Variational Consequence-Driven Offline Reinforcement Learning

Wen Jiang[1,2], **Ke Jiang**[1,2,3], **Jin Wang**[1,2], **Xiaoyang Tan**[1,2]
[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
[2]MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
[3]Graduate School of Information Science and Technology, The University of Osaka
{darren.jum, ke_jiang, jinw1999, x.tan}@nuaa.edu.cn

Abstract

Offline reinforcement learning (RL) typically mitigates *distribution shift* by imposing divergence constraints on the action distributions of the learned and behavior policies. However, this paradigm cannot ensure the underlying dynamic consistency structure: even minor action deviations can lead to drastically different transitions, inadvertently driving the agent away from the offline dataset. To address this, we explore policy constraints through their induced state-consequence distributions rather than relying on pure action-distribution matching. Nevertheless, directly enforcing such consequence-aware constraints is computationally intractable without access to the true underlying dynamics model. To bypass this issue, we propose Structural Behavior Regularization (StBR), which replaces the intractable objective with a closed-form latent divergence. Structuring this latent space to reflect transition dynamics renders the latent divergence a theoretically bounded surrogate for the true consequence-distribution divergence. By regularizing the learned policy with this latent divergence, StBR adaptively tightens the behavior constraint in dynamics-sensitive regions, effectively preventing deviations outside the offline dataset. Empirically, we demonstrate that StBR achieves superior performance across diverse offline RL benchmarks.

1 Introduction

Offline Reinforcement Learning (RL) aims to learn effective policies entirely from a fixed dataset collected by unknown behavior policies [18]. This paradigm holds immense promise for real-world applications where data collection is costly or dangerous, such as robotics [26], healthcare [28], and autonomous driving [5]. However, standard RL algorithms often fail in this setting due to the challenge of *distribution shift*, where the learned policy targets out-of-distribution (OOD) actions that are erroneously overestimated by the value function [19].

To avoid OOD actions, a dominant approach is *behavior regularization*, which restricts the learned policy to remain close to the behavior policy via divergence measures. Early methods employ simple Mean Squared Error (MSE) [7] for action cloning, which inherently assume unimodal distributions and often induce over-conservatism [16] or mode collapse [36] on complex datasets. To alleviate such drawbacks, subsequent approaches incorporate generative models (e.g., VAE [31], flow-GAN [36]) and distribution divergence (e.g., Kullback-Leibler (KL) [9], Maximum Mean Discrepancy (MMD) [16]) to constrain the learned policy within the behavior support, successfully generalizing beyond mere data points. While this paradigm shift brings initial improvements, the limited expressivity of these generative models eventually bottlenecks performance, prompting recent methods to leverage *diffusion* [37] and *flow matching* [23] for more precise modeling. However, these methods fundamentally remain rooted in direct action distribution matching, which is inherently

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

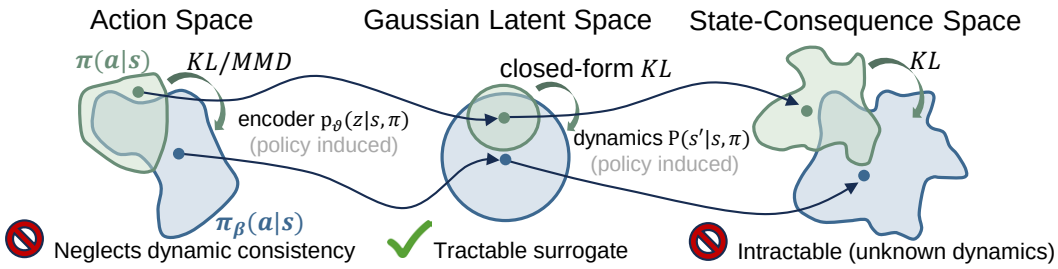

Figure 1: Conceptual illustration of divergence-based behavior regularization. Purely action distribution matching (left) neglects dynamic consistency, whereas ideal consequence-aware constraints (right) are computationally intractable without access to the true dynamics. Our StBR (middle) employs a Gaussian latent space as a tractable, closed-form surrogate of consequence-aware constraints.

agnostic to the underlying dynamic consistency. In practice, due to function approximation and finite dataset coverage, estimation errors in the learned action distribution are practically inevitable. As illustrated in Figure 1 (left), even minor action deviations can lead to drastically different state transitions. Consequently, the agent inadvertently deviates from the offline dataset, particularly in dynamics-sensitive regions.

To address this, we explore policy constraints through their induced state-consequence distributions rather than relying on pure action-distribution matching. However, directly enforcing such consequence-aware constraints is computationally intractable: it requires access to the true, unknown environment dynamics, and explicitly estimating dynamics density from offline data is notoriously infeasible (as shown in Figure 1, right). To bypass this issue, we replace the direct evaluation of consequence constraints with a tractable surrogate divergence in latent space. Specifically, we propose Structural Behavior Regularization (StBR)[1], a novel framework that approximates the intractable objective via a closed-form KL divergence computed within a structured, consequence-aware Gaussian latent space (Figure 1, middle).

To effectively construct this latent space, StBR introduces a State Belief Gaussian Fitting technique grounded in the Variational Information Bottleneck principle [1]. By mapping state-action pairs into the latent space, State Belief Gaussian Fitting extracts the essential consequence-aware information and structures the aggregated behavior embeddings into a unimodal Gaussian distribution, serving as a state belief. It is crucial to note that this unimodality is not an assumption about the real-world action or consequence distributions; rather, it is purely a geometric constraint imposed on the latent space. We theoretically establish that the discrepancy between the true consequence divergence and this surrogate is provably bounded, guaranteeing a reliable approximation. Furthermore, because this latent space captures environmental dynamics information, minor action deviations naturally trigger drastic shifts in the latent representations. By regularizing the learned policy with this latent divergence, StBR translates these representation shifts into an adaptive penalty, effectively tightening the behavior constraint in dynamics-sensitive regions. Empirically, StBR achieves superior performance across diverse offline RL benchmarks.

Our main contributions are summarized as follows:

- We propose Structural Behavior Regularization (StBR), a novel framework that enforces conservatism through the policy's induced consequences via a surrogate closed-form Gaussian latent divergence without dynamics density estimation.
- We introduce State Belief Gaussian Fitting to impose structure on the latent representations, and theoretically demonstrate that the proposed surrogate objective serves as a bounded approximation of the intractable consequence divergence.
- Empirically, StBR achieves superior performance across diverse offline RL benchmarks.

[1]Code is available at: `https://github.com/DarrenJum/StBR`

2 Preliminaries and Motivation

2.1 Reinforcement Learning

In RL, we generally model the environment as a Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma, \rho_0)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward function $R : \mathcal{S} \times \mathcal{A} \to [0, R_{\max}]$, discount factor $\gamma \in [0, 1)$, and initial state distribution ρ_0, where $\Delta(\cdot)$ denotes the probability simplex. The agent interacts with the environment and seeks a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ to maximize the expected discounted return $J(\pi) := \mathbb{E}_{s_0 \sim \rho_0, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)}[\sum_{t=0}^{\infty} \gamma^t R_t(s_t, a_t)]$. The objective of reinforcement learning is to find an optimal policy: $\pi^\star := \operatorname{argmax}_\pi J(\pi)$. It will be useful to define the *policy-specific* state-action value function (Q-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$) as $Q^\pi(s, a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R_t(s_t, a_t) | s_0 = s, a_0 = a]$. Then we define the discounted state occupancy for convenience $d^\pi(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t Pr(s_t = s; \pi, \rho_0)$.

In large and complex state-action spaces, function approximation is typically employed to model the value network Q_θ with parameters θ and the policy network π_ϕ with parameters ϕ. In practice, an off-policy iterative approach commonly initializes parameters arbitrarily, and iteratively updates the parameterized function with an experience replay dataset $\mathcal{D}$ and a target network $Q_{\bar{\theta}}$ [20]:

$$\mathcal{L}_Q(\theta) = \mathop{\mathbb{E}}_{(s,a,r,s') \sim \mathcal{D}} \left[Q_\theta(s, a) - r - \gamma Q_{\bar{\theta}}(s', \pi_\phi) \right]^2, \tag{1}$$

where $Q_{\bar{\theta}}(s', \pi_\phi)$ is shorthand for $\mathbb{E}_{a' \sim \pi_\phi(\cdot|s')}[Q_{\bar{\theta}}(s', a')]$. The policy π_ϕ can be updated by:

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{s \sim \mathcal{D}, a \sim \pi_\phi(\cdot|s)}[-Q_\theta(s, a)]. \tag{2}$$

2.2 Offline Reinforcement Learning

In offline RL, the agent has no direct interactive access to the environment but only to a fixed dataset $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^N$ collected by an unknown behavior policy π_β. The fundamental challenge in offline RL is the *distribution shift*: querying the learned value function on OOD actions can lead to severe overestimation.

2.3 Motivation: Rethinking Behavior Conservatism

A prominent category of offline RL methods follows the key insight of maintaining conservatism in policy learning, which constrains the learned policy to remain close to the dataset by avoiding OOD actions. Existing methods typically achieve this by regularizing the action distribution of the learned policy π_ϕ toward the behavior policy π_β via a divergence measure:

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{s \sim \mathcal{D}, a \sim \pi_\phi(\cdot|s)} \left[-Q_\theta(s, a) + \lambda D\big(\pi_\phi(a|s), \pi_\beta(a|s)\big) \right], \tag{3}$$

where λ is the Lagrange multiplier, D is a chosen divergence measure, such as Kullback-Leibler (KL) divergence in BRAC & AWAC [30, 21], Maximum Mean Discrepancy (MMD) in BEAR [16], or Mean Squared Error (MSE) in TD3BC [7]. However, enforcing regularizations purely on the action distribution is blind to the underlying dynamic consistency structure, leading to a critical issue:

Consequence-shift vulnerability: In dynamics-sensitive regions, even minor action deviations that incur a minimal divergence penalty can trigger drastically different transitions, inadvertently propelling the agent into OOD next states (consequences).

This limitation suggests that distribution shift is a problem related to *dynamic consistency*. Therefore, rather than constraining action-distributions, we propose constraining their induced state-consequence distributions $P_\pi(s'|s) := \mathbb{E}_{a \sim \pi(\cdot|s)} P(s'|s, a)$. We formulate this constraint using KL divergence, as its zero-forcing property strictly penalizes visiting out-of-support states, naturally preventing OOD extrapolation. Our ideal objective is thus:

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{s \sim \mathcal{D}, a \sim \pi_\phi(\cdot|s)} \left[-Q_\theta(s, a) + \lambda D_{\mathrm{KL}}\big(P_{\pi_\phi}(s'|s) \| P_{\pi_\beta}(s'|s)\big) \right]. \tag{4}$$

The Tractability Challenge. Directly optimizing Eq. (4) is computationally intractable in the standard offline RL setting, as enforcing such a constraint requires explicit density estimation of the true underlying dynamics P. This intractability directly motivates our Structural Behavior Regularization (StBR) framework, detailed in the following section.

3 Method

In this section, we present the Structural Behavior Regularization (StBR) framework. We first introduce a latent variable to derive a tractable surrogate divergence (Section 3.1). Next, we propose the State Belief Gaussian Fitting (SBGF) technique to structure this latent space (Section 3.2), provide theoretical guarantees for this approximation (Section 3.3), and finally integrate the surrogate divergence into offline RL policy optimization (Section 3.4).

3.1 Tractable Surrogate via Gaussian Latent Divergence

We begin by introducing a latent variable z as a sufficient statistic of the state-action pair (s, a) for predicting the next state s'. Specifically, we assume the dynamics follow the causal diagram:

$$(s, a) \to z \to s'. \tag{5}$$

This causal structure implies conditional independence, $(s, a) \perp\!\!\!\perp s'|z$ which naturally decouples the transition dynamics as $P(s'|s, a) = \int_z p(s'|z)p(z|s, a)\, dz$. Crucially, this decomposition provides a theoretical motivation formalized in Proposition 1. The detailed proof can be found in Appendix B.1. Let the subscript π denote expectation over $a \sim \pi(\cdot|s)$ (e.g., $p_\pi(z|s) = \mathbb{E}_{a\sim\pi}[p(z|s, a)]$).

Proposition 1 (Surrogate Consequence Divergence)**.** *Assume z is a latent variable acting as a sufficient statistic for the state-action pair (s, a) to predict the next state s', forming the causal diagram $(s, a) \to z \to s'$. For any policy π and behavior policy π_β, the KL divergence between their consequence state distributions is upper-bounded by the KL divergence between their corresponding latent distributions:*

$$D_{\mathrm{KL}}\big(P_\pi(s'|s)\|P_{\pi_\beta}(s'|s)\big) \le D_{\mathrm{KL}}\big(p_\pi(z|s)\|p_{\pi_\beta}(z|s)\big). \tag{6}$$

Proposition 1 provides a theoretical foundation for enforcing consequence conservatism entirely within the latent space. To make this distribution matching computationally tractable, we structurally design the representation bottleneck as a Gaussian latent space. This choice effectively transforms the intractable problem into an analytical, closed-form KL divergence computation between two Gaussian distributions in the latent space. Consequently, the burden of intractable dynamics density estimation is shifted to a tractable representation learning task, which we address in the next section.

3.2 State Belief Gaussian Fitting

To implement the representation learning task outlined above and enforce the causal structure $(s, a) \to z \to s'$, we build upon the Variational Information Bottleneck (VIB) principle [1]. Specifically, our objective is to maximize the mutual information between the latent representation Z and the next state S', while minimizing the information between Z and the input pair (S, A):

$$\max I(Z; S') - \beta I(Z; S, A), \tag{7}$$

where we use the chain rule of mutual information to expand $I(Z; S, A) = I(Z; S) + I(Z; A|S)$. To optimize this mutual information objective, we derive a tractable objective, detailed in Appendix B.2.

Proposition 2 (State Belief Gaussian Fitting based on VIB)**.** *Let Z be a stochastic latent representation of a state-action pair (S, A) generated by an encoder $p_\vartheta(z|s, a)$. Let $q_\psi(s'|z)$ be a variational decoder, $p_\varphi(z|s)$ be a state belief, and $p(z)$ be a fixed marginal prior (e.g., $\mathcal{N}(0, I)$). The VIB objective $\max I(Z; S') - \beta I(Z; S, A)$ can be optimized via the following approximation objective:*

$$\mathcal{L}_{\mathrm{SBGF}}(\psi, \vartheta, \varphi) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\Big[\underbrace{\mathbb{E}_{z\sim p_\vartheta(\cdot|s,a)}[-\log q_\psi(s'|z)]}_{\textit{Reconstruction}} + \beta\Big(\underbrace{D_{\mathrm{KL}}(p_\vartheta(z|s, a)\|p_\varphi(z|s))}_{\textit{Bidirectional Action IB}} + \underbrace{D_{\mathrm{KL}}(p_\varphi(z|s)\|p(z))}_{\textit{State IB}}\Big)\Big]. \tag{8}$$

To estimate the latent divergence in Eq. (6), the encoder $p_\vartheta(z|s, a)$ and decoder $q_\psi(s'|z)$ model environment dynamics, yielding the target policy's distribution by taking the expectation over actions: $\hat{p}_\pi(z|s) = \mathbb{E}_{a\sim\pi(\cdot|s)}p_\vartheta(z|s, a)$. Meanwhile, the state belief encoder $p_\varphi(z|s)$ directly captures the behavior policy's consequences, providing $\hat{p}_{\pi_\beta}(z|s) = p_\varphi(z|s)$.

The core innovation of our SBGF lies in the *Bidirectional Action IB* term. Directly integrating the encoder over the actions of the behavior policy would yield a complex Gaussian Mixture Model, making the analytical calculation of latent divergence in Eq. (6) intractable. To circumvent this, the bidirectional constraint fits the aggregated behavioral representations into a Gaussian state belief (p_φ), which simultaneously serves to regularize the encoder.

Through this representation learning framework, we establish the structured Gaussian latent space that enables the analytical computation of the surrogate divergence. However, a critical question naturally arises: *when using this surrogate divergence to approximate the true consequence divergence, how large is the resulting discrepancy?* We formally analyze this gap in the following section.

3.3 Theoretical Guarantee: Bounding the Surrogate Discrepancy

To address the question raised above, we now formalize a theoretical upper bound on the gap between the true consequence divergence and our surrogate latent divergence estimated from the dataset.

Theorem 1 (Surrogate Discrepancy Bound)**.** *Let $\hat{p}_\pi(z|s) = \mathbb{E}_{a\sim\pi(\cdot|s)} p_\vartheta(z|s,a)$ for any given policy π, and let $\hat{p}_{\pi_\beta}(z|s) = p_\varphi(z|s)$ be the models estimated by SBGF in Eq.* (8) *on an offline dataset $\mathcal{D}$ of size N. With probability at least $1-\delta$, the following bound holds for the discrepancy between the true consequence divergence and the empirical surrogate divergence:*

$$\left|\mathbb{E}_{s\sim d^{\pi_\beta}}\left[D_{\mathrm{KL}}(P_\pi(s'|s)\|P_{\pi_\beta}(s'|s))\right] - \mathbb{E}_{s\sim\mathcal{D}}\left[D_{\mathrm{KL}}(\hat{p}_\pi(z|s)\|\hat{p}_{\pi_\beta}(z|s))\right]\right| \tag{9}$$

$$\leq \underbrace{\mathbb{E}_{s,s'}\left[D_{\mathrm{KL}}(p_\pi(z|s,s')\|p_{\pi_\beta}(z|s,s'))\right]}_{\textbf{(I) Dynamics Info Compression (Bias) (Increases as } \beta\uparrow\textbf{)}} + \underbrace{\mathcal{O}\left(\sqrt{\frac{\mathrm{Var}_s(D_{\mathrm{KL}}(p_\pi(z|s)\|p_{\pi_\beta}(z|s))\log(1/\delta)}{N}}\right)}_{\textbf{(II) Estimation Error (Variance) (Decreases as } \beta\uparrow\textbf{)}} + \underbrace{C_\pi \epsilon_{\mathrm{opt},\beta}}_{\textbf{(III) Policy Shift}}.$$

where the policy ratio in the latent space is bounded by $C_\pi = \max_{s\in\mathcal{D},z} \frac{p_\pi(z|s)}{p_{\pi_\beta}(z|s)}$, and $\epsilon_{\mathrm{opt},\beta}$ is an error related to the bidirectional action IB loss in Eq. (8) *and fundamental approximation limit of fitting a single Gaussian to a potentially multimodal behavior-induced latent distribution.*

Remark. Theorem 1 reveals that our surrogate approximation is systematically governed by a fundamental Bias-Variance trade-off dictated by the information compression parameter β. *(Note that an excessively large β forces the latent z to collapse to the prior, losing all dynamics prediction capability. Thus, the meaningful trade-off occurs in the regime of relatively small β.)* The detailed proof can be found in Appendix B.3.

Term (I) acts as the Bias. When β is small, the mapping $z \to s'$ is near-deterministic. Consequently, $p_\pi(z|s,s')$ is largely policy-independent, keeping this bias low. As β increases, compression maps multiple z to the same s', exposing policy differences and increasing the bias.

Term (II) acts as the Variance. It captures the cross-state variance of the latent divergence defined in Eq. (6). Increasing β acts as a strong regularizer, smoothing out action details to effectively suppress the fluctuation of $D_{\mathrm{KL}}(p_\pi(z|s)\|p_{\pi_\beta}(z|s))$.

Term (III) highlights the conservatism. The policy ratio C_π indicates that this surrogate becomes ineffective when evaluating OOD actions, directly motivating the necessity of policy constraint.

This theoretical insight provides a principled *guideline for empirical β tuning*: a larger β mitigates variance in noisy or stochastic environments, whereas a smaller β preserves dynamics and minimizes bias in near-deterministic settings. Ultimately, Theorem 1 mathematically justifies the SBGF design, establishing a tractable surrogate for the policy optimization detailed in the next section.

3.4 Implementation: Structural Behavior Regularization

The surrogate divergence derived above can be plugged into standard off-policy algorithms with minimal modifications. We choose TD3 [7, 21, 31] as our base RL algorithm to verify the effectiveness of the proposed StBR framework.

Point-wise Constraint. While the surrogate divergence can be directly computed when π_ϕ is parameterized as a simple Gaussian or deterministic policy, it becomes intractable for more expressive generative model policies (e.g., flow policies [23]). A practically feasible alternative is to instead

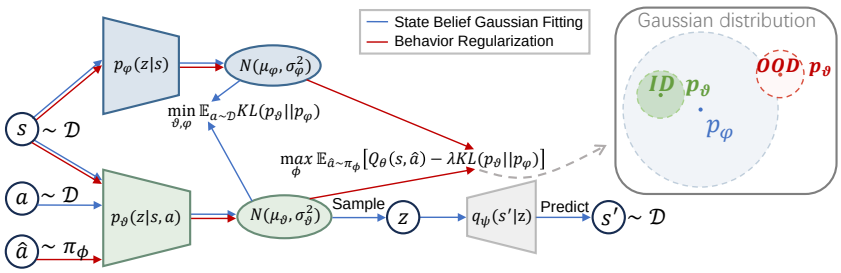

Figure 2: The architecture of the proposed StBR framework.The blue flow represents the learning of latent representations (Eq. (8)) through the proposed State Belief Gaussian Fitting. The red flow highlights the application of the surrogate latent divergence for behavior regularization in Eq. (12).

minimize its point-wise upper bound:

$$D_{\mathrm{KL}}(\hat{p}_{\pi_\phi}(z|s)\|\hat{p}_{\pi_\beta}(z|s)) \leq \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}\big[D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s))\big]. \tag{10}$$

Beyond computational efficiency, this surrogate objective avoids a risk-dilution issue where averaging over multiple samples can mask OOD actions. The proof sketch of Theorem 1 in Appendix B.4 details this mechanism.

Closed-form Gaussian Latent Divergence. Leveraging the Gaussian models $p_\vartheta(z|s,a) = \mathcal{N}(\mu_\vartheta, \sigma_\vartheta|s,a)$ and $p_\varphi(z|s) = \mathcal{N}(\mu_\varphi, \sigma_\varphi|s)$ learned via SBGF in Eq. (8), the point-wise divergence can be computed analytically without sampling:

$$D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s)) = -\frac{1}{2}\Big(1 + \log\frac{\sigma_\vartheta^2}{\sigma_\varphi^2} - \frac{\sigma_\vartheta^2 + (\mu_\vartheta - \mu_\varphi)^2}{\sigma_\varphi^2}\Big). \tag{11}$$

Policy Optimization. Enforcing our constraint is computationally elegant, as it simply requires aligning the representations from two encoders. Fusing this lightweight latent divergence constraint with the standard Q-value objective, we update the policy π_ϕ by:

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\phi(\cdot|s)}\big[-Q_\theta(s,a) + \lambda D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s))\big], \tag{12}$$

where λ is a hyperparameter controlling the regularization strength.

Overall algorithm. StBR follows a standard actor-critic paradigm and optimizes the SBGF via Eq. (8) right after the value and policy updates. Figure 2 gives the architecture of the proposed StBR framework, while the whole training process is shown in Appendix C.1.

4 Related Works

Behavior regularized offline RL. Behavior regularization has evolved from early MSE cloning to VAE-based divergence matching, and more recently toward high-expressivity diffusion and flow matching models [7, 17, 31, 36, 29, 23]. These advancements focus on achieving increasingly precise modeling of the underlying behavior policy distribution to capture its full complexity. However, such action-distribution matching paradigms remain agnostic to environmental dynamics, where minor action deviations can trigger significant state drift due to a lack of dynamic consistency.

Consequence-aware offline RL. Several offline RL approaches can be viewed as consequence-aware by anticipating future states. For instance, POR [33] employs hierarchical state-planning with an inverse model, while ODAC [12] constrains actions using explicit dynamics models and value uncertainty. Alternatively, out-of-sample state recovery methods (SDC [35] and OSR [9]) utilize inverse dynamics or dynamics ensembles to regularize the policy via transition alignment, ensuring agents can recover to the data manifold, thereby enhancing generalization. Unlike these methods, which struggle in high-dimensional environments due to their reliance on computationally heavy explicit state predictions or ensemble models, our StBR imposes a lightweight consequence-aware policy constraint directly via a latent divergence.

Other competitive offline RL methods. Besides density matching, more types of competitive offline RL methods exist. Conservative Q-Learning (CQL) [17] implicitly constrains action distributions via

Table 1: The mean and standard deviation of normalized scores after 1M steps of offline RL training, averaged over 5 seeds and the final 10 evaluations (the final 4 for AntMaze). The highest score in each task is bolded, and the second-best is underlined.

	Task Name	BC	CQL	TD3BC	IQL	PBRL	POR	SDC	SPOT	FQL	ENSP	StBR (Ours)
halfcheetah	medium	41.2±0.4	46.9±0.4	48.2±0.1	47.1±0.1	57.9±1.5	48.8±0.5	45.9±0.5	58.4±1.0	54.4±2.2	55.4±0.6	**60.7**±0.7
	medium-replay	32.9±3.4	45.3±0.3	44.6±0.1	44.3±0.3	45.1±8.0	43.5±0.9	45.1±0.5	52.2±1.2	53.2±0.6	43.4±1.5	**54.4**±1.0
	medium-expert	50.4±4.0	95.0±1.4	91.7±1.9	89.2±1.2	92.3±1.1	94.7±2.2	76.3±5.2	86.9±4.3	95.3±1.3	88.1±4.9	**97.8**±0.2
hopper	medium	46.1±1.8	61.9±6.4	59.0±0.7	63.6±2.9	75.3±31.2	78.6±7.2	64.7±3.5	86.0±8.7	77.1±10.3	80.4±14.7	**100.9**±1.0
	medium-replay	22.4±5.9	86.3±7.3	65.4±14.0	95.4±4.9	100.6±1.0	98.9±2.1	94.8±6.5	100.2±1.0	79.5±4.5	91.0±12.9	**100.8**±1.9
	medium-expert	62.4±13.9	96.9±15.1	101.6±2.0	101.6±7.4	110.8±0.8	90.0±12.1	99.9±8.5	99.3±7.1	103.5±6.7	65.5±12.5	**111.6**±0.9
walker2d	medium	62.0±7.7	79.5±3.2	83.8±0.3	80.2±2.8	**89.6**±0.7	81.1±2.3	82.7±1.9	86.4±2.7	81.9±3.4	80.4±3.8	87.2±0.9
	medium-replay	26.4±8.4	76.8±10.0	80.3±1.8	69.9±8.4	77.7±14.5	76.6±6.9	78.5±6.0	**91.6**±2.8	76.4±5.6	78.5±6.9	86.3±1.5
	medium-expert	81.1±12.5	109.1±0.2	110.1±0.3	109.6±7.4	110.1±0.3	109.1±0.7	109.2±1.4	**112.0**±0.5	111.3±0.3	109.4±0.8	111.3±1.4
MuJoCo Total		424.9	697.7	684.7	700.9	759.4	721.3	697.1	773.0	732.5	707.1	**811.0**
antmaze	umaze	58.8±5.1	92.8±1.9	70.8±39.2	89.6±4.2	0.0±0.0	90.6±7.1	81.4±3.8	93.5±2.4	**96.0**±2.0	94.3±3.0	**96.0**±2.3
	umaze-diverse	62.0±5.9	37.3±3.7	44.8±11.6	65.6±8.3	0.0±0.0	71.3±12.1	49.6±10.4	40.7±5.1	**89.0**±5.0	69.1±10.4	83.4±7.2
	medium-play	0.0±0.0	65.8±11.6	0.3±0.4	76.4±2.7	0.0±0.0	84.6±5.6	55.0±9.6	74.7±4.6	78.0±7.0	75.4±8.5	**94.2**±1.9
	medium-diverse	0.2±0.0	67.3±3.6	0.3±0.4	72.8±7.0	0.0±0.0	79.2±3.1	56.6±10.3	79.1±5.6	71.0±13.0	73.4±8.0	**90.6**±3.6
	large-play	0.0±0.0	20.8±7.3	0.0±0.0	42.0±3.8	0.0±0.0	58.0±12.4	20.8±8.0	35.3±8.3	84.0±7.0	55.4±4.8	**84.2**±2.7
	large-diverse	0.0±0.0	20.5±13.2	0.0±0.0	46.0±4.5	0.0±0.0	73.4±8.5	25.8±7.5	36.3±13.7	83.0±4.0	59.0±9.8	**85.8**±5.1
AntMaze Total		121.0	304.5	116.2	392.4	0.0	457.1	289.2	359.6	501.0	426.6	**534.2**
pen	human	70.2±9.4	13.7±17.0	-3.9±0.2	78.5±8.2	35.4±3.3	76.9±5.1	64.5±7.5	47.5±7.4	53.0±6.0	-3.1±0.0	**99.5**±3.4
hammer	human	1.2±0.9	0.1±0.1	-0.3±0.1	1.8±0.8	0.4±0.3	1.9±0.7	0.7±0.5	1.2±0.4	1.0±1.0	0.3±0.1	**3.1**±0.6
door	human	1.4±1.8	5.5±1.3	1.0±0.0	3.3±1.8	0.1±0.0	3.0±1.2	0.0±0.0	0.0±0.0	0.0±0.0	-0.2±0.1	**7.9**±2.0
Adroit Total		72.8	19.3	-3.2	83.6	35.9	81.8	65.2	48.7	54.0	-3.0	**110.5**
Wall time(min)		33	403	78	108	444	147	417	111	110	108	114

Q-value penalization. PBRL [2] enforces pessimism using ensemble variance; however, alongside high computational costs, this uncertainty becomes uninformative in sparse-reward environments due to lack of reward signals. Conversely, IQL [15] eliminates OOD queries via in-sample expectile regression, but its strict reliance on existing data limits generalization. We provide a broader discussion of additional related works in Appendix A.

5 Experiments

5.1 Performance Comparison

Tasks. We evaluate StBR across three domains: continuous-control MuJoCo locomotion, sparse-reward AntMaze, and high-dimensional Adroit human manipulation.

Hyperparameter tuning. The coefficient λ in Eq. (12) is essential for behavior regularization. Following prior works [16, 30, 31], we allow limited access to the online environment to tune this hyperparameter across a minimal candidate set of $\{0.03, 0.05, 0.1\}$ (only 3 choices), which is a reasonable setup for practical applications. See Appendix C.1 for additional discussion and details.

Main Results. Table 1 shows that StBR outperforms existing baselines across diverse benchmarks, underscoring two key advantages. First, unlike action-distribution-based methods (e.g., CQL, SPOT, FQL) that suffer from consequence-shift vulnerabilities, StBR effectively facilitates trajectory stitching in long-horizon, sparse-reward tasks like AntMaze. While specific baselines perform selectively, such as FQL's flow modeling succeeding in AntMaze but turning overly conservative in dense-reward MuJoCo, or PBRL collapsing in sparse rewards due to ungrounded Q-variance guidance, StBR delivers robust performance across both regimes. Second, StBR circumvents the dynamics modeling errors inherent to explicit consequence-based approaches (e.g., POR, SDC, and an explicit next-state prediction baseline, ENSP). Although explicit next-state modeling can aid stitching in low-dimensional tasks, it scales poorly to high-dimensional state spaces like Adroit due to the curse of dimensionality. For instance, ENSP achieves competitive scores in low-dimensional MuJoCo but exhibits severe variance, trails implicit planning in AntMaze, and fails catastrophically in Adroit (45-dim). By enforcing consequence constraints implicitly, StBR avoids intractable dynamics modeling while retaining consequence conservatism, achieving state-of-the-art performance throughout.

Runtime. We evaluated the runtime (in minutes) on the `hopper-medium`. As shown in Table 1, StBR incurs a computational overhead comparable to that of most behavior regularization methods, demonstrating the high efficiency of our implementation.

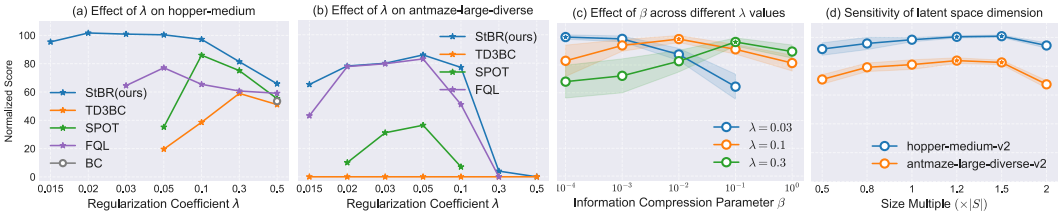

Figure 3: Hyperparameter sensitivity analysis. (a-b) When evaluating the performance at varying levels of constraint strength (λ) on two representative tasks, StBR takes shape of the "upper envelope" of all methods, showing that StBR is highly robust to parameter choices, making it significantly easier to achieve superior performance during practical deployment. (c-d) Sensitivity analysis of SBGF-related hyperparameters regarding β and latent space dimension.

5.2 Hyperparameter Analysis of StBR

Effect of λ. The Lagrangian multiplier λ in Eq. (12) is essential in regularization methods, as it controls the balance between policy improvement and conservatism. However, exhaustive hyperparameter tuning is prohibitively costly in offline settings. Consequently, existing methods [7, 31, 23] typically restrict tuning to a minimal candidate set. However, when deploying algorithms on new tasks in practice, the ability to quickly find an effective λ varies across different methods, which directly reflects their ease of use.

To evaluate method usability, we sweep λ from 0.015 (loose conservatism) to 0.5 (over-conservatism) to demonstrate which method can reliably achieve superior performance under a standardized tuning process. Note that for a fair comparison, we apply a Q-value normalization trick to align the objective scales across all baselines (more implementation details are provided in Appendix C.1).

Figure 3 (a-b) presents the results on the `hopper-medium-v2` and `antmaze-large-diverse-v2` tasks, two representative tasks from the D4RL benchmark widely known for their high sensitivity to hyperparameters due to a large proportion of suboptimal data. When evaluating performance across varying values of λ, StBR consistently outperforms all baseline methods. This demonstrates that StBR is highly robust to parameter choices, making it significantly easier to achieve superior performance during practical deployment.

SBGF-related hyperparameters. First, the information compression parameter β controls the information bottleneck constraint: a larger β forces the latent representation to be closer to a standard Gaussian distribution, resulting in a highly compressed latent space. Consequently, the numerical magnitude of the deviation within this space shrinks. To compensate for this reduced scale and maintain an effective behavior regularization penalty, a larger weight λ is required. Figure 3 (c) demonstrates this dynamic coupling: as λ increases, the optimal β (marked with a star) consistently shifts towards larger values. Notably, StBR achieves competitive performance across these configurations. Furthermore, Theorem 1 suggests that the optimal β is also inherently task-dependent, a property we further analyze in Section 5.3.

Next, we examine the sensitivity of StBR to the latent representation capacity by varying its dimension $|Z| \in [0.5, 2] \times |S|$, as illustrated in Figure 3 (d). StBR demonstrates strong robustness across a broad intermediate range (from 1 to 1.5 $\times|S|$). Performance degradation is primarily observed at the extremes: an excessively low dimension severely bottlenecks information, preventing the model from capturing necessary environmental dynamics, whereas an overly high dimension introduces representation redundancy and optimization challenges. Overall, the empirical results indicate that StBR is highly robust to SBGF-related hyperparameters within reasonable configurations.

5.3 Theoretical Validations

Validating the Bias-Variance Trade-off (β). Theorem 1 suggests that β should be appropriately larger in high-dimensional tasks and highly stochastic environments. To empirically validate this theoretical insight, we evaluate StBR under two distinct regimes:

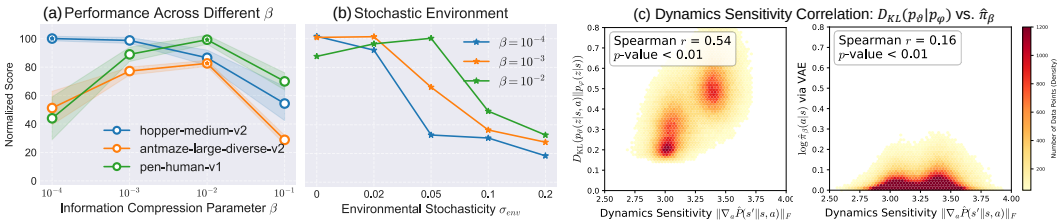

Figure 4: Theoretical validation experiments. (a-b) Demonstrating that a larger β should be chosen in high-dimensional tasks and highly stochastic environments to maintain superior performance. (c) Correlation analysis between dynamic sensitivity and different metrics: the proposed consequence-aware divergence and the action-distribution measure. Results validate that latent divergence (left, Spearman $r = 0.54$) exhibits a significantly stronger correlation with dynamics sensitivity than the VAE-based density measure (right, $r = 0.16$).

(a) *Task dimensions:* We compare the performance of StBR across tasks with different state dimensions by varying $\beta \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. As illustrated in Figure 4 (a), for the low-dimensional `hopper` task, the optimal β is 10^{-4}. In contrast, for the high-dimensional `antmaze` and `pen` tasks, the optimal β shifts to 10^{-2}.

(b) *Environment stochasticity:* We evaluate StBR on the `hopper-medium` task with artificially injected action execution stochasticity (i.e., simulated motor errors). As shown in Figure 4 (b), as the variance of the injected noise (σ^2_{env}) increases (higher environmental stochasticity), the optimal value of β correspondingly increases to maintain robust performance.

Correlation with Dynamics Sensitivity. A core motivation of StBR is to adaptively tighten behavior constraints in regions where transitions are highly sensitive to action deviations. To validate this, we estimate dynamic sensitivity using the Frobenius norm of the Jacobian matrix of a pre-trained dynamics model $\|\nabla_a \hat{P}(s'|s,a)\|_F$. Figure 4(c) illustrates the relationship between different metrics and dynamics sensitivity for state-action pairs from the `hopper-medium` dataset. The proposed consequence-aware divergence demonstrates a strong positive correlation, yielding a Spearman coefficient of 0.54 ($p < 0.01$). In contrast, the action-distribution measure via VAE [31], $\hat{\pi}_\beta(a|s)$, exhibits a significantly weaker relationship, with a correlation of only 0.16 ($p < 0.01$). These results confirm that latent divergence effectively serves as a consequence-aware surrogate, naturally imposing stricter regularization in dynamics-sensitive regions.

5.4 Ablation Study

Bidirectional Action IB. First, we ablate the bidirectional constraint in Eq. (8). Results are presented in Figure 5 (left). As expected, removing this constraint causes our objective to degenerate into a standard VIB. Under this formulation, the state belief completely collapses into a standard Gaussian prior, thereby losing the critical semantic information regarding the high-probability consequences induced by the behavior policy. Consequently, the variant suffers catastrophic performance degradation, even entirely ineffective on high-dimensional environments such as the AntMaze and Adroit tasks.

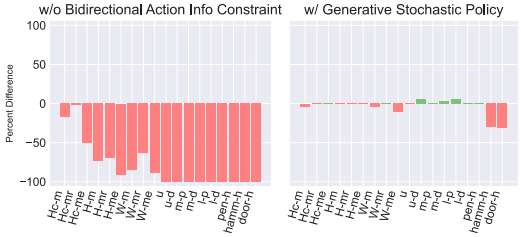

Figure 5: Percent performance difference of StBR variants compared to the original algorithm.

Are stochastic policies better than deterministic policies? Standard Gaussian stochastic policies are often disadvantageous in offline policy constraint methods due to the mode-averaging problem [31]. To investigate the performance of StBR when learning a stochastic policy, we train a one-step generative policy based on the FQL [23] framework. As depicted in Figure 5 (right), the results demonstrate that the generative stochastic policy primarily exhibits advantages in sparse-reward tasks

(AntMaze). Conversely, in tasks where the optimal actions are near-deterministic, the stochastic policy generally underperforms the deterministic one. This difference arises because stochastic policies assign non-zero probability mass to sub-optimal actions. Extensive supplementary experiments further validating the capabilities of StBR are detailed in Appendix C.

6 Conclusion

This paper proposes the Structural Behavior Regularization (StBR) framework, reformulating behavior regularization as a consequence-aware constraint process to solve the consequence-shift vulnerability problem in offline RL. The proposed surrogate latent divergence avoids dynamic density estimation via structured representation learning and is theoretically proven to be a reliable approximation. Experiments show StBR achieves superior performance across diverse offline RL benchmarks.

Limitations. While StBR achieves strong empirical performance, it comes with several limitations that suggest promising directions for future research. First, StBR imposes conservatism relative to the behavior policy, meaning that severely sub-optimal behavioral data may induce an inherent tension between conservatism and policy improvement; developing adaptive or evolving regularization mechanisms could mitigate this conflict. Second, the unimodal Gaussian latent assumption embodies a fundamental trade-off between expressiveness and tractability. Due to its light-tailed nature and propensity for mode-averaging, a single Gaussian struggles in regimes with severe multimodality or sharp transition boundaries—such as environments with heavy-tailed stochasticity where rare transitions govern task success (as observed in our stochastic environment), or systems with contact discontinuities and bifurcations where unimodal priors induce artificial smoothing. Extending our latent representation to expressive multimodal priors (e.g., mixtures or diffusion models) is a natural avenue to capture such dynamics. Finally, following standard practice in current offline RL literature, our setup relies on limited online evaluations for hyperparameter tuning due to the open challenges in offline policy evaluation. Nevertheless, this introduces non-trivial deployment overhead and safety concerns in high-stakes domains where online interactions are strictly prohibited. Developing reliable, strictly offline model selection and evaluation criteria remains an essential prerequisite for fully autonomous deployment.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China (62476128).

References

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

[2] Chenjia Bai, Lingxiao Wang, Zhuoran Yang, Zhihong Deng, Animesh Garg, Peng Liu, and Zhaoran Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. *arXiv preprint arXiv:2202.11566*, 2022.

[3] David Brandfonbrener, Will Whitney, Rajesh Ranganath, and Joan Bruna. Offline rl without off-policy evaluation. *Advances in neural information processing systems*, 34:4933–4946, 2021.

[4] Peng Cheng, Xianyuan Zhan, Wenjia Zhang, Youfang Lin, Han Wang, Li Jiang, et al. Look beneath the surface: Exploiting fundamental symmetry for sample-efficient offline rl. *Advances in Neural Information Processing Systems*, 36:7612–7631, 2023.

[5] Xing Fang, Qichao Zhang, Yinfeng Gao, and Dongbin Zhao. Offline reinforcement learning for autonomous driving with real world driving data. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 3417–3422. IEEE, 2022.

[6] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020.

[7] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20132–20145, 2021.

[8] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2052–2062. PMLR, 2019.

[9] Ke Jiang, Jia-Yu Yao, and Xiaoyang Tan. Recovering from out-of-sample states via inverse dynamics in offline reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[10] Ke Jiang, Yao Li, and Xiaoyang Tan. Toward reliable offline reinforcement learning via lyapunov uncertainty control. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[11] Ke Jiang, Wen Jiang, and Xiaoyang Tan. Variational ood state correction for offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 22327–22335, 2026.

[12] Ke Jiang, Wen Jiang, Li Yao, and Xiaoyang Tan. Beyond non-expert demonstrations: Outcome-driven action constraint for offline reinforcement learning. *Pattern Recognition*, 172:112663, 2026.

[13] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In *Advances in neural information processing systems*, volume 33, pages 21810–21823, 2020.

[14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

[16] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11761–11771, 2019.

[17] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[18] Sascha Lange, Thomas Gabel, and Martin A. Riedmiller. Batch reinforcement learning. In Marco A. Wiering and Martijn van Otterlo, editors, *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pages 45–73. Springer, 2012.

[19] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020.

[20] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015.

[21] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

[22] Mitsuhiko Nakamoto, Simon Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *Advances in Neural Information Processing Systems*, 36:62244–62269, 2023.

[23] Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. *arXiv preprint arXiv:2502.02538*, 2025.

[24] Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in neural information processing systems*, 35:16082–16097, 2022.

[25] Ohad Shamir, Sivan Sabato, and Naftali Tishby. Learning and generalization with the information bottleneck. *Theoretical Computer Science*, 411(29-30):2696–2711, 2010.

[26] Samarth Sinha, Ajay Mandlekar, and Animesh Garg. S4rl: Surprisingly simple self-supervision for offline reinforcement learning in robotics. In *Conference on Robot Learning*, pages 907–917. PMLR, 2022.

[27] Yihao Sun, Jiaji Zhang, Chengxing Jia, Haoxin Lin, Junyin Ye, and Yang Yu. Model-bellman inconsistency for model-based offline reinforcement learning. In *International Conference on Machine Learning*, pages 33177–33194. PMLR, 2023.

[28] Shengpu Tang, Maggie Makar, Michael Sjoding, Finale Doshi-Velez, and Jenna Wiens. Leveraging factored action spaces for efficient offline reinforcement learning in healthcare. *Advances in Neural Information Processing Systems*, 35:34272–34286, 2022.

[29] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

[30] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *CoRR*, abs/1911.11361, 2019.

[31] Jialong Wu, Haixu Wu, Zihan Qiu, Jianmin Wang, and Mingsheng Long. Supported policy optimization for offline reinforcement learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[32] Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 6683–6694, 2021.

[33] Haoran Xu, Li Jiang, Li Jianxiong, and Xianyuan Zhan. A policy-guided imitation approach for offline reinforcement learning. *Advances in neural information processing systems*, 35:4085–4098, 2022.

[34] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

[35] Hongchang Zhang, Jianzhun Shao, Yuhang Jiang, Shuncheng He, Guanwen Zhang, and Xiangyang Ji. State deviation correction for offline reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 9022–9030, 2022.

[36] Jing Zhang, Chi Zhang, Wenjia Wang, and Bingyi Jing. Constrained policy optimization with explicit behavior density for offline reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[37] Ruoqi Zhang, Ziwei Luo, Jens Sjölund, Thomas B Schön, and Per Mattsson. Entropy-regularized diffusion policy with q-ensembles for offline reinforcement learning. *Advances in neural information processing systems*, 37:98871–98897, 2024.

[38] Wenxuan Zhou, Sujay Bajracharya, and David Held. Plas: Latent action space for offline reinforcement learning. In *Conference on Robot Learning*, pages 1719–1735. PMLR, 2021.

A Related Work

Behavior-regularized methods. To mitigate the severe extrapolation errors caused by out-of-distribution (OOD) actions, the dominant paradigm in offline RL is behavior regularization, which penalizes the learned policy for deviating from the offline dataset. Early methods enforce this via explicit architectural constraints (e.g., BCQ [8], PLAS [38]) or by minimizing divergence metrics like MSE [7], MMD [16], and KL-divergence [30]. To address the limited expressivity of these early models on complex datasets, recent advancements leverage powerful generative models such as VAEs [31], diffusion models [37], and flow matching [23] to explicitly match the action distribution of the behavior policy. Despite their success, these methods are fundamentally action-distribution centric and inherently agnostic to the underlying dynamics structure. Due to function approximation and finite data, minor action deviations are inevitable. In dynamics-sensitive regions, such minor action deviations can compound into drastically different state transitions, inadvertently driving the agent toward unsafe OOD regions. In contrast, StBR shifts this paradigm by imposing policy constraints based on the induced state-consequences rather than pure action-distribution matching.

Consequence-aware methods. Several offline RL approaches can be viewed as consequence-aware by anticipating future states. For instance, POR [33] employs hierarchical state-planning with an inverse model, while ODAC [12] and LUC [10] constrain actions using explicit dynamics models and value uncertainty. Out-of-sample state recovery methods, such as OSR [9] and SDC [35], utilize inverse dynamics or ensemble models to pull the agent back to the data manifold via transition alignment. While effective in simple environments, these methods rely heavily on explicit state predictions or computationally burdensome ensemble models, making them intractable in high-dimensional continuous control tasks. To the best of our knowledge, StBR is the first consequence-aware method that eliminates the need for explicit future state generation, dynamics density estimation, and ensemble models. StBR elegantly translates the consequence penalty into a lightweight latent divergence, making it highly computationally efficient while retaining the full benefits of forward-looking constraints.

Model-based methods. Another line of research mitigating distribution shift operates by explicitly learning the environment dynamics. Model-based offline RL methods, such as MOPO [34] and MOReL [13], construct forward transition models and penalize policies for visiting states with high model uncertainty. However, such uncertainty estimation heavily relies on heuristics and is highly sensitive to hyperparameters due to the varying scales of uncertainty across states. RAMBO [24] further addresses this by replacing explicit uncertainty penalties with an adversarial framework, which directly modifies the transition dynamics to enforce pessimism. MOBILE [27] further addresses the inconsistency between the learned model and the Bellman operator. The fundamental limitation of these methods lies in their vulnerability to model errors. Accurately learning the underlying transition dynamics from static datasets is notoriously difficult, and inaccuracies in the forward model inevitably lead to severe compounding errors during multi-step rollouts. In contrast, StBR circumvents this by completely omitting explicit state-reconstruction decoders during the policy update phase, while still maintaining dynamics awareness through latent divergence regularization.

Dynamics-based representation learning. Representation learning in offline RL often incorporates dynamics modeling. A common intuition is that learning a superior latent state representation can implicitly capture the system's evolution. Representative methods like TSRL [4] and DASP [11] perform dynamic predictions within the latent space to mitigate over-conservatism and improve generalization. However, these approaches typically conduct the core RL optimization (e.g., value evaluation and policy optimization) entirely within the learned latent space. In contrast, StBR constructs the latent space solely to evaluate the consequence-aware penalty, while the actual RL learning remains grounded in the original observation space. This design makes StBR compatible with them and can be integrated with these approaches to pursue even stronger generalization performance.

B Proofs

This section provides the proofs for the propositions and theories in the paper.

B.1 Proof of Proposition 1

Proposition 1 *(Surrogate Consequence Divergence). Assume z is a latent variable acting as a sufficient statistic for the state-action pair (s, a) to predict the next state s', forming the causal diagram $(s, a) \to z \to s'$. For any policy π and behavior policy π_β, the KL divergence between their consequence state distributions is upper-bounded by the KL divergence between their corresponding latent distributions:*

$$D_{\mathrm{KL}}\big(P_\pi(s'|s) \| P_{\pi_\beta}(s'|s)\big) \le D_{\mathrm{KL}}\big(p_\pi(z|s) \| p_{\pi_\beta}(z|s)\big). \tag{13}$$

Proof. Given the causal diagram $(s, a) \to z \to s'$, the conditional independence $(s, a) \perp\!\!\!\perp s'|z$ naturally follows. Consequently, the transition dynamics can be decomposed via the latent variable z as:

$$P(s'|s, a) = \int p(s'|z, s, a)p(z|s, a)\, dz = \int p(s'|z)p(z|s, a)\, dz. \tag{14}$$

Accordingly, we define the latent distribution under policy π as $p_\pi(z|s) = \int p(z|s, a)\pi(a|s)da$. The transition dynamic then becomes $P_\pi(s'|s) = \int p(s'|z)p_\pi(z|s)\, dz$, where the decoder $p(s'|z)$ is an intrinsic property of the environment and remains independent of the policy.

To establish the bound, we consider the joint distribution $p_\pi(z, s'|s)$ and decompose its KL divergence. By definition, the state transition distribution marginalized over actions is $P_\pi(s'|s) = \sum_a P(s'|s, a)\pi(a|s)$. The complete error decomposition can be structured as follows:

$$\begin{aligned}
& D_{\mathrm{KL}}\big(p_\pi(z|s) \| p_{\pi_\beta}(z|s)\big) && (15)\\
\overset{(a)}{=} & D_{\mathrm{KL}}\big(p_\pi(z, s'|s) \| p_{\pi_\beta}(z, s'|s)\big) && (16)\\
\overset{(b)}{=} & \mathbb{E}_{(z,s')\sim p_\pi}\left[\log \frac{P_\pi(s'|s)p_\pi(z|s, s')}{P_{\pi_\beta}(s'|s)p_{\pi_\beta}(z|s, s')}\right] && (17)\\
\overset{(c)}{=} & D_{\mathrm{KL}}\big(P_\pi(s'|s) \| P_{\pi_\beta}(s'|s)\big) + \mathbb{E}_{s'\sim P_\pi(\cdot|s)}\left[D_{\mathrm{KL}}\big(p_\pi(z|s, s') \| p_{\pi_\beta}(z|s, s')\big)\right] && (18)\\
\overset{(d)}{\ge} & D_{\mathrm{KL}}\big(P_\pi(s'|s) \| P_{\pi_\beta}(s'|s)\big). && (19)
\end{aligned}$$

For (a): We construct the joint distribution conditioned on state s as $p_\pi(z, s'|s) = p_\pi(z|s)p(s'|z)$. Expanding the joint KL divergence along the latent space z yields:

$$\begin{aligned}
& D_{\mathrm{KL}}\big(p_\pi(z, s'|s) \| p_{\pi_\beta}(z, s'|s)\big) && (20)\\
= & \int\int p_\pi(z|s)p(s'|z)\left[\log \frac{p_\pi(z|s)}{p_{\pi_\beta}(z|s)} + \log \frac{p(s'|z)}{p(s'|z)}\right] dz\, ds' && (21)\\
= & \int p_\pi(z|s) \log \frac{p_\pi(z|s)}{p_{\pi_\beta}(z|s)} \underbrace{\left(\int p(s'|z)\, ds'\right)}_{=1} dz + \int p_\pi(z|s) \left(\int p(s'|z) \underbrace{\log 1}_{=0}\, ds'\right) dz && (22)\\
= & D_{\mathrm{KL}}\big(p_\pi(z|s) \| p_{\pi_\beta}(z|s)\big). && (23)
\end{aligned}$$

For (b): We apply Bayes' rule to factorize the joint distribution over (z, s') as: $p_\pi(z, s'|s) = P_\pi(s'|s)p_\pi(z|s, s')$. Here, $p_\pi(z|s, s') = \frac{p(s'|z)p_\pi(z|s)}{P_\pi(s'|s)}$ represents the posterior of the latent variable z conditioned on the next state s'. Substituting this factorization into the joint KL divergence yields Eq. (17).

For (c): We decompose the integral into two parts. Marginalizing the first term over z recovers the marginal KL divergence $D_{\mathrm{KL}}(P_\pi \| P_{\pi_\beta})$. The second term represents the expected KL divergence between the posteriors: $\mathbb{E}_{s'\sim P_\pi(\cdot|s)}\big[\int p_\pi(z|s, s') \log \frac{p_\pi(z|s,s')}{p_{\pi_\beta}(z|s,s')} dz\big]$.

For (d): By definition, the KL divergence is strictly non-negative. Consequently, the second term in Eq. (18) is also non-negative, which yields the final inequality bound.

This concludes the proof.

Remark: Eq. (18) further reveals that the exact gap between the two KL divergences is the expected KL divergence of the posterior latent distributions.

□

B.2 Proof of Proposition 2

Proposition 2 *(State Belief Gaussian Fitting based on VIB). Let Z be a stochastic latent representation of a state-action pair (S, A) generated by an encoder $p_\vartheta(z|s,a)$. Let $q_\psi(s'|z)$ be a variational decoder, $p_\varphi(z|s)$ be a state belief, and $p(z)$ be a fixed marginal prior (e.g., $\mathcal{N}(0, I)$). The VIB objective $\max I(Z;S') - \beta I(Z;S,A)$ can be optimized via the following approximation objective:*

$$\mathcal{L}_{\mathrm{SBGF}}(\psi,\vartheta,\varphi) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\Big[\underbrace{\mathbb{E}_{z\sim p_\vartheta(\cdot|s,a)}[-\log q_\psi(s'|z)]}_{\text{Reconstruction}} + \beta\Big(\underbrace{D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s))}_{\text{Bidirectional Action IB}} + \underbrace{D_{\mathrm{KL}}(p_\varphi(z|s)\|p(z))}_{\text{State IB}}\Big)\Big]. \quad (24)$$

Proof. To learn a predictive yet sufficient representation, we adopt the VIB framework based on the causal diagram $(s,a) \to z \to s'$. The core objective is to maximize the mutual information between the latent representation Z and the target next state S', while constraining the information Z retains about the state-action pair (S, A):

$$\max I(Z;S') - \beta I(Z;S,A) \quad (25)$$

We will examine each of these terms in turn.

Predictive Lower Bound for $I(Z;S')$. We first derive a tractable lower bound for the predictive term $I(Z;S')$. Expanding the mutual information, we have:

$$I(Z;S') = \mathbb{E}_{s'\sim p(s',z)}\Big[\log\frac{p(s'|z)}{p(s')}\Big] \quad (26)$$

$$= \mathbb{E}_{p(s',z)}[\log p(s'|z)] + H(S') \quad (27)$$

Since the true posterior $p(s'|z)$ is intractable, we introduce a parameterized variational decoder $q_\psi(s'|z)$. Using the non-negativity of the KL divergence, $D_{\mathrm{KL}}(p(s'|z)\|q_\psi(s'|z)) \geq 0$, we obtain $\mathbb{E}_{p(s'|z)}[\log p(s'|z)] \geq \mathbb{E}_{p(s'|z)}[\log q_\psi(s'|z)]$. Substituting this yields:

$$I(Z;S') \geq \mathbb{E}_{p(s',z)}[\log q_\psi(s'|z)] + H(S') \quad (28)$$

The entropy term $H(S')$ depends only on the environment dynamics and is constant with respect to our optimization parameters. Thus, it can be ignored during training. We can rewrite the joint distribution $p(s',z)$ by marginalizing over the dataset $\mathcal{D}$. Leveraging the causal diagram that S' and Z are conditionally independent given (S,A), we have $p(s,a,s',z) = p(s,a)p(s'|s,a)p_\vartheta(z|s,a)$. This yields the standard lower bound of the mutual information between Z and S':

$$I(Z;S') \geq \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\mathbb{E}_{z\sim p_\vartheta(\cdot|s,a)}[\log q_\psi(s'|z)]\right] + \text{const.} \quad (29)$$

Complexity Upper Bound for $I(Z;S,A)$. Next, we establish a tractable upper bound for the complexity penalty $I(Z;S,A)$. To introduce a state belief, we can decompose this term using the chain rule of mutual information:

$$I(Z;S,A) = I(Z;S) + I(Z;A|S) \quad (30)$$

Let $p_{\pi_\beta}(z|s) = \int p_\vartheta(z|s,a)\pi_\beta(a|s)da$ be the true state belief (state-conditional marginal representation) under the behavior policy π_β, and $p(z) = \int p_{\pi_\beta}(z|s)p(s)ds$ be the fully marginalized prior.

We can expand the mutual information as the expected KL divergence:

$$\begin{aligned} I(Z;S,A) &= \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p(z))\right] \\ &= \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\int p_\vartheta(z|s,a)\log\left(\frac{p_\vartheta(z|s,a)}{p_{\pi_\beta}(z|s)}\cdot\frac{p_{\pi_\beta}(z|s)}{p(z)}\right)dz\right] \\ &= \underbrace{\mathbb{E}_{(s,a)\sim\mathcal{D}}\left[D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_{\pi_\beta}(z|s))\right]}_{(1)\ I(Z;A|S)} + \underbrace{\mathbb{E}_{s\sim\mathcal{D}}\left[D_{\mathrm{KL}}(p_{\pi_\beta}(z|s)\|p(z))\right]}_{(2)\ I(Z;S)} \end{aligned} \tag{31}$$

Because both $p_{\pi_\beta}(z|s)$ and $p(z)$ are computationally intractable, we introduce variational approximations: a learned state belief network $p_\varphi(z|s)$ to approximate $p_{\pi_\beta}(z|s)$, and a fixed standard prior $p(z)$ (typically $\mathcal{N}(0,I)$).

To bound term (1) $I(Z;A|S)$, we introduce the state belief network $p_\varphi(z|s)$ and rewrite the term using the non-negativity property of the KL divergence $D_{\mathrm{KL}}(p_{\pi_\beta}\|p_\varphi)\geq 0$. This gives us a rigorous upper bound:

$$\begin{aligned} I(Z;A|S) &= \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\int p_\vartheta(z|s,a)\log\frac{p_\vartheta(z|s,a)}{p_\varphi(z|s)}dz - \int p_\vartheta(z|s,a)\log\frac{p_{\pi_\beta}(z|s)}{p_\varphi(z|s)}dz\right] && (32)\\ &= \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s))\right] - \mathbb{E}_{s\sim\mathcal{D}}\left[D_{\mathrm{KL}}(p_{\pi_\beta}(z|s)\|p_\varphi(z|s))\right] && (33)\\ &\leq \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s))\right] && (34) \end{aligned}$$

To address term (2) $I(Z;S)$, replacing the true marginal $p(z)$ with a prior $\mathcal{N}(0,I)$ yields a valid upper bound $\mathbb{E}_{s\sim\mathcal{D}}\left[D_{\mathrm{KL}}(p_{\pi_\beta}(z|s)\|p(z))\right]$, since the true marginal minimizes this KL divergence:

$$I(Z;S)\leq \mathbb{E}_{s\sim\mathcal{D}}\left[D_{\mathrm{KL}}(p_{\pi_\beta}(z|s)\|p(z))\right]. \tag{35}$$

However, the $p_{\pi_\beta}(z|s)$ remains intractable. Fortunately, as indicated by Eq. (34), optimizing the upper bound for $I(Z;A|S)$ implicitly drives $p_\varphi(z|s)$ to be a tightly optimized variational approximation of $p_{\pi_\beta}(z|s)$: $D_{\mathrm{KL}}(p_{\pi_\beta}(z|s)\|p_\varphi(z|s))\leq \mathbb{E}_{a\sim\pi_\beta(a|s)}\left[D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s))\right]$ (due to the convexity of the KL divergence). Substituting the intractable marginal $p_{\pi_\beta}(z|s)$ with the learned surrogate $p_\varphi(z|s)$ breaks the strict analytical inequality, meaning the resulting formula acts as a heuristic approximation rather than a rigorous bound. However, it remains a principled and widely adopted variational technique in practice to achieve tractability. Consequently, we formulate the following surrogate objective for term (2): $\mathcal{L}_{I(Z;S)}=\mathbb{E}_{s\sim\mathcal{D}}\left[D_{\mathrm{KL}}(p_\varphi(z|s)\|p(z))\right]$.

Final Objective. Combining the rigorous lower bound for $I(Z;S')$, the rigorous upper bound for $I(Z;A|S)$, and the tractable variational approximation for $I(Z;S)$, the original IB objective is approximated by our final empirical objective $\mathcal{L}_{\mathrm{SBGF}}$:

$$\begin{aligned} \mathcal{L}_{\mathrm{SBGF}}(\psi,\vartheta,\varphi) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}\Big[&\mathbb{E}_{z\sim p_\vartheta(\cdot|s,a)}[-\log q_\psi(s'|z)] \\ &+\beta\big(D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s)) + D_{\mathrm{KL}}(p_\varphi(z|s)\|p(z))\big)\Big], \end{aligned} \tag{36}$$

where the latent variable z can be sampled using the reparameterization trick [14]. We use an encoder of the form $p_\vartheta(z|s,a)=\mathcal{N}(\mu_\vartheta,\sigma_\vartheta|s,a)$ and state belief $p_\varphi(z|s)=\mathcal{N}(\mu_\varphi,\sigma_\varphi|s)$; the gradients can be easily backpropagated through the network parameters ψ,ϑ, and φ. Specifically, the two KL divergences have the following closed-form solutions:

$$D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s)) = -\frac{1}{2}(1+\log\frac{\sigma_\vartheta^2}{\sigma_\varphi^2}-\frac{\sigma_\vartheta^2+(\mu_\vartheta-\mu_\varphi)^2}{\sigma_\varphi^2}) \tag{37}$$

$$D_{\mathrm{KL}}(p_\varphi(z|s)\|p(z)) = -\frac{1}{2}(1+\log\sigma_\varphi^2-\mu_\varphi^2-\sigma_\varphi^2) \tag{38}$$

This concludes the proof. □

B.3 Proof of Theorem 1

Theorem 1 *(Surrogate Discrepancy Bound). Let $\hat{p}_\pi(z|s)=\mathbb{E}_{a\sim\pi(\cdot|s)}p_\vartheta(z|s,a)$ for any given policy π, and let $\hat{p}_{\pi_\beta}(z|s)=p_\varphi(z|s)$ be the models estimated by SBGF in Eq.* (8) *on an offline dataset $\mathcal{D}$ of*

size N. With probability at least $1-\delta$, the following bound holds for the discrepancy between the true consequence divergence and the empirical surrogate divergence:

$$\left|\mathbb{E}_{s\sim d^{\pi_\beta}}\left[D_{\mathrm{KL}}(P_\pi(s'|s)\|P_{\pi_\beta}(s'|s))\right]-\mathbb{E}_{s\sim\mathcal{D}}\left[D_{\mathrm{KL}}(\hat{p}_\pi(z|s)\|\hat{p}_{\pi_\beta}(z|s))\right]\right| \tag{39}$$

$$\leq \underbrace{\mathbb{E}_{s,s'}\left[D_{\mathrm{KL}}(p_\pi(z|s,s')\|p_{\pi_\beta}(z|s,s'))\right]}_{\substack{\textbf{(I) Dynamics Info Compression (Bias)}\\ \textbf{(Increases as } \beta\uparrow\textbf{)}}} + \underbrace{\mathcal{O}\left(\sqrt{\frac{\mathrm{Var}_s(D_{\mathrm{KL}}(p_\pi(z|s)\|p_{\pi_\beta}(z|s))\log(1/\delta)}{N}}\right)}_{\substack{\textbf{(II) Estimation Error (Variance)}\\ \textbf{(Decreases as } \beta\uparrow\textbf{)}}} + \underbrace{C_\pi\epsilon_{\mathrm{opt},\beta}}_{\textbf{(III) Policy Shift}} \tag{40}$$

where the policy ratio in the latent space is bounded by $C_\pi = \max_{s\in\mathcal{D},z}\frac{p_\pi(z|s)}{p_{\pi_\beta}(z|s)}$, and $\epsilon_{\mathrm{opt},\beta}$ is the optimization error related to the bidirectional action IB loss in Eq. (8).

Proof. The goal of our analysis is to bound the discrepancy between our learned empirical surrogate latent divergence and the true consequence divergence.

We begin by simplifying the notation for the divergence metrics involved in our analysis:

- True Consequence Divergence $D_{s'} := \mathbb{E}_{s\sim d^{\pi_\beta}}\left[D_{\mathrm{KL}}(P_\pi(s'|s)\|P_{\pi_\beta}(s'|s))\right]$, representing the expected divergence between consequence distributions across all states.
- Theoretical Latent Divergence $D_z := \mathbb{E}_{s\sim d^{\pi_\beta}}\left[D_{\mathrm{KL}}(p_\pi(z|s)\|p_{\pi_\beta}(z|s))\right]$. Here, the two distributions represent the theoretical consequence beliefs based on the encoder for the learned policy and the behavior policy, respectively: $p_{\pi_\beta}(z|s) := \mathbb{E}_{a\sim\pi_\beta(\cdot|s)}[p_\vartheta(z|s,a)]$ and $p_\pi(z|s) := \mathbb{E}_{a\sim\pi(\cdot|s)}[p_\vartheta(z|s,a)]$.
- Empirical Latent Divergence $\hat{D}_z := \mathbb{E}_{s\sim\hat{d}^{\pi_\beta}}\left[D_{\mathrm{KL}}(\hat{p}_\pi(z|s)\|\hat{p}_{\pi_\beta}(z|s))\right]$, where $\hat{d}^{\pi_\beta}(s)$ is the empirical state distribution defined by the offline dataset $\mathcal{D}$ of size N. The two empirical distributions are based on our encoder and the learned state belief: $\hat{p}_\pi(z|s) := \int p_\vartheta(z|s,a)\pi(a|s)\,da$ and $\hat{p}_{\pi_\beta}(z|s) := p_\varphi(z|s)$.

Our first goal is to bound the discrepancy between the intractable objective $D_{s'}$ and the objective optimized in practice $\hat{D}_z$. By introducing the theoretical latent divergence D_z and applying the triangle inequality, we decompose the total error into a theoretical gap and an empirical error:

$$|D_{s'}-\hat{D}_z| = |D_{s'}-D_z+D_z-\hat{D}_z| \tag{41}$$

$$\leq \underbrace{|D_{s'}-D_z|}_{\text{Theoretical Gap}} + \underbrace{|D_z-\hat{D}_z|}_{\text{Empirical Error}}. \tag{42}$$

We proceed to analyze these terms individually.

For the Theoretical Gap. Taking the expectation over $s\sim d^{\pi_\beta}(s)$ and applying the exact gap derived in Eq. (18), the theoretical gap is exactly the expected posterior divergence:

$$|D_{s'}-D_z| = \mathbb{E}_{s\sim d^{\pi_\beta}(s),s'\sim P_\pi(\cdot|s)}\left[D_{\mathrm{KL}}(p_\pi(z|s,s')\|p_{\pi_\beta}(z|s,s'))\right], \tag{43}$$

where $p_\pi(z|s,s') = \frac{p(s'|z)p_\pi(z|s)}{P_\pi(s'|s)}$ represents the posterior of the latent variable z conditioned on the next state s'.

For the Empirical Error. To further simplify notation, let $D(s) := D_{\mathrm{KL}}(p_\pi(z|s)\|p_{\pi_\beta}(z|s))$ and $\hat{D}(s) := D_{\mathrm{KL}}(\hat{p}_\pi(z|s)\|\hat{p}_{\pi_\beta}(z|s))$. By applying the triangle inequality again, we separate the empirical error into a sampling error due to finite data, and an estimation error due to model approximation:

$$|D_z-\hat{D}_z| = \left|\int ds\, d^{\pi_\beta}(s)D(s) - \int ds\,\hat{d}^{\pi_\beta}(s)\hat{D}(s)\right| \tag{44}$$

$$\leq \underbrace{\left|\int ds\left(d^{\pi_\beta}(s)-\hat{d}^{\pi_\beta}(s)\right)D(s)\right|}_{\text{Sampling Error}} + \underbrace{\int ds\,\hat{d}^{\pi_\beta}(s)\left|D(s)-\hat{D}(s)\right|}_{\text{Estimation Error}}. \tag{45}$$

Bounding Sampling Error. In the continuous-control setting, we consider $D_z = \mathbb{E}_{s\sim d^{\pi_\beta}(s)}[D(s)]$ as a bounded expectation under mild encoder regularity. Assuming there exists a constant D_{max} such that $|D(s) - D_z| \leq D_{max}$ similar to [25], we can apply the Bernstein inequality (following Lemma A.3 in [32]). With probability at least $1-\delta$, the sampling error is bounded by:

$$\text{Sampling Error} \leq \sqrt{\frac{2\text{Var}_{s\sim d^{\pi_\beta}}(D(s))\log(2/\delta)}{N}} + \frac{2D_{\max}\log(2/\delta)}{3N} \tag{46}$$

For sufficiently large N, the second term decays faster, making the first term dominant. Using big-$\mathcal{O}$ notation, the bound simplifies to:

$$\text{Sampling Error} \leq \mathcal{O}\left(\sqrt{\frac{\text{Var}_s(D(s))\log(1/\delta)}{N}}\right) \tag{47}$$

Bounding Estimation Error. The estimation error relies on the capacity of the learned models. By definition above, we decompose the discrepancy as follows:

$$\begin{aligned}
\text{Estimation Error} &= \mathbb{E}_{s\sim\mathcal{D}}\left|D(s) - \hat{D}(s)\right| && (48)\\
&= \mathbb{E}_{s\sim\mathcal{D}}\left|D_{\text{KL}}(p_\pi(z|s)\|p_{\pi_\beta}(z|s)) - D_{\text{KL}}(\hat{p}_\pi(z|s)\|\hat{p}_{\pi_\beta}(z|s))\right| \\
&\overset{(a)}{=} \mathbb{E}_{s\sim\mathcal{D}}\left|\int p_\pi(z|s)\log\frac{\hat{p}_{\pi_\beta}(z|s)}{p_{\pi_\beta}(z|s)}dz\right| && (49)\\
&\leq \mathbb{E}_{s\sim\mathcal{D}}\left[\int p_\pi(z|s)\left|\log\frac{p_{\pi_\beta}(z|s)}{\hat{p}_{\pi_\beta}(z|s)}\right|dz\right] \quad \text{(Triangle Inequality)} && (50)\\
&\overset{(b)}{\leq} C_\pi\mathbb{E}_{s\sim\mathcal{D}}\left[\int p_{\pi_\beta}(z|s)\left|\log\frac{p_{\pi_\beta}(z|s)}{\hat{p}_{\pi_\beta}(z|s)}\right|dz\right] && (51)\\
&\overset{(c)}{=} C_\pi\mathbb{E}_{s\sim\mathcal{D}}\left[\int p_{\pi_\beta}(z|s)\log\frac{p_{\pi_\beta}(z|s)}{\hat{p}_{\pi_\beta}(z|s)}dz - 2\int_{p_{\pi_\beta}<\hat{p}_{\pi_\beta}} p_{\pi_\beta}(z|s)\log\frac{p_{\pi_\beta}(z|s)}{\hat{p}_{\pi_\beta}(z|s)}dz\right] && (52)\\
&\overset{(d)}{\leq} C_\pi\mathbb{E}_{s\sim\mathcal{D}}\left[D_{\text{KL}}(p_{\pi_\beta}(z|s)\|\hat{p}_{\pi_\beta}(z|s)) + \int\left|p_{\pi_\beta}(z|s) - \hat{p}_{\pi_\beta}(z|s)\right|dz\right] && (53)\\
&\leq C_\pi\mathbb{E}_{s\sim\mathcal{D}}\left[D_{\text{KL}}(p_{\pi_\beta}\|\hat{p}_{\pi_\beta}) + \sqrt{2D_{\text{KL}}(p_{\pi_\beta}\|\hat{p}_{\pi_\beta})}\right]. \quad \text{(Pinsker's inequality)} && (54)
\end{aligned}$$

For (a): Based on an encoder defined by a given β, the policy-conditioned latent distribution is strictly formed by integrating over actions: $\hat{p}_\pi(z|s) = \int p_\vartheta(z|s,a)\pi(a|s)da = p_\pi(z|s)$. Expanding the KL divergences leaves the $\log\frac{\hat{p}_{\pi_\beta}}{p_{\pi_\beta}}$.

For (b): We apply the triangle inequality to move the absolute value inside the integral. We apply importance sampling $p_\pi(z|s) = \frac{p_\pi(z|s)}{p_{\pi_\beta}(z|s)}p_{\pi_\beta}(z|s)$, and bound the density ratio in latent space with a constant $C_\pi = \max_{s\in\mathcal{D},z}\frac{p_\pi(z|s)}{p_{\pi_\beta}(z|s)}$, representing the maximum policy shift in latent space.

For (c): We rewrite the absolute log term using the identity $|x| = x - 2x\cdot\mathbb{I}_{\{x<0\}}$.

For (d): The first integral is exactly the definition of KL divergence. For the second integral, when $p_{\pi_\beta} < \hat{p}_{\pi_\beta}$, the negative log inequality $-\log(p/q) \leq (q/p) - 1$ implies that $-2\int_{p<q} p\log(p/q) \leq 2\int_{p<q}(q-p)$. Using a standard property of probability distributions $2\int_{p<q}(q-p) = \int|p-q|dz$ yields the absolute difference integral term.

Recalling that $\hat{p}_{\pi_\beta}(z|s) := p_\varphi(z|s)$, we introduce the optimal projection $p^\star_\varphi(z|s)$, defined as $\varphi^\star = \arg\min_\varphi \mathbb{E}_s[D_{KL}(p_{\pi_\beta}|p_\varphi)]$, we can decompose the error into two distinct components:

$$\begin{aligned}
&\mathbb{E}_{s\sim\mathcal{D}}[D_{KL}(p_{\pi_\beta}(z|s)\|p_\varphi(z|s))] && (55)\\
=&\underbrace{\mathbb{E}_{s\sim\mathcal{D}}[D_{KL}(p_{\pi_\beta}(z|s)\|p_{\varphi^\star}(z|s))]}_{\text{realizability error }(\epsilon_\beta)} + \underbrace{\mathbb{E}_{s\sim\mathcal{D}}[\mathbb{E}_{z\sim p_{\pi_\beta}(\cdot|s)}[\log p_{\varphi^\star}(z|s) - \log p_\varphi(z|s)]]}_{\text{optimization error }(\epsilon_{\text{opt}})} && (56)
\end{aligned}$$

The first term explicitly captures the realizability gap (ϵ_β). This represents the fundamental approximation limit of fitting a single Gaussian to a potentially multimodal behavior-induced latent distribution. This term depends on β and vanishes only when β is sufficiently large, causing p_ϑ to collapse to a single Gaussian distribution across all actions. The second term now cleanly isolates the true optimization error (ϵ_{opt}), which stems strictly from the model training process.

We use Jensen's Inequality on concave function $\sqrt{\cdot}$ and simplify notation of the estimation error bound as:

$$\text{Estimation Error} \leq C_\pi \left((\epsilon_\beta + \epsilon_{\text{opt}}) + \sqrt{2(\epsilon_\beta + \epsilon_{\text{opt}})} \right) := C_\pi \epsilon_{\text{opt},\beta} \tag{57}$$

Combining all components yields the final bound:

$$\left| \mathbb{E}_{s\sim d^{\pi_\beta}} \left[D_{\text{KL}}(P_\pi(s'|s) \| P_{\pi_\beta}(s'|s)) \right] - \mathbb{E}_{s\sim \hat{d}^{\pi_\beta}(s)} \left[D_{\text{KL}}(\mathbb{E}_{a\sim\pi(\cdot|s)} p_\vartheta(z|s,a) \| p_\varphi(z|s)) \right] \right| \tag{58}$$

$$\leq \mathbb{E}_{s,s'} \left[D_{\text{KL}}(p_\pi(z|s,s') \| p_{\pi_\beta}(z|s,s')) \right] + \mathcal{O}\left(\sqrt{\frac{\text{Var}_s(D_{\text{KL}}(p_\pi(z|s) \| p_{\pi_\beta}(z|s)) \log(1/\delta)}{N}} \right)$$

$$+ C_\pi \epsilon_{\text{opt},\beta} \tag{59}$$

□

Remark *(Validating the Bias-Variance Trade-off (β)).* Theorem 1 reveals that our surrogate approximation is systematically governed by a fundamental Bias-Variance trade-off dictated by the information compression parameter β, guiding the representation learning in the causal diagram $s, a \to z \to s'$.

- **Term (I) acts as the Bias.** It measures the hidden divergence of the *policy latent consequences* conditioned on the *same physical outcome* s'. When β is sufficiently small, the model primarily maximizes the predictive information $I(Z; S')$, making the transition mapping $z \to s'$ near-deterministic. Consequently, given the start state s and the end state s', the intermediate latent z is almost uniquely determined. In this regime, the posterior $p_\pi(z|s, s')$ becomes largely independent of the specific policy, keeping Term (I) close to zero. However, as β moderately increases, the information bottleneck introduces stochasticity and compresses out action-specific details. Multiple different latent representations z (arising from different actions) might now map to the same physical outcome s'. Consequently, given (s, s'), the inferred z relies more heavily on the underlying policy's action distribution, which inflates the divergence between $p_\pi(z|s, s')$ and $p_{\pi_\beta}(z|s, s')$. *(While an excessively large β would force z to collapse entirely to the prior, trivially erasing policy differences and bringing Term (I) back to zero, such a heavily regularized model loses all dynamics prediction capability. Thus, the meaningful bias-variance trade-off occurs in the regime of relatively small β.)*
- **Term (II) acts as the Variance.** It captures the fluctuation of the latent policy discrepancy across different states. A large β acts as a strong regularizer, heavily penalizing the action information a in z. This smooths out the representational differences between π and π_β, effectively suppressing the cross-state variance. A small β preserves action details, increasing this variance.
- **Term (III) highlights the OOD Penalty.** C_π represents the coverage coefficient of the evaluated policy π relative to the behavior policy π_β in the latent space. If an OOD action maps far away from the support of $p_{\pi_\beta}(z|s)$, C_π can become unbounded. In such cases, Theorem 1's bound fails, and the surrogate metric becomes ineffective at guaranteeing the consequence distance. This establishes the theoretical necessity of our proposed regularizer. By explicitly constraining the learned policy π to stay within the latent support of π_β, the regularizer ensures that C_π remains bounded. As long as OOD behavior in the latent space is prevented, the surrogate's approximation of the consequence divergence is theoretically guaranteed.

Guideline for tuning β: This theoretical trade-off provides a principled guideline for environmental adaptation. In highly stochastic environments, the inherent variance of Term (II) dominates, necessitating a larger β to curb Term (II). In near-deterministic environments, a smaller β is optimal to strictly preserve the transition dynamics and minimize the bias of Term (I).

B.4 Proof Sketch of Theorem 1: Implementation for Policy Optimization

In practical policy optimization, directly minimizing the empirical latent divergence $D_{\mathrm{KL}}(\hat{p}_\pi(z|s)\|\hat{p}_{\pi_\beta}(z|s)) = D_{\mathrm{KL}}(\mathbb{E}_{a\sim\pi(\cdot|s)}[p_\vartheta(z|s,a)]\|p_\varphi(z|s))$ is computationally intractable. The optimization involves nested Monte Carlo sampling, which is expensive and prone to biased estimation. To address this, we instead optimize a tractable point-wise upper bound. By expanding the divergence, we have:

$$\mathbb{E}_{a\sim\pi(\cdot|s)}\big[D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s))\big] \tag{60}$$

$$= \int_a \pi(a|s) \int_z p_\vartheta(z|s,a) \log \frac{p_\vartheta(z|s,a)}{p_\varphi(z|s)}\, dz\, da \tag{61}$$

$$= \int_a \int_z \pi(a|s) p_\vartheta(z|s,a) \log \left(\frac{p_\vartheta(z|s,a)}{\hat{p}_\pi(z|s)} \cdot \frac{\hat{p}_\pi(z|s)}{p_\varphi(z|s)} \right) dz\, da \tag{62}$$

$$= \int_a \int_z \pi(a|s) p_\vartheta(z|s,a) \log \frac{p_\vartheta(z|s,a)}{\hat{p}_\pi(z|s)} dz\, da + \int_a \int_z \pi(a|s) p_\vartheta(z|s,a) \log \frac{\hat{p}_\pi(z|s)}{p_\varphi(z|s)} dz\, da \tag{63}$$

$$= I_\pi(A; Z|s) + D_{\mathrm{KL}}\big(\hat{p}_\pi(z|s)\|p_\varphi(z|s)\big) \tag{64}$$

where $I_\pi(A; Z|s)$ denotes the mutual information between the action a and the latent consequence z induced by the policy π conditioned on state s. Since mutual information is strictly non-negative, the left-hand side serves as a rigorous upper bound to the empirical latent divergence. By minimizing this upper bound, we can fuse the latent constraint with the Monte Carlo sampling of the Q-value, resulting in our final, highly efficient optimization objective in Eq. (12).

Crucially, utilizing this upper bound is not merely a computational compromise, but a highly desirable mechanism. The gap between the original marginal KL and our surrogate is exactly the mutual information penalty $I_\pi(A; Z|S)$:

$$|D_{s'} - \hat{D}_{po}| \le |D_{s'} - \hat{D}_z| + |\hat{D}_z - \hat{D}_{po}| \quad \text{(Triangle Inequality)} \tag{65}$$

$$= |D_{s'} - \hat{D}_z| + I_\pi(A; Z|S) \quad \text{(Eq. (64))} \tag{66}$$

where $\hat{D}_{po} := \mathbb{E}_{s\sim\hat{d}^{\pi_\beta}(s), a\sim\pi(\cdot|s)}\big[D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s))\big]$.

This yields the bound:

$$\Big|\mathbb{E}_{s\sim d^{\pi_\beta}}\big[D_{\mathrm{KL}}(P_\pi(s'|s)\|P_{\pi_\beta}(s'|s))\big] - \mathbb{E}_{s\sim\hat{d}^{\pi_\beta}(s), a\sim\pi(\cdot|s)}\big[D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s))\big]\Big| \tag{67}$$

$$\le \mathbb{E}_{s,s'}\big[D_{\mathrm{KL}}(p_\pi(z|s,s')\|p_{\pi_\beta}(z|s,s'))\big] + \mathcal{O}\left(\sqrt{\frac{\mathrm{Var}_s(D_{\mathrm{KL}}(p_\pi(z|s)\|p_{\pi_\beta}(z|s))\log(1/\delta)}{N}}\right)$$

$$+ C_\pi \epsilon_{\mathrm{opt},\beta} + I_\pi(A; Z|S) \tag{68}$$

Minimizing the mutual information penalty $I_\pi(A; Z|S)$ implicitly enforces action–consequence consistency, thereby resolving a critical risk-dilution issue. Specifically, solely matching marginal distributions allows dangerous OOD actions to be masked by expectation averaging, rendering the aggregate distribution $\hat{p}_\pi(z|s)$ deceptively safe. In contrast, penalizing this term enforces a point-wise constraint: every individually sampled action must strictly align with the safe behavior distribution, fundamentally preventing catastrophic deviations.

However, tightening this point-wise KL upper bound introduces a structural trade-off: On the one hand, it significantly benefits tasks where the optimal policy is near-deterministic. As illustrated in Figure 5, stochastic policies often assign non-negligible probability mass to sub-optimal actions, leading to inferior performance compared to deterministic policies. On the other hand, excessively penalizing policy-conditioned divergence risks entropy collapse by overly penalizing stochasticity, which may induce premature convergence and severe overfitting. Consequently, maintaining necessary policy stochasticity requires incorporating explicit entropy regularization.

C Implementation Details and Extended Experiments

C.1 Benchmark Experiments

Data. We use the datasets from the D4RL benchmark [6] of the latest versions, which are v2 for MuJoCo, AntMaze and v1 for Adroit. We provide an explanation of the abbreviations used in the Figure 5. Mujoco Tasks: Hc = halfcheetah, H = hopper, W = walker2d, m = medium, mr = medium-replay, me = medium-expert. AntMaze Tasks: u = AntMaze-umaze, u-d = AntMaze-umaze-diverse, m-p = AntMaze-medium-play, m-d = AntMaze-medium-diverse, l-p = AntMaze-large-play, l-d = AntMaze-large-diverse. Adroit Tasks: h = human, hamm = hammer.

Baselines. We report the baseline results from their original papers, and reproduce the missing experiments using the official code:

- TD3BC [7]: `https://github.com/sfujim/TD3_BC`
- SPOT [31]: `https://github.com/thuml/SPOT`
- FQL [23]: `https://github.com/seohongpark/fql`

Implementation details. We normalize the states in the dataset for MuJoCo and Adroit but do not normalize the states for AntMaze. Following CQL [17], we subtract 1 from rewards for the AntMaze datasets. Hyperparameters used are presented in Table 2. For policy evaluation, we average mean returns over 10 evaluation trajectories on the MuJoCo and Adroit tasks, and average over 100 evaluation trajectories on the AntMaze tasks. For stochastic policies implementation in Section 5.4, we directly replace the flow matching with our latent divergence. Our code is available at: `https://github.com/DarrenJum/StBR`.

To prevent the Q-value gradients from dominating the KL regularization term, we adopt the normalization strategy like most policy constraint methods [7, 31, 36]. We scale the Q-values by an absolute average over the current minibatch: $\mathcal{L}_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\phi(\cdot|s)}\left[-\frac{Q_\theta(s,a)}{\alpha} + \lambda D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s))\right]$, where the scaling factor $\alpha = \frac{1}{B}\sum_{i=1}^{B}|Q_\theta(s_i,a_i)|$ is dynamically computed from a minibatch of B transitions, ensuring stable optimization across different tasks.

Explicit Next-State Prediction (ENSP) Baseline. To demonstrate the advantage of constraining policy updates in the latent space rather than the raw observation space, we construct an explicit next-state prediction baseline. Instead of matching probability distributions via KL divergence, this baseline directly penalizes deviations in the state space using an explicit deterministic dynamics model $\hat{s}' = \hat{P}(s,a)$, pre-trained on transitions $(s,a,s') \sim \mathcal{D}$ via mean squared error. During policy optimization, we regularize the policy by pulling the predicted next state under the policy action toward the empirical transition target s' recorded in the dataset:

$$\max_{\pi_\phi} \mathbb{E}_{(s,s')\sim\mathcal{D},a\sim\pi_\phi(\cdot|s)}\left[\frac{Q_\theta(s,a)}{\alpha} - \lambda_{ENSP}\left\|\hat{P}(s,a) - s'\right\|_2^2\right], \tag{69}$$

where α is the batch-adaptive scaling factor defined as above.

Algorithm 1 Structural Behavior Regularization (StBR)

Input: The offline dataset $\mathcal{D} = \{(s,a,r,s')\}$

1: Initialize SBGF with parameters ψ, ϑ and φ, policy network π_ϕ, critic network Q_θ and target network $Q_{\bar{\theta}}$ with $\bar{\theta} \leftarrow \theta$
2: **for** $t = 1$ to T **do**
3: Sample mini-batch of transitions $(s,a,r,s') \sim \mathcal{D}$
4: Update θ minimizing $\mathcal{L}_Q(\theta)$ in Eq. (1)
5: Update ϕ minimizing $\mathcal{L}_\pi(\phi)$ in Eq. (12)
6: Update ψ, ϑ, φ minimizing $\mathcal{L}_{\mathrm{SBGF}}(\psi,\vartheta,\varphi)$ in Eq. (8)
7: Update target network: $\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}$
8: **end for**

Table 2: Hyperparameters in StBR

	Hyperparameter	Value
StBR	Optimizer	Adam
	Activation function	ReLU
	Critic learning rate	3×10^{-4}
	Actor learning rate	3×10^{-4}
	Batch size	256
	Discount factor	0.99
	Gradient Steps	1×10^{6}
	SBGF learning rate	1×10^{-3}
	Target network update rate	0.005
	Policy update frequency	2
	Number of Critics	4 (including 2 target networks)
	β	0.001 for MuJoCo, 0.01 for AntMaze and Adroit
	λ	{0.03,0.05,0.1} (Detail for each task see Table 3)
Architecture	Actor	input-256-256-action dim
	Critic	input-256-256-1
	Encoder $p_\vartheta(z\|s,a)$	input-750-750-1.2×state dim
	Encoder $p_\varphi(z\|s)$	input-750-750-1.2×state dim
	Decoder $q_\psi(s'\|z)$	input-750-750-state dim

Table 3: Detailed configurations of λ and β across different tasks.

Task Name		λ	β	λ_{ENSP}
halfcheetah	medium	0.03	0.001	0.05
	medium-replay	0.03	0.001	0.05
	medium-expert	0.05	0.001	0.2
hopper	medium	0.03	0.001	0.1
	medium-replay	0.05	0.001	0.05
	medium-expert	0.05	0.001	0.2
walker2d	medium	0.1	0.001	0.2
	medium-replay	0.1	0.001	0.2
	medium-expert	0.1	0.001	0.2
antmaze	umaze	0.1	0.01	0.1
	umaze-diverse	0.05	0.01	0.1
	medium-play	0.05	0.01	0.05
	medium-diverse	0.05	0.01	0.05
	large-play	0.05	0.01	0.05
	large-diverse	0.05	0.01	0.05
pen	human	0.03	0.01	0.05
hammer	human	0.03	0.01	0.1
door	human	0.03	0.01	0.2

Hyperparameter tuning and Learning curves. Approaches to offline policy evaluation and selection have garnered significant attention within the research community. As discussed by Brandfonbrener *et al.* [3], Wu *et al.* [31], assuming limited access to a real system to test a small number of trained policies constitutes a reasonable setup for practical applications such as robotics.

In StBR, the coefficient λ in Eq. (12) is a crucial hyperparameter that requires tuning. Following prior works [31, 16, 30], we allow limited access to the online environment to tune this parameter across a minimal candidate set of $\{0.03, 0.05, 0.1\}$. To ensure a rigorous evaluation, we tune λ using a single random seed. Once the optimal hyperparameter is identified, we train the policy from scratch using five entirely new seeds. The final reported results are based exclusively on these five evaluation seeds, omitting the initial tuning seed. Note that the Adroit Human domain presents a challenging setting characterized by high-dimensional spaces and limited dataset sizes (only ∼10,000 transitions). This leads to severe value overestimation, necessitating a stronger degree of conservatism.

To facilitate hyperparameter tuning, we specifically employ the following policy loss for this task: $\mathcal{L}_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D},a\sim\pi_\phi(\cdot|s)}\left[-\lambda\frac{Q_\theta(s,a)}{\alpha} + (1-\lambda)D_{\mathrm{KL}}(p_\vartheta(z|s,a)\|p_\varphi(z|s))\right]$.

Figures 6 and 7 present the training curves for all tasks in the benchmark experiments. The solid lines depict the mean scores across the five evaluation seeds, while the shaded regions indicate the minimum and maximum values. A natural question that arises is *how latent divergence performs when applied solely to BC*. To address this, these figures also include the training curves for this specific configuration, denoted as StBR+BC.

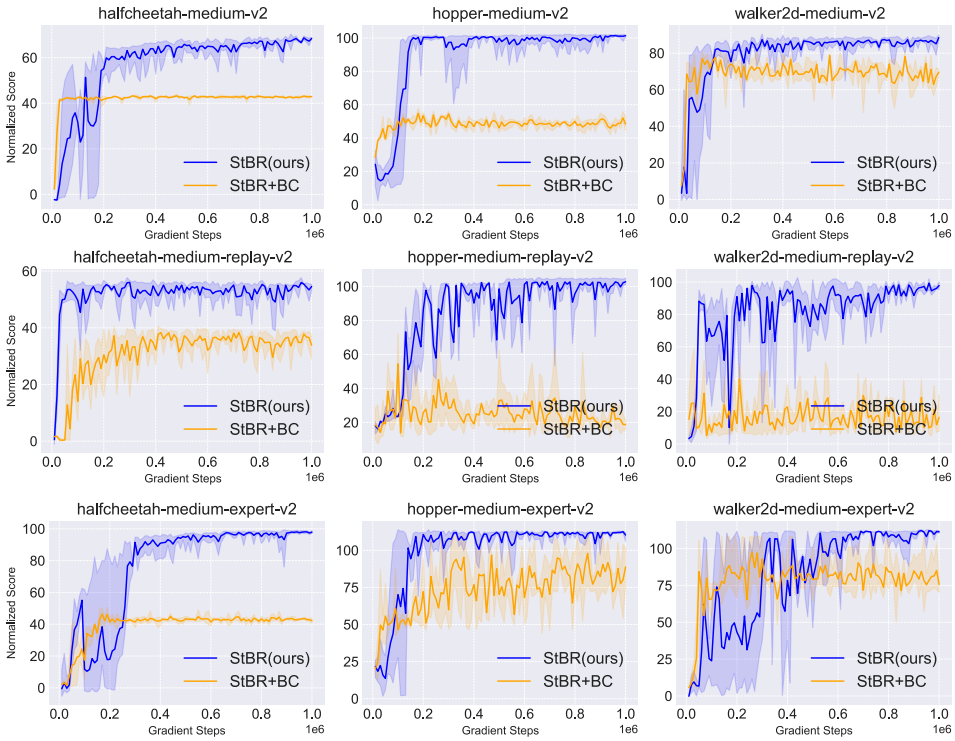

Figure 6: The results of StBR and StBR+BC on the MuJoCo task. The shaded area indicates the minimum and maximum score across 5 seeds, with the solid line depicting their mean.

C.2 Theoretical Validation Experiments

Design of environment stochasticity experiments We evaluate StBR on the `hopper-medium` task with artificially injected action execution stochasticity. To simulate transition stochasticity in real-world environments, we introduce action noise during both the offline data collection and online interaction phases as simulated motor errors. Specifically, we add Gaussian noise to the agent's intended action to emulate the control errors inherent in physical actuators $\tilde{a} \sim \mathcal{N}(a, \sigma^2_{env})$. From the agent's perspective, this discrepancy between the intended action and the executed action leads to unpredictable next states, thereby naturally transforming the deterministic Hopper simulator into a stochastic environment.

Visualization of OOD action separation. Furthermore, we evaluate whether StBR effectively shapes the divergence landscape to prevent the agent from taking OOD actions. Figure 8 tracks the latent divergence throughout the training process for both OOD actions ($a \sim \mathrm{Unif}(\mathcal{A})$) and learned policy's actions ($a \sim \pi_\phi$) in `hopper-medium` and `pen-human` tasks. As illustrated in Figure 8, the latent divergence for OOD actions remains substantially larger than that of the policy-constrained actions. This confirms that StBR successfully penalizes minor action deviations that could lead to drastically different, unknown transitions, thereby robustly keeping the policy within the offline dataset.

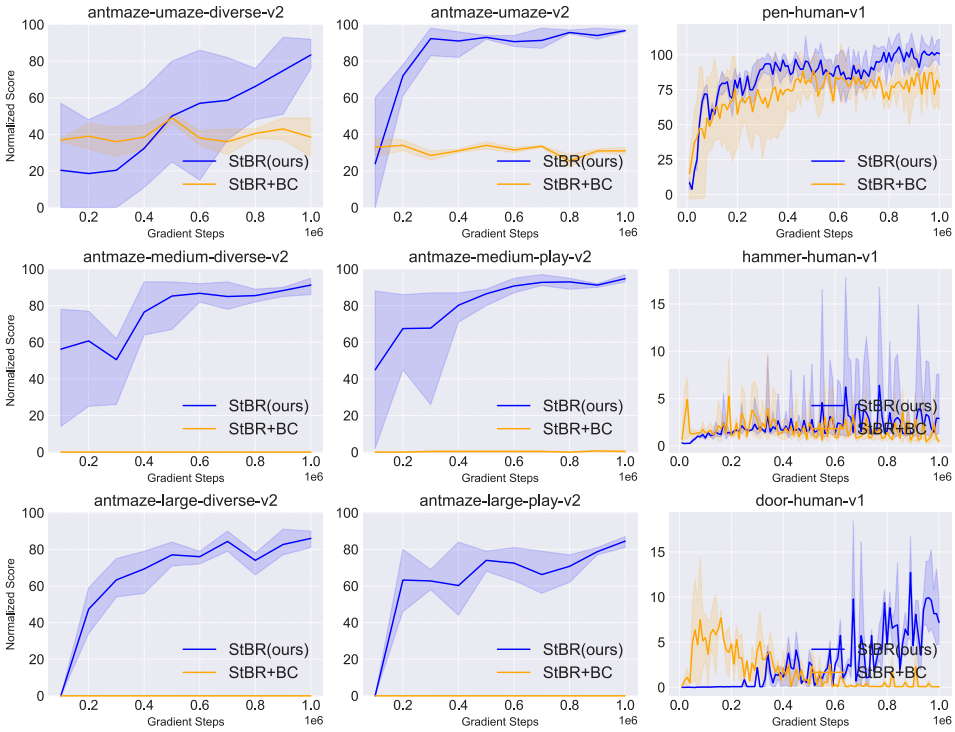

Figure 7: The results of StBR and StBR+BC on the AntMaze and Adroit task. The shaded area indicates the minimum and maximum score across 5 seeds, with the solid line depicting their mean.

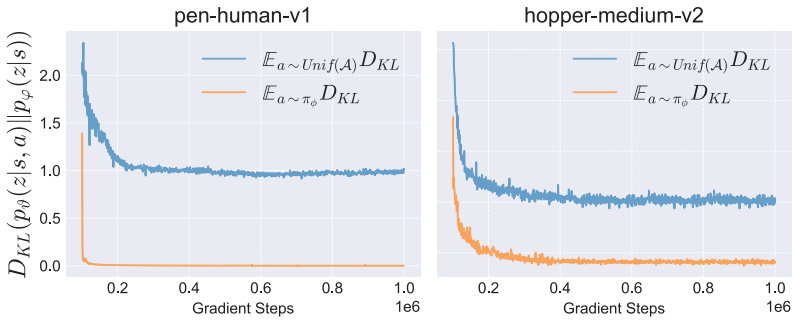

Figure 8: The figure tracks the latent divergence across gradient steps for OOD actions and learned policy's actions. Notably, the latent divergence of the constrained policy is significantly lower than that of the OOD actions, demonstrating that our latent divergence effectively separates in-distribution (ID) from OOD actions.

C.3 Additional Experimental Results

To comprehensively validate the capabilities and practical advantages of StBR beyond standard benchmark evaluations, we design three targeted sets of supplementary experiments. Specifically, we aim to answer the following questions: (1) How does StBR perform when high-quality data are scarce? (2) Does the consequence-aware nature of StBR practically enable out-of-sample (OOS) state recovery ability? and (3) How does StBR differ from model-based approaches when evaluating its learned decoder?

Robustness to limited valuable data. To investigate our StBR's performance on suboptimal datasets, we design an experiment to quantitatively evaluate scenarios with limited valuable data. Specifically, we construct mixed datasets using `walker2d-random` and `walker2d-expert`. By

progressively reducing the proportion of expert data from 50% down to 10%, we simulate scenarios where high-quality transitions are extremely scarce.

As shown in Table 4, severely limiting the expert data (e.g., down to 10%) causes traditional methods (CQL, SDC) to experience a performance collapse, largely due to over-conservatism affected by suboptimal transitions. While the advanced FQL achieves stronger robustness through more expressive modeling of the behavior policy, StBR consistently outperforms all these methods across all levels. This superior robustness is directly enabled by our consequence-aware support constraint.

Table 4: Normalized scores on mixed walker2d datasets with varying proportions of random (r) and expert (e) transitions. Results are averaged over four seeds. StBR demonstrates significant robustness against limited valuable data dilution.

Limited Valuable Dataset	CQL	SDC	FQL	StBR
Walker2d-r0.5-e0.5	88.3±4.9	88.8±3.9	108.9±4.5	**113.9**±3.1
Walker2d-r0.6-e0.4	81.5±6.3	65.9±5.0	91.7±6.6	**106.1**±2.4
Walker2d-r0.7-e0.3	48.3±5.8	85.5±4.7	88.7±9.1	**99.9**±8.1
Walker2d-r0.8-e0.2	50.5±7.2	49.8±5.6	70.4±11.5	**87.7**±5.5
Walker2d-r0.9-e0.1	17.6±3.0	26.8±2.1	66.7±23.6	**73.4**±22.9

Dataset-size ablation. We evaluated StBR on the hopper-medium dataset under varying data sizes, with the mean and standard deviation reported across 5 seeds. We randomly sampled trajectory subsets ranging from 1% to 100% of the offline dataset for standard offline RL training. As motivated by our theoretical analysis, a larger β is beneficial under data scarcity; thus, we used $\beta = 0.003$ for dataset-size below 10% and $\beta = 0.001$ for the rest. We also compared StBR against a VAE-based regularization baseline (SPOT).

The results in Table 5 demonstrate that StBR is robust to dataset size, maintaining strong performance and stability when at least 10% of the data is available. When the dataset is reduced to 5% and 1%, where the primary bottleneck becomes the quality of the randomly sampled trajectories, StBR still achieves stable learning.

Table 5: Normalized scores with varying dataset-size. Results are averaged over 5 seeds.

Dataset-size(%)	100	50	25	10	5	1
StBR(Ours)	100.9±1.0	100.4±1.2	99.5±1.5	97.2±4.1	78.2±2.7	65.6±14.5
SPOT	86.0±8.7	83.2±14.0	83.6±6.2	72.9±4.5	44.1±15.0	38.8±19.0

Out-of-sample (OOS) state recovery. We investigate whether StBR equips the agent with the ability to recover from OOS states back to the data manifold. Following the perturbation protocols established in prior OOS state recovery studies [35, 9], we conduct an OOS state recovery experiment on the AntMaze tasks. Specifically, we apply varying scales of Gaussian noise $s_{oos} \sim \mathcal{N}(s, \sigma^2_{oos})$ to the in-distribution states s sampled from the offline dataset, yielding perturbed OOS states s_{oos}. We then train the policy to pull the agent back to the behavior support from these OOS states by the following loss:

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D}}\mathbb{E}_{s_{oos}\sim\mathcal{N}(s,\sigma^2_{oos})}\big[- \mathbb{E}_{a\sim\pi_\phi(\cdot|s)}Q_\theta(s,a) + \lambda\mathbb{E}_{a'\sim\pi_\phi(\cdot|s_{oos})}D_{\mathrm{KL}}(p_\vartheta(z|s_{oos},a')\|p_\varphi(z|s_{oos}))\big]. \tag{70}$$

As illustrated in Table 6, introducing an appropriate level of OOS state perturbation (e.g., $\sigma_{oos} = 0.001$) effectively enhances the agent's generalization performance. However, when the perturbation scale is excessively large (e.g., $\sigma_{oos} = 0.1$), it completely distorts the underlying semantic structure of the states, making the OOS states unrecoverable and resulting in severe performance degradation. Overall, these results verify that StBR successfully endows the agent with remarkable state recovery capabilities, robustly navigating it back to safe regions when facing moderate distributional shifts.

Distinction from model-based methods. Since SBGF learns a decoder that can essentially be viewed as a complete dynamics model, it is natural to ask whether our approach can be explicitly

Table 6: The mean and standard deviation of averaged normalized scores after 1M steps of OOS state recovery training, averaged over 5 seeds and the final 4 evaluations. The highest scores in each task are bolded.

Task Name	$\sigma_{oos} = 0$	$\sigma_{oos} = 0.001$	$\sigma_{oos} = 0.01$	$\sigma_{oos} = 0.1$
AntMaze-medium-play	94.2±1.9	**97.1**±2.4	92.6±3.1	44.4±7.4
AntMaze-medium-diverse	90.6±3.6	**95.1**±1.8	91.0±4.5	47.4±4.8
AntMaze-large-play	84.2±2.7	**88.5**±2.7	84.5±2.2	19.6±6.2
AntMaze-large-diverse	85.8±5.1	**86.2**±3.3	83.1±6.1	12.8±7.4
Total	354.8	354.8→**366.9**	354.8→351.2	354.8→124.2

deployed as a model-based offline RL method. To investigate this, we implement an StBR+MB algorithm built upon the MOPO [34] framework. Specifically, we utilize the trained decoder as a forward model to generate rollout data and employ our latent divergence as the measure for model uncertainty with the same hyperparameter tuning protocol as model-free StBR.

The empirical results demonstrate that StBR+MB can indeed serve as an effective model-based variant, achieving performance slightly superior to MOPO. However, its overall efficacy falls short of more advanced model-based methods like RAMBO, primarily due to the inherent parameter sensitivity and the difficulty of acquiring perfectly reliable model uncertainty estimates. More importantly, our standard model-free StBR significantly outperforms all of these model-based baselines. This highlights a critical advantage: even if the learned forward model is imperfect for data generation, the latent space successfully acquires consequence-aware representations. This allows the latent divergence to provide a highly effective, robust constraint on the policy without relying on explicit rollouts, fundamentally distinguishing the mechanism of StBR from traditional model-based paradigms.

Table 7: The mean and standard deviation of averaged normalized scores after 3M steps of model-based offline RL training, averaged over 5 seeds and the final 10 evaluations. The top-2 highest scores in each task are bolded.

	Task Name	MOPO	RAMBO	StBR+MB	StBR(ours)
halfcheetah	medium	42.3±1.6	**71.0**±2.9	41.4±2.8	**60.7**±0.7
	medium-replay	53.1±2.0	**67.0**±1.5	**55.8**±3.1	54.4±1.0
	medium-expert	63.3±38.0	79.3±2.9	**90.6**±5.9	**97.8**±0.2
hopper	medium	28.0±12.4	**91.2**±16.3	80.3±7.3	**100.9**±1.0
	medium-replay	67.5±24.7	**97.6**±3.4	90.6±2.6	**100.8**±1.9
	medium-expert	23.7±6.0	**89.5**±11.1	77.8±6.5	**111.6**±0.9
walker2d	medium	17.8±19.3	**89.1**±2.7	7.8±6.0	**87.2**±0.9
	medium-replay	39.0±9.6	**88.5**±4.0	41.5±8.4	**86.3**±1.5
	medium-expert	44.6±12.9	**63.1**±31.3	46.7±23.7	**111.3**±1.4
Total		379.3	**747.5**	532.5	**811.0**

Table 8: The runtime of StBR on `hopper-medium-v2` task using a GeForce RTX 2080Ti GPU and Intel Xeon Gold 5218 CPU.

	BC	CQL	TD3BC	IQL	PBRL	POR	SDC	SPOT	FQL	StBR(ours)
Runtime (ms/step)	2.0	24.2	4.7	6.5	26.7	8.8	25.0	6.7	6.6	6.8
Wall time (minutes)	33	403	78	108	444	147	417	111	110	114
Inference time (ms)	-	-	476	-	-	909	-	-	492	476

C.4 Computation Cost

Benefiting from its pluggable design, StBR operates as a highly efficient regularization method. During inference, it requires only a single forward pass of the policy network. During the training phase, StBR introduces minimal computational overhead, necessitating only the additional forward passes of two encoders. This provides a distinct computational advantage over ensemble-based

methods, which require maintaining and querying multiple models, as well as flow-matching or diffusion models, which rely on numerous iterative forward passes.

As demonstrated in Table 8 on the `hopper-medium-v2` task, StBR maintains a wall-clock training time that is highly competitive with most standard regularization methods. Furthermore, because StBR trains a single-step policy and integrates seamlessly as a pluggable module, it achieves exceptionally fast inference speeds, making it highly practical for real-time deployment.

Table 9: Online fine-tuning results on AntMaze tasks, showing initial performance after offline RL and performance after 1M steps of online RL. All results are averaged over 5 seeds.

AntMaze	IQL	Cal-QL	StBR (Ours)
umaze	89.6 → 96.2	76.8 → **99.8**	99.2 → $99.6_{\pm 0.4}$
umaze-diverse	65.6 → 87.2	32.0 → 98.5	95.6 → **99.4**$_{\pm 0.7}$
medium-play	76.4 → 94.0	71.8 → 98.8	93.6 → **99.0**$_{\pm 0.8}$
medium-diverse	72.8 → 92.0	62.0 → 98.3	89.8 → **99.1**$_{\pm 0.7}$
large-play	42.0 → 51.0	31.8 → **97.3**	78.0 → $96.9_{\pm 1.1}$
large-diverse	46.0 → 58.4	44.0 → 91.5	77.8 → **96.5**$_{\pm 1.2}$
Average	65.4 → 79.8	53.0 → 97.3	89.0 → **98.4**

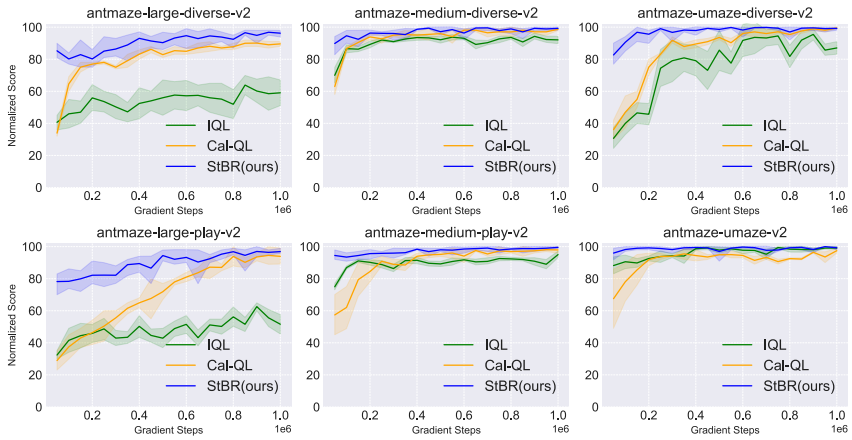

Figure 9: Online fine-tuning training curves on AntMaze tasks, showing initial performance after offline RL and performance after 1M steps of online RL. (5 seeds)

C.5 Can StBR be fine-tuned with online rollouts?

StBR naturally extends to the online fine-tuning setting. This is achieved by continuously appending new online transitions, $\mathcal{D}_{\text{online}}$, to the replay buffer while training the policy and value networks with the original offline objective.

Implementation details. We conduct our online fine-tuning experiments on the complex AntMaze domains. Due to their high-dimensional state-action spaces and sparse rewards, bootstrapping errors can easily accumulate during online exploration. To mitigate this, we retain the complete offline dataset in the replay buffer and augment it with new online data, which is actively collected using an exploration noise of 0.1. Notably, without altering the algorithmic framework, we continuously update the SBGF throughout the 1 million gradient steps of the online phase. All other hyperparameters remain identical to the baseline methods.

Comparison results. We evaluate StBR against Cal-QL [22], a performant method specifically designed for offline-to-online RL and IQL [15], a standard offline RL baseline. Table 9 summarizes the final performance, and Figure 9 plots the corresponding learning curves. The results demonstrate that the superior offline policy provided by StBR serves as a robust initialization, allowing the agent to achieve faster convergence and higher asymptotic performance compared to existing approaches.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The "abstract", "introduction" and "preliminaries and motivation" sections clearly state the problem of consequence-shift vulnerability, and the methodologies to solve this effectively. Theoretical and experimental evidences are provided to support our claims.

   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: The paper includes a dedicated "Limitations" paragraph in the Conclusion section.

   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 main paper and Appendix explicitly state the required assumptions.

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: Detailed experimental settings are referenced in the main text. These are extensively expanded upon in Appendix C, providing the necessary architectural configurations, hyperparameters, optimizers, and extensive supplementary experiments.

   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: The code were supplied to reviewers as an anonymous zip archive, and full public links will be added in the camera-ready version.

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 paper specifies baseline architectures, optimization protocols, hardware details, and hyperparameters comprehensively in Experiment section and Appendix C.

   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: Our paper reports appropriate information about statistical significance of the experiments.

   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: Appendix C details the specific compute hardware used (GeForce RTX 2080Ti GPU and Intel Xeon Gold 5218 CPU).

   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: The research involves fundamental algorithmic developments evaluated on standard, publicly available datasets (e.g., D4RL) and does not violate any ethical guidelines.

   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: [N/A]

    Justification: This paper focuses on foundational theoretical improvements to behavior regularization offline RL algorithms. It does not present direct, foreseeable negative societal impacts or obvious ties to malicious applications.

    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: The models and evaluation benchmarks used do not fall under the category of high-risk for dual-use or misuse. No scraped or sensitive data is released.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Existing datasets, platforms, and baseline models are properly credited with citations in the text and the references section.

    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: The paper does not introduce or release new datasets or proprietary assets.

    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: This research does not involve any human subjects or crowdsourcing.

    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: There is no research involving human subjects, making IRB approval unnecessary.

    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: Large language models are not part of the paper's core methods.

    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.