IQ-DIFF: ITERATIVE Q-GUIDED SAMPLING-TIME DIFFUSION ALIGNMENT WITH BLACK-BOX PHYSICS REWARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models provide generative priors for physical prediction, but statistically plausible outputs need not satisfy downstream physical objectives. We study sampling-time alignment when feedback is available as a scalar terminal reward from a black-box physics simulator. IQ-Diff keeps the diffusion backbone frozen and repeatedly learns action-value functions from trajectories of the current guided sampler. Re-evaluation accounts for changes in the continuation policy, while accumulated Q-scores implement successive KL-regularized updates over local denoising candidates. Deployment uses these learned scores without simulator gradients or further simulator calls. We analyze the finite-candidate selection process directly, establishing a final-iterate bound under controlled Q-estimation error and separating this optimization error from the proposal's coverage limitation. Exact current-policy evaluation yields monotonic reward improvement and an $O(1/K)$ final-iterate guarantee relative to the best selector with the same candidate budget, where K is the number of outer iterations. Experiments on fluid dynamics and thermal-process prediction demonstrate improved prediction accuracy and favorable quality–cost tradeoffs against unguided sampling, final-sample search, physics-reference guidance, single-Q guidance, and SMC-style resampling.

1 INTRODUCTION

Denoising Diffusion Models (DDMs) have emerged as a powerful framework for generative modeling, producing high-fidelity samples in image synthesis (Ho et al., 2020; Dhariwal & Nichol, 2021), audio generation (Huang et al., 2023), and robotic policy learning (Chi et al., 2025; Ren et al., 2025); see Yang et al. (2023) for a broader review. Rather than generating an output in a single feed-forward pass, diffusion models learn to reverse a gradual noising process (Song & Ermon, 2019; Ho et al., 2020). Each intermediate denoising step therefore provides an opportunity to guide generation toward an external objective while keeping the pretrained backbone fixed.

This opportunity is particularly valuable in scientific and engineering prediction, where statistical plausibility alone is not sufficient. A generated physical field may resemble observed data while violating physical constraints, deviating from a trusted simulator, or failing to meet a downstream engineering objective (Karniadakis et al., 2021; Wang et al., 2023a). The gap between *generative realism* and *physical reliability* motivates incorporating physical information into sampling. When high-fidelity observations are scarce or expensive to obtain, lower-fidelity simulators can provide complementary information for improving predictions (Shi et al., 2025). The goal is to make use of this information without repeatedly retraining the diffusion backbone.

The difficulty is that physical feedback is not always available in a form that can directly guide denoising. Explicit governing equations may support differentiable residuals and physics-informed objectives (Raissi et al., 2017; Cuomo et al., 2022). A simulator-based evaluation, however, may expose only a black-box interface: a completed prediction can be scored, but derivatives of that evaluation are unavailable to the alignment algorithm. We study this setting, where feedback arrives as a **scalar terminal reward**. Such feedback can distinguish better complete predictions, but does not directly identify which intermediate choices produced them. To guide sampling, it must first be translated into estimates of the consequences of earlier denoising decisions.

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

1.1 Background and Contributions

Existing alignment methods provide two broad ways to incorporate task-specific objectives. Training-time approaches learn or update diffusion policies using offline reinforcement learning (Wang et al., 2023b) or reinforcement-learning-based fine-tuning (Ren et al., 2025). Sampling-time approaches instead use the denoising trajectory to steer a fixed backbone. Classifier guidance uses an auxiliary classifier's gradients (Dhariwal & Nichol, 2021), while energy guidance and diffusion posterior sampling use differentiable energies, constraints, or measurement models (Bansal et al., 2023; Yu et al., 2023; Chung et al., 2023; Zhang et al., 2025). Classifier-free guidance takes a different route, combining conditional and unconditional score estimates without an auxiliary classifier (Ho & Salimans, 2022).

Reward gradients are not necessary for all forms of sampling-time control. Twisted diffusion sampling, Feynman–Kac correctors, and soft value-based decoding use weighting, resampling, or learned values to steer pretrained samplers (Wu et al., 2023; Skreta et al., 2025; Li et al., 2025); see also Uehara et al. (2025). Building on this perspective, we focus on what happens *after guidance changes the sampling policy*. An early denoising action's expected reward depends on how the remaining steps are performed. Once those later decisions change, values learned under an earlier sampler may no longer accurately describe the current process. This motivates learning from the guided sampler's own trajectories, rather than relying only on a fixed value estimate or increasing its guidance strength.

We introduce **IQ-Diff**, short for Iterative Q-Guided Diffusion, which combines *current-policy re-evaluation* with *accumulated Q-guidance*. IQ-Diff learns to assess a denoising choice by the terminal reward it is expected to produce when the remaining steps follow the current sampler. These learned Q-functions turn feedback on complete predictions into guidance for intermediate choices, without changing the pretrained diffusion backbone.

As illustrated in Fig. 1, IQ-Diff starts from the frozen base sampler and alternates between collecting trajectories and updating guidance. Each round evaluates complete predictions with the simulator and learns a new Q-function from their terminal rewards. Its scores are added to the existing guidance to update how candidates are selected, rather than replacing earlier Q-functions. The updated sampler then supplies the next round of trajectories.

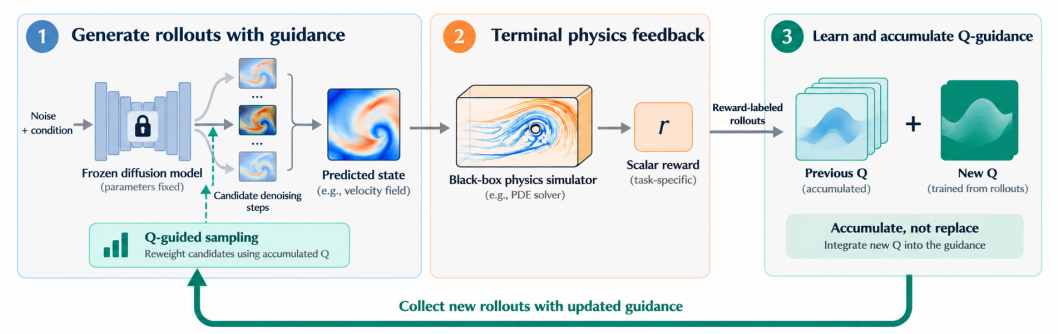

Figure 1: Overview of IQ-Diff. Rollouts from the current guided sampler receive terminal simulator rewards, which train a new Q-function. Its scores are accumulated into guidance for the next round. The diffusion backbone remains frozen throughout; deployment uses learned Q-guidance without simulator calls.

During sampling, IQ-Diff chooses among local candidates from the frozen diffusion transition along a single active trajectory. Learned Q-functions score these choices without modeling simulator dynamics; deployment requires no simulator queries or backbone updates.

Our main contributions are:

- **Iterative alignment through accumulated Q-guidance.** We convert terminal simulator feedback into reusable denoising guidance through repeated current-policy evaluation and Q-score accumulation. This combines updated return estimates with successive local sampling updates, while leaving the backbone frozen and eliminating simulator queries at deployment.

- **A direct theory of finite-candidate policy improvement.** We analyze the sampling rule actually executed and bound the final-iterate gap under controlled Q-estimation error. Exact evaluation gives monotonic reward improvement and an $O(1/K)$ guarantee within the same candidate budget; a separate coverage bound distinguishes improvements in selection from limitations of the frozen proposal.
- **Evaluation of prediction accuracy and deployment cost.** Experiments on fluid dynamics and thermal-process prediction show improved accuracy and favorable quality–cost tradeoffs against inference-time baselines. Component ablations examine iterative learning and Q accumulation, while the cost analysis quantifies the benefit of guiding a single trajectory without simulator calls at deployment.

Algorithm 1 specifies the iterative procedure; Appendix A provides an extended discussion of related work.

2 PRELIMINARIES

2.1 DIFFUSION MODELS

We build our framework upon Denoising Diffusion Probabilistic Models (DDPMs) (Ho et al., 2020). A diffusion model consists of a fixed forward process that gradually adds noise to data and a learned reverse process that generates samples by denoising from noise.

The forward process corrupts a clean sample x_0 over T noise levels according to a variance schedule $\beta_1, \ldots, \beta_T$:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right), \qquad t = 1, \ldots, T. \tag{1}$$

Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{r=1}^{t} \alpha_r$. A key property of the forward process is that x_t can be sampled directly from x_0:

$$q(x_t \mid x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I\right). \tag{2}$$

For sampling, we adopt the generalized Denoising Diffusion Implicit Model (DDIM) framework (Song et al., 2021a), which allows the stochasticity of the reverse process to be controlled. Given a pretrained noise prediction network $\epsilon_\theta(x_t, t, c)$, where c denotes conditioning information such as historical observations or process variables, the DDIM update takes the form

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \underbrace{\left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t, t, c)}{\sqrt{\bar{\alpha}_t}}\right)}_{\text{predicted } \hat{x}_0} + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\,\epsilon_\theta(x_t, t, c) + \sigma_t z,$$

where $z \sim \mathcal{N}(0, I)$. The parameter σ_t controls the stochasticity of the sampler:

$$\sigma_t = \eta\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}}\sqrt{1-\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}}, \tag{3}$$

with $\eta \in [0, 1]$. When $\eta = 0$, the sampler is deterministic; when $\eta = 1$, it recovers the stochastic DDPM sampling behavior.

In practice, inference often uses a subsampled reverse chain with horizon H, which we write as $x_H \to x_{H-1} \to \cdots \to x_0$. Here $h = H, \ldots, 1$ indexes the actual denoising steps used during sampling. The pretrained conditional diffusion model defines a frozen base transition distribution $\pi_0(x_{h-1} \mid x_h, h, c)$, which is induced by the chosen stochastic DDIM/DDPM sampling rule. For a fixed current state x_h, one forward pass through the frozen diffusion model specifies the transition distribution from which candidate next states x_{h-1} can be sampled.

2.2 DENOISING AS A TERMINAL-REWARD DECISION PROCESS

We formulate reverse diffusion sampling as a finite-horizon decision process. At denoising step h, the state is $s_h = (x_h, h, c)$, which contains the current noisy sample, the denoising index, and the conditioning information. The action is the next denoised sample $a_h = x_{h-1}$. The frozen diffusion sampler $\pi_0(a_h \mid s_h)$ serves as the base policy. The external physics simulator is treated as a terminal

reward oracle: after a full sample x_0 is produced, it returns a scalar reward $R(x_0; c)$ measuring physical consistency or downstream task quality. $R(x_0; c)$ is sparse and terminal, observed only at the final step $h = 1$ when $a_1 = x_0$, and is zero otherwise.

For a sampling policy π, the terminal-reward objective is

$$J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[R(x_0; c)\right], \tag{4}$$

where $\tau = (s_H, a_H, \ldots, s_1, a_1, x_0)$ denotes the full denoising trajectory. Under this formulation, sampling-time alignment amounts to improving the reverse diffusion policy with respect to the terminal simulator reward, while keeping the pretrained diffusion backbone fixed.

3 METHODOLOGY

We now present IQ-Diff, which learns action-value functions from denoising trajectories evaluated by the physics simulator, and uses them to steer the frozen diffusion sampler at inference time.

3.1 TRUST-REGION POLICY IMPROVEMENT FOR DENOISING

For a sampling policy π, define the action-value and value functions at denoising step h as

$$Q_h^\pi(s_h, a_h) = \mathbb{E}_\pi\left[R(x_0; c) \mid s_h, a_h\right], \qquad V_h^\pi(s_h) = \mathbb{E}_{a_h\sim\pi(\cdot|s_h)}\left[Q_h^\pi(s_h, a_h)\right]. \tag{5}$$

The corresponding advantage function is $A_h^\pi(s_h, a_h) = Q_h^\pi(s_h, a_h) - V_h^\pi(s_h)$.

For the current sampler π_k, the performance difference identity expresses the effect of a policy update:

$$J(\pi) - J(\pi_k) = \sum_{h=1}^{H} \mathbb{E}_{s_h\sim d_h^\pi,\, a_h\sim\pi(\cdot|s_h)}\left[A_h^{\pi_k}(s_h, a_h)\right], \tag{6}$$

where d_h^π is the state distribution at step h. To avoid optimizing this policy-dependent distribution directly, we follow the trust-region principle (Schulman et al., 2015) and fix the surrogate state distribution at $d_h^{\pi_k}$. Below, $\mathbb{E}_{d_h^{\pi_k},\pi}$ samples a state from $d_h^{\pi_k}$ and an action from $\pi(\cdot \mid s_h)$, while $\mathbb{E}_{d_h^{\pi_k}}$ averages over states alone:

$$\max_\pi \sum_{h=1}^{H} \mathbb{E}_{d_h^{\pi_k},\pi}[A_h^{\pi_k}(s_h, a_h)] \quad \text{s.t.} \quad \sum_{h=1}^{H} \mathbb{E}_{d_h^{\pi_k}}[D_{\mathrm{KL}}(\pi(\cdot \mid s_h) \,\|\, \pi_k(\cdot \mid s_h))] \le \epsilon. \tag{7}$$

This calculation motivates score accumulation; Section 4 instead analyzes the executed finite-candidate rule, not an exact continuous tilt.

For an unrestricted policy class, the penalized surrogate separates state-wise at states of positive sampling weight. We use the resulting update at every state. Since $V_h^{\pi_k}(s_h)$ is action-independent, maximizing expected advantage is equivalent to maximizing expected Q-value:

$$\pi_{k+1}(\cdot \mid s_h) = \arg\max_{\pi(\cdot|s_h)} \left\{ \mathbb{E}_{a_h\sim\pi(\cdot|s_h)}\left[Q_h^{\pi_k}(s_h, a_h)\right] - \frac{1}{\rho_k} D_{\mathrm{KL}}\left(\pi(\cdot \mid s_h) \,\|\, \pi_k(\cdot \mid s_h)\right) \right\}, \tag{8}$$

where $\rho_k > 0$ controls the update strength. The solution has the exponential-tilted form

$$\pi_{k+1}(a_h \mid s_h) = \frac{\pi_k(a_h \mid s_h) \exp\left\{\rho_k Q_h^{\pi_k}(s_h, a_h)\right\}}{Z_k(s_h)}, \tag{9}$$

with normalizing constant $Z_k(s_h) = \int \pi_k(a' \mid s_h) \exp\left\{\rho_k Q_h^{\pi_k}(s_h, a')\right\} da'$.

Recursively expanding (9) from the frozen base sampler π_0 gives the idealized Q-guided policy

$$\pi_K(a_h \mid s_h) \propto \pi_0(a_h \mid s_h) \exp\left\{ \sum_{i=0}^{K-1} \rho_i Q_h^{\pi_i}(s_h, a_h) \right\}. \tag{10}$$

This motivates accumulating current-policy action preferences. In the implemented algorithm, the rollouts come from the finite-candidate sampler $\widehat{\pi}_k$, and each Q_{ϕ_k} estimates $Q_h^{\widehat{\pi}_k}$, not the value of an unexecuted continuous policy. The candidate selection probabilities satisfy their own exact exponential-update identity, established in Section 4.

3.2 Iterative Q-Function Learning

Let $\widehat{\pi}_0 = \pi_0$ be the unguided frozen sampler. At iteration k, the current sampler $\widehat{\pi}_k$ uses the previously learned Q-functions $\{Q_{\phi_i}\}_{i=0}^{k-1}$. IQ-Diff collects M denoising trajectories (rollouts) from $\widehat{\pi}_k$, evaluates each final sample $x_0^{(m)}$ with the simulator, and trains a new action-value function Q_{ϕ_k}, where ϕ_k denotes the learnable parameters of Q_{ϕ_k}.

For trajectory m, the generated state-action pairs and terminal reward are

$$\tau_k^{(m)} = \{(s_h^{(m)}, a_h^{(m)})\}_{h=H}^{1}, \qquad r^{(m)} = R(x_0^{(m)}; c^{(m)}).$$

Because the objective is an undiscounted terminal reward, $\mathbb{E}[r \mid s_h, a_h; \widehat{\pi}_k] = Q_h^{\widehat{\pi}_k}(s_h, a_h)$ on the rollout distribution. The same terminal return $r^{(m)}$ is therefore a valid Monte Carlo target at every visited step. This gives the regression dataset $\mathcal{D}_k = \{(s_h^{(m)}, a_h^{(m)}, r^{(m)}) : m = 1, \dots, M,\ h = H, \dots, 1\}$. We fit Q_{ϕ_k} by Monte Carlo regression:

$$\mathcal{L}(\phi_k) = \frac{1}{MH} \sum_{m=1}^{M} \sum_{h=1}^{H} \left(Q_{\phi_k}(s_h^{(m)}, a_h^{(m)}) - r^{(m)} \right)^2. \tag{11}$$

The denoising index h and context c are included in the state input, so Q_{ϕ_k} learns horizon-dependent action values. After training, the new function is added to the accumulated guidance, and the next round of trajectories is collected with the updated sampler. The pairs within one trajectory share the same reward and are not independent regression observations; early-step return targets may also have substantial continuation variance.

3.3 Finite-Candidate Q-Guided Resampling

The continuous tilt in (10) cannot generally be sampled exactly in the high-dimensional action space. IQ-Diff instead implements local finite-candidate selection. The analysis treats this implemented process as a decision problem in its own right. At denoising step h, given $s_h = (x_h, h, c)$, we draw N candidate actions from the frozen base transition:

$$a_h^{(j)} \sim \pi_0(\cdot \mid s_h), \qquad j = 1, \dots, N. \tag{12}$$

Given the k learned Q-functions, assign an accumulated score to each candidate and compute its selection probability:

$$\begin{aligned} \ell_h^{(j)} &= \sum_{i=0}^{k-1} \rho_i Q_{\phi_i}(s_h, a_h^{(j)}), \qquad m_h = \max_{1 \le r \le N} \ell_h^{(r)}, \\ p_h^{(j)} &= \frac{\exp\{\ell_h^{(j)} - m_h\}}{\sum_{r=1}^{N} \exp\{\ell_h^{(r)} - m_h\}}. \end{aligned} \tag{13}$$

Subtracting m_h gives a numerically stable softmax without changing the distribution; no logit clipping is used, consistently with Table 6. IQ-Diff samples $J \sim \mathrm{Cat}(p_h^{(1)}, \dots, p_h^{(N)})$, sets $a_h = a_h^{(J)}$, and updates $x_{h-1} = a_h$. Each step remains anchored to the frozen proposal while the accumulated values bias candidate selection. The backbone is evaluated once per denoising step to specify the transition, irrespective of N. Algorithm 1 summarizes the procedure.

4 Theoretical Results

We analyze the executed finite-candidate sampler, connecting stochastic action sets (Boutilier et al., 2018) with policy mirror descent (Xiao, 2022). The Q-functions evaluate the same policy used to collect rollouts. Full proofs are in Appendix B.

Candidate selection. Fix N and let $\mathcal{C}_h = (a_h^{(1)}, \dots, a_h^{(N)})$ be the ordered candidate tuple. Write $F_k(s, a) = \sum_{i=0}^{k-1} \rho_i Q_{\phi_i}(s, a)$, with $F_0 = 0$, and denote the probabilities in (13) by $p_{k,h}(j \mid s_h, \mathcal{C}_h)$.

Algorithm 1 IQ-Diff

Require: Frozen sampler π_0, reward R, iterations K, trajectories per iteration M, candidates N.
1: Initialize $\mathcal{Q} \leftarrow \emptyset$.
2: **for** $k = 0, \dots, K-1$ **do**
3: Collect M trajectories using current sampler $\widehat{\pi}_k$.
4: Evaluate terminal rewards $r^{(m)} = R(x_0^{(m)}; c^{(m)})$.
5: Form $\mathcal{D}_k = \{(s_h^{(m)}, a_h^{(m)}, r^{(m)})\}$.
6: Train Q_{ϕ_k} by (11).
7: Update $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{Q_{\phi_k}\}$ and define $\widehat{\pi}_{k+1}$ by (12)–(13).
8: **end for**
9: **return** $\widehat{\pi}_K$.

Accumulation gives

$$p_{k+1,h}(j \mid s_h, \mathcal{C}_h) = \frac{p_{k,h}(j \mid s_h, \mathcal{C}_h) e^{\rho_k Q_{\phi_k}(s_h, a_h^{(j)})}}{\sum_{r=1}^{N} p_{k,h}(r \mid s_h, \mathcal{C}_h) e^{\rho_k Q_{\phi_k}(s_h, a_h^{(r)})}}. \tag{14}$$

This identity compares selection rules at the same possible tuple, not their marginal action kernels; tuples need not be reused. Let $J_N^\star$ be the best expected reward over measurable Markov selectors with the same proposal, horizon, initial distribution, and N independent candidates per step. Selectors can inspect the current tuple, not future draws, and need not share the Q-network parameterization.

Assumption 1 (Bounded terminal reward)**.** *The measurable terminal reward satisfies* $|R(x_0; c)| \le V_{\max}$. *State and action spaces are standard Borel; proposal kernels and learned scores are measurable.*

Assumption 2 (Fixed proposal and accumulated guidance)**.** *The horizon* H*, candidate count* N*, proposal* π_0*, and initial distribution are fixed. Candidates are conditionally independent draws from* $\pi_0(\cdot \mid s_h)$ *with fresh randomness. Starting from* $p_{0,h} = 1/N$*, which induces* $\widehat{\pi}_0 = \pi_0$*, IQ-Diff follows* (14) *with constant* $\rho_k = \rho > 0$*, finite scores, and no logit clipping.*

Assumption 3 (Relative Q-regression accuracy)**.** *For every* $k = 0, \dots, K-1$*, step* h*, state, and candidate tuple, the errors* $e_{k,h}^{(j)} = Q_{\phi_k}(s_h, a_h^{(j)}) - Q_h^{\widehat{\pi}_k}(s_h, a_h^{(j)})$ *satisfy*

$$\max_{1 \le j \le N} e_{k,h}^{(j)} - \min_{1 \le j \le N} e_{k,h}^{(j)} \le \varepsilon_Q. \tag{15}$$

This condition measures relative-value errors under the current sampler: common score offsets do not affect selection. Absolute error at most $\varepsilon_Q/2$ suffices, but small average training loss does not imply the condition.

Theorem 1 (Finite-candidate improvement with approximate Q-functions)**.** *Under Assumptions 1–3, every update satisfies*

$$J(\widehat{\pi}_{k+1}) - J(\widehat{\pi}_k) \ge -\frac{\rho H \varepsilon_Q^2}{16}, \qquad k = 0, \dots, K-1. \tag{16}$$

After $K \ge 1$ *updates, the returned sampler satisfies*

$$0 \le J_N^\star - J(\widehat{\pi}_K) \le \frac{H \log N}{\rho K} + \frac{2HV_{\max}}{K} + H\varepsilon_Q + \frac{\rho H(H+K)\varepsilon_Q^2}{32}. \tag{17}$$

Why iterative guidance helps. A denoising action is judged by the terminal reward expected when the current sampler completes generation. Re-evaluation keeps this assessment tied to the updated sampler; accumulation shifts probability toward candidates with higher expected returns. With exact values, these local changes cannot reduce expected terminal reward, and the returned sampler satisfies

$$J_N^\star - J(\widehat{\pi}_K) \le \frac{H \log N}{\rho K} + \frac{2HV_{\max}}{K}. \tag{18}$$

The $O(1/K)$ bound justifies improving sampling decisions without retraining the backbone. Inaccurate relative values can instead favor the wrong candidates, and stronger or repeated updates can reinforce those errors. Thus (17) gives finite-K stability, not a guarantee that every additional iteration improves a learned sampler.

Proposal coverage. Let $J^\star$ be the optimum without the candidate restriction and $\Delta_N = J^\star - J_N^\star$. Assume bounded measurable Bellman-optimal functions $V_h^\star, Q_h^\star$ exist, with $J^\star = \mathbb{E}[V_H^\star(s_H)]$.

Proposition 1 (Coverage of high-value candidates)**.** *Under Assumption 1 and independent candidate proposals, suppose some $\varepsilon \geq 0$ and $p_\varepsilon \in (0, 1]$ satisfy, for every state and step,*

$$\pi_0(\{a : Q_h^\star(s_h, a) \geq V_h^\star(s_h) - \varepsilon\} \mid s_h) \geq p_\varepsilon. \tag{19}$$

Then

$$0 \leq \Delta_N \leq H\varepsilon + 2HV_{\max}(1 - p_\varepsilon)^N. \tag{20}$$

Better selection still requires good candidates: $(1 - p_\varepsilon)^N$ bounds the probability that a tuple misses the near-optimal region. The theorem and proposition thus distinguish learning to choose from having useful choices available. Their guarantees concern simulator reward, not a separate high-fidelity evaluation metric.

5 Experiments

We evaluate IQ-Diff on two simulator-guided prediction tasks: buoyancy-driven fluid dynamics and a thermal process. Both use a frozen conditional diffusion predictor, terminal physics-based feedback for alignment, and held-out target fields for evaluation. We report task-specific prediction metrics, qualitative comparisons, online sampling costs, and ablations. Implementation details are provided in Appendix C, with the thermal reward described in Appendix D.3.

5.1 Experimental Setup and Baselines

Task. Given context information c, such as historical observations or process variables, the goal is to generate a future physical field x_0. A conditional diffusion model is first trained as the frozen base sampler π_0. During sampling-time guidance, an external simulator provides a terminal scalar reward $R(x_0; c)$ after a complete generated sample is obtained. IQ-Diff uses these terminal rewards to learn action-value functions, while keeping the diffusion backbone fixed.

Main benchmark: fluid dynamics. We use the 2D buoyancy-driven fluid prediction setup of Shi et al. (2025), governed by the Boussinesq approximation of the Navier–Stokes equations. Ground-truth trajectories are generated by a high-performance CFD simulator on 128×128 grids from randomized initial buoyancy and vorticity. Each simulation runs from $s = 0$ to $s = 10$, using the buoyancy field at $s = 0$ as context c and the field at $s^* = 10$ as the target. The dataset contains 6880 simulations, split 90%/10% for training/testing. During guidance, the terminal reward is computed from a faster 32×32 coarse-grid solver as the negative MSE between the generated prediction and the coarse solver reference, providing the scalar feedback used for guidance. Details are in Appendix C.3.

Baselines. We compare IQ-Diff with the following inference-time methods, all using the same frozen diffusion model and sampling horizon: (i) **DDIM**, the unguided base sampler (Song et al., 2021a); (ii) **BoN-16**, which generates 16 complete samples and selects the one with the highest terminal reward; (iii) **SGG**, energy-based guidance using simulator outputs as fixed conditioning references (Shi et al., 2025); (iv) **TDS-Reward**, SMC-style particle resampling with reward-induced twisting potentials from predicted clean samples (Wu et al., 2023); and (v) **Single-Q**, IQ-Diff with $K = 1$ using a single learned action-value function. Implementation details and hyperparameter choices are provided in Appendix C.

Results on fluid dynamics. Table 1 reports the main quantitative comparison. IQ-Diff achieves the best MSE and PSNR, with notably reduced variance compared to TDS-Reward and BoN-16, indicating more consistent physical alignment across test cases. The difference between Single-Q and IQ-Diff is consistent with a benefit from iterative guidance, although the reported configurations also differ in total labeled rollouts. A matched-budget comparison is needed to isolate iteration from additional training data. TDS-Reward underperforms the two Q-guided variants in this implementation; this result concerns the reward-potential adaptation described in Appendix C.5, not every TDS construction. Fig. 2 shows representative qualitative comparisons. DDIM produces scattered, spatially incoherent artifacts; BoN-16 and SGG smooth out fine structure; TDS-Reward captures coarse shape but loses boundary sharpness. IQ-Diff and Single-Q most faithfully recover plume curvature and arch geometry, with IQ-Diff showing tighter agreement to ground truth across all four rows. Additional examples are in Appendix D.1.

Table 1: Comparison on 2D fluid dynamics prediction at $s^* = 10$ (mean $\pm$ std over the test set)

Method	MSE ↓	PSNR ↑	SSIM ↑
DDIM	0.0061 ± 0.0024	22.48 ± 1.83	0.999223 ± 0.000330
BoN-16	0.0013 ± 0.0010	30.81 ± 5.17	0.999841 ± 0.000125
SGG	0.0029 ± 0.0018	26.72 ± 4.23	0.999606 ± 0.000274
TDS-Reward	0.0008 ± 0.0008	31.72 ± 4.02	0.999326 ± 0.000432
Single-Q	0.0007 ± 0.0005	32.56 ± 2.86	$\mathbf{0.999951 \pm 0.000036}$
IQ-Diff	$\mathbf{0.0005 \pm 0.0003}$	$\mathbf{33.88 \pm 2.36}$	0.999948 ± 0.000026

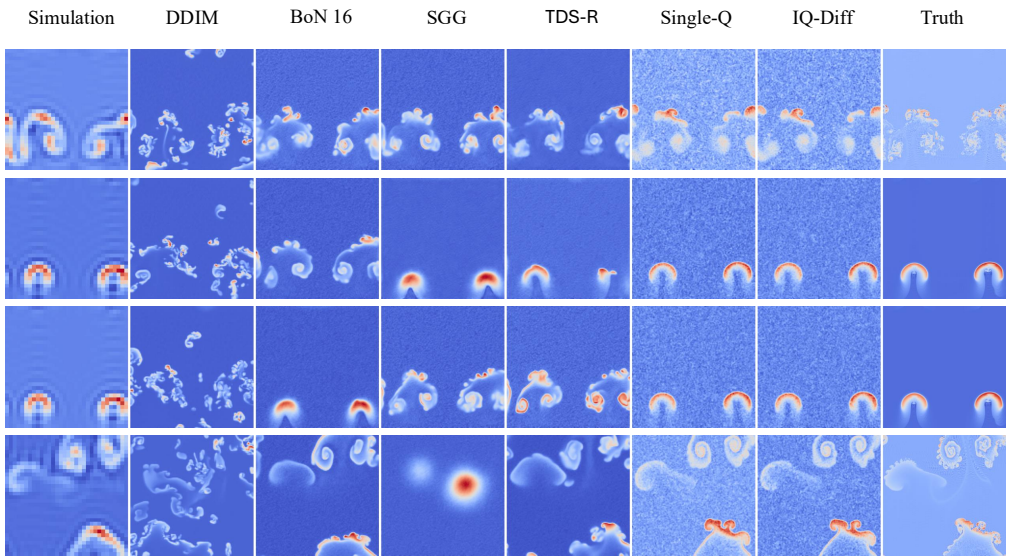

Figure 2: From left: coarse simulator input, five baselines (DDIM, BoN-16, SGG, TDS-Reward, Single-Q), IQ-Diff, and high-fidelity ground truth

Quality-cost tradeoff. Table 2 compares prediction quality with online sampling cost. IQ-Diff and Single-Q use a single U-Net denoising trajectory and require no online reward evaluations, adding only lightweight Q-scoring overhead over DDIM. In contrast, BoN-16 runs 16 complete trajectories and evaluates the reward oracle 16 times per sample, while TDS-Reward propagates P particles across all H denoising steps with PH reward evaluations, making it the most expensive baseline at $14.88\times$ relative time. Without any online reward calls, IQ-Diff achieves the best PSNR: it is $2.2\times$ faster than BoN-16 while improving PSNR by 3.07 dB, and $7.9\times$ faster than TDS-Reward with a 2.16 dB PSNR gain. The $1.88\times$ overhead over DDIM reflects Q-scoring alone. Single-Q at $1.24\times$ relative time shows that even a single Q-function iteration yields substantial gains at near-DDIM cost.

Table 2: Quality-cost comparison on fluid dynamics (per generated sample). Relative time is wall-clock time normalized by DDIM, excluding offline Q-function training. Call counts reflect logical computational scaling. Settings: $H = 50$, $N = 16$, $K = 4$, $P = 16$.

Method	U-Net calls	Q scores	Online reward evals	Relative time ↓	PSNR ↑
DDIM	H	0	0	$1.00\times$	22.48
BoN-16	$16H$	0	16	$4.22\times$	30.81
TDS-Reward	PH	0	PH	$14.88\times$	31.72
Single-Q	H	NH	0	$1.24\times$	32.56
IQ-Diff	H	NKH	0	$1.88\times$	**33.88**

Ablation study. Table 3 ablates the main components of IQ-Diff. **Shuffled-Q** randomly permutes terminal rewards across trajectories before Q training, testing whether reweighting works without rewards correctly attributed to the trajectories that generated them; it falls below DDIM (21.59 vs. 22.48 dB). **Single-Q** ($K = 1$) yields a large jump to 32.56 dB, while **Latest-Q only** drops to 31.93 dB, showing that accumulating Q-functions outperforms using only the most recent one. Full IQ-Diff with $K = 4$ reaches 33.88 dB, with monotone improvement from $K = 1$ to $K = 4$, supporting the role of both reward-aligned Q learning and iterative Q accumulation. Additional ablations on candidate count, guidance strength, reward variants and a 50-epoch diffusion backbone are in Appendix D.2.

Table 3: Ablation on the fluid dynamics benchmark. K denotes the number of outer Q-guidance iterations; $K = 0$ is the unguided sampler, $K = 1$ corresponds to Single-Q, and Latest-Q only uses $Q_{\phi_{K-1}}$ instead of accumulated Q-guidance.

Variant	MSE ↓	PSNR ↑
DDIM / $K = 0$	0.0061 ± 0.0024	22.48 ± 1.83
Shuffled-Q	0.0073 ± 0.0045	21.59 ± 2.37
Single-Q / $K = 1$	0.0007 ± 0.0005	32.56 ± 2.86
IQ-Diff / $K = 2$	0.0007 ± 0.0004	32.98 ± 2.86
Latest-Q only	0.0009 ± 0.0005	31.93 ± 1.89
IQ-Diff / $K = 4$	$\mathbf{0.0005 \pm 0.0003}$	$\mathbf{33.88 \pm 2.36}$

Thermal-process prediction. The thermal task predicts a future temperature field from two observed frames. Its terminal reward compares the prediction with a physics-based PDE prior through hot-region location, outside-support penalties, and structural agreement. We evaluate the same inference-time baselines against held-out temperature fields. Table 4 reports the main comparison, including LPIPS (Zhang et al., 2018) as an additional perceptual metric. IQ-Diff achieves the best mean performance across all three metrics: it reduces MSE relative to the strongest non-IQ-Diff baseline, improves PSNR by 1.92 dB over TDS-Reward, and obtains the lowest LPIPS. The comparison with Single-Q also shows that iterative Q-guidance improves over one-step action-value reweighting in this second physical setting. These results suggest that the benefit of IQ-Diff is not limited to the fluid dynamics benchmark. Detailed task description, reward definition, and additional results are provided in Appendix D.3.

Table 4: Comparison on thermal-process prediction (mean ± std over the test set).

Method	MSE ↓	PSNR ↑	LPIPS ↓
DDIM	0.0497 ± 0.0225	13.04 ± 1.95	0.384 ± 0.087
BoN-16	0.0041 ± 0.0019	23.87 ± 2.03	0.362 ± 0.101
SGG	0.0062 ± 0.0022	22.08 ± 1.55	0.262 ± 0.172
TDS-Reward	0.0028 ± 0.0005	25.53 ± 0.69	0.201 ± 0.074
Single-Q	0.0029 ± 0.0007	25.38 ± 1.05	0.177 ± 0.071
IQ-Diff	$\mathbf{0.0018 \pm 0.0003}$	$\mathbf{27.45 \pm 0.72}$	$\mathbf{0.139 \pm 0.036}$

6 Conclusion

IQ-Diff uses terminal simulator feedback to iteratively guide a frozen diffusion sampler. Each round evaluates the current sampling policy, and the resulting Q-function is accumulated into subsequent candidate selection. Our analysis treats this finite-candidate process directly: exact current-policy evaluation gives monotonic reward improvement and a final-iterate guarantee, while the approximate analysis exposes the effects of relative Q-error and limited proposal coverage. The fluid and thermal experiments report improved prediction accuracy under the tested configurations, with learned guidance replacing online reward queries. The method remains limited by the quality of its return estimates, the alignment of the simulator reward with the evaluation objective, and the high-value candidates proposed by the frozen model.

AI USE STATEMENT

Generative AI tools were used to (i) polish the manuscript's wording, grammar, and readability; (ii) assist in creating the overview illustration (Fig. 1); and (iii) check notation, language, and LaTeX formatting for consistency.

Author contributions and responsibility. The authors developed the original IQ-Diff method, implemented its sampling and Q-learning procedures, and conducted the reported fluid and thermal experiments. They retain responsibility for the final methodological choices, theoretical claims, references, results, and presentation, including all material prepared with AI assistance.

REFERENCES

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 843–852, 2023. URL https://openaccess.thecvf.com/content/CVPR2023W/GCV/html/Bansal_Universal_Guidance_for_Diffusion_Models_CVPRW_2023_paper.html.

Craig Boutilier, Alon Cohen, Avinatan Hassidim, Yishay Mansour, Ofer Meshi, Martin Mladenov, and Dale Schuurmans. Planning and learning with stochastic action sets. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 4674–4682. International Joint Conferences on Artificial Intelligence Organization, 2018. doi: 10.24963/ijcai.2018/650. URL https://www.ijcai.org/proceedings/2018/650.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025. doi: 10.1177/02783649241273668.

Hyungjin Chung, Jeongsol Kim, Michael T McCann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.14687.

Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022. doi: 10.1007/s10915-022-01939-z.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. IDQL: Implicit Q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023. URL https://arxiv.org/abs/2304.10573.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay N. Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-An-Audio: Text-to-audio generation with prompt-enhanced diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 13916–13932. PMLR, 2023. URL https://proceedings.mlr.press/v202/huang23i.html.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. doi: 10.1038/s42254-021-00314-5.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Bernie Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/5a1a10c2c2c9b9af1514687bc24b8f3d-Abstract.html.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/899af0d66d8850318a20781484416152-Abstract-Conference.html.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/260a14acce2a89dad36adc8eefe7c59e-Abstract-Conference.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part I): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017. URL https://arxiv.org/abs/1711.10561.

Allen Z. Ren, Justin Lidard, Lars L. Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c0749c39aaff9e9e4c91f7118bf21b1e-Abstract-Conference.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897. PMLR, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

Naichen Shi, Hao Yan, Shenghan Guo, and Raed Al Kontar. Diffusion-based surrogate modeling and multi-fidelity calibration. *IEEE Transactions on Automation Science and Engineering*, 22: 18217–18230, 2025. doi: 10.1109/TASE.2025.3582171.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alan Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-Kac correctors in diffusion: Annealing, guidance, and product of experts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55906–55949. PMLR, 2025. URL https://proceedings.mlr.press/v267/skreta25a.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://arxiv.org/abs/2010.02502.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/3001ef257407d5a371a96dcd947c7d93-Abstract.html.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems*, volume 34, 2021b. URL https://proceedings.neurips.cc/paper/2021/hash/0a9fdbb17feb6ccb7ec405cfb85222c4-Abstract.html.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025. URL https://arxiv.org/abs/2501.09685.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, Petar Veličković, Max Welling, Linfeng Zhang, Connor W. Coley, Yoshua Bengio, and Marinka Zitnik. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023a. doi: 10.1038/s41586-023-06221-2.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2208.06193.

Luhuan Wu, Brian Trippe, Christian Naesseth, David M. Blei, and John P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/63e8bc7bbf1cfea36d1d1b6538aecce5-Abstract.html.

Lin Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23(282):1–36, 2022. URL https://www.jmlr.org/papers/v23/22-0056.html.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023. doi: 10.1145/3626235. Article 105.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23174–23184, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Yu_FreeDoM_Training-Free_Energy-Guided_Conditional_Diffusion_Model_ICCV_2023_paper.html.

Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20895–20905, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Zhang_Improving_Diffusion_Inverse_Problem_Solving_with_Decoupled_Noise_Annealing_CVPR_2025_paper.html.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

APPENDIX OVERVIEW

Appendix A discusses related work on diffusion guidance, value-based sampling, and scientific prediction. Appendix B provides the complete proofs and clarifies the scope of the theoretical guarantees. Appendix C documents the fluid benchmark, reward oracles, implementation details, and cost accounting. Appendix D presents additional qualitative results, fluid ablations, and the thermal-task setup and results.

A EXTENDED RELATED WORK

Diffusion models for structured generation. Denoising diffusion models have become a central paradigm for high-dimensional generative modeling, with theoretical connections to score matching and stochastic differential equations (Song & Ermon, 2019; Song et al., 2021b; Ho et al., 2020). Their iterative reverse process has been studied for image synthesis (Dhariwal & Nichol, 2021; Rombach et al., 2022; Peebles & Xie, 2023), time-series forecasting (Kollovieh et al., 2023), physical-field calibration (Shi et al., 2025), and robotic policies (Chi et al., 2025; Ren et al., 2025). A large body of work has focused on improving either model architectures or sampling procedures. Latent diffusion models and diffusion transformers improve scalability and representation capacity (Rombach et al., 2022; Peebles & Xie, 2023), while DDIM, DPM-Solver, rectified flow, and consistency models accelerate sampling by reducing the number of denoising steps or straightening the generative trajectory (Song et al., 2021a; Lu et al., 2022; Liu et al., 2023; Song et al., 2023). IQ-Diff addresses guidance rather than acceleration: it uses denoising decisions to incorporate terminal feedback. Its candidate-selection mechanism requires a nondegenerate proposal; a deterministic transition offers no distinct local candidates to reweight.

Sampling-time guidance. Sampling-time guidance methods steer a frozen or partially frozen diffusion model without retraining the entire generative backbone. Classifier guidance modifies the reverse trajectory using gradients from a noisy classifier (Dhariwal & Nichol, 2021), while classifier-free guidance combines conditional and unconditional score estimates without an auxiliary classifier (Ho & Salimans, 2022). More general plug-and-play methods, such as Universal Guidance and FreeDoM, use gradients of external energies, discriminators, or constraint losses to guide the denoising process (Bansal et al., 2023; Yu et al., 2023). In scientific inverse problems, Diffusion Posterior Sampling and related posterior-sampling approaches guide samples using gradients of a measurement-consistency or likelihood term (Chung et al., 2023; Zhang et al., 2025). These methods are highly effective when the guidance objective, likelihood, or forward operator is differentiable, or when a reliable differentiable surrogate is available. IQ-Diff targets a different regime: the available feedback is a terminal scalar reward from an external physics simulator, and gradients through the simulator or a simulator likelihood are not assumed.

Derivative-free inference-time alignment, SMC, and value-based decoding. A complementary line of work shows that gradient access is not strictly necessary for inference-time alignment. Sequential Monte Carlo methods propagate, weight, and resample particles to approximate conditional or reward-tilted distributions. Twisted Diffusion Sampler uses SMC and twisting functions to target conditional diffusion distributions with asymptotic guarantees (Wu et al., 2023). Feynman–Kac Correctors provide a related weighted-simulation and resampling framework for annealing, guidance, and product-of-experts sampling in diffusion models (Skreta et al., 2025). Recent tutorials further unify SMC-based guidance, value-based sampling, and classifier-style guidance as different approximations to soft optimal denoising processes (Uehara et al., 2025). Soft value-based decoding methods perform derivative-free reward optimization by learning soft value approximations and using them to select value-weighted candidates during sampling (Li et al., 2025).

IQ-Diff shares the local value-weighted candidate-selection mechanism with value-based decoding. Its emphasis is the repeated evaluation of the executed sampler and accumulation of the resulting scores. Since the denoising action specifies the next state, the exact action value equals the next-state continuation value for $h > 1$; a transition-level neural parameterization alone is therefore not the claimed novelty. Iterative updates can change that continuation value, motivating new on-policy rollout data. Our theory concerns expected reward within a fixed candidate-selection class and does not establish superiority over all soft-value or SMC methods.

Diffusion policies, Q-learning, and reinforcement learning. Diffusion models have also been studied as expressive policy classes in reinforcement learning and control. Diffusion Q-Learning uses a diffusion policy together with a learned critic for offline reinforcement learning (Wang et al., 2023b), and Implicit Diffusion Q-Learning uses critic-based action selection without backpropagating through the entire denoising chain (Hansen-Estruch et al., 2023). Diffusion Policy and Diffusion Policy Policy Optimization formulate action generation as a denoising process and optimize diffusion policies for robotic control (Chi et al., 2025; Ren et al., 2025). These works typically focus on learning or fine-tuning policies for control tasks. IQ-Diff instead treats a pretrained conditional diffusion model as a frozen base policy and uses action-value functions for sampling-time improvement under terminal simulator rewards. IQ-Diff learns scalar return estimators for candidate ranking; it does not train a full simulator-output surrogate or differentiate through the reward oracle.

Physics-guided diffusion and simulator-based calibration. There is growing interest in incorporating physical knowledge into generative models for scientific prediction and inverse problems (Karniadakis et al., 2021; Wang et al., 2023a). Physics-guided diffusion methods often rely on differentiable residuals, measurement operators, or surrogate likelihoods, which are natural when governing equations are accessible and differentiable. Another related direction uses simulator outputs as conditioning or calibration signals in multi-fidelity physical modeling; for example, diffusion-based surrogate modeling and multi-fidelity calibration can improve physical consistency by conditioning on simulator predictions (Shi et al., 2025). IQ-Diff addresses a complementary use case: the simulator is treated as an external terminal reward oracle rather than as a differentiable forward model or a likelihood surrogate. This makes the method applicable when simulator feedback is useful but only available as sparse scalar evaluations after complete generation.

B Proofs and Theoretical Details

We prove Theorem 1 and Proposition 1. The proof keeps the proposal and candidate count fixed and evaluates the actual sampler $\widehat{\pi}_k$. Throughout, logarithms are natural, $\langle f, p\rangle = \sum_{j=1}^{N} f_j p_j$, and $\mathrm{TV}(p, q) = \frac{1}{2}\|p - q\|_1$. We use $z_h = (s_h, \mathcal{C}_h)$ only in this appendix to shorten expressions involving a state and its candidate tuple. The index h is the number of remaining denoising decisions, so trajectories proceed from H down to 1.

B.1 Finite-Candidate Representation

State and transition. At $s_h = (x_h, h, c)$, draw the ordered tuple $\mathcal{C}_h = (a_h^{(1)}, \dots, a_h^{(N)})$ independently from $\pi_0(\cdot \mid s_h)$. Tuples can contain repeated actions. A Markov selection rule $p_h(j \mid z_h)$ chooses an index j. For $h > 1$, the next original state is

$$s_{h-1} = (a_h^{(j)}, h - 1, c),$$

and its candidate tuple is drawn afresh. For $h = 1$, the selected action receives $R(a_1^{(j)}; c)$ and the process terminates. All preceding rewards are zero. The value after termination is zero. The initial law of s_H is shared by all policies, and $\mathcal{C}_H$ is sampled conditional on s_H by the same proposal.

For any measurable action set B, the marginal kernel induced by a selection rule p is

$$\widehat{\pi}_p(B \mid s_h) = \mathbb{E}_{\mathcal{C}_h \mid s_h}\left[\sum_{j=1}^{N} p_h(j \mid s_h, \mathcal{C}_h)\mathbf{1}\{a_h^{(j)} \in B\}\right]. \tag{21}$$

Integrating out the tuple and index at each step shows that the augmented process and $\widehat{\pi}_p$ have identical distributions of original states, selected actions, and rewards. In particular, the rule p_k in (14) induces $\widehat{\pi}_k$. Uniform selection induces π_0, since each entry of the tuple has law π_0.

Values and the regression target. Write the true candidate action values and augmented-state values at iteration k as

$$\begin{aligned} q_{k,h}(z_h, j) &:= Q_h^{\widehat{\pi}_k}(s_h, a_h^{(j)}), \\ \widetilde{V}_{k,h}(z_h) &:= \sum_{j=1}^{N} p_{k,h}(j \mid z_h) q_{k,h}(z_h, j). \end{aligned} \tag{22}$$

These are the action-value and value functions of the augmented process. To check this, at $h = 1$ both descriptions assign the reward $R(a_1^{(j)}; c)$ to index j. At $h > 1$, selection fixes s_{h-1}, and the next tuple is sampled independently of the unused current candidates. Hence, by induction,

$$\begin{aligned} q_{k,h}(z_h, j) &= \mathbb{E}_{\mathcal{C}_{h-1}|s_{h-1}}[\widetilde{V}_{k,h-1}(s_{h-1}, \mathcal{C}_{h-1})] \\ &= V_{h-1}^{\widehat{\pi}_k}(s_{h-1}) = Q_h^{\widehat{\pi}_k}(s_h, a_h^{(j)}). \end{aligned} \tag{23}$$

Averaging (22) over $\mathcal{C}_h$ gives the original value $V_h^{\widehat{\pi}_k}(s_h)$. This induction also proves that a rollout from the current sampler has population regression target

$$\mathbb{E}[r \mid s_h, a_h; \widehat{\pi}_k] = Q_h^{\widehat{\pi}_k}(s_h, a_h) \tag{24}$$

on the rollout distribution. No additional tuple input is needed by the Q-network. Bounded terminal reward implies $|q_{k,h}| \le V_{\max}$ and $|\widetilde{V}_{k,h}| \le V_{\max}$ for every k, h.

Existence of a candidate-optimal comparator. The optimum defining $J_N^\star$ is attained by a measurable Markov selection rule. At $h = 1$, maximize $R(a_1^{(j)}; c)$ over the finite index set. At each earlier decision, maximize the expected optimal continuation value, averaging over the next freshly drawn tuple. A finite maximum of bounded measurable functions is measurable; select the smallest maximizing index to resolve ties. Backward induction establishes optimality among all measurable Markov selection rules. Denote one such rule by $p_h^\star(\cdot \mid z_h)$, its marginal policy by $\widehat{\pi}_N^\star$, and its augmented-state distribution at step h by $\widetilde{d}_h^\star$. These objects are fixed across learning iterations. The comparator need not be representable by finite neural-network logits.

Exact exponential update. Since $F_{k+1} = F_k + \rho Q_{\phi_k}$, substituting into the softmax of the current tuple gives (14) exactly. This statement does not require shared tuples across iterations, and applies to the conditional index rule rather than the marginal action kernel (21). Finite logits and uniform initialization give strictly positive probabilities at every finite iteration.

B.2 Three Elementary Identities

Lemma B.1 (Performance difference)**.** *For any two selection rules p, p' with induced policies $\widehat{\pi}_p, \widehat{\pi}_{p'}$, let $\widetilde{d}_h^p$ be the augmented-state distribution under p. Then*

$$J(\widehat{\pi}_p) - J(\widehat{\pi}_{p'}) = \sum_{h=1}^{H} \mathbb{E}_{z_h \sim \widetilde{d}_h^p} \left[\sum_{j=1}^{N} Q_h^{\widehat{\pi}_{p'}}(s_h, a_h^{(j)}) \big(p_h(j \mid z_h) - p'_h(j \mid z_h) \big) \right]. \tag{25}$$

Proof. Let $\widetilde{V}_h^{p'}$ be the value under p' in the augmented process. For index j, its advantage is its immediate reward plus the expected continuation value $\widetilde{V}_{h-1}^{p'}$, minus $\widetilde{V}_h^{p'}$. At $h = 1$, continuation value is zero and immediate reward is terminal reward. Take expectations along a trajectory of p and sum over all steps. Consecutive continuation-value terms cancel. The remaining terms are $J(\widehat{\pi}_p)$ minus the initial expected value of p', which is $J(\widehat{\pi}_{p'})$ because the initial laws coincide. Finally, using the value identity in (22), the expected advantage at a fixed augmented state is the inner product in (25). □

Lemma B.2 (Exponential-update identities)**.** *Let p be a strictly positive probability vector, $f \in \mathbb{R}^N$, and $\rho > 0$. Define $p_j^+ = p_j e^{\rho f_j}/Z$, with $Z = \sum_j p_j e^{\rho f_j}$. For any probability vector u,*

$$\rho\langle f, p^+ - p\rangle = D_{\mathrm{KL}}(p^+ \| p) + D_{\mathrm{KL}}(p \| p^+), \tag{26}$$

$$\rho\langle f, u - p^+\rangle = D_{\mathrm{KL}}(u \| p) - D_{\mathrm{KL}}(u \| p^+) - D_{\mathrm{KL}}(p^+ \| p). \tag{27}$$

Proof. The logarithmic ratio satisfies $\log(p_j^+/p_j) = \rho f_j - \log Z$. Taking its inner product with $p^+ - p$ eliminates $\log Z$ and proves (26). Also,

$$D_{\mathrm{KL}}(u \| p) - D_{\mathrm{KL}}(u \| p^+) = \rho\langle f, u\rangle - \log Z,$$

whereas $D_{\mathrm{KL}}(p^+ \| p) = \rho\langle f, p^+\rangle - \log Z$. Subtracting the latter identity proves (27). Zero entries in u are allowed with the convention $0 \log 0 = 0$. □

Lemma B.3 (Value-change recursion)**.** *Define*

$$\delta_{k,h}(z) = \langle q_{k,h}(z,\cdot), p_{k+1,h}(\cdot \mid z) - p_{k,h}(\cdot \mid z)\rangle,$$
$$\Delta_{k,h}(z) = \widetilde{V}_{k+1,h}(z) - \widetilde{V}_{k,h}(z).$$

For $h > 1$,

$$\Delta_{k,h}(z) = \delta_{k,h}(z) + \mathbb{E}_{p_{k+1}}[\Delta_{k,h-1}(z_{h-1}) \mid z_h = z], \tag{28}$$

and $\Delta_{k,1}(z) = \delta_{k,1}(z)$. The expectation includes the new rule's current index choice and the next candidate draw.

Proof. In the difference of the two augmented values, add and subtract $\langle q_{k,h}, p_{k+1,h}\rangle$. The change in current index probabilities is $\delta_{k,h}$. The remaining term is the continuation-value difference averaged under the new current index choice and the common transition kernel, giving (28). At $h = 1$, the reward following a fixed index is policy-independent, so that remaining term is zero. □

B.3 Proof of Theorem 1

Proof. Set $b = \rho\varepsilon_Q^2/16$. We first control local and global degradation, then telescope a comparator inequality and convert it to a final-iterate bound.

Step 1: local degradation. Fix k, h, z_h. Write $p = p_{k,h}(\cdot \mid z_h)$, $p^+ = p_{k+1,h}(\cdot \mid z_h)$, $q = q_{k,h}(z_h,\cdot)$, and $f_j = Q_{\phi_k}(s_h, a_h^{(j)})$. The error vector $e = f - q$ satisfies $\max_j e_j - \min_j e_j \le \varepsilon_Q$. For any probability vectors u, v,

$$|\langle e, u - v\rangle| \le \varepsilon_Q \,\mathrm{TV}(u, v). \tag{29}$$

Indeed, subtract the midpoint of the largest and smallest entries from e. The resulting vector has supremum norm at most $\varepsilon_Q/2$, while the inner product is unchanged because $\sum_j (u_j - v_j) = 0$.

Using (26) for the score $f = q + e$, the true local improvement is

$$\delta_{k,h}(z_h) = \frac{D_{\mathrm{KL}}(p^+\|p) + D_{\mathrm{KL}}(p\|p^+)}{\rho} - \langle e, p^+ - p\rangle.$$

For $t = \mathrm{TV}(p^+, p)$, Pinsker's inequality gives a lower bound $4t^2$ on the sum of the two KL divergences. Therefore,

$$\begin{aligned}\delta_{k,h}(z_h) &\ge \frac{4t^2}{\rho} - \varepsilon_Q t \\ &= \frac{4}{\rho}\Big(t - \frac{\rho\varepsilon_Q}{8}\Big)^2 - \frac{\rho\varepsilon_Q^2}{16} \ge -b.\end{aligned} \tag{30}$$

The bound holds for every $\rho > 0$; minimizing the quadratic over all real t only weakens the bound compared with its feasible interval $[0, 1]$.

Step 2: global and downstream degradation. Apply Lemma B.1 with the new rule as the first argument. Equation (30) yields

$$J(\widehat{\pi}_{k+1}) - J(\widehat{\pi}_k) = \sum_{h=1}^{H} \mathbb{E}_{\tilde{d}_h^{p_{k+1}}}[\delta_{k,h}] \ge -Hb, \tag{31}$$

which proves (16). Starting from $\Delta_{k,1} \ge -b$ and using Lemma B.3, induction also gives

$$\Delta_{k,h}(z_h) \ge -hb, \qquad \delta_{k,h}(z_h) \le \Delta_{k,h}(z_h) + (h-1)b. \tag{32}$$

The first inequality counts the h decisions remaining. The second follows by rearranging the recursion and applying the first at step $h - 1$; it also holds at $h = 1$.

Step 3: comparison with the optimal candidate rule. Let $g_k = J_N^\star - J(\widehat{\pi}_k)$ and define

$$D_k = \sum_{h=1}^{H} \mathbb{E}_{z_h \sim \widetilde{d}_h^\star} \left[D_{\mathrm{KL}}\big(p_h^\star(\cdot \mid z_h) \| p_{k,h}(\cdot \mid z_h)\big) \right], \tag{33}$$

$$B_k = \sum_{h=1}^{H} \mathbb{E}_{z_h \sim \widetilde{d}_h^\star} [\widetilde{V}_{k,h}(z_h)]. \tag{34}$$

Both potentials use the same fixed comparator distributions at every iteration. They do not average under a distribution that changes with k.

At the fixed augmented state from Step 1, set $u = p_h^\star(\cdot \mid z_h)$ and apply (27) to $f = q + e$. Then

$$\begin{aligned}
\langle q, u - p \rangle &= \langle f, u - p^+ \rangle + \langle q, p^+ - p \rangle + \langle e, p^+ - u \rangle \\
&= \frac{D_{\mathrm{KL}}(u\|p) - D_{\mathrm{KL}}(u\|p^+) - D_{\mathrm{KL}}(p^+\|p)}{\rho} + \delta_{k,h}(z_h) + \langle e, p^+ - u \rangle \\
&\le \frac{D_{\mathrm{KL}}(u\|p) - D_{\mathrm{KL}}(u\|p^+)}{\rho} + \Delta_{k,h}(z_h) + (h-1)b + \varepsilon_Q.
\end{aligned}$$

The final line discards a nonpositive KL term and uses (29), $\mathrm{TV}(p^+, u) \le 1$, and (32). Average under $\widetilde{d}_h^\star$, sum over h, and apply Lemma B.1 with the comparator as the first argument. This gives

$$g_k \le \frac{D_k - D_{k+1}}{\rho} + B_{k+1} - B_k + H\varepsilon_Q + \frac{H(H-1)}{2} b. \tag{35}$$

Step 4: finiteness and telescoping. At a fixed tuple, the range of q is at most $2V_{\max}$, and the range of e is at most ε_Q. Thus the learned score range is at most $2V_{\max} + \varepsilon_Q$. Accumulation from uniform initialization implies

$$p_{k,h}(j \mid z_h) \ge \frac{1}{N} \exp\{-k\rho(2V_{\max} + \varepsilon_Q)\}.$$

Consequently each KL integrand in (33) is bounded above by $\log N + k\rho(2V_{\max} + \varepsilon_Q)$, so D_k is finite. This requires no bound on common score offsets and no iteration-independent bound on cumulative logits.

For any probability vector u, $D_{\mathrm{KL}}(u\|\,\mathrm{Unif}[N]) \le \log N$. Hence $D_0 \le H \log N$. Also $D_K \ge 0$ and, by bounded terminal reward, $B_K - B_0 \le 2HV_{\max}$. Sum (35) for $k = 0, \dots, K-1$ and divide by K:

$$\frac{1}{K} \sum_{k=0}^{K-1} g_k \le \frac{H \log N}{\rho K} + \frac{2HV_{\max}}{K} + H\varepsilon_Q + \frac{H(H-1)}{2} b. \tag{36}$$

Step 5: the final iterate. Equation (31) implies $g_{k+1} \le g_k + Hb$. Applying this successively through iteration K gives

$$g_K \le g_k + H(K-k)b, \qquad k = 0, \dots, K-1.$$

Averaging over these K indices and using $\sum_{k=0}^{K-1}(K-k) = K(K+1)/2$ yields

$$g_K \le \frac{1}{K} \sum_{k=0}^{K-1} g_k + \frac{H(K+1)}{2} b.$$

Combining with (36), the two quadratic-error coefficients sum to

$$\frac{H(H-1)}{2} b + \frac{H(K+1)}{2} b = \frac{H(H+K)}{2} b = \frac{\rho H(H+K)\varepsilon_Q^2}{32}.$$

This proves (17). Its left-hand nonnegativity follows because $\widehat{\pi}_K$ is induced by a feasible candidate selection rule. With exact current-policy evaluation, $\varepsilon_Q = 0$, so (31) gives monotonicity and the same bound reduces to (18). □

B.4 Proof of Proposition 1

Proof. The Bellman-optimal functions in the proposition are for the original decision process without the candidate restriction; they are distinct from the candidate-optimal value $J_N^\star$. Consider the feasible selection rule that chooses the smallest index maximizing $Q_h^\star(s_h, a_h^{(j)})$ in each current tuple. Denote its induced sampler by $\widehat{\pi}^{\mathrm{gr}}$. Finite maximization and the stated measurability assumptions make this rule measurable.

Fix s_h. By (19) and independent proposals, the probability that all N candidates miss the ε-optimal region is at most $(1-p_\varepsilon)^N$. If a candidate lies in that region, the greedy rule has Bellman loss at most ε. Otherwise its loss is at most $2V_{\max}$. Therefore,

$$\mathbb{E}_{\mathcal{C}_h|s_h}\left[V_h^\star(s_h) - \max_j Q_h^\star(s_h, a_h^{(j)})\right] \le \varepsilon + 2V_{\max}(1-p_\varepsilon)^N. \tag{37}$$

Along a trajectory of $\widehat{\pi}^{\mathrm{gr}}$, Bellman telescoping gives

$$J^\star - J(\widehat{\pi}^{\mathrm{gr}}) = \sum_{h=1}^{H} \mathbb{E}_{\widehat{\pi}^{\mathrm{gr}}}[V_h^\star(s_h) - Q_h^\star(s_h, a_h)]. \tag{38}$$

Specifically, for $h > 1$, $Q_h^\star(s_h, a_h) = V_{h-1}^\star(s_{h-1})$, and for $h = 1$, $Q_1^\star(s_1, a_1) = R(a_1; c)$. Successive value terms cancel, leaving $\mathbb{E}[V_H^\star(s_H)] - \mathbb{E}[R(x_0; c)]$. This uses the assumed equality $J^\star = \mathbb{E}[V_H^\star(s_H)]$, and does not require an unrestricted optimal action to be attained at every continuous state.

Applying (37) at each visited state gives

$$J^\star - J(\widehat{\pi}^{\mathrm{gr}}) \le H\varepsilon + 2HV_{\max}(1-p_\varepsilon)^N.$$

By optimality in the candidate class, $J_N^\star \ge J(\widehat{\pi}^{\mathrm{gr}})$. Also $J^\star \ge J_N^\star$, since candidate-based policies are feasible in the original process. These two inequalities prove (20). □

Combined error decomposition. Under the joint assumptions of the theorem and proposition,

$$\begin{aligned} J^\star - J(\widehat{\pi}_K) \le\ & \underbrace{H\varepsilon + 2HV_{\max}(1-p_\varepsilon)^N}_{\text{proposal coverage}} \\ & + \underbrace{\frac{H\log N}{\rho K} + \frac{2HV_{\max}}{K}}_{\text{optimization within the candidate class}} \\ & + \underbrace{H\varepsilon_Q + \frac{\rho H(H+K)\varepsilon_Q^2}{32}}_{\text{current-policy Q error}}. \end{aligned} \tag{39}$$

The coverage assumption is required only for the proposition and this combined bound, not for Theorem 1.

Candidate count and degenerate proposals. An $(N+1)$-candidate rule can ignore its last candidate and reproduce any N-candidate rule. Thus $J_{N+1}^\star \ge J_N^\star$; the $\log N$ optimization term does not mean that more candidates reduce the best achievable reward. When $N = 1$, the selected action necessarily follows π_0, so $\widehat{\pi}_k = \pi_0$ for every k. If every base transition is a point mass, all candidates coincide and the same conclusion holds for every N. Finally, whenever $\pi_0(B \mid s) = 0$, equation (21) gives $\widehat{\pi}_p(B \mid s) = 0$ for every candidate rule: reweighting cannot create local probability mass outside the proposal.

B.5 Scope and Implementation

Learning and coverage. Theorem 1 is conditional on a uniform relative-error bound for the current executed policy. Equation (24) identifies the correct population target but does not guarantee that finite rollouts cover every relevant state and action. Moreover, the H pairs within a rollout share

the same terminal reward and are dependent. A learning-rate result would need to account for the independent trajectory units, the function class, and transfer from the rollout distribution to the uniform norm in Assumption 3. We do not infer an $M^{-1/2}$ uniform Q-error rate from an average squared-loss bound.

Reward and evaluation metric. A negative squared-error reward is bounded on a bounded field domain, but not globally bounded merely because it is a squared-error expression. Assumption 1 therefore remains explicit; we do not silently clip or transform the experimental reward. The guarantees are for the training reward, not automatically for MSE or PSNR against a different high-fidelity reference.

Update and numerical stability. The theorem requires the accumulated categorical rule. Subtracting a common maximum before softmax preserves its probabilities, while active clipping or replacement by greedy selection changes the update. The per-step KL is relative to the previous conditional index rule, not a fixed global KL constraint to π_0. Changing the base model defines a different fixed-proposal problem with its own $J_N^\star$ and Q-functions. The base-model ablation compares such runs; it does not fall under a guarantee for changing the proposal during a single run.

C Experimental Details

This appendix provides implementation details for the main fluid dynamics benchmark in Sec. 5, including data generation, reward construction, baseline implementations, hyperparameters, and timing protocol. Details for the thermal-process task are provided separately in Appendix D.3.

C.1 Fluid Dynamics Setup

We use the 2D buoyancy-driven fluid prediction task as the main benchmark. The dynamics are governed by the Boussinesq approximation of the Navier–Stokes equations. The state variable is the buoyancy field. Following the simulator setup of Shi et al. (2025), high-fidelity ground-truth trajectories are generated by running a CFD simulator on 128×128 grids from randomized initial buoyancy and vorticity. Each simulation runs from $s = 0$ to $s = 10$. The buoyancy field at $s = 0$ is used as the context c, and the target x_0 is the future buoyancy field at $s^* = 10$.

The dataset contains 6880 simulations, split into 90% for training and 10% for testing. All fields are represented on 128×128 grids for model training and quantitative evaluation; generated outputs are upsampled to 256×256 only for visualization. During sampling-time guidance, the terminal reward is computed using a 32×32 coarse-grid simulator reference, whereas all reported MSE, PSNR, and SSIM metrics are computed against the high-fidelity ground truth at the evaluation resolution.

C.2 Base Diffusion Model

All inference-time methods use the same frozen conditional diffusion model and the same sampling horizon. The base model is a conditional U-Net denoiser with four resolution levels, width multipliers $(1, 2, 4, 8)$, and base channel size 32. Each resolution level uses ResNet blocks with GroupNorm and SiLU activations. We use lightweight attention at early scales and full attention at the coarsest scale. The diffusion timestep is encoded using sinusoidal embeddings followed by an MLP. Context observations are concatenated channel-wise with the noisy input. The base model is trained with the standard noise-prediction objective and is not updated during IQ-Diff guidance or any baseline comparison.

C.3 Fluid Reward Oracles

Low-fidelity reward oracle. The main fluid experiments use a lower-fidelity simulator reference. Specifically, we run the same Navier–Stokes–Boussinesq simulator from the same initial condition on a 32×32 coarse grid. Let $\Phi_{\mathrm{low}}(c)$ denote the resulting coarse-grid future buoyancy field at $s^* = 10$, mapped to the comparison resolution when needed. The terminal reward used for Q-function training

Table 5: Base diffusion training and sampling settings for the fluid benchmark.

Parameter	Value	Description
Training noise steps T	1000	Diffusion training schedule length
Inference horizon H	50	DDIM sampling steps
Noise schedule β_t	linear $[10^{-4}, 2 \times 10^{-2}]$	Training variance schedule
DDIM stochasticity η	0.5	Sampling stochasticity
Architecture	U-Net	Conditional denoising network
Base channels	32	Base U-Net channel width
Width multipliers	$(1, 2, 4, 8)$	U-Net resolution multipliers
Optimizer	AdamW	Base-model optimizer
Batch size	16	Training batch size
Learning rate	10^{-4}	Base-model learning rate
Weight decay	5×10^{-6}	Base-model weight decay
Epochs	201	Training epochs
Checkpoint interval	50 epochs	Checkpoint saving interval
Training resolution	128×128	Input resolution used by the model
Evaluation resolution	128×128	Output resolution used for metrics

is the negative mean squared error between the generated field and the simulator reference:

$$R_{\text{low}}(x_0; c) = -\,\text{MSE}(x_0, \Phi_{\text{low}}(c)) = -\frac{1}{d}\,\|x_0 - \Phi_{\text{low}}(c)\|_2^2\,,$$

where d is the number of scalar entries in each field at the comparison resolution, and the norm is taken over the vectorized fields. The mean is computed over the entries of each individual prediction, not across a batch. Thus, the reward is a fast coarse-simulator proxy available during sampling-time alignment, while final evaluation is performed against the high-fidelity ground truth.

Higher-fidelity reward variant. We also evaluate a higher-fidelity reward oracle as an additional reward-variant study. In this setting, the reward reference is generated by running the fluid simulator at 64×64 resolution. Let $\Phi_{\text{high}}(c)$ denote this higher-fidelity simulator reference, mapped to the same comparison resolution as x_0. Using the same per-sample mean squared error definition, the reward is

$$R_{\text{high}}(x_0; c) = -\,\text{MSE}(x_0, \Phi_{\text{high}}(c)) = -\frac{1}{d}\,\|x_0 - \Phi_{\text{high}}(c)\|_2^2\,.$$

This setting represents a more accurate but more expensive reward oracle. It is not used in the main experiments; results are reported in Table 9.

C.4 Q-Function Training

Each action-value function Q_{ϕ_k} takes the denoising state $s_h = (x_h, h, c)$ and action $a_h = x_{h-1}$ as input. The timestep h is encoded using sinusoidal embeddings followed by a two-layer MLP, and the state, action, and context tensors are combined by bilinearly resizing all tensors to a common spatial resolution and concatenating them channel-wise. We train Q_{ϕ_k} by the Monte Carlo regression objective in (11).

C.5 Baseline Implementations

All baselines use the same frozen diffusion model, the same DDIM sampling schedule, the same stochasticity parameter $\eta = 0.5$, the same denoising horizon $H = 50$, and the same test split.

DDIM. DDIM uses the frozen base sampler without simulator feedback.

BoN-16. BoN-16 draws 16 complete samples from the frozen base sampler and selects the one with the highest terminal simulator reward. All samples use the same DDIM schedule and stochasticity parameter as DDIM. This baseline requires 16 online reward evaluations per generated sample.

Table 6: IQ-Diff and Q-function training hyperparameters for the fluid benchmark.

Parameter	Value	Description
Outer iterations K	4	Number of Q-guidance iterations
Rollouts M	1000	Rollouts per Q-training iteration
Candidates N	16	Local candidates per denoising step
Guidance steps ρ_i	$(1, 1, 1, 1)$	Uniform Q-guidance weights across iterations
Logit clipping Λ	none	No explicit logit clipping is used
Q optimizer	AdamW	Optimizer for Q-function training
Q learning rate	10^{-4}	Q optimizer learning rate
Q weight decay	10^{-5}	Q optimizer weight decay
Q batch size	64	Batch size for Q regression
Q training epochs	10000 gradient steps	Optimization steps per Q-function
Gradient clip	1.0	Gradient clipping for Q training

SGG. SGG uses a differentiable physics-consistency correction inspired by diffusion-based surrogate modeling and multi-fidelity calibration (Shi et al., 2025). The predicted clean sample is adjusted using the gradient of a consistency loss with respect to that sample. This baseline has access to a simulator reference or differentiable correction signal, whereas IQ-Diff is supplied scalar reward labels. A fixed-reference squared-error loss is differentiable in the generated field even when the simulator producing the reference is a black box; this does not require differentiating through the simulator itself.

TDS-Reward. TDS-Reward is a TDS-style particle-resampling baseline inspired by Twisted Diffusion Sampling (Wu et al., 2023). It adapts the particle weighting and resampling structure of TDS to our terminal-reward setting, rather than serving as an exact reproduction of the original conditional-sampling algorithm. We define a reward-induced terminal likelihood

$$L(x_0; c) = \exp\{\alpha_{\mathrm{TDS}} R(x_0; c)\}.$$

Because the exact look-ahead likelihood $\mathbb{E}[L(x_0; c) \mid x_h]$ is unavailable, we use the predicted clean sample $\hat{x}_0(x_h, h, c)$ to construct the intermediate potential

$$G_h(x_h; c) = \exp\{\alpha_{\mathrm{TDS}} R(\hat{x}_0(x_h, h, c); c)\}.$$

The sampler maintains $P = 16$ particles. At each denoising step, each particle is propagated once by the frozen diffusion proposal, assigned the potential G_h, and resampled according to the normalized particle weights. We evaluate this reward-induced potential for every particle at every denoising step, so the online reward-evaluation cost is PH per generated sample. With $P = 16$ and $H = 50$, this corresponds to 800 online reward evaluations. We tune α_{TDS} on the validation set over $\{0.1, 0.3, 1, 3, 10\}$. The final sample is selected from the terminal particle population according to the highest weight.

Single-Q. Single-Q corresponds to $K = 1$. It trains one action-value function from base-sampler rollouts and uses it for finite-candidate local reweighting with the same candidate count $N = 16$ as IQ-Diff.

IQ-Diff. IQ-Diff uses $K = 4$ learned Q-functions and accumulates their Q-guided logits during candidate reweighting. Unless otherwise stated, $N = 16$ candidates are used at each denoising step.

C.6 Timing and Cost Accounting

We report online sampling cost separately from offline training cost. Online cost includes diffusion-backbone evaluations, Q-scoring, online reward evaluations, candidate selection, and wall-clock time per generated sample. Offline cost includes reward evaluations used to label trajectories for Q-function training.

Cost accounting. Logical U-Net calls count diffusion-backbone transitions. Q scores count scalar candidate scores; in implementation, candidate scores are evaluated in batches. Online reward

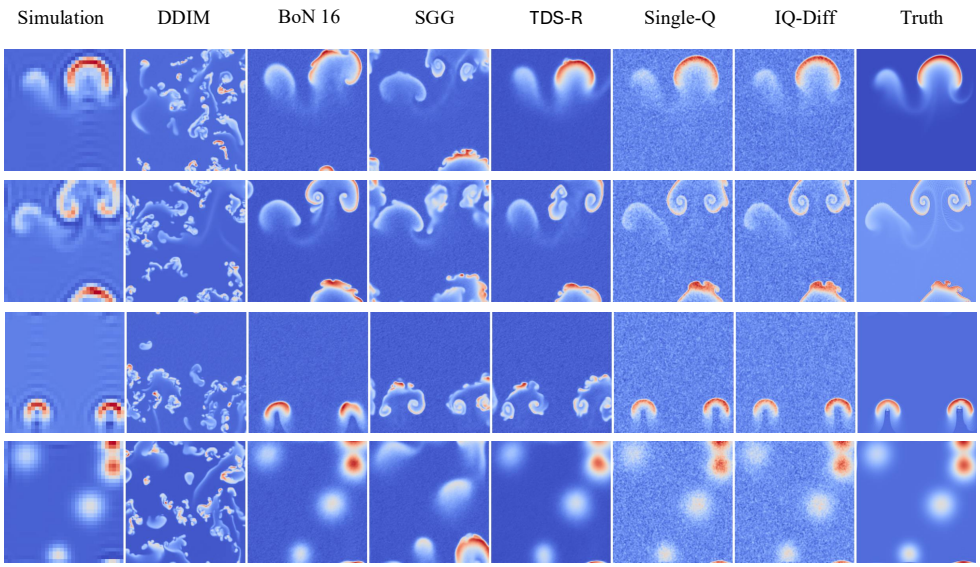

Figure 3: Additional qualitative comparisons on the fluid dynamics benchmark. Each row corresponds to a held-out test instance.

evaluations count test-time evaluations of the reward oracle used for final-sample selection or particle weighting. For IQ-Diff, the diffusion backbone is evaluated once per denoising step to specify the frozen transition distribution; candidates are sampled from this transition and scored by learned Q-functions. For BoN-16, 16 complete diffusion trajectories and 16 terminal reward evaluations are used for test-time selection. For TDS-Reward, P particles are propagated during denoising, resulting in PH diffusion transitions and PH online reward evaluations in the main setting.

D Additional Experimental Results

D.1 Additional Fluid Qualitative Results

We provide additional qualitative comparisons on the fluid dynamics benchmark in Fig. 3. The examples are selected from held-out test instances and complement the representative visualization in the main text.

D.2 Fluid Dynamics Ablations

We report additional ablations on the fluid dynamics benchmark.

Candidate count N. Keeping all other hyperparameters fixed, we vary the number of local candidates N used in finite-candidate reweighting. Table 7 shows that $N = 1$, which effectively removes candidate selection, performs close to unguided sampling. Increasing N substantially improves PSNR, with diminishing returns beyond $N = 16$. We therefore use $N = 16$ as the default setting in the main experiments to balance prediction quality and Q-scoring cost.

Guidance scale ρ. We vary the Q-guidance scale ρ_i while keeping all other hyperparameters fixed. Table 8 shows that the constant setting $\rho_i = 1.0$ achieves the highest PSNR among the tested choices. Larger guidance scale can over-emphasize learned Q-values, while smaller guidance scale may under-utilize the simulator feedback. The decreasing schedule $\rho_i = 1/\sqrt{i}$ is an empirical alternative. The theory analyzes a constant step size and does not identify the best empirical guidance scale.

Shuffled-Q control. The Shuffled-Q result is reported in the main text. In this control, we train the same Q-function architecture after randomly permuting terminal rewards across trajectories.

Table 7: Ablation on the number of local candidates N for IQ-Diff. Results are PSNR mean $\pm$ std over the evaluation set.

N	PSNR ↑
1	22.48 ± 1.83
4	28.58 ± 2.36
8	31.59 ± 1.64
16	33.88 ± 2.36
32	$\mathbf{34.03 \pm 1.87}$

Table 8: Ablation on the Q-guidance scale ρ_i, where $i = 1, \ldots, 4$. We report PSNR (mean $\pm$ std) on the test set.

Guidance scale setting	PSNR ↑
$\rho_i = 2.0$	32.07 ± 2.74
$\rho_i = 1.0$	$\mathbf{33.88 \pm 2.36}$
$\rho_i = 0.5$	32.31 ± 3.44
$\rho_i = 1/\sqrt{i}$	31.57 ± 3.21

This preserves the marginal reward distribution but breaks the correspondence between denoising actions and final simulator outcomes. The poor performance of Shuffled-Q confirms that candidate reweighting requires meaningful reward-action correspondence.

Reward variants. The main fluid experiments use the 32×32 low-fidelity reward oracle. Here we report a parallel comparison using a 64×64 higher-fidelity reward oracle. This experiment tests whether IQ-Diff can further benefit from a more accurate, but more expensive, simulator reward. The overall trend remains consistent with the main results: IQ-Diff achieves the best MSE, PSNR, and SSIM, while TDS-Reward and Single-Q are the strongest competing baselines.

Partially trained diffusion backbone. We replace the main pretrained backbone with the checkpoint saved after 50 epochs of the same diffusion training run and keep it frozen for all samplers. For Single-Q and IQ-Diff, we collect new rollouts and train new Q-functions using this checkpoint, rather than transferring Q-functions from the main backbone. The original reward, sampling configuration, and Q-training budget are unchanged. This ablation examines how guidance depends on the degree of backbone pretraining. Table 10 reports prediction accuracy against the high-fidelity ground truth.

D.3 Thermal-Process Experiments

Thermal-Process Setup We use the LBMAM thermal-process prediction task from the diffusion-based surrogate modeling and multi-fidelity calibration benchmark of Shi et al. (2025) as the second physical prediction task. Each raw clip contains 10 temperature frames on a 32×64 grid, together with a heat-source sequence. For each supervised prediction pair, the context c consists of the first two observed temperature frames, and the target x_0 is one future temperature frame with prediction index $t^* \in \{5, 6, 7, 8, 9\}$. The dataset contains 4566 clips, split into 4109/457 clips for training/testing, yielding 20545/2285 supervised prediction pairs. Unless otherwise stated, the diffusion and IQ-Diff training settings follow the fluid benchmark.

The benchmark provides a physics simulator constructed from the heat-source sequence using a lightweight PDE surrogate and sampled at the same target time. In our use of this benchmark, the PDE prior is used to construct the terminal reward for IQ-Diff, while evaluation is performed against held-out ground-truth temperature fields using MSE, PSNR, and LPIPS. We refer readers to Shi et al. (2025) for the construction of the PDE surrogate and simulator details; LPIPS is the perceptual metric of Zhang et al. (2018). Here we focus on how the simulator prior is converted into the terminal reward used by IQ-Diff.

Thermal Reward Construction For the thermal-process task, the terminal reward is computed from a physics-based PDE prior. Specifically, given the heat-source sequence of a clip, we use a

Table 9: Comparison on 2D fluid dynamics prediction at $s^* = 10$ using the 64×64 higher-fidelity reward oracle for sampling-time guidance (mean $\pm$ std over the test set).

Method	MSE ↓	PSNR ↑	SSIM ↑
DDIM	0.0061 ± 0.0024	22.48 ± 1.83	0.999223 ± 0.000330
SGG	0.0017 ± 0.0006	28.74 ± 2.59	0.999625 ± 0.000389
BoN-16	0.0008 ± 0.0003	32.11 ± 2.32	0.999961 ± 0.000235
TDS-Reward	0.0004 ± 0.0003	34.98 ± 1.98	0.999938 ± 0.000135
Single-Q	0.0004 ± 0.0002	34.77 ± 1.05	0.999955 ± 0.000042
IQ-Diff	$\mathbf{0.0003 \pm 0.0001}$	$\mathbf{36.36 \pm 1.48}$	$\mathbf{0.999978 \pm 0.000019}$

Table 10: Comparison on the fluid dynamics benchmark using the 50-epoch diffusion checkpoint (mean $\pm$ std over the test set).

Method	MSE ↓	PSNR ↑	SSIM ↑
DDIM	0.072 ± 0.022	11.57 ± 1.89	0.834 ± 0.029
SGG	0.055 ± 0.016	13.09 ± 2.03	0.901 ± 0.033
BoN-16	0.016 ± 0.004	17.90 ± 1.72	0.971 ± 0.004
TDS-Reward	0.091 ± 0.033	10.77 ± 1.62	0.791 ± 0.028
Single-Q	0.066 ± 0.009	11.83 ± 0.75	0.937 ± 0.055
IQ-Diff	0.027 ± 0.009	15.99 ± 0.94	0.977 ± 0.008

lightweight heat-source-to-PDE surrogate with lookback window 5 to generate a predicted thermal field at the target time. We denote this simulator prior by z_{pde}. The prior is not used as ground truth; it serves only as an inexpensive physics-based reference for sampling-time guidance.

Given a generated future temperature field x_0 and the corresponding PDE prior z_{pde}, we define the terminal reward as the negative of a weighted physics-consistency loss:

$$R_{\mathrm{thermal}}(x_0; c) = -\Big(\lambda_{\mathrm{loc}} L_{\mathrm{loc}} + \lambda_{\mathrm{out}} L_{\mathrm{out}} + \lambda_{\mathrm{ncc}} L_{\mathrm{ncc}}\Big).$$

Thus, larger reward corresponds to better agreement with the PDE prior. The three loss terms measure complementary aspects of physical consistency: location alignment of the hot region, suppression of spurious heat outside the predicted support, and local structural similarity within the predicted support.

Let $\tilde{x}_0$ and $\tilde{z}_{\mathrm{pde}}$ denote per-sample min–max normalized versions of the generated field and the PDE prior. The location term is

$$L_{\mathrm{loc}} = \big\|\mathrm{COM}_\beta(\tilde{x}_0) - \mathrm{COM}_\beta(\tilde{z}_{\mathrm{pde}})\big\|_2^2,$$

where COM_β is a soft center-of-mass operator. This term encourages the generated high-temperature region to be centered near the simulator-predicted high-temperature region.

To penalize spurious hot regions outside the simulator-predicted support, we construct a binary support mask m_{pde} by thresholding the normalized PDE prior at τ_{pde} and dilating the resulting mask. The outside-support penalty is

$$L_{\mathrm{out}} = \mathrm{mean}\Big(\big[(1 - m_{\mathrm{pde}}) \odot \mathrm{ReLU}(\tilde{x}_0 - \tau_{\mathrm{hot}})\big]^2\Big).$$

This term discourages the generated field from placing high temperature far outside the region suggested by the PDE prior.

Finally, we encourage local structural agreement inside the simulator-predicted support using a normalized cross-correlation loss:

$$L_{\mathrm{ncc}} = 1 - \mathrm{NCC}(G(\tilde{x}_0) \odot m_{\mathrm{pde}}, G(\tilde{z}_{\mathrm{pde}}) \odot m_{\mathrm{pde}}),$$

where $G(\cdot)$ is a 3×3 Gaussian blur. This term compares the local shape of the generated thermal pattern with the PDE prior within the predicted support.

We use

$$(\lambda_{\mathrm{loc}}, \lambda_{\mathrm{out}}, \lambda_{\mathrm{ncc}}) = (1.0, 0.7, 0.5),$$

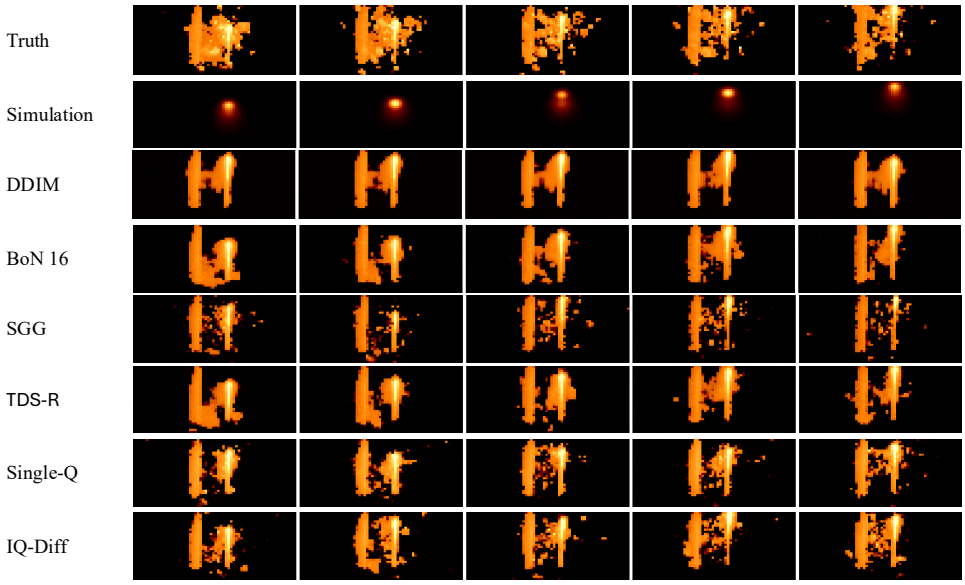

Figure 4: Qualitative comparison on the thermal-process prediction task. The examples illustrate that IQ-Diff better preserves localized high-temperature structures while reducing visible artifacts.

Table 11: Quality–cost comparison on the thermal-process task. Relative time is normalized by DDIM and excludes offline Q-function training.

Method	U-Net calls	Q scores	Online reward evals	Rel. time ↓	LPIPS ↓
DDIM	H	0	0	1.00×	0.384 ± 0.087
BoN-16	$16H$	0	16	5.68×	0.362 ± 0.101
TDS-Reward	PH	0	PH	20.36×	0.201 ± 0.074
Single-Q	H	NH	0	1.56×	0.177 ± 0.071
IQ-Diff	H	NKH	0	2.32×	$\mathbf{0.139 \pm 0.036}$

$\tau_{\text{pde}} = 0.15$, $\tau_{\text{hot}} = 0.08$, dilation kernel size 7, and softmax temperature $\beta = 12$ for the soft center-of-mass operator. The reward is used only for sampling-time guidance; evaluation is performed against held-out ground-truth temperature fields using MSE, PSNR, and LPIPS.

Thermal Qualitative Results Fig. 4 shows representative qualitative comparisons on the thermal-process task. IQ-Diff produces sharper and more localized thermal structures than the unguided sampler and better aligns with the high-temperature regions of the ground truth.

Thermal Quality–Cost Comparison Table 11 reports the quality–cost comparison on the thermal-process task. We use the same cost accounting protocol as in the fluid benchmark: relative time is normalized by DDIM, Q scores count scalar candidate scores, and online reward evaluations count test-time reward calls used for selection or particle weighting.