Critic-Guided Decision Transformer for Offline Reinforcement Learning

Yuanfu Wang[*1, 2], **Chao Yang**[*2], **Ying Wen**[†1], **Yu Liu**[2, 3], **Yu Qiao**[†2]

[1]Shanghai Jiao Tong University
[2]Shanghai Artificial Intelligence Laboratory
[3]SenseTime Research
{wangyuanfu,yangchao,qiaoyu}@pjlab.org.cn, ying.wen@sjtu.edu.cn, liuyuisanai@gmail.com

Abstract

Recent advancements in offline reinforcement learning (RL) have underscored the capabilities of Return-Conditioned Supervised Learning (RCSL), a paradigm that learns the action distribution based on target returns for each state in a supervised manner. However, prevailing RCSL methods largely focus on deterministic trajectory modeling, disregarding stochastic state transitions and the diversity of future trajectory distributions. A fundamental challenge arises from the inconsistency between the sampled returns within individual trajectories and the expected returns across multiple trajectories. Fortunately, value-based methods offer a solution by leveraging a value function to approximate the expected returns, thereby addressing the inconsistency effectively. Building upon these insights, we propose a novel approach, termed the Critic-Guided Decision Transformer (CGDT), which combines the predictability of long-term returns from value-based methods with the trajectory modeling capability of the Decision Transformer. By incorporating a learned value function, known as the critic, CGDT ensures a direct alignment between the specified target returns and the expected returns of actions. This integration bridges the gap between the deterministic nature of RCSL and the probabilistic characteristics of value-based methods. Empirical evaluations on stochastic environments and D4RL benchmark datasets demonstrate the superiority of CGDT over traditional RCSL methods. These results highlight the potential of CGDT to advance the state of the art in offline RL and extend the applicability of RCSL to a wide range of RL tasks.

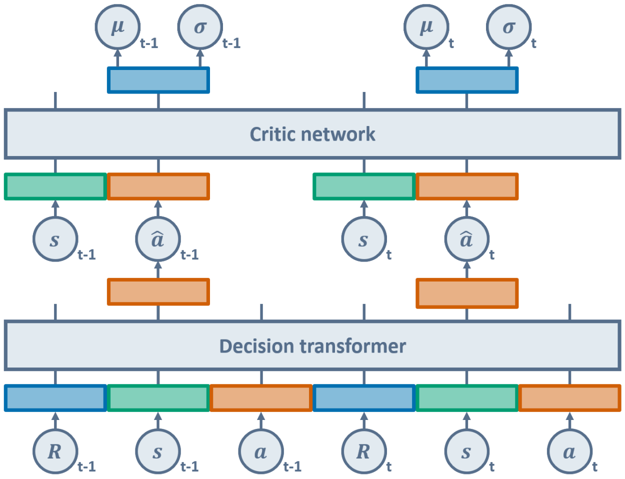

Figure 1: *Critic-Guided Decision Transformer framework*. The lower part is a vanilla Decision Transformer that takes the states s, actions a, and target returns R as inputs to predict the next action $\hat{a}_t$ for each state s_t. The predicted actions are then passed through a critic, which is a Gaussian distribution with expected return mean μ_t and variance σ_t learned from offline data. By minimizing the distance between the expected returns of the predicted actions and the target returns, e.g. $\|(R_t - \mu_t)/\sigma_t\|_2$, the critic guides the policy to take actions that are consistent with the target returns.

Introduction

Offline reinforcement learning (RL) addresses the problem of deriving effective policies from existing datasets that capture agent behaviors without interactions with environments. A naive solution to offline RL is imitation learning (IL) (Hussein et al. 2017), which aims to emulate the behaviors of policies represented in the dataset. However, IL is limited in its ability to distinguish between optimal and suboptimal trajectories without predefined returns, often resulting in suboptimal policies that mirror the distribution of the training data. To overcome the limitations of IL, recent research has introduced Return-Conditioned Supervised Learning (RCSL). Representative works such as the Decision Transformer (DT) (Chen et al. 2021) and RvS (Emmons et al. 2021) take cumulative returns or average returns as conditions to train a conditional policy that can differentiate desired (optimal) behaviors from the dataset.

However, RCSL struggles in stochastic environments and scenarios that require stitching abilities (Paster, McIlraith, and Ba 2022), where the policy needs to combine actions from suboptimal trajectories. The core issue limiting the performance of RCSL is the inconsistency between the sampled target (desired) returns and the expected returns of actions. In other words, trajectories with higher returns does not necessarily imply that their actions are superior to others; they can be a result of luck. RCSL treats the return-to-go

[*]Equal contribution.

[†]Correspondence to Ying Wen, Yu Qiao.

(RTG) as a quantity tied to a single trajectory, neglecting the stochastic state transitions and the broader distribution of future outcomes (Brandfonbrener et al. 2022; Bhargava et al. 2023). This inconsistency is further exacerbated by the inherent uncertainty and approximation errors within behavior policies, resulting in inferior performance in stitching problems where suboptimal data present.

Fortunately, value-based methods (Sutton, Barto et al. 1998), on the other hand, provide a robust solution to handle this inconsistency. These methods estimate the expected cumulative returns of actions for each state, enabling an agent to choose optimal actions that maximize long-term returns. Q-learning algorithms (Kumar et al. 2020; Peng et al. 2019; Kostrikov, Nair, and Levine 2021; Xu et al. 2023), in particular, utilize temporal difference (TD) updates to learn a value function, allowing effective policy learning even in stochastic environments with highly suboptimal trajectories.

To address the limitations of RCSL, we propose a novel approach called the Critic-Guided Decision Transformer (CGDT). Our approach combines the predictability of long-term returns from value-based methods with the Decision Transformer framework. By utilizing a value function, known as the critic, to guide policy training, CGDT ensures that the expected returns of actions align with the specified target returns. This integration enables CGDT to effectively handle both stochastic environments and stitching scenarios, while still allowing conditional action selection.

In this paper, we evaluate our proposed approach on stochastic environments and D4RL benchmark datasets. Our experimental results demonstrate significant improvements over pure RCSL in both stochastic environments and stitching problems. This showcases the potential of our method to advance the state-of-the-art in offline RL. Furthermore, our proposed approach holds promise for various RL tasks. We summarize our contributions as follows:

- We provide an intuitive explanation for the pitfalls of RCSL in stochastic environments and stitching scenarios, arising from the inconsistency between the target (desired) returns and the expected returns of actions.
- We propose a novel approach, Critic-Guided Decision Transformer (CGDT), which leverages a critic to handle stochasticity from environments and uncertainty from suboptimal data while preserving the capability to act on variable conditional inputs.
- We evaluate our method on various benchmarks, including a Bernoulli Bandit game with stochastic rewards and D4RL benchmark datasets, and analyze how the use of the critic handles stochasticity and benefits CGDT.

Related Work

Offline RL

Offline RL has seen the emergence of several methodologies to address the challenges of learning from fixed datasets (Prudencio, Maximo, and Colombini 2023). These methodologies can be categorized into value-free and value-based approaches. Value-free approaches do not necessarily rely on value functions. One such approach is *Imitation learning* (Hussein et al. 2017), which aims to imitate the behavior policy by training on collected or desired trajectories filtered by heuristics or value functions (Chen et al. 2020; Wang et al. 2020). *Trajectory Optimization*, e.g. Multi-Game Decision Transformer (MGDT) (Lee et al. 2022) and Trajectory Transformer (TT) (Janner, Li, and Levine 2021), models joint state-action distribution over complete trajectories, reducing out-of-distribution (OOD) action selection. To enable effective planning, this approach utilizes techniques such as beam search and reward estimates. Contrarily, value-based methods rely on value functions. Two common approaches in this category are *Policy Constraints* and *Regularization*. Policy Constraints ensure that the learned policy remains close to the behavioral policy, either through direct estimation (Fujimoto, Meger, and Precup 2019; Kostrikov et al. 2021) or implicit modifications of the learning objective (Kumar et al. 2019; Peng et al. 2019). Regularization, e.g. CQL (Kumar et al. 2020) and IQL (Kostrikov, Nair, and Levine 2021), introduces penalty terms to influence policy behaviors without explicitly estimating behavioral policy.

Return-Conditioned Supervised Learning

Return-Conditioned Supervised Learning (RCSL) is a newly emergent class of algorithms that learns action distribution based on future returns statistics for each state via supervised learning (Schmidhuber 2019; Kumar, Peng, and Levine 2019). By conditioning on target returns, the policy can generate actions that closely resemble the behaviors presented in the dataset. Decision Transformers (DT) and its variants (Siebenborn et al. 2022; Zheng, Zhang, and Grover 2022; Hu et al. 2023; Wen et al. 2023) use returns-to-go, i.e. cumulative future returns, as the conditional inputs and model trajectories with causal transformers (Vaswani et al. 2017). RvS (Emmons et al. 2021) investigates the effectiveness of conditioning on future states and average rewards. These approaches explore the capabilities of different conditional inputs in various environments. Generalized Decision Transformer (Furuta, Matsuo, and Gu 2021) reveals that all these conditional supervised learning approaches are doing *hindsight information matching* (HIM) indeed, i.e. to match the output trajectories with future information statistics.

Pitfalls of RCSL

In this section, we provide an intuitive explanation for the underlying reasons behind the limitations of RCSL in offline RL, particularly in the context of stochastic environments and scenarios involving stitching abilities. By focusing on these specific scenarios, we aim to reveal the fundamental factors that contribute to RCSL's failures in these settings.

Stochasticity of Transitions

RCSL faces significant challenges when applied to stochastic environments, even when provided with infinite data and without any approximation errors (Paster, McIlraith, and Ba 2022; Brandfonbrener et al. 2022). Consider the MDP depicted in Figure 2, which presents two available actions, a_1 and a_2. Taking action a_1 leads to a future state s_1' with a low probability $P = 0.01$ of attaining high returns $R_1' =$

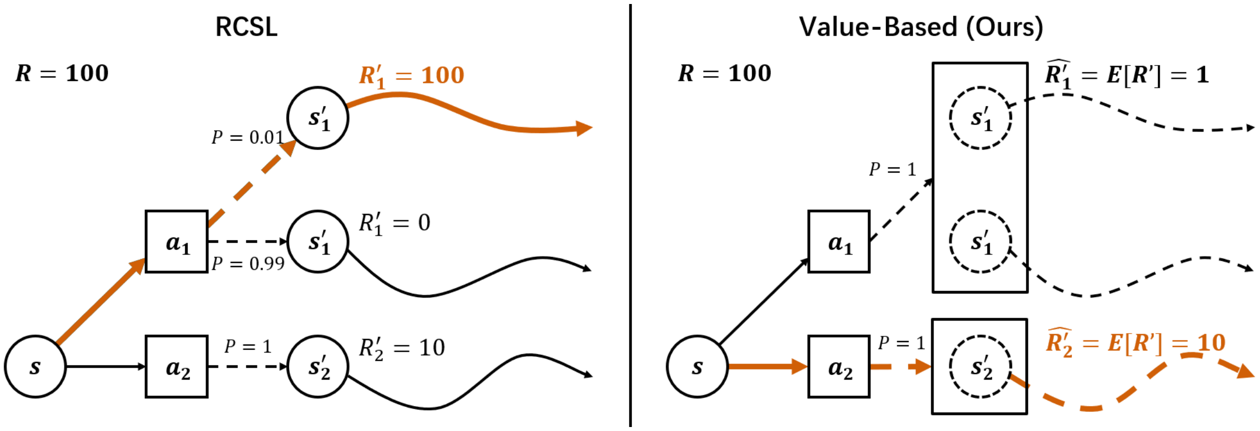

Figure 2: *Illustration of RCSL and Value-Based.* Considering a MDP where taking action a_1 has a low probability $P = 0.01$ of leading to a future state s'_1 with high return $R'_1 = 100$, while the optimal action a_2 deterministically results in a future state s'_2 with return $R'_2 = 10$. *Left*: RCSL selects action a_1 based on the target return $R = 100$, regardless of its low probability, following the action distribution in the dataset. *Right*: Value-based methods utilize a value function to estimate the expected returns over multiple trajectories and guide the policy to take actions aligned with the target return $R = 100$. The colored arrows indicate the learned behaviors conditioning on the specified target return $R = 100$, with both actions sampled in the dataset.

100, while the remaining probability yields no returns. Conversely, taking action a_2 guarantees a future state s'_2 with returns $R'_2 = 10$, making it the optimal choice. RCSL (left of Figure 2) models individual trajectories without considering transition probabilities. As a result, it may select the suboptimal action a_1 to attain the target return of $R = 100$, despite its low probabilities, following the action distribution in the dataset. This example illustrates that **trajectories with higher returns do not imply that their actions are superior to others; they could be a result of luck**.

To address the pitfall of RCSL in stochastic environments, alternative approaches have been proposed. For instance, Paster, McIlraith, and Ba (2022) propose environment-stochasticity-independent representations (ESPER), which cluster trajectories and utilize the average cluster returns as conditions for the policy. Similarly, Q-Learning Decision Transformer (QDT) (Yamagata, Khalil, and Santos-Rodriguez 2023), utilizes a conservative value function to relabel return-to-go in the dataset. DoC (Yang et al. 2022) conditions the policy on a latent representation of future trajectories that is agnostic to stochasticity, achieved by minimizing mutual information. These approaches reply on *Stochasticity-Independent Representations*, by defining or learning a representation of future trajectories that is independent of environment stochasticity.

Although these methods exhibit efficacy in stochastic contexts, they come with their limitations. Methods that rely on relabeling returns-to-go (ESPER, QDT) may struggle with unmatched return-to-go values during inference. DoC requires complex objectives for representation learning and additional steps like sampling and value estimation..

Stitching in Offline RL

Indeed, the limitations of RCSL are not confined to stochastic settings. RCSL methods may exhibit suboptimal performance even in deterministic environments when suboptimal data is prevalent (Li et al. 2023). In deterministic environments, **the uncertainty and approximation errors within the behavior policy introduce a form of stochasticity that resembles environmental stochasticity**. As a result, a tradeoff emerges between aggressive actions that may yield high returns but are hard to replicate, and conservative actions that consistently offer moderate returns. RCSL often prioritize the former, resulting in suboptimal performance.

Essentially, methods based on *Stochasticity-Independent Representations* leverage probabilistic statistics from multiple trajectories to guide action selection. Elastic Decision Transformer (Wu, Wang, and Hamaya 2023) dynamically adjusts the context length of DT during inference, allowing it to ”stitch” with an optimal trajectorywhile incorporating a value function without replacing the return-to-go. MGDT and TT use either reward estimates or value functions to sample optimal actions during the planning stage.

From our perspective, incorporating probabilistic statistics from multiple trajectories offers a promising solution for stochasticity and suboptimal data. As depicted in Figure 2 (right), these methods guide policy behaviors with learned expected returns from the entire distribution of future trajectories. These approaches ensure that the output actions align with the desired target returns in statistic, resolving the inconsistency arising from trajectory-level modeling.

Method

In this section, we first introduce the preliminary notations for offline RL and RCSL. Then, we propose the learning objective of training critics from the offline dataset and the learning objective of training policy with critic guidance which ensures the expected returns of actions are consistent with desired returns. Finally, we introduce Critic-Guided Decision Transformer, a practical framework for optimizing policy with critic-guided learning objectives.

Preliminaries

In the offline RL setup, the objective is to train an agent solely from an existing dataset $\mathcal{D}$ of trajectories $\tau = (s_0, a_0, r_0, \cdots, s_{T-1}, a_{T-1}, r_{T-1})$ of states $s_t \in \mathcal{S}$, actions $a_t \in \mathcal{A}$, and rewards $r_t \in \mathcal{R}$ sampled by a behavior policy π_β interacting with a finite horizon Markov Decision Process (Sutton, Barto et al. 1998) with horizon T. We use $\tau_{i:j} := (s_t, a_t, r_t)_{t=i}^{j}$ to denote a sub-trajectory. Let $R(\tau) = \sum_{t=0}^{T} r_t$ denotes the cumulative return of the trajectory τ. The goal of RL is to learn a policy that maximizes the expected cumulative return $\mathbb{E}\left[R(\tau)\right] = \mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$.

In RCSL, we denote the return of a trajectory at timestep t as $R_t := \sum_{t'=t}^{T} r_{t'}$. R_0 is also known as the target (desired) return. Let π_θ denotes the learning policy parameterized by θ. The objective of RCSL is typically to minimize the empirical negative log-likelihood loss (NLL), given by:

$$\mathcal{L}(\theta) = \mathbb{E}_{\tau \in \mathcal{D}}\left[-\sum_{0 \le t < H} \log\left(\pi_\theta(a_t|\tau_{0:t-1}, s_t, R_t)\right)\right].$$

During inference, the target return is substituted with a manually specified target return, often chosen as the maximum return among the trajectories in the dataset.

Asymmetric Critic Training

Following Bayes' rule, we can express the probability of taking an action a_t, given a return R_t and a state s_t as: $p(a_t|R_t, s_t) \propto p(a_t|s_t)p(R_t|s_t, a_t)$, which suggests that to guide the action selection towards desired returns, we can model the probability distribution $p(R_t|s_t, a_t)$. Nonetheless, this distribution is typically unknown. Instead, we propose a parameterized critic $Q_\phi(R_t|\tau_{0:t-1}, s_t, a_t)$, which approximates $p(R_t|s_t, a_t)$ as a Gaussian distribution with learnable mean and variance (Bellemare, Dabney, and Munos 2017). This critic is trained using an offline dataset $\mathcal{D}$ with NLL loss as the objective:

$$\mathcal{L}(\phi) = -\log Q_\phi(R_t|\tau_{0:t-1}, s_t, a_t). \tag{1}$$

However, the quality of the data in $\mathcal{D}$ can be unbalanced, containing both optimal and suboptimal trajectories. To address this issue, we introduce an asymmetric NLL loss as the revised learning objective of fitting the critic:

$$\mathcal{L}_Q(\phi) = -|\tau_c - \mathbb{I}(u > 0)| \log Q_\phi(R_t|\tau_{0:t-1}, s_t, a_t), \tag{2}$$

where $u = (R_t - \mu_t)/\sigma_t$, and $(\mu_t, \sigma_t) \sim Q_\phi(\cdot|\tau_{0:t-1}, s_t, a_t)$ represent the mean and variance of the estimated return at current state s_t and action a_t, respectively. Here, $\tau_c \in (0, 1)$ is an adjustable coefficient that controls the asymmetry of the loss. When $\tau_c > 0.5$, the critic is biased towards fitting optimal trajectories., while $\tau_c < 0.5$ biases the critic towards suboptimal trajectories. Setting $\tau_c = 0.5$ corresponds to using a scaled standard NLL loss in Equation 1.

Asymmetric Critic Guidance

To encourage the selection of optimistic actions with expected returns higher than the target returns, we adopt the approach of *Expectile Regression*, a variant of mean regression commonly used for estimating statistics of a random variable. Inspired by IQL (Kostrikov, Nair, and Levine 2021), our method utilizes the critic to guide action selection using the following objective:

Algorithm 1: Critic-Guided Decision Transformer

```
Input: Offline dataset $\mathcal{D}$, critic $Q_\phi$, policy $\pi_\theta$, iterations $M$, $N$, asymmetric critic coefficient $\tau_c$, expectile regression parameter $\tau_p$, and balance weight $\alpha$.
  \\ Asymmetric Critic Training
  for $i = 1, ..., M$ do
    Sample a batch of trajectories $(s_t, a_t, r_t)$ from $\mathcal{D}$;
    Compute return of sub-trajectory $\tau_{t:T}$, $R_t = \sum_t^T r_t$;
    Update $Q_\phi$ with gradient:
        $\mathbb{E}_{(s_t,a_t,R_t)}\left[\nabla_\phi \mathcal{L}_Q(\phi)\right];$
  end for
  \\ Critic-Guided Policy Training
  $\alpha' \leftarrow 0$
  for $j = 1, ..., N$ do
    $\alpha' \leftarrow \alpha' + \alpha/N$
    Sample a batch of trajectories $(s_t, a_t, r_t)$ from $\mathcal{D}$;
    Compute return of sub-trajectory $\tau_{t:T}$, $R_t = \sum_t^T r_t$;
    Predict action $\hat{a}_t \sim \pi_\theta(\cdot|\tau_{0:t-1}, s_t, R_t)$;
    Predict return $(\mu_t, \sigma_t) \sim Q_\phi(\cdot|\tau_{0:t-1}, s_t, \hat{a}_t)$;
    Compute expectile regression loss: $\mathcal{L}_2^{\tau_p}(\frac{R_t - \mu_t}{\sigma_t})$;
    Update $\pi_\theta$ with gradient:
        $\mathbb{E}_{(s_t,a_t,\hat{a}_t,R_t)}\left[\nabla_\theta \mathcal{L}_2(a_t, \hat{a}_t) + \alpha' \nabla_\theta \mathcal{L}_2^{\tau_p}(\frac{R_t - \mu_t}{\sigma_t})\right];$
  end for
  return $\pi_\theta$
```

$$\mathcal{L}_2^{\tau_p}(u) = |\tau_p - \mathbb{I}(u < 0)|u^2, \tag{3}$$

where $u = (R_t - \mu_t)/\sigma_t$ and $(\mu_t, \sigma_t) \sim Q_\phi(\cdot|\tau_{0:t-1}, s_t, \hat{a}_t)$. Here, $\hat{a}_t$ is sampled from the policy $\pi_\theta(\cdot|\tau_{0:t-1}, s_t, R_t)$. The variables μ_t and σ_t represent the mean and variance of the estimated return at the current state s_t and predicted action $\hat{a}_t$, respectively. The adjustable coefficient τ_p lies in the range $(0, 1)$. When $\tau_p = 0.5$, it is equivalent to mean regression, which estimates the mean of the random variables. By adjusting τ_p, we introduce asymmetry into the mean regression. In Equation 3, a large $\tau_p > 0.5$ approximates a lower expectile of the advantage of estimated expected returns over target returns, i.e. $u = (R_t - \mu_t)/\sigma_t > 0$, vice versa. Consequently, it guides the policy to select optimistic actions with higher expected returns than those conditioned on.

Critic-Guided Decision Transformer

Building upon the proposed learning objectives for critic training and critic guidance, we present the Critic-Guided Decision Transformer framework in Figure 1, which provides a practical approach for optimizing policy with critic-guided learning objective. Firstly, we evaluate the behavior policy by training a critic that estimates the cumulative returns of actions in the dataset through revised maximum likelihood estimation, as described in Equation 2. Then, we proceed with one-step policy improvement by optimizing the policy to select actions with expected returns that are consistent with or slightly better than the target returns. This approach eliminates the need for off-policy evaluation and has the potential to yield the same policy learned by multi-step critic regularization methods (Eysenbach et al. 2023).

Notably, the critic trained on the offline dataset may suffer from overestimating out-of-distribution actions. Solely optimizing the critic guidance term (Equation 3) may mislead the policy towards overestimated actions. To mitigate this problem and ensure that the policy remains close to the data distribution, we introduce the vanilla RCSL learning objective as a constraint. In practice, this objective is implemented as an l^2-norm term on actions, discouraging the selection of actions that deviate significantly from the data distribution. Consequently, the overall policy loss is formulated as:

$$\mathcal{L}_\pi(\theta;\alpha) = \mathcal{L}_2(a_t, \hat{a}_t) + \alpha \cdot \mathcal{L}_2^{\tau_p}(\tfrac{R_t - \mu_t}{\sigma_t}), \tag{4}$$

where $\hat{a}_t \sim \pi_\theta(\cdot|\tau_{0:t-1}, s_t, R_t)$, and (μ_t, σ_t) are sampled from the critic Q_ϕ. Additionally, α is a balance weight.

In practical implementation, we utilize validation errors as a means to detect overfitting during critic training, which can monitor the performance of the critic model and apply early stopping to mitigate overestimation issues. Additionally, we introduce a linearly increasing weight coefficient to balance the action regression loss and the critic guidance term during policy training. This coefficient eliminates the need for manual tuning and provides a mechanism to control policy behavior automatically. The algorithm implementation details are summarized in Algorithm 1.

Experiments

We conduct a series of experiments to address the following questions: **(i)** How effectively does CGDT handle stochasticity in the Bernoulli bandit problem with stochastic rewards? **(ii)** How does critic guidance benefit the performance of CGDT in the presence of suboptimal data with sparse and dense rewards? **(iii)** How consistent are the returns achieved by CGDT with the specified target returns? Additionally, we conduct ablation studies to examine the influence of asymmetry in critic training and critic guidance.

Bernoulli Bandit (Stochasticity)

To evaluate the ability of an agent to handle environmental stochasticity, we employ a two-armed Bernoulli bandit game with stochastic rewards, following Yang et al. (2022). As illustrated in Figure 3, the game consists of two arms, denoted as a_1 and a_2, which generate stochastic rewards drawn from Bernoulli distributions of $Bern(1-p)$ and $Bern(p)$, respectively. Arm a_1 yields a non-zero reward with probability $1-p$, while arm a_2 does so with probability p. In this setup, a smaller value of p (i.e., $p < 0.5$) makes arm a_1 the optimal choice with a higher expected return. To ensure a balanced occurrence of successful retrievals for both arms in the offline dataset, the behavior policy pulls arm a_1 with probability $\pi_\beta(a_1|s) = p$.

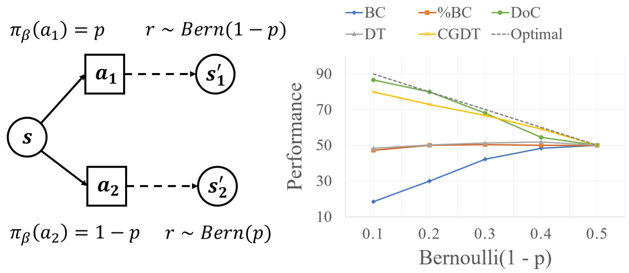

Figure 3: *Bernoulli Bandit*. *Left*: A two-armed bandit, following Yang et al. (2022), to evaluate the performance of CGDT under environment with stochastic rewards. *Right*: DoC, CGDT achieve close to Bayes-optimal (dotted), while BC and DT fail. Average normalized scores over 5 random seeds are reported, each evaluated for 1000 episodes.

We implement our approach and baselines as stochastic policies, except for DoC, which samples actions from a prior distribution and uses a value function to distinguish the action with highest return. We train these methods using 10,000 samples, where the probability p varied in the range $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. In Figure 3 (right), we observe that supervised learning (SL) methods (BC, %BC) and RCSL (DT) methods converge to suboptimal behaviors without value functions. However, both CGDT and DoC achieve results close to the Bayes-optimal. The ability to handle stochasticity using a value function is a significant advantage of CGDT compared to SL and RCSL.

OpenAI Gym (Stitching)

To evaluate how critic guidance benefits CGDT in stitching problems, we conduct further experiments on the D4RL datasets (Fu et al. 2020). These datasets provide standardized environments and various datasets with different qualities. For our evaluation, we specifically focus on the MuJoCo locomotion tasks using the *medium*, *medium-replay*, and *medium-expert* datasets, along with navigation tasks. We compare our approach with SL and RCSL algorithms, e.g. %BC, RvS, DT, and QDT. Besides, we also compare with value-based algorithms, e.g. CQL and IQL, and trajectory optimization algorithms like TT.

We present the results in Table 1 with baseline results retrieved from the original papers. SL and RCSL methods perform well in high-quality datasets but struggle to achieve optimal performance in suboptimal datasets. Conversely, value-based methods (CQL and IQL) exhibit strong stitching abilities in suboptimal datasets but do not generalize well to high-quality datasets. Trajectory optimization, e.g. TT, demonstrates exceptional overall performance, showcasing the advantage of trajectory modeling. Our approach, CGDT, shows significant improvement over SL and RCSL methods in suboptimal data regimes, while maintaining its optimal performance in high-quality datasets.

To further investigate the advantages of utilizing a value function with limited reward signals, we evaluate our ap-

Dataset	Environment	10%BC*	RvS*	DT*	QDT*	CQL*	IQL*	TT*	DT	CGDT
Medium	Halfcheetah	42.5	41.6	42.6	42.2	44.0	**47.4**	46.9	42.7	**43.0**
Medium	Hopper	56.9	60.2	**67.6**	65.3	58.5	66.3	61.1	67.5	**96.9**
Medium	Walker2d	75.0	71.7	74.0	70.1	72.5	78.3	**79.0**	76.8	**79.1**
Medium-Replay	Halfcheetah	40.6	38.0	36.6	35.7	**45.5**	44.2	41.9	40.2	**40.4**
Medium-Replay	Hopper	75.9	73.5	82.7	55.3	**95.0**	94.7	91.5	88.3	**93.4**
Medium-Replay	Walker2d	62.5	60.6	66.6	59.1	77.2	73.9	**82.6**	73.0	**78.1**
Medium-Expert	Halfcheetah	92.9	92.2	86.8	/	91.6	86.7	**95.0**	93.1	**93.6**
Medium-Expert	Hopper	**110.9**	101.7	107.6	/	105.4	91.5	110.0	**108.6**	107.6
Medium-Expert	Walker2d	109.0	106.0	108.1	/	108.8	**109.6**	101.9	109.0	**109.3**
Sum		666.2	645.5	672.6	/	698.5	692.6	**722.5**	699.2	**741.5**
Umaze	Antmaze	62.8	64.4	59.2	/	74.0	87.5	**100.0**	61.0	**71.0**
Umaze-Diverse	Antmaze	50.2	70.1	53.0	/	**84.0**	62.2	/	55.0	**71.0**
Sum		113.0	134.5	112.2	/	**158.0**	149.7	/	116.0	**142.0**

Table 1: *Overall performance*. The Critic-Guided Decision Transformer (CGDT) demonstrates competitive or superior performance compared to prior offline RL algorithms on D4RL datasets. Particularly, on *medium* and *medium-replay* datasets where suboptimal data present, CGDT significantly outperforms RCSL methods such as RvS, DT, and QDT. Its performance is on par with value-based algorithms such as CQL and IQL, and trajectory optimization algorithms such as TT. Average normalized scores over 5 random seeds are reported, each evaluated for 100 episodes; *baseline results are taken from original papers.

Dataset	Environment	DT (sparse)	CGDT (sparse)	δ_{sparse}	DT (dense)	CGDT (dense)	δ_{dense}
Medium	Halfcheetah	42.7	**43.1**	0.4	42.7	**43.0**	0.3
Medium	Hopper	65.9	**78.1**	12.2	67.5	**96.9**	29.4
Medium	Walker2d	76.9	**79.9**	3.0	76.8	**79.1**	2.3
Medium-Replay	Halfcheetah	**40.9**	40.2	-0.7	40.2	**40.4**	0.2
Medium-Replay	Hopper	84.5	**86.3**	1.8	88.3	**93.4**	5.1
Medium-Replay	Walker2d	69.4	**73.9**	4.5	73.0	**78.1**	5.2
Medium-Expert	Halfcheetah	93.6	**93.6**	0.0	93.1	**93.6**	0.5
Medium-Expert	Hopper	**106.3**	106.2	-0.1	**108.6**	107.6	-1.0
Medium-Expert	Walker2d	107.3	**109.4**	2.1	109.0	**109.3**	0.4
Sum		687.4	**710.5**	23.1	699.2	**741.5**	42.4

Table 2: *Sparse Reward*. We evaluate the performance of Critic-Guided Decision Transformer (CGDT) in sparse (delayed) reward settings on D4RL locomotion tasks using the same hyperparameters as in dense reward settings, which might not be optimal. Critic guidance benefits CGDT in sparse reward settings, while dense rewards lead to a larger improvement over sparse rewards. Average normalized scores over 5 random seeds are reported, each evaluated for 100 episodes.

proach in scenarios with sparse (delayed) rewards, where the rewards are only granted at the final timestep of each trajectory. The results for the sparse reward case are presented in Table 2. Notably, we do not extensively tune the hyperparameters and use the same settings as in the dense reward settings, which may not be optimal for the sparse reward scenario. Nonetheless, we observe improved performance over DT on the sparse reward tasks, though, with lesser improvement as in the dense reward case. These findings indicate that critics trained in both sparse and dense reward settings contribute to the benefits of CGDT, with more substantial improvements observed in dense reward scenarios.

Conditional Behaviors (Consistency)

One significant difference between RCSL and value-based algorithms is the ability to behave conditionally. Conditional behaviors, which are characteristic of RCSL, are not typically exhibited by most value-based algorithms. This is because the limited model capacity and potentially conflicting learning objectives in training may jeopardize the optimal performance. However, the ability to exhibit conditional behaviors is essential for controllability and flexibility, enabling application in a wider range of scenarios.

We evaluate the conditional behaviors of our approach and DT under different target returns $R' = \lambda R$, where $\lambda \in \{0, 0.2, \cdots, 1\}$ and R represents the original target return. From Figure 4, we observe that DT naturally exhibits conditional behaviors but shows weakened consistency, particularly when the datasets do not include suboptimal data (e.g., *medium* and *medium-expert* datasets). In contrast, CGDT sticks to target returns more closely, even in datasets where suboptimal data are absent. This result indicates an improved consistency of CGDT between the expected returns of actions and the target returns.

Ablation Study

To investigate the effect of asymmetry in critic training and policy training, we conduct a series of ablation experiments.

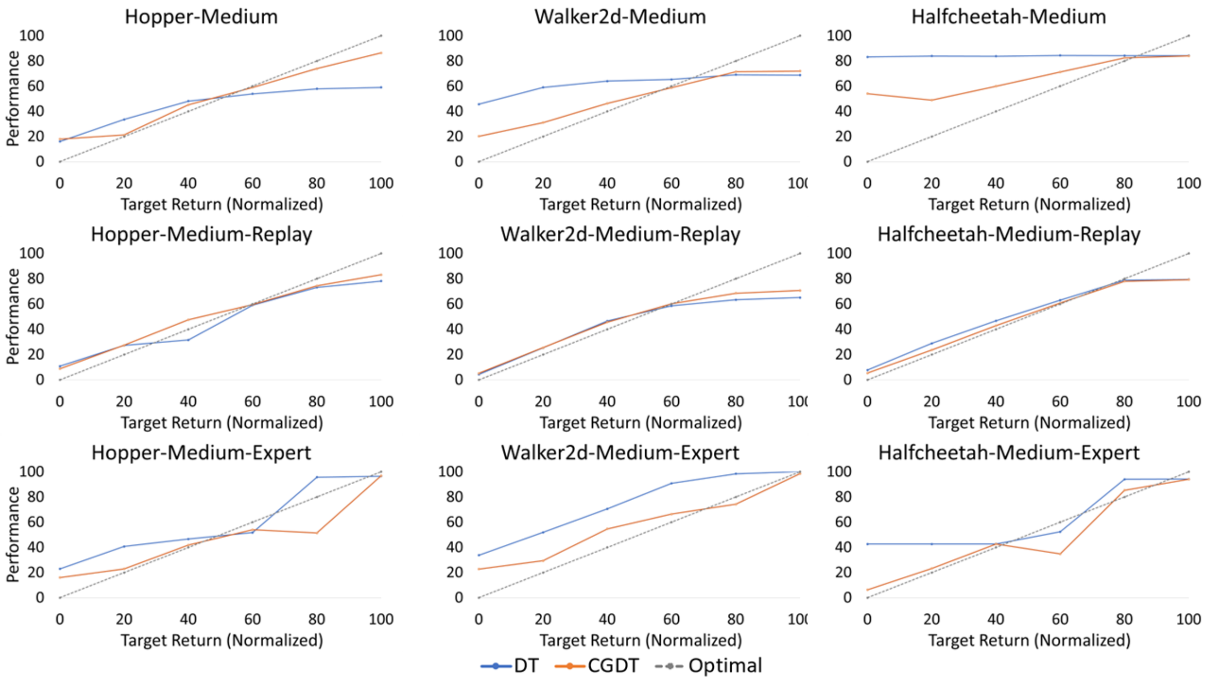

Figure 4: *Conditional Behaviors*. Evaluation returns achieved by DT and CGDT when conditioned on the specified target returns. The dotted lines denote the optimal behaviors. It is observed that CGDT sticks to the target returns more closely than DT, which indicates a more consistent behavior observed in CGDT.

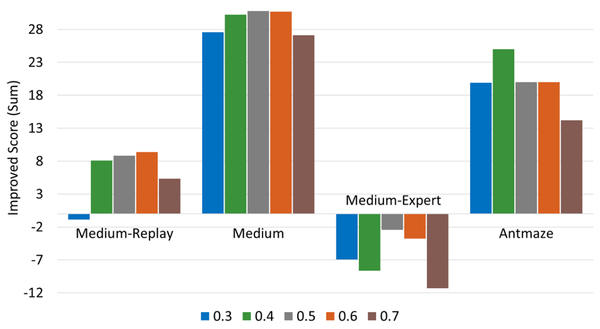

Figure 5: *Ablations of τ_c in critic training*. We show the improved performance of CGDT over DT with different $\tau_c \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ in Equation 2. A large τ_c indicates a bias over high-quality data, and vice versa.

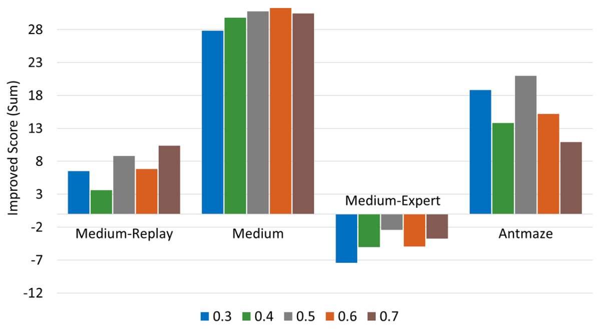

Figure 6: *Ablations of τ_p in policy training*. We show the improved performance of CGDT over DT with different $\tau_p \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ in Equation 3. A large τ_p favors actions with higher expected returns, and vice versa.

Initially, we set the hyperparameters τ_c and τ_p to 0.5. By varying τ_c and τ_p within the range of $[0.3, 0.7]$, we control the asymmetries during critic training and policy training, respectively. Notably, datasets with different distributions exhibit varying preferences. In general, during critic training, there is a tendency to fit the model towards high-quality data. On the other hand, during policy training, optimistic actions with higher expected returns are favored. The *Antmaze* datasets present a special case, which primarily consist of truncated trajectories with zero returns, where low returns do not necessarily indicate suboptimal performance.

Conclusion

We propose Critic-Guided Decision Transformer, a general framework for RCSL that utilizes a value function, to guide the trajectory modeling process. This framework effectively solves the inconsistency between the expected returns of actions and the target returns while preserving the conditional characteristic of RCSL. Our empirical results demonstrate that CGDT is highly capable of handling stochastic environments and addressing challenges in stitching problems posed by suboptimal data, which might provide fresh insights for extending the application of RCSL to broader domains.

Acknowledgments

This research is funded by the Shanghai Artificial Intelligence Laboratory. It also received partial support from the Shanghai Post-doctoral Excellent Program (Grant No. 2022234). Additionally, the team from Shanghai Jiao Tong University is supported by the National Key R&D Program of China (2022ZD0114804), the National Natural Science Foundation of China (No. 62106141), and the Shanghai Sailing Program (21YF1421900). This work is done during Yuanfu's internship at Shanghai Artificial Intelligence Laboratory. We sincerely thank our colleagues for engaging in insightful discussions and providing valuable feedback, with a particular acknowledgement to Ming Zhou and Zhanhui Zhou

References

Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A Distributional Perspective on Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 449–458. JMLR.org.

Bhargava, P.; Chitnis, R.; Geramifard, A.; Sodhani, S.; and Zhang, A. 2023. Sequence Modeling is a Robust Contender for Offline Reinforcement Learning. *arXiv preprint arXiv:2305.14550*.

Brandfonbrener, D.; Bietti, A.; Buckman, J.; Laroche, R.; and Bruna, J. 2022. When does return-conditioned supervisejanner2021offlined learning work for offline reinforcement learning? *Advances in Neural Information Processing Systems*, 35: 1542–1553.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34: 15084–15097.

Chen, X.; Zhou, Z.; Wang, Z.; Wang, C.; Wu, Y.; and Ross, K. 2020. Bail: Best-action imitation learning for batch deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 18353–18363.

Emmons, S.; Eysenbach, B.; Kostrikov, I.; and Levine, S. 2021. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*.

Eysenbach, B.; Geist, M.; Levine, S.; and Salakhutdinov, R. 2023. A Connection between One-Step Regularization and Critic Regularization in Reinforcement Learning. arXiv:2307.12968.

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv:2004.07219.

Fujimoto, S.; Meger, D.; and Precup, D. 2019. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, 2052–2062. PMLR.

Furuta, H.; Matsuo, Y.; and Gu, S. S. 2021. Generalized decision transformer for offline hindsight information matching. *arXiv preprint arXiv:2111.10364*.

Hu, S.; Shen, L.; Zhang, Y.; and Tao, D. 2023. Graph Decision Transformer. *arXiv preprint arXiv:2303.03747*.

Hussein, A.; Gaber, M. M.; Elyan, E.; and Jayne, C. 2017. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2): 1–35.

Janner, M.; Li, Q.; and Levine, S. 2021. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34: 1273–1286.

Kostrikov, I.; Fergus, R.; Tompson, J.; and Nachum, O. 2021. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, 5774–5783. PMLR.

Kostrikov, I.; Nair, A.; and Levine, S. 2021. Offline Reinforcement Learning with Implicit Q-Learning. arXiv:2110.06169.

Kumar, A.; Fu, J.; Soh, M.; Tucker, G.; and Levine, S. 2019. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32.

Kumar, A.; Peng, X. B.; and Levine, S. 2019. Reward-conditioned policies. *arXiv preprint arXiv:1912.13465*.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191.

Lee, K.-H.; Nachum, O.; Yang, M. S.; Lee, L.; Freeman, D.; Guadarrama, S.; Fischer, I.; Xu, W.; Jang, E.; Michalewski, H.; et al. 2022. Multi-game decision transformers. *Advances in Neural Information Processing Systems*, 35: 27921–27936.

Li, W.; Luo, H.; Lin, Z.; Zhang, C.; Lu, Z.; and Ye, D. 2023. A survey on transformers in reinforcement learning. *arXiv preprint arXiv:2301.03044*.

Paster, K.; McIlraith, S.; and Ba, J. 2022. You can't count on luck: Why decision transformers and rvs fail in stochastic environments. *Advances in Neural Information Processing Systems*, 35: 38966–38979.

Peng, X. B.; Kumar, A.; Zhang, G.; and Levine, S. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*.

Prudencio, R. F.; Maximo, M. R.; and Colombini, E. L. 2023. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*.

Schmidhuber, J. 2019. Reinforcement Learning Upside Down: Don't Predict Rewards–Just Map Them to Actions. *arXiv preprint arXiv:1912.02875*.

Siebenborn, M.; Belousov, B.; Huang, J.; and Peters, J. 2022. How crucial is transformer in decision transformer? *arXiv preprint arXiv:2211.14655*.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Z.; Novikov, A.; Zolna, K.; Merel, J. S.; Springenberg, J. T.; Reed, S. E.; Shahriari, B.; Siegel, N.; Gulcehre, C.; Heess, N.; et al. 2020. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33: 7768–7778.

Wen, M.; Lin, R.; Wang, H.; Yang, Y.; Wen, Y.; Mai, L.; Wang, J.; Zhang, H.; and Zhang, W. 2023. Large sequence models for sequential decision-making: a survey. *Frontiers of Computer Science*, 17(6): 176349.

Wu, Y.-H.; Wang, X.; and Hamaya, M. 2023. Elastic Decision Transformer. *arXiv preprint arXiv:2307.02484*.

Xu, H.; Jiang, L.; Li, J.; Yang, Z.; Wang, Z.; Chan, V. W. K.; and Zhan, X. 2023. Offline rl with no ood actions: In-sample learning via implicit value regularization. *arXiv preprint arXiv:2303.15810*.

Yamagata, T.; Khalil, A.; and Santos-Rodriguez, R. 2023. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. In *International Conference on Machine Learning*, 38989–39007. PMLR.

Yang, M.; Schuurmans, D.; Abbeel, P.; and Nachum, O. 2022. Dichotomy of control: Separating what you can control from what you cannot. *arXiv preprint arXiv:2210.13435*.

Zheng, Q.; Zhang, A.; and Grover, A. 2022. Online decision transformer. In *international conference on machine learning*, 27042–27059. PMLR.