DICHOTOMY OF CONTROL: SEPARATING WHAT YOU CAN CONTROL FROM WHAT YOU CANNOT

Anonymous authors
Paper under double-blind review

ABSTRACT

Future- or return-conditioned supervised learning is an emerging paradigm for offline reinforcement learning (RL), where the future outcome (i.e., return) associated with an observed action sequence is used as input to a policy trained to imitate those same actions. While return-conditioning is at the heart of popular algorithms such as decision transformer (DT), these methods tend to perform poorly in highly stochastic environments, where an occasional high return can arise from randomness in the environment rather than the actions themselves. Such situations can lead to a learned policy that is *inconsistent* with its conditioning inputs; i.e., using the policy to act in the environment, when conditioning on a specific desired return, leads to a distribution of real returns that is wildly different than desired. In this work, we propose the *dichotomy of control* (DoC), a future-conditioned supervised learning framework that separates mechanisms within a policy's control (actions) from those beyond a policy's control (environment stochasticity). We achieve this separation by conditioning the policy on a latent variable representation of the future, and designing a mutual information constraint that removes any information from the latent variable associated with randomness in the environment. Theoretically, we show that DoC yields policies that are *consistent* with their conditioning inputs, ensuring that conditioning a learned policy on a desired high-return future outcome will correctly induce high-return behavior. Empirically, we show that DoC is able to achieve significantly better performance than DT on environments that have highly stochastic rewards and transitions.

1 INTRODUCTION

Offline reinforcement learning (RL) aims to extract an optimal policy solely from an existing dataset of previous interactions (Fujimoto et al., 2019; Wu et al., 2019; Kumar et al., 2020). As researchers begin to scale offline RL to large image, text, and video datasets (Agarwal et al., 2020; Fan et al., 2022; Baker et al., 2022; Reed et al., 2022; Reid et al., 2022), a family of methods known as *return-conditioned supervised learning* (RCSL), including Decision Transformer (DT) (Chen et al., 2021; Lee et al., 2022) and RL via Supervised Learning (RvS) (Emmons et al., 2021), have gained popularity due to their algorithmic simplicity and ease of scaling. At the heart of RCSL is the idea of conditioning a policy on a specific future outcome, often a return (Srivastava et al., 2019; Kumar et al., 2019; Chen et al., 2021) but also sometimes a goal state or generic future event (Codevilla et al., 2018; Ghosh et al., 2019; Lynch et al., 2020). RCSL trains a policy to imitate actions associated with a conditioning input via supervised learning. During inference (i.e., at evaluation), the policy is conditioned on a desirable high-return or future outcome, with the hope of inducing behavior that can achieve this desirable outcome.

Despite the empirical advantages that come with supervised training (Emmons et al., 2021; Kumar et al., 2021), RCSL can be highly suboptimal in stochastic environments (Brandfonbrener et al., 2022), where the future an RCSL policy conditions on (e.g., return) can be primarily determined by randomness in the environment rather than the data collecting policy itself. Figure 1 (left) illustrates an example, where conditioning an RCSL policy on the highest return observed in the dataset ($r = 100$) leads to a policy (a_1) that relies on a stochastic transition of very low probability ($T = 0.01$) to achieve the desired return of $r = 100$; by comparison the choice of a_2 is much better in terms of average return, as it surely achieves $r = 10$. The crux of the issue is that the RCSL policy is *inconsistent* with its conditioning input. Conditioning the policy on a desired return (i.e., 100) to

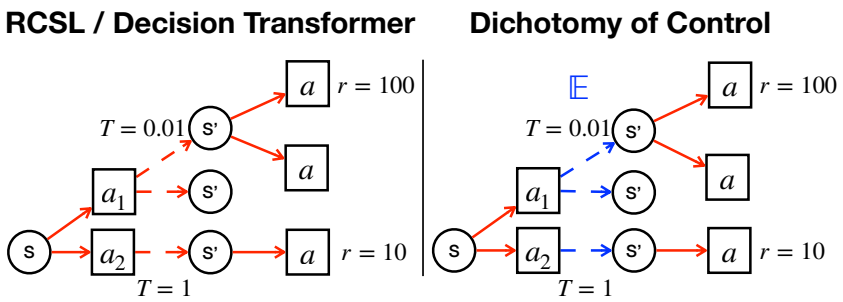

Figure 1: Illustration of DT (RCSL) and DoC. Circles and squares denote states and actions. Solid arrows denote policy decisions. Dotted arrows denote (stochastic) environment transitions. All arrows and nodes are present in the dataset, i.e., there are 4 trajectories, 2 of which achieve 0 reward. DT maximizes returns across an entire trajectory, leading to suboptimal policies when a large return ($r = 100$) is achieved only due to very low-probability environment transitions ($T = 0.01$). DoC separates policy stochasticity from that of the environment and only tries to control action decisions (solid arrows), achieving optimal control through maximizing expected returns at each timestep.

act in the environment leads to a distribution of real returns (i.e., $0.01 * 100$) that is wildly different from the return value being conditioned on. This issue would not have occurred if the policy could also maximize the transition probability that led to the high-return state, but this is not possible as transition probabilities are a part of the environment and not subject to the policy's control.

A number of works propose a generalization of RCSL, known as future-conditioned supervised learning methods. These techniques have been shown to be effective in imitation learning (Singh et al., 2020; Pertsch et al., 2020), offline Q-learning (Ajay et al., 2020), and online policy gradient (Venuto et al., 2021). It is common in future-conditioned supervised learning to apply a KL divergence regularizer on the latent variable – measured with respect to a learned prior conditioned only on past information – to limit the amount of future information captured in the latent variable. It is natural to ask whether this regularizer could remedy the insconsistency of RCSL. Unfortunately, as the KL regularizer makes no distinction between future information that is controllable versus that which is not, such an approach will still exhibit inconsistency, in the sense that the latent variable representation may contain information about the future that is due only to environment stochasticity.

It is clear that the major issue with both RCSL and naïve variational methods is that they make no distinction between stochasticity of the policy (controllable) and stochasticity of the environment (uncontrollable). An optimal policy should maximize over the controllable (actions) and take expectations over uncontrollable (e.g., transitions) as shown in Figure 1 (right). This implies that, under a variational approach, the latent variable representation that a policy conditions on should not incorporate any information that is solely due to randomness in the environment. In other words, while the latent representation can and should include information about future behavior (i.e., actions), it should not reveal any information about the rewards or transitions associated with this behavior.

To this end, we propose a future-conditioned supervised learning framework termed *dichotomy of control* (DoC), which, in Stoic terms (Shapiro, 2014), has *"the serenity to accept the things it cannot change, courage to change the things it can, and wisdom to know the difference."* DoC separates mechanisms within a policy's control (actions) from those beyond a policy's control (environment stochasticity). To achieve this separation, we condition the policy on a latent variable representation of the future while minimizing the mutual information between the latent variable and future stochastic rewards and transitions in the environment. By only capturing the controllable factors in the latent variable, DoC can maximize over each action step without also attempting to maximize environment transitions as shown in Figure 1 (right). Theoretically, we show that DoC policies are *consistent* with their conditioning inputs, ensuring that conditioning on a high-return future will correctly induce high-return behavior. Empirically, we show that DoC can outperform both RCSL and naïve variational methods on highly stochastic environments.

2 Related Work

Return-Conditioned Supervised Learning. Since offline RL algorithms (Fujimoto et al., 2019; Wu et al., 2019; Kumar et al., 2020) can be sensitive to hyper-parameters and difficult to apply in practice (Emmons et al., 2021; Kumar et al., 2021), return-conditioned supervised learning (RCSL) has become a popular alternative, particularly when the environment is deterministic and near-expert demonstrations are available (Brandfonbrener et al., 2022). RCSL learns to predict behaviors (actions) by conditioning on desired returns (Schmidhuber, 2019; Kumar et al., 2019) using an MLP policy (Emmons et al., 2021) or a transformer-based policy that encapsulates history (Chen et al., 2021). Richer information other than returns, such as goals (Codevilla et al., 2018; Ghosh et al., 2019) or trajectory-level aggregates (Furuta et al., 2021), have also been used as inputs to a conditional policy in practice. Our work also conditions policies on richer trajectory-level information in the form of a latent variable representation of the future, with additional theoretical justifications of such conditioning in stochastic environments.

RCSL Failures in Stochastic Environments. Despite the empirical success of RCSL achieved by DT and RvS, recent work has noted the failure modes in stochastic environments. Paster et al. (2020) and Štrupl et al. (2022) presented counter-examples where online RvS can diverge in stochastic environments. Brandfonbrener et al. (2022) identified near-determinism as a necessary condition for RCSL to achieve optimality guarantees similar to other offline RL algorithms but did not propose a solution for RCSL in stochastic settings. Paster et al. (2022) identified this same issue with stochastic transitions and proposed to cluster offline trajectories and condition the policy on the average cluster returns. However, the approach in Paster et al. (2022) has technical limitations (see Appendix C), does not account for reward stochasticity, and still conditions the policy on (expected) returns, which can lead to undesirable policy-averaging, i.e., a single policy covering two very different behaviors (clusters) that happen to have the same return.

Learning Latent Variables from Offline Data. Various works have explored learning a latent variable representation of the future (or past) transitions in offline data via maximum likelihood and use the latent variable to assist planning (Lynch et al., 2020), imitation learning (Kipf et al., 2019; Ajay et al., 2020; Hakhamaneshi et al., 2021), offline RL (Ajay et al., 2020; Zhou et al., 2020), or online RL (Fox et al., 2017; Krishnan et al., 2017; Goyal et al., 2019; Shankar & Gupta, 2020; Singh et al., 2020; Wang et al., 2021; Venuto et al., 2021). These works generally focus on the benefit of increased temporal abstraction afforded by using latent variables as higher-level actions in a hierarchical policy. However, latent variable models have not been fully incorporated into RCSL, which is one of the essential tools that enables our method. As we will see, latent variable models with appropriate constraints can allow RCSL to effectively combat environment stochasticity.

3 Preliminaries

Environment Notation We consider the problem of learning a decision-making agent to interact with a sequential, finite-horizon environment. At time $t = 0$, the agent observes an initial state s_0 determined by the environment. After observing s_t at a timestep $0 \leq t \leq H$, the agent chooses an action a_t. After the action is applied the environment yields an immediate scalar reward r_t and, if $t < H$, a next state s_{t+1}. We use $\tau := (s_t, a_t, r_t)_{t=0}^{H}$ to denote a generic *episode* generated from interactions with the environment, and use $\tau_{i:j} := (s_t, a_t, r_t)_{t=i}^{j}$ to denote a generic *sub-episode*, with the understanding that $\tau_{0:-1}$ refers to an empty sub-episode. The *return* associated with an episode τ is defined as $R(\tau) := \sum_{t=0}^{H} r_t$.

We will use $\mathcal{M}$ to denote the environment. We assume that $\mathcal{M}$ is determined by a stochastic reward function $\mathcal{R}$, stochastic transition function $\mathcal{T}$, and unique initial state s_0, so that $r_t \sim \mathcal{R}(\tau_{0:t-1}, s_t, a_t)$ and $s_{t+1} \sim \mathcal{T}(\tau_{0:t-1}, s_t, a_t)$ during interactions with the environment. Note that these definitions specify a history-dependent environment, as opposed to a less general Markovian environment.

Learning a Policy in RCSL In future- or return-conditioned supervised learning, one uses a fixed training data distribution $\mathcal{D}$ of episodes τ (collected by unknown and potentially multiple agents) to learn a policy π, where π is trained to predict a_t conditioned on the history $\tau_{0:t-1}$, the observation s_t, and an additional conditioning variable z that may depend on both the past and future of the episode. For example, in return-conditioned supervised learning, policy training minimizes the following

objective over π:

$$\mathcal{L}_{\text{RCSL}}(\pi) := \mathbb{E}_{\tau\sim\mathcal{D}}\left[\sum_{t=0}^{H} -\log \pi(a_t|\tau_{0:t-1}, s_t, z(\tau))\right], \tag{1}$$

where $z(\tau)$ is the return $R(\tau)$.

Inconsistency of RCSL To apply an RCSL-trained policy π during inference — i.e., interacting online with the environment — one must first choose a specific z.[1] For example, one might set z to be the maximal return observed in the dataset, in the hopes of inducing a behavior policy which achieves this high return. Using π_z as a shorthand to denote the policy π conditioned on a specific z, we define the expected return $V_{\mathcal{M}}(\pi_z)$ of π_z in $\mathcal{M}$ as,

$$V_{\mathcal{M}}(\pi_z) := \mathbb{E}_{\tau\sim\Pr[\cdot|\pi_z,\mathcal{M}]}\left[R(\tau)\right]. \tag{2}$$

Ideally the expected return induced by π_z is close to z, i.e., $z \approx V_{\mathcal{M}}(\pi_z)$, so that acting according to π conditioned on a high return induces behavior which actually achieves a high return. However, RCSL training according to Equation 1 will generally yield policies that are highly *inconsistent* in stochastic environments, meaning that the achieved returns may be significantly different than z (i.e., $V_{\mathcal{M}}(\pi_z) \neq z$). This has been highlighted in various previous works (Brandfonbrener et al., 2022; Paster et al., 2022; Štrupl et al., 2022; Eysenbach et al., 2022), and we provided our own example in Figure 1.

Approaches to Mitigating Inconsistency A number of future-conditioned supervised learning approaches propose to learn a stochastic latent variable embedding of the future, $q(z|\tau)$, while regularizing q with a KL-divergence from a learnable *prior* conditioned only on the past $p(z|s_0)$ (Ajay et al., 2020; Venuto et al., 2021; Lynch et al., 2020), thereby minimizing:

$$\mathcal{L}_{\text{VAE}}(\pi, q, p) := \mathbb{E}_{\tau\sim\mathcal{D}, z\sim q(z|\tau)}\left[\sum_{t=0}^{H} -\log \pi(a_t|\tau_{0:t-1}, s_t, z)\right] + \beta \cdot \mathbb{E}_{\tau\sim\mathcal{D}}\left[D_{\text{KL}}(q(z|\tau)\|p(z|s_0))\right]. \tag{3}$$

One could consider adopting such a future-conditioned objective in RCSL. However, since the KL regularizer makes no distinction between observations the agent can control (actions) from those it cannot (environment stochasticity), the choice of coefficient β applied to the regularizer introduces a 'lose-lose' trade-off. Namely, as noted in Ajay et al. (2020), if the regularization coefficient is too large ($\beta \geq 1$), the policy will not learn diverse behavior (since the KL limits how much information of the future actions is contained in z); while if the coefficient is too small ($\beta < 1$), the policy's learned behavior will be inconsistent (in the sense that z will contain information of environment stochasticity that the policy cannot reliably reproduce).

4 Dichotomy of Control

In this section, we first propose the DoC objective for learning future-conditioned policies that are guaranteed to be consistent. We then present a practical framework for optimizing DoC's constrained objective in practice and an inference scheme to enable better-than-dataset behavior via a learned value function and prior.

4.1 Dichotomy of Control via Mutual Information Minimization

As elaborated in the prevous section, whether $z(\tau)$ is the return $R(\tau)$ or more generally a stochastic latent variable with distribution $q(z|\tau)$, existing RCSL methods fail to satisfy consistency because they insufficiently enforce the type of future information z can contain. Our key observation is that z should not include any information due to environment stochasticity, i.e., any information about a future r_t, s_{t+1} that is not already known given the previous history up to that point $\tau_{0:t-1}, s_t, a_t$.

[1] For simplicitly, we assume z is chosen at timestep $t = 0$ and held constant throughout an entire episode. As noted in Brandfonbrener et al. (2022), this protocol also encompasses instances like DT (Chen et al., 2021) in which z at timestep t is the (desired) return summed starting at t.

Accordingly, we modify the RCSL objective from Equation 1 to incorporate a conditional mutual information constraint between z and each pair r_t, s_{t+1} in the future:

$$\mathcal{L}_{\text{DoC}}(\pi, q) := \mathbb{E}_{\tau\sim\mathcal{D}, z\sim q(z|\tau)}\left[\sum_{t=0}^{H} -\log \pi(a_t|\tau_{0:t-1}, s_t, z)\right]$$

$$\text{s.t.} \quad \text{MI}(r_t; z \mid \tau_{0:t-1}, s_t, a_t) = 0, \text{MI}(s_{t+1}; z \mid \tau_{0:t-1}, s_t, a_t) = 0, \tag{4}$$

$$\forall\, \tau_{0:t-1}, s_t, a_t \text{ and } 0 \leq t \leq H, \tag{5}$$

where $\text{MI}(r_t; z|\tau_{0:t-1}, s_t, a_t)$ denotes the mutual information between r_t and z given $\tau_{0:t-1}, s_t, a_t$ when measured under samples of r_t, z from $\mathcal{D}, q$; and analogously for $\text{MI}(s_{t+1}; z|\tau_{0:t-1}, s_t, a_t)$.

The first part of the DoC objective conditions the policy on a latent variable representation of the future, similar to the first part of the future-conditioned VAE objective in Equation 3. However, unlike Equation 3, the DoC objective enforces a much more precise constraint on q, given by the MI constraints in Equation 4.

4.2 Dichotomy of Control in Practice

Contrastive Learning of DoC Constraints. To satisfy the mutual information constraints in Equation 4 we transform the MI to a contrastive learning objective. Specifically, for the constraint on r and z (and similarly on s_{t+1} and z) one can derive,

$$\begin{aligned}
&\text{MI}(r_t; z|\tau_{0:t-1}, s_t, a_t) \\
&\quad = D_{\text{KL}}\left(\Pr[r_t, z|\tau_{0:t-1}, s_t, a_t] \| \Pr[r_t|\tau_{0:t-1}, s_t, a_t]\Pr[z|\tau_{0:t-1}, s_t, a_t]\right) \\
&\quad = \mathbb{E}_{\Pr[r_t, z|\tau_{0:t-1}, s_t, a_t]}\left[\log\left(\frac{\Pr[r_t|z, \tau_{0:t-1}, s_t, a_t]}{\Pr[r_t|\tau_{0:t-1}, s_t, a_t]}\right)\right] \\
&\quad = \mathbb{E}_{\Pr[r_t, z|\tau_{0:t-1}, s_t, a_t]} \log \Pr[r_t|z, \tau_{0:t-1}, s_t, a_t] - \mathbb{E}_{\Pr[r_t|\tau_{0:t-1}, s_t, a_t]} \log \Pr[r_t|\tau_{0:t-1}, s_t, a_t].
\end{aligned} \tag{6}$$

The second expectation above is a constant with respect to z and so can be ignored during learning. We further introduce a conditional distribution $\omega(r_t|\tau_{0:t-1}, s_t, a_t)$ parametrized by an energy-based function $f : \Omega \mapsto \mathbb{R}$:

$$\omega(r_t|z, \tau_{0:t-1}, s_t, a_t) \propto \rho(r_t) \exp\{f(r_t, z, \tau_{0:t-1}, s_t, a_t)\}, \tag{7}$$

where ρ is some fixed sampling distribution of rewards. In practice, we set ρ to be the marginal distribution of rewards in the dataset. Hence we express the first term of Equation 6 via an optimization over ω, i.e.,

$$\begin{aligned}
&\max_{\omega}\ \mathbb{E}_{\Pr[r_t, z|\tau_{0:t-1}, s_t, a_t]}\left[\log \omega(r_t|\tau_{0:t-1}, s_t, a_t)\right] \\
&= \max_{f} \mathbb{E}_{\Pr[r_t, z|\tau_{0:t-1}, s_t, a_t]}\left[f(r_t, z, \tau_{0:t-1}, s_t, a_t) - \log \mathbb{E}_{\rho(\tilde{r})}\left[\exp\{f(\tilde{r}, z, \tau_{0:t-1}, s_t, a_t)\}\right]\right].
\end{aligned}$$

Combining this (together with the analogous derivation for $\text{MI}(s_{t+1}; z|\tau_{0:t-1}, s_t, a_t)$) with Equation 4 via the Lagrangian, we can learn π and $q(z|\tau)$ by minimizing the final DoC objective:

$$\begin{aligned}
\mathcal{L}_{\text{DoC}}(\pi, q) &= \max_{f} \mathbb{E}_{\tau\sim\mathcal{D}, z\sim q(z|\tau)}\left[\sum_{t=0}^{H} -\log \pi(a_t|\tau_{0:t-1}, s_t, z)\right] \\
&+ \beta \cdot \sum_{t=0}^{H} \mathbb{E}_{\tau\sim\mathcal{D}, z\sim q(z|\tau)}\left[f(r_t, s_{t+1}, z, \tau_{0:t-1}, s_t, a_t) - \log \mathbb{E}_{\rho(\tilde{r}, \tilde{s}')}\left[\exp\{f(\tilde{r}, \tilde{s}', z, \tau_{0:t-1}, s_t, a_t)\}\right]\right].
\end{aligned} \tag{8}$$

Algorithm 1 Inference with Dichotomy of Control

Inputs Policy $\pi(\cdot|\cdot,\cdot,\cdot)$, prior $p(\cdot)$, value function $V(\cdot)$, initial state s_0, number of samples hyperparameter K.
Initialize $z^*; V^*$ ▷ `Track the best latent and its value.`
for $k = 1$ to K **do**
  Sample $z_k \sim p(z|s_0)$ ▷ `Sample a latent from the learned prior.`
  if $V(z_k) > V^*$ **then**
    $z^* = z_k; V^* = V$ ▷ `Set best latent to the one with the highest value.`
return $\pi(\cdot|\cdot,\cdot,z^*)$ ▷ `Policy conditioned on the best` z^*.

DoC Inference. As is standard in RCSL approaches, the policy learned by DoC requires an appropriate conditioning input z to be chosen during inference. To choose a desirable z associated with high return, we propose to (1) enumerate or sample a large number of potential values of z, (2) estimate the expected return for each of these values of z, (3) choose the z with the highest associated expected return to feed into the policy. To enable such an inference-time procedure, we need to add two more components to the method formulation: First, a prior distribution $p(z|s_0)$ from which we will sample a large number of values of z; second, a value function $V(z)$ with which we will rank the potential values of z. These components are learned by minimizing the following objective:

$$\mathcal{L}_{\text{aux}}(V, p) = \mathbb{E}_{\tau\sim\mathcal{D}, z\sim q(z|\tau)}\Big[\left(V(z) - R(\tau)\right)^2 + D_{\text{KL}}(\text{stopgrad}(q(z|\tau))\|p(z|s_0))\Big]. \tag{9}$$

Note that we apply a stop-gradient to $q(z|\tau)$ when learning p so as to avoid regularizing q with the prior. This is unlike the VAE approach, which by contrast advocates *for* regularizing q via the prior. See Algorithm 1 for inference pseudocode (and Appendix D for training pseudocode).

5 Consistency Guarantees for Dichotomy of Control

We provide a theoretical justification of the proposed learning objectives $\mathcal{L}_{\text{DoC}}$ and $\mathcal{L}_{\text{aux}}$, showing that, if they are minimized, the resulting inference-time procedure will be sound, in the sense that DoC will learn a V and π such that the true value of π_z in the environment $\mathcal{M}$ is equal to $V(z)$. More specifically we define the following notion of *consistency*:

Definition 1 (Consistency)**.** *A future-conditioned policy π and value function V are **consistent** for a specific conditioning input z if the expected return of z predicted by V is equal to the true expected return of π_z in the environment: $V(z) = V_{\mathcal{M}}(\pi_z)$.*

To guarantee consistency of π, V, we will make the following two assumptions:

Assumption 2 (Data and environment agreement)**.** *The per-step reward and next-state transitions observed in the data distribution are the same as those of the environment. In other words, for any $\tau_{0:t-1}, s_t, a_t$ with $\Pr[\tau_{0:t-1}, s_t, a_t|\mathcal{D}] > 0$, we have $\Pr[\hat{r}_t = r_t|\tau_{0:t-1}, s_t, a_t, \mathcal{D}] = \mathcal{R}(\hat{r}_t|\tau_{0:t-1}, s_t, a_t)$ and $\Pr[\hat{s}_{t+1} = s_{t+1}|\tau_{0:t-1}, s_t, a_t, \mathcal{D}] = \mathcal{T}(\hat{s}_{t+1}|\tau_{0:t-1}, s_t, a_t)$ for all $\hat{r}_t, \hat{s}_{t+1}$.*

Assumption 3 (No optimization or approximation errors)**.** *DoC yields policy π and value function V that are Bayes-optimal with respect to the training data distribution and q. In other words, $V(z) = \mathbb{E}_{\tau\sim\Pr[\cdot|z,\mathcal{D}]}\left[R(\tau)\right]$ and $\pi(\hat{a}|\tau_{0:t-1}, s_t, z) = \Pr\left[\hat{a} = a_t|\tau_{0:t-1}, s_t, z, \mathcal{D}\right]$.*

Given these two assumptions, we can then establish the following consistency guarantee for DoC.

Theorem 4. *Suppose DoC yields π, V, q with q satisfying the MI constraints:*

$$\text{MI}(r_t; z|\tau_{0:t-1}, s_t, a_t) = \text{MI}(s_{t+1}; z|\tau_{0:t-1}, s_t, a_t) = 0, \tag{10}$$

for all $\tau_{0:t-1}, s_t, a_t$ with $\Pr[\tau_{0:t-1}, s_t, a_t|\mathcal{D}] > 0$. Then under Assumptions 2 and 3, V and π are consistent for any z with $\Pr[z|q, \mathcal{D}] > 0$.

For proof, see Appendix A.

Consistency in Markovian environments. While the results above are focused on environments and policies that are non-Markovian, one can extend Theorem 4 to Markovian environments and policies. This result is somewhat surprising, as the assignments of z to episodes τ induced by q are necessarily history-dependent, and projecting the actions appearing in these clusters to a non-history-dependent policy would seemingly lose important information. However, a Markovian assumption on the rewards and transitions of the environment is sufficient to ensure that no 'important' information will be lost, at least in terms of the satisfying requirements for consistency in Definition 1. Alternative notions of consistency are not as generally applicable; see Appendix C.

We begin by stating our assumptions.

Assumption 5 (Markov environment)**.** *The rewards and transitions of $\mathcal{M}$ are Markovian; i.e., $\mathcal{R}(\tau_{0:t-1}, s_t, a_t) = \mathcal{R}(\tilde{\tau}_{0:t-1}, s_t, a_t)$ and $\mathcal{T}(\tau_{0:t-1}, s_t, a_t) = \mathcal{T}(\tilde{\tau}_{0:t-1}, s_t, a_t)$ for all $\tau, \tilde{\tau}, s_t, a_t$. We use the shorthand $\mathcal{R}(s_t, a_t), \mathcal{T}(s_t, a_t)$ for these history-independent functions.*

Assumption 6 (Markov policy, without optimization or approximation errors)**.** *The policy learned by DoC is Markov. This policy π as well as its corresponding learned value function V are Bayes-optimal with respect to the training data distribution and q. In other words, $V(z) = \mathbb{E}_{\tau \sim \Pr[\cdot|z,\mathcal{D}]}[R(\tau)]$ and $\pi(\hat{a}|s_t, z) = \Pr[\hat{a} = a_t|s_t, z, \mathcal{D}]$.*

With these two assumptions, we can then establish the analogue to Theorem 4, which relaxes the dependency on history for both the policy π and the MI constraints:

Theorem 7. *Suppose DoC yields π, V, q with q satisfying the MI constraints:*

$$\mathrm{MI}(r_t; z|s_t, a_t) = \mathrm{MI}(s_{t+1}; z|s_t, a_t) = 0, \tag{11}$$

for all s_t, a_t with $\Pr[s_t, a_t|\mathcal{D}] > 0$. Then under Assumptions 2, 5, and 6, V and π are consistent for any z with $\Pr[z|q, \mathcal{D}] > 0$.

For proof, see Appendix B.

6 Experiments

We conducted an empirical evaluation to ascertain the effectiveness of DoC. For this evaluation, we considered three settings: (1) a Bernoulli bandit problem with stochastic rewards, based on a canonical 'worst-case scenario' for RCSL (Brandfonbrener et al., 2022); (2) the FrozenLake domain from (Brockman et al., 2016), where the future VAE approach proves ineffective; and finally (3) a modified set of OpenAI Gym (Brockman et al., 2016) environments where we introduced environment stochasticity. In these studies, we found that DoC exhibits a significant advantage over RCSL/DT, and outperforms future VAE when the analogous to "one-step" RL is insufficient. For DT, we use the same implementation and hyperparameters as Chen et al. (2021). Both VAE and DoC are built upon the DT implementation and additionally learn a Gaussian latent variable over succeeding 20 future steps. See experiment details in Appendix E and additional results in Appendix F.

6.1 Evaluating Stochastic Rewards in Bernoulli Bandit

Bernoulli Bandit. Consider a two-armed bandit as shown in Figure 2 (left). The two arms, a_1, a_2, have stochastic rewards drawn from Bernoulli distributions of $\texttt{Bern}(1-p)$ and $\texttt{Bern}(p)$, respectively. In the offline dataset, the a_1 arm with reward $\texttt{Bern}(1-p)$ is pulled with probability $\pi_D(a_1) = p$. When p is small, this corresponds to the better arm only being pulled occasionally. Under this setup, $\pi_{\text{RCSL}}(a_1|r=1) = \pi_{\text{RCSL}}(a_2|r=1) = 0.5$, which is highly suboptimal compared to always pulling the optimal arm a_1 with reward $\texttt{Bern}(1-p)$ for $p < 0.5$.

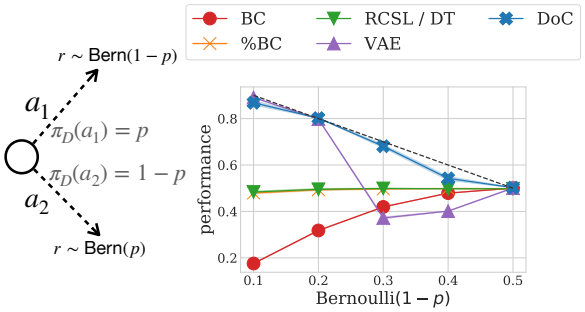

Figure 2: [Left] Bernoulli bandit where the better arm a_1 with reward $\texttt{Bern}(1-p)$ for $p < 0.5$ is pulled with probability $\pi_D(a_1) = p$ in the offline data. [Right] Average rewards achieved by DoC and baselines across 5 environment seeds. RCSL is highly suboptimal when p is small, whereas DoC achieves close to Bayes-optimal performance (dotted line) for all values of p.

Results. We train tabular DoC and baselines on 1000 samples where the superior arm with $r \sim \texttt{Bern}(1-p)$ is pulled with probability p for $p \in \{0.1, ..., 0.5\}$. Figure 2 (right) shows that RCSL and percentage BC (filtered by $r = 1$) always result in policies that are indifferent in the arms, whereas DoC is able to recover the Bayes-optimal performance (dotted line) for all p values considered. Future VAE performs similarly to DoC for small p values, but is sensitive to the KL regularization coefficient when p is close to 0.5.

6.2 Evaluating Stochastic Transitions in FrozenLake

FrozenLake. Next, we consider the FrozenLake environment with stochastic transitions where the agent taking an action has probability p of moving in the intended direction, and probability

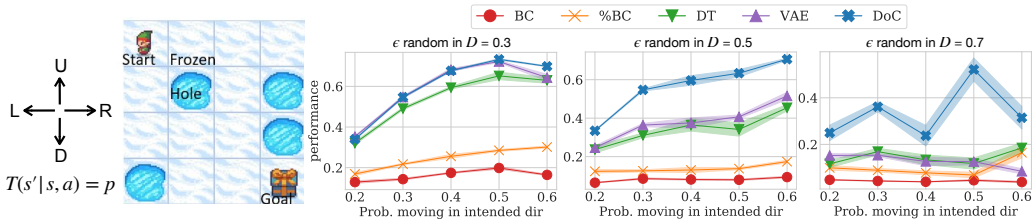

Figure 3: [Left] Visualization of the stochastic FrozenLake task. The agent has a probability p of moving in the intended direction and $1-p$ of slipping to either sides. [Right] Average performance (across 5 seeds) of DoC and baselines on FrozenLake with different levels of stochasticity (p) and offline dataset quality (ϵ). DoC outperforms DT and future VAE, where the gain is more salient when the offline data is less optimal ($\epsilon = 0.5$ and $\epsilon = 0.7$).

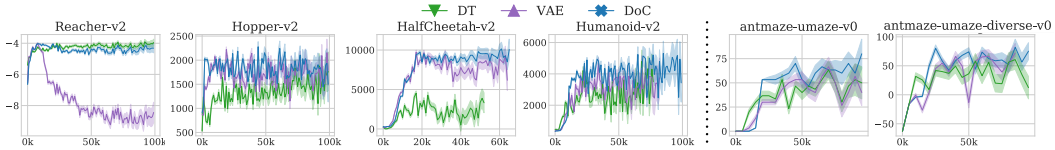

Figure 4: Average performance (across 5 seeds) of DoC and baselines on modified stochastic Gym MuJoCo and AntMaze tasks. DoC and future VAE generally provide benefits over DT, where DoC provide more benefits on harder tasks such as Humanoid. Future VAE can be sensitive to the KL coefficient β, which can result in the failure mode shown in Reacher-v2 if not tuned properly.

$0.5 \cdot (1-p)$ of slipping to either of the two sides of the intended direction. We collect 100 trajectories of length 100 using a DQN policy trained in the original environment ($p = \frac{1}{3}$) which achieves an average return of 0.7, and vary p during data collection and evaluation to test different levels of stochasticity. We also include uniform actions with probability ϵ to lower the performance of the offline data so that BC is highly suboptimal.

Results. Figure 3 presents the visualization (left) and results (right) for this task. When the offline data is closer to being expert ($\epsilon = 0.3$), DT, future VAE, and DoC perform similarly with better performance in more deterministic environments. As the offline dataset becomes more suboptimal ($\epsilon = 0.5$), DoC starts to dominate across all levels of transition stochasticity. When the offline data is highly suboptimal ($\epsilon = 0.7$), DT and future VAE has little advantage over BC, whereas DoC continues to learn policies with reasonable performance.

6.3 Evaluating Stochastic Gym MuJoCo

Environments. We now consider a set of Gym MuJoCo environments including Reacher, Hopper, HalfCheetah, and Humanoid. We additionally consider AntMaze from D4RL (Fu et al., 2020). These environments are deterministic by default, which we modify by introducing time-correlated Gaussian noise to the actions before inputing the action into the physics simulator during data collection and evaluation for all but AntMaze environments. Specifically, the Gaussian noise we introduce to the actions has 0 mean and standard deviation of the form $(1 - e^{-0.01 \cdot t}) \cdot \sin(t) \cdot \sigma$ where t is the step number and $\sigma \in [0, 1]$. For AntMaze where the dataset has already been collected in the deterministic environment by D4RL, we add gaussian noise with 0.1 standard deviation to the reward uniformly with probability 0.1 (both to the dataset and during evaluation).

Results. Figure 4 shows the average performance (across 5 seeds) of DT, future VAE, and DoC on these stochastic environments. Both future VAE and DoC generally provide benefits over DT, where the benefit of DoC is more salient in harder environments such as HalfCheetah and Humanoid. We found future VAE to be sensitive to the β hyperparameter, and simply using $\beta = 1$ can result in the falure case as shown in Reacher-v2.

7 CONCLUSION

Despite the empirical promise of return- or future-conditioned supervised learning (RCSL) with large transformer architectures, environment stochasticity hampers the application of supervised learning to sequential decision making. To address this issue, we proposed to augment supervised learning with the dichotomy of control principle (DoC), guiding a supervised policy to only control the controllable (actions). Theoretically, DoC learns *consistent* policies, guaranteeing that they achieve the future or return they are conditioned on. Empirically, DoC outperforms RCSL in highly stochastic environments. While DoC still falls short in addressing other RL challenges such as 'stitching' (i.e., composing sub-optimal trajectories), we hope that dichotomy of control serves as a stepping stone in solving sequential decision making with large-scale supervised learning.

REFERENCES

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pp. 104–114. PMLR, 2020.

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. Opal: Offline primitive discovery for accelerating offline reinforcement learning. *arXiv preprint arXiv:2010.13611*, 2020.

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *arXiv preprint arXiv:2206.11795*, 2022.

David Brandfonbrener, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. When does return-conditioned supervised learning work for offline reinforcement learning? *arXiv preprint arXiv:2206.01079*, 2022.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

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.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 4693–4700. IEEE, 2018.

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.

Benjamin Eysenbach, Soumith Udatha, Sergey Levine, and Ruslan Salakhutdinov. Imitating past successes can be very suboptimal. *arXiv preprint arXiv:2206.03378*, 2022.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *arXiv preprint arXiv:2206.08853*, 2022.

Roy Fox, Sanjay Krishnan, Ion Stoica, and Ken Goldberg. Multi-level discovery of deep options. *arXiv preprint arXiv:1703.08294*, 2017.

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.

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

Hiroki Furuta, Yutaka Matsuo, and Shixiang Shane Gu. Generalized decision transformer for offline hindsight information matching. *arXiv preprint arXiv:2111.10364*, 2021.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. *arXiv preprint arXiv:1912.06088*, 2019.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Kourosh Hakhamaneshi, Ruihan Zhao, Albert Zhan, Pieter Abbeel, and Michael Laskin. Hierarchical few-shot imitation with skill transition models. *arXiv preprint arXiv:2107.08981*, 2021.

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. Compile: Compositional imitation learning and execution. In *International Conference on Machine Learning*, pp. 3418–3428. PMLR, 2019.

Sanjay Krishnan, Roy Fox, Ion Stoica, and Ken Goldberg. Ddco: Discovery of deep continuous options for robot learning from demonstrations. In *Conference on robot learning*, pp. 418–437. PMLR, 2017.

Aviral Kumar, Xue Bin Peng, and Sergey Levine. Reward-conditioned policies. *arXiv preprint arXiv:1912.13465*, 2019.

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.

Aviral Kumar, Joey Hong, Anikait Singh, and Sergey Levine. Should i run offline reinforcement learning or behavioral cloning? In *International Conference on Learning Representations*, 2021.

Kuang-Huei Lee, Ofir Nachum, Mengjiao Yang, Lisa Lee, Daniel Freeman, Winnie Xu, Sergio Guadarrama, Ian Fischer, Eric Jang, Henryk Michalewski, et al. Multi-game decision transformers. *arXiv preprint arXiv:2205.15241*, 2022.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on Robot Learning*, pp. 1113–1132. PMLR, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Keiran Paster, Sheila A McIlraith, and Jimmy Ba. Planning from pixels using inverse dynamics models. *arXiv preprint arXiv:2012.02419*, 2020.

Keiran Paster, Sheila McIlraith, and Jimmy Ba. You can't count on luck: Why decision transformers fail in stochastic environments. *arXiv preprint arXiv:2205.15967*, 2022.

Karl Pertsch, Youngwoon Lee, and Joseph J Lim. Accelerating reinforcement learning with learned skill priors. *arXiv preprint arXiv:2010.11944*, 2020.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

Machel Reid, Yutaro Yamada, and Shixiang Shane Gu. Can wikipedia help offline reinforcement learning? *arXiv preprint arXiv:2201.12122*, 2022.

Juergen Schmidhuber. Reinforcement learning upside down: Don't predict rewards–just map them to actions. *arXiv preprint arXiv:1912.02875*, 2019.

Tanmay Shankar and Abhinav Gupta. Learning robot skills with temporal variational inference. In *International Conference on Machine Learning*, pp. 8624–8633. PMLR, 2020.

Fred R Shapiro. Who wrote the serenity prayer? *The Chronicle Review*, 28, 2014.

Avi Singh, Huihan Liu, Gaoyue Zhou, Albert Yu, Nicholas Rhinehart, and Sergey Levine. Parrot: Data-driven behavioral priors for reinforcement learning. *arXiv preprint arXiv:2011.10024*, 2020.

Rupesh Kumar Srivastava, Pranav Shyam, Filipe Mutz, Wojciech Jaśkowski, and Jürgen Schmidhuber. Training agents using upside-down reinforcement learning. *arXiv preprint arXiv:1912.02877*, 2019.

Miroslav Štrupl, Francesco Faccio, Dylan R Ashley, Jürgen Schmidhuber, and Rupesh Kumar Srivastava. Upside-down reinforcement learning can diverge in stochastic environments with episodic resets. *arXiv preprint arXiv:2205.06595*, 2022.

David Venuto, Elaine Lau, Doina Precup, and Ofir Nachum. Policy gradients incorporating the future. *arXiv preprint arXiv:2108.02096*, 2021.

Xiaofei Wang, Kimin Lee, Kourosh Hakhamaneshi, Pieter Abbeel, and Michael Laskin. Skill preferences: Learning to extract and execute robotic skills from human feedback. *arXiv preprint arXiv:2108.05382*, 2021.

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

Wenxuan Zhou, Sujay Bajracharya, and David Held. Plas: Latent action space for offline reinforcement learning. *arXiv preprint arXiv:2011.07213*, 2020.