Distribution Corrected Decision Transformer for Offline Reinforcement Learning with Imbalanced Datasets

Chufan Chen[1] **Bohao Tang**[2] **Dejiang Chen**[1] **Chao Zhang**[1*]

Ruiqi Liu[4,5] **Hanbin Zhao**[1] **Hui Qian**[1]

[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[2]Polytechnic Institute of Zhejiang University, Hangzhou, China
[4]Hangzhou International Innovation Institute, Beihang University, Hangzhou, China
[5]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China
{chufan.chen,12660052,22660092,zczju,zhaohanbin,qianhui}@zju.edu.cn
ruiqili@buaa.edu.cn

Abstract

Decision Transformer (DT) has emerged as a powerful paradigm for offline reinforcement learning. For complex tasks with imbalanced datasets, where high-quality samples constitute only a small fraction of the data while the majority is suboptimal, the performance of DT methods usually degrade significantly due to the the offline data distribution deviates substantially from the stationary distribution of the optimal policy. In this paper, we construct a dual-based correction mechanism to mitigate the distribution shift problem and propose a Distribution Corrected Decision Transformer method. Specifically, we investigate the Lagrangian duality of the reward maximizing objective in the standard Markov decision problem and recover the optimal correction weights between the static offline data and optimal stationary distribution. Note that a double-KL regularizer is integrated in the objective to prioritize high-return state-action pairs while remaining anchored to valid data support. Subsequently, we use the learned weights to correct distributional bias in two critical components of DT: stabilizing policy evaluation through weighted temporal-difference learning, and improving policy extraction by prioritizing DT training on weighted dataset aligned with the corrected target distribution. We theoretically analyze the performance guarantee, proving our method improves upon standard baselines, and empirically demonstrate strong performance on D4RL benchmarks, particularly on highly imbalanced datasets where prior methods fail.

1 Introduction

Offline reinforcement learning (RL)[36, 38] aims to learn optimal decision-making policies solely from static, pre-collected datasets, eliminating the need for costly or unsafe online interactions [11, 39, 42, 55]. Recent advancements in offline RL have been largely driven by Conditional Sequence Modeling (CSM), where the core idea is to learn the return-conditioned distribution of actions at each state given historical information [5, 7, 8]. A prominent paradigm within CSM is established by the Decision Transformer (DT) [7] and its variants [6, 23, 27, 59, 62]. These methods employ Transformer-based architectures to predict actions conditioned on target returns.

*Corresponding author.

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

By maximizing the likelihood of the training data, DT methods effectively perform return-conditioned behavioral cloning and show great superiority on standard benchmarks [10]. However, their effectiveness weakens markedly in the presence of *imbalanced datasets*, where high-quality samples constitutes only a small fraction of the dataset while most transitions are generated by suboptimal behavior [1, 5, 25], as illustrated in Figure 1. This regime is common in practice because high-quality trajectories are often substantially more expensive to collect than ordinary ones [21, 22, 28].

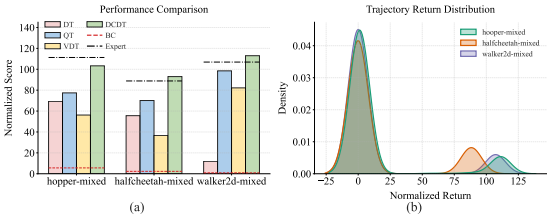


Figure 1: (a) Normalized scores of DCDT against baselines on imbalanced datasets. (b) Density plots of trajectory returns, highlighting the severe multi-modality of these datasets.

To mitigate this limitation, recent approaches integrate value-based critics into the vanilla behavioral cloning objective of DT, explicitly or implicitly guide the stationary distribution of the learned policy towards actions with higher returns [25, 65]. Typically, these methods first learn the Q-function via Temporal-Difference (TD) backup on the offline dataset. Then, the Q-value maximization term is augmented into the likelihood loss, implicitly biasing the policy towards high-return trajectories present in the dataset. Besides, advantage function based weighted regression has also been adopted to explicitly correct the data distribution [65]. However, when the dataset is highly skewed, standard TD learning yields Q-values that are themselves skewed, making them prone to severe extrapolation errors and collapse [14, 33, 34, 48]. Consequently, these approaches still face a fundamental limitation in imbalanced offline RL: the critic used to guide policy learning is itself estimated from a distribution dominated by suboptimal trajectories.

In this paper, a novel dual-weighted TD method is proposed to learn a Q-function that aligns the optimal one. In imbalanced datasets, the empirical occupancy d^U is dominated by suboptimal trajectories, causing standard TD learning to focus on frequently observed but low-quality regions. We first demonstrate that, if the optimal stationary distribution d^{π^*} is known, one can refine the Q-function by minimizing the Bellman error with a weighted norm, where the TD error of the state-action pair (s, a) in the offline dataset is weighted by the importance weight ratio $w^*(s, a) = d^{\pi^*}(s, a)/d^U(s, a)$. As estimating these weights is generally difficult, we then study the Lagrangian duality of the constraint reward maximization objective with respect to the policy occupancy density, and find the approximate optimal ratio by solving the equivalent dual convex optimization problem. In addition to rectifying the TD learning, this ratio is also suitable for adjusting the data distribution in the behavior cloning objective. This density-based weight adjustment approach is more robust than existing heuristic advantage weight strategy, particularly in imbalanced scenarios. Accordingly, we designate our proposed framework as the Distribution Corrected Decision Transformer (DCDT). In summary, our contributions are three-fold:

- We formulate a novel dual optimization objective to robustly estimate optimal density ratios from imbalanced data. In particular, a double-KL-regularized reward maximization problem with the Bellman flow constraint is constructed for an infinite horizon Markov decision process. The two KL terms respectively encourage alignment with the high-quality occupancy distribution and constrain the solution to the support of the offline data, thereby mitigating extrapolation. Moreover, the resulting dual problem admits a closed-form solution, enabling efficient computation of the approximate optimal ratio.
- We introduce the Distribution Corrected Decision Transformer (DCDT), a unified framework that implements a dual-level correction. DCDT utilizes weighted temporal-difference learning to stabilize policy value evaluation against distribution shift induced by suboptimal-dominated data, while simultaneously refining policy improvement by prioritizing high-quality state-action pairs through a weighted behavior cloning objective. Notably, DCDT remains well-defined even when no clearly distinguishable high-quality subset is available, in which case one KL term vanishes while the framework retains a valid and effective form.
- We provide a theoretical analysis establishing performance guarantees. We show that the proposed weighted TD-based Q-iteration converges to the optimal Q-function under the task-specific optimal-occupancy-weighted norm, in contrast to standard Q-learning which is analyzed under the task-agnostic supremum norm. Consequently, the proposed method,

which integrates weighted behavior cloning with value regularization, achieves improved cumulative rewards compared to existing value-regularized DT approaches.

We demonstrate that DCDT is robust across highly imbalanced offline RL benchmarks, consistently outperforming state-of-the-art baselines. Across the imbalanced Gym and Adroit benchmarks, DCDT achieves the best average performance in all six dataset settings, and improves the average normalized score by 33% over existing SOTA, highlighting the effectiveness of distribution correction under severe data imbalance, even when high-quality data constitutes as little as approximately 3% of the offline dataset.

2 Preliminary

2.1 Decision Transformer

Unlike the majority of prior RL approaches that estimate value functions or compute policy gradients, DT-based methods [7, 23, 24, 62] cast policy learning as a return-conditioned supervised learning problem, predicting desired future actions from a history of states $\mathbf{s}_t$, actions $\mathbf{a}_t$, and return-to-go $\hat{r}_t$. The return-to-go token quantifies the cumulative reward from the current time step to the end of the episode. During training with offline collected data, DT processes a trajectory sequence τ_t in an auto-regressive manner which encompasses the most recent K-step historical context:

$$\tau_t = (\hat{r}_{t-K+1}, \mathbf{s}_{t-K+1}, \mathbf{a}_{t-K+1}, \ldots, \hat{r}_t, \mathbf{s}_t, \mathbf{a}_t).$$

The prediction head associated with a state token $\mathbf{s}_t$ is trained to predict the corresponding action $\mathbf{a}_t$. Regarding continuous action spaces, the training objective is to minimize the mean-squared loss:

$$\mathcal{L}_{DT}(\pi) = \mathbb{E}_{\tau_t \sim \mathcal{D}} \left[\frac{1}{K} \sum_{i=t-K+1}^{t} (\mathbf{a}_i - \pi(\tau_t)_i)^2 \right],$$

where $\pi(\tau_t)_i$ denotes the i-th action output of the Transformer policy.

To address the challenge of imbalanced data distributions and align expected returns with optimal outcomes, QT [25, 63] incorporates a Q-value module during the training phase, enabling the preferential sampling of high-value actions. The final policy learning objective emerges as a linear combination of policy regularization and policy improvement elements:

$$\arg\min_{\pi} \mathcal{L}_{DT}(\pi) - \eta \cdot \underbrace{\mathbb{E}_{\tau_t \sim \mathcal{D}} \mathbb{E}_{(\mathbf{s}_i, \mathbf{a}_i) \sim \tau_t} Q_\phi(\mathbf{s}_i, \pi(\tau_t)_i)}_{\mathcal{L}_Q(\pi)},$$

where η is a hyper-parameter that balances the two loss terms, and Q_ϕ is the approximated Q-function. Besides, VDT [65] additionally applies advantage-weighted regression to the behavioral cloning loss, explicitly biasing the policy towards higher returns.

While aiming to steer the policy towards optimality, the Q-function Q_ϕ is usually derived from standard TD learning on the static offline dataset. This approach attempts to minimize the Bellman error over the dataset distribution d^U using Bellman operator B:

$$\begin{aligned} \min_{Q} \quad & \mathbb{E}_{(s,a,s') \sim d^U} \left[\left(Q(s,a) - BQ(s,a) \right)^2 \right], \\ \text{where} \quad & BQ(s,a) := R(s,a) + \gamma \max_{a'} Q(s', a'). \end{aligned} \tag{1}$$

However, this approach is highly susceptible to distribution shift, often yielding biased or collapsed Q-values when the data is imbalanced [14, 34, 48]. Consequently, these methods face a paradox: they attempt to guide the policy using an unreliable compass. In contrast, we propose a refined distribution correction framework. By deriving robust optimal importance weights from a dual formulation, we rectify the distribution shift in *both* the underlying value estimation and the final policy extraction, ensuring both components are grounded in the optimal stationary distribution.

2.2 Offline Reinforcement Learning

We model the agent-environment interaction as an infinite-horizon Markov Decision Process (MDP)[53] defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T}, \mu_0, \gamma)$. The normalized state-action occupancy

distribution $d^\pi(s,a)$ for a policy π is defined as:

$$d^\pi(s,a) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t P(s_t = s, a_t = a|\mu_0, \pi, \mathcal{T}).$$

In the imbalanced dataset setting, the return heterogeneity of the offline data $\mathcal{D}^U$ naturally induces a relatively high-quality reference subset denoted as $\mathcal{D}^E$[2], with the remaining data forming the imperfect subset $\mathcal{D}^I$. We regard $\mathcal{D}^E$ as sampled from a relatively high-return stationary distribution $d^E(s,a)$. Let $\mathcal{D}^U = \mathcal{D}^E \cup \mathcal{D}^I$ denote the full offline dataset with distribution d^U.

For a given MDP, the standard RL objective is defined as the reward-maximizing linear program $\max_\pi \mathbb{E}_{(s,a)\sim d^\pi(s,a)}[R(s,a)]$. While this objective is defined over the policy space, an equivalent optimization problem w.r.t. the occupancy density $d = d^\pi$ can be formulated as follows.

$$\max_d \mathbb{E}_{(s,a)\sim d(s,a)}[R(s,a)], \tag{2}$$

$$\text{s.t. } (\mathcal{B}_* d)(s) = (1-\gamma)\mu_0(s) + \gamma(\mathcal{T}_* d)(s), \ \ \forall s, a, \tag{3}$$

$$d(s,a) \geq 0 \ \ \forall s, a, \tag{4}$$

where $(\mathcal{B}_* d)(s) := \sum_{\bar{a}} d(s,\bar{a})$ is a marginalization operator, and $(\mathcal{T}_* d)(s) := \sum_{\bar{s},\bar{a}} T(s|\bar{s},\bar{a})d(\bar{s},\bar{a})$ is a transposed Bellman operator. Here, the equations (3-4) are often referred as the Bellman flow constraint. Equivalently, the discounted state-action occupancy measure satisfies $d(s,a) = (1-\gamma)\mu_0(s)\pi(a|s) + \gamma\sum_{\bar{s},\bar{a}} d(\bar{s},\bar{a})T(s|\bar{s},\bar{a})\pi(a|s), \ \ \forall s, a$.

In the offline setting, however, we do not have access to the environment dynamics to sample from arbitrary distributions d. Instead, we are restricted to a static dataset D^U collected by a behavior policy π^U, with the empirical distribution d^U. By applying Lagrangian duality [4] and performing a change of variables to optimize over the density ratio $w(s,a) := \frac{d(s,a)}{d^U(s,a)}$, problem (2) can be reformulated into a tractable unconstrained optimization problem:

$$\begin{aligned}\min_\nu \max_{w\geq 0} L(\nu, w) := (1-\gamma)\mathbb{E}_{s\sim\mu_0}[\nu(s)] \\ + \mathbb{E}_{(s,a)\sim d^U}\left[w(s,a)(R(s,a) + \gamma(\mathcal{T}\nu)(s,a) - \nu(s))\right],\end{aligned} \tag{5}$$

where ν is the dual value function. The corresponding optimal policy can be recovered by re-weighting the behavior distribution d^U using the learned optimal density ratios w^*.

Practically, the unconstrained optimization in (5) is prone to failure, often yielding highly suboptimal policies that drift outside the dataset support with a constant probability [51, 61, 64]. To mitigate this, empirical methods typically utilize the Kullback-Leibler (KL) penalty to anchor the learned policy to the behavior distribution [26, 58, 60], and consider the following regularized optimization problem:

$$\max_\pi \mathbb{E}_{(s,a)\sim d^\pi(s,a)}[R(s,a)] - D_{KL}(d^\pi \| d^U).$$

We argue that a single regularization term is insufficient for highly imbalanced datasets, as it creates a rigid trade-off between encouraging deviation towards optimality and enforcing adherence to the suboptimal data support. To address this, we propose a novel formulation that optimizes the density ratio w with double-KL penalties, explicitly decoupling the goal of alignment with the high-quality reference subset from the need to respect data-support constraints. The resulting estimator is developed in Section 3.2.

Moreover, prior works [37, 47, 52] restrict the application of these estimated weights to the policy extraction phase—specifically, by utilizing them solely for weighted behavioral cloning. Consequently, these approaches overlook the potential of density ratios to simultaneously correct distribution shift during TD-based value estimation. To the best of our knowledge, our method is the first to leverage these weights for a dual-level correction, applying them to rectify both the TD learning and the policy extraction phase.

[2]Throughout this paper, the superscript E denotes a *relatively high-return partition* of the offline data. In practice, $\mathcal{D}^E$ can be obtained by ranking trajectories by cumulative return and selecting the top-K. The only requirement is that its average return is higher than that of the remaining samples in the imbalanced dataset. When expert demonstrations are available, such as in autonomous-driving applications, they can also be directly used as $\mathcal{D}^E$.



(a) Behavior policy (b) Filtered BC (c) Single KL $\alpha_1 = 0$ (d) Single KL $\alpha_2 = 0$ (e) Double KL

Figure 2: Stationary distribution correction on the Four Rooms task **(a) Behavior Policy (π^U):** The stationary distribution of the data-collection policy. **(b) Filtered BC Policy ($\hat{d}^U$):** BC trained on high-return trajectories. Despite filtering, it fails to stitch a coherent strategy, yielding fragmented visitation. **(c) No High-return Alignment($\hat{d}^{\pi^*}_{\alpha_2}$):** Estimates weights without optimality incentives; fails to isolate the goal path.**(d)No Support Constraint($\hat{d}^{\pi^*}_{\alpha_1}$):** Estimates weights without support regularization; finds the goal but fails to stitch together the alternative feasible route from noisy data. **(e)Double KL($\hat{d}^{\pi^*}_{\alpha_1,\alpha_2}$):** By combining the two complementary constraints, our method recovers two forms of useful structure: the high-return alignment term induces a path that follows high-quality data, while the support constraint term enables the stitching of additional feasible routes from noisy data.

3 Methodology

We develop the Distribution Corrected Decision Transformer (DCDT) in three stages that move from an oracle analysis to a practical training objective: **(1) Oracle-weight analysis** (Section 3.1). Assuming the optimal density ratios $w^*(s,a) = d^{\pi^*}(s,a)/d^U(s,a)$ are given, we introduce *Distribution Corrected TD Learning* and prove it improves over standard TD learning (Theorem 3.1). **(2) Augmented DCDT objective** (Section 3.1). Using the corrected value function Q_{w^*}, we construct a weighted sequence-modeling objective in which the *same* w^* also reweights the behavioral-cloning term, and show that the fully corrected method dominates every partially corrected baseline (Theorem 3.3). **(3) Tractable regularized estimator** (Section 3.2). Because w^* is not available in practice, we derive a double-KL regularized objective whose closed-form solution w^*_α yields a tractable estimator well suited to highly imbalanced datasets. The training procedure is summarized as Algorithm 2 in Appendix F.

3.1 Distribution Corrected Learning

Standard offline Q-learning(1) minimizes the Bellman error under the empirical dataset distribution d^U. In imbalanced offline datasets, this distribution is dominated by suboptimal trajectories, while high-quality trajectories that better approximate the behavior of the optimal policy are scarce. Consequently, TD learning is biased toward frequently observed but low-quality regions of the dataset, leading to inaccurate value estimates for the optimal policy π^* when d^U deviates substantially from d^{π^*}. To mitigate this, we propose the Dual Weighted TD Learning procedure. We assume that the optimal stationary distribution d^* and the optimal density ratios $w^*(s,a) = d^{\pi^*}(s,a)/d^U(s,a)$ are available, and train the Q-function by minimizing the Bellman error weighted by $w^*(s,a)$ as follows.

$$Q_{k+1} \leftarrow \arg\min_Q \mathbb{E}_{(s,a)\sim d^U}\left[w^*(s,a)(BQ_k(s,a) - Q(s,a))^2\right],$$
$$\text{where } BQ_k(s,a) = R(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{T}(s,a)} \max_{a'} Q_k(s',a'). \tag{6}$$

We now analyze the theoretical properties of this update. We define the d^*-weighted l_2 norm as $\|Q\|^2_{d^*} := \mathbb{E}_{(s,a)\sim d^*}[Q(s,a)^2] = \mathbb{E}_{(s,a)\sim d^U}[w^*(s,a)Q(s,a)^2]$.

Theorem 3.1 (Convergence of Weighted Q-Iteration)**.** *Let $\{Q_k\}_{k=0}^\infty$ be the sequence of Q-functions generated by the weighted regression update in (6). Assuming the weighted function approximation error is bounded at each iteration, i.e., $\|Q_{k+1} - BQ_k\|_{d^*} \le \epsilon$, then the sequence converges to a neighborhood of the optimal value function Q^*:*

$$\limsup_{k\to\infty} \|Q_k - Q^*\|_{d^*} \le \frac{\epsilon}{1-\sqrt{\gamma}}.$$

Remark 3.2. Standard analyses of Q-learning typically focus on supremum-norm (L_∞) error bounds [3], which require the value function to be accurate uniformly over the entire state-action space. In offline RL, this requirement can be overly demanding due to limited data coverage. Seminal work in approximate dynamic programming [2, 46] shows that uniform accuracy is not always

necessary; instead, it can be sufficient to control the Bellman error under a weighted norm $\|\cdot\|_{d^*}$. Our method builds on this foundation by explicitly reweighting the Bellman objective toward the support of the optimal policy. Compared with the task-agnostic supremum norm, the optimal-occupancy-weighted norm is task-relevant: it emphasizes regions that matter for the target optimal behavior rather than requiring uniform accuracy over poorly covered or irrelevant regions of the offline dataset.

We denote the Q function learned under weighted d^* norm as Q_{w^*}. Our objective combines weighted behavioral cloning to anchor the policy to the reference distribution d^E within the data support, and weighted value maximization to explicitly guide the policy towards higher returns. The final optimization problem is given by:

$$\begin{aligned}\pi_{DCDT} &= \arg\min_{\pi} \mathcal{L}(\pi) := \mathcal{L}_{WDT}(\pi) + \lambda \mathcal{L}_{Q_{w^*}}(\pi) \\ &= \mathbb{E}_{\substack{\tau_t \sim \mathcal{D} \\ (s_t, a_t) \sim \tau_t}} \Big[w^*(s_t, a_t) \|\pi(\tau_t) - a_t\|^2 - \lambda Q_{w^*}(s_t, \pi(\tau_t)) \Big].\end{aligned}$$

Theorem 3.3. *Let π^* be the optimal policy and π^U be the behavior policy. Assuming policy-class realizability (i.e., π^* lies in the candidate class), the following performance hierarchy holds in terms of the expected return:*

$$J(\pi_{DCDT}) \geq J(\pi_{QWDT}) \geq J(\pi_{QDT}) \geq J(\pi_{DT})$$

and

$$J(\pi_{DCDT}) \geq J(\pi_{WDT}) \geq J(\pi_{DT}).$$

Different π denote the policy learned according to different $\mathcal{L}_{DT}$/$\mathcal{L}_{WDT}$ and/or $\mathcal{L}_Q$/$\mathcal{L}_{Q_{w^*}}$ combinations. The exact definition of each π, along with the full proof, is provided in the appendix D.2. Theorem 3.3 theoretically validates that correcting distribution shift in both the $\mathcal{L}_{WDT}$ and $\mathcal{L}_{Q_{w^*}}$ components yields additive performance gains.

Remark (idealized vs. tractable weights). Theorems 3.1–3.3 use the oracle ratio $w^* = d^{\pi^*}/d^U$, which has no closed form and is unbounded whenever $d^U \to 0$. Section 3.2 replaces it by the tractable regularized estimator w^*_α from (8). This estimator is tractable, admits a closed-form solution (Proposition 3.5), and comes with a finite-sample sample-complexity bound(Appendix D.5).

In practice, we utilize the Clipped Double Q-learning technique [12, 20] with two Q-networks, Q_{ϕ_1} and Q_{ϕ_2}, and their respective target networks $Q_{\phi'_1}, Q_{\phi'_2}$ to mitigate function approximation error, namely overestimation bias and a high variance build-up [44, 49, 56]. Given a trajectory segment τ_t of length K, the optimization of ϕ_i (for $i = \{1, 2\}$) is performed by minimizing the following weighted objective:

$$\mathbb{E}_{\substack{\tau_t \sim \mathcal{D}^U \\ \hat{\mathbf{a}}_t \sim \pi_{\theta'}}} \left[\sum_{\substack{m= \\ t-K+1}}^{t-1} w_{\nu_\psi}(s_m, a_m) \|\hat{Q}_m - Q_{\phi_i}(\mathbf{s}_m, \mathbf{a}_m)\|^2 \right], \tag{7}$$

$$\text{where } \hat{Q}_m = \sum_{j=m}^{t-1} \gamma^{j-m} r_j + \gamma^{t-m} \min_{i=1,2} Q_{\phi'_i}(\mathbf{s}_t, \hat{\mathbf{a}}_t).$$

Here, γ is the discount factor, $\hat{\mathbf{a}}_t$ denotes the predicted action output by the target model $\pi_{\theta'}$, and w_{ν_ψ} represents the parameterized importance weight function, which will be detailed in Section 3.2.

3.2 Recover optimal weight

To robustly estimate the optimal density ratios w^* in the presence of imbalanced data, we propose a novel double-KL optimization framework. Unlike standard approaches that rely on a single divergence constraint, our formulation synergizes two complementary Kullback-Leibler terms: (1) a divergence minimization towards the sparse reference subset d^E to induce optimality, and (2) a regularization term anchored to the full offline dataset d^U to enforce support constraints. This dual-constraint mechanism ensures that the learned weights prioritize high-return transitions while remaining anchored to valid data support, thereby mitigating extrapolation error in offline RL.

We consider the following optimization problem w.r.t. the stationary distribution d:

$$\max_{d \geq 0} J(d) = \mathbb{E}_{(s,a) \sim d}[R(s, a)] - \alpha_1 D_{\mathrm{KL}}(d \| d^E) - \alpha_2 D_{\mathrm{KL}}(d \| d^U), \tag{8}$$

with the bellman flow constraints (3-4). Here $\alpha_1, \alpha_2 \geq 0$ are hyperparameter that control the balance between minimizing KL divergence with d^E and preventing the deviation of d^π from d^U.

Note that this optimization relies on on-policy samples acquired through environment interaction, presenting a significant challenge in offline settings where such interaction is prohibited. We circumvent this limitation by leveraging Lagrangian duality. The derivation below follows the standard Lagrangian-duality and change-of-variable recipe used throughout the DICE family [29, 37, 43]; we therefore state only the resulting tractable objective and defer the algebra to Appendix D.3.

Proposition 3.4. *The optimization problem (8) can be transformed into a tractable optimization in the offline setting:*

$$\max_{w \geq 0} \min_{\nu} L_\alpha(w, \nu) := (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^U}\Big[w(s,a)(A_\nu^{\alpha_1}(s,a) - (\alpha_1+\alpha_2)\log w(s,a))\Big], \tag{9}$$

where $\nu(s)$ is the Lagrange multipliers, $w(s,a) := \frac{d(s,a)}{d^U(s,a)}$, and $A_\nu^{\alpha_1}(s,a)$ is the augmented advantage using ν defined in Equation (56).

Since the optimization (8) is a convex optimization and the strong duality holds, the optimization is equal to $\min_\nu \max_{w\geq 0} L_\alpha(w,\nu)$.

Proposition 3.5. *The closed-form solution of the inner maximization of (3.2), $w_\nu^* := \arg\max_{w\geq 0} L(w,\nu)$, is*

$$w_\nu^*(s,a) = e^{A_\nu^{\alpha_1}(s,a)/(\alpha_1+\alpha_2)-1}, \quad \forall s,a. \tag{10}$$

Remark 3.6. When no high-quality subset is available, taking $\alpha_1 = 0$ eliminates the d^E-anchored KL term, and the closed-form ratio in (10) reduces to $w_\nu^*(s,a) = \exp\big(A_\nu^0(s,a)/(1+\alpha_2) - 1\big)$. This yields a purely reward-driven correction that does not rely on an explicit reference subset. Empirically, this fallback variant already outperforms VDT and QT on standard D4RL datasets (Appendix L).

By using $w_\nu^*(s,a)$, optimization (3.2) can be reduced to a simple convex minimization problem as follows:

$$\min_\nu L(w_\nu^*, \nu) = (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + (\alpha_1+\alpha_2)\mathbb{E}_{(s,a)\sim d^U}\left[e^{A_\nu^{\alpha_1}(s,a)/(\alpha_1+\alpha_2)-1}\right]. \tag{11}$$

Illustrative example. To validate the intuition behind our method and verify that the learned weights correctly identify optimal transitions within an imbalance dataset, we apply our framework to a simplified 2D navigation task. *Figure 2* outlines how our approach works in the Four Rooms domain [54] where the agent aims to navigate to a goal location in a maze composed of four rooms. We describe the detailed setting of this experiment in the appendix E.3. As clearly depicted in *Figure 2e*, the double-KL objective effectively filters clean high-return transitions from the reference subset and stitches high-value segments from noisy data to form multiple complete paths to the goal. In contrast, *Figure 2c* and *Figure 2d* fail to recover a complete strategy, either because they cannot stitch useful segments or because they fail to discover the alternative valid path from the noisy dataset.

4 Experiment

We evaluate DCDT on the offline RL regime constructed from D4RL [10], covering Gym-MuJoCo and Adroit tasks. We compare against three baseline families—offline imitation learning (BC, IQ-Learn [16], RCE [9], ORIL [66]), distribution correction methods (SMODICE [43], ReCOIL [52]), and DT-based critic-guided sequence modeling (QT [25], VDT [65]). Full environment details and baseline configuration are in Appendix E.1.

4.1 Main Results

To mimic severe data imbalance, we construct each training set by mixing a large amount of suboptimal D4RL data ("random" or "medium" for Gym-MuJoCo locomotion, and the standard partitions for Adroit manipulation) with a small set of expert trajectories (200 for "+Expert" and 30 for "+Few-Expert"). The normalized scores on Gym and Adroit tasks are reported in Table 1.

Table 1: Normalized average returns on the D4RL Gym and Adroit tasks. Results for DCDT correspond to mean and standard errors of normalized scores over 30 random rollouts for each tasks. Our method outperforms all prior methods by a clear margin in almost all domains, demonstrating significant robustness in the imbalanced dataset settings where prior imitation learning methods, DT-based methods and distribution correction methods often fail.

Gym Tasks	Env	BC(only expert data)	BC(full data)	IQ-Learn	RCE	ORIL	SMODICE	ReCOIL	QT	VDT	DCDT
random+ expert	hopper	4.52±1.42	5.64±4.83	1.85± 2.19	51.41 ± 38.63	73.93±11.06	101.61±7.69	**108.18±3.28**	77.43 ± 0.41	56.22 ± 1.89	97.85 ± 5.45
	halfcheetah	2.2±0.01	2.25± 0.00	4.83±7.99	64.19 ± 11.06	60.49±3.53	80.16±7.30	80.20± 6.61	70.09 ± 0.23	36.72 ± 2.01	**93.03±0.52**
	walker2d	0.8±0.61	0.91±0.5	0.57±0.09	20.90 ± 26.80	2.86±3.39	105.8±3.47	102.16 ± 7.19	98.52 ± 0.54	82.22 ± 1.73	**113.00±0.25**
	ant	5.17±5.43	30.66±1.35	42.23±20.05	105.38 ± 14.15	73.67±12.69	126.78±5.12	126.7±4.63	70.73 ± 4.09	75.70 ± 6.27	**135.23±0.84**
	Average	3.17	9.87	12.37	60.47	52.74	103.59	104.31	79.19	62.72	**109.78**
random+ few-expert	hopper	4.84±3.83	3.0±0.54	1.37 ±1.23	25.31±18.97	42.04±13.76	60.11±18.28	**97.85±17.89**	77.38 ± 0.42	61.51 ± 0.13	86.78±4.13
	halfcheetah	-0.93±0.35	2.24±0.01	1.14±1.94	2.99±1.07	2.84±5.52	2.28±0.62	76.92±7.53	60.90 ± 1.49	30.50 ± 0.72	**83.51 ± 1.11**
	walker2d	0.98±0.83	0.74±0.20	0.39±0.27	40.49±26.52	3.22±3.29	**107.18±1.87**	83.23±19.00	78.88 ± 0.42	76.68 ± 2.33	103.62±1.71
	ant	0.91±3.93	35.38±2.66	32.99±3.12	67.62±15.81	25.41 ± 8.58	-6.10±7.85	67.14± 8.30	31.47 ± 3.82	70.58 ± 4.00	**130.08± 0.41**
	Average	1.45	10.34	8.97	34.10	18.38	40.87	81.29	62.16	59.82	**101.00**
medium+ expert	hopper	16.09±12.80	59.25±3.71	12.90±24.00	58.71±34.06	61.68±7.61	49.74±3.62	88.51±16.73	111.1 ± 1.31	112.50±0.08	**113.03± 0.07**
	halfcheetah	-1.79±0.22	42.45±0.42	25.67±20.82	65.14±13.82	54.66±0.88	59.50±0.82	81.15±2.84	90.61 ± 0.52	91.60 ± 0.12	**92.84±0.05**
	walker2d	2.43±1.82	72.76±3.82	59.37±30.14	96.24±14.04	8.19±7.70	2.62±0.93	108.54±1.81	101.04 ± 0.56	101.93 ± 0.94	**109.12 ± 0.44**
	ant	0.86±7.42	95.47±10.37	37.17±41.15	86.14±38.59	102.74±6.63	104.95±6.43	120.36±7.67	99.55 ± 4.15	122.84± 5.63	**134.68±0.41**
	Average	4.40	67.48	33.78	76.56	56.82	54.20	99.64	100.58	107.22	**112.42**
medium+ few-expert	hopper	7.37±1.13	46.87±5.31	11.05±20.59	66.15±35.16	17.40±15.15	47.61±7.08	50.01±10.36	96.16± 2.11	107.61 ± 0.18	**110.88±0.58**
	halfcheetah	-1.15±0.06	42.21±0.06	26.27±20.24	61.14±18.31	43.24±0.75	46.45±3.12	75.96±4.54	80.8 ± 0.42	80.52 ± 2.06	**82.36±6.41**
	walker2d	2.02±0.72	70.42±2.86	73.30±2.85	85.28±34.90	6.81±6.76	6.00±6.69	91.25±17.63	81.14± 0.93	90.11 ± 1.64	**109.12±0.58**
	ant	-10.45±1.63	81.63±6.67	35.12±50.56	67.95±36.78	81.53±8.618	81.53±8.618	110.38±10.96	94.91 ± 6.73	101.33±8.41	**128.44±1.29**
	Average	-0.55	60.28	36.44	70.13	37.25	45.40	81.90	88.25	94.89	**107.70**
Adroit Tasks	**Env**	**BC(only expert data)**	**BC(full data)**	**IQ-Learn**	**RCE**	**ORIL**	**SMODICE**	**ReCOIL**	**QT**	**VDT**	**DCDT**
cloned+ expert	pen	13.95±11.04	34.94±11.10	2.18±8.75	19.60±11.40	-3.10±0.40	-3.36±0.71	95.04±4.48	104.51 ± 3.42	93.98 ± 4.01	**165.45±1.36**
	door	-0.22±0.05	0.011±0.00	0.07±0.02	0.08± 0.15	-0.33±0.01	0.25±0.54	102.75±4.05	70.24 ± 2.38	97.86 ± 0.69	**106.46±0.17**
	hammer	2.41±4.48	5.45±7.84	0.27±0.02	1.95±3.89	0.25± 0.01	0.15±0.078	95.77±17.90	89.78 ± 2.30	126.49 ± 1.62	**130.08±0.39**
	relocate	-0.17±0.04	-0.24±0.01	-0.1±0.12	-0.25±0.04	-0.29±0.01	1.75±3.85	67.43±14.60	38.9 ± 1.57	21.20 ± 2.10	**100.79±2.61**
	Average	3.99	10.04	0.60	5.35	-0.87	-0.30	90.25	75.86	84.88	**125.70**
human+ expert	pen	13.83±10.76	90.76±25.09	14.29±28.82	17.81±5.91	-3.38±2.29	-2.20±2.40	103.72±2.90	145.90 ± 2.68	152.70 ± 4.27	**160.21±2.37**
	door	-0.03±0.05	103.71±1.22	5.6±7.29	-0.05±0.05	-0.33±0.01	-0.20±0.11	104.70±0.55	81.89 ± 1.73	104.29 ± 0.59	**105.08±0.70**
	hammer	0.18±0.14	122.61±4.85	5.32±1.38	5.00±5.64	1.89±0.70	-0.07±0.39	125.19±3.29	91.73 ± 2.13	125.52 ± 0.30	**126.85±1.00**
	relocate	-0.13±0.11	81.19±7.73	-0.04±0.22	0.02±0.10	-0.29±0.01	-0.16±0.04	91.98±2.89	103.9 ± 0.94	84.37± 2.31	**106.84±1.63**
	Average	3.46	99.57	6.29	5.70	-0.53	-0.66	106.40	105.86	116.72	**124.75**

Results for DCDT on the no-mixture original D4RL datasets are provided in Table 9 of Appendix K, where the top-200 trajectories are selected as the high-quality reference subset. Taken together, these results indicate that DCDT performs strongly on the imbalanced regimes targeted in this paper and remains robust across different dataset compositions. Overall, DCDT attains the highest average performance across all six imbalanced dataset settings, indicating strong robustness under severe data imbalance. Relative to QT, DCDT achieves a 33% improvement in average normalized score across these imbalanced settings, with particularly strong robustness in the most challenging cases (random+few expert) where high-quality data constitute only about 3% of the dataset.

Performance on Imbalanced Manipulation Tasks. The Adroit "Human + Expert" tasks provide a challenging testbed because human demonstrations introduce substantial behavioral variability and structural mismatch with expert trajectories. On this suite, DCDT attains the highest average score (125.7), exceeding both the distribution-correction baseline and the value-guided DT baseline. This result provides evidence that the dual-correction framework transfers effectively to high-dimensional, contact-rich manipulation tasks.

Robustness to Data Composition. We next examine performance when both dataset quality and expert quantity are severely degraded. In the challenging Gym "Random + Few-Expert" setting (Table 1, second block), standard DT-based methods deteriorate markedly, with QT achieving an average score of only 62.16, whereas DCDT maintains a substantially higher score of 101.00. This result suggests that the learned density-ratio weights help isolate useful high-return structure even when expert data are heavily diluted by random trajectories.

4.2 Ablation Studies

Contribution of algorithmic components. We ablate different components of DCDT across four distinct tasks, resulting in five variants (including DCDT), as detailed in Table 2. As shown in Table 2, we observe a clear additive benefit. While WDT improves upon the baseline by filtering suboptimal transitions and QDT improves performance via value maximization, the combination (DCDT) achieves the highest scores. This confirms that distribution correction and value maximization are complementary: w^* ensures the policy stays within the support of optimal transitions, while Q-regularization guides the policy towards higher returns within that support.

Table 2: Ablation of algorithmic components across distinct tasks on the random+expert dataset setting. The consistency of the improvement confirms that both distribution correction and dual-weighted Q-estimation are essential.

Method	Components			Normalized score				
	Policy w^*	Q-Reg	Weighted Q	Hopper	HalfCh.	Ant	Walk.	Average
(1) DT	×	×	×	69.3	55.6	89.0	11.9	56.5
(2) WDT	✓	×	×	77.3	65.3	130.7	110.8	96.0
(3) QDT	×	✓	×	77.4	70.1	131.2	98.5	94.3
(4) QWDT	✓	✓	×	88.9	92.3	134.3	111.2	106.7
(5) DCDT	✓	✓	✓	**97.9**	**93.0**	**135.2**	**113.0**	**109.8**

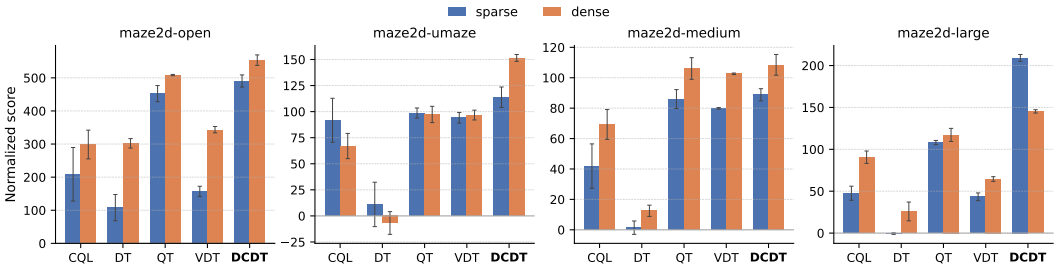


Figure 3: Maze2D stitching ability across four mazes (open/umaze/medium/large) under sparse and dense reward. Each panel shows five methods, and DCDT wins on every maze.

Dual vs. primal weight estimation. We validate our dual-derived importance weights (8) against standard advantage weights($w_{\text{primal}} \propto \exp((Q(s,a) - V(s))/\tau)$) used in prior works [50]. In the challenging setting of highly-imbalanced datasets, we find that primal weights exhibit significant instability; due to the dominance of low-quality data, the learned value function is prone to collapse, yielding noisy weights that fail to effectively isolate the sparse optimal trajectories. In contrast, our dual formulation leverages the Bellman flow constraint to robustly identify the optimal stationary distribution. This structural advantage translates to a 17.4% average performance improvement over the primal baseline across varying tasks(Table 3).

Table 3: Weight-estimation comparison on "Random+Few Expert". Primal: $w \propto \exp((Q-V)/\tau)$; Dual: our convex objective.

Task	Primal	Dual	Δ
hopper-r+e	54.9	**77.3**	+40.8%
walker2d-r+e	89.5	**110.8**	+23.9%
ant-r+e	126.3	**130.7**	+ 3.5%
hopper-m+e	108.7	**113.0**	+ 4.0%
walker2d-m+e	99.1	**108.9**	+ 9.6%
ant-m+e	120.1	**133.9**	+11.5%
pen-cloned+e	132.0	**170.0**	+28.8%
door-cloned+e	90.9	**106.2**	+16.8%
Avg.	102.7	**118.9**	**+17.4%**

Sparse reward and stitching ability. The Maze2D domain provides a challenging benchmark for evaluating whether offline RL algorithms can *stitch* suboptimal trajectory segments into near-optimal paths [10]. We construct an imbalanced dataset by mixing 20 expert trajectories with 100 random trajectories. We evaluate performance across four maze configurations of increasing complexity—Open, Umaze, Medium, and Large—under two reward settings: *sparse* and *dense*. The results are summarized in Figure 3. While CQL and QT demonstrate relatively strong performance, the standard DT struggles significantly due to its inherent inability to perform stitching. Although VDT improves upon DT, it lags behind QT; we conjecture that VDT's reliance on advantage-based re-weighting leads to policy collapse, particularly in sparse reward scenarios where advantage estimates are prone to high bias. Conversely, DCDT excels across all tasks, confirming that our dual-weighting mechanism effectively endows the Transformer with stitching capabilities.

Additional experiments. (i) **Hyperparameter Sensitivity.** The main hyperparameters of DCDT are η, α_1, and α_2. Following prior work[25, 29], we fix α_1=1 and adopt the reported η of QT obtained by grid-search, tuning only α_2. The hyperparameter sensitivity results in Appendix M show that DCDT is robust to α_2, with α_2=0.05 serving as a near-universal default and stable performance maintained across a wide range of values. (ii) **Role of the reference-subset penalty.** We evaluate the performance of DCDT on datasets with different imbalanced level (Appendix L), DCDT with only

the support-constrained KL($\alpha_1 = 0$) already outperforms QT on standard D4RL Gym tasks, and the importance of the reference-subset penalty increases with growth of the dataset's imbalanced level; (iii) **Runtime and memory.** As shown in Table 6 (Appendix I), DCDT incurs only modest overhead, adding approximately 6% in wall-clock time and 1.4% in peak memory compared to QT.

5 Conclusion

In this work, we proposed Distribution Corrected Decision Transformer(DCDT) to address the performance degradation of Decision Transformers on imbalanced offline data. By leveraging Lagrangian duality, we derived optimal correction weights to simultaneously stabilize value evaluation and refine policy improvement towards high-likelihood high-quality transitions. Theoretical analysis guarantees our method's superiority over SOTA baselines. The empirical results confirm state-of-the-art performance on the mixed D4RL datasets, particularly in highly imbalanced regimes.

Acknowledgments and Disclosure of Funding

This work receives financial supports from the National Natural Science Foundation of China (Grant No. 72371014), and supported by Shanghai Key Laboratory of Intelligent Information Processing, Fudan University (Grant No. IIPL-2025-RD1-05)

References

[1] 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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[2] Dimitri P Bertsekas. Weighted sup-norm contractions in dynamic programming: A review and some new applications. *Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. LIDS-P-2884*, page 5, 2012.

[3] Dimitri P Bertsekas and John N Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE conference on decision and control*, volume 1, pages 560–564. IEEE, 1995.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[5] David Brandfonbrener, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. When does return-conditioned supervised learning work for offline reinforcement learning? *Advances in Neural Information Processing Systems*, 35:1542–1553, 2022.

[6] Yevgen Chebotar, Quan Vuong, Karol Hausman, Fei Xia, Yao Lu, Alex Irpan, Aviral Kumar, Tianhe Yu, Alexander Herzog, Karl Pertsch, et al. Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In *Conference on Robot Learning*, pages 3909–3928. PMLR, 2023.

[7] 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.

[8] Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*, 2021.

[9] Ben Eysenbach, Sergey Levine, and Russ R Salakhutdinov. Replacing rewards with examples: Example-based policy search via recursive classification. *Advances in Neural Information Processing Systems*, 34:11541–11552, 2021.

[10] 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.

[11] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

[12] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[13] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

[14] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pages 2052–2062. PMLR, 2019.

[15] Scott Fujimoto, David Meger, and Doina Precup. A deep reinforcement learning approach to marginalized importance sampling with the successor representation. In *International Conference on Machine Learning*, pages 3518–3529. PMLR, 2021.

[16] Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34: 4028–4039, 2021.

[17] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International conference on machine learning*, pages 2160–2169. PMLR, 2019.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[19] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[20] Hado Hasselt. Double q-learning. *Advances in neural information processing systems*, 23, 2010.

[21] Zhang-Wei Hong, Pulkit Agrawal, Rémi Tachet des Combes, and Romain Laroche. Harnessing mixed offline reinforcement learning datasets via trajectory weighting. *arXiv preprint arXiv:2306.13085*, 2023.

[22] Zhang-Wei Hong, Aviral Kumar, Sathwik Karnik, Abhishek Bhandwaldar, Akash Srivastava, Joni Pajarinen, Romain Laroche, Abhishek Gupta, and Pulkit Agrawal. Beyond uniform sampling: Offline reinforcement learning with imbalanced datasets. *Advances in Neural Information Processing Systems*, 36:4985–5009, 2023.

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

[24] Shengchao Hu, Li Shen, Ya Zhang, and Dacheng Tao. Prompt-tuning decision transformer with preference ranking. *arXiv preprint arXiv:2305.09648*, 2023.

[25] Shengchao Hu, Ziqing Fan, Chaoqin Huang, Li Shen, Ya Zhang, Yanfeng Wang, and Dacheng Tao. Q-value regularized transformer for offline reinforcement learning. In *International Conference on Machine Learning*, pages 19165–19181. PMLR, 2024.

[26] Xiaohan Hu, Yi Ma, Chenjun Xiao, Yan Zheng, and HAO Jianye. Iteratively refined behavior regularization for offline reinforcement learning. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*, 2023.

[27] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021.

[28] Li Jiang, Sijie Cheng, Jielin Qiu, Haoran Xu, Wai Kin Chan, and Zhao Ding. Offline reinforcement learning with imbalanced datasets. *arXiv preprint arXiv:2307.02752*, 2023.

[29] Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2022.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, pages 5774–5783. PMLR, 2021.

[32] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

[33] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

[34] 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.

[35] Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *Journal of machine learning research*, 4(Dec):1107–1149, 2003.

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

[37] Jongmin Lee, Wonseok Jeon, Byungjun Lee, Joelle Pineau, and Kee-Eung Kim. Optidice: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning*, pages 6120–6130. PMLR, 2021.

[38] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[39] Jianxiong Li, Xianyuan Zhan, Haoran Xu, Xiangyu Zhu, Jingjing Liu, and Ya-Qin Zhang. When data geometry meets deep function: Generalizing offline reinforcement learning. *arXiv preprint arXiv:2205.11027*, 2022.

[40] Jianxiong Li, Xiao Hu, Haoran Xu, Jingjing Liu, Xianyuan Zhan, Qingyong Jia, and Ya-Qin Zhang. Mind the gap: Offline policy optimization for imperfect rewards. In *International Conference on Learning Representations (ICLR)*, 2023.

[41] Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *Advances in neural information processing systems*, 31, 2018.

[42] Jiafei Lyu, Xiaoteng Ma, Xiu Li, and Zongqing Lu. Mildly conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35, 2022.

[43] Yecheng Ma, Andrew Shen, Dinesh Jayaraman, and Osbert Bastani. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pages 14639–14663. PMLR, 2022.

[44] Shie Mannor, Duncan Simester, Peng Sun, and John N Tsitsiklis. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

[45] Rémi Munos. Error bounds for approximate policy iteration. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, pages 560–567, 2003.

[46] Rémi Munos. Performance bounds in l_p-norm for approximate value iteration. *SIAM journal on control and optimization*, 46(2):541–561, 2007.

[47] Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. DualDICE: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[48] Georg Ostrovski, Pablo Samuel Castro, and Will Dabney. The difficulty of passive learning in deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 23283–23295, 2021.

[49] Mark D Pendrith, Malcolm RK Ryan, and CC Sammut. *Estimator variance in reinforcement learning: Theoretical problems and practical solutions*. University of New South Wales, School of Computer Science and Engineering, 1997.

[50] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

[51] Paria Rashidinejad, Hanlin Zhu, Kunhe Yang, Stuart Russell, and Jiantao Jiao. Optimal conservative offline rl with general function approximation via augmented lagrangian. *arXiv preprint arXiv:2211.00716*, 2022.

[52] Harshit Sikchi, Qinqing Zheng, Amy Zhang, and Scott Niekum. Dual rl: Unification and new methods for reinforcement and imitation learning. In *The Twelfth International Conference on Learning Representations*, 2023.

[53] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[54] Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 1999.

[55] Shengpu Tang and Jenna Wiens. Model selection for offline reinforcement learning: Practical considerations for healthcare settings. In *Machine Learning for Healthcare Conference*, pages 2–35. PMLR, 2021.

[56] Sebastian Thrun and Anton Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the 1993 connectionist models summer school*, pages 255–263. Psychology Press, 2014.

[57] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[58] Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of kl regularization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:12163–12174, 2020.

[59] Yuanfu Wang, Chao Yang, Ying Wen, Yu Liu, and Yu Qiao. Critic-guided decision transformer for offline reinforcement learning. *arXiv preprint arXiv:2312.13716*, 2023.

[60] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

[61] Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021.

[62] Taku Yamagata, Ahmed Khalil, and Raul Santos-Rodriguez. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. In *International Conference on Machine Learning*, pages 38989–39007. PMLR, 2023.

[63] Kai Yan, Alex Schwing, and Yu-Xiong Wang. Reinforcement learning gradients as vitamin for online finetuning decision transformers. *Advances in Neural Information Processing Systems*, 37:38590–38628, 2024.

[64] Wenhao Zhan, Baihe Huang, Audrey Huang, Nan Jiang, and Jason Lee. Offline reinforcement learning with realizability and single-policy concentrability. In *Conference on Learning Theory*, pages 2730–2775. PMLR, 2022.

[65] Hongling Zheng, Li Shen, Yong Luo, Deheng Ye, Shuhan Xu, Bo Du, Jialie Shen, and Dacheng Tao. Value-guided decision transformer: A unified reinforcement learning framework for online and offline settings. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[66] Konrad Zolna, Alexander Novikov, Ksenia Konyushkova, Caglar Gulcehre, Ziyu Wang, Yusuf Aytar, Misha Denil, Nando De Freitas, and Scott Reed. Offline learning from demonstrations and unlabeled experience. *arXiv preprint arXiv:2011.13885*, 2020.

A Limitation and Future Work

All experiments on DCDT are conducted on proprioceptive D4RL benchmarks. Pixel-based observations, long-horizon manipulation tasks beyond FrankaKitchen, and safety-constrained settings remain untested.

The framework suggests three natural extensions. (i) *Vision-based offline RL.* The density-ratio estimator is architecture-agnostic; an MLP variant is reported in the supplementary material. Therefore, extending DCDT to pixel inputs mainly requires coupling it with a Transformer or CNN encoder trained on image sequences. (ii) *Online fine-tuning.* The regularized weight w_α^* and the discriminator c_κ can both be updated using fresh online samples. However, how the reference subset should be expanded or decayed during the online phase remains an open question. (iii) *Beyond KL.* Replacing the two KL penalties with more flexible f-divergences may improve robustness to misspecified or heavy-tailed reference distributions. Characterizing the corresponding closed-form solution is a concrete theoretical direction.

B Related Work

We discuss the literature most relevant to DCDT in three groups: offline RL under distribution shift, sequence-modeling approaches to offline RL, and DICE-style occupancy correction for offline control, including learning from mixed-quality data.

B.1 Offline RL with Distribution Shift

A fundamental challenge in offline RL is distribution shift: the learned policy may query state–action pairs that are weakly covered by the behavior dataset $\mathcal{D}^U$, which leads to unreliable value estimation. Existing methods typically address this issue in one of two ways. *Pessimistic value-regularization* methods, such as CQL [34], IQL [32], and Fisher-BRC [31], regularize the critic to suppress overestimation on unsupported actions. *Policy-constraint* methods, such as BCQ [13], BEAR [33], and TD3+BC [11], instead restrict the learned policy to remain close to the behavior policy. Although these approaches effectively control out-of-distribution action selection, they do not directly address a failure mode that becomes particularly severe under data imbalance: even within the support of $\mathcal{D}^U$, value learning can be dominated by the abundant low-return portion of the data. DCDT is motivated by this regime. Rather than imposing a direct policy-level constraint, DCDT performs occupancy-level correction through the density ratio $w^* = d^{\pi^*}/d^U$ and uses this correction to reweight both TD-based value learning and policy extraction. The resulting objective is therefore designed not only to prevent support mismatch but also to mitigate critic bias induced by skewed offline data.

B.2 Sequence-Modeling Approaches

A parallel line of work reformulates offline RL as supervised sequence modeling over trajectories. Decision Transformer (DT) [7] predicts actions conditioned on a target return, thereby recasting control as autoregressive sequence prediction. Trajectory Transformer [27] and Generalized Decision Transformer (GDT) [23] extend this formulation, while Critic-Guided DT (CGDT) [59], Prompt

DT (PTDT) [24], and Value DT (VDT) [65] enrich the training signal with additional value or prompt information. QT [25, 63] further augments DT-style learning with a Q-function that biases training toward high-value actions. Recent work has also studied sequence-model learning under skewed return distributions. RCSL [5] analyzes when return-conditioned supervised learning is effective, and return-rebalancing approaches [1] explicitly modify the training distribution to reduce domination by low-return trajectories. DCDT is closely related to this line of work, but differs in where the correction is applied. Prior methods primarily refine the sequence-model objective within the empirical distribution d^U. In contrast, DCDT first corrects the occupancy mismatch and then injects the same correction into both TD learning and sequence-model policy extraction. This distinction is important in our setting, since severe imbalance corrupts not only the behavioral-cloning objective but also the critic used for policy improvement.

B.3 DICE-Family Occupancy Correction and Mixed-Quality Data

A closely related line of work studies offline RL through density-ratio estimation and occupancy matching. DICE-family methods characterize distribution shift using ratios between stationary occupancies. DualDICE [47], GenDICE [41], and Off-Policy DICE [15] were originally developed for off-policy evaluation, whereas OptiDICE [37], SMODICE [43], and dual-RL [52] extend this perspective to offline control. A related method, *Mind the Gap* [40], also adopts a bi-level divergence-based formulation, although its goal is to correct imperfect rewards rather than imbalanced data distributions.

This line of work is also closely connected to learning from mixed-quality data. DemoDICE [29] and SMODICE [43] consider the regime in which a small high-quality dataset is combined with a much larger imperfect dataset, and perform occupancy matching with respect to a privileged high-quality reference source. DCDT is similar in motivation but differs in three respects. First, DCDT does not require privileged expert labels: the reference subset $\mathcal{D}^E$ only needs to be a *relatively* high-return partition of the offline data, and its median return can be substantially lower than that of a true expert dataset. Second, DCDT instantiates density-ratio correction within a *Decision Transformer* framework and propagates the resulting weights through both value learning and policy extraction, rather than using occupancy correction only for policy recovery. Third, the $\alpha_1{=}0$ variant shows that the reference subset serves as an optional inductive bias rather than a hard requirement: when meaningful quality stratification is unavailable, DCDT reduces to a purely reward-driven distribution-correction method. This distinction is important in practical offline RL, where rewards are often available but privileged expert annotations are not, and where the main challenge is to prevent the abundant low-quality portion of the data from dominating both critic learning and policy extraction.

C Additional Propositions for Section 3

We provide the sample complexity analysis of the optimization (11) under mild assumptions C.1,C.2,C.3. All involved constants are parameterized by $\alpha = \{\alpha_1, \alpha_2\}$ following the same treatment of [64].

Assumption C.1 (Realizability of $\mathcal{V}$, $\mathcal{W}$)**.** Suppose $\nu^*_\alpha \in \mathcal{V}, w^*_\alpha \in \mathcal{W}$, where $\mathcal{V}$ and $\mathcal{W}$ are function class.

Assumption C.2 (Boundedness of $\mathcal{W}$)**.** Suppose $B^{min}_{w,\alpha} \le w(s,a) \le B^{max}_{w,\alpha}$ for any $s \in \mathcal{S}, a \in \mathcal{A}, w \in \mathcal{W}$.

Assumption C.3 (Boundedness of $\mathcal{V}$)**.** Suppose $\|\nu^*_\alpha\|_\infty \le B_{\nu,\alpha}$ for any $s \in \mathcal{S}, \nu \in \mathcal{V}$.

Assumption C.4 (π^*_α-concentrability)**.**

$$\frac{d^*_\alpha(s,a)}{d^U(s,a)} \le B^{max}_{w,\alpha}, \forall s \in \mathcal{S}, a \in \mathcal{A}. \tag{12}$$

Definition C.5 (Statistical error of $\hat{L}_\alpha(\nu, w) - L_\alpha(\nu, w)$)**.**

$$\begin{aligned}\epsilon_{\text{stat}} &:= \mathcal{E}_{n,n_0,\alpha}(B_w, B_f, B_\nu, B_e) \\ &= (1-\gamma)\, B_{\nu,\alpha}\sqrt{\frac{2\log\big(4|\mathcal{V}|/\delta\big)}{n_0}} + \big((\alpha_1+\alpha_2)\, B_{f,\alpha} + B_{w,\alpha}B_{\nu,\alpha}\big)\cdot\sqrt{\frac{2\log\big(4|\mathcal{V}||\mathcal{W}|/\delta\big)}{n}}.\end{aligned} \tag{13}$$

Proposition C.6. *Suppose Assumptions C.1,C.2,C.3 (details in appendix)hold. Then with at least probability* $1-\delta$*,*

$$\|\hat{w} - w_\alpha^*\|_{2,d^U} \leq \sqrt{\frac{4B_{w,\alpha}^{max}\epsilon_{stat}}{\alpha_1+\alpha_2}}, \tag{14}$$

which implies that

$$\|\hat{d} - d_\alpha^*\|_1 \leq \sqrt{\frac{4B_{w,\alpha}^{max}\epsilon_{stat}}{\alpha_1+\alpha_2}}, \tag{15}$$

where $\hat{d}(s,a) = \hat{w}(s,a)d^U(s,a), \forall s,a.$

Remark C.7. Proposition C.6 establishes the finite-sample consistency of our estimator, confirming that $\hat{w}$ converges to the optimal regularized density ratio w_α^* as the statistical error ϵ_{stat} vanishes. Furthermore, Equation (15) demonstrates that our method effectively controls the Total Variation distance (via the L_1-norm) between the learned and optimal occupancy measures.

The proof is provided in Appendix D.5.

D Missing Proofs for Propositions and Theorems

D.1 Proof of Theorem 3.1

Assumption D.1 (No overestimation of the greedy value)**.** Every $Q \in \mathcal{Q}$ satisfies

$$\max_{a'} Q(s,a') \;\leq\; \max_{a'} Q^*(s,a') \qquad \text{for all } s \in \mathcal{S}. \tag{16}$$

Lemma D.2. *Under Assumption D.1, for any* $Q \in \mathcal{Q}$*, the Bellman optimality operator* B *satisfies the following inequality toward* Q^* *under the* $\|\cdot\|_{d^*}$ *norm:*

$$\|BQ - BQ^*\|_{d^*} \leq \sqrt{\gamma}\|Q - Q^*\|_{d^*}. \tag{17}$$

Proof. By the definition of the weighted norm $\|\cdot\|_{d^*}$ and the Bellman optimality operator B,

$$\|BQ - BQ^*\|_{d^*}^2 = \mathbb{E}_{(s,a)\sim d^*}\Big[\big(\gamma\mathbb{E}_{s'}[\max_{a'} Q(s',a') - \max_{a'} Q^*(s',a')]\big)^2\Big] \tag{18}$$

$$\leq \gamma^2\mathbb{E}_{(s,a)\sim d^*}\Big[\mathbb{E}_{s'}\big[(\max_{a'} Q(s',a') - \max_{a'} Q^*(s',a'))^2\big]\Big] \quad \text{(Jensen's Inequality)} \tag{19}$$

$$= \gamma^2\mathbb{E}_{(s,a)\sim d^*}\mathbb{E}_{s'\sim P(\cdot|s,a)}\Big[\big(\max_{a'} Q(s',a') - Q^*(s',\pi^*(s'))\big)^2\Big]. \tag{20}$$

Since π^* is greedy with respect to Q^*, Assumption D.1 implies

$$Q(s',\pi^*(s')) \leq \max_{a'} Q(s',a') \leq \max_{a'} Q^*(s',a') = Q^*(s',\pi^*(s')). \tag{21}$$

Therefore,

$$\|BQ - BQ^*\|_{d^*}^2 \tag{22}$$

$$\leq \gamma^2\mathbb{E}_{(s,a)\sim d^*, s'\sim P(\cdot|s,a)}\Big[\big(Q(s',\pi^*(s')) - Q^*(s',\pi^*(s'))\big)^2\Big] \tag{23}$$

$$= \gamma^2\mathbb{E}_{(s',a')\sim d^{*'}}\Big[\big(Q(s',\pi^*(s')) - Q^*(s',\pi^*(s'))\big)^2\Big]. \tag{24}$$

Here, $d^{*'}$ denotes the one-step state-action distribution induced by the optimal policy π^*. In operator form, $d^{*'} = (P^{\pi^*})^\top d^*$, i.e.,

$$d^{*'}(s',a') = \sum_{s,a} P(s'|s,a)\delta(a' = \pi^*(s'))d^*(s,a). \tag{25}$$

Recall that d^* satisfies the flow constraint

$$d^* = (1-\gamma)d_0 + \gamma(P^{\pi^*})^\top d^*, \tag{26}$$

$$d_0(s,a) = \mu_0(s)\delta(a = \pi^*(s)). \tag{27}$$

Substituting (26) into (25) gives $d^{*'} = \frac{1}{\gamma}(d^* - (1-\gamma)d_0)$. Let $g(s', a') = (Q(s', a') - Q^*(s', a'))^2 \geq 0$. Since $d^{*'}$ is supported on $a' = \pi^*(s')$, it follows that

$$\gamma^2 \mathbb{E}_{d^{*'}}[g] = \gamma \left(\mathbb{E}_{d^*}[g] - (1-\gamma)\mathbb{E}_{d_0}[g] \right) \tag{28}$$

$$\leq \gamma \mathbb{E}_{d^*}[g] = \gamma \|Q - Q^*\|_{d^*}^2. \tag{29}$$

Taking square roots yields (17). □

Proof of Theorem 3.1. This result follows the error propagation analysis for approximate value iteration [3, 45]. We analyze the distance between the current iterate Q_{k+1} and the optimal value function Q^* under the weighted norm $\| \cdot \|_{d^*}$. Using the triangle inequality:

$$\|Q_{k+1} - Q^*\|_{d^*} = \|Q_{k+1} - BQ_k + BQ_k - Q^*\|_{d^*} \tag{30}$$

$$\leq \underbrace{\|Q_{k+1} - BQ_k\|_{d^*}}_{\text{Approximation Error}} + \underbrace{\|BQ_k - Q^*\|_{d^*}}_{\text{Contraction}} \tag{31}$$

The first term is the approximation error introduced by the weighted regression (projection onto the function space $\mathcal{Q}$), which is bounded by ϵ by assumption.

For the second term, we recall that Q^* is the fixed point of B (i.e., $Q^* = BQ^*$). Using the γ-contraction property of B established in lemma D.2:

$$\|BQ_k - Q^*\|_{d^*} = \|BQ_k - BQ^*\|_{d^*} \tag{32}$$

$$\leq \sqrt{\gamma} \|Q_k - Q^*\|_{d^*} \tag{33}$$

Substituting this back into the inequality, we obtain the recursive bound:

$$\|Q_{k+1} - Q^*\|_{d^*} \leq \epsilon + \sqrt{\gamma} \|Q_k - Q^*\|_{d^*} \tag{34}$$

Unrolling this recurrence relation over k steps:

$$\|Q_k - Q^*\|_{d^*} \leq \gamma^{k/2} \|Q_0 - Q^*\|_{d^*} + \epsilon \sum_{j=0}^{k-1} \gamma^{j/2} \tag{35}$$

As $k \to \infty$, since $\gamma < 1$, the first term $\gamma^{k/2} \|Q_0 - Q^*\|_{d^*}$ vanishes. The geometric series sums to $\frac{1}{1-\sqrt{\gamma}}$. Thus:

$$\limsup_{k \to \infty} \|Q_k - Q^*\|_{d^*} \leq \frac{\epsilon}{1 - \sqrt{\gamma}} \tag{36}$$

□

D.2 Proof of Theorem 3.3

Assumption D.3 (Realizability of policy)**.** The policy class Π contains the expert policy, i.e. $\pi^* \in \Pi$.

Proof. Under assumption of D.3 and infinite data size [1], we can rewrite the learning objects for DCDT variants as following :

$$\pi^*_{DT} := \arg\max_{\pi \in \Pi} \mathcal{L}_{DT}(\pi) = \mathbb{E}_{\substack{\tau_t \sim \mathcal{D} \\ (s_t, a_t) \sim \tau_t}} \|\pi(\tau_t) - a_t\|^2 = \pi^U \tag{37}$$

$$\pi^*_{WDT} := \arg\max_{\pi \in \Pi} \mathcal{L}_{WDT}(\pi) = \mathbb{E}_{\substack{\tau_t \sim \mathcal{D} \\ (s_t, a_t) \sim \tau_t}} \left[w^*(s_t, a_t) \|\pi(\tau_t) - a_t\|^2 \right] := \pi^*_{primal} \tag{38}$$

$$\pi^*_{QDT} := \arg\max_{\pi \in \Pi} \mathcal{L}_{DT}(\pi) + \eta \mathbb{E}[Q_\pi(s, a)] \tag{39}$$

$$\pi^*_{QWDT} := \arg\max_{\pi \in \Pi} \mathcal{L}_{WDT}(\pi) + \eta \mathbb{E}[Q_\pi(s, a)] \tag{40}$$

$$\pi^*_{DCDT} := \arg\max_{\pi \in \Pi} \mathcal{L}_{WDT}(\pi) + \eta \mathbb{E}[Q_{w^*}(s, a)] \tag{41}$$

, where π^*_{primal} denotes optimal policy of (8). We first show that $J(\pi^*_{WDT}) \geq J(\pi^*_{DT})$. If we plug π^U into the (8), we get:

$$J_{primal}(\pi^U) = J(\pi^U) - \alpha_1 D_{\mathrm{KL}}(d^{\pi^U} \| d^E) - \alpha_2 D_{\mathrm{KL}}(d^{\pi^U} \| d^U) \tag{42}$$

$$= J(\pi^U) - \alpha_1 D_{\mathrm{KL}}(d^U \| d^E). \tag{43}$$

Since π^*_{primal} is the maximizer of the objective J_{primal}, we have:

$$J(\pi^*_{primal}) \geq J(\pi^U) + \alpha_2[D_{\mathrm{KL}}(d^U \| d^E) - D_{\mathrm{KL}}(d^* \| d^E)] + \alpha_1 D_{\mathrm{KL}}(d^* \| d^U). \tag{44}$$

Recall that $d^* = d^U w^*(s,a)$, we can express d^* as geometric mixture by plugging in the closed-form of w^*,

$$d^*(s,a) \propto d^U(s,a)^{\frac{\alpha_2}{\alpha_1+\alpha_2}} \cdot d^E(s,a)^{\frac{\alpha_1}{\alpha_1+\alpha_2}}. \tag{45}$$

By the non-negativity and convexity of the KL divergence, we can show that $D_{\mathrm{KL}}(d^U \| d^E) \geq D_{\mathrm{KL}}(d^* \| d^E)$ and $J(\pi^*_{WDT}) \geq J(\pi^*_{DT})$ follows.

Then we prove $J(\pi^*_{QDT}) \geq J(\pi^*_{DT})$.

$$\eta \mathbb{E}_{s\sim d^{\pi_\beta}}[V^{\pi^*_{QDT}}(s)] = \eta \mathbb{E}_{s\sim d^{\pi^U}, a\sim\pi^*_{QDT}(\cdot|s)}[Q^{\pi^*_{QDT}}] \tag{46}$$

$$\geq \eta \mathbb{E}_{s\sim d^{\pi^U}, a\sim\pi^*_{QDT}(\cdot|s)}[Q^{\pi^*_{QDT}}] - \mathbb{E}_{s\sim d^{\pi_\beta}}[D_{\mathrm{KL}}(\pi(\cdot|s)\|\pi^U(\cdot|s))] \tag{47}$$

$$\geq \eta \mathbb{E}_{s\sim d^{\pi^U}, a\sim\pi^*_{QDT}(\cdot|s)}[Q^{\pi^U}] - \mathbb{E}_{s\sim d^{\pi^U}}[D_{\mathrm{KL}}(\pi^U(\cdot|s)\|\pi^U(\cdot|s))] \tag{48}$$

$$= \eta \mathbb{E}_{s\sim d^{\pi_\beta}}[V^{\pi^U}(s)] \tag{49}$$

where the first inequality follows the non-negativity of KL divergence, the second inequality follows π^*_{QDT} is the optimal policy of (39). Similarly, we can prove that $J(\pi^*_{QWDT}) \geq J(\pi^*_{WDT})$.

Then we prove $J(\pi^*_{QWDT}) \geq J(\pi^*_{QDT})$.Both methods maximize the standard policy Q-value Q_π, but differ in their regularization. We express the objectives as KL-regularized policy updates:

$$\pi^*_{QDT} = \arg\max_\pi \left(\eta \mathbb{E}_{s\sim d^U, a\sim\pi}[Q_\pi(s,a)] - D_{\mathrm{KL}}(\pi(\cdot|s)\|\pi^*_{DT}(\cdot|s))\right) \tag{50}$$

$$\pi^*_{QWDT} = \arg\max_\pi \left(\eta \mathbb{E}_{s\sim d^U, a\sim\pi}[Q_\pi(s,a)] - D_{\mathrm{KL}}(\pi(\cdot|s)\|\pi^*_{WDT}(\cdot|s))\right) \tag{51}$$

From the previous result, we know that π^*_{WDT} (the optimal weighted BC policy) has a higher value than π^*_{DT} (vanilla BC), i.e., $V^{\pi^*_{WDT}}(s) \geq V^{\pi^*_{DT}}(s)$. Since the regularization π^*_{WDT} is structurally aligned with the optimal policy π^* due to distribution correction, the KL-divergence term acts as a tighter regularizer compared to the uncorrected behavior policy used in π^*_{QWDT}. Following the theory of regularized MDPs [17, 60], reducing the divergence between the prior and the optimal policy lowers the bias of the regularization. Consequently, π^*_{QWDT} can achieve a higher value solution within the feasible trust region.

Finally, we establish the inequality $J(\pi^*_{DCDT}) \geq J(\pi^*_{QWDT})$ The distinction here lies in the Q-function target. π^*_{QWDT} uses the current policy value Q_π (standard policy iteration), whereas π^*_{DCDT} uses the optimal Q-value Q_{w^*} learned via Dual Weighted TD (Section 3.1).

By the definition of the optimal value function, $Q_{w^*}(s,a) = Q_{\pi_{w^*}}(s,a) = Q^*(s,a) \geq Q^\pi(s,a)$ for any sub-optimal policy π[3]. Consequently, the objective for π^*_{DCDT} constitutes a strict upper bound on the objective for π^*_{QWDT}. By directly distilling the global optimal value Q^*-rather than iteratively climbing via local greedy steps—π^*_{DCDT} avoids the local optima inherent in standard policy iteration [35, 53].Thus:

$$J(\pi^*_{DCDT}) \geq J(\pi^*_{QWDT}). \tag{52}$$

Combining these results with the base inequalities, we arrive at the full performance hierarchy shown in theorem 3.3. □

D.3 Proof of Proposition 3.4

Proof. We can obtain the following Lagrangian for the constrained optimization problem(8-4) in

$$\max_{d\geq 0}\min_\nu \mathbb{E}_{(s,a)\sim d}[R(s,a)] - \alpha_1 D_{\mathrm{KL}}(d\|d^E) - \alpha_2 D_{\mathrm{KL}}(d\|d^U) + \sum_s \nu(s)\Big((1-\gamma)p_0(s)\gamma(\mathcal{T}_* d)(s) - (\mathcal{B}_* d)(s)\Big), \tag{53}$$

where $\nu(s)$ are the Lagrange multipliers.

We then introduce the following derivations for the optimization (53) to obtain tractable optimization in the offline setting:

$$\begin{aligned}
&\mathbb{E}_{(s,a)\sim d}[R(s,a)] - \alpha_1 D_{\mathrm{KL}}(d\|d^E) - \alpha_2 D_{\mathrm{KL}}(d\|d^U) \\
&\quad + \sum_s \nu(s)\Big((1-\gamma)p_0(s) + \gamma(\mathcal{T}_* d)(s) - (\mathcal{B}_* d)(s)\Big) \qquad (54)\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d}\Big[R(s,a) + \gamma(\mathcal{T}\nu)(s,a) - (\mathcal{B}\nu)(s,a) \\
&\qquad - \alpha_1 \log \frac{d(s,a)}{d^E(s,a)} - \alpha_2 \log \frac{d(s,a)}{d^U(s,a)}\Big] \qquad (55)\\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d}\Big[\underbrace{\alpha_1 \log \frac{d^E(s,a)}{d^U(s,a)} + R(s,a)}_{r^{\alpha_1}(s,a)} + \gamma(\mathcal{T}\nu)(s,a) \\
&\qquad - (\mathcal{B}\nu)(s,a) - (\alpha_1+\alpha_2)\log \underbrace{\frac{d(s,a)}{d^U(s,a)}}_{w(s,a)}\Big]. \qquad (56)
\end{aligned}$$

We introduce another log ratio $\log \frac{d^E(s,a)}{d^U(s,a)}$ in Equation (56) to avoid using $\log \frac{d(s,a)}{d^E(s,a)}$, which requires on-policy samples to estimate. Unlike $\log \frac{d(s,a)}{d^E(s,a)}$, we can estimate the distribution-shaped reward $r^{\alpha_1}(s,a)$, in the offline setting using d^E and d^U using a discriminator[29, 43]. Details in Appendix E.2.

We then change the distribution used in the expectation of Equation (56) from d to d^U by following the standard trick of importance sampling as follows:

$$\begin{aligned}
&(1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d}\Big[\underbrace{r^{\alpha_1} + \gamma(\mathcal{T}\nu)(s,a) - (\mathcal{B}\nu)(s,a)}_{:=A_\nu^{\alpha_1}(s,a)\text{(advantage using }\nu)} - (\alpha_1+\alpha_2)\log w(s,a)\Big] \\
&= (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d^U}\Big[w(s,a)(A_\nu^{\alpha_1}(s,a) - (\alpha_1+\alpha_2)\log w(s,a))\Big] \\
&=: L_\alpha(w,\nu). \qquad (57)
\end{aligned}$$

□

D.4 Proof of Proposition 3.5

Proof. We first show that our origin problem (equation 8,equation 4) is an instance of convex programming due to the convexity of KL.The objective function $J(d)$ is concave in d because the negative KL-divergence is concave, and a non-negative linear combination of concave functions preserves concavity . Additionally, the Bellman flow constraints are affine in d, and the non-negativity constraints are linear. In addition, by using the strong duality and the change-of-variable from d to w, we can rearrange the original maximin optimization to the minimax optimization.

Let us denote the Lagrangian of the constraint optimization (equation 8 with equation 4) as $L_\alpha(d,\nu,\mu) := \mathbb{E}_d[R(s,a)] - \alpha_1 D_{KL}(d\|d^E) - \alpha_2 D_{KL}(d\|d^U) + \sum_s \nu(s)\Big((1-\gamma)p_0(s)\gamma(\mathcal{T}_* d)(s) - (\mathcal{B}_* d)(s)\Big) + \sum_{s,a}\mu(s,a)d(s,a)$ with Lagrange multipliers $\nu(s)\forall s$ and $\mu(s,a)\forall s,a$. With the Lagrangian $L_\alpha(d,\nu,\mu)$, the original problem (8) can be represented by

$$\max_{d\geq 0}\min_{\nu} L_\alpha(d,\nu,0) = \max_{d}\min_{\nu,\mu\geq 0} L_\alpha(d,\nu,\mu). \qquad (58)$$

For an MDP where every $s\in\mathcal{S}$ is reachable, there always exists d such that $d(s,a) > 0$. From Slater's condition, the strong duality holds, i.e., we can change the order of optimizations:

$$\begin{aligned}
\max_{d}\min_{\nu,\mu\geq 0} L_\alpha(d,\nu,\mu) &= \min_{\nu,\mu\geq 0}\max_{d} L_\alpha(d,\nu,\mu) \\
&= \min_{\nu}\max_{d\geq 0} L_\alpha(d,\nu,0). \qquad (59)
\end{aligned}$$

Here, the last equality holds since $\max_{d\geq 0} L_\alpha(d,\nu,0) = \max_d \min_{\mu\geq 0} L_\alpha(d,\nu,\mu)$ for fixed ν due to strong duality. Finally, by applying the change of variable $w = d/d^U$, we have

$$\max_{w\geq 0}\min_{\nu} L_\alpha(w,\nu) = \min_{\nu}\max_{w\geq 0} L_\alpha(w,\nu). \tag{60}$$

Finally, the solution of the inner maximization $\max_{w\geq 0} L_\alpha(w,\nu)$ can be derived as follows: For a fixed ν, let the maximization be the primal problem. Then its corresponding dual problem is

$$\max_{w}\min_{\nu\geq 0} L_\alpha(w,\nu) + \sum_{s,a}\mu(s,a)w(s,a). \tag{61}$$

Since the strong duality holds, satisfying KKT condition is both necessary and sufficient conditions for the solutions w^* and μ^* of primal and dual problems (we will use w^* and μ^* instead of w^*_ν and μ^*_ν for notational brevity). From Stationarity and $d^U \geq 0$, we have

$$w^*(s,a) = \exp\left(\frac{A^{\alpha_1}_\nu(s,a) + \mu^*(s,a)}{\alpha_1+\alpha_2} - 1\right) \quad \forall s,a. \tag{62}$$

Due to Complementary Slackness $w^*(s,a)\mu^*(s,a) = 0$. Since the exponential function is strictly positive, $w^*(s,a) > 0$ implies $\mu^*(s,a) = 0$. Thus, we obtain the closed-form solution:

$$w^*_\nu(s,a) = \exp\left(\frac{A^{\alpha_1}_\nu(s,a)}{\alpha_1+\alpha_2} - 1\right) \quad \forall s,a. \tag{63}$$

□

D.5 Proof of Lemma C.6

Lemma D.4. *Suppose Assumption C.2,C.3 hold. Then with at least probability $1-\delta$, for all $\nu \in \mathcal{V}$ and $w \in \mathcal{W}$ we have:*

$$|\hat{L}_\alpha(\nu,w) - L(\nu,w)| \leq \mathcal{E}_{n,n_0,\alpha}(B_{w,\alpha}, B_{\nu,\alpha}, B_{e,\alpha}) := \epsilon_{stat}, \tag{64}$$

Proof of Lemma D.4. Let $l_i^{\nu,w} = -(\alpha_1+\alpha_2)\log w(s_i,a_i) + w(s_i,a_i)A_\nu(s_i,a_i,r_i,s_i')$. From Assumption C.3, we know

$$|A_\nu(s,a,r,s')| = |r(s,a) + \gamma\nu(s') - \nu(s)| \leq (1+\gamma)B_{\nu,\alpha} + 1 = B_{e,\alpha}. \tag{65}$$

Therefore, by Assumption C.2, we have:

$$|l_i^{\nu,w}| \leq (\alpha_1+\alpha_2)B_{f,\alpha} + B_{w,\alpha}B_{\nu,\alpha}, \tag{66}$$

where $B_{f,\alpha} = \max\{|\log B^{min}_{w,\alpha}|, |\log B^{max}_{w,\alpha}|\}$. Notice that $l_i^{\nu,w}$ is independent from each other, thus we can apply Hoeffding's inequality and for any $t > 0$,

$$\Pr[|\frac{1}{n}\sum_{i=1}^n l_i^{\nu,w} - \mathbb{E}[l_i^{\nu,w}]| \leq t] \geq 1 - 2\exp\left(\frac{-nt^2}{2((\alpha_1+\alpha_2)B_{f,\alpha} + B_{w,\alpha}B_{\nu,\alpha})^2}\right). \tag{67}$$

Let $t = ((\alpha_1+\alpha_2)B_{f,\alpha} + B_{w,\alpha}B_{\nu,\alpha})\sqrt{\frac{2\log\frac{4|\mathcal{V}||\mathcal{W}|}{\delta}}{n}}$, we have with at least probability $1 - \frac{\delta}{2|\mathcal{V}||\mathcal{W}|}$,

$$|\frac{1}{n}\sum_{i=1}^n l_i^{\nu,w} - \mathbb{E}[l_i^{\nu,w}]| \leq ((\alpha_1+\alpha_2)B_{f,\alpha} + B_{w,\alpha}B_{\nu,\alpha})\sqrt{\frac{2\log\frac{4|\mathcal{V}||\mathcal{W}|}{\delta}}{n}}. \tag{68}$$

Therefore by union bound, with at least probability $1-\frac{\delta}{2}$, we have for all $\nu \in \mathcal{V}$ and $w \in \mathcal{W}$,

$$|\frac{1}{n}\sum_{i=1}^n l_i^{\nu,w} - \mathbb{E}[l_i^{\nu,w}]| \leq ((\alpha_1+\alpha_2)B_{f,\alpha} + B_{w,\alpha}B_{\nu,\alpha})\sqrt{\frac{2\log\frac{4|\mathcal{V}||\mathcal{W}|}{\delta}}{n}}. \tag{69}$$

Similarly, we have with at least probability $1-\frac{\delta}{2}$, for all $\nu \in \mathcal{V}$,

$$|\frac{1}{n_0}\sum_{j=1}^{n_0} v(s_{0,j}) - \mathbb{E}_{s\sim\mu_0}[v(s)]| \leq B_{\nu,\alpha}\sqrt{\frac{2\log\frac{4|\mathcal{V}|}{\delta}}{n_0}}. \tag{70}$$

Therefore, with at least probability $1-\delta$ we have

$$|\hat{L}_\alpha(\nu,w)-L_\alpha(\nu,w)| \leq ((\alpha_1+\alpha_2)B_{f,\alpha}+B_{w,\alpha}B_{\nu,\alpha})\sqrt{\frac{2\log\frac{4|\mathcal{V}||\mathcal{W}|}{\delta}}{n}}+(1-\gamma)B_{\nu,\alpha}\sqrt{\frac{2\log\frac{4|\mathcal{V}|}{\delta}}{n_0}}. \tag{71}$$

□

Proof of Lemma C.6. First we can show $L_\alpha(\nu^*_\alpha, w)$ is $\frac{\alpha_1+\alpha_2}{B^{max}_{w,\alpha}}$-strongly-concave with respect to w and $\|\cdot\|_{2,d^D}$. On the other hand, since $(\nu^*_\alpha, w^*_\alpha)$ is a saddle point of $L_\alpha(\nu, w)$ over $\mathbb{R}^{|\mathcal{S}|}\times\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, we have $w^*_\alpha = \arg\max_{w\geq B^{min}_{w,\alpha}} L_\alpha(\nu^*_\alpha, w)$. Then we have:

$$\|\hat{w}-w^*_\alpha\|_{2,d^D} \leq \sqrt{\frac{2B^{max}_{w,\alpha}(L_\alpha(\nu^*_\alpha,w^*_\alpha)-L_\alpha(\nu^*_\alpha,\hat{w}))}{\alpha_1+\alpha_2}}. \tag{72}$$

Substituting Lemma D.4 into the above equation we can obtain (14). For (15), it can be observed that

$$\|\hat{d}-d^*_\alpha\|_1 = \|\hat{w}-w^*_\alpha\|_{1,d^D} \leq \|\hat{w}-w^*_\alpha\|_{2,d^D} \leq \sqrt{\frac{4B^{max}_{w,\alpha}\epsilon_{\text{stat}}}{\alpha_1+\alpha_2}}. \tag{73}$$

□

E Experimental Details

E.1 Environments and Datasets

Environments: We focus on 10 locomotion and manipulation environments from the MuJoCo physics engine [57]. These environments include Hopper, Walker2d, HalfCheetah, Ant, Pen, Door, Hammer, and Relocate. The MuJoCo environments used in this work are licensed under CC BY 4.0 and the datasets used from D4RL are also licensed under Apache 2.0.

Suboptimal Datasets: We utilize offline datasets consisting of environment interactions from the D4RL framework [10]. Specifically, we construct suboptimal datasets following the composition approach introduced in SMODICE [43]. The suboptimal datasets, denoted as 'random+expert', 'random+few-expert', 'medium+expert', and 'medium+few-expert' combine expert trajectories with low-quality trajectories obtained from the "random-v2" and "medium-v2" datasets, respectively. For locomotion tasks, the 'x+expert' dataset (where x is 'random' or 'medium') contains a mixture of some number of expert trajectories (≤ 200) and ≈ 1 million transitions from the "x" dataset. The 'x+few-expert' dataset is similar to 'x+expert,' but with only 30 expert trajectories included. For manipulation environments we consider only 30 expert trajectories mixed with the complete 'x' dataset of transitions obtained from D4RL.

High-quality Reference Subset. In our main experiments, the reference subset $\mathcal{D}^E$ consists of 200 trajectories for the "+expert" mixtures and 30 trajectories for the "+few-expert" mixtures on locomotion tasks, and 30 trajectories on manipulation tasks. For the ReCOIL baseline, we follow its original protocol and use a single trajectory from the "expert-v2" dataset per environment; this choice is specific to ReCOIL and does not apply to DCDT or other baselines.

Baselines: To benchmark and analyze the performance of our proposed methods of offline reinforcement learning with suboptimal data, we consider seven representative baselines in this work: RCE [9], ORIL [66], SMODICE [43], IQLearn [16], ReCOIL [52], QT [25], and VDT [65].

The open-source implementations of the baselines RCE, ORIL, SMODICE, ReCOIL, QT and VDT provided by the authors are employed in our experiments. We use the hyperparameters provided by the authors, which are consistent with those used in the original paper, for all the MuJoCo locomotion and manipulation environments.

E.2 Discriminator Training

In continuous environment, we can train a discriminator $c : \mathcal{S}\times\mathcal{A}\to(0,1)$ via the standard binary cross-entropy objective:

$$\min_c \mathbb{E}_{(s,a)\sim d^E}[\log c(s,a)] + \mathbb{E}_{(s,a)\sim d^U}[\log 1 - c(s,a)]. \tag{74}$$

The optimal discriminator is $c^*(s) = \frac{d^U(s,a)}{d^E(s,a)+d^U(s,a)}$[18]. Combined with (56), we can use $r^{\alpha_1}(s,a) = -\alpha_1 \log(\frac{1}{c^*(s,a)} - 1) + R(s,a)$.

In practice, we parameterize c with a neural network c_κ trained via (74). Consequently, the augmented advantage is computed as:

$$A^{\alpha_1}_{\nu_\psi}(\mathbf{s}_t, \mathbf{a}_t) = r^{\alpha_1}_{c_\kappa}(\mathbf{s}_t, \mathbf{a}_t) + \gamma \nu_\psi(\mathbf{s}_{t+1}) - \nu_\psi(\mathbf{s}_t). \tag{75}$$

Here, ν_ψ is the function approximation of Lagrangian variable defined in (53).

E.3 Finite MDP Experiment

Algorithm 1 3.2 for Tabular MDPs

Require: MLE MDP $\hat{M} = \langle S, A, \mathcal{T}, r, \gamma, p_0 \rangle$, data-collection policy $\pi_U = \beta \pi_E + (1-\beta)\pi_D$, regularization hyperparameter $\alpha > 0$.

$d^{\pi_U} \leftarrow$ COMPUTESTATIONARYDISTRIBUTION$(\hat{M}, \pi_U)$
$d^{\pi_E} \leftarrow$ COMPUTESTATIONARYDISTRIBUTION$(\hat{M}, \pi_E)$
$D_U \leftarrow \mathrm{diag}(d^{\pi_U})$
$\nu \leftarrow$ (random initialization)
while ν is not converged **do**
  $e \leftarrow r + \log d^{\pi_E}/d^{\pi_U} + \gamma \mathcal{T} v - \mathcal{B} v$
  $w \leftarrow \exp(\frac{e}{1+\alpha_2})$
  $J \leftarrow \frac{1}{1+\alpha_2} \mathrm{diag}(w)(\gamma \mathcal{T} - \mathcal{B})$
  $g \leftarrow (1-\gamma)p_0 + \frac{1}{D_U^\intercal w}(D_U^\intercal \mathrm{diag}(w))(\gamma \mathcal{T} - \mathcal{B})$
  $\nu \leftarrow \nu - \eta g$ (where η is a step-size)
end while

$$\pi^*(a|s) \leftarrow \frac{w(s,a) d^{\pi_U}(s,a)}{\sum_{a'} w(s,a') d^{\pi_U}(s,a')} \quad \forall s, a$$

Ensure: π^*, w

Figure 2 outlines how our approach works in the Four Rooms domain [54] where the agent aims to navigate to a goal location in a maze composed of four rooms. We collect a static dataset D^U consisting of 50 episodes with maximum time step 50 using the data-collection policy $\pi^U = 0.3\pi^*_{\text{true}} + 0.7\pi_{\text{rand}}$, where π^*_{true} is the optimal policy of the underlying true MDP and π_{rand} is the random policy sampled from the Dirichlet distribution, i.e., $\pi_{\text{rand}}(s) \sim \mathrm{Dir}(1,1,1,1)$, $\forall s$.

For Finite MDPs, we can reformulate the expected objective of (8) into vector-matrix form. To simplify the derivation and for illustrative purposes, we set $\alpha_1 = 1$.

$$\begin{aligned} \min_\nu \max_{w \geq 0} L_{\alpha_2}(w, \nu) = (1-\gamma) p_0^\intercal \nu - (1+\alpha_2)(\log w)^\intercal D w \\ + w^\intercal D(R + \gamma \mathcal{T} \nu - \mathcal{B} \nu), \end{aligned} \tag{76}$$

where $\nu \in \mathbb{R}^{|S|}$ is a $|S|$-dimensional vector, $w \in \mathbb{R}^{|S||A|}$ is a $|S||A|$-dimensional vector, $R \in \mathbb{R}^{|S||A|}$ is a $|S||A|$-dimensional reward vector, and $D = \mathrm{diag}(d^U)$ is the diagonal matrix induced by offline dataset D^U. Note that $R = \log \frac{d^E(s,a)}{d^U(s,a)} + r_{env}$ and we can estimate log ratio by counting. Let $\mathcal{T} \in \mathbb{R}^{|S||A| \times |S|}$ and $\mathcal{B} \in \mathbb{R}^{|S||A| \times |S|}$ be the matrices satisfying the following equations:

$$\mathcal{T}\nu \in \mathbb{R}^{|S||A|} \quad \text{s.t.} \quad (\mathcal{T}\nu)((s,a)) = \sum_{s'} T(s'|s,a)\nu(s')$$

$$\mathcal{B}\nu \in \mathbb{R}^{|S||A|} \quad \text{s.t.} \quad (\mathcal{B}\nu)((s,a)) = \nu(s).$$

So $A_\nu = R + \gamma \mathcal{T}\nu - \mathcal{B}\nu$. Then, by substituting the closed-form equation of the inner maximization $w^*_\nu = \exp(\frac{A_\nu}{1+\alpha_2})$ into $L_\alpha(w^*_\nu, \nu)$, we can get:

$$\min_\nu L_\alpha(w^*_\nu, \nu) = L_\alpha(\nu) := (1-\gamma) p_0^\intercal \nu - (1+\alpha_2)(\log w^*_\nu)^\intercal D w^*_\nu + w^{*\intercal}_\nu D(R + \gamma \mathcal{T}\nu - \mathcal{B}\nu). \tag{77}$$

Since $L_\alpha(\nu)$ is convex in ν, we perform Nesterov's Accelerated Gradient Descent to compute the optimal ν^* efficiently. For almost every ν, we can compute the first and second derivatives as follows:

$$
\begin{aligned}
A_\nu &:= R + \gamma \mathcal{T}\nu - \mathcal{B}\nu && \text{(advantage via } \nu\text{)} \\
w_\nu^* &:= \exp\Big(\tfrac{A_\nu}{1+\alpha_2}\Big) && \text{(closed form)} \\
J &:= \tfrac{1}{1+\alpha_2}\,\mathrm{diag}(w_\nu^*)(\gamma\mathcal{T} - \mathcal{B}) && \text{(Jacobian)} \\
g &:= \tfrac{\partial L(\nu)}{\partial \nu} && \text{(gradient)} \\
&\quad = (1-\gamma)\,p_0 \\
&\quad + \tfrac{1}{D^\intercal w_\nu^*}\big(D^\intercal\,\mathrm{diag}(w_\nu^*)\big)(\gamma\mathcal{T} - \mathcal{B})
\end{aligned}
$$

We iteratively update ν in the direction of $-g$ until convergence. Finally, $w_{\nu^*}^*$ and the corresponding optimal policy $\pi^*(a|s) \propto w_{\nu^*}^* d^U(s,a)$ are computed. The pseudo-code of these procedures is presented in Algorithm 1.

F DCDT Training Algorithm

We provide the complete training procedure of DCDT in Algorithm 2. The algorithm summarizes the main steps of the proposed framework, discriminator training, dual density-ratio estimation, reweighted TD learning, weighted Decision Transformer policy extraction, and target-network updates.

Algorithm 2 DCDT: Distribution Correction Decision Transformer

Input: Sequence horizon K, offline datasets $\mathcal{D}^U$, high-quality reference subset $\mathcal{D}^E$, coefficient ρ, discount γ.
Initialize policy network π_θ, dual network ν_ψ, critic networks Q_{ϕ_1}, Q_{ϕ_2}, discriminator c_κ, and target networks $\pi_{\theta'}, Q_{\phi_1'}, Q_{\phi_2'}$.
`// Train the DCDT`
for iteration $= 1$ **to** T **do**
  Sample mini-batch $\mathcal{B}^U \sim \mathcal{D}^U$, $\mathcal{B}^E \sim \mathcal{D}^E$.
  `// (i) Discriminator update`
  Update c_κ by binary cross-entropy on $\mathcal{B}^E$ vs. $\mathcal{B}^U$.
  `// (ii) Dual update`
  Update dual network ν_ψ by Equation (11).
  Compute augmented advantage $A_{\nu_\psi}^{\alpha_1}$ via Equation (56).
  `// (iii) Compute density-ratio weights`
  $w_{\nu_\psi}(s,a) = \exp\big(A_{\nu_\psi}^{\alpha_1}(s,a)/(\alpha_1+\alpha_2) - 1\big)$.
  `// (iv) Weighted Q update`
  Update Q_{ϕ_1}, Q_{ϕ_2} by Equation (7), using weights w_{ν_ψ}.
  `// (v) Weighted policy update`
  Update π_θ by minimizing
  $\mathbb{E}_{\mathcal{B}}[w_{\nu_\psi}\|\pi_\theta(\tau_t) - a_t\| - \lambda \min_i Q_{\phi_i'}(s_t, \pi_\theta(\tau_t))]$
  `// (vi) Target networks`
  $\theta' = \rho\theta' + (1-\rho)\theta$, $\phi_i' = \rho\phi_i' + (1-\rho)\phi_i$, for $i = \{1,2\}$.
end for.

G Implementation Details

Conditional Transformer Policy. We build our policy as a Transformer-based model, which is based on minGPT open-source code. The detailed model parameters are in Table 4.

Q networks. We build two Q networks with the same MLP setting as our diffusion policy, which has 3-layer MLPs with Mish activations and 256 hidden units for all networks.

Discriminator Network. We build our discriminator network using a simple 2-layer ReLU network with hidden size 256. For stable discriminator learning, we use gradient penalty regularization on the $r(s, a)$ function, which was proposed in [19] to enforce 1-Lipschitz constraint.

We use the Adam [30] optimizer for the training of Conditional Transformer Policy, Q networks and Discriminator Network.

Table 4: Hyperparameters of DCDT in our experiments.

Parameter	Value
Number of layers	4
Number of attention heads	4
Embedding dimension	256
Nonlinearity function	ReLU
Batch size	256
Context length K	20
Dropout	0.1
Learning rate	3.0e-4
Gradient penalty	10

H Hyperparameters Configuration

For DCDT, we consider three hyper-parameters in total: Q-value regularization weight η and offline data regularization strength α_1, α_2. For the Q-value regularization weight η, it can be determined via grid search, and we directly use the value reported in QT [25][3] in our experiment without a further fine search. For the offline data regularization strength, we fix the $\alpha_1 = 1$ and consider α_2 values in the grid $\{0.05, 0.5, 1.0, 2.0\}$. Based on these considerations, we provide our hyper-parameter setting in Table 5.

Table 5: Hyperparameter settings of all selected tasks.

Tasks	η	α_2	**Tasks**	η	α_2
halfcheetah-random+expert	5.0	0.05	ant-random+expert	0.05	0.05
halfcheetah-random+few-expert	5.0	0.05	ant-random+few-expert	0.05	0.05
halfcheetah-medium+expert	5.0	0.05	ant-medium+expert	0.05	0.05
halfcheetah-medium+few-expert	2.0	0.5	ant-medium+few-expert	0.05	0.05
hopper-random+expert	2.0	0.05	pen-cloned+expert	0.1	0.05
hopper-random+few-expert	0.01	0.5	pen-human+expert	0.1	0.05
hopper-medium+expert	1.0	0.05	door-cloned+expert	0.001	0.05
hopper-medium+few-expert	1.0	0.05	door-human+expert	0.005	0.05
walker2d-random+expert	2.0	0.05	hammer-cloned+expert	0.01	0.05
walker2d-random+few-expert	0.2	0.5	hammer-human+expert	0.1	0.05
walker2d-medium+expert	0.5	0.5	relocate-cloned+expert	0.001	0.05
walker2d-medium+few-expert	2.0	0.05	relocate-human+expert	0.001	0.05

I Runtime and Memory Comparison

To evaluate the computational overhead introduced by DCDT, Table 6 reports per-task wall-clock training time and peak GPU memory across four Gym tasks, evaluated on a single GPU with identical batch size and sequence length to ensure a controlled comparison. This experiment is designed to

[3]https://github.com/charleshsc/QT

assess the computational overhead introduced by DCDT relative to existing methods. DCDT incurs approximately $\sim 6\%$ additional wall-clock time and $\sim 1.4\%$ additional peak memory compared to QT. This overhead is primarily attributable to the discriminator and the dual network, both implemented as 2-layer MLPs and modest in size relative to the Transformer policy and Q-network updates shared by DCDT and QT.

Table 6: Per-task wall-clock training time (hours) and peak GPU memory (MB) on locomotion tasks.

	Wall-clock (hr)				Peak GPU memory (MB)			
Task	DT	QT	VDT	DCDT	DT	QT	VDT	DCDT
halfcheetah-m	6.3	11.8	11.9	12.3	1175	2389	2391	2424
hopper-m	3.5	9.2	9.4	10.1	1175	2313	2315	2348
halfcheetah-m-r	7.2	12.2	12.2	12.4	1176	2267	2268	2296
hopper-m-r	3.5	8.5	8.5	9.1	1175	2254	2256	2283
Average	**5.1**	**10.4**	**10.5**	**11.0**	**1175**	**2306**	**2308**	**2338**

J Gaussian (MLP) Policy Variant

To verify that the dual-correction mechanism is not specific to the Transformer backbone, we replace the conditional Transformer policy with a conventional 3-layer MLP Gaussian policy (denoted **DCDT-MLP**) while keeping the rest of the pipeline—density-ratio estimation, weighted TD learning, and weighted behavioral cloning—unchanged. Table 7 compares DCDT-MLP to the Transformer-based DCDT per task across three mixed D4RL suites. DCDT-MLP is already competitive on Gym

Table 7: Per-task normalized scores of DCDT with a Gaussian (MLP) policy versus the Transformer policy on mixed D4RL datasets. The dual-correction mechanism remains effective under an MLP backbone; the Transformer's gain is largest on long-horizon Adroit manipulation and FrankaKitchen.

Setting	Environment	Gaussian (MLP)	Transformer (DCDT)
medium+expert (Gym)	hopper	112.1	**113.0**
	halfcheetah	90.5	**92.8**
	walker2d	**110.7**	109.8
	ant	133.9	**134.7**
	Average	111.8	**112.6**
human+expert (Adroit)	pen	98.5	**160.2**
	door	103.4	**105.1**
	hammer	125.2	**126.9**
	relocate	97.9	**106.8**
	Average	106.3	**124.8**
partial+expert (Kitchen)	kitchen	45.0	**52.5**

locomotion tasks, confirming that density-ratio reweighting is architecture-agnostic. The advantage of the Transformer backbone is most pronounced on Adroit, such as pen with a gain of $+61.7$, and FrankaKitchen, with a gain of $+7.5$, where long-horizon dependencies and multi-stage subgoals favor the inductive bias of sequence modeling. We therefore use the Transformer as the default backbone of DCDT.

K Per-task Results on Original D4RL

Table 9 provides the per-task results of DCDT on the original D4RL datasets. Each score is the normalized return averaged over 5 evaluation seeds, where the top-200 trajectories ranked by cumulative return are used as the reference subset $\mathcal{D}^E$. These results show that DCDT remains effective even when the reference subset is not at the expert level, suggesting that the proposed correction requires only a relatively high-return partition rather than access to true expert demonstrations.

Table 8: Average normalized return of trajectories in each data split for the 6 D4RL tasks used in Table 9. The top-200 split (our reference subset $\mathcal{D}^E$) is the 200 highest-return trajectories of the source dataset. The Expert column reports `expert-v2` dataset of each task.

Task	Original	Top-200 of Original	Expert
halfcheetah-medium	40.7	42.7	88.1
hopper-medium	44.3	65.5	108.5
walker2d-medium	62.0	83.2	107.1
halfcheetah-medium-replay	27.2	27.5	88.1
hopper-medium-replay	15.0	52.8	108.5
walker2d-medium-replay	14.8	50.4	107.1

To quantify the quality gap between the reference subset $\mathcal{D}^E$ and true expert data, Table 8 reports the average normalized return of trajectories in each data split used for the standard-D4RL experiments in Table 9. "Original" denotes the full D4RL dataset. "Top-200 of Original" denotes the subset of the highest-return trajectories selected from the original dataset, which is used as the reference subset $\mathcal{D}^E$. "Expert" denotes the corresponding D4RL `expert-v2` dataset for comparison. The top-200 split has a averaged return far below that of the expert dataset, confirming that the reference subset used by DCDT in the standard-D4RL setting contains only *moderately good* trajectories compared to the rest of original dataset rather than true expert demonstrations.

Table 9: Per-task normalized scores on the original D4RL datasets. Scores are reported against a broad set of offline-RL and sequence-modelling baselines; DCDT obtains the best or tied-best score on most individual tasks across Gym locomotion tasks.

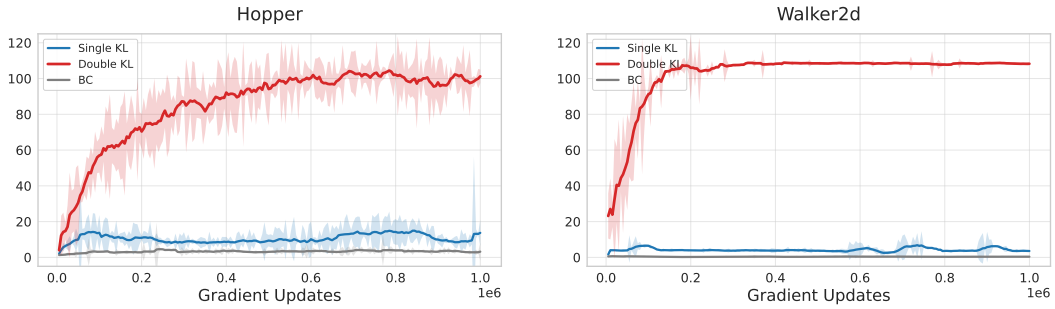
Domain	Task	CQL	IQL	BCQ	TD3+BC	MoRel	BC	DD	DT	StAR	GDT	CGDT	QT	**DCDT**
Gym	halfcheetah-m	49.2	47.4	41.5	48.4	42.1	42.6	49.1	42.6	42.9	42.9	43.0	51.0	**51.9**
Gym	hopper-m	69.4	66.3	65.1	59.3	95.4	52.9	79.3	67.6	59.5	77.1	96.9	96.9	**101.3**
Gym	walker2d-m	83.0	78.3	52.0	83.7	77.8	75.3	82.5	74.0	73.8	76.5	79.1	87.2	**92.9**
Gym	halfcheetah-m-r	45.5	44.2	34.8	44.6	40.2	36.6	39.3	36.6	36.8	40.5	40.4	**48.8**	46.3
Gym	hopper-m-r	95.0	94.7	31.1	60.9	93.6	18.1	100.0	82.7	29.2	85.3	93.4	102.1	**102.4**
Gym	walker2d-m-r	77.2	73.9	13.7	81.8	49.8	32.3	75.0	79.4	39.8	77.5	78.1	97.8	**98.7**
Gym	**Average**	69.9	67.5	39.7	63.1	66.5	43.0	70.9	63.8	47.0	66.6	71.8	80.6	**82.3**

L Effect of Dual objective Composition

The dual objective (8) has two regularizers: α_1 aligns the learned occupancy with the reference d^E, and α_2 keeps it within the support of d^U. Setting $\alpha_1{=}0$ drops the d^E-anchored term and reduces the closed-form ratio to $w^* \propto \exp(A_\nu/(1+\alpha_2))$—a purely reward-driven correction. To evaluate the role of expert matching, we ablate this term by setting $\alpha_1 = 0$. We keep α_2 active in this ablation, since controlling divergence from the data distribution is essential in offline RL for reliable value estimation. Figure 4 shows the resulting dual-objective learning curves. In the imbalanced dataset regime, removing α_1 degrades performance, as the policy tends to overfit suboptimal modes in d^U. On six standard D4RL Gym tasks (Table 10), the imbalance between high- and low-quality trajectories is much milder, DCDT($\alpha_1{=}0$) outperforms VDT and QT . Furthermore, after tuning α_2, our method, shown in the fourth column of Table 10, exceeds prior state-of-the-art results by a larger margin. These results suggest that the importance of the reference-subset penalty increases as the level of dataset imbalance grows.

M Sensitivity to α_2 and reference-subset size

This appendix gives the full evidence behind the one-sentence sensitivity claim in additional experiments: per-task plots for the three Gym tasks (Figure 5), and the full numerical table (Table 11). On D_1, the variation across a $40\times$ range of α_2 is bounded within 2.5% on `hopper`, 3.5% on `walker2d`, and 10.8% on `halfcheetah`; no tested value yields a catastrophic failure, and $\alpha_2{=}0.05$ is a near-universal best default.



(a) Hopper-Random+Expert

(b) Walker2d-Random+Expert

Figure 4: Learning curves of dual objectives: the double-KL form outperforms single-KL (α_1=0) and BC under imbalanced data.

Table 10: α_1 ablation on standard D4RL Gym datasets. Setting α_1=0 removes the reference-subset KL term; with α_2 and η retuned for the single-KL setting, DCDT(α_1=0) matches or exceeds the α_1=1 variant on every task.

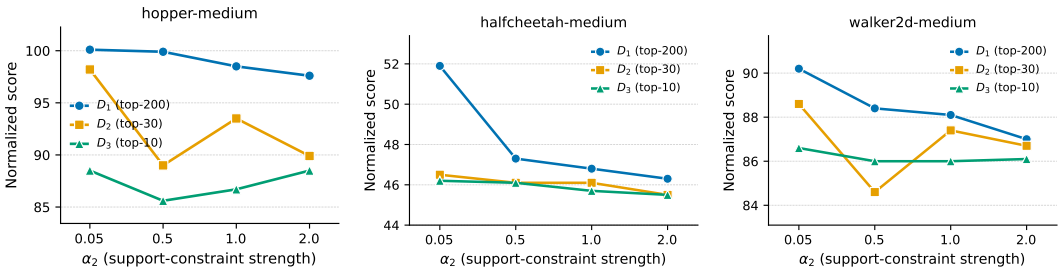
Task	VDT	QT	DCDT$^{\alpha_2=0.05\ \text{default}}_{\alpha_1=0}$	DCDT$^{\alpha_2\ \text{tuned}}_{\alpha_1=0}$	DCDT$_{\alpha_1=1}$
hopper-medium-expert	111.5	113.0	113.1	113.3	**113.9**
walker2d-medium-expert	110.4	112.0	112.9	114.5	**114.8**
hopper-medium	98.3	96.9	99.9	100.6	**101.3**
walker2d-medium	81.6	87.2	90.1	91.8	**92.9**
hopper-medium-replay	96.0	102.1	102.9	102.9	**103.4**
walker2d-medium-replay	82.3	97.8	98.2	102.0	**102.2**
Average	96.7	101.5	102.9	104.2	**104.5**

Table 11: Per-task normalized scores across four values of α_2 for all three reference-subset sizes.

Subset	Task	α_2=0.05	α_2=0.5	α_2=1.0	α_2=2.0
D_1 (top-200)	hopper-medium	**101.3**	99.9	98.5	97.6
	halfcheetah-medium	**51.9**	47.3	46.8	46.3
	walker2d-medium	**92.9**	88.4	88.1	87.0
D_2 (top-30)	hopper-medium	**98.2**	89.0	93.5	89.9
	halfcheetah-medium	**46.5**	46.1	46.1	45.5
	walker2d-medium	**88.6**	84.6	87.4	86.7
D_3 (top-10)	hopper-medium	**88.5**	85.6	86.7	**88.5**
	halfcheetah-medium	**46.2**	46.1	45.7	45.5
	walker2d-medium	**86.6**	86.0	86.0	86.1

N Maze2D Stitching — Full Numerical Results

Table 12 provides the numerical results underlying Figure 3, including per-maze and per-method normalized scores with standard deviations on the Maze2D suite under both sparse and dense rewards. The mixed dataset consists of 20 expert trajectories and 100 random trajectories. The highest score in each row is bolded. These results show that DCDT consistently improves stitching performance under imbalanced data, indicating its ability to recover feasible high-return paths from mixed-quality trajectories.



(a) hopper-medium (b) halfcheetah-medium (c) walker2d-medium

Figure 5: Per-task sensitivity of DCDT to α_2 across three reference-subset sizes (D_1=top-200, D_2=top-30, D_3=top-10 trajectories). α_2=0.05 is consistently the best or tied-best setting on every task; performance degrades gracefully across a $40\times$ range without catastrophic failure.

Table 12: Maze2D stitching benchmark: mean $\pm$ std normalized score across four mazes under sparse and dense reward.

Reward	Task	CQL	DT	QT	VDT	DCDT
Sparse	maze2d-open	208.7 ± 80.9	107.9 ± 39.6	452.3 ± 24.4	156.9 ± 15.5	$\mathbf{490.5 \pm 18.3}$
	maze2d-umaze	91.7 ± 21.1	11.0 ± 21.3	98.7 ± 4.8	94.1 ± 5.1	$\mathbf{113.8 \pm 9.8}$
	maze2d-medium	41.9 ± 14.6	1.4 ± 4.4	86.0 ± 6.2	79.9 ± 0.5	$\mathbf{88.8 \pm 4.0}$
	maze2d-large	47.6 ± 8.4	-0.6 ± 0.9	108.1 ± 2.5	43.34 ± 4.6	$\mathbf{209.1 \pm 4.1}$
Dense	maze2d-open	298.6 ± 43.4	302.2 ± 14.3	508.4 ± 1.9	343.5 ± 9.6	$\mathbf{553.5 \pm 15.8}$
	maze2d-umaze	67.0 ± 12.1	-6.8 ± 10.9	97.3 ± 7.8	96.7 ± 4.7	$\mathbf{151.5 \pm 3.3}$
	maze2d-medium	69.3 ± 9.8	12.5 ± 3.7	106.1 ± 7.1	102.6 ± 0.5	$\mathbf{108.5 \pm 6.8}$
	maze2d-large	90.6 ± 7.4	25.88 ± 11.2	117.2 ± 7.8	64.54 ± 2.8	$\mathbf{145.3 \pm 2.0}$

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:

   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 have a limitations paragraph.

   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: The proof is included in the appendix text.

   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 details are reported between main text and the appendix. Furthermore, we have released the code.

   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: [Yes]

   Justification: Code has been released.

   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: The most important details are available in the main text and the rest is reported in the Appendix.

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: Yes, we report the standard deviation.

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: We provide information in appendix.

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:

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:

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:

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: [N/A]

    Justification:

    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:

    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:

    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:

    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:

    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.