Off-Policy Martingale q-Learning for Reward-Directed Diffusion with Replay and Normalization Constraints

Liz Lemma Future Detective

January 19, 2026

Abstract

Reward-directed diffusion models aim to generate samples that maximize a reward while remaining close to the underlying data distribution. Recent continuous-time formulations treat the unknown score in the reverse-time SDE as the action and introduce a running KL-type penalty to the true score; however, actor–critic training is expensive because it is typically on-policy. We develop an off-policy variant of martingale-based little q-learning for diffusion control that uses a replay buffer, importance weighting, and an explicit q-normalization regularizer consistent with entropy-regularized optimal control. Our method minimizes a weighted squared martingale-residual objective over replayed transitions of the discretized diffusion environment. In a linear/NTK approximation regime and under bounded importance weights, we establish convergence and finite-time bounds that separate (i) stochastic approximation error, (ii) score-signal noise from data-driven running rewards, and (iii) discretization error. Empirically, we expect a drop-in replacement for on-policy diffusion RL that attains equal (or better) FID at matched reward with 2–5× fewer environment rollouts. Experiments on CIFAR-10 incompressibility and a modern 2026 reward (aesthetic/CLIP-based) would validate the compute–quality gains and characterize when off-policy replay remains stable.

Table of Contents

3. 3. Problem Setup: discretized controlled reverse SDE, running reward via score-signal estimator, terminal reward oracle, and the entropy-regularized objective.

4. 4. Off-Policy Formulation: replay distribution, behavior policy dominance, importance weighting, and identification of martingale moment equations from off-policy data.

5. 5. Algorithm: replay-based critic regression + actor improvement with importance weights; q-normalization penalty and practical stabilizers (clipping, trust region, target networks).

6. 6. Main Theorems: (i) correctness of weighted moment equations, (ii) almost sure convergence under linear/NTK assumptions, (iii) finite-time bounds, (iv) discretization + score-signal error propagation, (v) lower bounds via bandit reduction and necessity of bounded importance weights.

7. 7. Complexity and Optimality: time/space costs, rollouts vs gradient steps trade-off, and when replay yields provable improvements.

8. 8. Experiments (recommended): CIFAR-10 incompressibility + aesthetic/CLIP reward; comparisons to on-policy q-learning and DPOK; ablations on replay size, importance clipping, and normalization penalty.

9. 9. Discussion and Extensions: ODE samplers, conditional diffusion, retrieval-augmented score signals, constrained/safe objectives.

1 Introduction

We consider the problem of constructing a diffusion sampler whose trajectories are biased toward configurations of high utility while remaining close, in a controlled sense, to a given data distribution. The setting is motivated by "reward-guided" generation tasks in which one possesses (i) samples from an unknown data law p_0 (e.g. images, molecules, or other structured objects) and (ii) an external preference signal that assigns larger values to desirable outcomes. In the diffusion formalism, the baseline generative mechanism is specified through a forward corruption process with known coefficients, and sampling is performed by simulating an associated reverse-time dynamics. Our central observation is that, once the reverse dynamics are written in controlled form, the score function (or a score-like surrogate) becomes a natural control variable, and diffusion sampling may be treated as a continuous-time control problem whose state is the latent variable and whose action is a drift adjustment.

Concretely, let $t \in [0, T]$ denote reverse-time, and let $y_t \in \mathbb{R}^d$ be the reverse process state. For standard forward diffusions (e.g. Ornstein–Uhlenbeck or variance-preserving schedules), the reverse-time evolution admits a representation of the schematic form

$$dy_t \;=\; \big(\text{known drift}(t, y_t) \;+\; g(T-t)^2\, a(t, y_t)\big)\, dt \;+\; g(T-t)\, d\bar{B}_t,$$

where $a(t, y)$ is the control applied at time t in state y and $\bar{B}_t$ is a reverse-time Brownian motion. In the uncontrolled model, the optimal choice is $a(t, y) = \nabla \log p_{T-t}(y)$, which is the score of the intermediate-time marginal of the forward process. Thus, in a reward-directed variant, it is mathematically natural to interpret a as a *score control*, namely a field that deforms the reverse drift away from the baseline score so as to increase some reward functional, while paying a cost for deviating from the data-consistent dynamics.

We focus on an entropy-regularized formulation in which the objective combines a running penalty for score deviation with a terminal reward at time T. At a high level, we seek a policy $\pi(\cdot \mid t, y)$ over actions a maximizing

$$\mathbb{E}\Big[\int_0^T \big(r(t, y_t, a_t) - \theta \log \pi(a_t \mid t, y_t)\big)\, dt + \beta\, h(y_T)\Big],$$

where $\theta > 0$ is the entropy temperature, $\beta \geq 0$ weights the terminal utility $h(y_T)$, and the running reward takes the quadratic form

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

This running term encourages faithfulness to the data distribution by penalizing departures from the (unknown) true score, while leaving freedom to trade fidelity for terminal utility. The entropy term makes the optimal policy stochastic and yields a convenient Gaussian structure in many cases; moreover, it provides numerical stability when learning from noisy reward signals.

The principal difficulty is that the score $\nabla \log p_t$ is unknown, since p_0 is unknown. We assume only data access $x_0 \sim p_0$, together with a computable score-value estimator $\widehat{\nabla} \log p_t(\cdot)$ (potentially biased) obtained from denoising-score training or related techniques. The running reward is therefore not directly available; instead we observe, along simulated reverse trajectories, a sample $\hat{r}$ computed from $\widehat{\nabla} \log p_{T-t}(y)$. In addition, the terminal reward $h(y_T)$ may only be observed through an oracle that returns noisy samples. The resulting learning problem is thus an off-policy reinforcement learning problem in a controlled diffusion environment with partial knowledge: the environment dynamics are simulatable given actions, but the reward is only available through noisy, data-driven signals.

A naive approach is to run an on-policy actor–critic method: generate trajectories under the current policy, estimate gradients from those trajectories, update the actor and critic, and repeat. In the present setting this is typically rollouts-limited. The reverse-time diffusion horizon T is discretized into $K = T/\Delta t$ steps, so a single trajectory already requires K environment steps; when K is large (as is common for faithful diffusion sampling), the marginal cost of collecting fresh trajectories becomes substantial. More importantly, when the terminal oracle is noisy, estimation of the effect of policy changes on $\mathbb{E}[h(y_T)]$ exhibits the familiar variance barrier: improving terminal reward by ε generically requires $\Omega(1/\varepsilon^2)$ episodes in the worst case. In such regimes, "on-policy" usage of trajectories is statistically and computationally inefficient, because each expensive rollout contributes only once to each update.

Our thesis is that the relevant efficiency gains come from *off-policy replay*: we should collect trajectories using a behavior policy b that maintains adequate coverage, store transitions in a replay buffer, and perform many gradient updates per environment rollout. While the statistical lower bound on the number of distinct terminal observations remains in force, replay permits us to extract more learning signal per observed transition by repeatedly solving the same conditional moment equations under different parameter values. The method is especially well-matched to the diffusion control viewpoint because the environment is simulatable and time-inhomogeneous but otherwise regular; hence we can separate the cost of data collection (rollouts) from the cost of computation (replay-based regression).

To make this precise, we work with a discretization $t_k = k\Delta t$ and a controlled Markov chain $(y_k)_{k=0}^{K}$ driven by actions a_k chosen according to a policy $\pi_\psi(\cdot \mid t_k, y_k)$. The critic is a value approximation $J_\Theta(t, y)$, and the actor is represented by a Gaussian policy with mean $\mu_\psi(t, y)$ and known covariance $\Sigma(t)$ determined by the entropy regularization and diffusion scale. The key analytical tool is a martingale characterization of optimality for the entropy-regularized controlled diffusion. In discretized form, it yields a one-step residual

$$\delta_k(\Theta, \psi) \;=\; J_\Theta(t_{k+1}, y_{k+1}) - J_\Theta(t_k, y_k) + \hat{r}_k \,\Delta t - q_\psi(t_k, y_k, a_k)\,\Delta t,$$

whose conditional expectation vanishes at the solution. The residual allows us to cast learning as a regression problem against a martingale difference, rather than as a direct policy-gradient estimate. This perspective is convenient for off-policy learning, because such conditional moment restrictions remain valid under trajectories generated by any behavior policy, provided we correct for distribution shift.

Off-policy learning introduces the standard obstacle of support mismatch. We therefore impose a dominance condition: for all visited (t, y), the behavior density $b(\cdot \mid t, y)$ must dominate the target policy density $\pi_\psi(\cdot \mid t, y)$, and the importance ratio

$$w(t, y, a) \;=\; \frac{\pi_\psi(a \mid t, y)}{b(a \mid t, y)}$$

must be bounded (or clipped) by a constant W. Under this coverage condition, we may estimate target-policy moments by replaying transitions collected under b and weighting by w. The resulting objective takes the form of a weighted squared residual $\mathbb{E}_b[w\,\delta^2]$, which we minimize over critic and actor parameters using stochastic gradient steps on minibatches drawn from replay.

Two additional features are essential in our diffusion context. First, because the running reward depends on a score estimator $\widehat{\nabla} \log p_t$, the signal $\hat{r}$ contains both noise and (potential) bias. We explicitly accommodate this by treating $\hat{r}$ as a corrupted version of the ideal reward and tracking its second-moment boundedness, so that replay-based stochastic approximation remains well-posed. Second, to ensure that the actor corresponds to a normalized policy (and, in generalized variants, that the associated soft q-function defines a valid Gibbs distribution), we incorporate an explicit q-normalization constraint or penalty. For the Gaussian policy class induced by entropy regularization, this normalization is exact; in more flexible parameterizations it must be enforced approximately.

The overall conclusion of the introduction is a structural one. In reward-directed diffusion, the controlled reverse-time dynamics provide a natural reinforcement learning environment with long horizons and expensive rollouts. On-policy updates are therefore constrained primarily by the number of full diffusion trajectories one can afford to simulate and evaluate. Off-policy replay, combined with a martingale-residual learning principle and bounded-importance correction, yields an algorithmic route to reduce the number of required rollouts for a desired level of performance by increasing the number of gradient steps per collected transition, while retaining a clean convergence theory under linear/NTK assumptions.

In the remainder of the paper we formalize the controlled reverse diffusion model, the entropy-regularized objective, and the martingale characterization that underlies our residual formulation. We then present the off-policy replay algorithm and analyze its convergence and finite-time behavior under bounded importance weights and standard regularity conditions, and we complement these upper bounds with a rollout lower bound obtained by a bandit reduction, clarifying which aspects of sample efficiency can and cannot be improved by replay.

2 Background: diffusion models, reverse-time control, and soft optimality

We briefly recall the continuous-time diffusion formalism and isolate the aspects that will later be used to formulate reward-directed sampling as an entropy-regularized control problem. Throughout, we work on a fixed time horizon $[0, T]$ and in ambient dimension d, and we assume that the forward diffusion coefficients are known through scalar schedules $f(\cdot), g(\cdot)$.

Forward diffusions and intermediate-time marginals. A canonical class of generative diffusions is given by the linear Itô SDE

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

where $(B_t)_{t\in[0,T]}$ is standard Brownian motion. Under mild conditions on f, g, (1) defines a Markov process with a family of marginal densities $(p_t)_{t\in[0,T]}$ (with p_0 unknown), and a transition density $p_{t|0}(\cdot \mid x_0)$ that is explicit for the usual choices of schedules (e.g. Ornstein–Uhlenbeck and variance-preserving cases). In particular, for each fixed $t > 0$, the forward corruption map $x_0 \mapsto x_t$ is Gaussian conditional on x_0, and thus can be simulated exactly (or to arbitrary precision) without knowing p_0.

The score $\nabla \log p_t(x)$ plays a central structural role in reverse-time sampling and in the control viewpoint we adopt. Since p_0 is unknown, the score is not directly available; nevertheless, the forward process (1) provides the standard denoising identity that motivates learning a parametric approximation to $\nabla \log p_t$ from data by score matching. We do not commit here to a particular estimator; it suffices that, given data and a time t, we can evaluate a score-like signal $\widehat{\nabla} \log p_t(\cdot)$ at arbitrary locations, potentially with bias and noise.

Reverse-time SDE and the score drift. Fix $T > 0$ and consider the time-reversed process $y_t := x_{T-t}$, $t \in [0, T]$. Under standard regularity assumptions, (y_t) satisfies a reverse-time SDE driven by a reverse Brownian motion $(\bar{B}_t)$ of the form

$$dy_t \;=\; \Big(-f(T-t)\, y_t \;+\; g(T-t)^2\, \nabla \log p_{T-t}(y_t)\Big)\, dt \;+\; g(T-t)\, d\bar{B}_t. \quad (2)$$

The key point is that the only term in (2) depending on the unknown data law is the score $\nabla \log p_{T-t}$. Hence, if we can approximate the score sufficiently well, then simulating (2) yields approximate samples from p_0 at time $t = T$. Conversely, if we wish to bias samples toward high utility, it is natural to deform precisely this score drift term, because doing so is the minimal intervention that preserves the diffusion structure while steering the terminal distribution.

Controlled reverse dynamics. We therefore introduce a controlled reverse-time SDE in which the score drift is replaced by a control field (the "action") $a(t, y) \in \mathbb{R}^d$:

$$dy_t \;=\; \Big(-f(T-t)\, y_t \;+\; g(T-t)^2\, a_t\Big)\, dt \;+\; g(T-t)\, d\bar{B}_t, \qquad a_t \sim \pi(\cdot \mid t, y_t). \quad (3)$$

The uncontrolled sampler is recovered by taking $a_t = \nabla \log p_{T-t}(y_t)$ (in which case the law of y_T matches p_0). From the control perspective, (3) is a time-inhomogeneous diffusion with affine control in the drift and fixed dispersion. This structure is particularly convenient: it admits well-developed dynamic programming characterizations, and it yields tractable "soft" optimal controls under entropy regularization.

Entropy regularization and soft dynamic programming. Let π be a (possibly stochastic) Markov policy. For a running reward $r(t, y, a)$ and a terminal reward $\beta h(y_T)$, we consider an entropy-regularized objective of the schematic form

$$J^\pi(t, y) \;:=\; \mathbb{E}^\pi\left[\int_t^T \Big(r(s, y_s, a_s) \;-\; \theta \log \pi(a_s \mid s, y_s)\Big)\, ds \;+\; \beta h(y_T) \;\Big|\; y_t = y\right], \quad (4)$$

where $\theta > 0$ is the temperature. In continuous time, one may regard the term $-\theta \log \pi$ as a control cost that penalizes low-entropy (overly concentrated) action distributions, thereby stabilizing learning and inducing an analytically tractable "soft" Bellman structure. In particular, if $V(t, y) := \sup_\pi J^\pi(t, y)$ denotes the optimal value, then V solves a soft Hamilton–Jacobi–Bellman (HJB) equation whose Hamiltonian is the log-partition (or convex conjugate) associated with the entropy term. Concretely, the HJB takes the form

$$\partial_t V(t, y) + \sup_{\pi(\cdot|t,y)} \Big\{\mathbb{E}_{a\sim\pi}\big[r(t, y, a)\big] - \theta\, \mathbb{E}_{a\sim\pi}\big[\log \pi(a \mid t, y)\big] + \mathcal{L}^\pi V(t, y)\Big\} \;=\; 0, \quad (5)$$

with terminal condition $V(T, y) = \beta h(y)$. Here $\mathcal{L}^\pi$ denotes the controlled diffusion generator applied to the value function:

$$\mathcal{L}^\pi V(t, y) \; = \; \mathbb{E}_{a \sim \pi(\cdot|t,y)} \left[\Big(-f(T-t)y + g(T-t)^2 a \Big)^\top \nabla_y V(t, y) \right] + \frac{1}{2} g(T-t)^2 \, \Delta_y V(t, y). \tag{6}$$

The variational form (5) implies a Gibbs characterization of the optimal policy in terms of an associated soft q-function, and it is this characterization that leads to a Gaussian policy structure in the diffusion setting.

Gaussian structure induced by quadratic control costs. In reward-directed diffusion, the canonical running term is quadratic in the deviation between a and the data-consistent score. Abstractly, if the immediate preference is of the form

$$r(t, y, a) \; = \; -\alpha(t) \, \|a - s(t, y)\|^2 \; + \; (\text{terms independent of } a), \tag{7}$$

for some weight $\alpha(t) > 0$ and some "reference" field $s(t, y)$ (in our application $s(t, y) = \nabla \log p_{T-t}(y)$), then the entropy-regularized maximization over action distributions at fixed (t, y) yields a Gaussian optimizer. Indeed, writing the soft advantage as an affine-quadratic function of a, the optimal density satisfies

$$\pi^*(a \mid t, y) \; \propto \; \exp\!\left(\frac{1}{\theta} \, Q^*(t, y, a) \right), \tag{8}$$

where $Q^*(t, y, a)$ collects the immediate reward and the value-gradient coupling coming from the generator term in (5). When $Q^*(t, y, a)$ is (at most) quadratic in a, (8) is a Gaussian density whose covariance is determined by the quadratic coefficient and whose mean is a linear transform of the linear coefficient. In the diffusion-control parameterization (3), the action enters the drift linearly as $g(T-t)^2 a$, so the contribution of $\mathcal{L}^\pi V$ to Q^* is linear in a, while the running reward is chosen quadratic; consequently, the optimal policy is Gaussian with a covariance that is known up to the diffusion scale and temperature. This observation motivates restricting attention to Gaussian policies with known covariance schedule and learnable mean, which reduces actor learning to estimating a mean field $\mu(t, y)$.

Martingale characterizations of (soft) optimality. A second structural fact we use is that, for a fixed policy π, the value function J^π admits a martingale characterization along trajectories of the controlled diffusion. Formally, if J^π is sufficiently smooth, then applying Itô's formula to $J^\pi(t, y_t)$ under (3) yields

$$dJ^\pi(t, y_t) \; = \; \Big(\partial_t J^\pi(t, y_t) + \mathcal{L}^\pi J^\pi(t, y_t) \Big) \, dt \; + \; g(T-t) \, \nabla_y J^\pi(t, y_t)^\top d\bar{B}_t. \tag{9}$$

By the definition (4), J^π satisfies the soft Bellman equation in differential form,

$$\partial_t J^\pi(t,y) + \mathcal{L}^\pi J^\pi(t,y) + \mathbb{E}_{a\sim\pi(\cdot|t,y)}\Big[r(t,y,a) - \theta \log \pi(a \mid t,y)\Big] \;=\; 0, \qquad J^\pi(T,y) = \beta h(y), \tag{10}$$

which, when substituted into (9), implies that the process

$$M_t \;:=\; J^\pi(t,y_t) \;+\; \int_0^t \Big(r(s,y_s,a_s) - \theta \log \pi(a_s \mid s, y_s)\Big)\, ds \tag{11}$$

is a local martingale (and, under standard integrability conditions, a martingale). At the optimal pair (V, π^*), the same identity holds with J^π replaced by V and with π^* attaining the soft supremum in (5). This martingale viewpoint is more than a reformulation: it yields conditional moment restrictions that remain valid when data are generated under an arbitrary behavior policy, a fact that is central for off-policy learning with replay.

The subsequent development will exploit (3)–(11) after discretizing time and replacing the unknown score in the running reward by a data-driven signal. In particular, once we pass to a time grid and a simulatable controlled Markov chain, we will obtain a one-step martingale residual whose conditional expectation vanishes at the correct value–policy pair, thereby enabling weighted regression updates from replayed transitions.

3 Problem setup: discretized controlled reverse diffusion with score-based running rewards

We now formalize the learning problem as a discrete-time control task induced by the controlled reverse-time SDE (3). Fix a discretization step $\Delta t > 0$ and let $K := T/\Delta t \in \mathbb{N}$ with time grid $t_k := k\Delta t$ for $k = 0, \ldots, K$. Our primitive interface is the ability to (i) initialize the reverse process from a known prior ν (typically Gaussian), (ii) apply an action over each interval $[t_k, t_{k+1})$, (iii) simulate one step of the resulting discretized reverse dynamics, and (iv) query a noisy running-reward signal derived from a data-driven score estimator, together with a noisy terminal-reward oracle evaluated at the final state.

Discretized controlled reverse dynamics. We work with the Euler–Maruyama discretization of (3), which yields a controlled Markov chain $(y_k)_{k=0}^K$ in $\mathbb{R}^d$. Given (t_k, y_k) and an action $a_k \in \mathbb{R}^d$ applied on $[t_k, t_{k+1})$, we define

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

with $(\xi_k)_{k=0}^{K-1}$ i.i.d. and independent of $y_0 \sim \nu$. We emphasize that the environment transition (12) is simulatable since the schedules f, g are assumed known and the injected noise is explicit. The action a_k is interpreted as a *score control*: in the ideal (uncontrolled) sampler one would take $a_k = \nabla \log p_{T-t_k}(y_k)$, whereas in reward-directed sampling we allow a_k to deviate from the data score in a state- and time-dependent manner.

A (stochastic) policy π is a family of conditional densities $\pi(\cdot \mid t, y)$ on $\mathbb{R}^d$; the interaction protocol is

$$a_k \sim \pi(\cdot \mid t_k, y_k), \qquad y_{k+1} \sim P(\cdot \mid t_k, y_k, a_k), \tag{13}$$

where P denotes the transition kernel induced by (12). Our target policy class is Gaussian with known covariance schedule,

$$\pi_\psi(\cdot \mid t, y) \;=\; \mathcal{N}\big(\mu_\psi(t, y), \Sigma(t)\big), \qquad \Sigma(t) \;=\; \frac{\theta}{2\, g(T-t)^2}\, I_d, \tag{14}$$

so that learning reduces to fitting the mean field $\mu_\psi(t, y)$ (parametrized linearly or in an NTK regime as assumed in the enclosing scope). The particular choice (14) is aligned with the quadratic running reward introduced below: the diffusion scale $g(T-t)$ and temperature θ jointly determine the natural action variance in the entropy-regularized optimum, and fixing $\Sigma(t)$ removes an otherwise ill-conditioned degree of freedom.

Score-signal estimator and running reward samples. The defining constraint in diffusion-based generation is that the unknown data law p_0 only enters the reverse dynamics through the score $\nabla \log p_{T-t}(y)$. We assume access to i.i.d. data $(x_0^i)_{i=1}^M \sim p_0$ and, using any standard score-learning or denoising mechanism, we may evaluate a score-like signal $\widehat{\nabla} \log p_t(\cdot)$ at arbitrary (t, y). We do not require this signal to be unbiased; rather, we treat it as an exogenous estimator with controlled second moments and (possibly) nonzero bias, which will later appear explicitly in error decompositions.

Given an action a_k and state y_k at time t_k, we define the *ideal* running reward (unknown to the learner) by

$$r(t_k, y_k, a_k) \;:=\; -\, g(T-t_k)^2 \left\|\nabla \log p_{T-t_k}(y_k) - a_k\right\|^2. \tag{15}$$

This choice is canonical in our setting: it penalizes deviation from the data-consistent reverse drift at a scale commensurate with the diffusion dispersion, and therefore enforces fidelity to the data distribution except where reward incentives justify controlled deviations. In practice we only observe a sample-based surrogate computed from the score-signal estimator,

$$\hat{r}_k \;:=\; -\, g(T-t_k)^2 \left\|\widehat{\nabla} \log p_{T-t_k}(y_k) - a_k\right\|^2, \tag{16}$$

which is available during simulation. It is useful to record the decomposition

$$\hat{r}_k \;=\; r(t_k, y_k, a_k) \;+\; \varepsilon_k, \tag{17}$$

where ε_k aggregates both estimation noise and systematic error induced by replacing $\nabla \log p_{T-t_k}$ with $\widehat{\nabla} \log p_{T-t_k}$. Our standing assumption is that ε_k has bounded conditional second moment (and, when needed, bounded conditional bias) given (t_k, y_k, a_k). No further structure is required at the level of the problem definition.

Terminal reward oracle. In addition to the running penalty (15), we seek to bias the terminal sample toward high utility as measured by an application-specific functional $h : \mathbb{R}^d \to \mathbb{R}$. We do not assume that h is known analytically nor differentiable. Instead, we assume an oracle that produces a noisy observation of h at the terminal state:

$$\hat{h}_K \;=\; h(y_K) \;+\; \zeta, \tag{18}$$

where ζ is observation noise with bounded variance. The scalar $\beta \geq 0$ weights the terminal reward relative to the running score-deviation penalty; large β encourages aggressive steering, while small β favors fidelity to p_0. Importantly, the terminal reward is only observed at the end of each rollout, so the algorithm must propagate its effect backward through the dynamics via value estimation rather than by direct per-step supervision.

Entropy-regularized discrete-time objective. For a policy π and initial distribution ν, we define the entropy-regularized return of the discretized controlled reverse process by

$$J^{\pi}_{\Delta t}(0, \nu) \;:=\; \mathbb{E}^{\pi}\left[\sum_{k=0}^{K-1}\Big(r(t_k, y_k, a_k) \;-\; \theta \log \pi(a_k \mid t_k, y_k)\Big)\Delta t \;+\; \beta\, h(y_K)\right], \tag{19}$$

where the expectation is taken over $y_0 \sim \nu$, actions sampled from π, and the Gaussian innovations in (12). The term $-\theta \log \pi$ is interpreted as an entropy regularizer (equivalently, a control cost) and is scaled by Δt to match the continuous-time objective as $\Delta t \to 0$. Although (19) depends on the unknown score through r, it is well defined as a population objective and will serve as our learning target.

Operationally, the learner has access only to $\hat{r}_k$ and $\hat{h}_K$. Accordingly, each rollout produces an unbiased (or biased-but-controlled) sample of the form

$$\widehat{G} \;:=\; \sum_{k=0}^{K-1}\Big(\hat{r}_k \;-\; \theta \log \pi(a_k \mid t_k, y_k)\Big)\Delta t \;+\; \beta\, \hat{h}_K, \tag{20}$$

whose expectation differs from $J^{\pi}_{\Delta t}(0, \nu)$ by the cumulative score-signal and terminal-oracle biases. The role of the value function approximation introduced later is to enable learning from these samples in a way that is stable under noise and compatible with off-policy replay.

Soft q-parameterization under Gaussian policies. Because our policy class (14) is Gaussian with known covariance, the entropy term can be absorbed into a convenient "soft q" representation. In particular, defining

$$q_\psi(t, y, a) \;:=\; \theta \log \pi_\psi(a \mid t, y), \tag{21}$$

we obtain a quadratic function of a (up to terms independent of a):

$$q_\psi(t, y, a) \;=\; -\frac{\theta}{2}\,(a - \mu_\psi(t, y))^\top \Sigma(t)^{-1}(a - \mu_\psi(t, y)) \;+\; c(t), \tag{22}$$

with $c(t)$ collecting the log-normalizer. This explicit normalization is not merely cosmetic: it guarantees that $\int \exp(q_\psi(t, y, a)/\theta)\, da \;=\; 1$ for every (t, y), which will later eliminate the need for approximate partition-function penalties in the Gaussian case.

Learning problem. The problem is therefore to choose parameters ψ (and, in actor–critic form, auxiliary value parameters Θ) such that the induced sampler—the terminal state y_K obtained by simulating (12) under π_ψ from $y_0 \sim \nu$—achieves high terminal utility while remaining close to the data distribution in the precise sense enforced by the running penalty (15). The central difficulty is that the per-step reward depends on the unknown score and is only observed through the noisy proxy (16), while the terminal reward is available only via the oracle (18). In the next section we recast this problem in an off-policy framework suitable for replay, and we identify moment equations that remain valid under arbitrary data-collection policies.

4 Off-policy formulation: replay, dominance, and martingale moment equations

We now recast the discretized control problem (12)–(19) into an off-policy learning setting in which data are collected under an arbitrary *behavior* policy b and subsequently reused via replay to learn a (possibly different) *target* policy π_ψ. The goal of this section is twofold: (i) to formalize the replay distribution induced by b and the role of importance weighting, and (ii) to identify martingale-based moment equations that characterize the optimal entropy-regularized solution and remain valid under off-policy data collection.

Behavior policy and replay distribution. A behavior policy is any family of densities $b(\cdot \mid t, y)$ from which we can sample actions during rollouts. Running the interaction protocol (13) with $\pi = b$ produces trajectories

$$(y_0, a_0, y_1, a_1, \ldots, y_{K-1}, a_{K-1}, y_K),$$

together with observed running rewards $(\hat{r}_k)_{k=0}^{K-1}$ from (16) and a terminal observation $\hat{h}_K$ from (18). Each rollout therefore yields a collection of transitions of the form

$$(t_k, y_k, a_k, \hat{r}_k, y_{k+1}), \qquad k = 0, \dots, K-1,$$

which we store in a replay buffer $\mathcal{D}$. Sampling uniformly (or with a fixed priority rule) from $\mathcal{D}$ induces an empirical distribution over transitions; in analysis we idealize this by a *replay distribution* ρ_b over $(t, y, a, \hat{r}, y')$ defined as the average occupancy measure under b,

$$\rho_b(\cdot) \;:=\; \frac{1}{K} \sum_{k=0}^{K-1} \mathcal{L}\big(t_k, y_k, a_k, \hat{r}_k, y_{k+1}\big) \quad \text{when } a_k \sim b(\cdot \mid t_k, y_k), \tag{23}$$

where $\mathcal{L}(\cdot)$ denotes the law of the indicated random tuple. This distribution is the fundamental sampling distribution for all subsequent regression objectives.

Coverage and dominance. Since our aim is to learn a target policy π_ψ using data generated by b, we require the standard dominance condition: whenever (t, y) is visited with non-negligible probability during replay, the behavior density must be strictly positive wherever the target density is positive. Concretely, we assume that for all (t, y) in the support of the replay state distribution,

$$\operatorname{supp}\big(\pi_\psi(\cdot \mid t, y)\big) \;\subseteq\; \operatorname{supp}\big(b(\cdot \mid t, y)\big), \tag{24}$$

so that the importance ratio

$$w(t, y, a) \;:=\; \frac{\pi_\psi(a \mid t, y)}{b(a \mid t, y)} \tag{25}$$

is well-defined. We additionally impose (or enforce by clipping) a uniform bound

$$0 \le w(t, y, a) \le W, \tag{26}$$

which is essential for controlling variance of off-policy estimates and will explicitly enter our convergence and finite-time guarantees. In practice (26) is achieved by choosing b as a mixture of recent target policies and exploratory noise, and by storing (or recomputing) the behavior log-density needed to evaluate (25).

Value functions and a one-step residual. Let us fix any target policy π (not necessarily equal to b). We define the entropy-regularized value function at grid time t_k as the conditional expected return from (19):

$$J^{\pi}_{\Delta t}(t_k, y) \;:=\; \mathbb{E}^{\pi}\left[\sum_{\ell=k}^{K-1}\Big(r(t_\ell, y_\ell, a_\ell) - \theta \log \pi(a_\ell \mid t_\ell, y_\ell)\Big)\Delta t + \beta\, h(y_K) \;\middle|\; y_k = y\right]. \tag{27}$$

As usual, $J^{\pi}_{\Delta t}(t_K, y) = \beta\, h(y)$ serves as the terminal condition. For subsequent identification arguments it is convenient to rewrite the per-step entropy term using a generic function $q(t, y, a)$ (later instantiated as $q_\psi = \theta \log \pi_\psi$ as in (21)). Given any candidate pair (J, q), we define the (ideal, unobserved) one-step residual

$$\delta_k^{\mathrm{true}}(J, q) \;:=\; J(t_{k+1}, y_{k+1}) - J(t_k, y_k) \;+\; r(t_k, y_k, a_k)\Delta t \;-\; q(t_k, y_k, a_k)\Delta t, \tag{28}$$

and the corresponding residual based on the observed running reward proxy,

$$\delta_k(J, q) \;:=\; J(t_{k+1}, y_{k+1}) - J(t_k, y_k) \;+\; \hat{r}_k \Delta t \;-\; q(t_k, y_k, a_k)\Delta t. \tag{29}$$

By (17), these differ by an additive noise term:

$$\delta_k(J, q) \;=\; \delta_k^{\mathrm{true}}(J, q) \;+\; \varepsilon_k \Delta t. \tag{30}$$

We emphasize that ε_k may be biased; our assumption is only that it admits bounded conditional second moments (and, when needed, bounded conditional bias) given (t_k, y_k, a_k).

Martingale characterization and off-policy identification. Let $(\mathcal{F}_{t_k})_{k=0}^{K}$ denote the natural filtration generated by the trajectory up to time t_k (including states, actions, and reward observations up to that time). The defining property of the optimal entropy-regularized solution in our setting is that, for the appropriate optimal pair (J^*, q^*), the process of one-step residuals forms a martingale difference sequence. At the discrete level this may be stated as the conditional moment restriction

$$\mathbb{E}\big[\delta_k^{\mathrm{true}}(J^*, q^*) \,\big|\, \mathcal{F}_{t_k}\big] = 0, \qquad k = 0, \ldots, K-1, \tag{31}$$

with the understanding that $J^*(t_K, \cdot) = \beta h(\cdot)$. Importantly, (31) is a *model-based* identity induced by the controlled Markov structure of (12) and does not depend on how the data were collected: as long as the transition (y_k, a_k, y_{k+1}) is generated by the environment dynamics with some action a_k, the conditional expectation of the residual at the solution vanishes. This is the sense in which the martingale equations are identifiable from off-policy data.

Passing from conditional to unconditional moments, (31) implies that for any square-integrable test function ϕ measurable with respect to (t_k, y_k, a_k),

$$\mathbb{E}\big[\phi(t_k, y_k, a_k)\, \delta_k^{\mathrm{true}}(J^*, q^*)\big] = 0. \tag{32}$$

When we replace r by $\hat{r}$, the same moment equation holds up to the additive score-signal error:

$$\mathbb{E}\big[\phi(t_k, y_k, a_k)\,\delta_k(J^*, q^*)\big] \;=\; \Delta t\,\mathbb{E}\big[\phi(t_k, y_k, a_k)\,\varepsilon_k\big], \tag{33}$$

so the bias and variance of the score estimator enter explicitly through ε_k.

Why importance weighting still appears. Although (32) is valid under any data-collection strategy, the *projection* implicit in function approximation depends on the distribution with respect to which we fit J_Θ and q_ψ. In particular, if we wish to approximate a target-policy object using samples from b, we may correct the mismatch between action distributions by importance weighting. For any integrable function $F(t, y, a)$ and any fixed (t, y), dominance (24) yields

$$\mathbb{E}_{a\sim\pi_\psi(\cdot|t,y)}[F(t, y, a)] \;=\; \mathbb{E}_{a\sim b(\cdot|t,y)}\big[w(t, y, a)\,F(t, y, a)\big]. \tag{34}$$

Accordingly, actor and critic objectives that are naturally expressed as expectations under π_ψ can be estimated from replay by weighting each transition with w. The bound (26) ensures that such corrections do not introduce uncontrolled variance.

Projected martingale equations under replay. Let J_Θ and q_ψ be our parametric approximators. In the Gaussian policy setting (14) we will typically take $q_\psi = \theta \log \pi_\psi$ as in (21), but for the present discussion it suffices to treat q_ψ as a function indexed by ψ. A canonical way to exploit (32) with replay is to choose a class of test functions ϕ spanning the feature space (e.g., components of the critic feature map) and solve the resulting system in least-squares form. This motivates the weighted squared-residual objective

$$L(\Theta, \psi) \;:=\; \mathbb{E}_{(t,y,a,\hat{r},y')\sim\rho_b}\Big[w(t, y, a)\,\big(J_\Theta(t', y') - J_\Theta(t, y) + \hat{r}\,\Delta t - q_\psi(t, y, a)\Delta t\big)^2\Big], \tag{35}$$

where (t', y') denotes the successor time-state pair. Minimizing (35) corresponds to solving a projected version of the martingale equations under the replay sampling distribution, while approximately correcting for action mismatch via w. The specific algorithmic choices for optimizing (35) (including clipping w, trust regions for the actor, and stabilization via target networks) are deferred to the next section.

Terminal condition under noisy oracle access. Finally, we note that the terminal boundary condition $J(t_K, y) = \beta h(y)$ is not directly evaluable. In practice we enforce it through samples $\beta\hat{h}_K$ by treating the final step as a regression target: at $k = K - 1$ the term $J(t_{k+1}, y_{k+1})$ in (29) is replaced by $\beta\hat{h}_K$ (equivalently, one may define an absorbing terminal time with observed terminal value). This introduces an additional noise source independent of the score-signal error ε_k, and it is handled in the same martingale-residual framework by tracking its contribution to the residual variance (and, when present, bias).

The outcome of this section is that off-policy rollouts under a behavior policy b provide replayable samples whose statistics identify the martingale moment conditions characterizing optimality; bounded importance weights then permit stable correction for the discrepancy between b and the evolving target policy π_ψ. In the next section we instantiate these principles into a practical replay-based actor–critic algorithm.

5 Replay-based martingale q-learning: critic regression and actor improvement

We now instantiate the off-policy objective (35) into a replay-based actor–critic procedure. The algorithmic design is guided by two constraints that are specific to our setting: (i) the only per-step reward access is through the noisy score-driven signal $\hat{r}_k$, and (ii) the entropy-regularized control problem admits a convenient parameterization in which the policy normalization is either exact (Gaussian policy-induced q) or must be enforced (general q-network). We describe both cases and the corresponding stabilizers.

Data structures and bookkeeping. Each rollout under the behavior policy b produces transitions $(t_k, y_k, a_k, \hat{r}_k, y_{k+1})$. In addition, for reliable off-policy correction we record the behavior log-density $\log b(a_k \mid t_k, y_k)$ (or enough information to recompute it), and we optionally store an "old" target log-density $\log \pi_{\psi_{\text{old}}}(a_k \mid t_k, y_k)$ when using trust regions. We denote by $\mathcal{D}$ the replay buffer containing these tuples and by $B \subset \mathcal{D}$ a minibatch sampled uniformly (or via a fixed priority rule).

Clipped importance weights. Given a current target policy π_ψ, each replayed transition is assigned an importance weight

$$w_\psi(t, y, a) = \frac{\pi_\psi(a \mid t, y)}{b(a \mid t, y)}, \qquad \bar{w}_\psi(t, y, a) := \min\{w_\psi(t, y, a), W\}, \quad (36)$$

where W is the prescribed bound. In all objectives below we use $\bar{w}_\psi$ rather than w_ψ; this preserves the desired correction when $w_\psi \le W$ and prevents uncontrolled variance otherwise.

Critic regression via weighted squared martingale residuals. We update the critic by weighted regression on the one-step residual (29). In practice we employ a target network (or Polyak-averaged copy) Θ^- to stabilize the bootstrapped term $J(t_{k+1}, y_{k+1})$. Concretely, for a transition

$(t, y, a, \hat{r}, y')$ with successor time $t' := t + \Delta t$ we define the residual used for learning as

$$\delta^{-}_{\Theta,\psi}(t, y, a, \hat{r}, y') \;:=\; J_{\Theta^-}(t', y') \;-\; J_{\Theta}(t, y) \;+\; \hat{r}\,\Delta t \;-\; q_{\psi}(t, y, a)\,\Delta t. \quad (37)$$

At the terminal step (i.e. when $t' = T$) we replace $J_{\Theta^-}(T, y')$ by the sampled terminal value $\beta\hat{h}(y')$, which yields the same expression with $J_{\Theta^-}(T, y') := \beta\hat{h}(y')$. The critic loss is then

$$L_V(\Theta; \psi) \;:=\; \mathbb{E}_{(t,y,a,\hat{r},y')\sim\rho_b}\Big[\bar{w}_{\psi}(t, y, a)\,\big(\delta^{-}_{\Theta,\psi}(t, y, a, \hat{r}, y')\big)^2\Big], \quad (38)$$

approximated by the empirical average over a minibatch. The gradient step is the usual stochastic gradient descent on (38) with Θ^- updated slowly, e.g.

$$\Theta^- \leftarrow (1-\tau)\Theta^- + \tau\Theta, \qquad \tau \in (0, 1]. \quad (39)$$

When J_Θ is linear (or in the NTK regime), (38) is a (nearly) convex weighted least-squares objective, and replay simply increases the number of effective stochastic approximation steps per collected transition.

Actor update as residual minimization (projected martingale fitting). The most direct actor update is to minimize the same weighted residual objective with respect to ψ while holding Θ fixed:

$$L^{\mathrm{res}}_{\pi}(\psi; \Theta) \;:=\; \mathbb{E}_{(t,y,a,\hat{r},y')\sim\rho_b}\Big[\bar{w}_{\psi}(t, y, a)\,\big(\delta^{-}_{\Theta,\psi}(t, y, a, \hat{r}, y')\big)^2\Big]. \quad (40)$$

This update is natural from the moment-equation viewpoint: it attempts to choose ψ so that the projected martingale restriction is better satisfied under the target-policy parameterization. In the Gaussian policy class $\pi_\psi(\cdot \mid t, y) = \mathcal{N}(\mu_\psi(t, y), \Sigma(t))$ with known $\Sigma(t)$, gradients of $\log \pi_\psi$ and hence of $q_\psi = \theta \log \pi_\psi$ are explicit, so (40) admits low-variance gradient estimates.

Actor update as soft policy improvement. Although (40) is conceptually aligned with the martingale equation, in practice we often prefer an improvement-style update that uses the critic to define an advantage-like signal. One convenient choice is to interpret the negative squared residual as a surrogate for local consistency and maximize its expectation, equivalently minimizing (40); another is to adopt a soft actor–critic form when we maintain an explicit soft q-estimate q_φ:

$$L^{\mathrm{imp}}_{\pi}(\psi; \varphi) \;:=\; -\,\mathbb{E}_{(t,y)\sim\rho_b}\Big[\mathbb{E}_{a\sim\pi_\psi(\cdot|t,y)}\big[q_\varphi(t, y, a) - \theta \log \pi_\psi(a \mid t, y)\big]\Big], \quad (41)$$

with the inner expectation computed via reparameterization $a = \mu_\psi(t, y) + \Sigma(t)^{1/2}\xi$, $\xi \sim \mathcal{N}(0, I)$. The importance weights then enter through the standard identity (34) when we estimate (41) from behavior actions; in particular, if we reuse sampled actions $a \sim b(\cdot \mid t, y)$ then we weight by $\bar{w}_\psi(t, y, a)$.

q-normalization: exact in the Gaussian case, penalized otherwise. A distinctive feature of our setting is that the entropy term is not an auxiliary regularizer but part of the correct continuous-time control objective, and thus the normalization of the policy-induced quantity

$$\log Z_q(t,y) \;:=\; \log \int \exp\bigl(q(t,y,a)/\theta\bigr)\,da \tag{42}$$

matters. If we parameterize q_ψ *by definition* as $q_\psi = \theta \log \pi_\psi$ for a normalized Gaussian density, then $\log Z_{q_\psi}(t,y) = 0$ holds identically for all (t,y), and no additional constraint is needed.

If instead we use a flexible q-network $q_\varphi(t,y,a)$ (e.g. to decouple critic and actor parameterizations), then $\log Z_{q_\varphi}(t,y)$ is not automatically zero. To prevent drift that would otherwise corrupt the entropy-regularized interpretation, we add a penalty

$$L_{\mathrm{norm}}(\varphi) \;:=\; \lambda\,\mathbb{E}_{(t,y)\sim\rho_b}\Bigl[\bigl(\log Z_{q_\varphi}(t,y)\bigr)^2\Bigr], \qquad \lambda > 0, \tag{43}$$

and optimize $L_V(\Theta;\psi)$ (or the corresponding q-loss) augmented by L_{norm}. The integral (42) is approximated either by Monte Carlo sampling $a^{(j)} \sim \tilde{\pi}(\cdot \mid t,y)$ from a proposal distribution (typically the current Gaussian policy) with log-sum-exp stabilization, or by low-dimensional quadrature when d_a is small. The role of (43) is not to enforce optimality by itself, but to ensure that the learned q remains interpretable as a log-density up to the temperature θ, which is required for stable actor updates of the form (41).

Practical stabilizers: trust regions, delayed updates, and reward/weight clipping. We briefly record the stabilizers we use in implementations and in the proofs when needed.

1. *Trust region / KL-to-old policy.* To maintain the dominance condition operationally and to limit extrapolation error from replay, we constrain policy updates by penalizing deviation from a lagged policy $\pi_{\psi_{\mathrm{old}}}$:

$$L_{\mathrm{KL}}(\psi) \;:=\; \eta_{\mathrm{KL}}\,\mathbb{E}_{(t,y)\sim\rho_b}\bigl[\mathrm{KL}(\pi_\psi(\cdot \mid t,y)\,\|\,\pi_{\psi_{\mathrm{old}}}(\cdot \mid t,y))\bigr], \tag{44}$$

   with $\eta_{\mathrm{KL}} > 0$, updating ψ_{old} only occasionally.

2. *Delayed actor updates.* We may update the actor less frequently than the critic, which empirically reduces nonstationarity of targets in (37).

3. *Reward and residual clipping.* Since $\hat{r}$ is computed from a noisy score estimate, we optionally clip $\hat{r}$ or the residual magnitude in (37) to control heavy tails; this is analytically compatible with bounded-moment assumptions by replacing them with explicit bounds.

4. *Weight clipping.* As already encoded in (36), clipping $\bar{w} \le W$ is the primary mechanism by which we prevent variance blow-up in off-policy regression and ensure the weighted moment equations remain well-conditioned.

In summary, the algorithm alternates between (i) collecting rollouts under a behavior policy b to populate $\mathcal{D}$, and (ii) performing many replay-based gradient steps on (38) (and either (40) or (41)) using clipped importance weights. The next section formalizes the guarantees of this procedure: the weighted martingale moments are correctly identified off-policy, and under linear/NTK assumptions and bounded weights the replay-based stochastic approximation converges with explicit finite-time error bounds.

6 Main theoretical guarantees

We collect in this section the formal statements underlying the replay-based procedure of Section 5. Our results address five points: (i) the off-policy correctness of the weighted martingale moment equations induced by the discretization, (ii) almost sure convergence of the replay-based stochastic approximation under linear/NTK function classes and bounded importance weights, (iii) finite-time rates with an explicit decomposition into estimation/approximation/bias/discretization terms, (iv) propagation of score-signal and discretization errors into control performance, and (v) lower bounds showing that (a) coverage/bounded weights are necessary for off-policy learning and (b) noisy terminal feedback enforces an $\Omega(1/\varepsilon^2)$ rollout complexity via a bandit reduction.

6.1 Off-policy identification of the martingale moments

The key structural fact is that the optimal entropy-regularized value function $J^\star$ and its associated $q^\star$ satisfy a martingale restriction that is invariant to the policy used to generate the data. In discrete time this yields a one-step residual whose conditional expectation vanishes at the solution. Since our replay objective is precisely a weighted squared residual, this invariance is what legitimizes replay under an arbitrary behavior policy b.

Theorem 6.1 (Off-policy identification of the martingale equations)**.** *Let $\delta_k(J, q)$ denote the one-step residual at step k constructed from the discretized controlled reverse dynamics and the observed running reward sample $\hat{r}_k$ (cf. (37) with $\Theta^- = \Theta$ for notational simplicity). Suppose $\hat{r}_k = r_k + \epsilon_k$ where $\mathbb{E}[\epsilon_k \mid \mathcal{F}_{t_k}] = \mathrm{bias}_k$ and $\mathbb{E}[\epsilon_k^2 \mid \mathcal{F}_{t_k}] < \infty$. Then for the optimal pair $(J^\star, q^\star)$ we have, for every k,*

$$\mathbb{E}[\delta_k(J^\star, q^\star) \mid \mathcal{F}_{t_k}] \;=\; \mathrm{bias}_k \, \Delta t, \tag{45}$$

regardless of the policy generating (y_k, a_k, y_{k+1})*. In particular, for any square-integrable test function* ϕ *measurable with respect to* (t_k, y_k, a_k)*,*

$$\mathbb{E}_b\big[\phi(t_k, y_k, a_k)\,\big(\delta_k(J^\star, q^\star) - \mathrm{bias}_k \Delta t\big)\big] \;=\; 0. \tag{46}$$

When the score-signal is conditionally unbiased ($\mathrm{bias}_k = 0$*), the residual has zero conditional mean and the corresponding weighted moments are exactly identified off-policy.*

Proof sketch. We apply the martingale characterization of the entropy-regularized control problem to the discretely sampled filtration $\{\mathcal{F}_{t_k}\}_{k=0}^K$. The defining property of $(J^\star, q^\star)$ is that the drift component of $J^\star(t, y_t)$ cancels, leaving only a martingale increment plus the running reward and entropy terms; discretization yields (45). The policy generating the data affects the law of (y_k, a_k) but not the conditional identity itself, hence (46). The observation noise in $\hat{r}_k$ contributes only through its conditional mean (bias) term.

6.2 Almost sure convergence under linear/NTK approximation and bounded weights

The replay objectives in Section 5 amount to solving a system of weighted orthogonality conditions induced by (46). Under linear features (or in the NTK regime where the dynamics of parameters are well-approximated by linearization), the critic regression is a weighted least-squares problem and the coupled actor–critic updates can be treated as stochastic approximation with martingale-difference noise.

Theorem 6.2 (Almost sure convergence with replay and clipped importance weights)**.** *Assume (a) linear (or NTK-linearized) parameterizations* $J_\Theta(t, y) = \langle \Theta, \Psi(t, y)\rangle$ *and either* $q_\psi(t, y, a) = \langle \psi, \Phi(t, y, a)\rangle$ *or a Gaussian policy-induced* $q_\psi = \theta \log \pi_\psi$ *with linear mean* μ_ψ*, (b) bounded feature norms and bounded second moments of* $\hat{r}_k$*, and (c) clipped importance weights* $\bar{w}_\psi \le W$ *as in* (36)*. Suppose moreover that the weighted replay covariance (the projected moment matrix) is positive definite on the feature span. Then, under Robbins–Monro step sizes and standard two-timescale conditions (critic faster than actor), the replay-based stochastic approximation converges almost surely to the unique minimizer of the projected weighted residual risk,*

$$(\Theta^\dagger, \psi^\dagger) \;=\; \arg\min_{\Theta,\psi}\; \mathbb{E}_b\big[\bar{w}_\psi(t_k, y_k, a_k)\,\delta_k(\Theta, \psi)^2\big], \tag{47}$$

where $\delta_k(\Theta, \psi)$ *is the residual induced by* (J_Θ, q_ψ) *(with the appropriate terminal substitution).*

Discussion. Theorem 6.2 is a convergence-to-projection statement: we converge to the best solution representable within the chosen function class under the replay distribution. The role of replay is algorithmic rather than statistical: by increasing the number of stochastic approximation steps per environment transition, we approach (47) with fewer rollouts, provided that the replay distribution remains well-conditioned and weights remain bounded.

6.3 Finite-time bounds and explicit error decomposition

We next state a representative finite-time guarantee illustrating how importance-weight control and replay updates translate into quantitative accuracy. The bound separates the effects of (i) optimization/estimation from a finite number of replay updates, (ii) approximation error due to restricted function classes, (iii) score-signal bias and noise, and (iv) discretization error due to $\Delta t > 0$.

Theorem 6.3 (Finite-time parameter error and induced performance gap)**.** *Under the assumptions of Theorem 6.2, assume additionally that the reward-signal noise is conditionally sub-Gaussian and that the weighted least-squares objective for the critic is strongly convex in Θ over the feature span. Run M replay gradient updates with constant step size η and Polyak averaging to obtain $(\bar{\Theta}_M, \bar{\psi}_M)$. Then there exist problem-dependent constants such that*

$$\mathbb{E}\|\bar{\Theta}_M - \Theta^\dagger\|^2 + \mathbb{E}\|\bar{\psi}_M - \psi^\dagger\|^2 \;\leq\; \tilde{O}\left(\frac{W^2}{M}\right) + \text{Approx} + \text{ScoreBias} + \text{Disc}, \tag{48}$$

where Approx *is the misspecification error of the function classes,* ScoreBias *scales with $\sum_k \mathbb{E}[\text{bias}_k^2]\Delta t^2$ (and similarly for higher-order bias terms), and* $\text{Disc} = O(\Delta t)$ *(or $O(\Delta t^2)$ under higher-order integrators) captures discretization. Moreover, if the Gaussian policy mean map is Lipschitz in ψ and the controlled reverse dynamics are stable under perturbations of the drift control, then the value gap obeys*

$$J^\star(0,\nu) - J^{\pi_{\bar{\psi}_M}}(0,\nu) \;\leq\; C\sqrt{\mathbb{E}\|\bar{\psi}_M - \psi^\dagger\|^2} + \text{Approx} + \text{ScoreBias} + \text{Disc}. \tag{49}$$

Interpretation. The factor W^2 reflects the variance inflation intrinsic to off-policy correction: even after clipping, the effective noise scales with the largest permitted weights. Theorem 6.3 therefore makes explicit the central trade-off: more aggressive off-policy extrapolation (larger W) can reduce bias but increases variance, while stronger clipping reduces variance but changes the target of (47).

6.4 Propagation of discretization and score-signal errors

The preceding theorems separate *optimization* (convergence to $\Theta^\dagger, \psi^\dagger$) from *modeling* (how far $\Theta^\dagger, \psi^\dagger$ are from the true continuous-time optimum). Two error sources are specific to our diffusion setting. First, discretization changes the control problem: even with perfect scores and unlimited function classes, a fixed Δt induces a gap between the continuous-time objective and its discretized counterpart. Second, the running reward uses $\widehat{\nabla} \log p_t$, which is computed from finite data and may be biased; this bias enters the learning problem as a systematic perturbation of the martingale moment condition (45). In both cases the effect is additive in the final guarantees: the learned policy is optimal for the *perturbed* problem (projected and biased) and the resulting performance gap is controlled by stability properties of the controlled reverse dynamics together with Lipschitz properties of the reward functional in the score control.

6.5 Lower bounds: necessity of bounded weights and rollout limits under noisy terminal feedback

Finally, we record two complementary impossibility statements that delimit what replay can and cannot achieve.

Theorem 6.4 (Necessity of coverage / bounded importance weights)**.** *If there exists a measurable set of states S with positive probability under the target policy such that, for some $(t, y) \in S$, the behavior density satisfies $b(a \mid t, y) = 0$ on a subset of the support of $\pi_\psi(\cdot \mid t, y)$, then no algorithm using only replay data collected under b can consistently estimate the target-policy update direction at those states. Moreover, if $\mathbb{E}_b[w_\psi(t, y, a)^2] = \infty$, then importance-weighted estimators of the moments in* (46) *can have infinite variance.*

Theorem 6.5 (Rollout lower bound via a bandit reduction)**.** *Consider a special case in which actions affect only the terminal distribution (equivalently, the dynamics are action-independent for $k < K - 1$) and terminal reward samples have variance σ^2. Any algorithm that outputs a policy with expected terminal reward within ε of optimal with probability at least* $2/3$ *requires $\Omega(\sigma^2/\varepsilon^2)$ rollouts.*

Consequence. Theorem 6.5 formalizes that replay cannot beat the statistical cost of acquiring reward information from the environment when terminal feedback is noisy; rather, replay improves sample efficiency by reusing each rollout for many gradient updates. Theorem 6.4 shows that this benefit is contingent on maintaining coverage (or clipping) so that off-policy correction remains well-posed. Together, these results justify the algorithmic emphasis on bounded weights, trust regions, and replay: they are not merely stabilizers but necessary ingredients for provable learning in reward-directed diffusion control.

6.6 Complexity and optimality: rollouts versus replay

We quantify the algorithmic cost of the replay-based updates and clarify in which regimes replay yields provable gains (and in which regimes it cannot). Throughout, we view a *rollout* as one simulated reverse trajectory of length $K = T/\Delta t$, producing K transitions for the replay buffer, and we view a *gradient update* as one stochastic-gradient step on the replay objective using a minibatch of size $|B|$ drawn from $\mathcal{D}$.

Time per rollout. A single rollout requires simulating the discretized controlled reverse dynamics and evaluating the score-signal used to form $\hat{r}_k$. Writing C_{env} for the cost of one environment step (sampling the next state y_{k+1} given (t_k, y_k, a_k)) and C_{score} for the cost of computing $\widehat{\nabla} \log p_{T-t_k}(y_k)$ and forming $\hat{r}_k$, the time per rollout scales as

$$\text{Time}_{\text{rollout}} \;=\; O(K\,(C_{\text{env}} + C_{\text{score}}))\,. \tag{50}$$

In diffusion applications, C_{env} is typically $O(d)$ for simple Euler–Maruyama dynamics but may be larger if the environment includes additional learned components; C_{score} is dominated either by a score network forward pass or by the construction of a dataset-based score-value estimator. Importantly, both costs are paid *once* per transition regardless of how many replay updates reuse that transition.

Time per gradient update. Each replay update computes (i) the martingale residual $\delta_k(\Theta, \psi)$, (ii) the importance weight $w_k = \pi_\psi(a_k \mid t_k, y_k)/b(a_k \mid t_k, y_k)$ (clipped if needed), and (iii) gradients for the actor/critic (and the normalization penalty if q is not explicitly normalized). If C_{actor} and C_{critic} denote the per-sample forward/backward costs of the actor and critic, and C_w the cost of evaluating log-densities for π_ψ and b, then one update costs

$$\text{Time}_{\text{update}} \;=\; O(|B|\,(C_{\text{actor}} + C_{\text{critic}} + C_w))\,. \tag{51}$$

For the Gaussian policy class $\pi_\psi(\cdot \mid t, y) = \mathcal{N}(\mu_\psi(t, y), \Sigma(t))$ with known $\Sigma(t)$, we have $C_w = O(d)$ from quadratic forms in $\mathbb{R}^d$. Moreover, in the canonical parameterization $q_\psi = \theta \log \pi_\psi$ the normalization constraint $\int \exp(q_\psi/\theta)\, da = 1$ holds identically, so the q-normalization penalty is zero and contributes no additional computation.

Total time and the rollout–replay trade-off. With N rollouts and G gradient updates per rollout, the total time is

$$\text{Time}_{\text{total}} \;=\; O(NK\,(C_{\text{env}} + C_{\text{score}})) \;+\; O(NG\,|B|\,(C_{\text{actor}} + C_{\text{critic}} + C_w))\,. \tag{52}$$

This decomposition makes the intended regime explicit: replay is beneficial when environment interaction is the dominant cost (large $C_{\text{env}} + C_{\text{score}}$) and additional computation is comparatively cheap, so that increasing G can reduce the required N for a target accuracy. Conversely, if gradient updates are as expensive as (or more expensive than) collecting additional transitions, replay does not provide a computational advantage.

Space complexity. The replay buffer stores $|\mathcal{D}| = NK$ transitions, each containing $(t_k, y_k, a_k, \hat{r}_k, y_{k+1})$ and optionally metadata (e.g. $\log b(a_k \mid t_k, y_k)$). Thus,

$$\text{Space}_{\text{replay}} \;=\; O(NK\,(d + d_a + 1))\,, \tag{53}$$

where d_a is the action dimension (in our diffusion control interpretation, $d_a = d$). This linear scaling in NK is standard for off-policy methods; in practice it motivates either fixed-capacity buffers or prioritized retention, but our theoretical statements only require that replay sampling induces a well-conditioned weighted covariance on the feature span.

When replay yields provable improvements. Theorems 6.2–6.3 quantify the effect of replay updates through the number of stochastic approximation steps M (with $M \approx NG$ when counting one minibatch update as one step). In the idealized streaming setting where replay samples are effectively draws from a stationary distribution induced by b, Theorem 6.3 yields an optimization/estimation term of order $\tilde{O}(W^2/M)$. Thus, for a fixed behavior distribution and fixed function class, increasing G at fixed N drives the iterates closer to the projected solution (47) without requiring additional environment interaction.

However, the same theorem also clarifies the ceiling: replay cannot remove the terms labeled Approx, ScoreBias, and Disc, and it cannot create reward information that is not present in the collected transitions. In particular, even in the realizable setting, there remains a *data* (as opposed to *optimization*) limitation due to the finite set of collected rollouts. A useful way to express this is to separate

$$\text{total error} \;\approx\; \underbrace{\text{optimization error}}_{\tilde{O}(W^2/M)} + \underbrace{\text{statistical error from finite } |\mathcal{D}|}_{\text{typically decreases with } NK} + \text{Approx} + \text{ScoreBias} + \text{Disc}.$$

While we do not specialize a single closed form for the statistical term (as it depends on mixing along trajectories and feature concentration), the qualitative implication is standard: once M is large enough that the optimization error is below the statistical floor set by $|\mathcal{D}| = NK$, additional replay updates cannot improve population performance and may only refine the empirical solution on the buffer.

Optimal allocation under a fixed budget. Suppose we have a wall-clock budget that allows either collecting additional rollouts or increasing replay. Then (52) suggests allocating updates until the marginal reduction in $\tilde{O}(W^2/M)$ is comparable to the marginal reduction attainable by increasing NK (which reduces statistical error and, when $\hat{r}_k$ depends on finite-data score estimation, can also reduce ScoreBias if the score estimator is improved with more data). In particular, when the environment is expensive, it can be preferable to take $G \gg 1$ to approach the projected fixed point for the current replay distribution, whereas when the environment is cheap, taking $G \approx 1$ (nearly on-policy) may be adequate.

Limits: lower bounds and the role of W. Two constraints delimit the maximal benefit of replay. First, Theorem 6.5 implies that when the reward information enters only through noisy terminal observations, any method must incur $\Omega(\sigma^2/\varepsilon^2)$ rollouts to obtain ε-accurate reward optimization; replay can only improve computation per rollout, not overcome this information-theoretic barrier. Second, Theorem 6.4 implies that replay is only well-posed under coverage: either b must dominate π_ψ or we must enforce clipping $w \leq W$. From the finite-time bound (48), increasing W expands the range of effective off-policy correction but inflates the variance and hence the number of replay updates required. Consequently, in regimes where b is far from π_ψ, the potential gain from replay is offset by the W^2 dependence unless one introduces additional structure (e.g. trust regions that keep π_ψ close to b in KL).

Summary. Replay yields provable improvements when (i) the environment interaction cost dominates gradient computation, (ii) the replay distribution is sufficiently rich to keep the projected moment matrix well-conditioned, and (iii) importance weights are controlled so that variance remains finite. In that regime, increasing replay updates per transition reduces the optimization component of error at essentially no additional rollout cost, up to the statistical and information-theoretic limits imposed by finite data, score-signal bias, discretization, and noisy reward feedback.

7 Experiments

We empirically evaluate the proposed off-policy martingale q-learning procedure in a standard image-generation setting where (i) the uncontrolled reverse diffusion already produces samples close to the data distribution, and (ii) an additional terminal preference signal induces a distribution shift that must be handled with care. Our goals are to (a) quantify the rollout savings enabled by replay at fixed target quality/reward, (b) compare against representative on-policy diffusion policy optimization baselines, and

(c) validate the algorithmic design choices suggested by the theory, namely importance-weight control and (when applicable) q-normalization.

7.1 Environment, policy class, and rewards

We work with an unconditional diffusion model trained on CIFAR-10 at resolution 32×32. The "environment" is the discretized controlled reverse process with horizon T and $K = T/\Delta t$ steps. At each step k we apply an action $a_k \in \mathbb{R}^d$ interpreted as a score control, and the simulator returns the next state y_{k+1} together with a running reward sample $\hat{r}_k$ computed from the available score-value estimator $\widehat{\nabla} \log p_{T-t_k}(y_k)$ according to the quadratic deviation penalty described earlier. Concretely, we use

$$\hat{r}_k \ \approx \ -g^2(T - t_k) \left\| \widehat{\nabla} \log p_{T-t_k}(y_k) - a_k \right\|^2,$$

where the approximation hides the score-signal noise/bias induced by the estimator and minibatch randomness.

Our target policy is Gaussian with time-varying covariance fixed by the entropy temperature, $\pi_\psi(\cdot \mid t, y) = \mathcal{N}(\mu_\psi(t, y), \Sigma(t))$ and $\Sigma(t) = \frac{\theta}{2g^2(T-t)} I$. In implementation we parameterize the mean as a residual correction to the base score estimate,

$$\mu_\psi(t, y) \ = \ \widehat{\nabla} \log p_{T-t}(y) \ + \ u_\psi(t, y),$$

so that $u_\psi \equiv 0$ recovers the base sampler and the learned control acts only through a shift u_ψ. This parameterization makes the "stay close to data" inductive bias explicit and interacts favorably with importance weighting, since early iterates remain close to the behavior policy.

Terminal rewards are obtained from (noisy) evaluations of a black-box preference functional $h(y_T)$, combined with a weight β. We consider two families of h:

1. *CLIP/aesthetic reward.* We score the final image $x = y_T$ by a pretrained image–text model or an aesthetic predictor, e.g. $h(x) = s_{\text{CLIP}}(x, \tau)$ for a fixed text prompt τ, or $h(x) = s_{\text{aes}}(x)$. We treat the observed reward as $h(x) + \xi$ with ξ mean-zero noise to reflect stochastic evaluation and finite-precision effects.

2. *Incompressibility regularization on CIFAR-10.* We use a simple description-length proxy to discourage pathological, highly structured artifacts that can arise under pure preference maximization. Let $\mathrm{DL}(x)$ denote the compressed bit-length of x under a fixed lossless codec (equivalently, an MDL-style proxy). Empirically, CIFAR-10 images occupy a relatively narrow range of DL; extreme deviations correlate with unnatural samples. We therefore define a terminal term of the form

$$h_{\text{IC}}(x) \ = \ -\big(\mathrm{DL}(x) - m_{\text{data}}\big)^2,$$

where m_{data} is the empirical mean of $\mathrm{DL}(x_0)$ over the training set. In combined experiments we take $h(x) = s_{\text{CLIP}}(x, \tau) + \lambda_{\text{IC}} h_{\text{IC}}(x)$, which operationally enforces a weak "stay in-distribution" constraint using only observable quantities.

7.2 Baselines and evaluation protocol

We compare against two baselines designed to isolate the effect of replay and off-policy correction.

On-policy martingale q-learning. This variant uses the same residual objective and the same actor/critic parameterization, but discards replay: each gradient update uses only the most recent rollout(s), and importance weights are identically 1. This isolates the value of reusing past transitions.

DPOK-style on-policy diffusion policy optimization. We implement an on-policy KL-regularized preference optimization method tailored to diffusion sampling, where the policy update is driven primarily by terminal rewards with a KL penalty to a reference sampler (the base diffusion). While implementations differ across the literature, the defining characteristic for our purposes is that data are used once (or a small constant number of times) and the update is effectively on-policy, with no explicit importance weighting across a large replay buffer.

Metrics. We report (i) mean terminal reward (CLIP/aesthetic score, optionally with incompressibility regularization), (ii) classical generative-quality metrics (FID to CIFAR-10 and Inception Score), and (iii) a distributional diagnostic based on the empirical distribution of $\mathrm{DL}(x)$ relative to the training set. To compare sample efficiency, we plot each metric against the number of *rollouts* (environment episodes) rather than wall-clock time, since rollouts are the resource whose reduction is our principal objective.

7.3 Main results: replay reduces rollouts for a fixed reward target

Across prompts and reward types, we observe a consistent separation between off-policy replay and purely on-policy training when environment interaction is the bottleneck. For a fixed terminal-reward target, replay typically attains the target using substantially fewer rollouts than on-policy martingale learning; correspondingly, for a fixed rollout budget, replay attains higher reward. This is the expected qualitative behavior suggested by the error decomposition in Theorem 6.3: with large G updates per rollout, we reduce the optimization term driven by the number of stochastic approximation steps, while holding the rollout count fixed.

Relative to the DPOK-style baseline, the advantage of replay is most pronounced when terminal rewards are noisy or sparse. In such cases, the ability to amortize each expensive terminal evaluation across many gradient steps is decisive, whereas on-policy preference optimization spends most rollouts collecting reward-labeled samples that are used only once. At the same time, we observe the anticipated trade-off: overly aggressive off-policy updates without weight control can lead to instability and degraded perceptual quality, which motivates the ablations below.

In the combined CLIP+DL experiments, the incompressibility regularizer materially improves fidelity at high reward: for comparable CLIP scores, the DL distribution of generated images remains closer to the CIFAR-10 reference and FID degrades less severely. This supports the interpretation of DL as a simple, fully observable constraint proxy that partially mitigates reward hacking. Importantly, this effect is obtained without modifying the running reward; it is entirely attributable to terminal shaping, which is compatible with the algorithmic framework.

7.4 Ablations: replay size, importance clipping, and q-normalization

Replay buffer size. We vary the capacity $|\mathcal{D}|$ while holding the total rollout count fixed. Small buffers (retaining only the most recent trajectories) behave similarly to on-policy training and lose most of the rollout advantage, while sufficiently large buffers improve stability and reward at fixed rollouts. The effect saturates: beyond a moderate capacity, additional storage yields diminishing returns, consistent with the view that once the projected fixed point for the empirical replay distribution is well-approximated, the remaining limitation is statistical (finite $|\mathcal{D}|$) rather than optimization.

Importance-weight clipping. We test clipping thresholds W spanning a wide range. Without clipping, we observe occasional large-weight events leading to high-variance critic updates and brittle actor steps, which in turn can collapse sample quality. Moderate clipping stabilizes learning and improves the reward–quality trade-off. Excessively small W yields conservative updates that under-correct the behavior mismatch and reduce attainable reward. This reproduces the qualitative dependence predicted by the W^2 factor in the finite-time term: allowing larger effective correction requires either more replay updates (to average out the increased variance) or additional trust-region constraints to keep π_ψ near b.

q-normalization penalty. In the canonical Gaussian setting $q_\psi = \theta \log \pi_\psi$, normalization is exact and the penalty is identically zero, so the ablation is vacuous by construction; we include it only to confirm numerically that enabling/disabling the penalty does not change results. In a generalized variant where we represent $q_\psi(t, y, a)$ with a flexible network not constrained to integrate to one, we observe that removing the penalty leads to pathological growth of $\log \int \exp(q_\psi/\theta)\, da$ and degraded policy updates, whereas a modest penalty weight restores stable learning and improves reward at fixed rollouts. This supports the role of normalization as an identifiability/stability constraint rather than a cosmetic regularizer.

Overall, these experiments support the central claim: when rollouts are expensive, replay provides a principled mechanism to exchange additional computation for fewer environment interactions, while importance-weight control and (when needed) q-normalization are necessary to realize this gain without sacrificing stability and sample quality.

8 Discussion and Extensions

We briefly discuss several directions in which the off-policy martingale viewpoint extends beyond the basic stochastic reverse-time sampler considered above. The common thread is that, once we regard diffusion sampling as a controlled dynamical system with observable (possibly biased) reward surrogates, many variations amount to changing either (i) the environment transition map, (ii) the information set used to form $\hat{r}$, or (iii) the objective functional. In each case, the martingale residual framework continues to provide moment equations whose roots define the relevant value functions and policy improvements, subject to the same coverage and bounded-weight requirements.

8.1 ODE and probability-flow samplers

Many practical diffusion samplers use deterministic or partially deterministic dynamics, e.g. probability-flow ODEs or predictor–corrector schemes with a reduced noise schedule. Formally, if we replace the controlled reverse SDE by a controlled ODE

$$\dot{y}_t \;=\; F(t, y_t, a_t),$$

then the discrete-time simulator becomes $y_{k+1} = \mathsf{Env}(t_k, y_k, a_k)$ with no injected stochasticity beyond that induced by the policy. In this setting the martingale terminology is slightly abusive—the residual is no longer a martingale difference arising from Brownian increments—but the same *temporal consistency* equation remains: at the optimum, the (entropy-regularized) value satisfies a one-step relation of the form

$$J(t_{k+1}, y_{k+1}) - J(t_k, y_k) + r(t_k, y_k, a_k)\Delta t - q(t_k, y_k, a_k)\Delta t \;=\; 0$$

along trajectories. Consequently, our weighted squared-residual objective continues to define a projected fixed point under replay sampling. The principal change is variance: for an ODE environment, the conditional variance of y_{k+1} given (t_k, y_k, a_k) can be smaller, which reduces the noise in δ_k and can improve critic learning for a fixed number of rollouts. At the same time, deterministic dynamics can exacerbate coverage issues: if the behavior policy induces a narrow set of visited states, then replay may fail to cover the states reached by a substantially improved target policy, yielding large importance weights or support mismatch. This suggests that ODE samplers benefit disproportionately from explicit exploration in b, mixture behavior policies, or trust-region constraints limiting $\mathrm{KL}(\pi_\psi(\cdot \mid t, y) \,\|\, \pi_{\mathrm{old}}(\cdot \mid t, y))$.

A related practical extension is to non-uniform and adaptive step sizes. If the sampler uses a time grid $\{t_k\}_{k=0}^{K}$ with variable Δt_k, then the residual becomes

$$\delta_k(\Theta, \psi) \;=\; J_\Theta(t_{k+1}, y_{k+1}) - J_\Theta(t_k, y_k) + \hat{r}_k \Delta t_k - q_\psi(t_k, y_k, a_k)\Delta t_k,$$

and the replay objective is unchanged aside from using Δt_k. This is conceptually useful because many high-order solvers concentrate steps in regions where $g(T-t)$ is large; our formulation accommodates this without modifying the learning rule, provided the transition tuples record Δt_k (or equivalently t_k, t_{k+1}).

8.2 Conditional diffusion and guided generation

In conditional generation we introduce an observed condition c (class label, text embedding, or other side information) and seek a policy $\pi_\psi(\cdot \mid t, y, c)$ whose induced terminal samples satisfy both reward preference and conditional fidelity. From the present perspective, c is part of the state; we may simply augment $y \leftarrow (y, c)$ with trivial dynamics for c. The running reward and terminal reward become $r(t, y, c, a)$ and $h(y_T, c)$, and all moment equations continue to hold conditional on c. In particular, if the base score estimator is conditional, $\widehat{\nabla} \log p_t(y \mid c)$, we may use the same residual mean parameterization

$$\mu_\psi(t, y, c) \;=\; \widehat{\nabla} \log p_{T-t}(y \mid c) + u_\psi(t, y, c),$$

thereby preserving the inductive bias that $u_\psi \equiv 0$ recovers the base conditional sampler.

Classifier-free guidance and related techniques can also be seen as specifying a restricted policy family: for example, guidance with scale s corresponds to actions of the form

$$a(t, y, c) \;=\; \widehat{\nabla} \log p_{T-t}(y) + s\Big(\widehat{\nabla} \log p_{T-t}(y \mid c) - \widehat{\nabla} \log p_{T-t}(y)\Big),$$

which is a one-dimensional control subspace. Our algorithm may be applied directly with $\psi = s$ (or a time-dependent scale $s(t)$), in which case replay learns an adaptive guidance schedule optimized for the terminal preference signal, while importance weighting controls the distribution shift relative to the behavior guidance. More generally, conditional diffusion introduces an additional axis along which coverage can fail: if the behavior policy is collected under a narrow set of conditions c, off-policy learning cannot generalize to unseen c without explicit function approximation assumptions and training data spanning those conditions.

8.3 Retrieval-augmented score signals

Our formulation treats $\widehat{\nabla} \log p_t(y)$ as a data-driven score-value signal that may be biased and noisy, and whose noise propagates to $\hat{r}_k$. A natural extension is to enhance $\widehat{\nabla} \log p_t(y)$ using retrieval or nearest-neighbor conditioning on the training set. Concretely, for a query (t, y) we retrieve a set of data points $\mathcal{N}(y) = \{x_0^{(j)}\}$ (possibly in a learned embedding space), and construct an estimator

$$\widehat{\nabla} \log p_t(y) \;=\; \mathsf{Agg}\Big(\big\{\nabla_y \log p_{t|0}(y \mid x_0^{(j)})\big\}_{x_0^{(j)} \in \mathcal{N}(y)}\Big),$$

where $p_{t|0}$ is known and Agg is an averaging or attention operator. In OU/VP settings, each $\nabla_y \log p_{t|0}(y \mid x_0)$ is explicit, hence retrieval provides a nonparametric mechanism to approximate the score by localizing around training examples that plausibly explain y at time t.

This augmentation affects learning only through the reward signal $\hat{r}_k$ (and through any parameterization using $\widehat{\nabla} \log p$ as a baseline). The theoretical consequences are therefore captured by the "score-signal bias" terms in our bounds: retrieval may reduce variance (by averaging multiple neighbors) and reduce bias (by improving local fit), but it can also introduce selection bias if the retrieval distribution depends sharply on y in a way not modeled by the estimator class. From an off-policy perspective, retrieval can be implemented without changing the replay interface; however, it changes the stationarity of $\hat{r}$ if the retrieval index is updated online. In that case one should either (i) store enough metadata in replay to recompute $\hat{r}_k$ under the current retrieval model, or (ii) treat the reward non-stationarity as additional noise and rely on small step sizes and frequent replay refresh. The former is preferable when $\hat{r}$ is used as a regression target for the critic.

8.4 Constrained and safe objectives

Preference maximization in generative models is often accompanied by constraints: fidelity constraints (stay close to data), safety constraints (avoid disallowed content), or resource constraints (e.g. inference-time budgets). Our framework already enforces a soft "stay close" effect through the running quadratic penalty and the entropy term, but one may wish to impose explicit constraints of the form

$$\mathbb{E}\big[c(y_T)\big] \;\le\; \kappa \qquad \text{or} \qquad \mathbb{E}\Big[\int_0^T \ell(t, y_t, a_t)\, dt\Big] \;\le\; \kappa,$$

where c is an observable terminal cost and ℓ an observable running cost. A standard approach is Lagrangian relaxation: introduce a multiplier $\lambda \ge 0$ and optimize

$$\mathbb{E}\Big[\int_0^T \big(r(t, y_t, a_t) - \lambda \ell(t, y_t, a_t) - \theta \log \pi(a_t \mid t, y_t)\big)\, dt + \beta h(y_T) - \lambda c(y_T)\Big] + \lambda\kappa.$$

For fixed λ, this is again an entropy-regularized control problem with modified rewards; hence the same residual equations apply with $\hat{r}_k \leftarrow \hat{r}_k - \lambda \hat{\ell}_k$ and terminal reward $\beta h(y_T) - \lambda c(y_T)$. One may then update λ by stochastic dual ascent using samples from rollouts, reusing replay to reduce variance. Off-policy correction remains necessary because the state–action visitation induced by the constrained optimum can differ materially from that of the behavior sampler.

Beyond expectations, risk-sensitive criteria (e.g. CVaR) can be incorporated by augmenting the state with an auxiliary variable and defining a Markovian objective in an expanded space; likewise, hard constraints can be approximated via barrier penalties in h or via rejection sampling at terminal time. The practical lesson is that constrained objectives intensify the need for weight control: as constraints activate, optimal policies can become sharply peaked, which increases the mismatch with exploratory behavior policies and can inflate importance weights. Consequently, conservative updates (clipping, trust regions, or explicit mixtures $b = \alpha \pi_\psi + (1 - \alpha)\pi_{\text{base}}$) are not merely stabilizers but plausibly necessary for consistent learning in constrained regimes.

Taken together, these extensions suggest that the off-policy martingale residual is best viewed as a modular interface between (i) a controlled sampler (SDE/ODE, conditional/unconditional) and (ii) an objective specified via observable running and terminal signals (possibly augmented by retrieval and constraints). The main limitations remain those already highlighted by the theory: coverage, bounded effective importance weights, and control of bias in the score-derived reward surrogates.