SELECTIVE FACTORIZED RISK REGULARIZATION FOR OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline reinforcement learning must balance policy improvement against unreliable extrapolation beyond the available data. Local action support and marginal successor-state support provide useful evidence, but do not establish whether a specific state–action–successor relation is supported or whether subsequent policy decisions will remain supported. Moreover, insufficient data support does not necessarily imply low return, motivating the use of value information when determining how support evidence should constrain learning. We propose Selective Factorized Path Risk (SFPR), a framework that regularizes policy improvement through factorized, value-guided future support costs. SFPR separately models action support conditioned on the current state and transition support conditioned on the state–action pair. Two branch-specific cost estimators recursively connect these local assessments to subsequent decisions. Data-support and behavior-relative value assessments regulate batch-level cost strengths, while support-dependent continuation reduces recursive propagation at weakly supported transitions. SFPR integrates these future costs with behavior-distribution fitting and selective value regularization to constrain both policy updates and the value estimates guiding them. SFPR achieves the highest average performance among the compared methods on both the D4RL locomotion and Maze2D navigation benchmarks. These results demonstrate SFPR's overall performance advantages across continuous-control and navigation tasks.

1 INTRODUCTION

Offline reinforcement learning (RL) uses reward information to learn policies with high long-term returns from a fixed dataset, without additional environment interaction during training (Fujimoto et al., 2019; Kumar et al., 2019; Siegel et al., 2020). However, limited coverage of states, actions, and outcomes may leave candidate decisions insufficiently supported by the available data (Chen & Jiang, 2019; Laroche et al., 2019). For out-of-distribution (OOD) actions, overestimation arising from extrapolation can bias policy updates (Fujimoto et al., 2019), while poorly covered states can undermine subsequent action selection and value estimation (Mao et al., 2024a). Without fresh environment feedback during training, these estimation errors are difficult to identify and correct. Yet generalization beyond observed actions can yield higher returns under suitable conditions (Mao et al., 2024b). The challenge is therefore to control the risks of unreliable extrapolation without unnecessarily restricting beneficial policy improvement.

Existing offline RL methods reduce reliance on unreliable extrapolation through behavioral constraints, conservative value estimation, or in-sample learning (Fujimoto et al., 2019; Kumar et al., 2019; 2020; Kostrikov et al., 2022). With finite data and function approximation, however, these training mechanisms do not automatically eliminate state distribution shift during execution, motivating explicit consideration of action consequences (Mao et al., 2024a). SCAS combines value-aware state correction with OOD action suppression, while DASP uses potential outcomes and their density to guide policies toward data-supported regions (Mao et al., 2024a; Jiang et al., 2026). DOSER uses action support together with the support and value of predicted successor states to selectively regularize OOD action values, moving beyond uniform suppression (Wang et al., 2026). Its marginal successor-state support score, however, does not by itself establish whether the corresponding state–action–successor relation is supported. Support evidence for a single decision does not establish that subsequent policy decisions will remain supported, even when the reward critic

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

estimates long-term returns. These distinctions motivate examining what action and conditional consequence evidence contribute to control, and how their use should extend to subsequent decisions.

Our perspective separates the objects of assessment, actions and their conditional consequences, from the criteria used to guide control: data support and behavior-relative value. Modeling the two conditional support relations separately makes evidence individually queryable, allowing us to examine the contribution of conditional consequences beyond action-support scores. Data compatibility alone does not determine whether a decision improves return, so relative-value judgments inform how support evidence is used rather than treating every support deficiency as harmful behavior. Current decisions also affect later states and actions; extending these assessments to future behavior therefore calls for control of both constraint strength and recursive propagation, whose estimates may be unreliable.

Building on this perspective, we propose *Selective Factorized Path Risk* (SFPR), which uses factorized support evidence and behavior-relative value assessments to guide policy improvement through selective future-cost constraints. SFPR separately models behavioral action support conditioned on the current state and transition support conditioned on the state–action pair. Branch-specific cost estimators connect these assessments to subsequent policy decisions. Behavior-relative value comparisons control the influence of the estimated costs, while support-dependent continuation limits the propagation of transition costs through poorly supported consequences. We instantiate SFPR in a Gaussian actor–critic method that combines these future-cost constraints with behavioral distribution fitting and DOSER-style selective value regularization (Wang et al., 2026). Distribution fitting grounds policy learning in recorded actions, while selective value regularization constrains reward-value estimates for selected policy candidates. Together with future-cost control, they form the policy and value optimization used by our method. We evaluate SFPR on nine D4RL locomotion datasets and three Maze2D navigation tasks. We report the Tail score, defined as the mean normalized return over the final five evaluations of a one-million-update run, to measure performance retained at the end of training. SFPR achieves the highest reported mean score across the nine locomotion datasets, reaching 94.62, and attains a Tail score of 221.72 on Maze2D-large. Our contributions are threefold:

- **Factorized conditional support evidence.** We distinguish behavioral action support given the current state from transition support given a state–action pair. This provides separately queryable evidence about actions and their conditional consequences, beyond the marginal support of predicted successor states.
- **Selective control of future support costs.** We connect the two sources of evidence to subsequent policy decisions through branch-specific recursive cost estimates. Behavior-relative value comparisons determine how strongly these costs influence policy improvement, while support-dependent continuation limits the propagation of transition costs through poorly supported consequences.
- **Empirical evaluation.** Across nine D4RL locomotion datasets, SFPR's reported mean exceeds the next-highest table entry by 1.44 normalized-score points. We also evaluate end-of-training performance on Maze2D navigation tasks and use targeted ablations to examine the roles of conditional transition evidence, selective weighting, and temporal propagation.

2 Related Work

Reliable policy improvement under limited support. Offline RL methods address unreliable extrapolation through behavioral constraints and regularized policy objectives (Fujimoto et al., 2019; Kumar et al., 2019; Fujimoto & Gu, 2021; Wu et al., 2022), conservative value estimation and diversified critic ensembles (Kumar et al., 2020; An et al., 2021), or alternative value regressions and in-sample learning (Kostrikov et al., 2022; Garg et al., 2023; Xu et al., 2023). Diffusion- and flow-based approaches further improve behavioral expressivity and the efficiency of behavior-guided policy optimization (Wang et al., 2023; Chen et al., 2024a;b; Park et al., 2025; Nguyen & Yoo, 2026). To address poorly covered states encountered during deployment, recovery methods guide policies toward data-supported regions (Zhang et al., 2022; Jiang et al., 2023), while conservative smoothing improves local robustness (Yang et al., 2022). SCAS unifies value-aware state correction and OOD action suppression, and DASP uses potential outcomes and their density to encourage

data-supported behavior (Mao et al., 2024a; Jiang et al., 2026). Complementary approaches preserve useful generalization through mildly conservative value targets, data-geometric constraints, or in-sample-guided OOD value smoothing (Lyu et al., 2022; Li et al., 2023; Yao et al., 2025). DOSER combines action support with the marginal support and value of predicted successors for selective value regularization (Wang et al., 2026). These methods already connect support, consequences, and value, although marginal successor-state support alone does not establish support for a specific state–action–successor relation.

Future consequences and controlled propagation. Prior methods account for unreliable future decisions through reward design, model rollouts, and value updates. MOPO and MOReL incorporate model unreliability into long-term optimization through uncertainty-penalized rewards and a pessimistic decision process, respectively (Yu et al., 2020; Kidambi et al., 2020). MOBILE links model uncertainty to Bellman-estimate inconsistency, while NUNO controls model rollout length using learned uncertainty (Sun et al., 2023; Koprulu et al., 2025). DMG limits the propagation of action generalization, CPQL uses conservative multi-step value estimation, and UVU estimates policy-conditional value uncertainty that incorporates future uncertainty (Mao et al., 2024b; Kim & Oh, 2026; Zanger et al., 2026). SFPR instead introduces separate recursive estimates of action-support and conditional-transition costs, using relative-value assessments to modulate their influence and support-dependent continuation to control transition-cost propagation. These costs are support-based control signals, not estimates of model uncertainty or reward loss; their incremental contribution is examined against matched behavior-fitting and value-regularization objectives.

3 PRELIMINARIES

Offline actor–critic learning. Consider a discounted Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma_r)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, P is the transition kernel, r the reward function, ρ_0 the initial-state distribution, and $\gamma_r \in [0, 1)$ the reward discount. Offline RL learns a policy $\pi_\theta(a \mid s)$ from a fixed dataset $\mathcal{D} = \{(s_n, a_n, r_n, s'_n, d_n)\}_{n=1}^N$, collected by a behavior policy or a mixture of policies, without further environment interaction. Let $p_\mathcal{D}(a \mid s)$ denote the conditional behavior distribution underlying the data. The objective is to maximize $J(\pi) = \mathbb{E}_{\rho_0, P, \pi}[\sum_{t=0}^\infty \gamma_r^t r_t]$; $Q^\pi(s, a)$ is the expected discounted return given the state and action. The binary indicator $d = 1$ disables bootstrapping; time-limit truncations follow the dataset protocol.

Let $\underline{Q}(s, a) = \min_i Q_i(s, a)$ and $\bar{Q}(s, a) = \min_i \bar{Q}_i(s, a)$ denote the minimum online and target reward-critic estimates. Reward critics learn from recorded transitions without an entropy bonus:

$$
\begin{aligned}
y_Q &= \mathrm{sg}\big[r + \gamma_r(1 - d)\bar{\underline{Q}}(s', a')\big], \\
\mathcal{L}_{\mathrm{TD}} &= \mathbb{E}_{\substack{(s,a,r,s',d)\sim\mathcal{D} \\ a'\sim\bar{\pi}(\cdot|s')}} \left[\sum_i \big(Q_i(s, a) - y_Q\big)^2\right].
\end{aligned}
\tag{1}
$$

$\bar{Q}_i$ and $\bar{\pi}$ denote the target reward critic and target policy, respectively. sg denotes stop-gradient, and k indexes training updates, distinct from environment time t. The actor uses $\underline{Q}$ for policy improvement; behavior-distribution fitting and value regularization are specified in Section 4.4.

Conditional denoising models. Conditional diffusion models represent the distribution of a variable x given context u. Under a clean-sample parameterization (Karras et al., 2022), a denoiser D_ϕ reconstructs x from Gaussian-corrupted inputs:

$$
\mathcal{L}_{\mathrm{denoise}}(\phi) = \mathbb{E}_{x,u,\sigma,\epsilon}\left[w(\sigma)\,\|D_\phi(x + \sigma\epsilon, u; \sigma) - x\|_2^2\right]. \tag{2}
$$

The expectation is over training pairs (x, u), sampled noise scales $\sigma > 0$, and $\epsilon \sim \mathcal{N}(0, I)$; $w(\sigma)$ weights the noise levels. After pretraining, the denoiser is frozen, and its reconstruction errors provide inputs to support scoring. This fitting loss does not itself define support cost, behavioral harm, or failure probability. Section 4.2 specifies the branch variables, conditioning information, and scoring and normalization rules for SFPR's local support evidence.

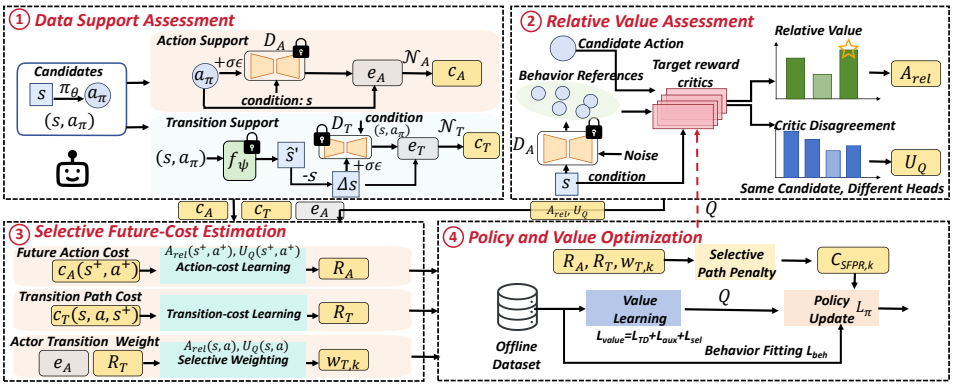

Figure 1: Overview of SFPR. Action support and conditional transition support are assessed alongside behavior-relative value and critic disagreement. Branch-specific future costs guide policy optimization, with value-guided coefficients shared within the respective minibatches for action-cost learning and the actor-side transition-cost term. Behavior-distribution fitting and selective value regularization provide complementary policy and value-learning constraints.

4 METHOD

SFPR separately models action support and conditional transition support, combining support evidence with behavior-relative value assessments to regulate future support costs. Branch-specific estimators connect local costs to subsequent decisions. These assessments determine batch-shared coefficients for action-cost learning and actor-side transition costs, controlling aggregate constraint strengths in both branches. The method comprises factorized risk formulation (Section 4.1), data support and relative value assessment (Section 4.2), value-guided future-cost estimation (Section 4.3), and policy and value optimization (Section 4.4), as summarized in Figure 1.

4.1 FACTORIZED RISK FORMULATION

At a given state s, a candidate decision raises two distinct questions: whether the action a is compatible with recorded behavior, and whether its consequence s' is compatible with recorded state–action–successor relations. Their data evidence corresponds to the conditional factorization

$$p_{\mathcal{D}}(a, s' \mid s) = \underbrace{p_{\mathcal{D}}(a \mid s)}_{\text{action support}} \quad \underbrace{p_{\mathcal{D}}(s' \mid s, a)}_{\text{conditional transition support}} . \tag{3}$$

A successor state can be common without the particular state–action–successor relation being supported. We therefore model the two factors separately, making their evidence separately queryable and controllable. This is a decomposition of data-support evidence, not of independent failure events, and does not automatically eliminate overlapping penalties.

We obtain local support costs $c_A(s, a)$ and $c_T(s, a, s')$. Branch-specific estimators connect them to subsequent decisions: $R_A(s, a)$ estimates batch-weighted support costs of policy actions following the current decision, whereas $R_T(s, a)$ includes the current conditional-consequence cost and subsequent transition costs. Relative-value comparisons determine how these estimates constrain policy improvement, rather than treating insufficient support as sufficient evidence of harm. At training update k, consider a policy minibatch $\mathcal{B} = \{(s_i, a_i)\}_{i=1}^{B}$, with states sampled from $\mathcal{D}$ and actions sampled from π_θ. The two cost estimates form the batch-dependent policy penalty

$$\begin{aligned} H_k^{\mathcal{B}}(s, a) &= \lambda \left(\eta_k [R_A(s, a)]_+ + \bar{w}_{T,k}^{\mathcal{B}} [R_T(s, a)]_+ \right) , \\ C_{\mathrm{SFPR},k}^{\mathcal{B}}(s, a) &= \Phi\big(H_k^{\mathcal{B}}(s, a)\big) . \end{aligned} \tag{4}$$

Here λ is the shared cost strength, η_k controls training-stage activation of the action contribution, and $\bar{w}_{T,k}^{\mathcal{B}} \in [0, 1]$ is shared by all samples in the policy minibatch. The action branch also uses a batch-shared coefficient, $\bar{w}_{A,k}^{\mathcal{C}}$, within its cost-learning target; its effect is already incorporated in R_A and is not applied again in equation 4. Section 4.3 defines both coefficients from the respective batches' support and relative-value assessments. A shared coefficient does not make the penalties identical:

their values and gradients still depend on each sample's branch costs. We use $[x]_+ = \max(x, 0)$ and $\Phi(h) = 1 - \exp(-h)$ for $h \geq 0$; this bounded transform does not calibrate failure probabilities.

The policy combines return optimization, behavioral distribution fitting, and future support costs:

$$\mathcal{L}_\pi = \mathbb{E}_{\mathcal{B}}\left[\frac{1}{B}\sum_{i=1}^{B}\left(-\underline{Q}(s_i, a_i) + \beta_R C^{\mathcal{B}}_{\mathrm{SFPR},k}(s_i, a_i)\right)\right] + \lambda_{\mathrm{beh}}\mathcal{L}_{\mathrm{beh}}, \tag{5}$$

where $\underline{Q}(s, a) = \min_i Q_i(s, a)$ is the aggregated reward-value estimate and $\beta_R, \lambda_{\mathrm{beh}} \geq 0$. The shared coefficient is held fixed during each actor gradient computation. The behavior-fitting term provides a data-derived reference for policy updates, while the future cost adds constraints on subsequent behavior. Section 4.4 specifies this distribution-fitting objective and the selective regularization used to learn the reward values.

4.2 Data Support and Relative Value Assessment

Following equation 3, we train separate conditional action and transition models using the denoising formulation in Section 3, and freeze them during policy learning. The transition model uses the state increment $\Delta s = s' - s$. Given s, the increment and successor are mutually recoverable; this choice changes the parameterization, rather than the information represented by the conditional consequence. For branch $B \in \{A, T\}$, the support score is the denoising reconstruction error $\mathfrak{e}_B(x, u) = \mathbb{E}_{\sigma,\epsilon}\left[\|D_B(x + \sigma\epsilon, u; \sigma) - x\|_2\right], \quad \epsilon \sim \mathcal{N}(0, I)$. The action branch uses $(x, u) = (a, s)$, and the transition branch uses $(x, u) = (s' - s, (s, a))$. We obtain the local costs

$$\begin{aligned} c_A(s, a) &= \mathcal{N}_A(\mathfrak{e}_A(a, s)), \\ c_T(s, a, s') &= \mathcal{N}_T(\mathfrak{e}_T(s' - s, (s, a))), \end{aligned} \tag{6}$$

where each $\mathcal{N}_B$ is a fixed, branch-specific quantile normalization fitted on offline data and clipped to $[0, 1]$. These costs provide proxies for incompatibility with recorded relations, rather than direct measurements of return loss. Finite-sample scoring, preprocessing, and normalization are specified in Appendix A.

Recorded transitions are scored using their observed successors. For a policy candidate a_π, a separately trained and frozen dynamics predictor supplies $\hat{s}' = f_\psi(s, a_\pi)$, which is evaluated through $c_T(s, a_\pi, \hat{s}')$. The predictor proposes a consequence, whereas the conditional scoring model assesses its compatibility with recorded transition relations. These roles are distinct, but both models may extrapolate under weakly covered state–action conditions. Scores on predicted consequences therefore do not establish that those predictions are reliable.

To assess whether a candidate may offer an improvement, we compare its estimated return with the estimated values of behavioral reference actions $a_b^{(j)}$ sampled from the frozen conditional action model. Using target reward critics, let $q_c(s, a) = \min_i \bar{Q}_i(s, a)$ and $b_j(s) = \min_i \bar{Q}_i(s, a_b^{(j)})$, with $b_*(s) = \max_j b_j(s)$. The normalized behavior-relative value is

$$A_{\mathrm{rel}}(s, a) = \frac{q_c(s, a) - b_*(s)}{D(s, a)}, \tag{7}$$

where $D(s, a) > 0$ is a value scale. The sign of $A_{\mathrm{rel}}(s, a)$ indicates an estimated advantage or disadvantage relative to the highest-valued sampled reference. The comparison is evaluated at the state–action pairs relevant to each branch; it does not introduce a separate reward objective for conditional consequences.

Because these comparisons can themselves be affected by extrapolation, we also compute normalized target-critic disagreement:

$$U_Q(s, a) = \frac{\mathrm{Std}_i[\bar{Q}_i(s, a)]}{D(s, a)}. \tag{8}$$

The scale D combines reference-value dispersion, candidate critic disagreement, and reference-value magnitude, with a positive lower bound (Appendix B). A_{rel} is an estimated relative advantage rather than a ground-truth return difference, and U_Q is a disagreement proxy rather than calibrated confidence: agreeing critics may still share a common bias. Together, these signals determine batch-level coefficients for action-cost learning and transition-cost control in the following construction.

4.3 SELECTIVE FUTURE-COST ESTIMATION

Local assessments do not directly describe the behavior that follows a decision. We use the support and value assessments to construct a detached reference signal $G_k(s,a,z) = \mathrm{sg}[\mathcal{G}_k(A_{\mathrm{rel}}(s,a), U_Q(s,a), z)] \in [0,1]$, where z is a cost input and $\mathcal{G} \in [0,1]$ increases for lower relative value, smaller critic disagreement, and appreciable cost. Disagreement moderates reliance on the value comparison: attenuating this additional constraint under disagreement does not declare the candidate safe, and behavioral fitting and value regularization remain active. This signal supplies reference assessments for computing batch-level coefficients, rather than directly weighting individual action targets or actor-side transition costs. Its smooth function and activation schedule are specified in Appendix B.

In our instantiation, the action branch starts with the next policy action, complementing constraints on the current action, whereas the transition branch includes the current conditional consequence. For a cost-learning minibatch $\mathcal{C} = \{(s_i, a_i, s_i^+, m_i)\}_{i=1}^N$, sample $a_i^+ \sim \pi_\theta(\cdot \mid s_i^+)$ and define $z_{A,i} = c_A(s_i^+, a_i^+) + \gamma_A[\bar{R}_A(s_i^+, a_i^+)]_+, q_{A,i} = G_k\big(s_i^+, a_i^+, c_A(s_i^+, a_i^+)\big)$. We aggregate these assessments into one coefficient shared by every sample in $\mathcal{C}$:

$$\bar{w}_{A,k}^{\mathcal{C}} = \mathrm{sg}\left[\frac{\sum_{i=1}^N m_i q_{A,i} z_{A,i}}{\sum_{i=1}^N m_i z_{A,i}}\right] \in [0,1]. \tag{9}$$

We set this coefficient to zero when the denominator is zero. The termination masks are included in the aggregation, so terminal transitions do not determine the strength of future-action supervision. This construction preserves the mean reference action target before clipping, without retaining its sample-specific allocation. The recursive targets are $\widetilde{y}_{A,i} = m_i\, \bar{w}_{A,k}^{\mathcal{C}} z_{A,i}, \widetilde{y}_{T,i} = c_T(s_i, a_i, s_i^+) + m_i\, \kappa_T(s_i, a_i, s_i^+)[\bar{R}_T(s_i^+, a_i^+)]_+$, where $\bar{R}_A, \bar{R}_T$ are target cost estimates and $\gamma_A \in [0,1)$. For recorded transitions, $(s^+, m) = (s', 1-d)$; for model candidates, $(s^+, m) = (f_\psi(s, a_\pi), 1)$. The shared action coefficient modulates both the next-action cost and its continuation uniformly within the cost-learning minibatch. It is recomputed for each such minibatch; uniformity does not imply identical costs across samples or a constant coefficient throughout training. For transitions, $\kappa_T \in [0, \gamma_T^{\max}]$, with $\gamma_T^{\max} < 1$, is non-increasing in the current conditional-transition cost. Poor support thus reduces reliance on subsequent recursive estimates without removing the immediate cost. These coefficients govern continuation along environment time, separately from the reward discount γ_r; training-stage activation is specified in Appendix B. Cost estimators are fitted to detached, clipped targets from both sources, as detailed in Appendix C.

For the actor-side transition term, define $I_A(s,a) = \mathbf{1}[\mathfrak{e}_A(a,s) > \theta_A]$, $q_{T,k}(s,a) = \mathrm{sg}[I_A(s,a) G_k(s,a,R_T(s,a))]$, where θ_A is the action-support threshold. We choose a batch-shared coefficient $\bar{w}_{T,k}^{\mathcal{B}} \in [0,1]$ to satisfy, up to numerical tolerance, $\frac{1}{B}\sum_{i=1}^B C_{\mathrm{SFPR},k}^{\mathcal{B}}(s_i, a_i) = M_k^{\mathcal{B}}$, where $M_k^{\mathcal{B}}$ is the mean reference penalty formed using the samplewise assessments $q_{T,k}$. The coefficient is computed from detached quantities on the current policy minibatch and held fixed during actor differentiation. Appendix B.7 gives the reference construction and matching procedure; Appendix G analyzes the cost operators.

4.4 POLICY AND VALUE OPTIMIZATION

We optimize the policy using equation 5. The frozen conditional action model assesses candidate support, while behavior-distribution fitting trains the policy on recorded actions using a distributional scoring loss ℓ_{dist}:

$$\mathcal{L}_{\mathrm{beh}} = \mathbb{E}_{(s,a_{\mathcal{D}})\sim\mathcal{D}}\left[\ell_{\mathrm{dist}}(\pi_\theta(\cdot \mid s), a_{\mathcal{D}})\right]. \tag{10}$$

Our instantiation uses an established sample-based distributional scoring rule (Gneiting & Raftery, 2007), specified in Appendix D. Together with the return term, this objective provides a data-derived reference for improvement, while $C_{\mathrm{SFPR},k}^{\mathcal{B}}$ adds constraints derived from future support costs.

Because policy improvement relies on reward-value estimates, we additionally regularize the values of selected candidates with insufficient action support:

$$\begin{aligned} \mathcal{L}_{\mathrm{value}} &= \mathcal{L}_{\mathrm{TD}} + \mathcal{L}_{\mathrm{aux}} + \mathcal{L}_{\mathrm{sel}}, \\ \mathcal{L}_{\mathrm{sel}} &= \mathbb{E}_{\substack{s\sim\mathcal{D} \\ a\sim\pi_\theta(\cdot|s)}}\left[m_Q(s,a)\sum_i \left(Q_i(s,a) - \mathrm{sg}[y_{\mathrm{ref}}(s,a)]\right)^2\right]. \end{aligned} \tag{11}$$

Table 1: Normalized scores on nine D4RL locomotion-v2 datasets. Baselines retain their published reporting protocols; SFPR reports Tail at 1M updates. The **best** is shown in bold.

Dataset	CQL	TD3+BC	IQL	ReBRAC	Diffusion-QL	DTQL	DMG	SCAS-ht	CPQL	DOSER	**SFPR**
HalfCheetah-M	44.00	48.30	47.40	65.60	51.10	57.90	54.90	58.50	66.60	67.50	**69.03**±0.60
Hopper-M	58.50	59.30	66.30	102.00	90.50	99.60	100.60	102.50	99.70	104.00	**104.36**±1.23
Walker2d-M	72.50	83.70	78.30	82.50	87.00	89.40	**92.40**	90.80	90.00	86.70	88.60±2.56
HalfCheetah-MR	45.50	44.60	44.20	51.00	47.80	50.90	51.40	52.90	60.30	63.00	**64.32**±1.25
Hopper-MR	95.00	60.90	94.70	98.10	101.30	100.00	101.90	101.60	103.00	104.40	**105.57**±0.84
Walker2d-MR	77.20	81.80	73.90	77.30	95.50	88.50	89.70	88.10	97.40	94.40	**97.56**±0.35
HalfCheetah-ME	91.60	90.70	86.70	**101.10**	96.80	92.70	91.10	91.70	95.30	96.20	98.20±2.02
Hopper-ME	105.40	98.00	91.50	107.00	111.10	109.30	110.40	109.70	111.30	111.50	**112.72**±0.72
Walker2d-ME	108.80	110.10	109.60	111.60	110.10	110.00	**114.40**	110.80	112.90	110.90	111.21±0.23
Mean	77.61	75.27	76.96	88.47	87.91	88.70	89.64	89.62	92.94	93.18	**94.62**±0.44

Here, $\mathcal{L}_{\mathrm{TD}}$ is the recorded-transition TD objective in Section 3, and $\mathcal{L}_{\mathrm{aux}}$ fits the auxiliary values used to construct reference targets. The nonnegative weight m_Q encodes candidate selection and regularization strength, and y_{ref} is the corresponding reference target; both are stop-gradient quantities. We retain the DOSER-style selective value-regularization interface (Wang et al., 2026). Its action-support, predicted marginal-state-support, and reference-value comparisons select either a conservative value anchor or a reference-based compensation target (Appendix D). This inherited rule constrains reward-value estimates; it is distinct from the batch-shared action-cost and transition-cost coefficients and does not use an identical selection or target construction.

We first train and freeze the support models and required consequence predictors, then alternate reward-value, branch-cost, and policy updates on the fixed offline dataset while maintaining their target networks. Predicted consequences support cost learning and reference assessments, but do not replace recorded transitions in reward TD targets. The present instantiation omits entropy bonuses from both the actor objective and the reward TD target; gradient handling and update schedules are given in Appendix E. Evaluation requires only the learned policy and its normalization statistics. Our Gaussian-policy instantiation samples one action per state, without online support-model queries, candidate-action search, or risk-trajectory planning.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate SFPR on the locomotion and Maze2D navigation benchmarks in D4RL (Fu et al., 2020). The locomotion benchmark comprises nine datasets, combining HalfCheetah, Hopper, and Walker2d with the *medium*, *medium-replay*, and *medium-expert* settings. These settings vary in behavioral quality and data composition, allowing us to assess policy improvement under different levels of data support. The *medium-expert* datasets combine medium-quality and expert behavior. For navigation, we use *maze2d-umaze-v1*, *maze2d-medium-v1*, and *maze2d-large-v1* with sparse rewards and fixed goals, evaluating performance across different maze layouts and scales.

Baselines. For locomotion, we compare against ten baselines: CQL (Kumar et al., 2020), TD3+BC (Fujimoto & Gu, 2021), IQL (Kostrikov et al., 2022), ReBRAC (Tarasov et al., 2023), Diffusion-QL (Wang et al., 2023), DTQL (Chen et al., 2024b), DMG (Mao et al., 2024b), SCAS-ht (Mao et al., 2024a), CPQL (Kim & Oh, 2026), and DOSER (Wang et al., 2026). These methods cover conservative value learning, behavior-constrained optimization, generative policies, and selective treatment of out-of-distribution behavior. For Maze2D, we include CQL (Kumar et al., 2020), IQL (Kostrikov et al., 2022), ROMI-BCQ (Wang et al., 2021), OptiDICE-MSE (Lee et al., 2021), CDE (Cen et al., 2024), Diffuser (Janner et al., 2022), Diffuser+RGG+ (Lee et al., 2023), Diffuser+TAT (Feng et al., 2024), Diffuser+LoMAP (Lee & Choi, 2025), and DOSER (Wang et al., 2026), covering value-based learning, model-assisted learning, and generative planning. Locomotion baseline scores are taken from the literature, including the officially reported DOSER results. For Maze2D, we use our local DOSER evaluation and literature-reported results for the remaining methods.

Table 2: Normalized scores on sparse-reward, fixed-goal Maze2D-v1. The **best** is shown in bold.

Dataset	CQL	IQL	ROMI-BCQ	OptiDICE-MSE	CDE	Diffuser	Diffuser +RGG+	Diffuser +TAT	Diffuser +LoMAP	DOSER	**SFPR (ours)**
U-Maze	5.70	47.40	139.50	105.80	134.10	113.90	109.50	114.50	126.00	30.24	**146.27**±2.43
Medium	5.00	34.90	82.40	145.20	146.10	121.50	132.10	130.70	131.00	98.04	**150.23**±2.34
Large	12.50	58.60	83.10	155.70	210.00	123.00	143.90	133.40	151.90	105.75	**221.72**±1.41
Mean	7.73	46.97	101.67	135.57	163.40	119.47	128.50	126.20	136.30	78.01	**172.74**±1.22

Implementation. SFPR uses a Gaussian policy network. The action-support model, conditional-transition support model, and consequence predictor are pretrained on the offline dataset and held fixed during policy learning. At evaluation time, actions are sampled directly from the policy. Ablations use a matched learning backbone: except for the factor under investigation, we keep the policy architecture, pretrained assets, and training and evaluation settings unchanged. Network architectures, pretraining configurations, and task-specific hyperparameters are provided in the appendix.

Training and evaluation. Local policy experiments use 10^6 gradient updates, with evaluation every 2×10^4 updates over 10 episodes. We use D4RL normalized returns and define the *Tail* score of each run as the average return over its final five evaluations. For our method, we report the mean and standard deviation of Tail scores across two independent randomized runs.

5.2 Main Results

Locomotion performance. Table 1 shows that SFPR achieves the highest average normalized score of 94.62 among the compared methods and ranks first on six of the nine datasets. SFPR outperforms DOSER (Wang et al., 2026) on all nine datasets by an average of 1.44 points. These gains span medium, medium-replay, and medium-expert data, highlighting its ability to learn high-return policies from suboptimal behavior while also performing well on datasets containing expert demonstrations. SFPR exceeds CPQL (Kim & Oh, 2026) on seven datasets by approximately 1.7 points on average, demonstrating that its combination of data-support assessment and value-guided control remains competitive with a strong approach that explicitly exploits multi-step offline trajectories. SFPR surpasses DMG (Mao et al., 2024b) by 14.13 and 12.92 points on HalfCheetah-medium and HalfCheetah-medium-replay, respectively, showing its capacity to learn high-return policies from suboptimal datasets. SFPR also outperforms SCAS-ht (Mao et al., 2024a) on eight datasets, including all three medium-replay tasks, indicating that its advantages over explicit state-correction approaches extend across multiple environments and behavioral data compositions. Results demonstrate both strong aggregate performance and broad competitiveness across the evaluated settings.

Maze2D performance. Table 2 shows that SFPR achieves the highest scores on all three Maze2D datasets, with an average normalized return of 172.74. SFPR exceeds our local DOSER evaluation by 94.73 points on average and by 115.97 points on Maze2D-large, showing that its performance advantages extend beyond locomotion to sparse-reward navigation. SFPR outperforms CDE (Cen et al., 2024), which employs conservative occupancy-distribution estimation, by 12.17, 4.13, and 11.72 points on U-Maze, Medium, and Large, respectively. These results demonstrate the competitiveness of combining data-support assessment with value-guided control across different navigation layouts. SFPR also surpasses all Diffuser-based planning methods included in the table, exceeding Diffuser+LoMAP (Lee & Choi, 2025), the strongest among them in average score, by 36.44 points. This comparison shows that directly learning an executable policy can achieve higher returns than the compared generative trajectory-planning approaches on these tasks. In particular, SFPR achieves a Tail score of 221.72±1.41 on Maze2D-large, retaining its lead in the larger layout. Results demonstrate strong navigation performance across the evaluated maze layouts and competitiveness against approaches based on value regularization, occupancy-distribution control, and generative planning.

5.3 Ablation Study

Contributions of factorized support costs. To assess whether support-cost regularization helps maintain policy performance, we compare SFPR with *NoPath*, which removes both branch-cost penalties from policy optimization. As shown in Figure 2, SFPR exhibits fewer pronounced return drops on Hopper-medium-replay and higher late-training returns on Maze2D-large, while achieving performance comparable to NoPath on Walker2d-medium-expert. These observations suggest that

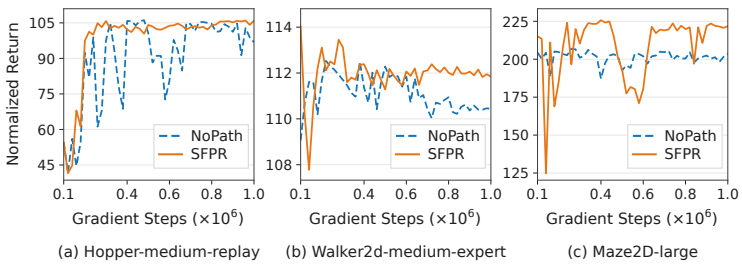

Figure 2: Learning curves of SFPR and NoPath, which removes the additional support-cost penalty while retaining behavior-distribution fitting and selective value regularization.

Table 3: Branch ablation on Maze2D-large. Bold denotes the highest Tail score.

Variant	Action	Transition	Tail↑
Action-Only	✓	✗	211.72
Transition-Only	✗	✓	201.46
SFPR	✓	✓	**221.72**

Table 4: Tail comparison of samplewise weighting (SAM), zero cost continuation (G-zero), and SFPR.

Dataset	SAM	G-zero	SFPR
Hopper-ME	104.95	107.74	**112.72**
Walker2d-ME	110.46	110.59	**111.21**
Maze2D-large	218.47	214.34	**221.72**

the additional support-cost constraints can help preserve high returns. Additional learning curves are provided in Appendix H.

To assess whether combining both cost branches is beneficial, we compare SFPR with Action-Only and Transition-Only, which retain only the action-support or conditional-transition cost, respectively, in the policy objective. As shown in Table 3, SFPR achieves a Tail score of 221.72 on Maze2D-large, exceeding Action-Only and Transition-Only by 10.00 and 20.26 points, respectively. These results favor jointly using the two support-cost branches in this setting, as neither branch alone recovers the late-training performance of the complete method.

Allocation of support evidence. We compare SFPR with SAM that a samplewise-weighted variant. As shown in Table 4, SFPR outperforms samplewise weighting by 7.77, 0.75, and 3.25 points on Hopper-medium-expert, Walker2d-medium-expert, and Maze2D-large, respectively. Results support batch-shared weighting as an effective way to allocate support costs: data-support and relative-value assessments need not assign an independent weight to every sample, but can instead guide policy improvement by regulating the aggregate constraint strength within each minibatch.

Recursive propagation of support costs. To examine the role of recursive support-cost propagation, we compare SFPR with variants G-ZERO that disable continuation in both cost branches. These variants retain the current conditional-transition cost and the next-state policy-action cost, while leaving the reward discount unchanged. As shown in Table 4, SFPR achieves Tail scores that are 4.98 and 7.38 points higher on Hopper-medium-expert and Maze2D-large, respectively. The reported the advantage of recursive propagation across datasets.

6 Conclusion

A central challenge in offline reinforcement learning is to improve policies reliably under limited data support. We introduced Selective Factorized Path Risk (SFPR), which separately models action support and transition support conditioned on the state–action pair. SFPR combines behavior-relative value assessments with two recursive support-cost estimators to selectively constrain subsequent actions and conditional consequences associated with a current decision, while controlling cost propagation across future decisions. The path-cost mechanism is integrated with behavior-distribution fitting and an inherited selective Q regularizer; deployment requires only the learned policy and its normalization statistics. On the D4RL locomotion and Maze2D benchmarks reported here, SFPR achieves higher average performance than the compared methods. SFPR offers a testable design approach for finer-grained constraints on subsequent behavior. Future work could investigate task-adaptive adjustment of the two support costs and their propagation in response to different feedback structures.

AI USE STATEMENT

In this work, we used ChatGPT to assist with translation, drafting selected passages, and language editing of the manuscript, including grammar correction, sentence rephrasing, and improvements to clarity and readability. We did not use generative AI tools for research ideation, experimental design, method implementation, data generation or processing, or analysis and interpretation of results. The authors reviewed and revised all AI-assisted text, checking its accuracy and consistency with the underlying research. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study relies exclusively on standard, publicly available datasets and does not involve the recruitment of human participants or animal experimentation. All datasets were used in strict accordance with their respective licenses and terms of use. Given the public and standardized nature of the data, this work does not raise any concerns regarding privacy, security, or legal compliance.

REPRODUCIBILITY STATEMENT

We describe the proposed method in Section 4 and the experimental setup and evaluation protocols in Section 5. Appendices A–E provide additional details on support scoring, relative-value assessment, selective weighting, cost-critic fitting, policy and value learning, and the training procedure and hyperparameters. The source code will be made publicly available upon acceptance of this paper.

REFERENCES

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified Q-ensemble. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Zhepeng Cen, Zuxin Liu, Zitong Wang, Yihang Yao, Henry Lam, and Ding Zhao. Learning from sparse offline datasets via conservative density estimation. In *International Conference on Learning Representations*, 2024.

Huayu Chen, Cheng Lu, Zhengyi Wang, Hang Su, and Jun Zhu. Score regularized policy optimization through diffusion behavior. In *International Conference on Learning Representations*, 2024a.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1042–1051. PMLR, 2019.

Tianyu Chen, Zhendong Wang, and Mingyuan Zhou. Diffusion policies creating a trust region for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Lang Feng, Pengjie Gu, Bo An, and Gang Pan. Resisting stochastic risks in diffusion planners with the trajectory aggregation tree. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 13175–13198. PMLR, 2024.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning, 2020.

Scott Fujimoto and Shixiang (Shane) Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 2019.

Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. Extreme Q-learning: MaxEnt RL without entropy. In *International Conference on Learning Representations*, 2023.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

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

Ke Jiang, Jia-Yu Yao, and Xiaoyang Tan. Recovering from out-of-sample states via inverse dynamics in offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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, pp. 22327–22335, 2026.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Byeongchan Kim and Min-hwan Oh. Peng's $Q(\lambda)$ for conservative value estimation in offline reinforcement learning. In *International Conference on Learning Representations*, 2026.

Cevahir Koprulu, Franck Djeumou, and Ufuk Topcu. Neural stochastic differential equations for uncertainty-aware offline RL. In *International Conference on Learning Representations*, 2025.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

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

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Romain Laroche, Paul Trichelair, and Rémi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3652–3661. PMLR, 2019.

Jongmin Lee, Wonseok Jeon, Byungjun Lee, Joelle Pineau, and Kee-Eung Kim. OptiDICE: Offline policy optimization via stationary distribution correction estimation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6120–6130. PMLR, 2021.

Kyowoon Lee and Jaesik Choi. Local manifold approximation and projection for manifold-aware diffusion planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 32987–33004. PMLR, 2025.

Kyowoon Lee, Seongun Kim, and Jaesik Choi. Refining diffusion planner for reliable behavior synthesis by automatic detection of infeasible plans. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jianxiong Li, Xianyuan Zhan, Haoran Xu, Xiangyu Zhu, Jingjing Liu, and Ya-Qin Zhang. When data geometry meets deep function: Generalizing offline reinforcement learning. In *International Conference on Learning Representations*, 2023.

Jiafei Lyu, Xiaoteng Ma, Xiu Li, and Zongqing Lu. Mildly conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Yixiu Mao, Qi Wang, Chen Chen, Yun Qu, and Xiangyang Ji. Offline reinforcement learning with OOD state correction and OOD action suppression. In *Advances in Neural Information Processing Systems*, volume 37, 2024a.

Yixiu Mao, Qi Wang, Yun Qu, Yuhang Jiang, and Xiangyang Ji. Doubly mild generalization for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024b.

Xuan Thanh Nguyen and Chang Yoo. One-step flow Q-learning: Addressing the diffusion policy bottleneck in offline reinforcement learning. In *International Conference on Learning Representations*, 2026.

Seohong Park, Qiyang Li, and Sergey Levine. Flow Q-learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 48104–48127. PMLR, 2025.

Noah Y. Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavior modelling priors for offline reinforcement learning. In *International Conference on Learning Representations*, 2020.

Yihao Sun, Jiaji Zhang, Chengxing Jia, Haoxin Lin, Junyin Ye, and Yang Yu. Model-Bellman inconsistency for model-based offline reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33177–33194. PMLR, 2023.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jianhao Wang, Wenzhe Li, Haozhe Jiang, Guangxiang Zhu, Siyuan Li, and Chongjie Zhang. Offline reinforcement learning with reverse model-based imagination. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Qingjun Wang, Hongtu Zhou, Hang Yu, Junqiao Zhao, Yanping Zhao, Chen Ye, Ziqiao Wang, and Guang Chen. Beyond penalization: Diffusion-based out-of-distribution detection and selective regularization in offline reinforcement learning. In *International Conference on Learning Representations*, 2026.

Zhendong Wang, Jonathan J. Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations*, 2023.

Jialong Wu, Haixu Wu, Zihan Qiu, Jianmin Wang, and Mingsheng Long. Supported policy optimization for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Wai Kin Chan, and Xianyuan Zhan. Offline RL with no OOD actions: In-sample learning via implicit value regularization. In *International Conference on Learning Representations*, 2023.

Rui Yang, Chenjia Bai, Xiaoteng Ma, Zhaoran Wang, Chongjie Zhang, and Lei Han. RORL: Robust offline reinforcement learning via conservative smoothing. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Qingmao Yao, Zhichao Lei, Tianyuan Chen, Ziyue Yuan, Xuefan Chen, Jianxiang Liu, Faguo Wu, and Xiao Zhang. Offline RL with smooth OOD generalization in convex hull and its neighborhood. In *International Conference on Learning Representations*, 2025.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Moritz Akiya Zanger, Max Weltevrede, Yaniv Oren, Pascal R. Van der Vaart, Caroline Horsch, Wendelin Boehmer, and Matthijs T. J. Spaan. Universal value-function uncertainties. In *International Conference on Learning Representations*, 2026.

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, pp. 9022–9030, 2022.

A Support Scoring and Consequence Prediction

A.1 Conditional reconstruction scores

We train conditional denoising models for the action and transition factors in Eq. equation 3. Each model is frozen during policy learning, together with its preprocessing and scoring protocol. Here, s denotes the coordinates expected by the pretrained models, and $\Delta s = s' - s$ is computed in the same coordinates. Raw and normalized increments are not interchangeable. Given s, the increment is an invertible reparameterization of the successor state, not an additional source of information.

For $B \in \{A, T\}$, let $(x_A, u_A) = (a, s)$ and $(x_T, u_T) = (s' - s, (s, a))$. We approximate the expected reconstruction error by

$$\widehat{\mathfrak{e}}_B(x, u) = \frac{1}{L_B} \sum_{\ell=1}^{L_B} \left\| D_B(x + \sigma_\ell \epsilon_\ell, u; \sigma_\ell) - x \right\|_2, \qquad \epsilon_\ell \sim \mathcal{N}(0, I). \tag{12}$$

Noise scales follow the pretrained diffusion protocol. The score uses the Euclidean norm, not its square. The main-text notation $\mathfrak{e}_B$ suppresses the finite-sample approximation. Scoring calls may use fresh noise, while the frozen model and noise-sampling law remain unchanged.

A.2 Normalization and action-support detection

Let q_B^{lo} and q_B^{hi} be prescribed quantiles of branch-specific reconstruction scores on offline data, fitted before policy learning and then fixed. The normalization in Eq. equation 6 is

$$\mathcal{N}_B(e) = \mathrm{clip}_{[0,1]} \left(\frac{e - q_B^{\mathrm{lo}}}{q_B^{\mathrm{hi}} - q_B^{\mathrm{lo}} + \varepsilon} \right). \tag{13}$$

We use the 50th and 95th percentiles and $\varepsilon = 10^{-6}$. If the fitted quantiles are numerically indistinguishable, the upper value is set to $q_B^{\mathrm{lo}} + \varepsilon$ before normalization. Range normalization does not establish probability calibration; scores above the clipping threshold lose their relative ordering.

The action detector uses a separate threshold:

$$I_A(s, a) = \mathbf{1}[\mathfrak{e}_A(a, s) > \theta_A]. \tag{14}$$

Its task-specific percentile is distinct from the percentiles used to normalize c_A.

A.3 Observed and predicted consequences

Recorded pairs use the observed successor s'. For a policy candidate at a dataset state, the frozen predictor supplies $\hat{s}' = f_\psi(s, a_\pi)$. It is trained on recorded transitions separately from the conditional support-scoring model. Its output is converted to the scorer's coordinates before constructing the increment. The Gaussian-policy reference instantiation forms a point prediction by averaging five samples from a single-member predictive distribution; it does not add an ensemble-disagreement cost.

The consequence predictor f_ψ and the predictor f_G used by selective value regularization (Appendix D) use different frozen checkpoints. The latter is a pretrained MLP that predicts successors for the candidate and its behavior-reference action. The marginal-state scorer e_S uses a separate frozen state-denoising checkpoint, whereas conditional-transition scoring uses a state–action-conditioned increment-denoising checkpoint. These models serve distinct roles; their separate specification does not imply statistical independence. Neither model-generated cost labels nor predicted successors replace recorded transitions in reward TD learning.

B Relative Values, Cost Weights, and Continuation

B.1 Reference values and normalization

Future-cost assessments use target reward critics. For behavior references, compute $b_j = \min_i \bar{Q}_i(s, a_b^{(j)})$ and $b_* = \max_j b_j$. For the candidate, use $q_c = \min_i \bar{Q}_i(s, a)$ and $\sigma_Q =$

$\mathrm{Std}_i[\bar{Q}_i(s,a)]$. The standard deviation uses the population convention across critic heads. The scale in Eqs. equation 7 and equation 8 is

$$D(s,a) = \max\left\{d_0,\ \mathrm{MAD}_j(b_j) + \sigma_Q + \zeta\,|\mathrm{median}_j(b_j)|\right\}. \tag{15}$$

Here MAD is the median absolute deviation from the median. We use $d_0 = 1$ and $\zeta = 0.05$. The implementation's median operator selects the lower middle order statistic for an even-sized set. The reference value maximizes over sampled actions, not critic heads. This normalization neither calibrates disagreement nor removes shared value-estimation bias.

B.2 Reference assessment function

Let $\mathrm{sigm}(v) = (1 + e^{-v})^{-1}$. Define

$$\begin{aligned} g_V(A) &= b_V + (1 - b_V)\,\mathrm{sigm}\left(\frac{-A - m_V}{\tau_V}\right), \\ g_U(U) &= \mathrm{sigm}\left(\frac{u_0 - U}{\tau_U}\right), \\ g_C(z) &= \mathrm{sigm}\left(\frac{z - z_0}{\tau_C}\right), \\ \mathcal{G}(A,U,z) &= g_V(A) g_U(U) g_C(z), \\ \mathcal{G}_k(A,U,z) &= \eta_k \mathcal{G}(A,U,z). \end{aligned} \tag{16}$$

The main-text G_k applies stop-gradient to this assessment. All temperatures are positive, and $b_V \in [0,1]$ is a configurable floor on the relative-value factor. The gate temperature τ_V is distinct from the auxiliary-value expectile level ν_V. Numerical settings are collected in Appendix F.

High disagreement attenuates reliance on the value comparison when imposing additional costs; it does not establish that the candidate is safe. Nonfinite candidate values, reference values, or costs yield a zero reference assessment in the numerical guard. This guard does not make an otherwise nonfinite training update valid.

B.3 Branch-specific insertion points

For the action branch, the next-action reference assessment is

$$q_{A,k}(s^+, a^+) = G_k(s^+, a^+, c_A(s^+, a^+)). \tag{17}$$

For the transition branch, use

$$q_{T,k}(s,a) = \mathrm{sg}[I_A(s,a) G_k(s,a,R_T(s,a))]. \tag{18}$$

These assessments determine the detached coefficients $\bar{w}^{\mathcal{C}}_{A,k}$ and $\bar{w}^{\mathcal{B}}_{T,k}$ defined below. The former is shared across action-cost targets in one cost-learning minibatch; the latter is shared across actor-side transition costs in one policy minibatch. The transition-cost target retains its immediate cost and support-dependent continuation without the actor-side multiplier. The two shared coefficients need not be equal.

B.4 Policy-cost aggregation

Writing $[x]_+ = \max(x, 0)$ and $\Phi(h) = 1 - \exp(-h)$ for $h \geq 0$, the policy penalty is

$$\begin{aligned} H^{\mathcal{B}}_k(s,a) &= \lambda\left(\eta_k [R_A(s,a)]_+ + \bar{w}^{\mathcal{B}}_{T,k} [R_T(s,a)]_+\right), \\ C^{\mathcal{B}}_{\mathrm{SFPR},k}(s,a) &= \Phi\left(H^{\mathcal{B}}_k(s,a)\right). \end{aligned} \tag{19}$$

The positive part prevents negative cost estimates from reversing the penalty. The bounded transform is not a calibrated failure probability. The action-target coefficient is already incorporated into R_A and is not multiplied into the actor penalty again. The schedule η_k appears in the reference assessments and explicitly on the actor's action-cost term; these insertion points are distinct.

B.5 SUPPORT-DEPENDENT CONTINUATION

The transition-cost continuation coefficient is

$$\kappa_T(s,a,s^+) = \mathrm{sg}\left[\gamma_T^{\min} + (\gamma_T^{\max} - \gamma_T^{\min})\,\mathrm{sigm}\left(\frac{c_0 - c_T(s,a,s^+)}{\tau_\kappa}\right)\right], \tag{20}$$

where $0 \le \gamma_T^{\min} \le \gamma_T^{\max} < 1$ and $\tau_\kappa > 0$. The action branch uses $\gamma_A \in [0,1)$, with effective bootstrap coefficient $m_i \bar{w}^{\mathcal{C}}_{A,k} \gamma_A$. The shared action weight is constant within $\mathcal{C}$, whereas $\kappa_T(s_i,a_i,s_i^+)$ remains sample-dependent. Lower continuation reduces reliance on subsequent estimates without removing the immediate transition cost or identifying model error.

B.6 TRAINING-STAGE ACTIVATION

For warm-up length $K_{\mathrm{w}} \ge 0$ and ramp length $K_{\mathrm{r}} > 0$,

$$\eta_k = \mathrm{clip}_{[0,1]}\left(\frac{k - K_{\mathrm{w}}}{K_{\mathrm{r}}}\right). \tag{21}$$

This schedule enters both reference assessments and the explicit action-cost term in Eq. equation 19. Because transition aggregation matches a nonlinear penalty, its effect is not generally equivalent to multiplying a fixed shared coefficient by η_k. Warm-up leaves behavior fitting, selective value regularization, and transition-cost learning active. Action-cost fitting remains active with zero gated targets. Unlike η_k, γ_A and κ_T govern continuation along environment time.

B.7 BATCH-SHARED COST WEIGHTS

Action-cost supervision. For a cost-learning minibatch $\mathcal{C} = \{(s_i,a_i,s_i^+,m_i)\}_{i=1}^N$, sample $a_i^+ \sim \pi_\theta(\cdot \mid s_i^+)$ and define

$$\begin{aligned} z_{A,i} &= c_A(s_i^+,a_i^+) + \gamma_A[\bar{R}_A(s_i^+,a_i^+)]_+, \\ S_A^{\mathcal{C}} &= \sum_{i=1}^N m_i z_{A,i}, \\ \bar{w}^{\mathcal{C}}_{A,k} &= \mathrm{sg}\left[\begin{cases} \dfrac{\sum_{i=1}^N m_i q_{A,i} z_{A,i}}{S_A^{\mathcal{C}}}, & S_A^{\mathcal{C}} > 0, \\ 0, & S_A^{\mathcal{C}} = 0. \end{cases}\right] \end{aligned} \tag{22}$$

Here $q_{A,i} = q_{A,k}(s_i^+,a_i^+)$. Since $m_i, z_{A,i} \ge 0$ and $q_{A,i} \in [0,1]$, the shared coefficient lies in $[0,1]$. The raw targets $\widetilde{y}_{A,i} = m_i \bar{w}^{\mathcal{C}}_{A,k} z_{A,i}$ satisfy

$$\frac{1}{N}\sum_i \widetilde{y}_{A,i} = \frac{1}{N}\sum_i m_i q_{A,i} z_{A,i}. \tag{23}$$

This also holds when $S_A^{\mathcal{C}} = 0$. The rule preserves the reference mean before clipping, not individual targets, clipped means, regression losses, or gradients. Recorded and model-candidate sources use separate minibatches and separate shared coefficients.

Actor-side transition cost. On the policy minibatch $\mathcal{B} = \{(s_i,a_i)\}_{i=1}^B$, use detached values

$$h_{A,i} = \lambda \eta_k [R_A(s_i,a_i)]_+, \qquad h_{T,i} = \lambda [R_T(s_i,a_i)]_+, \qquad q_i = q_{T,k}(s_i,a_i).$$

Define

$$F_{\mathcal{B}}(w) = \frac{1}{B}\sum_i \Phi(h_{A,i} + w h_{T,i}), \qquad w \in [0,1], \tag{24}$$

and the reference batch mean

$$M_k^{\mathcal{B}} = \mathrm{sg}\left[\frac{1}{B}\sum_i \Phi(h_{A,i} + q_i h_{T,i})\right]. \tag{25}$$

The shared coefficient approximately satisfies $F_{\mathcal{B}}(\bar{w}^{\mathcal{B}}_{T,k}) = M^{\mathcal{B}}_k$. A supported current action can still receive a nonzero transition penalty under a nonzero shared coefficient.

For finite nonnegative hazards and $q_i \in [0,1]$, $F_{\mathcal{B}}$ is continuous and nondecreasing, and

$$F_{\mathcal{B}}(0) \le M^{\mathcal{B}}_k \le F_{\mathcal{B}}(1).$$

Thus a matching coefficient exists. If some $h_{T,i} > 0$, then

$$F'_{\mathcal{B}}(w) = \frac{1}{B}\sum_i h_{T,i}\exp(-h_{A,i} - w h_{T,i}) > 0,$$

so the exact matching coefficient is unique. We approximate it by fixed-budget bisection as specified in Appendix F.3. When all $h_{T,i} = 0$, we set $\bar{w}^{\mathcal{B}}_{T,k} = 0$.

For the actor update, evaluate the penalty using the same sampled actions and critic parameters used for matching. Hold the shared coefficient and critic parameters fixed, retaining derivatives of critic outputs with respect to policy actions. Matching preserves the current batch's mean composed penalty approximately, not samplewise penalties, actor gradients, or another run's trajectory. Nonfinite inputs do not define a valid matching problem.

C Cost-Critic Fitting

C.1 Estimators and supervision sources

Each branch uses two estimators with

$$R_B(s,a) = \max_{j=1,2} R_{B,j}(s,a), \qquad \bar{R}_B(s,a) = \max_{j=1,2} \bar{R}_{B,j}(s,a). \tag{26}$$

The maximum discourages underestimation but is not a calibrated statistical upper bound. Recorded tuples are $(s, a_{\mathcal{D}}, s', 1-d)$; model-candidate tuples are $(s, a_\pi, f_\psi(s,a_\pi), 1)$ at dataset source states. For each source $v \in \{\mathcal{D}, \pi\}$, form $\mathcal{C}_v = \{(s_{v,i}, a_{v,i}, s^+_{v,i}, m_{v,i})\}_{i=1}^{N_v}$, sample next actions at that source's successors, and compute $\bar{w}^{\mathcal{C}_v}_{A,k}$ separately. Source regression weights ξ_v are distinct from shared cost weights.

C.2 Targets and regression

For each source, construct

$$\begin{aligned}
\widetilde{y}^{(v)}_{A,i} &= m_{v,i}\bar{w}^{\mathcal{C}_v}_{A,k}\left(c_A(s^+_{v,i}, a^+_{v,i}) + \gamma_A[\bar{R}_A(s^+_{v,i}, a^+_{v,i})]_+\right), \\
\widetilde{y}^{(v)}_{T,i} &= c_T(s_{v,i}, a_{v,i}, s^+_{v,i}) + m_{v,i}\kappa_T(s_{v,i}, a_{v,i}, s^+_{v,i})[\bar{R}_T(s^+_{v,i}, a^+_{v,i})]_+.
\end{aligned} \tag{27}$$

Each target is detached and clipped:

$$y^{(v)}_{B,i} = \mathrm{sg}\left[\mathrm{clip}_{[0,R_{\max}]}(\widetilde{y}^{(v)}_{B,i})\right]. \tag{28}$$

Using the unit-threshold Smooth-L1 loss

$$h(e) = \begin{cases} \frac{1}{2}e^2, & |e| < 1, \\ |e| - \frac{1}{2}, & |e| \ge 1, \end{cases} \tag{29}$$

the fitting objective is

$$\mathcal{L}_R = \sum_{v\in\{\mathcal{D},\pi\}} \xi_v \mathbb{E}_{\mathcal{C}_v}\left[\frac{1}{N_v}\sum_i \sum_{B\in\{A,T\}}\sum_{j=1}^{2} h\Big(R_{B,j}(s_{v,i},a_{v,i}) - y^{(v)}_{B,i}\Big)\right]. \tag{30}$$

The expectation includes minibatch construction, next-action sampling, and scoring randomness. Both sources use weight one in the reference configuration: their separately averaged losses are added, rather than forming a normalized half-and-half mixture. Targets, shared coefficients, support scores, and reference assessments are detached; only the cost estimators receive gradients from this loss. Clipped targets do not by themselves bound neural outputs.

Model candidates use mask one because the predictor does not supply validated termination labels. This may propagate costs beyond a true termination. Recorded terminal transitions retain their immediate transition cost but have no next-action contribution. These estimates are support-cost proxies, not verified counterfactual environment losses.

D Policy and Value Learning Details

D.1 Behavior-distribution fitting

The actor is a tanh-squashed Gaussian. At a dataset state, draw independent actor samples a_1, a_2 and let $a_{\mathcal{D}}$ be the recorded action. With componentwise scaling $u = a \oslash a_{\max}$, the behavior-fitting loss uses the Energy Score (Gneiting & Raftery, 2007):

$$\ell_{\text{dist}}(\pi_\theta(\cdot \mid s), a_{\mathcal{D}}) = \frac{1}{2}\mathbb{E}_{a_1,a_2\sim\pi_\theta(\cdot|s)}\big[\|u_1 - u_{\mathcal{D}}\|_2 + \|u_2 - u_{\mathcal{D}}\|_2 - \|u_1 - u_2\|_2\big]. \tag{31}$$

Two independent samples provide the training estimator. $\mathcal{L}_{\text{beh}}$ is the dataset expectation of this score, not a second behavior loss. The pairwise term distinguishes distribution fitting from regression to a single conditional mean. This scoring rule is adopted rather than proposed here.

D.2 Reward and auxiliary-value targets

The reference instance uses $K_Q = 4$ reward-Q heads and two auxiliary state-value heads. For recorded transitions,

$$\begin{aligned} y_Q &= \text{sg}\left[r + \gamma_r(1-d)\min_{i\le K_Q}\bar{Q}_i(s', a'_{\text{tar}})\right], \quad a'_{\text{tar}} \sim \bar{\pi}(\cdot \mid s'), \\ y_V &= \text{sg}\left[r + \gamma_r(1-d)\min_{j=1,2}\bar{V}_j(s')\right]. \end{aligned} \tag{32}$$

The corresponding losses are

$$\begin{aligned} \mathcal{L}_{\text{TD}} &= \mathbb{E}_{\mathcal{D}}\sum_{i=1}^{K_Q}(Q_i(s, a_{\mathcal{D}}) - y_Q)^2, \\ \mathcal{L}_{\text{aux}} &= \mathbb{E}_{\mathcal{D}}\sum_{j=1}^{2}\ell_{\nu_V}(y_V - V_j(s)), \end{aligned} \tag{33}$$

where $\ell_{\nu_V}(e) = |\nu_V - \mathbf{1}[e < 0]|e^2$ and $\nu_V \in (0,1)$. This is auxiliary state-TD regression, not an IQL-style expectile fit to $Q(s, a_{\mathcal{D}}) - V(s)$. Neither reward targets nor actor updates include an entropy bonus in this instantiation.

D.3 Selective value regularization

We retain the DOSER-style interface (Wang et al., 2026). Unlike future-cost assessments, it uses online reward critics and auxiliary state values. For a finite set of behavior-model samples,

$$a_b^G \in \operatorname*{arg\,max}_{a_j\sim p_{\text{beh}}(\cdot|s)}\min_i Q_i(s, a_j), \qquad q_b^G = \min_i Q_i(s, a_b^G). \tag{34}$$

Here p_{beh} is the frozen conditional action model; ties follow the implementation's sample ordering. Let $\underline{V} = \min(V_1, V_2)$ and define

$$\begin{aligned} \hat{s}_\pi^G &= f_G(s, a_\pi), & \hat{s}_b^G &= f_G(s, a_b^G), \\ \Delta V_G &= \underline{V}(\hat{s}_\pi^G) - \underline{V}(\hat{s}_b^G), & & \\ I_{\text{OOD}} &= \mathbf{1}[\mathfrak{e}_A(a_\pi, s) > \theta_A], & I_{\text{next}} &= \mathbf{1}[e_S(\hat{s}_\pi^G) \le \theta_S], \\ M_+ &= I_{\text{OOD}} I_{\text{next}}\mathbf{1}[\Delta V_G \ge 0], & M_- &= I_{\text{OOD}} - M_+, \\ q_+ &= \text{sg}\big[\eta_G(q_b^G + [\Delta V_G]_+)\big]. & & \end{aligned} \tag{35}$$

η_G belongs to the compensation rule, not the activation schedule. The masks are mutually exclusive proxy categories, not independently verified beneficial and harmful behavior labels. The regularizer is

$$\mathcal{L}_{\text{sel}} = \mathbb{E}_{s,a_\pi}\frac{1}{K_Q}\sum_{i=1}^{K_Q}\Big[\lambda_- M_-(Q_i(s, a_\pi) - q_{\text{floor}})^2 + \lambda_+ M_+(Q_i(s, a_\pi) - q_+)^2\Big]. \tag{36}$$

The task-specific anchor q_{floor} is distinct from the minimum across Q heads. The compact main-text expression is recovered with

$$m_Q(s, a_\pi) = \frac{\lambda_- M_- + \lambda_+ M_+}{K_Q},$$
$$y_{\mathrm{ref}}(s, a_\pi) = \begin{cases} q_{\mathrm{floor}}, & M_- = 1, \\ q_+, & M_+ = 1, \\ 0, & M_- = M_+ = 0. \end{cases} \tag{37}$$

Both m_Q and y_{ref} are detached. The last target is immaterial because its weight is zero. The action-scoring operator is shared with local action assessment, although separate calls can use different noise. The marginal-state scorer e_S is not the conditional-transition cost c_T. This value regularizer therefore has a different selection rule and learning interface from the future-cost assessments.

E Training Procedure and Matched Comparisons

E.1 Updates and gradient flow

The complete procedure is:

1. Train or load the conditional action and increment models, the marginal-state scorer, and the consequence predictors. Freeze their parameters, preprocessing, and normalization statistics.
2. Initialize the Gaussian actor, reward and auxiliary-value heads, two estimators per cost branch, and their target networks.
3. At update k, form recorded and model-candidate minibatches at dataset source states and sample next actions at their successors.
4. Without gradients, compute support scores, reference assessments, continuation coefficients, and the source-specific $\bar{w}_{A,k}^{\mathcal{C}_v}$. Construct and clip cost targets.
5. Update both cost critics and softly update their targets.
6. On policy-update iterations, compute $\bar{w}_{T,k}^{\mathcal{B}}$ from the policy minibatch and minimize Eq. equation 5. Use the same sampled actions and cost parameters as in matching. Freeze shared coefficients and critic parameters while retaining derivatives with respect to policy actions.
7. Update reward and auxiliary values using Eq. equation 11, with terms expanded in Eqs. equation 33 and equation 36. Update their targets and the target actor at the specified frequency.
8. Evaluate using one stochastic policy action per state and the training normalization statistics, without online support-model queries, value-based candidate selection, or trajectory search.

Target parameters follow $\bar{\vartheta} \leftarrow (1 - \tau_{\mathrm{tar}})\bar{\vartheta} + \tau_{\mathrm{tar}}\vartheta$. Cost targets are updated after every cost-critic update; the actor, target actor, and reward/value targets use a two-step interval. No policy gradient passes through reference assessments, matching, support scorers, or consequence predictors.

E.2 Matched comparisons

Matched comparisons retain the actor, pretrained assets, behavior fitting, value regularization, reward protocol, budget, and evaluation protocol unless explicitly changed. NoPath removes the additional actor support-cost penalty, providing a matched-backbone control. Actor-only branch ablations retain cost-critic training and change only the corresponding contribution to policy optimization. Disabling critic training or removing a local score are different interventions.

Zero continuation sets $\gamma_A = 0$ and $\gamma_T^{\min} = \gamma_T^{\max} = 0$ without changing γ_r. The next-action local cost and current transition cost remain. Shared coefficients are recomputed on each variant's own minibatches; changing γ_A can consequently change the shared action weight. Action-target

Table 5: Common training settings of the Gaussian-policy reference configuration.

Hyperparameter	Value
Policy-training updates	10^6
Minibatch size	256
Reward discount, γ_r	0.99
Target soft-update coefficient, τ_{tar}	0.005
Initial actor learning rate	3×10^{-4}
Reward/value-critic learning rate	3×10^{-4}
Cost-critic learning rate	3×10^{-4}
Actor and reward/value-target update interval	2
Cost-critic and cost-target update interval	1
Base strength of each cost branch, λ	0.1
Outer support-cost coefficient, β_R	1.0
Behavior-fitting coefficient, λ_{beh}	1.0
Recorded-source loss weight, $\xi_{\mathcal{D}}$	1.0
Model-candidate loss weight, ξ_{π}	1.0
Perturbed-source and local-consistency weights	0
Cost-target upper bound, $R_{\max}$	2
Smooth-L1 transition point	1.0
Warm-up updates, K_{w}	100,000
Ramp duration, K_{r}	200,000
Evaluation interval	20,000
Episodes per evaluation	10
Tail evaluation window	Last 5 evaluations

aggregation, actor-side transition weighting, and training-stage activation are separate interventions. Historical experiments with different weighting interfaces must be distinguished from matched comparisons of the formulation above.

F Hyperparameters and Numerical Settings

F.1 Common training and assessment settings

Tables 5 and 6 collect the reference settings. Both branches use base strength $\lambda = 0.1$, distinct from the outer coefficient $\beta_R = 1$ and the source-loss weights. Shared coefficients are computed from minibatches, not assigned fixed numerical values. The activation schedule reaches one at 300,000 updates. Task-specific settings and ablation changes are separate from these common reference values.

F.2 Task-specific reference presets

Table 7 lists task-specific reference presets for value regularization and support detection. The implementation fields map to the mathematical notation as $\lambda_- = \texttt{beta}$, $\lambda_+ = \texttt{lam}$, $q_{\mathrm{floor}} = \texttt{Q_min}$, and $\nu_V = \texttt{expectile}$. All listed presets use $\lambda_+ = 0.001$ and $\eta_G = 0.9$. These coefficients are distinct from λ and β_R. Detector percentiles are also distinct from the 50th/95th-percentile normalization of support costs. The table specifies reference presets; it does not identify tuned runs solely from their reported scores.

F.3 Numerical implementation

Action-target sharing uses the ratio in Eq. equation 22, with a separate zero- denominator branch and no arbitrary additive offset in that ratio. For transition-cost matching, initialize $l = 0, u = 1$ and perform 22 bisection iterations. At midpoint $v = (l + u)/2$, set $l = v$ if $F_{\mathcal{B}}(v) < M_k^{\mathcal{B}}$ and $u = v$ otherwise. Return $(l+u)/2$ after the final iteration. When all transition hazards are zero, return zero instead.

Table 6: Reference settings for support assessment and continuation. The two behavior-reference sample counts correspond to different interfaces.

Hyperparameter	Value
Relative-value margin, m_V	0
Relative-value temperature, τ_V	0.5
Relative-value factor floor, b_V	0
Disagreement threshold, u_0	0.25
Disagreement temperature, τ_U	0.05
Cost threshold, z_0	0.2
Cost temperature, τ_C	0.1
Value-scale lower bound, d_0	1.0
Reference-value magnitude coefficient, ζ	0.05
Reference actions for future-cost assessment	4
Reference actions for value regularization	10
Diffusion steps for future-cost reference actions	10
Action/conditional-transition score samples, L_A, L_T	$1, 1$
Support-cost normalization percentiles	$50, 95$
Normalization offset, ε	10^{-6}
Action-cost continuation, γ_A	0.25
Transition continuation lower bound, $\gamma_T^{\min}$	0
Transition continuation upper bound, $\gamma_T^{\max}$	0.25
Continuation threshold, c_0	0.5
Continuation temperature, τ_κ	0.2

Table 7: Task-specific reference presets. M, MR, and ME denote medium, medium-replay, and medium-expert. P_A, P_S are detector percentiles; L_S is the marginal-state scoring sample count.

Dataset	q_{floor}	λ_-	ν_V	$P_A = P_S$	L_S
HalfCheetah-M	-366	0.001	0.9	99	1
HalfCheetah-MR	-366	0.001	0.9	99	10
HalfCheetah-ME	-366	0.050	0.7	80	1
Hopper-M	-125	0.001	0.9	99	10
Hopper-MR	-125	0.001	0.9	99	1
Hopper-ME	-125	0.050	0.7	80	1
Walker2d-M	-471	0.010	0.9	99	10
Walker2d-MR	-471	0.001	0.9	99	1
Walker2d-ME	-471	0.050	0.7	80	1
Maze2D-umaze	0	0.001	0.9	99	1
Maze2D-medium	0	0.001	0.9	99	1
Maze2D-large	0	0.001	0.9	99	1

This is a fixed-iteration solver, not a residual-tolerance stopping rule. In exact arithmetic, its final bracket width is $2^{-22} \approx 2.38 \times 10^{-7}$; this is a coefficient-space resolution, not a guaranteed penalty-residual bound. Finite-precision arithmetic can further limit matching accuracy. Matching inputs must be finite, hazards nonnegative, and reference assessments in $[0, 1]$. The bounded transform can be evaluated as $-\operatorname{expm1}(-h)$ for numerical accuracy near zero. Coefficient computation is detached, while branch-cost derivatives with respect to actor actions are retained in the policy update.

G Analysis and Scope

The following results concern idealized support-cost operators. They do not establish policy improvement or convergence of joint neural actor–critic training.

G.1 FROZEN-COEFFICIENT SUPPORT-COST BACKUPS

Fix the policy, update index, support models, reward critics, scoring-noise and reference-action laws, and the conditional law of successors and cost-learning minibatches. The successor law may be recorded, predicted, or a fixed mixture. Write $x = (s, a)$ and let ξ include $x^+ = (s^+, a^+)$, the mask $m \in [0, 1]$, the other samples in the minibatch, and all sampling randomness.

Also fix a bounded reference cost function R_A^{ref} for computing the shared action coefficient. Let $\alpha_A(x, \xi) \in [0, 1]$ be the resulting coefficient from Eq. equation 22. It can depend on the sampled minibatch, but its rule is independent of the function R to which the operator is applied. All coefficient functions and conditional sampling kernels are measurable and fixed. For $0 \le c_A, c_T \le c_{\max}$, define

$$\begin{aligned} u_T(x, \xi) &= c_T(s, a, s^+), & b_T(x, \xi) &= m\kappa_T(s, a, s^+), \\ u_A(x, \xi) &= m\alpha_A(x, \xi)c_A(s^+, a^+), & b_A(x, \xi) &= m\alpha_A(x, \xi)\gamma_A. \end{aligned} \tag{38}$$

The actor-side transition coefficient is not part of these operators. For $M = R_{\max} > 0$, let

$$(\mathcal{T}_B^M R)(x) = \mathbb{E}_{\xi|x}\left[\Pi_M\left(u_B(x, \xi) + b_B(x, \xi)[R(x^+)]_+\right)\right], \quad \Pi_M(z) = \mathrm{clip}_{[0,M]}(z). \tag{39}$$

Clipping is inside the expectation, as in the samplewise targets. Assume

$$\beta_B := \sup_x \mathbb{E}_{\xi|x}[b_B(x, \xi)] < 1. \tag{40}$$

The bounds give $\beta_T \le \gamma_T^{\max} < 1$ and $\beta_A \le \gamma_A < 1$.

[Contraction of frozen-coefficient backups] Under these assumptions, $\mathcal{T}_B^M$ is a β_B-contraction on bounded measurable functions under the supremum norm. It has a unique fixed point in $[0, M]$. The two-branch operator is a contraction under the product supremum norm with factor $\max(\beta_A, \beta_T)$.

Proof. Clipping and the positive part are 1-Lipschitz, and $b_B \ge 0$. Therefore,

$$\begin{aligned} |(\mathcal{T}_B^M R)(x) - (\mathcal{T}_B^M S)(x)| &\le \mathbb{E}_{\xi|x}\left[b_B(x, \xi)|[R(x^+)]_+ - [S(x^+)]_+|\right] \\ &\le \mathbb{E}_{\xi|x}[b_B(x, \xi)]\|R - S\|_\infty. \end{aligned}$$

Taking the supremum gives the contraction bound. Bounded measurable functions form a complete supremum-norm space, and the operator maps it into itself with outputs in $[0, M]$. The contraction mapping theorem gives existence and uniqueness. Applying the bound to both branches proves the joint claim. □

Unclipped interpretation. Replacing Π_M by the identity retains a contraction with a unique bounded, nonnegative fixed point. Under the fixed sampling kernel,

$$R_B^{\star,\mathrm{unc}}(x_0) = \mathbb{E}\left[\sum_{t=0}^{\infty}\left(\prod_{j=0}^{t-1} b_B(x_j, \xi_j)\right) u_B(x_t, \xi_t) \,\middle|\, x_0\right]. \tag{41}$$

The empty product is one. The action branch includes next-action costs weighted by the frozen batch-dependent coefficient. Samplewise clipping generally changes the fixed point, so this path sum is not an identity for the clipped operator.

Relation to training. In training, the shared action coefficient is recomputed from current target-cost estimates. The proof does not cover an operator that recomputes this coefficient as a function of its own argument R. Stop-gradient within an update does not freeze coefficients across updates. Neural approximation, target networks, and minibatch regression differ from these idealized backups. In particular, a population Smooth-L1 minimizer need not equal a random target's conditional mean. The 1-Lipschitz property of twin-critic maximum aggregation alone does not establish convergence of neural training.

G.2 BACKUP DISCREPANCY AND PROPAGATION

Let $\mathcal{T}_B$ have fixed point $R_B^\star$, and let $\widehat{\mathcal{T}}_B$ be a fixed comparison operator on the same bounded-function domain with contraction factor $\widehat{\beta}_B < 1$. Suppose a bounded estimate $\widehat{R}_B$ satisfies

$$\begin{aligned} \|\widehat{\mathcal{T}}_B R_B^\star - \mathcal{T}_B R_B^\star\|_\infty &\le \epsilon_B, \\ \|\widehat{R}_B - \widehat{\mathcal{T}}_B \widehat{R}_B\|_\infty &\le \delta_B. \end{aligned} \tag{42}$$

[Propagation of backup discrepancy] Under these assumptions,

$$\|\widehat{R}_B - R_B^\star\|_\infty \le \frac{\delta_B + \epsilon_B}{1 - \widehat{\beta}_B}. \tag{43}$$

If $\widehat{R}_B$ is the exact fixed point of $\widehat{\mathcal{T}}_B$, then $\delta_B = 0$.

Proof. Using $\mathcal{T}_B R_B^\star = R_B^\star$,

$$\begin{aligned} \|\widehat{R}_B - R_B^\star\|_\infty &\le \|\widehat{R}_B - \widehat{\mathcal{T}}_B \widehat{R}_B\|_\infty \\ &\quad + \|\widehat{\mathcal{T}}_B \widehat{R}_B - \widehat{\mathcal{T}}_B R_B^\star\|_\infty \\ &\quad + \|\widehat{\mathcal{T}}_B R_B^\star - \mathcal{T}_B R_B^\star\|_\infty \\ &\le \delta_B + \widehat{\beta}_B \|\widehat{R}_B - R_B^\star\|_\infty + \epsilon_B. \end{aligned}$$

Rearranging proves the result. □

This perturbation bound concerns support-cost backups, not reward returns. It does not show that support scores estimate model error or bound discrepancies on OOD queries. An average empirical loss is not a certificate for the uniform residual δ_B. Lower continuation reduces the amplification factor only with the discrepancy bounds held fixed; it also changes the estimand and can change those bounds. No universal horizon choice or monotonic return improvement follows.

G.3 A CRITERION FOR INCREMENTAL PREDICTIVE EVIDENCE

Choose a common outcome $Y \in L^2$, such as a prespecified measured outcome under a fixed policy and horizon. Let X be a baseline evidence representation and Z an additional conditional-consequence representation. Hold the outcome, distribution, and horizon fixed. [Squared-loss incremental-evidence identity] Let $m_X = \mathbb{E}[Y \mid X]$ and $m_{XZ} = \mathbb{E}[Y \mid X, Z]$. Then

$$\mathbb{E}[(Y - m_X)^2] - \mathbb{E}[(Y - m_{XZ})^2] = \mathbb{E}[(m_{XZ} - m_X)^2] \ge 0. \tag{44}$$

The improvement is strict exactly when $m_{XZ} \neq m_X$ on a set of positive probability.

Proof. Decompose $Y - m_X = (Y - m_{XZ}) + (m_{XZ} - m_X)$. The cross term vanishes because $\mathbb{E}[Y - m_{XZ} \mid X, Z] = 0$. Expanding the square gives the identity. □

This identity is a criterion, not a proof that SFPR's scores are complementary or that fitted finite-sample predictors improve. If Z is a deterministic function of X, the right-hand side is zero. In particular, a deterministic score adds no oracle information when X already contains the complete (s, a) and fixed model parameters. Practical gains may instead reflect representation, inductive bias, or restricted model capacity. Conditional-mean equality is not statistical independence. The criterion supplies neither unavailable OOD outcome labels nor a causal attribution or policy-improvement guarantee.

G.4 BOUNDED CONTROL COSTS AND REMAINING SENSITIVITY

$\Phi(H) = 1 - e^{-H}$ is monotone, bounded by one, and 1-Lipschitz on $H \ge 0$. Consider two sets of branch estimates on the same policy minibatch, with errors bounded by δ_A, δ_T. For the same fixed numerical coefficient $w = \bar{w}_{T,k}^{\mathcal{B}}$, each sample satisfies

$$|C_{\mathrm{SFPR},k}^{\mathcal{B}} - \widehat{C}_{\mathrm{SFPR},k}^{\mathcal{B}}| \le \lambda(\eta_k \delta_A + w \delta_T). \tag{45}$$

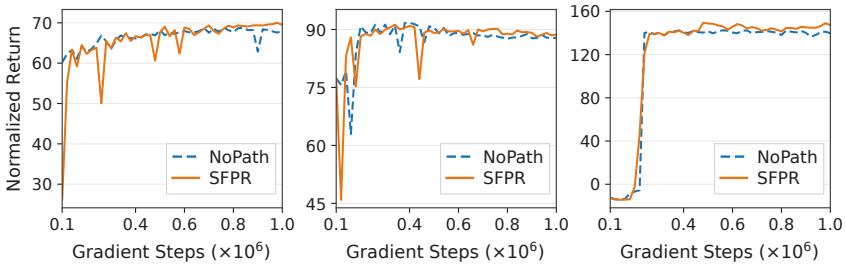

Figure 3: Additional learning curves of SFPR and NoPath on HalfCheetah-medium, Walker2d-medium, and Maze2D-umaze. Curves show individual runs.

This follows from the 1-Lipschitz properties of Φ and the positive part. If the coefficient is recomputed, let its two values be $w, \widehat{w} \in [0, 1]$. If positive-part transition estimates are bounded by M_T on the batch, then

$$|C^{\mathcal{B}}_{\mathrm{SFPR},k} - \widehat{C}^{\mathcal{B}}_{\mathrm{SFPR},k}| \le \lambda(\eta_k \delta_A + \delta_T + M_T |w - \widehat{w}|). \tag{46}$$

Indeed, for $r = [R_T]_+$ and $\widehat{r} = [\widehat{R}_T]_+$, $|wr - \widehat{w}\widehat{r}| \le w|r - \widehat{r}| + \widehat{r}|w - \widehat{w}| \le \delta_T + M_T|w - \widehat{w}|$. The bound isolates coefficient recomputation but does not control $|w - \widehat{w}|$ using branch errors alone. Clipped targets do not guarantee the assumed output bound. A bounded penalty also does not bound actor gradients or ensure correction of Q overestimation.

H Additional Ablation Results

Figure 3 provides the additional SFPR–NoPath learning curves on HalfCheetah-medium, Walker2d-medium, and Maze2D-umaze. The individual-run curves complement the main-text comparisons by showing the evolution of normalized return over the training budget. NoPath removes the additional actor support-cost penalty while retaining behavior-distribution fitting and selective value regularization. These curves are not averages over independent training runs.