Trajectory-Consistent Diffusion Policies for Offline Reinforcement Learning

Yichao Fu[*]
Zhejiang University
fuyichao@zju.edu.cn

Shangde Gao[*§]
Zhejiang University
gaosde@zju.edu.cn

Zhuoling Li
Lancaster University
z.li81@lancaster.ac.uk

Wen Wang
Zhejiang University
wwenxyz@zju.edu.cn

Shangqi Gao
University of Cambridge
sg2162@cam.ac.uk

Ke Liu[†]
Zhejiang University
lk2017@zju.edu.cn

Abstract

Diffusion policies have emerged as a powerful policy class for offline reinforcement learning (RL) due to their ability to model complex, multimodal action distributions through iterative denoising. However, in offline RL, expressivity alone is not enough. Generated actions are supposed to remain within value-reliable regions of the action space to avoid extrapolation error. We find that standard diffusion policies often produce unstable denoising trajectories whose intermediate and final action candidates drift toward weakly supported regions, making critic-guided action selection unreliable and degrading teacher quality for one-step distillation. This issue is particularly pronounced on mixed-quality offline datasets, where suboptimal behaviors amplify trajectory instability. To address this problem, we propose *T*rajectory *C*onsistency *R*ectification (*TCR*), a framework designed to promote denoising stability during both training and inference. At inference time, TCR aggregates locally consistent, high-value action candidates across denoising steps to suppress critic-unreliable outliers. During training, it adaptively reweights denoising steps to improve trajectory-level coherence. We show that this stabilization provides a more reliable teacher for one-step flow-matching distillation. Extensive experiments on D4RL and OGBench show that TCR consistently improves over diffusion-policy baselines, with significant gains in sparse-reward domains like AntMaze and Adroit. Moreover, one-step policies distilled from TCR achieve competitive performance among efficient single-step methods. These results suggest that enhancing the temporal consistency of denoising trajectories is a key ingredient for robust diffusion policy in offline reinforcement learning. Code is available at: https://github.com/yichaofu/TCR-Diffusion.

1 Introduction

Offline reinforcement learning (RL) aims to learn policies from fixed datasets without additional interaction with the environment, making it particularly attractive in domains where online exploration is costly, unsafe, or impractical [Lange et al., 2012, Prudencio et al., 2023, Oh et al., 2025]. A central challenge in offline RL is that policy improvement must be achieved under limited data support: actions that deviate excessively from the behavior distribution can receive unreliable value estimates

[*]Equal contribution.
[†]Corresponding author.
[§]Shangde Gao is affiliated with the Second Affiliated Hospital of Zhejiang University School of Medicine; Transvascular Implantation Devices Research Institute and State Key Laboratory of Transvascular Implantation Devices.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

due to extrapolation error, leading to poor decision making [Li et al., 2023, Zhu et al., 2023]. Recently, diffusion policies have emerged as a powerful policy class for offline RL, since iterative denoising provides a flexible mechanism for modeling complex and multimodal action distributions [Kang et al., 2023, Venkatraman et al., 2024, Zhang et al., 2024, Chi et al., 2025]. This expressive formulation has led to strong empirical performance in continuous control and robotic manipulation, where simple unimodal policies often struggle to capture heterogeneous behaviors.

However, in offline RL, expressive action generation alone is not sufficient. Generated actions are supposed to lie in regions where value estimates are reliable [Bai et al., 2025]. Standard diffusion-policy training optimizes denoising accuracy at each timestep independently, but does not explicitly constrain the full denoising trajectory to remain within value-reliable regions. As a result, the reverse diffusion process may produce unstable action candidates that drift toward weakly supported regions, where critic estimates become unreliable due to extrapolation error [Wang et al., 2024]. This issue is particularly pronounced on mixed-quality offline datasets: unstable denoising trajectories can corrupt final action execution and weaken the quality of teacher policies used for one-step distillation. The concern, therefore, is not whether intermediate denoising steps improve monotonically, but whether the denoising trajectory is sufficiently stable and consistent to yield reliable final decisions. Unlike previous methods that use the critic only for final action selection or re-ranking, we argue that the iterative nature of diffusion allows for a more holistic stabilization of the entire denoising path [Hansen-Estruch et al., 2023, Chen et al., 2024a].

To address this problem, we propose *T*rajectory *C*onsistency *R*ectification (*TCR*), a framework that stabilizes diffusion policies during both training and inference. The key idea is to *exploit trajectory-level structure in the reverse diffusion chain rather than relying solely on the final denoising step.* At inference time, TCR aggregates locally consistent, high-value action candidates across denoising steps to suppress critic-unreliable outliers. By requiring both high critic values and local temporal agreement, this mechanism effectively filters out candidate actions that might otherwise exploit critic error, thereby promoting more support-consistent behavior without requiring explicit density modeling. During training, TCR adaptively reweights denoising steps, encouraging the policy to focus on more reliable trajectory segments. In this way, TCR improves not only the quality of the diffusion policy itself, but also the quality of the supervision it provides as a teacher for one-step flow-matching distillation.

We evaluate TCR on standard offline RL benchmarks, including D4RL and OGBench. Across these benchmarks, TCR consistently improves over standard diffusion-policy baselines, with particularly strong gains on sparse-reward and mixed-quality domains such as AntMaze and Adroit. Moreover, one-step policies distilled from TCR achieve state-of-the-art performance among efficient single-step methods, showing that trajectory stabilization also improves teacher quality for downstream compression. Together, these results suggest that *enhancing the temporal and value-consistency of denoising trajectories* is a key ingredient for robust and deployable offline diffusion reinforcement learning. Our main contributions are summarized as follows:

- We revisit the failure mode of diffusion policies in offline RL and argue that the key issue is not merely non-monotonic intermediate behavior but the generation of weakly supported and critic-unreliable action candidates along denoising trajectories.
- We propose **Trajectory Consistency Rectification (TCR)**, a framework that stabilizes diffusion policies through trajectory-level aggregation at inference and adaptive denoising-step rectification during training.
- We show that stabilizing denoising trajectories produces a stronger teacher for one-step flow-matching distillation, improving both policy efficiency and final performance.
- We demonstrate on D4RL and OGBench that TCR consistently outperforms standard diffusion-policy baselines and achieves state-of-the-art performance after distillation, especially on mixed-quality and sparse-reward tasks.

2 Related Work

2.1 Diffusion Policies for Offline RL

Recent advances have transitioned offline RL from traditional parametric policies to conditional score-based generative modeling [Kang et al., 2023, Berrueta et al., 2024, Zhang et al., 2025]. By

recasting decision-making as a denoising process, frameworks such as Trajectory Diffuser [Janner et al., 2022], Decision Diffuser [Ajay et al., 2023], and Diffusion Policy [Chi et al., 2025] leverage iterative refinement to capture complex multimodal action distributions, effectively mitigating the mode collapse prevalent in Gaussian-based baselines. This empirical success is increasingly supported by theoretical analyses on sample complexity, convergence, and the advantages of score-matching objectives [Wang et al., 2023, Sun et al., 2025]. However, existing methods primarily focus on improving the quality of the final generated action through architectural or conditioning design [Bai et al., 2025], while leaving the denoising trajectory itself largely unconstrained. In offline RL, such unconstrained denoising can produce weakly supported action candidates that are difficult for the critic to evaluate them reliably. Unlike these approaches that primarily regularize the objective or architecture, TCR shifts the focus to the internal dynamics of the reverse diffusion process, ensuring that intermediate states are not just noise transitions but are value-reliable.

2.2 Policy Distillation and One-Step Policy Learning

To bridge the gap between the strong expressivity of diffusion policies and the real-time requirements of robotic control, several works have proposed distilling multi-step denoising into efficient one-step policies. OneDP [Wang et al., 2025] and Consistency Policy [Prasad et al., 2024] represent foundational efforts in this direction, respectively leveraging KL-divergence minimization along the diffusion chain and self-consistency constraints to achieve substantial inference acceleration. CED [Zhu et al., 2025] further introduces an energy-based distillation framework, while Flow Q-learning (FQL) [Park et al., 2025] trains an expressive one-step RL policy that avoids recursive backpropagation through the sampling process. Despite these advances, such methods largely assume that the diffusion teacher provides stable supervision. Our work challenges this by showing that denoising instability in teachers directly degrades distillation. By using TCR to refine the teacher trajectory, we provide a more consistent target that enables the distilled policy to avoid suboptimal modes captured by standard diffusion.

2.3 Consistency and Stability in Iterative Models

Consistency is a central principle in iterative generation and sequential refinement. In robotics, Diffusion Policy [Chi et al., 2025] employs stochastic Langevin dynamics to model complex action manifolds, though at the cost of high inference latency. To improve efficiency, Consistency Policy and related methods enforce self-consistency across different noise levels or solver trajectories [Song et al., 2023, Ding and Jin, 2024, Prasad et al., 2024]. However, these approaches mainly emphasize generative efficiency or smoothness. Our work differs by introducing value-aware consistency, where stability is defined not only by the model's self-agreement but also by its alignment with high-value regions supported by the offline dataset. This directly addresses the extrapolation error unique to the offline RL setting.

3 Method

3.1 Problem Setup and Notation

Offline reinforcement learning setting. We consider offline reinforcement learning from a fixed dataset $\mathcal{D} = \{(s_i, \mathbf{a}_{0,i})\}_{i=1}^{N}$, where $s \in \mathcal{S}$ denotes the state or observation, and $\mathbf{a}_0 = (a_0, a_1, \ldots, a_{H-1}) \in \mathbb{R}^{H\times d}$ denotes a sequence of H continuous actions. We define the vectorized representation $\mathbf{A}_0 := \mathrm{vec}(\mathbf{a}_0) \in \mathbb{R}^D$, where $D = Hd$. The goal of diffusion policy learning is to model the conditional policy $\pi(\mathbf{A}_0 \mid s)$ and generate actions by sampling from this distribution at inference time.

Forward diffusion (noising) process. Diffusion models introduce a sequence of latent variables $\{\mathbf{A}_t\}_{t=0}^{T}$ with $\mathbf{A}_0$ being the clean action chunk, and define a Markov forward process that gradually injects Gaussian noise:

$$q(\mathbf{A}_t \mid \mathbf{A}_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\,\mathbf{A}_{t-1},\, (1-\alpha_t)I_D)\,, \quad t = 1, \ldots, T.$$

Here, $\{\alpha_t\}_{t=1}^{T} \subset (0,1)$ is a predefined variance schedule and I_D is the $D \times D$ identity matrix. Let $\bar{\alpha}_t = \prod_{k=1}^{t} \alpha_k$. Marginalizing the forward process yields the closed-form distribution:

$$q(\mathbf{A}_t \mid \mathbf{A}_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\,\mathbf{A}_0,\, (1-\bar{\alpha}_t)I_D\big)\,, \tag{1}$$

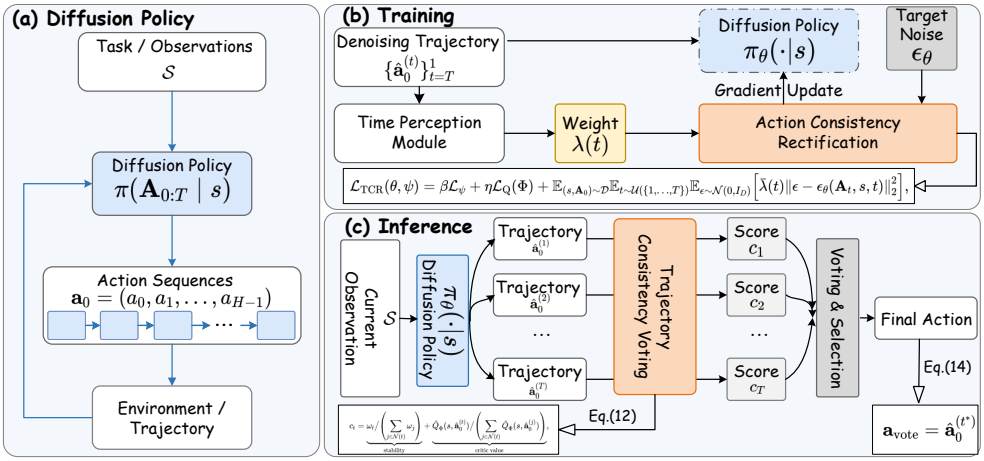


Figure 1: Temporal Consistency Rectification framework Overview. Our framework consists of two main components: The action consistency rectification module for training and the temporal consistency voting module for inference.

or equivalently,

$$\mathbf{A}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{A}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon, \quad \epsilon \sim \mathcal{N}(0, I_D). \tag{2}$$

Reverse process (denoising) conditioned on state. The diffusion policy defines a learned reverse-time Markov chain conditioned on s:

$$\pi_\theta(\mathbf{A}_{0:T} \mid s) := \pi(\mathbf{A}_T) \prod_{t=1}^{T} \pi_\theta(\mathbf{A}_{t-1} \mid \mathbf{A}_t, s), \tag{3}$$

with a standard Gaussian prior $\pi(\mathbf{A}_T) = \mathcal{N}(0, I_D)$. Each reverse transition is modeled as a Gaussian:

$$\pi_\theta(\mathbf{A}_{t-1} \mid \mathbf{A}_t, s) = \mathcal{N}(\mu_\theta(\mathbf{A}_t, s, t),\, \Sigma_t)\,, \quad \Sigma_t = \sigma_t^2 I_D.$$

where μ_θ is a neural network parameterized by θ and σ_t^2 is determined by the noise schedule (or learned).

Noise-prediction parameterization. We adopt the standard ϵ-prediction parameterization, in which the model predicts the injected noise $\epsilon_\theta = \epsilon_\theta(\mathbf{A}_t, s, t)$. Then an estimate of the clean action chunk is recovered as

$$\hat{\mathbf{A}}_0(\mathbf{A}_t, s, t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{A}_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(\mathbf{A}_t, s, t)\right). \tag{4}$$

Under this parameterization, the reverse-process mean can be written as

$$\mu_\theta(\mathbf{A}_t, s, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{A}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(\mathbf{A}_t, s, t)\right). \tag{5}$$

Training objective. Diffusion policies are trained to maximize the conditional log-likelihood $\log p_\theta(\mathbf{A}_0 \mid s)$ via a variational lower bound. In practice, this results in the denoising score-matching objective:

$$\mathcal{L}_{\mathrm{DP}}(\theta) = \mathbb{E}_{(s,\mathbf{A}_0)\sim\mathcal{D}}\,\mathbb{E}_{t\sim\mathcal{U}(\{1,\ldots,T\})}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I_D)}\left[\|\epsilon - \epsilon_\theta(\mathbf{A}_t, s, t)\|_2^2\right], \tag{6}$$

where $\mathbf{A}_t$ is obtained using Eq. (2). Optionally, timestep weights $\lambda(t)$ can be incorporated:

$$\mathcal{L}_{\mathrm{DP}}(\theta) = \mathbb{E}_{(s,\mathbf{A}_0),t,\epsilon}\left[\lambda(t)\,\|\epsilon - \epsilon_\theta(\mathbf{A}_t, s, t)\|_2^2\right]. \tag{7}$$

Sampling and policy execution. At inference time, given a state s, the diffusion policy samples from $\pi_\theta(\cdot \mid s)$ by running the reverse chain. We initialize $\mathbf{A}_T \sim \mathcal{N}(0, I_D)$, and for $t = T, T-1, \ldots, 1$, sample

$$\mathbf{A}_{t-1} = \mu_\theta(\mathbf{A}_t, s, t) + \sigma_t z, \qquad z \sim \mathcal{N}(0, I_D).$$

The final sample $\mathbf{A}_0$ is reshaped into $\mathbf{a}_0 = \mathrm{unvec}(\mathbf{A}_0)$. In receding-horizon control, only the first action a_0 is executed, and the policy replans at the next environment step.

Problem Definition. Given denoising trajectories $\{\mathbf{A}_t\}_{t=T}^0$ generated by the diffusion policy for a state s, we define the intermediate decoded action estimates and denoising-trajectory instability as:

Definition 3.1 (Intermediate Decoded Actions)**.** Conditioned on the state s and the current noisy sample $\mathbf{A}_t$ at diffusion timestep t, we define the *intermediate decoded action estimate* $\hat{\mathbf{A}}_0^{(t)}$ as the clean action reconstructed directly from $\mathbf{A}_t$ via the noise predictor ϵ_θ:

$$\hat{\mathbf{A}}_0^{(t)} := \hat{\mathbf{A}}_0(\mathbf{A}_t, s, t) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{A}_t - \sqrt{1-\bar{\alpha}_t}\, \epsilon_\theta(\mathbf{A}_t, s, t) \right), \quad t = 1, \ldots, T.$$

The corresponding action chunks are denoted by $\hat{\mathbf{a}}_0^{(t)} = \mathrm{unvec}(\hat{\mathbf{A}}_0^{(t)})$. Collectively, $\{\hat{\mathbf{a}}_0^{(t)}\}_{t=1}^T$ forms a denoising-time trajectory of action predictions.

Definition 3.2 (Denoising-Trajectory Instability)**.** Given a state s, let $\{\hat{\mathbf{a}}_0^{(t)}\}_{t=1}^T$ be the sequence of intermediate decoded actions. We define *denoising-trajectory instability* as the variance in action quality and support across denoising steps, formally characterized by the fluctuation $\mathcal{V} = \frac{1}{T-1} \sum_{t=1}^{T-1} \|\hat{\mathbf{a}}_0^{(t+1)} - \hat{\mathbf{a}}_0^{(t)}\|^2$ and the sensitivity of $Q_\Phi(s, \hat{\mathbf{a}}_0^{(t)})$ to the timestep t. This phenomenon causes action candidates to vary substantially in their local consistency, making final selection sensitive to denoising dynamics.

In offline RL, this is particularly problematic when actions drift toward weakly supported regions where critic estimates can become unreliable. Therefore, our goal is therefore not to enforce strict monotonic improvement across all denoising steps, but to *stabilize the denoising trajectory to ensure that the resulting action candidates remain within behaviorally-consistent and value-reliable regions.* By suppressing fluctuations in the action sequence, we aim to mitigate the risk of the policy exploiting unreliable critic estimates in weakly-supported regions.

3.2 Trajectory Consistency Rectification

To mitigate the denoising-trajectory instability defined in Definition 3.2, we propose *T*rajectory *C*onsistency *R*ectification (*TCR*), as shown in Fig. 1. TCR stabilizes the denoising trajectory $\{\hat{\mathbf{a}}_0^{(t)}\}_{t=1}^T$ generated by the diffusion policy through two complementary components: (i) *Trajectory Consistency Voting*, which aggregates locally stable and high-value predictions from multiple timesteps at inference time; and (ii) *Action Consistency Rectification*, which adaptively reweights the training loss toward more reliable denoising steps.

3.2.1 Action Consistency Rectification

To improve denoising stability during training, we introduce an *Action Consistency Rectification* module that reweights the diffusion loss at each timestep. Specifically, we modify the timestep weights $\lambda(t)$ in Eq. (7) so that the model focuses more on reliable denoising steps. Concretely, we estimate the weights with a small neural network $f_\psi : \mathbb{R}^D \times \mathcal{S} \times \{1, \ldots, T\} \to \mathbb{R}^+$, parameterized by ψ, which takes as input the noisy action $\mathbf{A}_t$, state s, and timestep t, and outputs a positive weight:

$$\lambda(t) = f_\psi(\mathbf{A}_t, s, t). \tag{8}$$

To avoid trivial solutions, *i.e.*assigning zero weights to all timesteps, we normalize the weights within each training batch:

$$\bar{\lambda}(t) = \frac{\lambda(t)}{\sum_{j=1}^B \lambda(j)}, \tag{9}$$

where B is the batch size. We additionally introduce a regularization term

$$\mathcal{L}_\psi = -\mathbb{E}_{(s,\mathbf{A}_0),t,\epsilon} \left[\log \lambda(t) \right], \tag{10}$$

which prevents weight collapse and encourages informative timestep reweighting.

We further introduce a value function $Q_\Phi(s, a)$, parameterized by Φ, to estimate action quality. The critic is trained on the offline dataset $\mathcal{D}$ using the standard Bellman error:

$$\mathcal{L}_{\mathrm{Q}}(\Phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left(r + \gamma \max_{a'} Q_{\bar{\Phi}}(s', a') - Q_\Phi(s, a)\right)^2,$$

where $\bar{\Phi}$ denotes the parameters of a target network updated via exponential moving average.

The overall training objective for the diffusion policy with TCR is

$$\mathcal{L}_{\mathrm{TCR}}(\theta, \psi) = \beta\mathcal{L}_\psi + \eta\mathcal{L}_{\mathrm{Q}}(\Phi) + \mathbb{E}_{(s,\mathbf{A}_0)\sim\mathcal{D}}\mathbb{E}_{t\sim\mathcal{U}(\{1,\ldots,T\})}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I_D)}\left[\bar{\lambda}(t)\,\|\epsilon - \epsilon_\theta(\mathbf{A}_t, s, t)\|_2^2\right],$$

where $\beta > 0$ and $\eta > 0$ are hyperparameters. This rectification mechanism reduces the influence of unstable denoising steps and encourages the learned policy to follow more support-consistent trajectory segments.

3.2.2 Trajectory Consistency Voting

The core intuition of TCR is that a reliable diffusion process should produce locally consistent predictions of $\mathbf{a}_0$ across neighboring denoising steps. We therefore measure the reliability of the intermediate prediction $\hat{\mathbf{a}}_0^{(t)}$ by its agreement with predictions in a local trajectory neighborhood. Let

$$\mathcal{N}(t) = \{j \in \{1, \ldots, T\} \mid |t - j| \le k\}$$

denote a trajectory window of size $2k + 1$ centered at t. We compute a non-normalized consistency score ω_t for each timestep:

$$\omega_t = \sum_{j\in\mathcal{N}(t)} \mathcal{K}\left(\hat{\mathbf{a}}_0^{(t)}, \hat{\mathbf{a}}_0^{(j)}\right) \cdot w_t, \tag{11}$$

where $\mathcal{K}(\mathbf{u}, \mathbf{v}) = \exp(-\|\mathbf{u} - \mathbf{v}\|_2^2/\gamma^2)$ is a Gaussian kernel with bandwidth γ. A high ω_t indicates that the prediction at step t is stable and agrees with its neighboring denoising steps. Here w_t denotes a trajectory weight used to emphasize certain timesteps, e.g., later steps that are closer to the final output:

$$w_t = \frac{\mathcal{G}(t)}{\sum_{j\in\mathcal{N}(t)} \mathcal{G}(j)},$$

where $\mathcal{G}(t) = \exp\left(\alpha \cdot \frac{t}{T-1}\right)$ is an exponential function with temperature parameter α.

Crucially, to combine local consistency with action quality, we use a normalized critic value to avoid issues with task-specific reward scales or negative values. The final score c_t for timestep t is defined as:

$$c_t = \frac{\omega_t}{\sum_{j\in\mathcal{N}(t)} \omega_j} + \frac{\hat{Q}_\Phi(s, \hat{\mathbf{a}}_0^{(t)})}{\sum_{j\in\mathcal{N}(t)} \hat{Q}_\Phi(s, \hat{\mathbf{a}}_0^{(j)})}, \tag{12}$$

where $\hat{Q}_\Phi(s, a)$ represents the critic value after a min-max normalization over the candidates in the current trajectory:

$$\hat{Q}_\Phi(s, a) = \frac{Q_\Phi(s, a) - \min_j Q_\Phi(s, \hat{\mathbf{a}}_0^{(j)})}{\max_j Q_\Phi(s, \hat{\mathbf{a}}_0^{(j)}) - \min_j Q_\Phi(s, \hat{\mathbf{a}}_0^{(j)}) + \delta_{norm}}. \tag{13}$$

where $\delta_{norm} > 0$ is a small constant for numerical stability. The normalized aggregation ensures that the voting process is robust to the absolute magnitude of Q and focuses on relative action quality. Finally, the trajectory-voted action is

$$\mathbf{a}_{\text{vote}} = \hat{\mathbf{a}}_0^{(t^*)}, \quad \text{where} \quad t^* = \arg\max_{t=1,\ldots,T} c_t. \tag{14}$$

The voting mechanism suppresses critic-unreliable outliers caused by stochastic denoising and favors action candidates that are both locally stable and highly valued.

3.3 Distillation to One-Step Policies

To further improve inference efficiency, we distill the TCR-enhanced diffusion policy into a one-step flow-matching policy. Previous works have shown that flow-matching policies can effectively approximate diffusion policies while enabling single-step action generation. However, the instability of standard diffusion policies often leads to suboptimal distillation targets. By first stabilizing the diffusion policy with TCR, we obtain a more reliable teacher for one-step policy learning.

To distill the TCR diffusion policy into the flow-matching policy, we generate high-quality action targets using the trajectory-voted action $\mathbf{a}_{\text{vote}}$ from Section 3.2.2. Flow-matching policies model the conditional distribution $p_\phi(\mathbf{A}_0 \mid s)$ directly, allowing for single-step action generation without iterative denoising. We parameterize the flow-matching policy with a neural network ϕ and train it to minimize

$$\mathcal{L}_{\text{FM}}(\phi) = \mathbb{E}_{(s,\mathbf{A}_0)\sim\mathcal{D},\, z\sim\mathcal{N}(0,I_D)} \left[\|\mathbf{A}_{\text{vote}} - \mu_\phi(s,z)\|_2^2 \right],$$

where $\mathbf{A}_{\text{vote}} := \text{vec}(\mathbf{a}_{\text{vote}})$ and $\mu_\phi(s,z)$ is the flow-matching model's prediction given state s and noise $z \sim \mathcal{N}(0, I_D)$. In this way, the student policy learns from cleaner and more stable supervision, which leads to better one-step policy performance.

Theoretical analysis of TCR as support regularization in offline RL is provided in Appendix B.

4 Experiments

We conduct extensive experiments on D4RL and OGBench to evaluate the effectiveness of TCR in offline RL. Our evaluation focuses on three aspects: overall diffusion-policy performance, teacher quality for one-step distillation, and generalization across diverse datasets and environments. Across both benchmarks, TCR consistently improves over standard diffusion-policy baselines, with particularly strong gains on sparse-reward and mixed-quality tasks such as AntMaze and Adroit.

4.1 Experimental Setup

Datasets. We mainly evaluate our method on the *D4RL* benchmark [Fu et al., 2020] and OGBench [Park et al., 2024], which cover diverse offline RL settings with varying data quality, reward sparsity, and action complexity.

Evaluation metrics. We report the normalized score for D4RL, where 100 indicates expert-level performance, while a score of 0 corresponds to random behavior. For OGBench, we follow the standard benchmark protocol and report the corresponding task performance scores.

Compared approaches. We compare TCR diffusion policies and distilled one-step policies against three classes of baselines: *standard offline RL methods*, including CQL [Kumar et al., 2020], IQL [Kostrikov et al., 2022], and TD3+BC [Fujimoto and Gu, 2021]; *transformer-based methods*, including DT [Chen et al., 2021], GDT [Hu et al., 2023], and QT [Hu et al., 2024]; *generative-policy methods*, including DD [Ajay et al., 2023], IDQL [Hansen-Estruch et al., 2023], SRPO [Chen et al., 2024b], FQL [Park et al., 2025], and standard Diffusion Policy (DP) [Chi et al., 2025].

4.2 Main Results on Offline RL Benchmarks

4.2.1 Comparison with Previous Methods

TCR consistently achieves superior performance over standard diffusion policies and strong offline RL baselines across different tasks and datasets as shown in Table 1. In particular, our method achieves a Gym average of 94.3, compared with 63.0 for standard Diffusion Policy, 62.0 for FQL, and 87.1 for SRPO. On Adroit, TCR reaches an average score of 54.6, outperforming DP (44.2), FQL (42.5), and QT (48.8). On AntMaze, TCR obtains an average of 94.5, substantially improving over DP (31.0), IDQL (73.9), SRPO (77.0), and FQL (84.8). These results show that stabilizing denoising trajectories provides a robust benefit beyond architectural expressivity alone, especially in domains where support mismatch and critic unreliability are severe.

Table 1: **Normalized scores on D4RL benchmarks.** Normalized scores across standard, transformer-based, and generative policies. Best results are highlighted in **bold**.

	Standard			Transformer			Generative					
Task	**CQL**	**IQL**	**TD3+BC**	**DT**	**GDT**	**QT**	**DD**	**IDQL**	**SRPO**	**FQL**	**DP**	**Ours**
Gym Locomotion Tasks												
Medium-Expert												
HalfCheetah	91.6	86.7	90.7	86.8	93.2	93.2	90.6	94.4	92.2	91.5	73.2	**110.1**
Hopper	105.4	91.5	98.0	107.6	111.1	113.0	111.8	105.3	100.1	73.6	92.7	**113.4**
Walker2d	108.8	109.6	110.1	108.1	107.7	112.0	108.8	111.6	**114.0**	101.4	108.5	**114.0**
Medium												
HalfCheetah	49.2	47.4	48.4	42.6	42.9	51.0	49.1	49.7	60.4	58.6	42.5	**61.0**
Hopper	69.4	66.3	59.3	67.6	77.1	99.6	79.3	63.1	95.5	55.0	57.5	**103.3**
Walker2d	83.0	78.3	83.7	74.0	76.5	87.2	82.5	80.2	84.4	43.0	71.1	**90.0**
Medium-Replay												
HalfCheetah	45.5	44.2	44.6	36.6	40.5	48.8	39.3	45.1	51.4	51.9	38.9	**52.7**
Hopper	95.0	94.7	60.9	82.7	85.3	102.1	100.0	82.4	101.2	40.0	35.9	**103.3**
Walker2d	77.2	73.9	81.8	79.4	77.5	97.8	75.0	79.8	84.6	43.0	46.6	**100.7**
Gym Average	80.6	77.0	75.3	76.2	79.1	89.4	81.8	79.1	87.1	62.0	63.0	**94.3**
Adroit Tasks												
Pen-cloned	39.2	37.3	26.5	75.8	86.2	85.8	42.8	64.0	61.0	74.0	87.6	**94.0**
Hammer-cloned	2.1	2.1	0.3	3.0	8.9	11.8	1.7	2.0	2.0	11.0	0.7	**15.1**
Adroit Average	20.7	19.7	13.4	39.4	47.6	48.8	22.3	33.0	31.5	42.5	44.2	**54.6**
AntMaze Tasks												
AntMaze-U	74.0	87.5	78.6	59.2	76.0	76.0	73.1	93.8	97.1	96.0	62.0	**100.0**
AntMaze-U-D	84.0	62.2	71.4	53.0	69.0	69.0	49.2	62.0	82.1	89.0	62.0	**98.0**
AntMaze-M-D	53.7	70.0	3.0	0.0	6.0	0.0	24.6	83.5	75.0	71.0	0.0	**86.0**
AntMaze-L-D	14.9	47.5	0.0	0.0	0.0	0.0	7.5	56.4	53.6	83.0	0.0	**94.0**
AntMaze Average	56.6	66.8	38.3	28.1	37.8	36.3	38.6	73.9	77.0	84.8	31.0	**94.5**

Comparison with diffusion policies. TCR substantially improves diffusion-based policies by stabilizing denoising trajectories and producing more reliable action candidates during inference. On Gym locomotion tasks, our method achieves 110.1, 113.4, and 114.0 on the three medium-expert tasks, and further reaches 61.0, 103.3, and 90.0 on the medium tasks. On the more difficult medium-replay tasks, TCR still improves over DP by a large margin, e.g., from 38.9 to 52.7 on HalfCheetah, from 35.9 to 103.3 on Hopper, and from 46.6 to 100.7 on Walker2d. Similar gains are observed on AntMaze, where TCR achieves 100.0, 98.0, 86.0, and 94.0 on the reported tasks, and on Adroit, where it improves substantially over standard diffusion baselines in both cloned and human settings.

Comparison with one-step methods. The distilled one-step flow-matching policies derived from TCR-enhanced diffusion teachers also exhibit strong performance. As shown in Table 1, the final one-step policies achieve state-of-the-art results compared with strong efficient-policy baselines such as QT and FQL (The *Ours* column denotes the final distilled one-step policy derived from the TCR-enhanced diffusion teacher). This is especially clear on challenging tasks where the quality of teacher supervision is critical. These results indicate that stabilizing the diffusion teacher with TCR not only improves the multi-step diffusion policy itself, but also enhances downstream one-step policy learning by providing cleaner and more reliable supervision.

For more results on additional tasks, please refer to Appendix D.

4.2.2 Generalization of TCR

To further validate the robustness and generalization capability of TCR, we evaluate it on *OGBench* [Park et al., 2024], which contains a diverse collection of tasks with complex dynamics. As shown in Table 2, our method consistently outperforms diffusion-policy baselines across navigation, manipulation, and puzzle tasks.

Table 2: **Comparison on OGBench benchmarks.** Best and second-best results are highlighted in bold and underlined, respectively.

	Diffusion Baselines				Proposed
Task Name	**IDQL**	**SRPO**	**CAC**	**DP**	**Ours**
Maze Navigation					
AntMaze (Large)	0	0	<u>42</u>	2	**96**
AntMaze (Giant)	0	0	0	0	**12**
HumanoidMaze (Med)	1	0	**38**	2	<u>34</u>
HumanoidMaze (Large)	0	0	<u>1</u>	0	**2**
Manipulation & Scene					
Cube (Single)	<u>96</u>	82	80	70	**99**
Cube (Double)	7	**16**	0	0	<u>10</u>
Scene (Single)	33	2	<u>50</u>	4	**88**
High-Dim Puzzles					
Puzzle (3×3)	<u>2</u>	0	0	0	**36**
Puzzle (4×4)	<u>26</u>	7	1	2	**40**

For example, on the reported single-task subset, TCR improves over DP from 2 to 96 on AntMaze (Large), from 2 to 34 on HumanoidMaze (Medium), from 70 to 99 on Cube (Single), from 4 to 88 on Scene (Single), and from 0 to 36 and 2 to 40 on Puzzle (3×3) and (4×4), respectively.

These gains show that the benefits of TCR are not tied to a specific benchmark, but reflect a broader principle: improving denoising-trajectory stability leads to more support-consistent and value-reliable policy generation.

4.2.3 Ablation Study

To isolate the contributions of the two components in TCR, we conduct a detailed ablation study on Adroit, as shown in Table 3. Both Consistency Voting (CV) and Consistency Rectification (CR) individually improve over the standard Diffusion Policy baseline, while their combination yields the strongest teacher performance. Averaged across Adroit tasks, DP obtains 50.8, DP+CV improves to 53.0, DP+CR to 52.5, and the full DP+TCR to 55.9. This shows that inference-time aggregation and training-time rectification are complementary rather than redundant.

A particularly notable finding is that the final distilled one-step policy (*Ours*) frequently outperforms its teacher (DP+TCR). This suggests that the distillation process is not merely compressing the teacher, but can further refine it by learning from trajectory-voted targets that filter out unstable and low-quality denoising outputs.

4.3 Stability and Efficiency Analysis

We further analyze TCR from two practical perspectives: denoising-trajectory stability and inference efficiency. Across denoising steps, standard Diffusion Policy exhibits substantial fluctuation in success rate, indicating that intermediate decoded actions can vary significantly in quality along the reverse diffusion chain. In contrast, TCR yields a more stable trajectory with both higher success rates and smaller fluctuations over a broader range of denoising steps. Quantitatively, the standard deviation of success rates over the last 50 denoising steps decreases from 0.1157 for standard DP to 0.0884 for DP+TCR, showing that TCR effectively suppresses large performance fluctuations without requiring strict monotonic improvement at every step.

We also evaluate the computational overhead of TCR. The batch inference time changes only from 8.467 ms for DP to 8.665 ms for DP+TCR at batch size 256, indicating only a marginal overhead. In contrast, the distilled one-step student requires only 0.208 ms per batch, achieving about $40.72\times$ speedup over the original diffusion policy. These results suggest that TCR improves stability and policy quality while maintaining practical inference efficiency.

Table 3: **Normalized scores on Adroit benchmarks.** Comparison with baseline methods and ablations on Human, Cloned, and Expert tasks. Best and second-best results are highlighted in **bold** and underlined, respectively.

	Baselines				Ablation & Ours				
Task	**IDQL**	**SRPO**	**CAC**	**FQL**	**DP**	**DP+CV**	**DP+CR**	**DP+TCR**	**Ours**
Adroit Tasks: Human									
Pen	76.0	69.0	64.0	53.0	77.95	81.79	82.84	101.73	**101.88**
Hammer	1.0	1.0	2.0	3.0	2.39	2.56	6.17	3.25	**6.25**
Door	6.0	3.0	5.0	0.0	2.61	3.69	3.64	4.26	**6.27**
Relocate	0.0	0.0	0.0	0.0	0.31	0.32	0.13	0.54	**0.68**
Adroit Tasks: Cloned									
Pen	64.0	61.0	56.0	74.0	69.72	79.17	71.22	80.13	**94.05**
Hammer	2.0	2.0	1.0	11.0	0.66	2.69	0.55	2.99	**15.12**
Door	0.0	**3.0**	2.0	0.0	1.90	0.37	0.72	1.12	2.89
Relocate	0.0	0.0	0.0	0.0	0.02	0.14	0.15	0.17	**0.22**
Adroit Tasks: Expert									
Pen	**152.0**	140.0	103.0	142.0	119.23	126.30	125.77	133.94	147.85
Hammer	125.0	127.0	92.0	125.0	123.93	125.19	125.23	126.71	**130.25**
Door	105.0	105.0	98.0	104.0	104.91	105.39	105.29	105.90	**106.15**
Relocate	107.0	106.0	93.0	107.0	105.51	108.42	108.28	110.05	**113.15**
Average	53.2	51.4	43.0	51.6	50.8	53.0	52.5	55.9	**60.4**

5 Conclusion

In this work, we investigate denoising-trajectory instability in diffusion policies for offline RL. Rather than requiring monotonic improvement, we propose *Trajectory Consistency Rectification* (TCR) to enhance the temporal consistency and value-reliability of the full denoising path. TCR stabilizes diffusion policies through adaptive loss rectification during training and trajectory-level consistency voting at inference. Experiments on D4RL and OGBench demonstrate that TCR significantly outperforms standard diffusion baselines, particularly in sparse-reward and mixed-quality tasks. Furthermore, we show that TCR provides high-quality supervision for one-step flow-matching distillation, where the student can purify the teacher trajectory to achieve superior performance. A **limitation** is that our current theory is based on simplifying assumptions and does not establish a general convergence guarantee.

Acknowledgments and Disclosures

This work was supported by the Ningbo Global Innovation Center, Zhejiang University (Grant No. 701G26003). The authors declare no competing interests.

References

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*, pages 45–73. Springer, 2012.

Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8):10237–10257, 2023.

Junhyuk Oh, Greg Farquhar, Iurii Kemaev, Dan A Calian, Matteo Hessel, Luisa Zintgraf, Satinder Singh, Hado Van Hasselt, and David Silver. Discovering state-of-the-art reinforcement learning algorithms. *Nature*, pages 1–2, 2025.

Anqi Li, Byron Boots, and Ching-An Cheng. Mahalo: Unifying offline reinforcement learning and imitation learning from observations. In *Proceedings of the 40th International Conference on Machine Learning*, pages 19360–19384. PMLR, 2023.

Zhengbang Zhu, Hanye Zhao, Haoran He, Yichao Zhong, Shenyu Zhang, Haoquan Guo, Tingting Chen, and Weinan Zhang. Diffusion models for reinforcement learning: A survey. *arXiv preprint arXiv:2311.01223*, 2023.

Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 67195–67212, 2023.

Siddarth Venkatraman, Shivesh Khaitan, Ravi Tej Akella, John Dolan, Jeff Schneider, and Glen Berseth. Reasoning with latent diffusion in offline reinforcement learning. In *Proceedings of the 12th International Conference on Learning Representations*, pages 1–22, 2024.

Ruoqi Zhang, Ziwei Luo, Jens Sjölund, Thomas Schön, and Per Mattsson. Entropy-regularized diffusion policy with q-ensembles for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37:98871–98897, 2024.

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.

Wensong Bai, Chufan Chen, Yichao Fu, Qihang Xu, Chao Zhang, and Hui Qian. Rebalancing return coverage for conditional sequence modeling in offline reinforcement learning. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems*, 2025.

Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, et al. Diffusion actor-critic with entropy regulator. *Advances in Neural Information Processing Systems*, 37:54183–54204, 2024.

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.

Yuhui Chen, Haoran Li, and Dongbin Zhao. Boosting continuous control with consistency policy. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 335–344, 2024a.

Thomas A Berrueta, Allison Pinosky, and Todd D Murphey. Maximum diffusion reinforcement learning. *Nature Machine Intelligence*, 6(5):504–514, 2024.

Shiyuan Zhang, Weitong Zhang, and Quanquan Gu. Energy-weighted flow matching for offline reinforcement learning. In *Proceedings of the 13th International Conference on Learning Representations*, 2025.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 9902–9915. PMLR, 2022.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B Tenenbaum, Tommi S Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision making? In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *Proceedings of the 17th International Conference on Learning Representations*, pages 1–17, 2023.

Mingyang Sun, Pengxiang Ding, Weinan Zhang, and Donglin Wang. Score-based diffusion policy compatible with reinforcement learning via optimal transport. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Zhendong Wang, Max Li, Ajay Mandlekar, Zhenjia Xu, Jiaojiao Fan, Yashraj Narang, Linxi Fan, Yuke Zhu, Yogesh Balaji, Mingyuan Zhou, et al. One-step diffusion policy: Fast visuomotor policies via diffusion distillation. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: accelerated visuomotor policies via consistency distillation. In *Robotics: Science and Systems*, 2024.

Huaisheng Zhu, Teng Xiao, Shijie Zhou, Zhimeng Guo, Hangfan Zhang, Siyuan Xu, and Vasant G Honavar. Simple distillation for one-step diffusion models. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems*, 2025.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 1–24. PMLR, 2025.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. In *Proceedings of the 40th International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023.

Zihan Ding and Chi Jin. Consistency models as a rich and efficient policy class for reinforcement learning. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. *arXiv preprint arXiv:2410.20092*, 2024.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:20132–20145, 2021.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in Neural Information Processing Systems*, 34:15084–15097, 2021.

Shengchao Hu, Li Shen, Ya Zhang, and Dacheng Tao. Graph decision transformer. *arXiv preprint arXiv:2303.03747*, 2023.

Shengchao Hu, Ziqing Fan, Chaoqin Huang, Li Shen, Ya Zhang, Yanfeng Wang, and Dacheng Tao. Q-value regularized transformer for offline reinforcement learning. *arXiv preprint arXiv:2405.17098*, 2024.

Huayu Chen, Cheng Lu, Zhengyi Wang, Hang Su, and Jun Zhu. Score regularized policy optimization through diffusion behavior. In *Proceedings of the 12th International Conference on Learning Representations*, pages 1–22, 2024b.

A Method Details

A.1 Motivation

A.1.1 Why Denoising-Trajectory Instability Arises in Offline RL

We provide a theoretical perspective on why diffusion policies trained with standard objectives can exhibit *denoising-trajectory instability* in offline reinforcement learning. The key issue is not whether intermediate denoising steps improve monotonically, but whether decoded action candidates remain support-consistent and value-reliable throughout the reverse diffusion process.

Independent optimization across denoising steps. Diffusion policies are typically trained by optimizing a denoising objective independently at each timestep:

$$\mathcal{L}_{\text{DP}} = \mathbb{E}_{t,(s,a_0),\epsilon}\left[\|\epsilon - \epsilon_\theta(a_t, s, t)\|_2^2\right]. \tag{15}$$

While this objective ensures that each conditional distribution $p_\theta(a^{(t)} \mid s)$ matches the data distribution marginally, it does not impose any explicit constraint on the *relationship between decoded predictions at different timesteps*. As a result, denoising steps are not jointly optimized to form a stable refinement trajectory, and decoded actions from neighboring steps may differ substantially in both local consistency and support with respect to the offline dataset.

Mismatch between likelihood optimization and offline value reliability. The diffusion objective is derived from likelihood or score matching, which encourages consistency with the data distribution rather than reliable value-aware action selection. In offline RL, this mismatch is particularly problematic because the usefulness of a generated action depends not only on whether it resembles the dataset, but also on whether it lies in a region where the critic can evaluate it reliably. Let $Q(a, s)$ denote an action-quality measure. Along the denoising trajectory, decoded action candidates may drift toward weakly supported regions of the action space, where value estimates become less reliable due to extrapolation error. Consequently, even when some intermediate predictions are locally good, subsequent denoising steps may move toward less reliable candidates, making final action selection sensitive to denoising dynamics. This phenomenon is especially pronounced on mixed-quality offline datasets, where high-return and low-return behaviors coexist.

A.2 Q Function Learning.

In our framework, the Critic network is optimized by minimizing the Bellman error to evaluate the quality of the current policy. This process consists of the following three core components:

Target Value Computation. To mitigate the distribution shift in offline datasets and the resulting value overestimation, we employ the Double Q-learning mechanism to construct the Temporal Difference (TD) target:

$$y = r + \gamma \cdot \text{mask} \cdot \min_{j=1,2} Q_{\theta'_j}(s', a')$$

Where:y denotes the TD target value (Target Q).r is the reward obtained from the environment or dataset at the current step.γ is the discount factor, which is set to 0.99 in our implementation.mask is the terminal state mask ($1.0 -$ terminated), used to truncate value propagation at the end of a task.$Q_{\theta'_j}$ represents the Target Critic networks, whose parameters θ' are maintained through a soft update mechanism to ensure target stability.a' is the action for the next state.

Loss Function. The Critic network updates its parameters θ by minimizing the Mean Squared Error (MSE) between the current Q-value prediction and the target value y:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\theta(s, a) - y\right)^2\right]$$

This loss function ensures that the predicted Q-values converge toward the current Bellman optimal estimate.

Target Network Soft Update. The target network parameters θ' do not undergo direct gradient descent updates. Instead, they track the parameters of the online network through Polyak averaging (soft update) to enhance training robustness:

$$\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$$

In our implementation, the update rate τ is set to 0.005. This slow parameter update method effectively smoothes the TD target and prevents oscillations during the training process.

A.3 Additional Preliminary

Markov Decision Process (MDP). We formulate the agent-environment interaction as a finite-horizon Markov Decision Process (MDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mu_0, H)$. Here,

$\mathcal{S}$ and $\mathcal{A}$ represent the continuous state and action spaces, respectively. The stochastic transition dynamics are governed by $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ denotes the probability simplex over the state space. At each timestep t, the agent receives a scalar reward $r_t = \mathcal{R}(s_t, a_t)$, where $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The process initiates with a state s_1 sampled from the initial distribution μ_0 and terminates after a fixed horizon H.

A.4 Distillation to Single-Step Flow Matching

To distill the TCR-enhanced diffusion policy into a one-step flow-matching policy, we follow a similar procedure as outlined in Section 3.4 of the main text. The key steps are as follows:

- **Data Collection:** We first generate a dataset of state-action pairs by rolling out the TCR-enhanced diffusion policy in the environment. This dataset captures the behavior of the diffusion policy.
- **Flow-Matching Objective:** We then train a one-step flow-matching policy using the collected dataset. The flow-matching objective is defined as:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\|a - f_\phi(s)\|_2^2 \right], \tag{16}$$

  where $f_\phi(s)$ is the one-step flow-matching policy parameterized by ϕ.
- **Training:** We optimize the flow-matching objective using stochastic gradient descent, ensuring that the one-step policy learns to approximate the behavior of the TCR-enhanced diffusion policy.

B Theoretical Analysis of TCR as Support Regularization

In this section, we provide a theoretical perspective on why TCR improves the stability of diffusion policies in offline reinforcement learning. Our goal is not to prove strict monotonic improvement along denoising steps, but to show that, under suitable assumptions, TCR can be interpreted as a form of *support regularization* that reduces off-support leakage and yields a tighter critic error bound on the behavior manifold.

Preliminaries and the restricted behavior manifold. Consider a discounted Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\gamma \in (0, 1)$. In offline reinforcement learning, the dataset $\mathcal{D}$ is collected by an unknown behavior policy π_β.

A central challenge in offline RL is that evaluating actions outside well-covered dataset regions can induce severe extrapolation error in Bellman backup. To focus the analysis on the behavior-supported region, we define the state-action behavior manifold as

$$Z_\beta^* := \{(s, a) \mid s \in S_\beta^*,\ a \in \operatorname{supp}(\pi_\beta(\cdot \mid s))\},$$

where

$$S_\beta^* := \operatorname{supp}(d^{\pi_\beta}),$$

and d^{π_β} denotes the state visitation distribution induced by π_β.

For the restricted-manifold analysis below, we assume the following one-step transition closure property: for any $(s, a) \in Z_\beta^*$, the next state $s' \sim P(\cdot \mid s, a)$ satisfies $s' \in S_\beta^*$ almost surely. This is a technical assumption introduced to make the restricted-manifold Bellman analysis self-contained.

We define the restricted infinity norm over Z_β^* as

$$\|Q\|_{\infty, Z_\beta^*} := \sup_{(s,a)\in Z_\beta^*} |Q(s, a)|.$$

We further define the support-constrained Bellman optimality operator

$$(B_{\pi_\beta}^* Q)(s, a) = r(s, a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)} \left[\max_{a' \in \operatorname{supp}(\pi_\beta(\cdot|s'))} Q(s', a') \right].$$

Its fixed point is denoted by $Q_{\beta,\text{supp}}^*$, which serves as the reference optimum under support-constrained policy improvement induced by the behavior-policy support. It should be interpreted as an in-support reference optimum rather than the globally optimal value function over the full action space.

Lemma B.1 (Contraction on the restricted state-action manifold)**.** *The operator $B^*_{\pi_\beta}$ is a γ-contraction under the restricted norm $\|\cdot\|_{\infty,Z^*_\beta}$:*

$$\|B^*_{\pi_\beta}Q_1 - B^*_{\pi_\beta}Q_2\|_{\infty,Z^*_\beta} \le \gamma\|Q_1 - Q_2\|_{\infty,Z^*_\beta}.$$

Proof. For any $(s,a) \in Z^*_\beta$, the transition closure property ensures that $s' \sim P(\cdot \mid s,a)$ satisfies $s' \in S^*_\beta$ almost surely. For each such s', the maximization is restricted to $a' \in \mathrm{supp}(\pi_\beta(\cdot \mid s'))$, so $(s',a') \in Z^*_\beta$. Therefore,

$$\begin{aligned}
&|(B^*_{\pi_\beta}Q_1)(s,a) - (B^*_{\pi_\beta}Q_2)(s,a)| \\
&\le \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[\max_{a'\in\mathrm{supp}(\pi_\beta(\cdot|s'))}|Q_1(s',a') - Q_2(s',a')|\right] \\
&\le \gamma \sup_{(s',a')\in Z^*_\beta}|Q_1(s',a') - Q_2(s',a')| \\
&= \gamma\|Q_1 - Q_2\|_{\infty,Z^*_\beta}.
\end{aligned}$$

Taking the supremum over $(s,a) \in Z^*_\beta$ completes the proof. □

TCR as support regularization and the projected policy. In standard diffusion policies, intermediate denoising steps may drift away from the behavior-supported region due to stochastic reverse dynamics. In offline RL, evaluating such off-support actions can destabilize policy improvement and amplify Bellman backup error.

We interpret TCR as a form of support regularization: it biases denoising trajectories toward action candidates that remain closer to the underlying behavior manifold. To formalize this effect, we define the worst-case support deviation of a policy π over the behavior state manifold as

$$\Delta_{\infty,\beta}(\pi) := \sup_{s\in S^*_\beta}\mathrm{dist}\big(\pi(s), \mathrm{supp}(\pi_\beta(\cdot \mid s))\big),$$

where $\mathrm{dist}(a,\mathcal{A}_0)$ denotes the shortest distance from an action a to a set $\mathcal{A}_0$.

We assume that for every $s \in S^*_\beta$, the support set $\mathrm{supp}(\pi_\beta(\cdot \mid s))$ is nonempty and closed, so that the following nearest-point projection is well-defined. We define the projected policy by selecting any nearest in-support action:

$$\tilde{\pi}_k(s) \in \arg\min_{a\in\mathrm{supp}(\pi_\beta(\cdot|s))}\mathrm{dist}\big(\pi_k^{\mathrm{TCR}}(s), a\big).$$

Since $\mathrm{supp}(\pi_\beta(\cdot \mid s))$ is nonempty and closed, the minimum is attained, although the minimizer need not be unique. By construction, $\tilde{\pi}_k(s) \in \mathrm{supp}(\pi_\beta(\cdot \mid s))$, so the Bellman evaluation operator $B_{\tilde{\pi}_k}$ is closed on Z^*_β.

Assumption B.2 (Restricted support proximity induced by TCR)**.** Let π_k^{TCR} be the policy induced by TCR at iteration k. There exists a sequence $\{\delta_k\}_{k\ge0}$ such that

$$\Delta_{\infty,\beta}(\pi_k^{\mathrm{TCR}}) \le \delta_k.$$

Remark (Scope of the support-proximity assumption). Assumption B.2 characterizes a sufficient condition for controlling critic error rather than a property guaranteed by TCR. While TCR is designed to encourage trajectory consistency and suppress unstable action candidates, it does not explicitly enforce proximity to the behavior-policy support. Therefore, the following analysis is conditional on this assumption and does not establish a general support-proximity guarantee for the proposed algorithm.

Assumption B.3 (Lipschitz critic and OOD leakage control)**.** For each k, the critic $Q_k(s,\cdot)$ is L-Lipschitz with respect to the action, i.e.,

$$|Q_k(s,a_1) - Q_k(s,a_2)| \le L\,\mathrm{dist}(a_1,a_2), \quad \forall s \in S^*_\beta,\ \forall a_1,a_2 \in \mathcal{A}.$$

Under Assumption 1, the projection definition implies

$$\operatorname{dist}\big(\pi_k^{\mathrm{TCR}}(s), \tilde{\pi}_k(s)\big) \le \delta_k, \quad \forall s \in S_\beta^*.$$

Therefore,

$$\|B_{\pi_k^{\mathrm{TCR}}} Q_k - B_{\tilde{\pi}_k} Q_k\|_{\infty, Z_\beta^*} \le \gamma L \delta_k.$$

Thus, the off-support leakage of $B_{\pi_k^{\mathrm{TCR}}}$ is controlled by its distance to a closed in-support operator.

Assumption B.4 (In-support optimization gap)**.** There exists a sequence $\{\eta_k\}_{k\ge 0}$ such that

$$\|B_{\tilde{\pi}_k} Q_k - B^*_{\pi_\beta} Q_k\|_{\infty, Z_\beta^*} \le \gamma \eta_k.$$

Assumption B.5 (Bounded critic approximation error)**.** The fitted critic update satisfies

$$\|Q_{k+1} - B_{\pi_k^{\mathrm{TCR}}} Q_k\|_{\infty, Z_\beta^*} \le \epsilon_{\mathrm{approx}},$$

where $\epsilon_{\mathrm{approx}}$ captures the approximation and statistical estimation error of the fitted critic when evaluated on the behavior manifold.

Theorem B.6 (Approximate critic error bound under TCR)**.** *Under Lemma 1 and Assumptions 1–4, the critic sequence satisfies*

$$\|Q_{k+1} - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*} \le \epsilon_{\mathrm{approx}} + \gamma(L\delta_k + \eta_k) + \gamma \|Q_k - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*}.$$

Consequently, if $\sup_k \delta_k \le \bar{\delta}$ *and* $\sup_k \eta_k \le \bar{\eta}$*, then*

$$\limsup_{k\to\infty} \|Q_k - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*} \le \frac{\epsilon_{\mathrm{approx}} + \gamma(L\bar{\delta} + \bar{\eta})}{1-\gamma}.$$

In particular, if $\delta_k \to 0$ *and* $\eta_k \to 0$*, then*

$$\limsup_{k\to\infty} \|Q_k - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*} \le \frac{\epsilon_{\mathrm{approx}}}{1-\gamma}.$$

Proof. By repeated application of the triangle inequality on the restricted norm,

$$\begin{aligned}
&\|Q_{k+1} - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*} \\
&\le \|Q_{k+1} - B_{\pi_k^{\mathrm{TCR}}} Q_k\|_{\infty, Z_\beta^*} + \|B_{\pi_k^{\mathrm{TCR}}} Q_k - B_{\tilde{\pi}_k} Q_k\|_{\infty, Z_\beta^*} \\
&\quad + \|B_{\tilde{\pi}_k} Q_k - B^*_{\pi_\beta} Q_k\|_{\infty, Z_\beta^*} + \|B^*_{\pi_\beta} Q_k - B^*_{\pi_\beta} Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*}.
\end{aligned}$$

The first term is bounded by $\epsilon_{\mathrm{approx}}$ from Assumption 4. The second term is bounded by $\gamma L \delta_k$ from Assumptions 1–2. The third term is bounded by $\gamma \eta_k$ from Assumption 3. The fourth term is bounded by

$$\gamma \|Q_k - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*}$$

by Lemma 1. Combining the four bounds yields

$$\|Q_{k+1} - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*} \le \epsilon_{\mathrm{approx}} + \gamma L \delta_k + \gamma \eta_k + \gamma \|Q_k - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*}.$$

Unrolling the recursion gives

$$\|Q_k - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*} \le \gamma^k \|Q_0 - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*} + \sum_{i=0}^{k-1} \gamma^{k-1-i} \big(\epsilon_{\mathrm{approx}} + \gamma(L\delta_i + \eta_i)\big).$$

If $\delta_i \le \bar{\delta}$ and $\eta_i \le \bar{\eta}$, then taking the limit supremum as $k \to \infty$ yields

$$\limsup_{k\to\infty} \|Q_k - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*} \le \frac{\epsilon_{\mathrm{approx}} + \gamma(L\bar{\delta} + \bar{\eta})}{1-\gamma}.$$

If $\delta_k \to 0$ and $\eta_k \to 0$, the residual extrapolation and optimization gaps vanish asymptotically, yielding

$$\limsup_{k\to\infty} \|Q_k - Q^*_{\beta,\mathrm{supp}}\|_{\infty, Z_\beta^*} \le \frac{\epsilon_{\mathrm{approx}}}{1-\gamma}.$$

□

Discussion and limitations. Theorem B.6 provides a sufficient-condition analysis of critic error rather than a convergence guarantee for TCR. Specifically, the bound holds under support proximity, Lipschitz continuity of the critic, transition closure on the behavior-supported manifold, and bounded approximation and optimization errors. These assumptions facilitate the analysis but may be difficult to verify in practical offline RL settings. Importantly, our analysis does not establish that TCR necessarily satisfies the support-proximity condition. Instead, it characterizes how critic error can be controlled when the learned policy remains sufficiently close to the behavior support. TCR encourages trajectory consistency through training-time rectification and inference-time voting, while our empirical analyses of action deviation and Q-value fluctuation provide supporting evidence of improved stability. These observations are consistent with the theoretical motivation but do not constitute a formal verification of the assumptions.

C Experiment Details

C.1 Experimental Environments and Datasets

We evaluate our method on an extensive set of benchmarks spanning five distinct domains within the D4RL suite (Gym-MuJoCo, Adroit, Kitchen, Maze2D, and AntMaze), complemented by the challenging OGBench datasets and complex robotic manipulation scenarios. This diverse selection allows us to investigate performance across varying degrees of data quality, trajectory sub-optimality, and task complexity. Consequently, our experiments provide a holistic view of the agent's ability to learn effective policies in both dense and sparse reward settings.

Gym-MuJoCo Gym-MuJoCo locomotion tasks serve as standard benchmarks for continuous control. These tasks, such as HalfCheetah, Hopper, and Walker2d, are characterized by smooth reward functions and dense feedback signals. The datasets include trajectories generated by policies of varying quality: random, medium, expert, and medium-expert, representing different levels of optimality. While expert and medium-expert provide high-quality trajectories, medium, random, and medium-replay contain suboptimal and exploratory behavior, offering diverse state-action coverage for offline learning.

Adroit Adroit tasks involve the precise manipulation of a high-dimensional, 24-DoF robotic hand for tasks such as Pen (twirling), Hammer (hitting a nail), Door (opening a handle), and Relocate (moving an object). Unlike locomotion, these tasks feature extremely complex action spaces and highly sparse reward signals. The datasets typically consist of a small number of human demonstrations or trajectories generated by behavioral cloning. The primary challenge lies in learning complex coordination and fine motor skills from extremely narrow and non-structured data distributions.

Maze2D Maze2D is a navigation benchmark that requires an agent to reach a target location within 2D layouts of varying complexity, such as U-maze, Medium, and Large mazes. The datasets primarily consist of non-goal-directed, random-walk trajectories that do not contain complete demonstrations from start to finish. This makes Maze2D an ideal platform for evaluating the "stitching" capability of offline algorithms—testing whether they can extract and re-compose fragmented, suboptimal experiences into a single, globally optimal path.

AntMaze AntMaze combines the complex dynamics of an 8-DoF quadruped (Ant) with challenging maze navigation. Compared to Maze2D, the Ant agent introduces higher dimensionality and the risk of flipping over, making the physics much more difficult to manage. The datasets typically feature expert-generated trajectories for various start and goal pairs, representing a quintessential long-horizon, sparse-reward task. Algorithms must not only demonstrate strong stitching capabilities but also handle significant distribution shifts in high-dimensional non-linear systems.

OGBench The Offline Generalization Benchmark (OGBench) is a novel suite designed to evaluate large-scale generalization across a diverse set of tasks, categorized as follows:

- **AntMaze (Large/Giant):** These tasks require an Ant agent to navigate massive maze environments with extremely long-horizon goals. The datasets contain disjointed exploratory paths, testing the limits of trajectory stitching under high-dimensional dynamics.

- **HumanoidMaze (Medium/Large):** This task elevates navigation difficulty by introducing a humanoid agent. It serves as a critical test for handling high-dimensional non-linear systems and severe distribution shifts where the agent must maintain balance while planning.
- **Cube (Single/Double):** These tasks focus on object manipulation, ranging from basic grasping of a single cube to complex inter-object interactions like stacking two cubes. They evaluate fine-grained tactile control and coordination within continuous action spaces.
- **Scene (Single):** This task places manipulation within complex visual backgrounds with various distractors, emphasizing the ability to extract key features from high-dimensional visual inputs and maintain spatial generalization.
- **Puzzle (3x3/4x4):** These are high-dimensional logical reasoning tasks requiring the agent to restore a slider puzzle. With exponential state-space growth and strict logical dependencies, they challenge the algorithm's ability to solve combinatorial optimization problems.

Robotic Manipulation Tasks These tasks focus on fine-grained manipulation and contact-rich interactions using various robotic platforms (e.g., Panda or Sawyer arms):

- **Blockpushing/PushT:** These tasks require the agent to push objects (blocks or T-shaped structures) to target poses. They emphasize the learning of multimodal action distributions and precise spatial coordination.
- **Lift / Can / Square (MH):** These environments, often sourced from the Robomimic benchmark, involve picking up a cube, placing a can into a bin, or fitting a square nut onto a peg. The "MH" (Multi-Human) datasets are particularly challenging as they contain diverse, non-optimal demonstrations from multiple operators with varying skill levels.
- **Tool Hang (PH):** This task requires an agent to use a tool to hang an object on a frame, involving complex kinematic chains and precise insertion. The "PH" (Proficient Human) data provides high-quality but narrow demonstrations that test an algorithm's ability to stay within the supported state-action distribution.
- **Transport:** A multi-stage task where two robotic arms must coordinate to move an object from one bin to another. This environment evaluates the ability of offline RL algorithms to handle long-horizon temporal dependencies and inter-agent (or dual-arm) coordination.

C.2 Implementation Details

C.2.1 Hardware & Hyper-parameters

All the experiments were conducted on a server with 8 NVIDIA ® A800 GPUs. The diffusion policy and one-step flow-matching policy were implemented using PyTorch. We used the Adam optimizer with a learning rate of 3e-4 for training both diffusion policies and one-step policies. The batch size was set to 256 for all experiments. The training process for diffusion policies typically took around 4 hours, while distilling into one-step policies took approximately 1 hours. Related Hyperparameters are listed in Table 4.

Table 4: Key hyperparameter of our algorithm.

Hyperparameter	**Value**
Learning rate	0.0003
Gradient steps	1000000
Minibatch size	256
Actor hidden dims	[512, 512, 512, 512]
Value hidden dims	[512, 512, 512, 512]
Target network smoothing coefficient τ	0.005
Discount factor γ	0.99
Diffusion steps	100
Q aggregation strategy	min
Voting alpha	10

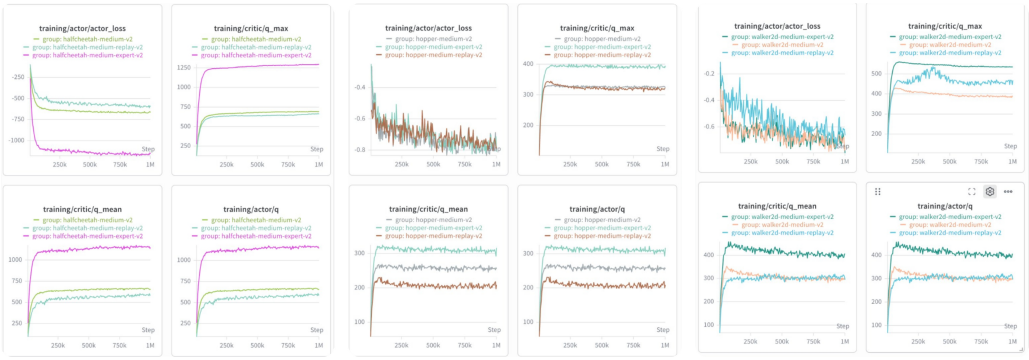


Figure 2: Training dynamics of TCR on representative Gym locomotion tasks. We report actor loss, critic statistics, and actor-side Q-values under different dataset qualities. The curves show stable optimization behavior across medium, medium-replay, and medium-expert settings.

C.2.2 Compared Approaches

For diffusion-policy baselines, we use the standard diffusion-policy training described in Section 3.1 without TCR. For one-step flow-matching baselines, we follow the distillation procedure in Section 3.3 using standard diffusion policies as teachers. All baselines are reproduced under the same experimental settings with their official implementations whenever available for fair comparison.

C.2.3 Metrics

For D4RL, we report the normalized score:

$$\text{Normalized_Score} = \frac{\text{Return}_{\text{current}} - \text{Return}_{\text{random}}}{\text{Return}_{\text{expert}} - \text{Return}_{\text{random}}} \times 100.$$

A normalized score of 100 indicates expert-level performance, while a score of 0 corresponds to random behavior. For OGBench, we follow the standard benchmark protocol and report the corresponding task performance scores.

D Additional Results

D.1 Additional Analysis

D.1.1 Training Dynamics Analysis

We provide additional training curves in Fig. 2 to examine the optimization behavior of TCR. The curves show that TCR remains numerically stable throughout training and does not introduce noticeable optimization instability. In particular, the actor loss decreases smoothly without divergence, while `critic/q_mean` and `actor/q` remain closely aligned and quickly enter stable regimes. This suggests that TCR does not destabilize critic learning; instead, it helps maintain a coherent value signal between actor and critic during optimization. Overall, these training dynamics are consistent with our interpretation that TCR improves denoising-trajectory stability without sacrificing optimization stability.

D.1.2 Efficiency Analysis

We further report the inference efficiency of TCR and its distilled one-step student in Table 5. We compare standard Diffusion Policy (DP), DP enhanced with TCR, and the final distilled student under a batch size of 256. For DP-based methods, we use 100 reverse diffusion steps, while the student generates actions in a single step.

The results show that the voting mechanism in TCR introduces only a marginal inference overhead on the multi-step diffusion teacher: the batch inference time changes from 8.467 ms for DP to

Table 5: Inference efficiency comparison between standard Diffusion Policy (DP), DP enhanced with TCR, and the distilled one-step student.

Method	Steps	Batch Time (ms)	Sample Time (ms)	Throughput (samples/s)	Relative Speed
DP	100	8.467	0.0331	30234.9	1.00×
DP+TCR	100	8.665	0.0338	29549.9	0.98×
Student	1	0.208	0.0008	1230769.2	40.72×

8.665 ms for DP+TCR, corresponding to only about 2.3% additional overhead. In contrast, the distilled one-step student requires only 0.208 ms per batch, achieving about $40.72\times$ speedup over the original diffusion policy. These results indicate that TCR improves policy quality with negligible additional cost on the teacher side, while the distilled student provides a substantial efficiency gain for deployment.

D.1.3 Additional Value Trend Analysis

We further visualize the Q-value trend of TCR across diffusion steps in Figure 3. The critic value remains stable and gradually improves during denoising, suggesting that TCR reduces late-stage degradation in decoded action quality. We emphasize that this result is intended as an empirical observation rather than a guarantee of strict monotonic improvement.

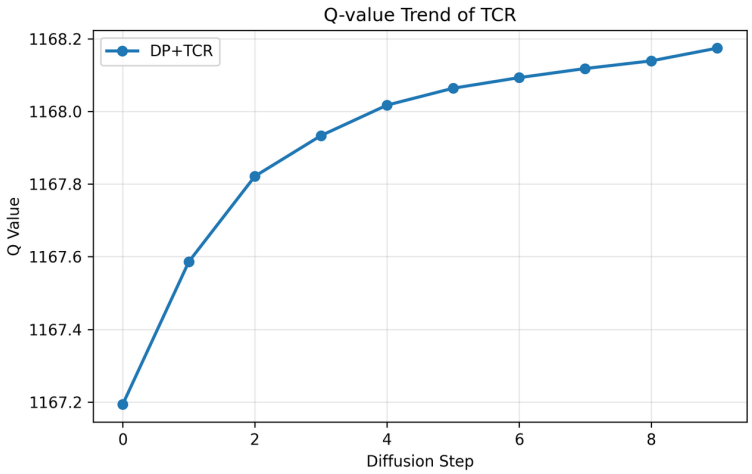


Figure 3: Q-value trend of TCR across diffusion steps. The critic value under TCR remains stable and gradually improves across the denoising process, suggesting that TCR reduces late-stage degradation and supports more reliable value-guided action selection.

Effectiveness of Consistency Components. The results demonstrate that both Consistency Voting and Action Consistency Rectification are essential for stabilizing policy performance.

- **Individual Contributions:** Incorporating either CV or CR alone yields improvements over the standard DP baseline. For instance, the average normalized score increases from 50.8 (DP) to 53.0 (DP+CV) and 52.5 (DP+CR). This confirms that mitigating policy oscillation—whether through inference-time voting or training-time rectification—benefits action quality.
- **Synergistic Effect:** The combination of both components (DP+TCR) results in a further performance boost, raising the average score to 55.9. This indicates that enforcing trajectory coherence during training (ACR) and refining outputs during inference (Voting) are complementary, leading to a stronger teacher policy.

Table 6: Performance of the one-step distilled version (*Ours*) on Gym Locomotion tasks. Results are reported as mean ± standard deviation.

Dataset	HalfCheetah	Hopper	Walker2d
Medium-Expert	110.1 ± 1.1	113.4 ± 0.7	114.0 ± 1.2
Medium	61.0 ± 0.5	103.3 ± 0.6	90.0 ± 0.2
Medium-Replay	52.7 ± 1.3	103.3 ± 2.6	100.7 ± 2.2

Table 7: Performance of the one-step distilled version (*Ours*) on Adroit tasks. Results are reported as mean ± standard deviation.

Dataset	Pen-cloned	Hammer-cloned
Cloned	94.0 ± 1.0	15.1 ± 1.5

Distillation as Policy Refinement (Student > Teacher). A remarkable finding in our experiments is that the final distilled one-step policy (**Ours**) significantly outperforms its teacher (**DP+TCR**), achieving the highest average score of 60.4.

- This phenomenon is particularly pronounced in the **Hammer-cloned** task, where the student policy achieves a score of 15.12, compared to the teacher's 2.99.
- This validates the core design of our distillation pipeline: the **Trajectory Consistency Voting** mechanism successfully acts as a data curator. By utilizing the learned critic Q_Φ to selectively filter out oscillatory or low-quality trajectories from the teacher and aggregating only the stable, high-value actions as targets, the student policy effectively "cleans" the behavior distribution. This results in a self-improvement effect where the distilled policy transcends the performance limitations of the teacher.

Robustness on Suboptimal Data. Our method demonstrates superior robustness on the challenging *Human* and *Cloned* datasets, which are characterized by high noise and multimodality.

- In the **Pen-human** task, our method achieves a state-of-the-art score of 101.88, drastically surpassing the strongest baseline (IDQL: 76.0) and the standard DP (77.95).
- These results suggest that TCR is uniquely capable of stabilizing learning in scenarios where data quality is inconsistent, making it a highly effective solution for offline RL with imperfect demonstrations.

D.2 Additional Quantitative Results

We provide the complete experimental results on Maze2D benchmarks in Table 11 and AntMaze benchmarks in Table 12. Our method consistently outperforms existing baselines across all tasks, demonstrating its effectiveness in stabilizing diffusion policies and enhancing action quality. Furthermore, results on the full OGBench suite, as shown in Table 13, highlight our method's versatility. By achieving superior results in both state-based and pixel-based domains, our approach underscores its scalability and effectiveness for high-dimensional, goal-conditioned offline RL.

D.3 Statistic Significance

We also provide the performance of the one-step distilled version of our method on Gym Locomotion, Adroit, and AntMaze tasks in Tables 6, 7, and 10, respectively. The results are reported as mean ± standard deviation across multiple runs, demonstrating the statistical significance of our method's performance improvements over baselines. The consistent performance gains across different seeds and tasks further validate the robustness and reliability of our approach in various offline RL settings.

Table 8: Hyper-parameter sensitivity of TCR under different temperature α and window size k.

Window size k	Temperature α			
	0	1	5	10
1	108.63866	109.13496	108.19550	109.34100
3	109.06708	108.37460	108.14693	107.96318
5	109.95129	108.87620	108.99180	110.17854
7	110.37451	109.30017	108.76131	107.94976

Table 9: Comparison between standard Diffusion Policy (DP), critic-guided Diffusion Q-Learning (DQL), and our method on Gym tasks. Results of *Ours* are reported as mean $\pm$ standard deviation.

Gym Task	DP	DQL	Ours
halfcheetah-medium-v2	42.5	51.1	61.0 ± 0.5
hopper-medium-v2	57.5	90.5	103.3 ± 0.6
walker2d-medium-v2	71.1	87.0	90.0 ± 0.2
halfcheetah-medium-replay-v2	38.9	47.8	52.7 ± 1.3
hopper-medium-replay-v2	35.9	101.3	103.3 ± 2.6
walker2d-medium-replay-v2	46.6	95.5	100.7 ± 2.2
halfcheetah-medium-expert-v2	73.2	96.8	110.1 ± 1.1
hopper-medium-expert-v2	92.7	111.1	113.4 ± 0.7
walker2d-medium-expert-v2	108.5	110.1	114.0 ± 1.2

D.4 Results on hyper-parameter sensitivity

To examine the robustness of TCR with respect to hyper-parameter choices, we conduct a small ablation on the temperature parameter α and the local window size k used in trajectory consistency voting. Table 8 reports the results under different combinations of α and k.

The results show that TCR does not rely on a brittle hand-tuned parameter combination, but remains stable under moderate changes of both α and k. Across all tested settings, the performance variation is relatively small, suggesting that the proposed trajectory consistency voting mechanism is robust to reasonable hyper-parameter choices.

For the kernel bandwidth γ, we do not treat it as a heavily hand-tuned free hyper-parameter. Instead, we adopt a standard data-adaptive heuristic from kernel methods:

$$\gamma^2 = \operatorname{median}\left(\left\| \hat{a}_0^{(t)} - \hat{a}_0^{(j)} \right\|_2^2 \right), \tag{17}$$

where the median is computed over action pairs within the current local trajectory window. In this way, the bandwidth is automatically matched to the intrinsic scale of decoded actions, rather than requiring extensive manual tuning.

D.5 Final-Step Action Quality and Critic Guidance

In offline RL, the quality of the final decoded action is particularly important, since value-guided action selection depends on the critic estimate of the generated action. If the policy produces weakly supported or out-of-distribution action candidates, critic estimation can become less reliable due to extrapolation error, which in turn degrades final action selection. To further examine this effect, we compare standard Diffusion Policy (DP), its critic-guided variant (denoted as DQL), and our method on representative Gym tasks.

Table 9 shows that adding critic guidance already improves over standard DP, while TCR further improves over both baselines. This result supports our motivation that, in critic-guided offline RL, stabilizing denoising trajectories and improving the reliability of the final decoded action are both important for strong policy performance.

Table 10: Performance of the one-step distilled version (*Ours*) on AntMaze tasks. Results are reported as mean ± standard deviation.

Dataset	AntMaze-U	AntMaze-U-D	AntMaze-M-D	AntMaze-L-D
MazeResults	100.0 ± 2.0	98.0 ± 6.0	86.0 ± 4.0	94.0 ± 6.0

Table 11: **Normalized scores on Maze2D benchmarks.**

Task	**CQL**	**IQL**	**TD3+BC**	**DT**	**GDT**	**QT**	**DD**	**IDQL**	**Diffuser**	**FQL**	**DP**	**Ours**
Maze2D-U	94.7	42.1	14.8	31.0	50.4	57.3	113.9	42.3	113.9	83.1	10.7	**160.6**
Maze2D-Med	41.8	34.9	62.1	8.2	7.8	13.3	121.5	-2.1	121.5	107.1	15.9	**122.1**
Maze2D-Large	49.6	61.7	88.6	2.3	0.7	31.0	123.0	31.8	123.0	160.0	13.0	**219.1**
Average	62.0	46.2	55.2	13.8	19.6	33.9	119.5	24.0	119.5	116.7	13.2	**167.3**

Table 12: **Normalized scores on AntMaze benchmarks.**

	Generative Policies				
Task	**IDQL**	**SRPO**	**FQL**	**DP**	**Ours**
AntMaze-U	93.8	97.1	96.0	62.0	**100.0**
AntMaze-U-D	62.0	82.1	89.0	62.0	**98.0**
AntMaze-M-P	86.6	80.7	78.0	0.0	86.0
AntMaze-M-D	83.5	75.0	71.0	0.0	**86.0**
AntMaze-L-P	57.0	53.6	84.0	0.0	72.0
AntMaze-L-D	56.4	53.6	83.0	0.0	**94.0**
Average	73.2	73.7	83.5	20.7	**89.3**

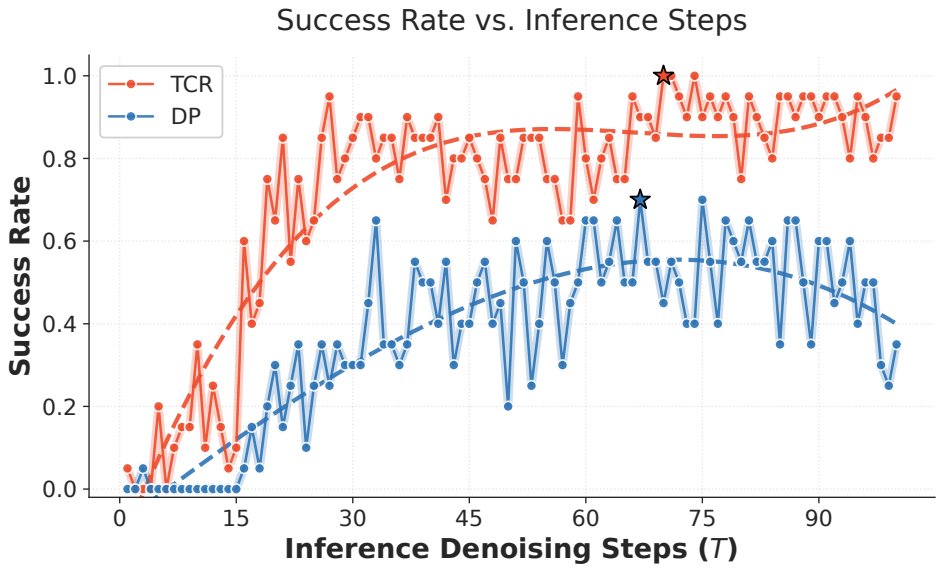


Figure 4: Comparison of policy oscillation before and after applying TCR. The figure illustrates how our method effectively stabilizes the success rate across denoising steps.

Table 13: **Full OGBench Experimental Results.** The left panel presents navigation environments, while the right panel presents manipulation environments. We compare our method against diffusion-based baselines. **Bold** indicates the highest score for each task, and $(*)$ denotes the primary evaluation task.

	Diffusion Baselines			Proposed
Navigation Task	IDQL	SRPO	CAC	**Ours**
AntMaze Large Navigate				
Task 1 $(*)$	0	0	42	**96**
Task 2	**14**	4	1	2
Task 3	26	3	49	**98**
Task 4	62	45	17	**94**
Task 5	2	1	55	**96**
AntMaze Giant Navigate				
Task 1 $(*)$	0	0	0	**12**
Task 2	0	0	0	**2**
Task 3	0	0	0	0
Task 4	0	0	0	**2**
Task 5	0	0	0	**48**
HumanoidMaze Medium Navigate				
Task 1 $(*)$	1	0	**38**	34
Task 2	1	1	47	**52**
Task 3	0	2	**83**	12
Task 4	1	1	**5**	3
Task 5	1	3	**91**	64
HumanoidMaze Large Navigate				
Task 1 $(*)$	0	0	1	**2**
Task 2	0	0	0	0
Task 3	**3**	1	2	2
Task 4	0	0	0	0
Task 5	0	0	0	0
AntSoccer Arena Navigate				
Task 1	**44**	2	1	12
Task 2	**15**	3	0	4
Task 3	0	0	**8**	6
Task 4 $(*)$	0	0	0	**2**
Task 5	0	0	0	**4**
Average	6.80	2.64	17.60	**25.88**

	Diffusion Baselines			Proposed
Manipulation Task	IDQL	SRPO	CAC	**Ours**
Cube Single Play				
Task 1	95	89	77	**98**
Task 2 $(*)$	96	82	80	**99**
Task 3	**99**	96	98	**99**
Task 4	93	70	91	**98**
Task 5	90	61	80	**96**
Cube Double Play				
Task 1	**39**	7	21	36
Task 2 $(*)$	**16**	2	2	10
Task 3	**17**	0	3	4
Task 4	0	0	0	0
Task 5	1	0	3	**4**
Scene Play				
Task 1	**100**	94	**100**	**100**
Task 2 $(*)$	33	2	50	**88**
Task 3	94	4	49	**100**
Task 4	4	0	0	**6**
Task 5	0	0	0	0
Puzzle 3x3 Play				
Task 1	52	89	97	**98**
Task 2	0	0	0	**86**
Task 3	0	0	0	**28**
Task 4 $(*)$	0	0	0	**36**
Task 5	0	0	0	**76**
Puzzle 4x4 Play				
Task 1	48	24	44	**64**
Task 2	**14**	0	0	6
Task 3	**34**	21	29	24
Task 4 $(*)$	26	7	1	**40**
Task 5	**24**	1	0	4
Average	39.00	25.96	33.00	**52.00**

D.6 Additional Visual Results

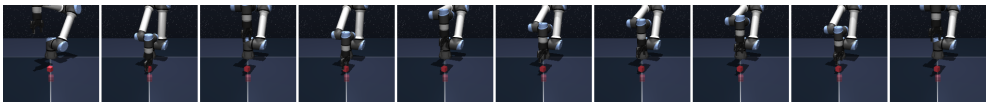
D.6.1 Policy oscillations

We first provide additional results on the policy oscillation mitigating effect of our TCR framework within the cube-single task (Fig. 4). The results clearly demonstrate that our method effectively stabilizes the success rate across denoising steps, significantly reducing oscillations compared to the baseline diffusion policy. Besides, TCR show a trend of performance increasing with the inference steps.

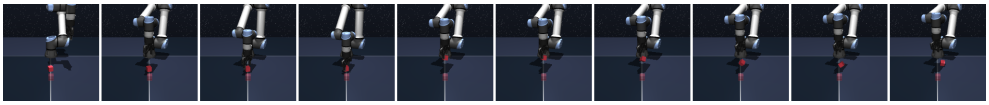
D.6.2 Generated Trajectory Visulization

We visualize a successful trajectory generated by our model in Fig. 5. The filmstrip demonstrates dual deficits in the standard diffusion policy regarding spatial precision and action consistency. As shown in Fig. 5a , stochastic noise in the final denoising steps causes terminal spatial drift, resulting in 'Air Grasping' where the end-effector fails to enclose the object. Simultaneously, Fig. 5b illustrates gripper command oscillation (the 'Leaky Grip') during the lifting phase, leading to object slippage due to insufficient retention force. In contrast, TCR acts as a spatio-temporal stabilizer by enforcing consistency constraints. It effectively filters out positional jitter to ensure geometric alignment while

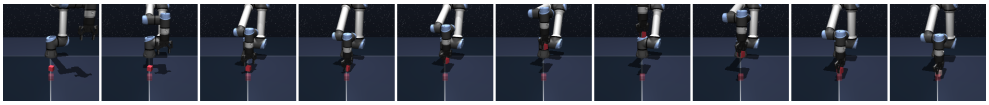
locking the gripper state to prevent accidental release, thereby significantly boosting manipulation success rates.



(a) Air Grasping (Standard DP): Due to the instability in the final denoising steps, the end-effector exhibits terminal spatial drift. Although the approach is generally correct, the gripper closes at a misaligned position, resulting in a missed grasp.



(b) Gripper Oscillation (Standard DP): During the lifting phase, the baseline fails to maintain temporal consistency in the gripper action dimension. The "Leaky Grip" phenomenon causes the object to slip due to intermittent relaxation.



(c) Stabilized Trajectory (Ours - TCR): By rectifying inconsistent predictions, TCR generates a spatially smooth approach and a temporally consistent grasping primitive , ensuring successful manipulation.

Figure 5: Qualitative comparison of policy stability on the Cube Single task.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims of the paper accurately reflect the contributions and scope of our paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations of our work in the section of Conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provided the proofs in Section B in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the information needed to reproduce the main experimental results of the paper is disclosed in Section 4 and Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: All the data in the paper is open access and the code will be released upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All the experimental details are provided in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide error bars for the main experiments in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All the information on the compute resources needed to reproduce the experiments is provided in Appendix C.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: All the research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential positive and negative societal impacts of our work in the conclusion section.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release any data or models that have a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the creators or original owners of assets used in the paper are properly credited and the license and terms of use are explicitly mentioned and properly respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not introduce new assets in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not conduct crowdsourcing experiments or research with human subjects in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We do not conduct research with human subjects in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We do not use LLMs as any important, original, or non-standard components of the core methods in this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.