains insufficiently estimated unless $N = \Omega(m\varepsilon^{-2})$.

Discussion. We emphasize what the lower bound does and does not say. It does not preclude faster rates in regimes where (a) the terminal oracles are known analytically, (b) one can query gradients of $\mathbb{E}[h(y_T)]$ and $\mathbb{E}[c_i(y_T)]$, or (c) additional side information is available beyond noisy terminal evaluations. Rather, it asserts that, under the black-box model where h and c_i are accessed only through noisy terminal samples, the episode complexity is dominated by the need to estimate means accurately enough to certify feasibility and near-optimality. This obstruction persists regardless of computational power, function approximation capacity (including overparameterized or NTK-like regimes), or the availability of accurate simulation of the controlled diffusion.

7 Implementation Notes and Recommended Experiments

We collect here practical remarks for implementing PD-CTQL and a set of experiments that, in our view, exercise the distinctive aspects of the model: (i) terminal-only black-box reward/constraints, (ii) on-policy training in a controlled reverse diffusion, and (iii) primal–dual updates with projected multipliers. Throughout, we assume that the reverse-time controlled dynamics are simulated on a grid $t_k = k\Delta t$, $k = 0, \dots, K$, and that the policy is the Gaussian family prescribed by the entropy-regularized quadratic control structure,

$$\pi_\psi(\cdot \mid t, y) = \mathcal{N}\Big(\mu_\psi(t, y),\ \frac{\theta}{2g(T-t)^2} I\Big), \qquad a_k = \mu_\psi(t_k, y_k) + \sqrt{\frac{\theta}{2g(T-t_k)^2}}\,\zeta_k,\ \ \zeta_k \sim \mathcal{N}(0, I).$$

We recommend implementing action sampling via the above reparameterization so that actor gradients propagate cleanly through μ_ψ. For the controlled sampler, Euler–Maruyama suffices for the theory-aligned baseline:

$$y_{k+1} = y_k + \big[f(T-t_k)y_k + g(T-t_k)^2 a_k\big]\Delta t + g(T-t_k)\sqrt{\Delta t}\,\xi_k, \qquad \xi_k \sim \mathcal{N}(0, I),$$

with the understanding that higher-order solvers (or predictor–corrector variants) can be substituted to reduce discretization error in the integral reward surrogate.

Score-signal computation and numerical stability. The running reward depends on $\|\nabla \log p_{T-t}(y) - a\|^2$, where the score is unavailable. In our interaction model we replace this by a ratio-estimator-based signal $\widehat{r}(t, y, a) = -g(T-t)^2 \|\widehat{s}(t, y) - a\|^2$. In practice, the dominant numerical issue is evaluating $\widehat{s}(t, y)$ stably for small t and large d. A robust implementation uses log-sum-exp normalization when estimating quantities of the form

$$\widehat{p}_t(y) \approx \frac{1}{M} \sum_{j=1}^{M} p_{t|0}(y \mid x_0^{(j)}), \qquad \widehat{s}(t, y) = \nabla_y \log \widehat{p}_t(y),$$

with $x_0^{(j)}$ drawn from the dataset minibatch. We recommend (i) evaluating $\log p_{t|0}(y \mid x_0^{(j)})$ and normalizing by subtracting the maximum log-density across j, (ii) computing gradients analytically using the Gaussian form of $p_{t|0}$, and (iii) clipping $\|\widehat{s}(t, y)\|$ to a moderate threshold to prevent rare large ratios from destabilizing the critic loss. Since $\widehat{r}$ is only used as a stochastic signal inside a stochastic approximation scheme, mild clipping typically improves performance without changing qualitative behavior.

Critic loss and trajectory storage. The martingale-residual objective requires correlating successive states along each trajectory. If memory permits, we store $(y_k, a_k, \widehat{r}_k)$ for $k = 0, \ldots, K-1$ and compute the residuals

$$G_k = J_\Theta(t_{k+1}, y_{k+1}) - J_\Theta(t_k, y_k) + \widehat{r}_k \Delta t + \mathbf{1}\{k = K-1\}\, \widehat{h}_L - q_\psi(t_k, y_k, a_k)\Delta t,$$

then minimize $\sum_k G_k^2$ over the batch. When K is large (as in image diffusion), one may instead use truncated backpropagation through time or a sliding window, at the cost of some bias. We also found it helpful to normalize the residuals by an estimate of their running standard deviation, to avoid the terminal term dominating early in training.

Dual update and two-timescale tuning. The practical role of the dual update is to trade off constraint satisfaction against reward. The projected ascent step

$$\lambda_i \leftarrow \Pi_{[0,\Lambda]}\Big(\lambda_i + \eta_\lambda \big(\widehat{\mathbb{E}}[c_i(y_T)] - \tau_i\big)\Big)$$

is sensitive to η_λ and Λ. We recommend the following protocol. First, fix Λ large enough that projection is inactive at the eventual solution (this can be checked post hoc by monitoring how often λ hits the boundary). Second, tune η_λ so that multipliers evolve noticeably slower than the actor–critic parameters; concretely, if α denotes the actor/critic learning rate, we often choose $\eta_\lambda \in [10^{-3}\alpha, 10^{-1}\alpha]$ depending on the noise level in $\widehat{c}_i$. Third, report the final λ as a certificate of how hard the constraints are: persistent large multipliers indicate an inherently tight feasible set for the chosen policy class.

Experiment 1: Constrained Swiss-roll with terminal property constraints. A minimal continuous experiment is a $d = 2$ synthetic dataset (Swiss-roll or mixtures of Gaussians) with a terminal reward that encourages a property not aligned with the data likelihood. For instance, let p_0 be Swiss-roll in $\mathbb{R}^2$, let $h(y)$ reward landing in a target region (e.g. $h(y) = \mathbf{1}\{y \in \mathcal{R}\}$ smoothed, or $h(y) = -\mathrm{dist}(y, \mathcal{R})^2$), and impose a constraint on a different terminal attribute, e.g. $c(y) = \|y\|^2$ with τ controlling radial spread, or $c(y) = \mathbf{1}\{y_1 \geq 0\}$ with a target proportion. The point is to force a nontrivial tradeoff between reward shaping via the terminal oracle and plausibility via the diffusion score penalty. We recommend sweeping β and reporting Pareto curves of (estimated) $\mathbb{E}[h(y_T)]$ versus constraint violation $\big(\mathbb{E}[c(y_T)] - \tau\big)_+$. Baselines include (i) unconstrained PD-CTQL with $\lambda \equiv 0$, (ii) a penalty method replacing λ by a fixed scalar tuned on a validation set, and (iii) post hoc rejection sampling (which typically degrades sample efficiency sharply when τ is tight).

Experiment 2: Synthetic LQG with known optimum (sanity check). To separate algorithmic issues from representation and score-signal approximation, we recommend an LQG instance where an analytic solution is available. Take linear dynamics with constant g and choose a quadratic terminal reward $h(y) = -\frac{1}{2}y^\top Qy$ and linear/quadratic constraints $c_i(y)$ (e.g. $c(y) = u^\top y$ or $c(y) = \frac{1}{2}\|y\|^2$). In this setting, for linear actor/critic, one can compute the optimal feasible control (or at least the optimal control for a fixed λ) and use it to validate: (i) convergence of λ to a stabilizing value, (ii) scaling of suboptimality and violation with N, and (iii) sensitivity to discretization K. This experiment also allows one to inject controlled noise into $\widehat{h}, \widehat{c}$ and empirically confirm the anticipated ε^{-2} scaling in episode complexity by plotting the number of episodes needed to reach fixed target tolerances.

Experiment 3: Image generation with safety-classifier constraints. For images, we interpret y_t as a latent (or pixel) variable in a pre-trained diffusion model whose forward SDE coefficients f, g are fixed. Terminal reward and constraints are provided by black-box evaluators on decoded images. A representative configuration is: reward $h(y_T)$ is an aesthetic score or text-image alignment score (e.g. CLIP similarity), and constraints $c_i(y_T)$ are safety-related (e.g. NSFW probability, toxicity score, or a content policy classifier). Thresholds τ_i encode desired safety levels, and the algorithm learns a control that improves the reward while keeping expected unsafe scores below τ_i. We recommend evaluating against: (i) classifier guidance (which does not directly enforce expectation constraints), (ii) simple constrained sampling via rejection (which may be infeasible at high resolution), and (iii) Lagrangian-guidance heuristics that tune a penalty weight offline rather than via dual ascent. In addition to standard sample-quality metrics, we must report constraint metrics with confidence intervals, since constraints are expectations under a stochastic generator.

Ablations: dual stepsize, projection radius, temperature, and solver order. We recommend four ablations that directly probe the design choices in PD-CTQL. (i) *Dual stepsize* η_λ: too large yields oscillatory feasibility; too small yields slow constraint enforcement. Plot trajectories of λ_i and the running estimate of $\mathbb{E}[c_i(y_T)] - \tau_i$. (ii) *Projection radius* Λ: vary Λ over orders of magnitude; if Λ is too small, constraints may remain violated even at convergence due to dual clipping. (iii) *Temperature* θ: higher θ increases exploration (and variance) via the fixed policy covariance; we recommend reporting how θ affects constraint satisfaction and sample diversity. (iv) *Discretization/solver*: compare Euler–Maruyama with a higher-order sampler at matched compute; quantify the change in reward and violation at equal N to illustrate discretization effects.

Evaluation metrics and reporting. We recommend reporting (a) estimated terminal reward $\widehat{\mathbb{E}}[h(y_T)]$ and constraint violation $\max_i(\widehat{\mathbb{E}}[c_i(y_T)] - \tau_i)_+$ using an *independent* set of rollouts (not reused for training updates), (b) the empirical distribution of $c_i(y_T)$ (not only its mean) to expose heavy tails, and (c) the final dual variables λ together with the fraction of iterations at which projection is active. For image generation, we additionally report FID (or KID) and a diversity statistic (e.g. LPIPS diversity) alongside safety violation rates. Since our constraints are in expectation, it is essential to include confidence intervals: for each i, report $\widehat{\mathbb{E}}[c_i(y_T)] \pm 1.96\,\widehat{\mathrm{se}}$, and declare feasibility only when the upper confidence bound is below τ_i (or adopt a deliberately conservative margin). This reporting discipline aligns empirical practice with the theoretical interpretation of feasibility as an expectation constraint under the learned policy.

8 Implementation Notes and Recommended Experiments

We record here implementation details that materially affect the stability of PD-CTQL, together with experiments that isolate (i) terminal-only black-box reward/constraints, (ii) on-policy interaction with a controlled reverse diffusion, and (iii) the projected primal–dual mechanism. We assume throughout a uniform grid $t_k = k\Delta t$ for $k = 0, \dots, K$ and a Gaussian policy with fixed covariance prescribed by the entropy-regularized quadratic structure,

$$\pi_\psi(\cdot \mid t, y) = \mathcal{N}\Big(\mu_\psi(t,y),\, \frac{\theta}{2g(T-t)^2} I\Big), \qquad a_k = \mu_\psi(t_k, y_k) + \sqrt{\frac{\theta}{2g(T-t_k)^2}}\, \zeta_k,\ \ \zeta_k \sim \mathcal{N}(0, I).$$

We implement action sampling via this reparameterization so that gradients propagate only through μ_ψ and not through sampling. For the controlled sampler, Euler–Maruyama provides the theory-aligned baseline,

$$y_{k+1} = y_k + \big[f(T-t_k)y_k + g(T-t_k)^2 a_k\big]\Delta t + g(T-t_k)\sqrt{\Delta t}\,\xi_k, \qquad \xi_k \sim \mathcal{N}(0, I),$$

and all reported quantities (objective and constraints) should be interpreted as approximations to the continuous-time quantities up to discretization error of order $O(\Delta t)$ (or better if a higher-order solver is used).

Computing the score-based running signal. The running reward involves $\|\nabla \log p_{T-t}(y) - a\|^2$ and thus requires a score surrogate. In our interaction model this appears only as a stochastic signal, so we implement

$$\widehat{r}(t, y, a) = -g(T-t)^2 \big\|\widehat{s}(t, y) - a\big\|^2, \qquad \widehat{s}(t, y) = \nabla_y \log \widehat{p}_{T-t}(y), \qquad \widehat{p}_t(y) = \frac{1}{M}\sum_{j=1}^{M} p_{t|0}(y \mid x_0^{(j)}),$$

with $\{x_0^{(j)}\}_{j=1}^M$ drawn i.i.d. from the dataset minibatch. Numerically, the relevant object is a mixture of Gaussians; the dominant failure mode is underflow/overflow when d is large or t is small. We therefore recommend implementing $\log \widehat{p}_t(y)$ by log-sum-exp stabilization: compute $\ell_j = \log p_{t|0}(y \mid x_0^{(j)})$, set $\ell_{\max} = \max_j \ell_j$, and evaluate

$$\log \widehat{p}_t(y) = \ell_{\max} + \log\Big(\frac{1}{M}\sum_{j=1}^{M} e^{\ell_j - \ell_{\max}}\Big), \qquad \widehat{s}(t, y) = \sum_{j=1}^{M} w_j \nabla_y \log p_{t|0}(y \mid x_0^{(j)}),$$

where $w_j \propto e^{\ell_j - \ell_{\max}}$. Since $p_{t|0}$ is Gaussian, $\nabla_y \log p_{t|0}$ is available in closed form and should be coded analytically rather than by automatic differentiation through a density routine. Two practical heuristics are consistently beneficial: (i) clipping $\|\widehat{s}(t, y)\|$ (or equivalently clipping $\widehat{r}$) to control rare large ratios, and (ii) using a moderately large M early in training (to reduce signal variance), then reducing M once the policy concentrates. Because the critic loss involves squared residuals, heavy-tailed $\widehat{r}$ can dominate updates; clipping typically improves optimization without changing the qualitative fixed point.

Critic residuals, parameterization, and variance control. PD-CTQL uses martingale orthogonality via a squared residual loss along sampled trajectories. Concretely, for each trajectory we form

$$G_k = J_\Theta(t_{k+1}, y_{k+1}) - J_\Theta(t_k, y_k) + \widehat{r}_k \Delta t + \mathbf{1}\{k = K-1\}\, \widehat{h}_L - q_\psi(t_k, y_k, a_k)\Delta t, \qquad \widehat{h}_L = \beta \widehat{h} - \sum_{i=1}^{m} \lambda_i (\widehat{c}_i - \tau$$

and minimize $\sum_{k=0}^{K-1} G_k^2$ over a batch. When K is modest, storing the full trajectory $(y_k, a_k, \widehat{r}_k)$ is simplest. When K is large, we recommend either (i) truncating the residual loss to a sliding window (introducing a controlled bias) or (ii) accumulating residuals online and discarding intermediate states. In either case, gradient clipping for (Θ, ψ) is usually necessary once $\widehat{h}_L$ has high variance.

The q-term may be implemented in several equivalent ways. If we explicitly enforce the Gaussian form of π_ψ with fixed covariance, then $\log \pi_\psi(a \mid t, y)$ is available in closed form and we may set $q_\psi = \theta \log \pi_\psi$ up to an additive constant independent of a (which cancels in policy normalization). This avoids learning an additional network for q_ψ and often stabilizes training; alternatively, one may learn J_Θ and μ_ψ only and treat q_ψ as implied by the Gaussian policy. Independently of the parameterization, we recommend normalizing residuals by an empirical running scale estimate (e.g. dividing G_k by a moving standard deviation) so that the terminal term does not overwhelm the early-time residuals.

Dual ascent and timescale separation. The projected update

$$\lambda_i \leftarrow \Pi_{[0,\Lambda]}\Big(\lambda_i + \eta_\lambda\big(\widehat{\mathbb{E}}[c_i(y_T)] - \tau_i\big)\Big)$$

is the mechanism enforcing expectation constraints. Empirically, stability depends more on *timescale separation* than on the absolute value of η_λ. We thus tune η_λ relative to the actor–critic learning rate α so that λ changes slowly compared to (ψ, Θ). A practical protocol is: choose Λ so that projection is inactive at the apparent solution (monitored by the fraction of updates hitting the boundary), then set $\eta_\lambda \in [10^{-3}\alpha, 10^{-1}\alpha]$ depending on the noise of $\widehat{c}_i$. If constraints oscillate (alternating over- and under-satisfaction), we decrease η_λ and/or increase the batch size for terminal estimates. We interpret persistently large λ_i as a diagnostic: either the constraint is tight for the chosen policy class, or the horizon/temperature prevents sufficiently targeted control.

Experiment A: constrained Swiss-roll (or mixtures) with terminal property constraints. We begin with a low-dimensional ($d = 2$) dataset (Swiss-roll, rings, or Gaussian mixtures) to visualize trajectories and directly inspect constraint effects. We choose a terminal reward h that induces a distributional shift (e.g. attraction to a designated region $\mathcal{R}$), and a terminal constraint c that competes with this shift. Examples that are easy to interpret are

$$h(y) = -\mathrm{dist}(y, \mathcal{R})^2, \qquad c(y) = \|y\|^2 \text{ with threshold } \tau, \qquad \text{or} \qquad c(y) = \mathbf{1}\{y_1 \geq 0\} \text{ with threshold } \tau$$

We recommend sweeping β and τ to obtain a Pareto front, reporting $(\widehat{\mathbb{E}}[h(y_T)]$, $(\widehat{\mathbb{E}}[c(y_T)] - \tau)_+)$ and also the empirical distribution of $c(y_T)$ to reveal whether feasibility is achieved via tail suppression or via a global shift. Baselines should include: (i) the unconstrained variant with $\lambda \equiv 0$, (ii) a fixed-penalty method where λ is tuned offline (which typically underperforms when noise or nonstationarity is present), and (iii) rejection sampling at terminal time (which becomes inefficient as τ tightens). In this setting we can additionally plot the evolution of λ and the running estimate of constraint violation across training, which serves as an operational check of the intended primal–dual behavior.

Experiment B: synthetic LQG instance with known optimum. To disentangle representational issues from the algorithmic mechanism, we recommend a linear–quadratic instance where (for fixed λ) the optimal policy can be computed, and where feasibility/optimality can be measured precisely. A typical choice is constant g, linear drift, quadratic terminal reward $h(y) = -\frac{1}{2}y^\top Q y$, and a linear or quadratic constraint such as $c(y) = u^\top y$ or $c(y) = \frac{1}{2}\|y\|^2$. With a linear actor/critic, we can verify: (i) convergence of λ to a stabilizing value, (ii) the $N^{-1/2}$ scaling by plotting suboptimality and violation versus number of episodes on log–log axes, and (iii) sensitivity to discretization by varying K at fixed wall-clock compute. This experiment is also the natural place to inject controlled additive noise into $\widehat{h}$ and $\widehat{c}$ and empirically confirm the ε^{-2} episode scaling predicted by the lower bound: fix target tolerances $(\varepsilon_{\mathrm{opt}}, \varepsilon_{\mathrm{feas}})$ and measure the number of episodes needed for both to be met with a prescribed confidence level.

Experiment C: image generation with safety-classifier constraints. For images we treat the diffusion backbone (and thus f, g) as fixed and use PD-CTQL to learn a control that changes the sampling distribution at terminal time. The terminal reward $h(y_T)$ is provided by a black-box evaluator (e.g. a text-image alignment model), while constraints $c_i(y_T)$ are safety-related (e.g. NSFW probability, policy violation score). Thresholds τ_i encode acceptable expected risk. We emphasize two implementation points. First, training must be on-policy: constraint satisfaction is a property of the current generator, and stale rollouts can mislead dual ascent. Second, constraint metrics should be estimated using an *independent* evaluation set of rollouts (not reused in updates), since otherwise the same terminal noise that drives learning can bias reported feasibility. Baselines include classifier guidance (which lacks expectation-constraint enforcement), rejection at terminal time (often infeasible computationally at scale), and offline-tuned penalty guidance (which fails to adapt as the policy changes). In addition to quality metrics, we report safety metrics with confidence intervals; since constraints are expectations, it is methodologically appropriate to declare feasibility only when an upper confidence bound is below τ_i.

Ablations and reporting discipline. We recommend four ablations that directly interrogate the algorithmic design: (i) η_λ (dual timescale), (ii) Λ (dual clipping), (iii) θ (exploration through fixed covariance), and (iv) solver order / Δt (discretization). For each ablation we report the induced changes in reward and constraint violation at fixed compute. For images we additionally report FID (or KID) and a diversity statistic (e.g. LPIPS diversity) alongside $\max_i(\widehat{\mathbb{E}}[c_i(y_T)] - \tau_i)_+$. Finally, we report the terminal constraint distribution (not only its mean) and the final λ together with the frequency of projection activity; these quantities are essential for interpreting whether feasibility is achieved robustly or by exploiting tails.

9 Extensions

Conditional diffusion and contextual control. Many applications require sampling from a conditional data law $p_0(\cdot \mid C)$ given a context variable C (class labels, text embeddings, or other side information). The forward SDE and its Gaussian transition kernel naturally extend by conditioning: for each realized C, we run the same forward OU/VP dynamics from $x_0 \sim p_0(\cdot \mid C)$, yielding marginals $p_t(\cdot \mid C)$ and conditionals $p_{t|0}(\cdot \mid x_0, C)$ (the latter often independent of C for OU/VP coefficients). The reverse-time controlled dynamics become

$$dy_t = \big[f(T-t)y_t + g(T-t)^2 a_t\big]dt + g(T-t)\,dW_t, \qquad y_0 \sim \nu(\cdot \mid C),$$

with a contextual policy $\pi_\psi(a \mid t, y, C) = \mathcal{N}(\mu_\psi(t,y,C), \frac{\theta}{2g(T-t)^2} I)$. The running reward is replaced by the conditional score mismatch,

$$r_C(t,y,a) = -g(T-t)^2 \big\|\nabla \log p_{T-t}(y \mid C) - a\big\|^2,$$

and terminal feedback may also be context-dependent, $h(y_T, C)$ and $c_i(y_T, C)$, with constraints imposed either per-context (harder) or in expectation over the context distribution (simpler):

$$\mathbb{E}[c_i(y_T, C)] \le \tau_i, \qquad \text{or} \qquad \mathbb{E}[c_i(y_T, C) \mid C] \le \tau_i(C) \ \text{ for all } C.$$

Algorithmically, the only substantive change is that the ratio-estimator signal for $\nabla \log p_t(\cdot \mid C)$ should draw minibatches from the conditional dataset (or reweight a pooled dataset by a learned propensity model for C). When C is high-dimensional (e.g. text embeddings), a pragmatic alternative is to treat the score surrogate as $\widehat{s}(t, y, C) = \nabla_y \log \widehat{p}_t(y \mid C)$ where $\widehat{p}_t(\cdot \mid C)$ is a kernel mixture over training samples whose contexts lie in a neighborhood of C. The convergence and finite-sample statements carry through with the same structure, with constants depending on the conditional moment bounds and the conditional signal quality (the latter entering the SignalErr term).

ODE samplers and deterministic controlled flows. While our interaction model is phrased as a controlled reverse SDE, it is often computationally advantageous to replace stochastic sampling with an ODE solver. In score-based diffusion this corresponds to the probability-flow ODE; in our notation the uncontrolled reverse drift is typically of the form $f(T-t)y_t - g(T-t)^2 \nabla \log p_{T-t}(y_t)$, and we have replaced the score by an action a_t. The corresponding controlled probability-flow dynamics therefore take the deterministic form

$$\dot{y}_t = f(T-t)y_t + g(T-t)^2 a_t, \qquad y_0 \sim \nu,$$

with terminal-time feedback $\beta h(y_T) - \sum_i \lambda_i (c_i(y_T) - \tau_i)$ as before. Two issues then arise. First, exploration no longer arises from Brownian noise, so an entropy-regularized Gaussian policy should be interpreted as injecting *policy noise* into the control channel, i.e. sampling $a_t \sim \pi_\psi(\cdot \mid t, y_t)$ while the state evolves deterministically given the sampled control. This preserves the variational structure of Theorem 1 (quadratic action penalty yields Gaussian optimality) while enabling deterministic integration schemes (Runge–Kutta, adaptive step sizes) for the state. Second, the discretization error changes character: for a p-th order ODE solver the analogue of DiscErr becomes $O(\Delta t^p)$ under standard smoothness assumptions on f, g and on the learned mean map μ_ψ. In practice, when $g(T-t)$ becomes small near terminal time, the ODE is often stiff; adaptive solvers reduce error but also introduce nonuniform computational budgets across trajectories. One convenient compromise is a hybrid scheme: early time steps use an SDE integrator to maintain exploration when noise is large, and late time steps switch to an ODE solver once the policy has concentrated and stochasticity primarily increases estimator variance.

Multiple constraints, continuation, and empirical Pareto fronts. Our formulation already allows $m > 1$ constraints with a dual vector $\lambda \in \mathbb{R}_+^m$, but in applications we typically require a family of tradeoffs rather than a single feasible point. We may view the thresholds $\tau \in \mathbb{R}^m$ and reward weight β as knobs defining a family of constrained problems; running PD-CTQL across a grid yields an empirical Pareto front in the plane of $(\mathbb{E}[h(y_T)], \mathbb{E}[c(y_T)])$ (or its higher-dimensional analogue). To reduce the cost of sweeping, we recommend a continuation strategy: start from an "easy" constraint vector $\tau^{(0)}$

(loose feasibility) and gradually tighten $\tau^{(s)} \downarrow \tau^\star$, warm-starting (ψ, Θ, λ) at each stage. Formally, if the saddle point mapping $\tau \mapsto (\psi^\star(\tau), \lambda^\star(\tau))$ is locally Lipschitz (as in strongly monotone linear settings), then continuation reduces transient oscillations by tracking the solution path. A similar continuation may be used in β to trade terminal reward against score-matching regularization, which is useful when h is very noisy: begin with small β (prioritizing score-consistency) and increase β once the generator stabilizes. We also note that multiple constraints with heterogeneous noise levels benefit from constraint-specific dual stepsizes $\eta_{\lambda,i}$ and from constraint normalization. Since the dual ascent uses estimates of $\mathbb{E}[c_i(y_T)] - \tau_i$, rescaling c_i to comparable magnitudes reduces ill-conditioning and improves the practical validity of timescale separation.

Robust constraints under distribution shift and risk-sensitive variants. Expectation constraints are brittle when the terminal cost oracle drifts (e.g. a safety classifier whose calibration changes) or when the deployment context differs from the training context. A robust extension is to impose constraints uniformly over an ambiguity set of environments $\mathcal{Q}$:

$$\sup_{Q \in \mathcal{Q}} \mathbb{E}_{Q,\pi_\psi}[c_i(y_T)] \le \tau_i,$$

where Q may index perturbations of the terminal cost functional, the initial prior ν, or even mild misspecification of the forward coefficients used in the score signal. One tractable choice is an f-divergence ball around a nominal environment Q_0, which leads (by convex duality) to a risk-sensitive penalty of the form $\rho(c_i(y_T))$ such as an entropic risk or a variance-regularized surrogate. Another practically useful variant is CVaR-type constraints, replacing $\mathbb{E}[c_i(y_T)]$ by $\mathrm{CVaR}_\alpha(c_i(y_T))$ to explicitly control tails; this is particularly relevant when c_i measures a rare but severe violation. In both cases the primal–dual mechanism remains natural: we introduce additional dual variables for the robustified constraints and estimate the corresponding gradients from terminal samples. What changes in the analysis is not the stochastic approximation skeleton but the concentration behavior of the terminal statistics: robust objectives typically amplify tail noise, and thus the constants hidden in $\tilde{O}(N^{-1/2})$ may become large unless one uses larger batches or explicit variance reduction (e.g. control variates based on cheap proxies for c_i). Empirically, robust constraints also interact with the projection radius Λ: if Λ is too small, the algorithm cannot express the required robustness and will settle at a point with persistent violation; if Λ is too large, dual-driven nonstationarity can destabilize the actor unless the dual stepsize is reduced accordingly.

Privacy-preserving score signals and constrained access to data. Our interaction model accesses the dataset only through minibatch computations of $\widehat{p}_t(y)$ and $\widehat{s}(t,y) = \nabla_y \log \widehat{p}_t(y)$, which suggests privacy mechanisms localized to the score-signal pathway. A simple approach is to make the ratio estimator differentially private by (i) clipping per-sample contributions to the mixture score and (ii) adding calibrated Gaussian noise. Concretely, writing $\widehat{s}(t,y) = \sum_{j=1}^{M} w_j u_j$ with $u_j = \nabla_y \log p_{t|0}(y \mid x_0^{(j)})$, we may clip u_j to $\|u_j\| \le S$ and release

$$\widehat{s}_{\mathrm{DP}}(t,y) = \sum_{j=1}^{M} w_j \operatorname{clip}(u_j; S) + \sigma_{\mathrm{DP}}\, \zeta, \qquad \zeta \sim \mathcal{N}(0, I),$$

with σ_{DP} set by a privacy accountant for subsampled Gaussian mechanisms. This produces a private running signal $\widehat{r}_{\mathrm{DP}}(t,y,a) = -g(T-t)^2 \|\widehat{s}_{\mathrm{DP}}(t,y) - a\|^2$. From the viewpoint of our bounds, this modification simply increases SignalErr by an additive variance term proportional to σ_{DP}^2, and introduces a bias term if clipping is active. Hence the same convergence statements apply provided the second moments remain bounded; however, the achievable ε for a fixed privacy budget is limited by the unavoidable noise floor in $\widehat{s}_{\mathrm{DP}}$. More structured mechanisms are possible: one may precompute privatized sufficient statistics for Gaussian mixtures, use random-feature approximations to privatize density evaluation, or train a privatized conditional density model for p_0 and use it as a drop-in replacement for dataset minibatches. The common theme is that PD-CTQL does not require learning a globally accurate score, only sufficiently informative local signals along the on-policy state distribution; this locality can be exploited to concentrate the privacy budget on regions the policy actually visits.

Limitations and open problems. Our main theoretical guarantees are stated for linear actor–critic parametrizations (or, equivalently, for settings in which the induced stochastic approximation (SA) dynamics are effectively linear after feature lifting). This leaves open the regime most relevant to modern diffusion models, namely deep nonlinear parametrizations of μ_ψ and J_Θ with nonconvex objectives and potentially unstable coupled updates. While one may hope to import arguments from the neural tangent kernel (NTK) literature or from overparameterized actor–critic analyses, two obstacles are specific to our setting: (i) the running signal $\hat{r}$ is itself a learned, on-the-fly functional of the dataset through the ratio estimator, and thus its error distribution depends on the visited states; (ii) the dual variables induce an additional slow nonstationarity which is benign in two-timescale linear SA but can amplify instabilities in nonlinear training. A principled extension would require a stability theory for primal–dual martingale-residual learning with *state-dependent* reward noise and biased score surrogates, ideally yielding conditions under which the iterates remain in a compact set without imposing explicit projections on ψ, Θ. Even in the NTK limit, it is not immediate that the induced kernel regression viewpoint respects the constrained saddle-point structure, because the critic loss is not a standard supervised objective but enforces martingale orthogonality under the evolving policy.

A second limitation concerns the modeling of constraints and, more broadly, the gap between a terminal expectation constraint and practical notions of safety. Our basic constraints take the form $\mathbb{E}[c_i(y_T)] \leq \tau_i$, estimated from noisy terminal observations. This is statistically convenient but may be inadequate when violations are rare yet catastrophic, or when c_i is itself a proxy produced by a misspecified classifier. Although robust and tail-sensitive variants (e.g. CVaR, entropic risk) can be incorporated at the level of the Lagrangian, the analysis becomes sensitive to higher moments and to the calibration of the oracle. A concrete open problem is to design constraint surrogates that (a) admit unbiased (or controlled-bias) stochastic gradients from black-box terminal samples, (b) preserve a tractable saddle-point structure, and (c) yield interpretable guarantees such as

$$\mathbb{P}\big(c_i(y_T) > \tau_i\big) \leq \delta \quad \text{or} \quad \mathrm{CVaR}_\alpha(c_i(y_T)) \leq \tau_i$$

with finite-episode bounds that do not scale poorly in $1/\delta$ or $1/(1-\alpha)$. One promising direction is to combine primal–dual updates with *calibration* of the constraint oracle (e.g. conformalized classifiers) so that the learned multiplier λ certifies a constraint with respect to a statistically valid upper confidence bound on c_i. This would shift the burden from assuming correct c_i to estimating it conservatively, but it requires integrating uncertainty quantification into the policy-induced sampling distribution, which is itself changing during training.

A third open issue is the on-policy nature of the algorithm. The martingale residual construction is naturally compatible with online rollouts, yet the sample complexity of diffusion generation is dominated by simulator steps, making off-policy reuse appealing. Off-policy learning in controlled diffusions raises nontrivial measure-change questions: trajectories generated under an earlier policy $\pi_{\psi'}$ are distributed according to a different path measure than those under π_ψ, and importance weighting must account for continuous-time likelihood ratios (Girsanov transforms). In discretized form, one can write a product of Gaussian action likelihood ratios, but this can have high variance over long horizons K. An open problem is to develop a replay scheme with provable variance control, for example by (i) truncating the horizon and using multi-step objectives, (ii) employing pathwise control variates linked to the quadratic structure in a, or (iii) learning a density ratio model for the state–action occupancy measure. The constrained setting further complicates matters because off-policy estimates of $\mathbb{E}[c_i(y_T)]$ can be biased if replay trajectories come from policies with systematically different terminal distributions. Establishing that replay buffers preserve feasibility (even approximately) appears to require new arguments beyond standard off-policy actor–critic analyses.

Fourth, our discretization model is intentionally simple: we assume a fixed step size Δt and treat the induced error as an additive DiscErr term. This is unsatisfactory in regimes where $g(T-t)$ varies by orders of magnitude or where the dynamics are stiff, as is common in variance-preserving diffusions near $t \approx T$. Adaptive step-size solvers are then practically necessary, but they introduce two conceptual difficulties. First, the effective time grid becomes random and policy-dependent, which interacts with SA since the update noise is no longer identically distributed across steps. Second, if one switches between SDE and ODE modes or uses higher-order solvers, one must specify which objective is being optimized: the continuous-time control problem, the discretized controlled Markov chain, or a solver-dependent approximation. A precise theory would treat the solver as part of the environment and quantify the bias in the estimated martingale residuals as a function of local truncation error. It remains open to obtain end-to-end bounds of the form

$$\text{primal–dual gap} \;\leq\; \tilde{O}(N^{-1/2}) \;+\; O(\mathbb{E}[\text{solver error}]),$$

where the solver error is controlled adaptively per trajectory under a fixed compute budget.

Fifth, the restriction to a Gaussian policy with fixed covariance is both a strength and a limitation. It is a strength because it yields a closed-form variational structure (Theorem 1) and prevents degeneracy in exploration by maintaining a minimum-entropy behavior proportional to θ. However, it limits expressivity: the optimal surrogate score a_t may be multimodal given (t, y) (e.g. when the conditional score is itself multimodal under ambiguous contexts), and a unimodal Gaussian may be an inefficient parametrization. Moreover, fixing the covariance to $\frac{\theta}{2g(T-t)^2} I$ hard-codes an exploration schedule that may be mismatched to the local geometry of p_{T-t} and to the noise level of the score signal. Natural extensions include (i) learning a state-dependent covariance $\Sigma_\psi(t, y)$, (ii) using mixture-of-Gaussians policies, or (iii) employing normalizing-flow policies in a to capture heavy tails and skew. The challenge is that the quadratic running penalty $-g^2 \|\nabla \log p - a\|^2$ interacts favorably with Gaussian entropy regularization; once we depart from this family, the identity $\pi_\psi \propto \exp(q_\psi/\theta)$ may no longer yield tractable sampling or stable gradients. A key open problem is to identify alternative regularizers (e.g. f-divergences or Wasserstein penalties) that preserve computability and admit a comparable martingale-residual learning rule.

Sixth, the quality of the ratio-based score signal remains a practical and theoretical bottleneck in high dimension. Our assumptions treat $\hat{r}$ as a noisy but controlled approximation of the true running reward. Yet the ratio estimator relies on finite minibatches from p_0 and on evaluating $p_{t|0}(y \mid x_0)$, which can concentrate sharply as d grows. This raises a question of *informational sufficiency*: for which classes of p_0 and time horizons T does an on-policy, local score signal contain enough information to guide the policy toward high terminal reward while respecting constraints? Conversely, can one prove impossibility results showing that, even ignoring terminal noise, any algorithm that only accesses p_0 through such local ratios requires exponentially many samples in d to achieve a nontrivial approximation? Clarifying this would help delineate when reward-directed diffusion via local score surrogates is viable and when one must fall back to globally learned score networks.

Finally, there is a conceptual open problem regarding certificates and stopping criteria. In constrained optimization one often desires a posteriori guarantees (upper bounds on constraint violation and primal suboptimality) at finite N. Our primal–dual iterates provide a natural certificate in the form of λ, but turning this into a quantitative bound requires estimating the primal–dual gap, which in turn depends on quantities (true scores, true constraint expectations) that are not directly observable. Developing data-dependent confidence intervals for feasibility and performance, based solely on terminal samples and score-signal diagnostics collected along trajectories, would substantially improve practical reliability. We view such certification as essential for deploying constrained diffusion samplers in safety-critical settings and as a natural next step in the theory.