PREDICTIVE SIGNATURE DIVERSITY FOR MODEL-BASED OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline reinforcement learning (offline RL) aims to learn policies from static datasets, but suffers from distributional shift when reaching states not represented by the data. Model-based approaches mitigate this issue using learned dynamics models, but are highly sensitive to model uncertainty, especially in regions less covered by the dataset. Existing approaches fail to adequately capture such uncertainty, particularly in terms of worst-case discrepancies between potential models. We propose a predictive, policy-independent framework for model selection based on *predictive signature diversity*. Models are differentiated by their predictions on shared anchor inputs, enabling comparison of behaviors in a common space. We encourage diversity during model selection by selecting a subset that maximizes worst-case separation of their predictions, yielding a dynamically updated ensemble for policy learning via diverse rollout generation. This decouples diversity estimation from policy rollouts and directly captures model disagreement in uncertain regions. Our empirical evaluation shows improved robustness to model uncertainty over a broad range of tasks compared to state-of-the-art baselines.

1 INTRODUCTION

Offline reinforcement learning (offline RL) aims to learn effective policies from a fixed dataset of previously collected environment interactions. In contrast to online RL, the learner cannot actively gather additional experience and must rely entirely on data collected by one or more agents whose behavior policies are often unknown. This setting is crucial for real-world applications such as robotics, healthcare, and autonomous systems, where learning on-the-go is costly, unsafe, or impractical.

Offline RL presents fundamental challenges. Since the dataset is collected by unknown behavior policies, the coverage and quality of observed state–action pairs are determined by the data collection process and may be limited or biased. As a result, the learned policy will likely explore actions or encounter regions of the state space that are not well represented in the data, leading to unreliable value estimates and compounding errors due to distributional shift. Model-based offline RL mitigates this issue by learning dynamics models from the dataset and generating synthetic rollouts for policy improvement. However, learned models are inherently uncertain, especially in regions with limited data coverage, and model errors can severely degrade policy performance.

A common strategy to address model uncertainty is to use an ensemble of dynamics models. Prior work leverages multiple models to capture epistemic uncertainty Lakshminarayanan et al. (2017) and mitigate model bias in model-based RL, as demonstrated by MBPO Janner et al. (2019). In the offline setting, methods such as MOReL Kidambi et al. (2020) and MOPO Yu et al. (2020) use ensemble-based uncertainty estimates to mitigate the exploitation of inaccurate learned dynamics. More recently, Model-based Diverse Adaptive Policy Learning (MoDAP Choi et al. (2024)) explicitly promotes diversity among dynamics models through an information-theoretic objective defined over policy-induced trajectories. While effective in some settings, this diversity measure depends on trajectories generated by the current adaptive policy and optimizes an expected information-theoretic objective which can cause extreme deviations between individual models. In contrast, our approach explicitly targets worst-case pairwise discrepancies among model predictions.

In this work, we propose **PS-DAP**, a novel model-selection framework for model-based offline RL based on *predictive signature diversity*. Rather than characterizing model diversity through

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.

trajectories induced by the current policy, PS-DAP evaluates candidate dynamics models on shared anchor states and action sequences sampled from the offline dataset. Each model is represented by the resulting multi-step sequence of predicted states and rewards, which we define as its *predictive signature*. Evaluating all models on the same anchor inputs provides a direct, comparable, and policy-independent characterization of their predictive behavior.

PS-DAP quantifies model diversity through distances between predictive signatures and selects a subset of models using a max–min criterion. Specifically, for each candidate model, we measure its distance to the closest model already selected and choose the candidate that maximizes this minimum distance. This criterion discourages the selection of models that are distinct from some members of the selected set but remain redundant with others. Consequently, each newly selected model is encouraged to contribute predictive behavior that is distinct relative to the entire selected subset. This anchor-based formulation provides a notion of diversity tailored to the offline setting and enables model selection without relying on policy-induced trajectory sampling.

Our approach offers several advantages. First, it decouples diversity estimation from the evolving policy, providing a more stable and directly comparable basis for model selection. Second, the max–min criterion explicitly promotes separation across the selected ensemble rather than relying on an average measure of disagreement, reducing redundancy among selected models. Third, PS-DAP can be incorporated into existing model-based offline RL pipelines by modifying the model-selection and refinement stages without changing the underlying policy-learning algorithm.

The contributions of this work are as follows:

- **Predictive Signature Diversity.** We introduce *predictive signatures*, a policy-independent representation of dynamics-model behavior based on multi-step state and reward predictions over shared anchor inputs, providing a direct basis for comparing ensemble members in model-based offline RL.
- **PS-DAP: predictive diversity-aware model selection.** We propose **PS-DAP**, an anchor-based model-selection framework that uses a max–min predictive diversity criterion to construct subsets of models with strong worst-case separation. By maximizing each selected model's minimum distance from the current subset, PS-DAP promotes distinct predictive behaviors while reducing redundancy among ensemble members.
- **Seamless integration into model-based offline RL.** We show that PS-DAP can be integrated into existing model-based offline RL pipelines through modifications to the ensemble selection and refinement stages, without requiring policy-dependent trajectories for diversity estimation or changes to the underlying policy-optimization procedure.
- **Extensive empirical validation.** We evaluate PS-DAP on standard D4RL benchmarks and demonstrate strong performance relative to existing baselines. We further examine the contribution of the diversity regularizer and model-selection strategy, examine sensitivity to the real-to-synthetic data ratio and rollout horizon, and study learning dynamics across the evaluated domains.

2 Related Work

Offline and Model-Based Reinforcement Learning. Offline reinforcement learning has received significant attention due to its applicability in settings where online interaction is limited or unsafe Levine et al. (2020). A central challenge in offline RL is distributional shift, where the learned policy may encounter state-action pairs that are insufficiently represented in the dataset, leading to unreliable value estimates Fujimoto et al. (2019); Kumar et al. (2020). To address this issue, prior work has explored conservative value estimation, such as CQL Kumar et al. (2020); behavior-constrained or regression-based policy learning, such as BCQ Fujimoto et al. (2019) and CRR Wang et al. (2020); and uncertainty-aware methods that discourage actions in poorly supported regions.

Model-based approaches instead learn a dynamics model and use model-generated transitions for policy optimization. MBPO Janner et al. (2019) demonstrated the effectiveness of short model-generated rollouts, a mechanism later adopted by offline model-based methods such as MOPO Yu et al. (2020). MOPO and MOReL Kidambi et al. (2020) further account for model inaccuracies in the fully offline setting through uncertainty penalties or pessimistic treatment of unreliable model

predictions. These methods highlight the importance of controlling model error when policy learning relies on learned dynamics. In contrast, our work focuses on the composition of the dynamics ensemble itself, aiming to preserve predictive accuracy while selecting models that exhibit distinct predictive behaviors.

Ensemble Methods and Model Uncertainty. Ensembles of dynamics models are widely used to represent epistemic uncertainty in model-based RL Janner et al. (2019); Kidambi et al. (2020); Yu et al. (2020); Chua et al. (2018). By maintaining multiple plausible models consistent with the data, ensembles can support more robust planning and policy learning. Different methods use the resulting uncertainty in different ways. PETS Chua et al. (2018) propagates uncertainty across predicted trajectories during planning, MOPO Yu et al. (2020) penalizes rewards in regions where the learned dynamics are uncertain, and MOReL Kidambi et al. (2020) discourages the policy from entering regions in which the learned model is considered unreliable. The effectiveness of these approaches depends not only on the predictive accuracy of individual models but also on whether the ensemble captures sufficiently distinct plausible dynamics.

Adaptive Policies in Offline Model-Based RL. MAPLE (Model-based Adaptable Policy LEarning) proposes learning policies that adapt at deployment time by conditioning on recent transition history Chen et al. (2021). Instead of learning a fixed policy, MAPLE uses a latent representation of observed transitions, implemented through a recurrent architecture, to adjust behavior according to the inferred dynamics. This line of work highlights the importance of adaptivity in offline RL. However, MAPLE does not explicitly optimize the diversity of the dynamics-model ensemble. Our work is complementary: we focus on the model selection stage and propose a predictive notion of diversity that can be integrated into adaptive-policy frameworks such as MAPLE.

Diversity in Model-Based RL. Recent work has placed greater emphasis on the diversity of learned dynamics models. In particular, MoDAP Choi et al. (2024) aims to construct an effective set of dynamics models under a limited ensemble size. MoDAP promotes diversity through an information-theoretic objective based on the mutual information between policy-induced trajectories and the identity of the underlying dynamics model, encouraging different models to induce distinguishable behaviors while using an adaptive policy conditioned on past transitions.

PS-DAP differs in how model diversity is defined and measured. Rather than estimating diversity through trajectories generated by the current policy, PS-DAP evaluates all candidate models on the same anchor states and action sequences sampled from the offline dataset. The resulting multi-step state and reward predictions form each model's *predictive signature*, enabling direct and policy-independent comparison between models. PS-DAP then applies a max–min selection criterion that explicitly promotes separation from the closest already-selected model, reducing redundancy within the selected ensemble.

3 PRELIMINARIES

Offline Model-Based Reinforcement Learning. We consider a Markov Decision Process (MDP) Bellman (1957) with state space $\mathcal{S}$, action space $\mathcal{A}$, transition dynamics $P(s' \mid s, a)$, and reward function $r(s, a)$, and discount factor γ. In offline reinforcement learning, the agent is given a fixed dataset $\mathcal{D}_{\text{off}} = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^{N}$, where $N \in \mathbb{N}_{>0}$, collected by one or more unknown behavior policies, without any further interaction with the environment. The goal is to learn a policy π that achieves high expected return under the true dynamics P using only $\mathcal{D}_{\text{off}}$. Model-based offline RL methods address this problem by learning an approximate transition model $\hat{P}_\phi(s' \mid s, a)$ from the offline data and using it to generate synthetic transitions for policy optimization. However, approximation errors in $\hat{P}_\phi$ may accumulate over multi-step rollouts, especially in regions with limited data support, leading to distributional shift when the learned policy queries out-of-distribution state-action pairs.

Ensemble Dynamics Models. An ensemble of models $\{P_{\phi_m}\}_{m=1}^{n}$, trained independently on $\mathcal{D}_{\text{off}}$ under different initializations and data perturbations, represents multiple dynamics hypotheses consistent with the data. Disagreement among models serves as a proxy for epistemic uncertainty and can be leveraged during policy learning Lakshminarayanan et al. (2017).

Adaptive Policy Learning Following MAPLE Chen et al. (2021), the policy $\pi_\theta(a \mid s', z)$ and its critic are conditioned on a latent variable z produced by a recurrent Gated Recurrent Unit (GRU)

encoder Chung et al. (2014) over recent transitions, with $z_{t+1} = GRU_\psi(s_{t+1}, a_t, z_t)$. This recurrent representation allows the policy to infer and adapt to the dynamics model underlying the current rollout. The policy, critic, and encoder are trained using Soft Actor-Critic (SAC) Haarnoja et al. (2018) on a mixture of transitions from $\mathcal{D}_{\text{off}}$ and synthetic transitions generated by the learned dynamics models.

4 Predictive Signature Diverse Adaptive Policy Learning

We address the challenge of capturing model uncertainty in offline reinforcement learning by introducing **PS-DAP** (Predictive Signature Diverse Adaptive Policy Learning), a predictive, policy-independent framework for model selection based on predictive signatures. Instead of evaluating models through trajectories generated by a policy, we compare their multi-step predictions on a shared set of anchor inputs derived from the offline dataset. Each model is thus represented by a predictive signature, enabling direct comparison of model behaviors under identical state–action sequences. We select a subset of models by maximizing worst-case pairwise separation in this predictive space, subject to a loss-based constraint on the dataset. This yields a compact set of models that exhibit low prediction loss and are diverse in their predicted dynamics, providing a robust basis for model-based policy learning.

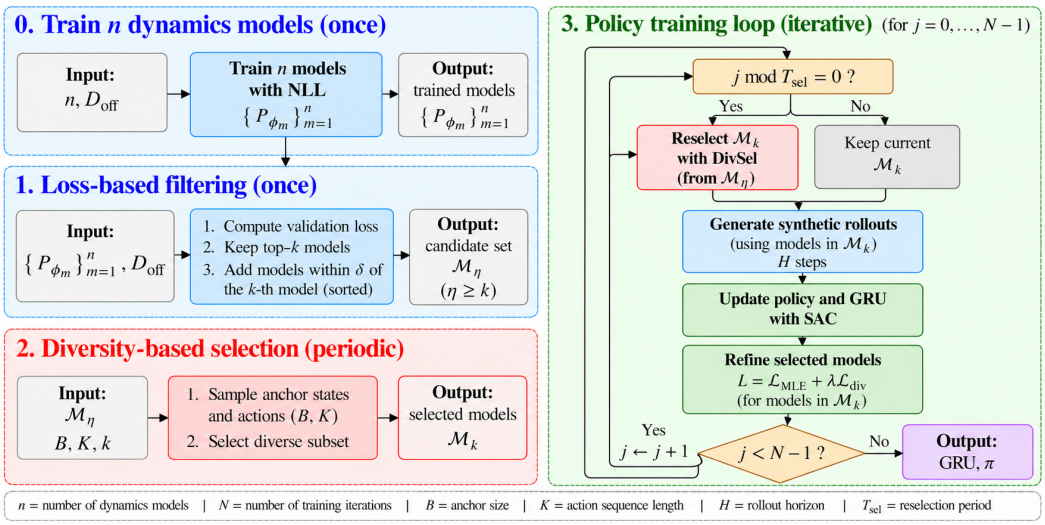


Figure 1: Overview of the PS-DAP framework, including ensemble filtering, diversity-based model selection, and adaptive policy training.

An overview of the complete PS-DAP framework is shown in Fig. 1. The method first trains an ensemble of dynamics models and filters it according to validation prediction loss. From the resulting candidate set, PS-DAP periodically selects a subset of models using predictive signature diversity. The selected models are used to generate synthetic rollouts for policy learning and are subsequently refined using a diversity-regularized objective. Detailed pseudocode for the individual components is provided in Appendix C.

As illustrated in Fig. 1, PS-DAP consists of four main components: (i) **ensemble training**, where n dynamics models are learned from $\mathcal{D}_{\text{off}}$; (ii) **loss-based filtering**, which constructs a candidate set $\mathcal{M}_\eta$ by retaining low-loss models using a top-k criterion with tolerance (Algorithm 1 in Appendix C.1); (iii) **diversity-based selection**, where DivSel selects a subset $\mathcal{M}_k \subseteq \mathcal{M}_\eta$ according to predictive diversity (Algorithm 2 in Appendix C.2); and (iv) **policy training**, where the selected models generate synthetic rollouts for SAC-based policy learning and are refined using the diversity-regularized objective, with $\mathcal{M}_k$ periodically recomputed during training (Algorithm 3 in Appendix C.3).

Component 0: Learning an Ensemble of Dynamics Models. Component 0 in Fig. 1 trains an ensemble of n parameterized dynamics models $\{P_{\phi_m}\}_{m=1}^n$ on the offline dataset $\mathcal{D}_{\text{off}}$. We split

$\mathcal{D}_{\text{off}}$ into 80% training data and 20% validation data, and optimize each model using the maximum-likelihood objective

$$\mathcal{L}_{\text{MLE}}(\phi_m) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}_{\text{off}}}\left[-\log P_{\phi_m}(s', r \mid s, a)\right].$$

This component yields an initial ensemble of n candidate models that fit the observed data but may differ substantially in regions where the dataset provides limited coverage.

Component 1: Loss-Based Filtering. Component 1 in Fig. 1 filters the learned ensemble according to validation prediction loss. The 20% validation split is used to estimate predictive accuracy for model filtering, and we assume that the initial ensemble contains sufficiently many plausible models such that at least k models remain after filtering. The detailed filtering procedure is provided in Algorithm 1 in Appendix C.1.

For each model P_{ϕ_m}, we compute the mean squared error on a validation $\mathcal{D}_{\text{val}}$, $L_{\text{MSE}(\phi_m)} = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}_{\text{val}}}\left[\|\hat{s}'_m - s'\|_2^2 + (\hat{r}_m - r)^2\right]$, where $(\hat{s}'_m, \hat{r}_m) = \mu_{\phi_m}(s,a)$ is the mean model prediction. This deterministic error metric provides a consistent basis for comparing models by predictive accuracy, independent of their predicted uncertainty.

Rather than keeping only the strict top-k models, we retain all models whose loss is within a tolerance δ of the k-th best model. This gives

$$\mathcal{M}_\eta = \{m \in \{1,\dots,n\} : L_{\text{MSE}}(\phi_m) \le L_{\text{MSE}}(\phi_k) + \delta\}.$$

Thus, $|\mathcal{M}_\eta| \ge k$, and the resulting candidate set contains models with comparable predictive accuracy while preserving the opportunity for subsequent diversity-based selection.

Component 2: Diversity-Based Subset Selection (DivSel). Component 2 in Fig. 1 performs diversity-based subset selection. From the filtered candidate set $\mathcal{M}_\eta$, we select a subset of k models that are maximally separated in predictive space in a procedure we call DivSel. The detailed procedure is provided in Algorithm 2 in Appendix C.2. First, we sample anchor states $\{s_b\}_{b=1}^B \sim D_{\text{off}}$ and shared action sequences $\{a_b^{(0)},\dots,a_b^{(K-1)}\}_{b=1}^B$, which are used for all filtered models. For each model $P_{\phi_m} \in \mathcal{M}_\eta$ and anchor b, we generate a K-step rollout starting from $\hat{s}_{m,b}^{(0)} = s_b$ using the model mean prediction μ_{ϕ_m}:

$$(\hat{s}_{m,b}^{(t+1)}, \hat{r}_{m,b}^{(t)}) = \mu_{\phi_m}\left(\hat{s}_{m,b}^{(t)}, a_b^{(t)}\right), \qquad t = 0,\dots,K-1.$$

Let $\{(\tilde{s}_{m,b}^{(t+1)}, \tilde{r}_{m,b}^{(t)})\}_{t=0}^{K-1}$ denote the z-score normalized predicted states and rewards with anchor b relative to statistics from the offline dataset. The *predictive signature* of model m for anchor b is defined as:

$$g_{m,b} = \left[\sqrt{w_0}\,\tilde{s}_{m,b}^{(1)}, \sqrt{w_0}\,\beta\tilde{r}_{m,b}^{(0)}, \dots, \sqrt{w_{K-1}}\,\tilde{s}_{m,b}^{(K)}, \sqrt{w_{K-1}}\,\beta\tilde{r}_{m,b}^{(K-1)}\right], \tag{1}$$

The parameters $w_t \ge 0$ are temporal weights, indicating the significance of each time step. The parameter $\beta > 0$ determines the significance of state discrepancies relative to reward discrepancies. Using the predictive signatures for all anchors, we compute the pairwise distance between models i and ℓ:

$$\begin{aligned} d(i,\ell) &= \frac{1}{B}\sum_{b=1}^{B} \|g_{i,b} - g_{\ell,b}\|_2 \\ &= \frac{1}{B}\sum_{b=1}^{B}\left(\sum_{t=0}^{K-1} w_t\left(\|\tilde{s}_{i,b}^{(t+1)} - \tilde{s}_{\ell,b}^{(t+1)}\|_2^2 + \beta^2\left|\tilde{r}_{i,b}^{(t)} - \tilde{r}_{\ell,b}^{(t)}\right|^2\right)\right)^{1/2} \end{aligned} \tag{2}$$

Since all models are evaluated from identical anchors and action sequences, differences in rollout predictions reflect only differences in learned dynamics.

Finally, DivSel selects the subset whose minimum pairwise predictive distance is maximized:

$$\mathcal{M}_k \leftarrow \arg\max_{|\mathcal{I}|=k} \min_{i\neq\ell\in\mathcal{I}} d(i,\ell).$$

Component 3: Policy Learning with Periodic Subset Updates. Component 3 in Fig. 1 performs policy learning using rollouts generated from a dynamically maintained subset $\mathcal{M}_k \subseteq \mathcal{M}_\eta$. The

detailed training procedure is provided in Algorithm 3 in Appendix C.3. At iteration j, the subset is periodically recomputed via DIVSEL every T_{sel} steps and otherwise reused across iterations. Synthetic data is generated by sampling an initial state from D_{off} and initializing the recurrent latent state z_0. The GRU then encodes the recent state–action history into z_t, and the policy conditions each action on both the current state and this latent representation, i.e., $\pi_\theta(a_t \mid s_t, z_t)$. A model $P_m \sim \text{Uniform}(\mathcal{M}_k)$ is then sampled. A rollout of horizon H is constructed, where actions are sampled from the policy, transitions from the selected model, and the recurrent state is updated via the GRU. The resulting transitions are aggregated into D_{P_ϕ}. The policy and critic are updated using SAC on the combined dataset $D_{\text{off}} \cup D_{P_\phi}$. This process ensures that policy learning is guided by a subset of models that is both accurate (via filtering) and diverse (via periodic reselection).

Model Refinement with Diversity Regularization. As part of Component 3 in Fig. 1, we update the selected subset $\mathcal{M}_k$ using a joint objective that combines maximum-likelihood estimation with a diversity regularizer specifically introduced in this work to preserve predictive separation during training. The refinement procedure is included in Algorithm 3 in Appendix C.3:

$$\mathcal{L}_{\text{div}} = \sum_{\substack{i<\ell \\ i,\ell \in \mathcal{M}_k}} \max\big(0, \epsilon - d(i,\ell)\big), \tag{3}$$

and optimize

$$\mathcal{L} = \sum_{m \in \mathcal{M}_k} \mathcal{L}_{\text{MLE}}(\phi_m) + \lambda \text{div} \mathcal{L}_{\text{div}}. \tag{4}$$

The diversity loss in Eq. (3) is introduced only during the refinement stage through the total objective in Eq. (4), and is not used during the initial ensemble training. Initially, all dynamics models are trained solely with the maximum-likelihood objective to establish predictive accuracy with respect to the offline data before encouraging separation between ensemble members. Applying the diversity objective from the beginning could promote disagreement before reliable dynamics are learned, causing the separation to reflect prediction error rather than meaningful alternative dynamics.

During refinement, updates are restricted to the currently selected subset $\mathcal{M}_k \subseteq \mathcal{M}_\eta$, since these models are used for rollout generation and directly influence policy learning. At each refinement update, a new batch of B anchor states and shared action sequences of length K is sampled from D_{off} and used for all models in $\mathcal{M}_k$. In Eq. (3), $d(i,\ell)$ denotes the predictive distance between models i and ℓ, and the hinge term $\max(0, \epsilon - d(i,\ell))$ penalizes only pairs whose distance falls below the margin ϵ.

The maximum-likelihood term keeps each model grounded in observed transitions, while the diversity regularizer preserves predictive separation among the selected models. In practice, λ is chosen such that the data-fitting objective remains dominant on in-distribution samples, limiting divergence from the offline data. Models outside $\mathcal{M}_k$ remain unchanged and may be reselected in later iterations. By maintaining diversity among the models used for synthetic data generation, the refinement exposes the policy to a broader range of plausible dynamics and can improve the robustness of policy updates.

Summary of Ensemble Reduction. The reduction process is: $n \longrightarrow |\mathcal{M}_\eta| \longrightarrow k$. We begin with n learned models, filter them according to prediction loss to obtain a candidate set $\mathcal{M}_\eta$, and repeatedly select a subset of k models from this set based on predictive diversity for rollout generation and policy learning. This subset is dynamically updated every T_{sel} iterations, allowing the policy to be trained under a continually refreshed set of well-fitting and behaviorally diverse dynamics.

5 EMPIRICAL EVALUATION

We examine whether PS-DAP outperforms prior methods for model-based offline RL under the similar configurations. To that end, we evaluate PS-DAP on standard offline RL benchmarks and compare it to state-of-the-art methods for offline RL in peak performance as well as performance throughout training.

Table 1: Normalized scores on D4RL datasets Fu et al. (2021).

Environment	Dataset	Model-free				Model-based		
		SAC	BEAR	BC	CQL	MAPLE	MOPO	PS-DAP (Ours)
Walker2d	random	4.1	6.7	9.8	7.0	21.8 ± 0.04	13.6 ± 2.6	**25.3 ± 2.5**
Walker2d	medium	0.9	33.2	6.6	79.2	65.5 ± 10.1	11.8 ± 19.3	**91.6 ± 2.6**
Walker2d	medium-replay	3.5	25.3	11.3	26.7	68.7 ± 7.6	39.0 ± 9.6	**95.3± 7.9**
Walker2d	medium-expert	-0.1	26.0	6.4	**111.0**	78.3 ± 15.7	44.6 ± 12.9	**109.6± 3.02**
HalfCheetah	random	30.5	25.5	2.1	35.4	**40.2 ± 2.2**	35.4 ± 1.5	36.7 ± 1.9
HalfCheetah	medium	-4.3	38.6	36.1	44.4	51.9 ± 3.1	42.3 ± 1.6	**71.2 ±0.9**
HalfCheetah	medium-replay	-2.4	36.2	38.4	46.2	64.9 ± 1.9	53.1 ± 2.0	**66.9 ± 3.9**
HalfCheetah	medium-expert	1.8	51.7	35.8	62.4	73.5 ± 6.1	63.3 ± 38.0	**93.3 ± 8.9**
Hopper	random	11.3	9.5	1.6	10.8	10.9 ± 0.4	**11.7 ± 0.4**	8.6 ± 1.5
Hopper	medium	0.8	47.6	29.0	58.0	18.6 ± 4.9	28.0 ± 12.4	**104.7 ± 0.1**
Hopper	medium-replay	1.9	10.8	11.8	48.6	92.1 ± 1.3	67.5 ± 24.7	**98.8 ± 2.7**
Hopper	medium-expert	1.6	4.0	**111.9**	98.7	31.7 ± 1.7	23.7 ± 6.0	102.3 ± 6.3

Domains We evaluate our method and the baselines on continuous control tasks from the D4RL suite Fu et al. (2021). Similarly to prior work, we focus on the `Hopper`, `Walker2d`, and `HalfCheetah` MuJoCo environments Todorov et al. (2012), where the objective is to control a unipedal, bipedal, and quadrapedal agent to travel as far as possible in one direction within the time limit, respectively. The growing complexity of the agent's degrees of freedom relates directly to the difficulty of fitting a dynamics model for that environment.

For each of the tested domains, the offline data was collected using one of four behavior policies. `random` uses a random policy, `medium` uses a partially trained SAC policy, `medium-replay` uses multiple checkpoints of the partially trained SAC policy, and `medium-expert` uses a partially trained and a fully trained SAC policy. Each behavior policy exposes the underlying model differently, e.g., `medium-expert` mostly exposes the parts required to complete the task, while `random` exposes mostly information that is irrelevant for the task.

Experimental Setup All dynamics models and policies are trained purely offline using the provided datasets, without any additional environment interaction. Transition models are first pretrained on the offline dataset prior to policy learning, and are further updated during training using offline data. The policy is evaluated on the environment itself throughout training. Performance is evaluated using normalized return in the true environment following the standard D4RL evaluation protocol.

The policy itself is optimized using a mixture of transitions from the offline dataset and synthetic transitions generated by the learned dynamics models. Policies are trained for 1000 policy update iterations, and results are reported as the mean and standard deviation over three random seeds. In our method, we train an ensemble of 10 dynamics models and initially select the $k = 6$ models with the lowest predictive loss on the offline dataset. Additional models whose predictive loss falls within a predefined threshold of the selected set are included in the candidate pool for diversity-aware selection. We use a rollout horizon of 10 for both predictive-signature evaluation and synthetic rollout generation. For MAPLE, we follow the original implementation settings, where 20 dynamics models are trained for each task and 14 models are selected as the ensemble used for policy learning. Additional implementation details and computational resource specifications are provided in Appendix A and Appendix E, respectively.

Baselines We compare our method against representative offline RL baselines from both model-free and model-based categories. The model-free baselines include BC: Imitate the behavior policy via supervised learning, BEAR Kumar et al. (2019), SAC Haarnoja et al. (2018), and CQL Kumar et al. (2020). The model-based baselines include MOPO Yu et al. (2020), MAPLE Chen et al. (2021), and MoDAP Choi et al. (2024)[1]. Our implementation builds upon the MoDAP codebase.

[1]We note that despite extensive reproduction efforts, including correspondence with the authors, we consistently obtained lower performance for MoDAP than reported in Choi et al. (2024). We report MoDAP results obtained using the same configurations as PS-DAP; see Appendix B for more details on our reproduction efforts.

Results: As shown in Table 1, PS-DAP outperforms the baseline methods across most environments and dataset regimes. The gains are particularly pronounced in the `Walker2d` and `HalfCheetah` domains, where PS-DAP achieves the highest returns on most dataset variants. Even when PS-DAP does not achieve the highest mean score, it remains highly competitive with the best-performing method while outperforming the other model-based approaches. For example, on `walker2d-medium-expert`, PS-DAP achieves 109.6 ± 3.02, compared with 111.0 obtained by the model-free CQL method, while substantially outperforming the model-based MAPLE and MOPO baselines. A similar result is observed on `hopper-medium-expert`, where PS-DAP remains close to the best-performing baseline while outperforming the other model-based methods. Overall, these results show that PS-DAP achieves consistently strong performance across the evaluated settings, attaining the highest score in most cases and remaining competitive with strong model-free baselines otherwise.

We additionally report the learning curve of our reproduced MoDAP implementation on `walker2d-random` under the same 1000-epoch training budget used for PS-DAP. As shown in Fig. 2, PS-DAP achieves higher returns than the reproduced MoDAP implementation. A detailed discussion of the MoDAP reproduction, including extended training results, is provided in Appendix B.

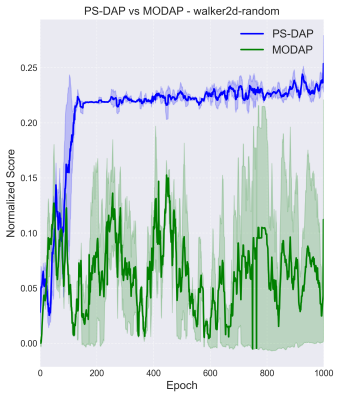


Figure 2: PS-DAP vs. reproduced MoDAP on `walker2d-random`.

All learning curves comparing PS-DAP with MAPLE across the D4RL domains and dataset settings are provided in Appendix D.1, along with a detailed discussion of their convergence behavior, stability, and relative performance throughout training.

Discussion:
Our results demonstrate that PS-DAP achieves strong performance relative to SOTA baseline methods across the evaluated benchmarks, attaining the highest score in most settings and remaining competitive with the best-performing methods otherwise. In particular, PS-DAP consistently outperforms the other model-based baselines, supporting the effectiveness of combining loss-based filtering with diversity-driven model selection.

The results further highlight the importance of maintaining predictive diversity within the dynamics ensemble. Loss-based filtering first restricts the candidate set to models that remain consistent with the offline data, while diversity-based selection identifies a compact subset with distinct predictive behaviors. Periodically refining the selected models using the combined maximum-likelihood and diversity objectives further preserves this separation while maintaining predictive accuracy. Rather than estimating diversity from policy-induced trajectories, PS-DAP evaluates disagreement directly through predictive signatures, allowing model selection to reflect differences in learned dynamics that are relevant to the synthetic data used for policy training.

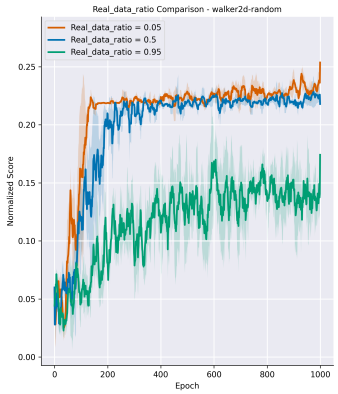


Figure 3: Real-to-synthetic data ratio on `walker2d-random`.

To further examine the role of synthetic data, we evaluate three real-data ratios on the `walker2d-random` dataset, as shown in Fig. 3. The results show a clear dependence on the composition of the policy-training data. When the real-data ratio is increased to 0.95, performance degrades substantially, as policy learning relies predominantly on transitions from the fixed offline dataset, whose coverage is limited and may provide insufficient information outside well-represented regions of the state-action space. Reducing the real-data ratio to 0.5 leads to considerably stronger and more balanced performance. However, the default ratio of 0.05 achieves the strongest results, converging faster and attaining the highest final performance among the three settings. This suggests that, when the learned dynamics models are sufficiently accurate and diverse, model-generated transitions can provide useful additional coverage beyond the offline dataset and improve policy learning. These results further motivate our emphasis on maintaining both predictive fidelity and diversity among the models used for synthetic

data generation. Additional ablation and sensitivity studies examining other components and design choices of PS-DAP are presented in Appendix D.2.

6 CONCLUSION

We showed that diversity in dynamics model ensembles can improve policy learning in offline RL, particularly in poorly covered and out-of-support regions of the dataset. This is especially important in settings where online interaction is unavailable and the policy must be learned entirely from a fixed dataset. In such cases, the training data becomes a central limiting factor. Regardless of the strength of the policy optimizer, data that is biased, low-quality, or insufficient can fundamentally restrict the behaviors to which the learned policy can generalize.

This consideration is particularly relevant in our setting, where 95% of the policy-training data is synthesized from learned dynamics models and only 5% is sampled directly from the offline dataset. Consequently, the quality, coverage, and diversity of the synthesized transitions directly determine the range of states, actions, and behaviors on which the policy is trained. If the ensemble members collapse to similar predictions, the synthetic data provides only a narrow extension of the original dataset, limiting the policy's ability to generalize beyond well-covered regions. In contrast, generating transitions from models with diverse yet plausible predictive behaviors can expose the policy to a broader range of dynamics, including regions that are weakly represented or absent in the offline data, thereby improving its robustness to distribution shift.

Our approach therefore encourages the policy to train on models with diverse predictive behaviors, while constraining model selection according to prediction loss in order to preserve consistency with the offline data distribution. This balance is crucial: predictive diversity broadens the effective support of the training data and increases exposure to plausible out-of-distribution behaviors, while predictive accuracy limits drift toward unrealistic transitions. Experimental results on standard D4RL benchmarks demonstrate that jointly promoting predictive diversity and model fidelity can improve policy robustness, generalization, and overall performance across multiple environments and dataset settings.

Limitations: Despite its advantages, the proposed approach has several limitations. First, periodic model reselection and diversity-based dynamics updates introduce additional computational overhead compared to standard ensemble methods. Second, performance is sensitive to several hyperparameters, requiring careful tuning to achieve stable training. Third, experiments were conducted only on standard D4RL benchmarks, and evaluation on more diverse or higher-dimensional domains remains limited. In particular, PS-DAP depends on a trade-off between diversity estimation, computational efficiency, and training stability. The reselection interval T_{sel} controls how frequently the rollout ensemble is updated. Small values allow faster adaptation to improved diversity estimates but may destabilize policy learning due to rapidly changing synthetic data distributions, whereas large values reduce the benefits of adaptive selection. In practice, reselection is performed only when the filtered candidate set satisfies $|\mathcal{M}_\eta| > k$, since otherwise the filtered set itself is used directly for rollout generation. Training stability is also affected by the frequency and magnitude of dynamics model updates.

Future Work: Future work includes evaluating PS-DAP on broader offline RL benchmarks and more complex environments, as well as exploring adaptive diversity mechanisms that dynamically balance diversity and predictive accuracy during training. Another promising direction is extending predictive-signature diversity to uncertainty-aware planning and online model-based reinforcement learning.

Impact: Our primary contribution is methodological. While the proposed approach may be applicable to domains such as robotics and autonomous systems, this work does not introduce new application-specific capabilities. We thus do not anticipate immediate negative societal impacts arising directly from this research.

AI Use Statement. Generative AI tools were used to assist with editing and improving the clarity and presentation of the manuscript. All technical content, methodology, experiments, results, and conclusions were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REFERENCES

Richard Bellman. A Markovian Decision Process. *Indiana University Mathematics Journal*, 6(4): 679–684, 1957. ISSN 0022-2518. doi: 10.1512/iumj.1957.6.56038.

Xiong-Hui Chen, Yang Yu, Qingyang Li, Fan-Ming Luo, Zhiwei Qin, Wenjie Shang, and Jieping Ye. Offline model-based adaptable policy learning. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021.

Yunseon Choi, Li Zhao, Chuheng Zhang, Lei Song, Jiang Bian, and Kee-Eung Kim. Diversification of adaptive policy for effective offline reinforcement learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, pp. 3863–3871, Jeju, South Korea, August 2024. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/427. URL https://www.ijcai.org/proceedings/2024/427.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, Montréal, Canada, 2018.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*, 2014.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning, February 2021. URL https://arxiv.org/abs/2004.07219.

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, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, Stockholm, Sweden, 2018.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada, 2020.

Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada, 2020.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, November 2020. URL https://arxiv.org/abs/2005.01643.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Proceedings of the 24th IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, Vilamoura, Portugal, 2012.

Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S. Merel, Jost Tobias Springenberg, Scott E. Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, and Nando de Freitas. Critic regularized regression. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada, 2020.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada, 2020.

A IMPLEMENTATION DETAILS

Table 2 summarizes the neural network architecture and optimization hyperparameters inherited from the original MAPLE implementation. To ensure a fair comparison and isolate the effect of the proposed diversity-aware model selection mechanism, we retain the same backbone architecture and optimizer settings used in MAPLE for the dynamics models, actor, and critic networks.

Table 2: Neural network hyperparameters .

Hyperparameter	Value
Dynamics hidden layers	4
Dynamics hidden units	256
Dynamics hidden activation	Swish
Dynamics output activation	Linear
Policy GRU hidden units	128
Policy GRU layers	1
Policy history projection	[16]
Policy projection activation	tanh
Actor hidden layers	[256, 256]
Actor hidden activation	ReLU
Actor output	Gaussian (μ, $\log \sigma$), tanh-squashed
Critic GRU hidden units	128
Critic GRU layers	1
Critic history projection	[16]
Critic projection activation	tanh
Critic hidden layers	[256, 256]
Critic hidden activation	ReLU
Critic output	Linear scalar Q

Following the original MAPLE implementation, we use a rollout horizon of $H = 10$, an uncertainty-penalty coefficient of $\lambda_{\text{unc}} = 0.25$, and train the policy for 1000 iterations. MAPLE additionally trains 20 dynamics models and selects 14 models for policy learning. In PS-DAP, we instead train an ensemble of 10 dynamics models and select $k = 6$ models using the proposed predictive-signature diversity criterion. The loss-based filtering tolerance is set to $\delta = 0.001$. This smaller ensemble size reduces the additional computational overhead introduced by diversity-based selection while maintaining sufficient model variability for robust rollout generation.

For predictive-diversity estimation, we use an anchor batch of $B = 100$ offline samples and a multi-step prediction horizon of $K = 10$. The reward component in the predictive signature is weighted by $\beta = 0.7$. Temporal weights follow an exponentially decaying schedule, $w_t \propto 0.9^t$, and are normalized to sum to one. Model reselection is performed every $T_{\text{sel}} = 20$ policy-training epochs.

During refinement, we use a diversity margin of $\epsilon = 2$ and set the diversity weight to $\lambda = 0.01$, increasing it to $\lambda = 0.1$ from epoch 40 onward. We adopt this periodic refinement schedule to balance continued improvement of the selected dynamics models with stability of the synthetic data used for policy training.

The reselection interval is particularly important in our setting, where synthetic transitions constitute approximately 95% of the policy-training data due to the real-data ratio of 0.05. As a result, frequent changes in the subset of dynamics models used for rollout generation can induce substantial changes

in the synthetic data distribution and affect downstream policy learning. By periodically reselecting the model subset rather than changing it at every policy-training epoch, PS-DAP limits abrupt shifts in the synthetic data used for policy optimization while still allowing the rollout ensemble to adapt throughout training.

B MoDAP Reproduction

To provide an additional comparison with MoDAP, we reproduced the method using the publicly available implementation and the experimental details reported in the original work. However, we were unable to recover the performance reported in the paper. Several details required for exact reproduction, including some hyperparameter values and the precise training schedule, are not fully specified. We also sought clarification from the original authors but were unable to obtain additional information prior to submission. Consequently, these settings were reconstructed from the available implementation and tuned within our computational budget. The main text reports the comparison

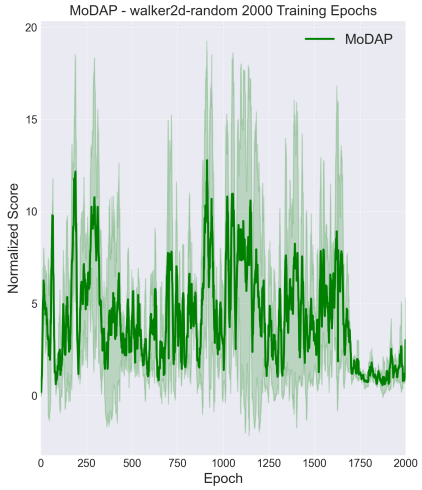


Figure 4: Reproduced MoDAP performance on `walker2d-random` over 2000 training epochs.

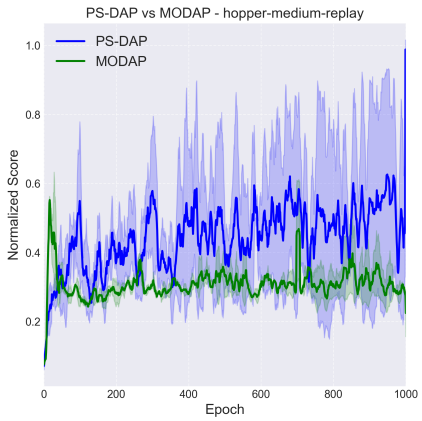


Figure 5: PS-DAP vs. reproduced MoDAP on `hopper-medium-replay`.

between PS-DAP and our reproduced MoDAP implementation on `walker2d-random` under a common 1000-epoch training budget (Fig. 2). To assess whether the observed MoDAP performance was primarily due to insufficient training duration, we extended the same reproduced configuration to 2000 epochs. As shown in Fig. 4, extending the training does not eliminate the observed behavior: substantial fluctuations persist throughout training, followed by a decline in performance at later

epochs. Thus, under our reproduced configuration, increasing the training duration from 1000 to 2000 epochs is insufficient to recover the performance reported in the original MoDAP work.

We additionally compare PS-DAP with the reproduced MoDAP implementation on `hopper-medium-replay` (Fig. 5). The reproduced MoDAP policy exhibits a pronounced initial overshoot, after which its performance drops and oscillates within a substantially lower range, approximately between normalized scores of 0.2 and 0.5. In contrast, PS-DAP maintains higher performance over most of training and finishes with a normalized score of 0.98, compared with 0.22 for the reproduced MoDAP implementation. This additional comparison indicates that the degraded performance observed in our MoDAP reproduction is not limited to the `walker2d-random` setting.

We therefore present these results as reproduction attempts rather than direct replications of the original MoDAP results. The comparisons with PS-DAP should accordingly be interpreted as comparisons against our reproduced MoDAP configuration under the available experimental and computational constraints, rather than as evidence that the originally reported MoDAP performance cannot be attained under the authors' exact experimental setup.

C Algorithmic Details

This section provides the detailed procedures for loss-based filtering, predictive diversity-based subset selection (DivSel), and policy training used in PS-DAP.

C.1 Loss-Based Filtering

Algorithm 1 Loss-Based Filtering

Require: Models $\{P_{\phi_m}\}_{m=1}^{n}$, validation losses $\{L_{\mathrm{MSE}}(\phi_m)\}_{m=1}^{n}$, tolerance δ, subset size k

1: $M_{\mathrm{sorted}} \leftarrow \textsc{Sort}(\{P_{\phi_m}\}_{m=1}^{n}$ by $L_{\mathrm{MSE}}(\phi_m))$
2: $\mathcal{M}_\eta \leftarrow \emptyset$
3: **for** $i = 1$ **to** n **do**
4: **if** $i \leq k$
5: add $M_{\mathrm{sorted}}[i]$ to $\mathcal{M}_\eta$
6: **else if** $L_{\mathrm{MSE}}(M_{\mathrm{sorted}}[i]) - L_{\mathrm{MSE}}(M_{\mathrm{sorted}}[k]) \leq \delta$
7: add $M_{\mathrm{sorted}}[i]$ to $\mathcal{M}_\eta$
8: **else**
9: **break**
10: **end if**
11: **end for**
12: **return** $\mathcal{M}_\eta$

Algorithm 1 describes the loss-based filtering stage used before diversity-aware model selection. Lines 1–2 sort the learned dynamics models according to validation loss and initialize the candidate set $\mathcal{M}_\eta$. Lines 3–10 then construct this set. Lines 4–5 guarantee that the top-k lowest-loss models are always retained, ensuring that at least k sufficiently accurate dynamics models remain available for subsequent selection. Lines 6–7 additionally include models whose validation loss differs from that of the k-th model by at most a tolerance threshold δ. This allows the candidate set to preserve multiple plausible dynamics hypotheses with comparable predictive quality rather than restricting it to the strict top-k. The filtering process terminates once the validation loss exceeds the admissible threshold. Finally, Line 12 returns the filtered candidate set $\mathcal{M}_\eta$, which is subsequently used for diversity-aware subset selection.

C.2 Diversity-Based Subset Selection (DivSel)

Algorithm 2 describes the predictive diversity-based subset selection procedure. Lines 1–2 sample B anchor states from $\mathcal{D}_{\mathrm{off}}$ and shared action sequences of length K. Using the same anchor states and action sequences for all candidate models ensures that differences in their predictions reflect differences in the learned dynamics rather than differences in the inputs. Lines 3–11 generate a K-step

Algorithm 2 DivSel – Diverse Selection

Require: Candidate set $\mathcal{M}_\eta$, subset size k,
dataset D_{off}
anchor size B, rollout length K

1: Sample $\{s_b\}_{b=1}^{B} \sim D_{\text{off}}$
2: Sample shared actions $\{a_b^{(0)}, \ldots, a_b^{(K-1)}\}_{b=1}^{B}$
3: **for** each model $P_{\phi_m} \in \mathcal{M}_\eta$
4: **for** $b = 1$ to B
5: Initialize $\hat{s}_{m,b}^{(0)} \leftarrow s_b$
6: **for** $t = 0$ to $K-1$
7: $(\hat{s}_{m,b}^{(t+1)}, \hat{r}_{m,b}^{(t)}) \leftarrow \mu_{\phi_m}(\hat{s}_{m,b}^{(t)}, a_b^{(t)})$
8: **end for**
9: Construct $g_{m,b}$
10: **end for**
11: **end for**
12: **for** each pair $P_{\phi_i}, P_{\phi_\ell} \in \mathcal{M}_\eta$
13: $d(i,\ell) \leftarrow \frac{1}{B}\sum_{b=1}^{B} \|g_{i,b} - g_{\ell,b}\|_2$
14: **end for**
15: $\mathcal{M}_k \leftarrow \arg\max_{|\mathcal{I}|=k} \min_{i\neq\ell\in\mathcal{I}} d(i,\ell)$
16: **return** $\mathcal{M}_k$

prediction sequence for each model $P_{\phi_m} \in \mathcal{M}_\eta$ and each anchor, and construct the corresponding predictive signature $g_{m,b}$. Lines 12–14 compute the pairwise predictive distance $d(i,\ell)$ between candidate models by averaging the distance between their predictive signatures across all anchors. Finally, Line 15 selects the subset $\mathcal{M}_k$ of size k that maximizes the minimum pairwise predictive distance, encouraging the selected models to remain well separated in predictive space. Line 16 returns the resulting subset $\mathcal{M}_k$.

C.3 Policy Training

Algorithm 3 Policy Training

Require: Policy θ, GRU ψ, dataset D_{off},
candidate set $\mathcal{M}_\eta$,
k, T_{sel}, K rollout length,
B anchor size, horizon H,
iterations N, diversity weight λ

1: $D_{P_\phi} \leftarrow \emptyset$
2: **for** $j = 0$ to $N-1$
3: **if** $j \bmod T_{\text{sel}} = 0$
4: $\mathcal{M}_k \leftarrow \textsc{DivSel}(\mathcal{M}_\eta, k, D_{\text{off}}, B, K)$
5: **end if**
6: Sample $s_0 \sim D_{\text{off}}$, initialize z_0
7: Sample $P_m \sim \text{Uniform}(\mathcal{M}_k)$
8: **for** $t = 0$ to $H-1$
9: $a_t \sim \pi_\theta(\cdot \mid s_t, z_t)$
10: $s_{t+1} \sim P_m(\cdot \mid s_t, a_t)$
11: $r_t \leftarrow \hat{r}_t$
12: $z_{t+1} \leftarrow \text{GRU}_\psi(s_{t+1}, a_t, z_t)$
13: $(s_t, a_t, r_t, s_{t+1}, z_t, z_{t+1}) \cup D_{P_\phi}$
14: **end for**
15: Update π_θ, GRU_ψ by SAC with $D_{\text{off}} \cup D_{P_\phi}$
16: Update $\mathcal{M}_k$ by minimizing $L = L_{\text{MLE}} + \lambda L_{\text{div}}$
17: **end for**

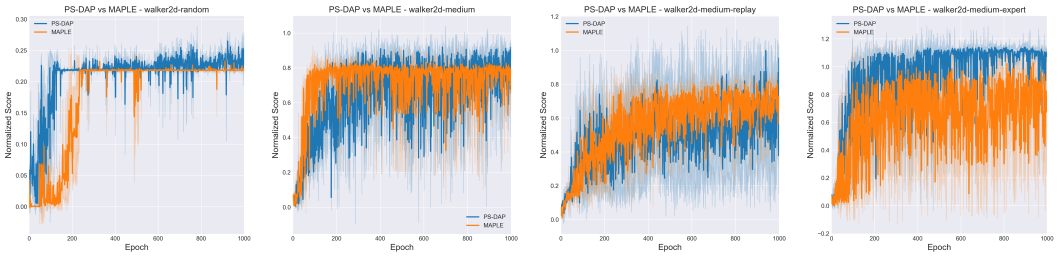
Algorithm 3 describes the PS-DAP policy-training procedure. Line 1 initializes the synthetic transition dataset D_{P_ϕ}. Lines 2–5 periodically recompute the selected subset $\mathcal{M}_k$ using DivSel every T_{sel} iterations, while retaining the current subset between selection updates.

Lines 6–7 initialize each synthetic rollout by sampling a starting state from D_{off}, initializing the recurrent latent state z_0, and sampling a dynamics model uniformly from $\mathcal{M}_k$. Lines 8–14 generate a rollout of horizon H: the policy samples an action conditioned on the current state and latent representation, the selected dynamics model generates the next transition, and the GRU updates the latent state. The generated transitions are accumulated in D_{P_ϕ}.

Finally, Line 15 updates the policy, critic, and recurrent encoder using SAC on a mixture of offline and synthetic transitions. Line 16 updates the currently selected dynamics models using the joint maximum-likelihood and diversity objective, preserving predictive accuracy while encouraging predictive separation among the models used for rollout generation.

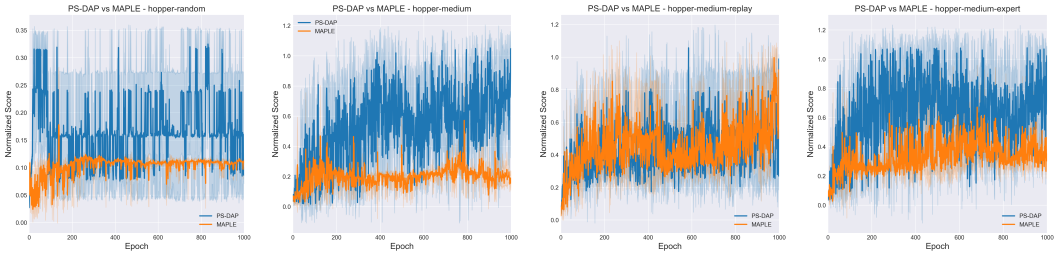
D Additional Results

D.1 Learning Curves Across All Domains



(a) `random` (b) `medium` (c) `medium-replay` (d) `medium-expert`

Figure 6: Learning curves on the Walker2d D4RL datasets. PS-DAP is compared with MAPLE across the `random`, `medium`, `medium-replay`, and `medium-expert` datasets.



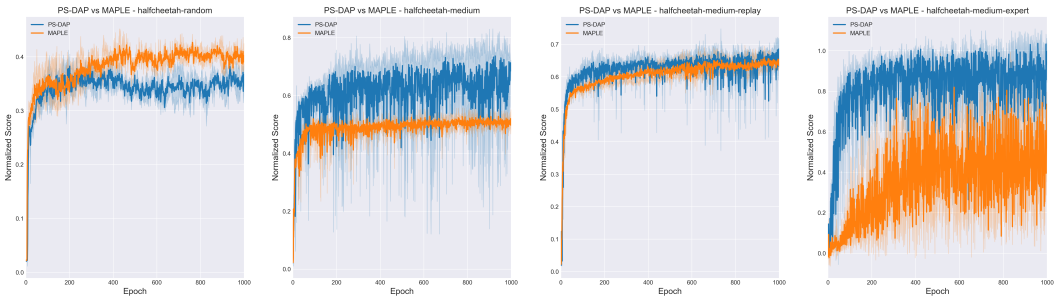
(a) `random` (b) `medium` (c) `medium-replay` (d) `medium-expert`

Figure 7: Learning curves on the Hopper D4RL datasets. PS-DAP is compared with MAPLE across the `random`, `medium`, `medium-replay`, and `medium-expert` datasets.

To further analyze training dynamics, we examine the learning curves across all dataset settings.

Walker2d. Figure 6 presents the learning curves across the Walker2d datasets. On `walker2d-random` (Figure 6a), PS-DAP improves substantially faster during the early stages of training, reaching a normalized score of approximately 0.22 after about 100 epochs. In contrast, MAPLE remains at considerably lower performance during this period and reaches a similar level only after roughly 220 epochs. After its initial rapid improvement, PS-DAP maintains high performance while continuing to improve gradually throughout training.

On `walker2d-medium` (Figure 6b), MAPLE appears to converge faster during the early stages of training, while PS-DAP improves more gradually and eventually reaches higher performance. Both methods exhibit considerable variability throughout training.



(a) `random` (b) `medium` (c) `medium-replay` (d) `medium-expert`

Figure 8: Learning curves on the HalfCheetah D4RL datasets. PS-DAP is compared with MAPLE across the `random`, `medium`, `medium-replay`, and `medium-expert` datasets.

On `walker2d-medium-replay` (Figure 6c), both methods improve rapidly during the early stages of training, after which MAPLE generally maintains higher performance throughout most of training. PS-DAP continues to improve gradually but exhibits greater variability, with occasional peaks reaching performance levels comparable to MAPLE.

On `walker2d-medium-expert` (Figure 6d), PS-DAP maintains higher performance than MAPLE throughout most of training. Both methods exhibit similar overall learning behavior and substantial variability, but MAPLE experiences more frequent and substantially deeper performance drops, whereas PS-DAP remains at consistently higher performance levels.

Hopper. Figure 7 presents the learning curves across the four Hopper dataset settings. On `hopper-random` (Figure 7a), PS-DAP exhibits substantially larger fluctuations than MAPLE. While MAPLE stabilizes around a relatively low normalized return, PS-DAP repeatedly reaches considerably higher scores but does not maintain them consistently throughout training. This behavior is consistent with the policy being exposed to a broader and dynamically changing synthetic training distribution, which may enable the discovery of higher-return behaviors while also increasing optimization variability. Additional stabilization mechanisms, such as more conservative hyperparameter tuning, smoother updates of the selected model subset, or stronger regularization of the synthetic-data distribution, may help preserve these performance gains while reducing the observed fluctuations.

On `hopper-medium` (Figure 7b), PS-DAP consistently outperforms MAPLE throughout training and converges to a substantially higher normalized score.

On `hopper-medium-replay` (Figure 7c), the two methods exhibit substantially more similar learning behavior, with their curves remaining relatively close over much of training. Nevertheless, PS-DAP ultimately reaches a higher final normalized score. Thus, while the benefit of predictive diversity is less pronounced in this dataset than in the `medium` and `medium-expert` settings, the final performance still indicates an advantage for the proposed method.

A similar pattern is observed on `hopper-medium-expert` (Figure 7d), where PS-DAP remains above MAPLE throughout the learning process and reaches a higher final performance. These results indicate that, in these settings, incorporating predictive diversity into both dynamics-model training and model selection provides a persistent advantage during policy optimization rather than only improving the final stages of training.

HalfCheetah. Figure 8 presents the learning curves across the four HalfCheetah dataset settings. On `halfcheetah-random` (Figure 8a), PS-DAP improves rapidly during the early stages of training, but remains below MAPLE over the later epochs, while MAPLE continues to improve and ultimately achieves the higher normalized return.

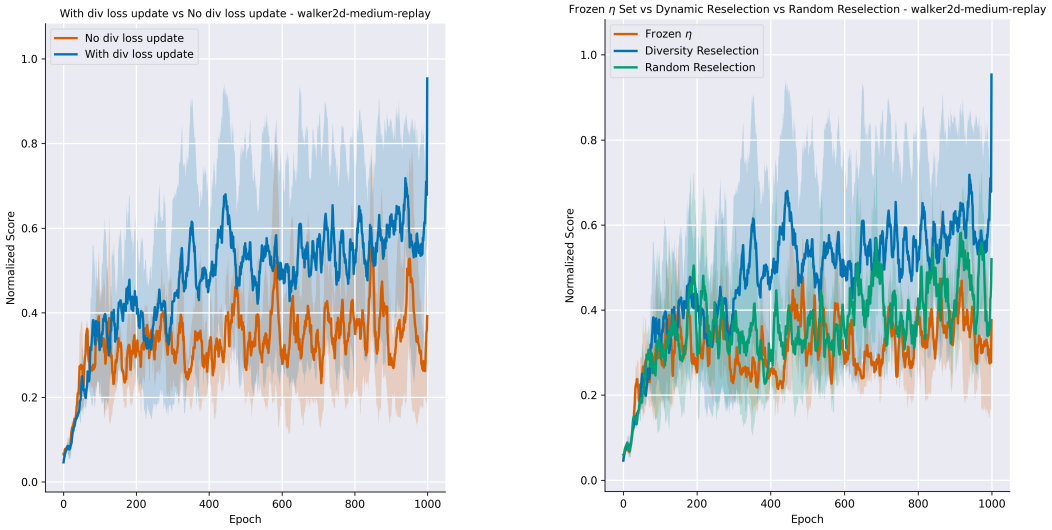
On `halfcheetah-medium` (Figure 8b), PS-DAP reaches a higher-performance region early in training and maintains a clear advantage over MAPLE throughout the optimization process. Although PS-DAP exhibits greater short-term variability, the separation between the two methods remains persistent and the proposed method converges to a substantially higher normalized return.

On `halfcheetah-medium-replay` (Figure 8c), the two methods exhibit very similar learning dynamics. Both improve rapidly during the initial training phase and subsequently evolve along comparable trajectories, with only minor differences in performance. This indicates that the contribution of the diversity-based components is less pronounced in this setting.

The largest separation is observed on `halfcheetah-medium-expert` (Figure 8d). PS-DAP rapidly reaches a high-return regime and remains substantially above MAPLE throughout training. While both methods exhibit noticeable variability, PS-DAP fluctuates around a consistently higher performance level, whereas MAPLE improves more gradually and remains considerably below the proposed method even at later training epochs.

Overall, the learning curves reveal that PS-DAP achieves faster convergence, more stable training, and stronger or comparable final performance across many of the evaluated settings. These results highlight the advantage of diversity-aware model selection in improving both training dynamics and robustness, particularly in challenging offline reinforcement learning settings.

D.2 Ablation Studies



(a) Effect of the diversity loss

(b) Effect of the model selection strategy

Figure 9: Ablation studies on `walker2d-medium-replay`, evaluating the effect of the diversity loss and the model selection strategy.

Figure 9 presents ablation results on `walker2d-medium-replay` for the diversity objective and the model-selection strategy. From Figure 9a, we observe that removing the diversity loss leads to a qualitatively similar learning trend, indicating that the method can still benefit from naturally emerging differences between ensemble members. However, the version trained with the diversity loss achieves a substantially larger performance gain. From this, we infer that the improvement does not arise solely from selecting models according to their diversity at policy-training time. Rather, explicitly optimizing the dynamics models with a diversity-promoting objective during model training also contributes meaningfully to the final performance. This suggests that the two components play complementary roles: diversity-aware training increases the pool of distinct predictive behaviors within the ensemble, while diversity-based selection identifies the most useful subset of models for generating synthetic transitions.

Figure 9b presents the ablation results for the model-selection strategy on `walker2d-medium-replay`. We compare three variants: (i) using all η models that satisfy the prediction-loss criterion, (ii) randomly selecting k models from this set, and (iii) our proposed strategy, which selects k models from the η admissible models according to their predictive diversity. Despite using the largest number of dynamics models for synthetic data generation, the variant that retains all η models achieves the lowest performance among the

three. This result indicates that increasing the number of models used for policy training is not necessarily beneficial. Although the prediction-loss criterion ensures that the retained models remain sufficiently consistent with the offline data, it does not ensure that their predictive behaviors are complementary. Consequently, the η-model set may contain multiple models producing highly similar transitions, leading to substantial redundancy in the synthetic training distribution. In this case, increasing the ensemble size primarily increases the amount of repeated or closely related model-generated experience rather than expanding the effective coverage available to the policy.

Interestingly, randomly selecting only k models from the same admissible set already improves over using all η models. This suggests that restricting the number of models can itself reduce redundancy and prevent highly represented predictive modes from dominating synthetic data generation. However, random selection provides no mechanism for ensuring that the retained models span distinct predictive behaviors. The proposed diversity-based selection achieves the strongest performance by explicitly choosing a subset whose predictions are complementary while restricting the candidate set to models that satisfy the prediction-loss threshold. We therefore infer that the benefit of the proposed method arises not from maximizing the number of models used for policy training, but from selecting a compact set that balances model fidelity with predictive diversity. Together with the diversity-loss ablation, these results suggest that both encouraging diversity during dynamics-model training and exploiting that diversity during model selection contribute to the final performance. Figure 10 presents the effect of the synthetic rollout horizon on `walker2d-random`. Overall,

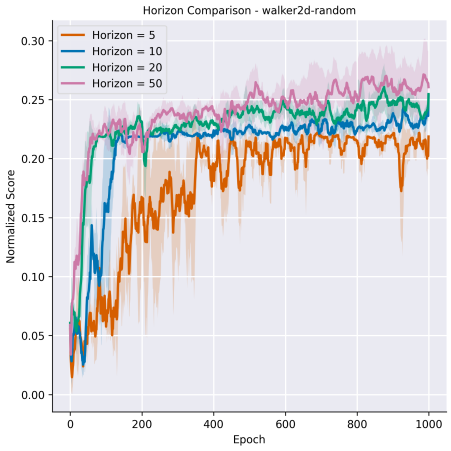


Figure 10: Effect of the rollout horizon on `walker2d-random`.

performance improves as the rollout horizon is increased. Short rollouts with $H = 5$ result in slower learning and the lowest final normalized return, whereas increasing the horizon to $H = 10$ and $H = 20$ leads to faster improvement and higher performance. The longest rollout, $H = 50$, achieves the highest overall return, although the improvement over $H = 20$ is more moderate, suggesting diminishing gains as the horizon becomes sufficiently long.

Longer rollouts allow the selected dynamics models to generate more temporally extended synthetic trajectories, thereby increasing the temporal extent of the synthetic data used for policy optimization. This can be particularly useful in the `random` setting, where high-quality behavioral trajectories may be sparsely represented in the offline dataset. At the same time, longer rollouts increase the potential for compounding model error, since later transitions are conditioned on previously predicted states. The improvement observed up to $H = 50$ suggests that, in this setting, the additional synthetic experience provided by longer rollouts remains beneficial despite the increased exposure to accumulated prediction error.

This improvement, however, comes at an increased computational cost. Extending the rollout horizon increases the number of model evaluations and generated transitions, resulting in longer training times and higher memory consumption. Therefore, the rollout horizon introduces a trade-off between policy performance and computational efficiency. In particular, the relatively smaller gain from $H = 20$ to $H = 50$ suggests that substantially longer horizons may not always justify their additional computational and memory requirements.

E COMPUTATIONAL RESOURCES

Experiments were conducted on both a high-performance computing cluster and a local workstation. On the cluster, we used nodes equipped with $2\times$ Intel Xeon 6710E CPUs (128 cores total), $4\times$ NVIDIA L40S GPUs (48GB memory each), and approximately 1TB RAM, running CUDA 12.4. On the local workstation, we used an Intel i7-12700 CPU and an NVIDIA RTX 3070 Ti GPU (8GB), with CUDA 13.0. Each training run used a single GPU, with one process per GPU. Experiments were executed in parallel across multiple GPUs when available.

Policies were trained for 1000 epochs, and results are averaged over 3 random seeds. The typical runtime per training run was approximately 7-12 hours per seed, with peak GPU memory usage remaining below approximately 1-2GB per run.