POLICY PLASTICITY MATTERS IN OFFLINE-TO-ONLINE REINFORCEMENT LEARNING: REFITTING OFFLINE POLICIES FOR ONLINE ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

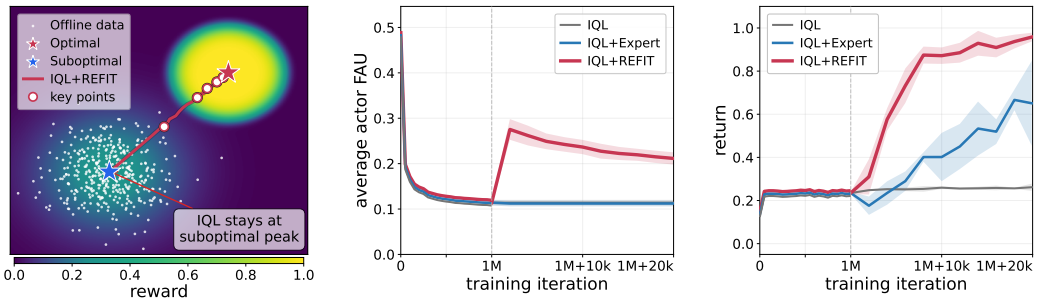
Offline-to-Online Reinforcement Learning (O2O RL) has emerged as a practical paradigm that pre-trains the policy using static offline datasets and subsequently adapts the policy through online interactions. Existing O2O methods primarily address the transition through value calibration, while generally treating the offline-trained policy as a given initialization. We instead study O2O adaptation from the perspective of network plasticity, asking whether the offline-trained policy remains sufficiently adaptable for online learning. Controlled experiments show that prolonged optimization on static offline data progressively reduces network plasticity even after offline performance has largely saturated, and that lower plasticity is associated with weaker subsequent online improvement. Motivated by these observations, we propose REstoring plasticity via Fresh Initialization and policy Transfer (REFIT), a lightweight model-level method for the O2O transition. Before online fine-tuning, REFIT distills the offline policy into a freshly initialized student while temporarily freezing a random subset of student units, transferring the learned offline behavior to a more plastic policy initialization. Extensive experiments on D4RL and OGBench demonstrate that REFIT consistently achieves higher aggregate performance than existing O2O plug-in methods across both Cal-QL and IQL backbones, while plasticity diagnostics and ablations provide further evidence of restored network plasticity.

1 INTRODUCTION

Deep Reinforcement Learning (DRL) has achieved strong performance across diverse real-world applications, such as robot control (Qing et al., 2024; 2025) and power grid control (Xu et al., 2024; Chen et al., 2025). To learn such effective policies, DRL typically relies on either collecting experience through online interaction or leveraging previously collected datasets. However, online interaction can be costly and unsafe (Lillicrap et al., 2015; Haarnoja et al., 2018), while learning solely from fixed datasets is limited by insufficient data distribution coverage and the challenge of out-of-distribution (OOD) actions (Fujimoto et al., 2019; Kostrikov et al., 2021; Kumar et al., 2020b). Offline-to-Online (O2O) RL bridges these two paradigms by pre-training on offline data and subsequently improving the policy through online interaction, thereby reducing the reliance on costly exploration and enabling adaptation beyond the coverage of the offline data (Zhou et al., 2025b; Nakamoto et al., 2023; Zhang et al., 2023).

Despite these promising results, O2O RL often suffers from limited asymptotic performance during online fine-tuning (Zhang et al., 2024; Luo et al., 2023). To tackle this issue, existing methods generally take the offline-trained policy as a given initialization and focus on recalibrating value estimates under the offline-to-online shift, alleviating the excessive conservatism inherited from offline training (Nakamoto et al., 2023; Luo et al., 2024; Shin et al., 2025). These methods implicitly assume that the offline-trained policy remains sufficiently plastic to benefit from subsequent online updates. However, we argue that the network plasticity of the offline-trained policy can also be a critical bottleneck to subsequent online improvement. Specifically, the loss of plasticity during offline training can cause the network to become rigid, severely hindering its ability to move beyond

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.



(a) Environment setup (b) Actor FAU (c) Learning curve

Figure 1: Network plasticity and online adaptation in a controlled O2O setting. The dashed line marks the transition from offline pre-training to online fine-tuning. (a) The bimodal reward landscape, offline data distribution, and policy trajectories. (b) Actor FAU throughout offline training and subsequent online fine-tuning. (c) Learning curve throughout training.

the suboptimal behavior learned offline and adapt toward new behavior patterns that yield higher returns. To exemplify the above issue, we design a controlled 2D continuous bandit experiment and track the Fraction of Active Units (FAU) as a proxy for network plasticity. As shown in Figure 1a, the offline data cover only the suboptimal peak. During online fine-tuning, IQL remains near the suboptimal peak, while even with the guidance of expert data, the IQL+Expert variant can only improve slowly. Their low actor FAU (Figure 1b) suggests that insufficient network plasticity, rather than data coverage alone, can hinder the offline-trained policy from adapting toward better behaviors.

Motivated by these observations, we propose **REstoring plasticity via Fresh Initialization and policy Transfer (REFIT)** for O2O RL. Unlike prior methods that focus on value correction (Nakamoto et al., 2023; Luo et al., 2024; Shin et al., 2025) to improve online learning efficiency, REFIT restores the plasticity of the offline-trained policy before online fine-tuning, allowing it to better adapt to the online distribution while preserving the offline prior. Specifically, REFIT transfers the offline behavioral prior by distilling the teacher policy into a freshly initialized student network. Crucially, by temporarily freezing a random subset of units in the student network during this process, REFIT preserves a subset of unoptimized parameters that preserve high plasticity, facilitating rapid adaptation during the subsequent online phase. As shown in Figure 1b and Figure 1c, even when the offline policy converges to a highly suboptimal solution, REFIT uses distillation to retain the offline prior while restoring network plasticity, moving beyond its offline behavior and continuing to improve during online fine-tuning.

Our contributions are summarized as follows:

- We identify an overlooked issue in O2O RL: Offline optimization can reduce the plasticity of the policy network, hindering its ability to continue adapting during online fine-tuning.
- We propose REFIT, a simple plug-in method that distills the offline policy into a freshly initialized student policy while temporarily freezing a random subset of student units. This procedure transfers the learned offline behavioral prior to a more plastic network, enabling continued adaptation during online fine-tuning.
- We demonstrate the effectiveness of REFIT through extensive experiments on D4RL and OGBench across two O2O RL backbones, showing that it consistently outperforms existing plug-in O2O RL methods and achieves stronger performance across various settings.

2 Related Works

2.1 Offline-to-Online Reinforcement Learning

O2O RL first pre-trains a policy on a static dataset, and then fine-tunes it through online environment interactions (Nair et al., 2020; Nakamoto et al., 2023; Fujimoto et al., 2019; Fujimoto & Gu, 2021). It combines the strong initialization provided by offline RL with the continual improvement

enabled by online interactions. Early O2O methods directly extend conservative offline RL algorithms to the online phase, including Implicit Q-Learning (Kostrikov et al., 2021) and Conservative Q-Learning (Kumar et al., 2020b). More recent methods have dived deep into the inner mechanisms of O2O RL such as recalibrating the value function to mitigate value underestimation (Nakamoto et al., 2023) and reducing distribution mismatch via a warm-up phase (Zhou et al., 2025b). However, existing O2O methods primarily focus on these algorithmic challenges, while offline-induced plasticity degradation has received comparatively limited attention. In contrast, our method explicitly restores network plasticity during O2O adaptation, leading to substantially improved performance.

2.2 Plasticity in Reinforcement Learning

Plasticity refers to the ability of a neural network to continually adapt to new data distributions (Klein et al., 2024; Dohare et al., 2021). Recent studies have shown that deep RL agents progressively suffer from degrading plasticity, limiting their ability to adapt to new distributions despite having sufficient model capacity (Abbas et al., 2023; Lyle et al., 2022; Wang et al., 2025). Existing work attributes this phenomenon to the non-stationarity of RL training and measures it through symptoms such as dormant neurons, representation collapse, and abnormal parameter or gradient dynamics (Sokar et al., 2023; Klein et al., 2024; Zhou et al., 2025a). To mitigate plasticity loss, prior methods have explored weight resets (Nikishin et al., 2022), normalization (Lyle et al., 2023) and regularization techniques (Kumar et al., 2024), as well as architectural designs that explicitly inject plasticity into the critic during training (Nikishin et al., 2023). Meanwhile, recent work (Kong et al., 2024) has realized that primacy bias can also be a key factor limiting subsequent O2O adaptation. However, direct assessment of policy plasticity during the O2O pipeline and corresponding solutions specifically targeting its restoration remain underexplored. To fill this gap, we first systematically characterize offline-induced plasticity loss and propose REFIT to restore network plasticity at the O2O transition by distilling the offline teacher policy to an initialized and partially frozen online student policy.

3 Preliminaries

Offline-to-Online Reinforcement Learning Traditional RL can be viewed as a Markov Decision Process (MDP) (Puterman, 1990) paradigm, defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, d_0, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s' \mid s, a)$ represents the transition dynamics, $r(s, a)$ is the reward function, $d_0(s)$ is the initial state distribution, and $\gamma \in [0, 1)$ denotes the discount factor. The objective of RL is to learn a policy $\pi(a \mid s)$ that maximizes the expected discounted return, $\pi^* = \arg\max_\pi \mathbb{E}_{s_0 \sim d_0,\, a_t \sim \pi(\cdot|s_t),\, s_{t+1} \sim P(\cdot|s_t, a_t)} [\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$. The policy is evaluated by the state-value function $V^\pi(s) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s]$ and the action-value function $Q^\pi(s, a) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a]$. As for O2O RL, the agent is given a dataset $\mathcal{D}_{\text{off}} = \{(s_t, a_t, r_t, s_{t+1})\}_{t=1}^{N}$ collected by a behavior policy π_β. The agent leverages $\mathcal{D}_{\text{off}}$ to pre-train the value functions Q_{offline}, V_{offline} and policy π_{offline}, which then serve as the starting point for online interaction and optimization.

Model Plasticity Network plasticity (Lyle et al., 2023; Dohare et al., 2021; Nikishin et al., 2022) refers to the capacity of a neural network to continually adapt to new data distributions, which can be quantitatively monitored through activation patterns, parameter scales, and gradient behaviors. In this work, we primarily focus on the *Fraction of Active Units (FAU)* (Ma et al., 2023) as a key metric for diagnosing network plasticity. Additional plasticity metrics are provided in Appendix F.1. Specifically, for a hidden layer l with width H_l, we define the layer-wise FAU as well as the network FAU under a state distribution $\mathcal{B}$ as

$$\Phi_l = \mathbb{E}_{s \sim \mathcal{B}} \left[\frac{1}{H_l} \sum_{n=1}^{H_l} \mathbb{I}\Big(h_n^{(l)}(s) > 0\Big) \right], \qquad \Phi_{\text{network}} = \frac{1}{L} \sum_{l=1}^{L} \Phi_l, \tag{1}$$

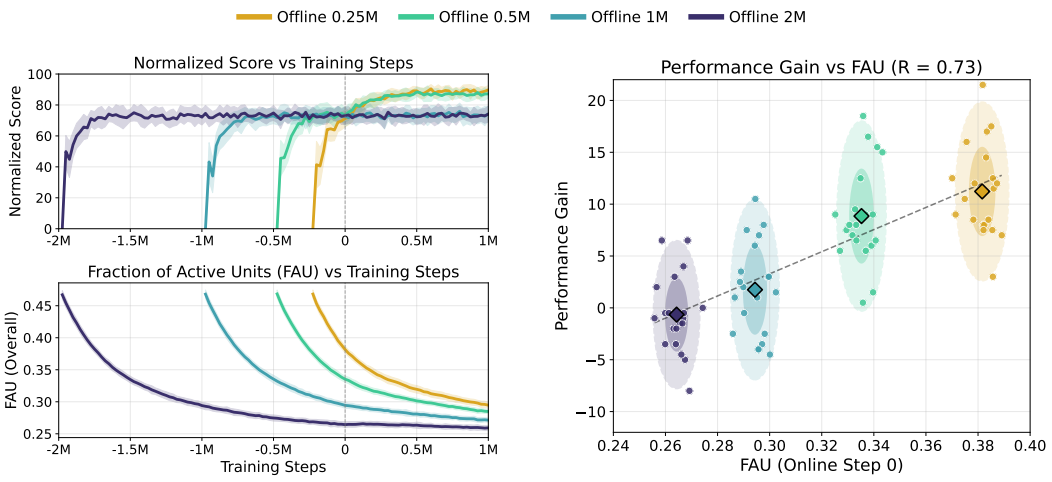
where $h_n^{(l)}(s)$ denotes the pre-activation of unit n in layer l, L is the number of hidden layers, and $\mathbb{I}(\cdot)$ is the indicator function. This statistic measures the fraction of hidden units with nonzero post-ReLU activation.

4 Understanding and Mitigating Plasticity Loss in O2O RL

To understand what limits online adaptation in O2O RL, we examine the plasticity of the offline-trained policy. Specifically, we first provide controlled empirical evidence that offline over-optimization progressively reduces network plasticity and suppresses subsequent online adaptation. This observation motivates **Plasticity Restoration** as a key design principle for O2O RL. Following this principle, we propose REstoring plasticity via Fresh Initialization and policy Transfer, termed **REFIT**, a plug-in method orthogonal to existing O2O RL algorithms that transfers the behavioral prior of the offline teacher policy to a freshly initialized online student policy through randomized partial freezing and behavioral distillation.

4.1 Empirical Analysis: Plasticity Loss in O2O RL

To investigate the relationship between offline training, network plasticity, and subsequent online adaptation, we conduct a controlled empirical study based on IQL (Kostrikov et al., 2021) on the D4RL `antmaze-medium-play` environment, as shown in Figure 2. To disentangle the effect of network plasticity from that of offline policy quality at the start of online fine-tuning, we pre-train identical policies with four offline training budgets: 0.25M, 0.5M, 1M, and 2M gradient steps. These checkpoints attain comparable near-saturated offline performance while exhibiting distinct levels of network plasticity, thereby allowing us to study how plasticity affects subsequent online adaptation. During the whole process, we track the normalized score to measure adaptation performance and FAU (Ma et al., 2023) as a proxy for network plasticity. A decline in FAU reflects a reduced fraction of active units, further indicating increased plasticity loss and diminished adaptability. All the above experiments are conducted over 20 random seeds. Given the results shown in Figure 2, we have the following observations.



(a) Performance and FAU of O2O IQL

(b) Correlation between FAU and performance gain

Figure 2: Offline over-optimization significantly degrades model plasticity, and the plasticity metric FAU is shown to be positively correlated with online performance improvements. (a) Overall performance and FAU curves across varying offline pre-training budgets. (b) Correlation between online initial policy FAU and performance improvement.

Observation 1: Offline Over-Optimization Leads to Plasticity Degradation.

Continued optimization on static offline data progressively reduces FAU even after offline performance has largely saturated, indicating a gradual plasticity loss before online fine-tuning.

As shown in Figure 2a, all of the settings achieve comparable offline performance, while FAU consistently decreases with continued offline optimization. This indicates that network plasticity can degrade even after offline performance has largely saturated. The results suggest that over-optimization

on a static offline dataset may progressively constrain the diversity of network activations, reducing the fraction of active units and potentially limiting the capacity of the model to adapt to novel distributions encountered during online fine-tuning.

Observation 2: Higher Plasticity Correlates with Stronger Online Adaptation.

Policies with higher FAU exhibit larger online performance gains, whereas lower-FAU policies adapt more slowly and tend to plateau earlier.

During online fine-tuning, the policy must adapt to online interaction data beyond the fixed offline distribution. As shown in Figure 2a, online improvement steadily diminishes as the offline training budget increases from 0.25M to 2M steps. In particular, the 1M- and 2M-step policies show little further improvement during online fine-tuning, consistent with weaker adaptability at lower plasticity. This relationship is further reflected in Figure 2b, where mean FAU is positively correlated with online performance gain, linking higher retained plasticity to stronger subsequent online adaptation.

Design Implication: Plasticity Restoration for O2O RL These empirical observations suggest that offline optimization can progressively reduce policy plasticity and weaken subsequent online adaptation. This motivates treating plasticity restoration as an explicit design objective at the O2O transition. Guided by this insight, we design REFIT to transfer the behavioral prior from the offline teacher policy to a freshly initialized student policy, restoring the capacity of the agent for subsequent online adaptation without discarding the learned offline behavior.

4.2 REFIT FOR PLASTICITY RESTORATION

Motivated by the above analysis, we introduce **REFIT**, a lightweight model-level method for restoring policy plasticity at the O2O transition. Rather than directly continuing online optimization from the offline-trained policy parameters, REFIT instantiates a freshly initialized student policy and transfers the behavior learned by the offline teacher through policy distillation. During distillation, a random subset of student units is temporarily frozen at their initial values, allowing the student to acquire the offline behavior while retaining part of its freshly initialized structure. Once distillation is complete, the frozen units are released, and the resulting online student policy undergoes online fine-tuning with all parameters trainable.

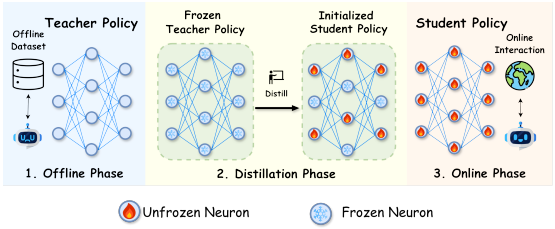


Figure 3: Schematic overview of REFIT.

Randomized Partial Freezing. Before online fine-tuning, REFIT instantiates a fresh online student policy π_{student} from the standard initialization distribution and randomly partitions its parameters θ_{student} into two disjoint subsets:

$$\theta_{\text{student}} = \theta_{\text{train}} \cup \theta_{\text{freeze}}, \quad \theta_{\text{train}} \cap \theta_{\text{freeze}} = \emptyset. \tag{2}$$

Specifically, REFIT randomly selects a fraction ρ of units in each dense layer and freezes their associated parameters at their initial values throughout distillation, forming θ_{freeze}. The remaining parameters θ_{train} are optimized to reproduce the behavior of the offline teacher policy. By leaving part of the freshly initialized student unoptimized during behavioral transfer, partial freezing allows the trainable parameters to acquire the offline behavioral prior through distillation while preserving a freshly initialized parameter subset for subsequent online adaptation.

Plasticity-Preserving Distillation. Given the partition above, REFIT transfers the behavior of the offline teacher policy π_{teacher} to the student using the offline dataset $\mathcal{D}_{\text{off}}$. During distillation, only the trainable subset θ_{train} is optimized, while θ_{freeze} remains fixed at its initial values:

$$\mathcal{L}_{\text{distill}}(\theta_{\text{train}}) = \mathbb{E}_{s\sim\mathcal{D}_{\text{off}}}\left[\left\|\pi_{\theta_{\text{student}}}(s) - \pi_{\text{teacher}}(s)\right\|_2^2\right]. \tag{3}$$

This objective allows the trainable parameters to recover the behavior learned by the offline teacher, while the frozen subset remains unaffected by distillation and retains its fresh initialization. REFIT

is applied only to the policy network; the critic and value networks are directly inherited from the offline phase. After distillation, the freeze mask is removed and all student parameters are set to trainable for subsequent online fine-tuning. In this way, REFIT combines fresh initialization, behavioral distillation, and temporary partial freezing to transfer the offline policy behavior while providing a more plastic initialization for online adaptation. Pseudocode of our algorithm is provided in Appendix A.

5 EXPERIMENTS

Our experiments aim to answer the following questions: whether REFIT (1) improves the online performance of original O2O RL algorithms over other plug-in methods (Section 5.1), (2) restores network plasticity during online fine-tuning (Section 5.2), (3) benefits from each design component (Section 5.3), (4) remains effective without manually tuning the offline training budget (Section 5.4), and (5) introduces only modest computational overhead (Appendix C).

Experiment Setting We conduct experiments on D4RL (Fu et al., 2020) MuJoCo locomotion and AntMaze tasks and three OGBench (Park et al., 2025) manipulation tasks. We select two representative O2O RL algorithms as baselines: IQL (Kostrikov et al., 2021) and Cal-QL (Nakamoto et al., 2023). For each algorithm, we compare against a dedicated O2O plug-in method: OPT (Shin et al., 2025), which adds an online critic pre-training stage to IQL, and PARS (Kim et al., 2025), which regularizes OOD actions for Cal-QL. More experimental details are provided in Appendix B.

5.1 MAIN PERFORMANCE RESULTS

Table 1: Performance of our REFIT and baselines on AntMaze, MuJoCo, and OGBench, averaged over five random seeds. **Bold** and underlined values denote the best and second-best results within each backbone, respectively.

Environment	IQL (Kostrikov et al., 2021)			Cal-QL (Nakamoto et al., 2023)		
	Baseline	OPT	Ours (REFIT)	Baseline	PARS	Ours (REFIT)
antmaze-m-p	76.2 ± 1.5	**88.0** ± 3.0	86.6 ± 4.8	92.2 ± 2.1	92.4 ± 3.0	**95.4** ± 1.7
antmaze-m-d	71.6 ± 2.7	88.8 ± 1.2	**89.2** ± 2.7	93.2 ± 0.8	89.6 ± 2.9	**96.2** ± 0.8
antmaze-l-p	46.0 ± 2.8	61.6 ± 5.1	**63.8** ± 8.3	70.6 ± 7.5	**78.6** ± 3.6	78.2 ± 5.0
antmaze-l-d	50.6 ± 5.7	63.6 ± 3.6	**65.2** ± 4.0	53.6 ± 17.4	79.2 ± 12.4	**80.2** ± 5.0
AntMaze total	244.4	302.0	**304.8**	309.6	339.8	**350.0**
halfcheetah-r	13.8 ± 3.9	33.2 ± 11.8	**46.3** ± 1.4	33.4 ± 3.8	27.0 ± 2.4	**39.2** ± 3.5
hopper-r	8.0 ± 0.3	11.4 ± 0.5	**15.7** ± 4.1	8.3 ± 1.7	15.0 ± 0.8	**31.2** ± 0.9
walker2d-r	7.1 ± 1.0	10.0 ± 1.0	**10.7** ± 1.1	11.5 ± 1.5	11.7 ± 0.4	**13.5** ± 3.8
halfcheetah-m	48.1 ± 0.1	**54.3** ± 2.7	50.0 ± 0.3	**65.3** ± 3.0	64.2 ± 1.1	51.1 ± 0.9
hopper-m	63.9 ± 3.2	**91.6** ± 12.1	79.8 ± 3.9	76.3 ± 6.1	71.7 ± 1.8	**82.2** ± 2.2
walker2d-m	71.9 ± 2.5	78.9 ± 3.0	**87.9** ± 0.6	83.4 ± 2.2	88.8 ± 0.7	**97.3** ± 2.9
halfcheetah-m-r	43.9 ± 0.2	46.9 ± 2.8	**47.1** ± 0.4	**58.9** ± 0.8	54.4 ± 0.3	55.8 ± 0.7
hopper-m-r	77.5 ± 18.1	**104.8** ± 1.4	102.7 ± 0.4	68.9 ± 7.8	85.7 ± 4.6	**87.6** ± 7.5
walker2d-m-r	74.8 ± 2.8	93.6 ± 1.1	**96.9** ± 2.2	86.7 ± 8.7	**101.7** ± 1.3	97.9 ± 5.8
MuJoCo total	409.0	524.7	**537.1**	492.7	520.2	**555.8**
cube-single-play	70.7 ± 9.7	73.0 ± 9.4	**81.1** ± 5.3	15.8 ± 16.2	63.2 ± 1.9	**79.6** ± 2.2
puzzle-3x3-play	25.3 ± 4.6	27.7 ± 6.4	**98.6** ± 1.8	1.4 ± 3.1	**17.0** ± 9.7	9.0 ± 8.3
scene-play	76.7 ± 5.7	77.0 ± 3.0	**96.9** ± 2.4	**11.8** ± 11.6	1.4 ± 1.1	6.0 ± 5.4
OGBench total	172.7	177.7	**276.6**	29.0	81.6	**94.6**
All total	826.1	1004.4 (+21.6%)	**1118.5** (+35.4%)	831.3	941.6 (+13.3%)	**1000.4** (+20.3%)

Following prior work (Shin et al., 2025; Kim et al., 2025), we report performance after 300K online fine-tuning steps. Table 1 shows that REFIT improves the aggregate score over vanilla IQL and Cal-QL by 35.4% and 20.3% respectively, while also outperforming the corresponding plug-in methods. The training curves in Figure 4 further show pronounced gains on OGBench and sparse-reward AntMaze tasks, where the base algorithms and even their corresponding plug-in method often plateau during online fine-tuning. Overall, these results demonstrate that REFIT improves aggregate

O2O performance across both backbones, especially in settings where the base algorithms struggle to improve further during online fine-tuning. Additional comparisons are provided in Appendix D.

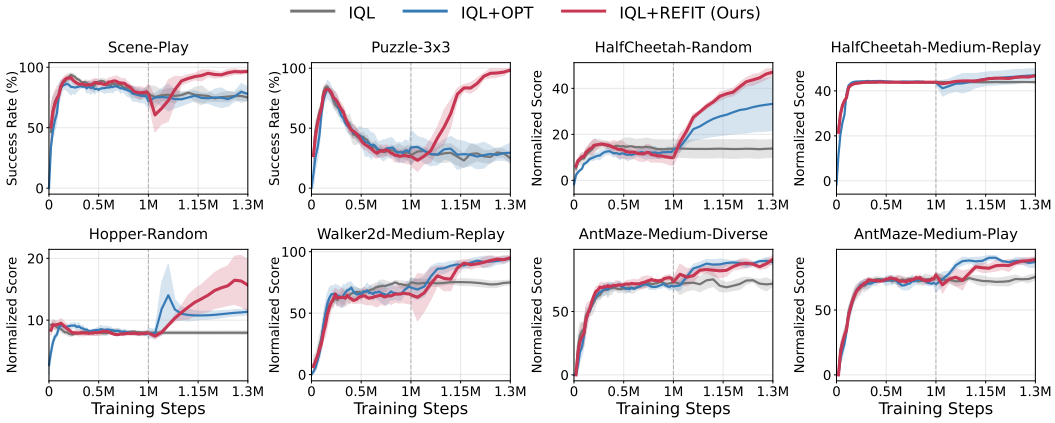


Figure 4: O2O training curves on D4RL and OGBench. The dashed line marks the transition from offline pre-training to online fine-tuning, at which REFIT is applied.

5.2 Plasticity Diagnostics

Figure 5 tracks the FAU throughout offline and online training. The results show that FAU declines during offline pre-training for both methods and remains low under vanilla online fine-tuning. In contrast, REFIT sharply restores FAU at the O2O transition, consistent with its goal of restoring network plasticity for subsequent online adaptation. Meanwhile, Appendix F.2 further shows consistent trends across complementary plasticity metrics, including weight norm and SRank, providing additional support for the plasticity-restoring effect of REFIT.

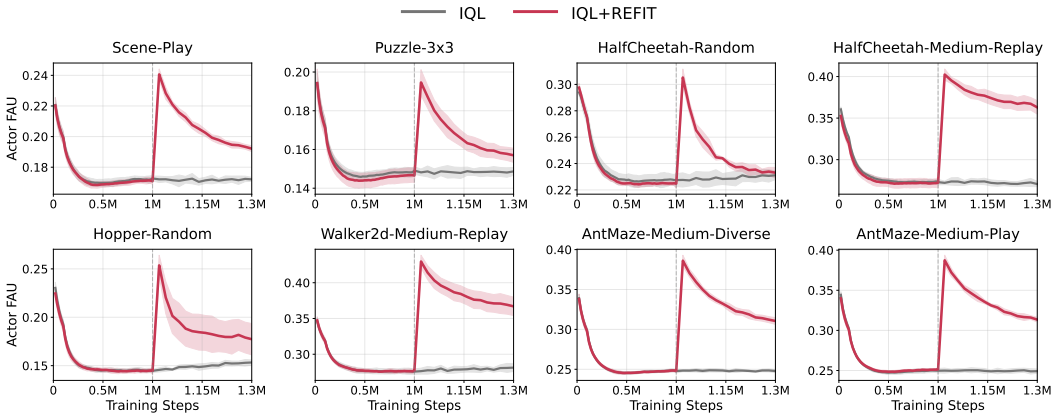


Figure 5: Evolution of actor FAU across the offline-to-online transition.

5.3 Ablation Analysis

Ablation of REFIT Design Components We investigate the contributions of fresh initialization, distillation, and partial freezing. The *Baseline* directly deploys the classic IQL algorithm, while *Reset* reinitializes the policy network before online fine-tuning. *Reset + Distillation* further transfers the behavior learned by the offline teacher policy to the freshly initialized student through distillation, without partial freezing. Finally, *REFIT* combines all three components. As shown in Table 2, none of these isolated variants can match the performance of the complete REFIT configuration, which achieves the highest overall score. These results suggest that fresh initialization, policy distillation, and partial freezing provide complementary benefits for subsequent online adaptation.

Table 2: Ablation of REFIT design components. All results are averaged over 5 random seeds.

Method	hc-m	hc-r	wal-m-r	hop-m-r	ant-m-p	ant-m-d	Score
Baseline	48.1 ± 0.1	13.8 ± 3.9	74.8 ± 2.8	77.5 ± 18.1	76.2 ± 1.5	71.6 ± 2.7	362.0 (-23.3%)
Reset	47.3 ± 0.4	38.6 ± 2.4	84.0 ± 10.6	86.5 ± 29.9	71.2 ± 6.8	74.0 ± 5.5	401.6 (-14.9%)
Reset + Distillation	49.8 ± 0.2	45.7 ± 1.9	89.3 ± 11.2	97.9 ± 6.2	83.5 ± 1.8	79.2 ± 23.6	445.4 (-5.6%)
REFIT	**50.0** ± 0.3	**46.3** ± 1.4	**96.9** ± 2.2	**102.7** ± 0.4	**86.6** ± 4.8	**89.2** ± 2.7	**471.7**

Ablation of Freeze Ratio We vary the freeze ratio over $\rho \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$, where $\rho = 0.0$ corresponds to distillation without freezing and $\rho = 1.0$ freezes the entire student network during distillation. As shown in Table 3, intermediate freeze ratios achieve comparable aggregate performance, whereas both extreme settings perform notably worse. When $\rho = 0$, all student parameters are updated during distillation, which may reduce the fraction of active units and limit the plasticity gains from fresh initialization. Conversely, when $\rho = 1$, the student is entirely frozen and retains only the initialized parameters without inheriting the offline behavioral prior. The robustness of REFIT across the intermediate range suggests that the core mechanism driving performance is the *combination* of partial freezing and distillation itself, rather than any particular freeze ratio: as long as a sufficient subset of parameters is frozen to preserve plasticity while the remaining parameters absorb the teacher distribution, REFIT maintains comparable performance across these intermediate freeze ratios. Given its highest aggregate performance across the evaluated tasks, we adopt $\rho = 0.5$ as the default in the remaining experiments.

Table 3: Performance of REFIT under various freeze ratio settings. The two lowest scores in each row are marked in red.

Env	0.0	0.25	0.5	0.75	1.0
halfcheetah-m	49.8 ± 0.2	50.0 ± 0.2	50.0 ± 0.3	**50.1** ± 0.4	47.3 ± 0.4
halfcheetah-r	45.7 ± 1.9	46.2 ± 1.3	46.3 ± 1.4	**46.5** ± 1.8	38.6 ± 2.4
antmaze-m-p	83.5 ± 1.8	86.0 ± 3.7	**86.6** ± 4.8	85.8 ± 4.1	71.2 ± 6.8
antmaze-m-d	79.2 ± 23.6	86.4 ± 3.0	**89.2** ± 2.7	84.4 ± 1.5	74.0 ± 5.5
Total	258.2	268.6	**272.1**	266.8	231.1

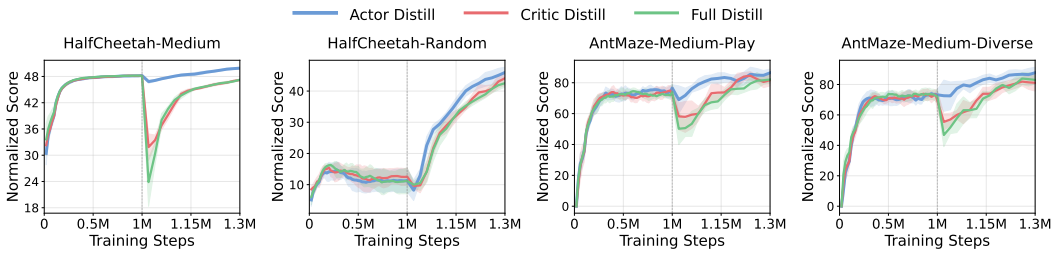


Figure 6: Learning curves of REFIT with different distillation targets.

Ablation of Distillation Target We compare three distillation targets: actor-only, critic-only, and full actor-critic distillation. As shown in Figure 6, variants involving critic distillation exhibit a pronounced performance drop at the start of online fine-tuning, whereas actor-only distillation largely avoids this degradation and maintains more stable adaptation. This suggests that critic distillation introduces additional perturbations to the learned value function, making the O2O transition less stable. Based on this observation, REFIT distills only the actor network while allowing the critic to adapt freely during online fine-tuning.

Ablation of Distillation Budget We vary the policy-distillation budget from 0K, corresponding to no distillation, to 500K updates. As shown in Figure 7, all nonzero distillation budgets consistently outperform the 0K baseline, highlighting the importance of transferring the offline policy before online fine-tuning. Meanwhile, performance remains relatively stable across various settings, indicating limited sensitivity to the exact distillation duration. Together with the freeze ratio ablation, REFIT remains effective across a wide range of distillation configurations without requiring careful hyperparameter tuning. Based on this robustness, we adopt 100K updates as the default setting, which provides strong performance without incurring unnecessary additional computation.

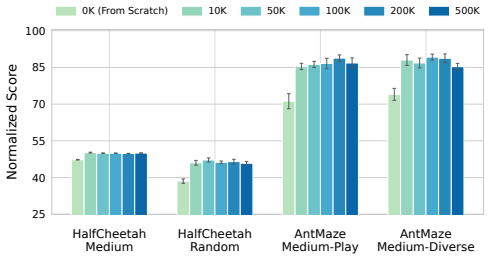
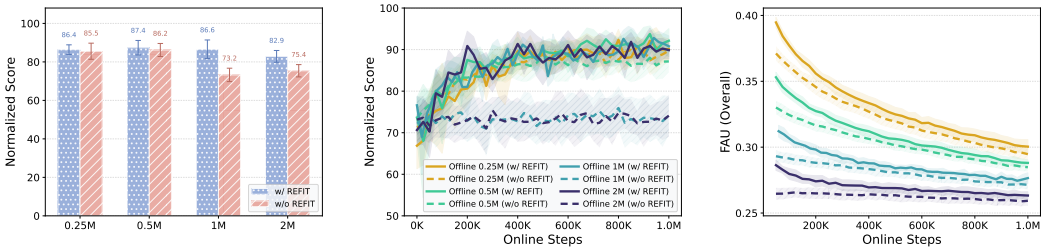


Figure 7: Performance of REFIT under different distillation budget settings.

5.4 Robustness to Offline Training Budget

The motivation study in Section 4.1 shows that prolonged offline optimization can degrade network plasticity and hinder subsequent online adaptation. We therefore evaluate REFIT across offline training budgets ranging from 0.25M to 2M steps under the same setting as in Section 4.1. As shown in Figure 8a and Figure 8b, REFIT consistently outperforms the corresponding baseline across all offline budget settings. Notably, as the offline training budget increases, the baseline exhibits increasingly pronounced plateaus during online fine-tuning, whereas REFIT continues to improve, suggesting that restoring network plasticity enables effective adaptation after prolonged offline optimization. Figure 8c further shows that REFIT maintains higher FAU throughout online fine-tuning across all offline budget settings, providing further evidence of improved network plasticity. Together, these results demonstrate the effectiveness of REFIT in mitigating the plasticity degradation induced by prolonged offline optimization and facilitating subsequent online adaptation.



(a) Online performance (b) Online learning curves (c) FAU

Figure 8: Robustness of REFIT across different offline training budgets. (a) Normalized online performance of REFIT and the corresponding IQL baseline after online fine-tuning under different offline training budgets. (b) Online learning curves of REFIT and the baseline across different offline training budgets. (c) FAU during online fine-tuning.

6 Conclusion

In this work, we investigate O2O RL from the perspective of network plasticity. Our empirical analysis shows that prolonged optimization on static offline data can progressively reduce network plasticity even after offline performance has largely saturated, and that policies retaining higher plasticity tend to exhibit stronger subsequent online adaptation. Motivated by these observations, we propose REFIT, a lightweight model-level method that distills the behavior of an offline teacher into a freshly initialized student policy while temporarily freezing a random subset of student units. This procedure preserves the offline behavioral prior while restoring policy plasticity for subsequent online adaptation. Experiments across D4RL and OGBench demonstrate that REFIT improves aggregate performance for IQL and Cal-QL and achieves higher aggregate scores than the corresponding O2O plug-in methods. These results highlight policy plasticity as an important factor in O2O adaptation and demonstrate the effectiveness of explicitly restoring plasticity at the O2O transition.

AI USE STATEMENT

In this paper, we have used generative AI tools for implementing part of the code of our method. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims and interpreting results. The rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to refine the expression of the paper. All LLM-generated content has been reviewed by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies plasticity loss in O2O RL and proposes a model-level method for improving policy fine-tuning. All experiments are conducted in simulated continuous-control, navigation, and robotic-manipulation benchmarks, including D4RL and OGBench, and do not involve human subjects, personal data or safety-critical systems. Our method does not introduce capabilities that directly target surveillance, identity inference, autonomous weapons, or other clearly harmful applications. While improved O2O RL could in principle accelerate the deployment of learned policies in downstream systems, we do not identify ethical concerns specific to this work beyond those common to general RL research. We therefore do not believe this work requires additional mitigation protocols.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our work. To facilitate reproducibility, we provide our code, including hyperparameter configuration files in the supplementary material, with implementation details, hyperparameters, and hardware specifications documented in Appendix B.

REFERENCES

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado. Loss of plasticity in continual deep reinforcement learning. In *Conference on lifelong learning agents*, pp. 620–636. PMLR, 2023.

Kaixuan Chen, Wei Luo, Shunyu Liu, Yaoquan Wei, Yihe Zhou, Yunpeng Qing, Quan Zhang, Yong Wang, Jie Song, and Mingli Song. Powerformer: A section-adaptive transformer for power flow adjustment. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2204–2215, 2025.

Shibhansh Dohare, Richard S Sutton, and A Rupam Mahmood. Continual backprop: Stochastic gradient descent with persistent randomness. *arXiv preprint arXiv:2108.06325*, 2021.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

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

Caglar Gulcehre, Srivatsan Srinivasan, Jakub Sygnowski, Georg Ostrovski, Mehrdad Farajtabar, Matt Hoffman, Razvan Pascanu, and Arnaud Doucet. An empirical study of implicit regularization in deep offline rl. *arXiv preprint arXiv:2207.02099*, 2022.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, August 2018. URL `http://arxiv.org/abs/1801.01290`. arXiv:1801.01290 [cs].

Jeonghye Kim, Yongjae Shin, Whiyoung Jung, Sunghoon Hong, Deunsol Yoon, Youngchul Sung, Kanghoon Lee, and Woohyung Lim. Penalizing infeasible actions and reward scaling in reinforcement learning with offline data. *arXiv preprint arXiv:2507.08761*, 2025.

Timo Klein, Christoph Luther, Manus McAuliffe, Lukas Miklautz, Claudia Plant, and Sebastian Tschiatschek. Plasticity loss in deep reinforcement learning: A survey. *arXiv preprint arXiv:2411.04832*, 2024.

Rui Kong, Chenyang Wu, Chen-Xiao Gao, Zongzhang Zhang, and Ming Li. Efficient and stable offline-to-online reinforcement learning via continual policy revitalization. In *IJCAI*, pp. 4317–4325, 2024.

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

Aviral Kumar, Rishabh Agarwal, Dibya Ghosh, and Sergey Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. *arXiv preprint arXiv:2010.14498*, 2020a.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020b.

Saurabh Kumar, Henrik Marklund, and Benjamin Van Roy. Maintaining plasticity in continual learning via regenerative regularization, 2024. URL https://arxiv.org/abs/2308.11958.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Qin-Wen Luo, Ming-Kun Xie, Ye-Wen Wang, and Sheng-Jun Huang. Optimistic critic reconstruction and constrained fine-tuning for general offline-to-online rl. *Advances in Neural Information Processing Systems*, 37:108167–108207, 2024.

Yicheng Luo, Jackie Kay, Edward Grefenstette, and Marc Peter Deisenroth. Finetuning from offline reinforcement learning: Challenges, trade-offs and practical solutions, 2023. URL https://arxiv.org/abs/2303.17396.

Clare Lyle, Mark Rowland, and Will Dabney. Understanding and preventing capacity loss in reinforcement learning. *arXiv preprint arXiv:2204.09560*, 2022.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In *International Conference on Machine Learning*, pp. 23190–23211. PMLR, 2023.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, Hado Van Hasselt, Razvan Pascanu, James Martens, and Will Dabney. Disentangling the causes of plasticity loss in neural networks. *arXiv preprint arXiv:2402.18762*, 2024.

Guozheng Ma, Lu Li, Sen Zhang, Zixuan Liu, Zhen Wang, Yixin Chen, Li Shen, Xueqian Wang, and Dacheng Tao. Revisiting plasticity in visual reinforcement learning: Data, modules and training stages. *arXiv preprint arXiv:2310.07418*, 2023.

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

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

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International conference on machine learning*, pp. 16828–16847. PMLR, 2022.

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and André Barreto. Deep reinforcement learning with plasticity injection. *Advances in Neural Information Processing Systems*, 36:37142–37159, 2023.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, volume 2025, pp. 94937–94982, 2025.

Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

Yunpeng Qing, Shunyu Liu, Jingyuan Cong, Kaixuan Chen, Yihe Zhou, and Mingli Song. A2po: Towards effective offline reinforcement learning from an advantage-aware perspective. *Advances in Neural Information Processing Systems*, 37:29064–29090, 2024.

Yunpeng Qing, Yixiao Chi, Shuo Chen, Shunyu Liu, Kexuan Zhou, Sixu Lin, Litao Liu, and Changqing Zou. Bitrajdiff: Bidirectional trajectory generation with diffusion models for offline reinforcement learning. *arXiv preprint arXiv:2506.05762*, 2025.

Yongjae Shin, Jeonghye Kim, Whiyoung Jung, Sunghoon Hong, Deunsol Yoon, Youngsoo Jang, Geonhyeong Kim, Jongseong Chae, Youngchul Sung, Kanghoon Lee, et al. Online pre-training for offline-to-online reinforcement learning. *arXiv preprint arXiv:2507.08387*, 2025.

Ghada Sokar, Rishabh Agarwal, Pablo Samuel Castro, and Utku Evci. The dormant neuron phenomenon in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 32145–32168. PMLR, 2023.

Haoyu Wang, Wei Dai, Jiawei Zhang, Jialun Ma, Mingyi Huang, and Yuguo Yu. The dual nature of plasticity loss in deep continual learning: Dissection and mitigation. *Advances in Neural Information Processing Systems*, 38:69412–69444, 2025.

Feiyang Xu, Shunyu Liu, Yunpeng Qing, Yihe Zhou, Yuwen Wang, and Mingli Song. Temporal prototype-aware learning for active voltage control on power distribution networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3598–3609, 2024.

Haichao Zhang, Wei Xu, and Haonan Yu. Policy expansion for bridging offline-to-online reinforcement learning. *arXiv preprint arXiv:2302.00935*, 2023.

Yinmin Zhang, Jie Liu, Chuming Li, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. A perspective of q-value estimation on offline-to-online reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 16908–16916, 2024.

Huaicheng Zhou, Zifeng Zhuang, and Donglin Wang. Stay hungry, keep learning: Sustainable plasticity for deep reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025a.

Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient online reinforcement learning fine-tuning need not retain offline data. In *International Conference on Learning Representations*, volume 2025, pp. 32343–32368, 2025b.

A PSEUDOCODE

Algorithm 1 REFIT Training Procedure

Require: Offline dataset $\mathcal{D}_{\text{off}}$; environment E; O2O algorithm; freeze ratio ρ.
Require: Budgets $N_{\text{off}} = 10^6$, $N_{\text{distill}} = 10^5$, and $N_{\text{on}} = 3 \times 10^5$.
Phase 1: Offline Pre-training
for $t = 1$ to N_{off} **do**
  Sample batch $(s, a, r, s') \sim \mathcal{D}_{\text{off}}$.
  Update policy π_θ and backbone value functions using the offline RL objective.
end for
Phase 2: Plasticity-Preserving Policy Distillation
Set $\pi_{\text{teacher}} \leftarrow \pi_\theta$ and initialize a fresh student policy π_{student}.
Randomly partition θ_{student} into frozen parameters θ_{freeze} and trainable parameters θ_{train} according to the freeze ratio ρ.
for $k = 1$ to N_{distill} **do**
  Sample $s \sim \mathcal{D}_{\text{off}}$ and update unfrozen student units using $\mathcal{L}_{\text{distill}}$ in Eq. (3)
end for
Replace $\pi_\theta \leftarrow \pi_{\text{student}}$; retain the offline-trained value functions.
Phase 3: Online Fine-tuning
Unfreeze θ_{freeze}, making all student policy parameters trainable.
for $t = 1$ to N_{on} **do**
  Interact with E, add the transition to the replay buffer, update with online RL objective.
end for

B EXPERIMENT DETAILS

B.1 ENVIRONMENT DETAILS

We evaluate D4RL MuJoCo and AntMaze (Fu et al., 2020) and three OGBench manipulation tasks (Park et al., 2025). Table 4 defines the compositional abbreviations used in our result tables. To fit space-constrained tables, we additionally shorten the environment prefixes to `hc`, `hop`, `wal`, and `ant`; for example, `halfcheetah-m`, `walker2d-m-r`, and `antmaze-m-p` become `hc-m`, `wal-m-r`, and `ant-m-p`, respectively. OGBench task names are not abbreviated.

Table 4: Environment naming convention. Tokens are composed from left to right, as illustrated in the examples.

Suite	Token	Meaning	Example
Environment	`hc`	`halfcheetah`	`hc-m`
	`hop`	`hopper`	`hop-m-r`
	`wal`	`walker2d`	`wal-r`
	`ant`	`antmaze`	`ant-l-d`
MuJoCo Postfix	`r`	`random`	`hc-r`
	`m`	`medium`	`hop-m`
	`m-r`	`medium-replay`	`wal-m-r`
AntMaze Postfix	`m-d`	`medium-diverse`	`ant-m-d`
	`m-p`	`medium-play`	`ant-m-p`
	`l-d`	`large-diverse`	`ant-l-d`
	`l-p`	`large-play`	`ant-l-p`

B.2 REFIT IMPLEMENTATION

All experiments use NVIDIA A100 GPUs with CUDA 12.4.

Three-phase training. We first train the selected backbone on $\mathcal{D}_{\text{off}}$ for 10^6 steps. At the O2O transition, we distill the offline teacher into a freshly initialized student policy for 10^5 steps while applying a fixed mask that freezes the kernel columns and biases of $\lfloor \rho d_{\text{out}} \rfloor$ randomly selected output channels in each actor Dense layer, including the policy output head; the critic and value networks are inherited from the offline phase. We then remove the mask and fine-tune all networks for 3×10^5 online steps. Algorithm 1 summarizes the full procedure.

Hyperparameters. Across all environments and both backbones, we use 10^6 offline training steps, 10^5 policy-distillation updates, and 3×10^5 online fine-tuning steps. The default freeze ratio is $\rho = 0.5$; for Cal-QL on MuJoCo, we use $\rho = 0.1$ for HalfCheetah and Hopper and $\rho = 0.5$ for Walker2d. Other settings, such as network architecture, policy learning rate, IQL expectile and temperature, and Cal-QL conservative weight, follow the corresponding backbone implementations. Complete environment-specific configurations, including batch size and normalization, are provided in the configuration files of the supplementary code.

B.3 Backbone Implementation

We evaluate REFIT on IQL (Kostrikov et al., 2021) and Cal-QL (Nakamoto et al., 2023), comparing against OPT (Shin et al., 2025) and PARS (Kim et al., 2025), respectively. Our implementations build on the following public repositories:

- IQL: `https://github.com/ikostrikov/implicit_q_learning`
- Cal-QL: `https://github.com/nakamotoo/Cal-QL`
- OPT: `https://github.com/LGAI-Research/opt`
- PARS: `https://github.com/LGAI-Research/pars`

C Computational Cost

In this section, we evaluate the computational overhead of REFIT by comparing the wall-clock time of IQL+REFIT with that of the IQL baseline on `halfcheetah-random-v2`. All runs are conducted on a single NVIDIA A100 GPU and averaged over five random seeds.

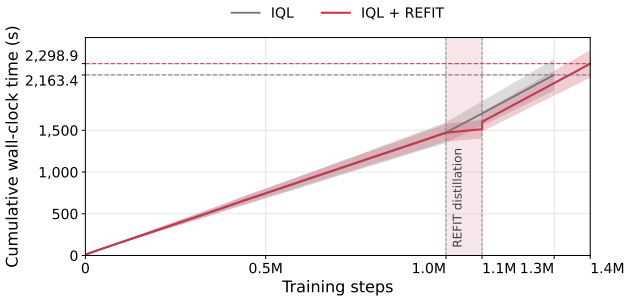


Figure 9: Wall-clock time of IQL+REFIT and the IQL baseline, averaged over five random seeds.

As shown in Figure 9, the complete pipeline requires 2163.4 seconds for IQL and 2298.9 seconds for IQL+REFIT, corresponding to a 6.26% increase in wall-clock time. The small offset near the end of distillation arises from one-time JAX just-in-time compilation when the distillation update and student policy are initialized. After compilation, the two curves progress approximately in parallel, indicating that REFIT introduces limited sustained overhead beyond the additional distillation phase.

D Additional Comparison with CPR

CPR (Kong et al., 2024) links primacy bias to limited O2O adaptation, but does not directly quantify offline-induced policy plasticity loss with network-level diagnostics. Its periodic policy revitaliza-

tion retains earlier policies for action selection. REFIT instead diagnoses this loss and applies a single model-level intervention at the O2O transition, then fine-tunes one actor online. Table 5 compares both methods on IQL and Cal-QL across six D4RL tasks. We adapt CPR from their released code (`https://github.com/LAMDA-RL/CPR`), following its method-specific hyperparameters where applicable.

Table 5: Normalized scores on six representative tasks.

Method	hc-m	hc-r	wal-m-r	hop-m-r	ant-m-p	ant-m-d	Total
IQL	48.1 ± 0.1	13.8 ± 3.9	74.8 ± 2.8	77.5 ± 18.1	76.2 ± 1.5	71.6 ± 2.7	362.0
IQL+CPR	**51.7** ± 0.5	26.0 ± 4.0	86.0 ± 11.7	101.8 ± 3.9	66.0 ± 3.3	71.0 ± 6.8	402.5
IQL+REFIT	50.0 ± 0.3	**46.3** ± 1.4	**96.9** ± 2.2	**102.7** ± 0.4	**86.6** ± 4.8	**89.2** ± 2.7	**471.7**
Cal-QL	**65.3** ± 3.0	33.4 ± 3.8	86.7 ± 8.7	68.9 ± 7.8	92.2 ± 2.1	93.2 ± 0.8	439.7
Cal-QL+CPR	43.3 ± 0.6	2.2 ± 0.1	17.6 ± 2.8	30.3 ± 7.9	27.0 ± 10.3	26.0 ± 13.9	146.3
Cal-QL+REFIT	51.1 ± 0.9	**39.2** ± 3.5	**97.9** ± 5.8	**87.6** ± 7.5	**95.4** ± 1.7	**96.2** ± 0.8	**467.4**

These results demonstrate that REFIT outperforms CPR in most cases by directly restoring policy plasticity at the model level.

E Supplementary Ablation Analysis

This section provides supplementary ablations that complement the analysis in Section 5.3. We examine the necessity of policy distillation and network reinitialization, the sensitivity to the distillation loss function, the effect of standard-deviation transfer on exploration, and the role of unfreezing during online adaptation.

E.1 Ablation of Policy Distillation

To evaluate the role of distillation in mitigating policy degradation during the O2O transition, we compare REFIT against a *direct reset* baseline (referred to as From-Scratch), which randomly reinitializes the network parameters at the start of online fine-tuning without any behavioral prior from the offline teacher.

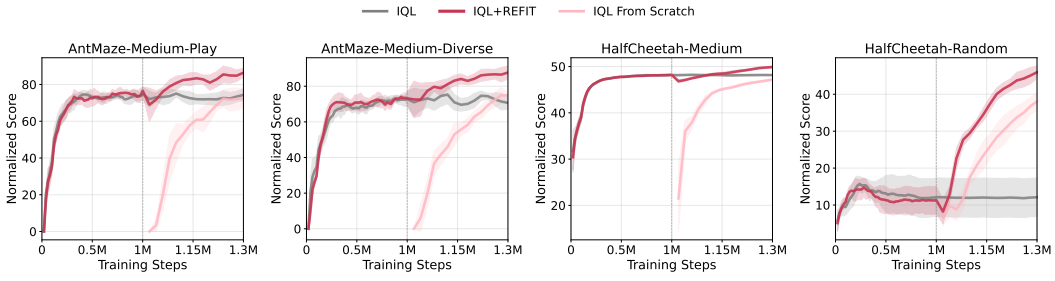


Figure 10: Performance comparison between IQL, IQL From Scratch and IQL+REFIT

As shown in Figure 10, although the direct reinitialization baseline outperforms standard fine-tuning on aggregate by restoring network plasticity, it consistently falls short of the full REFIT configuration. This result confirms that combining policy distillation with structured reinitialization effectively preserves the offline behavioral prior, whereas naive parameter resetting discards the behavioral prior acquired during offline training.

E.2 Ablation of Policy Reinitialization

A natural question arises regarding the distillation phase: *is it essential to freshly initialize the online student policy*, or could one simply freeze parts of the network during offline pre-training and unfreeze them for online fine-tuning? To isolate the effect of network reinitialization, we compare the default REFIT method against this unfreezing baseline across four representative environments. As shown in Figure 11, directly unfreezing offline pre-trained parameters leads to substantial performance degradation, particularly on tasks requiring significant adaptation (e.g., `halfcheetah-r` drops from 46.3 to 14.7). These results indicate that fresh initialization is essential for overcoming the optimization bottlenecks accumulated during offline training and enabling effective online policy improvement.

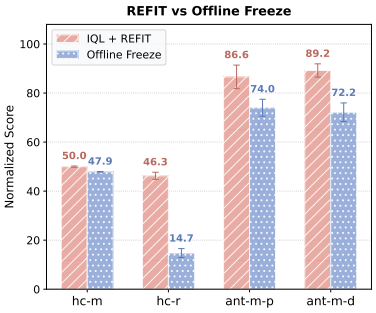


Figure 11: Impact of network reset during distillation.

E.3 Ablation of Distillation Loss Function

REFIT adopts mean squared error (MSE) as its default distillation objective due to computational simplicity: MSE directly regresses the student policy toward the mean action of the frozen teacher policy, without requiring action sampling or full distribution matching. To verify that the method is not tied to a particular choice of loss function, we compare MSE against forward KL divergence and reverse KL divergence, keeping all other components unchanged.

Table 6: Ablation of distillation loss. Results report mean ± standard deviation.

Environment	KL_{fwd}	KL_{rev}	MSE
antmaze-m-d	86.9 ± 6.7	87.0 ± 5.2	**89.2** ± 2.7
antmaze-m-p	**87.8** ± 2.7	84.8 ± 5.2	86.6 ± 4.8
halfcheetah-m	49.9 ± 0.2	50.0 ± 0.6	50.0 ± 0.3
halfcheetah-r	46.3 ± 1.5	**47.5** ± 1.7	46.3 ± 1.4
Total Score	270.9 ± 2.9	269.2 ± 2.9	**272.1** ± 2.8

As shown in Table 6, all three objectives yield comparable aggregate performance across the evaluated environments, with per-task differences largely within the reported standard deviations. This result confirms that the effectiveness of REFIT stems from its structural design—fresh initialization, partial freezing, and knowledge transfer—rather than from the specific form of the distillation loss. We therefore adopt MSE as the default for its simplicity.

E.4 Ablation of Standard-Deviation Transfer

Because REFIT reinitializes the student network, the log-standard-deviation parameters of the policy are also reset to their default initial values, which may differ from those learned during offline training. A natural concern is whether the performance gain of REFIT is primarily attributable to the increased policy standard deviation induced by this reinitialization, rather than to the restoration of network plasticity.

Table 7: Ablation of standard-deviation transfer. "Default" denotes the standard REFIT configuration where the log-standard-deviation parameters are reset along with the rest of the student network; "Copy Std" denotes the variant that inherits the standard deviation from the offline teacher. Results are reported as mean ± standard deviation across random seeds.

Method	antmaze-m-d	antmaze-m-p	halfcheetah-m	halfcheetah-r	Total Score
Default	**89.2** ± **2.7**	**86.6** ± **4.8**	50.0 ± 0.3	46.3 ± 1.4	**272.1** ± **2.8**
Copy Std	86.4 ± 4.5	84.3 ± 4.0	**50.1** ± **0.3**	**46.9** ± **2.9**	267.7 ± 2.3

To disentangle these two factors, we evaluate a *Copy Std* variant that directly inherits the standard-deviation parameters from the offline teacher after distillation, so that the student begins online fine-tuning with the same exploration profile as the teacher. As shown in Table 7, the Copy Std variant achieves performance comparable to the default REFIT configuration across all evaluated environments, with aggregate scores differing by only a small margin. This result indicates that

the performance gain of REFIT does not rely on increased exploration induced by resetting the standard deviation, and is instead consistently attributable to the plasticity restoration provided by the structured reinitialization and distillation procedure.

E.5 Ablation of Online Unfreezing

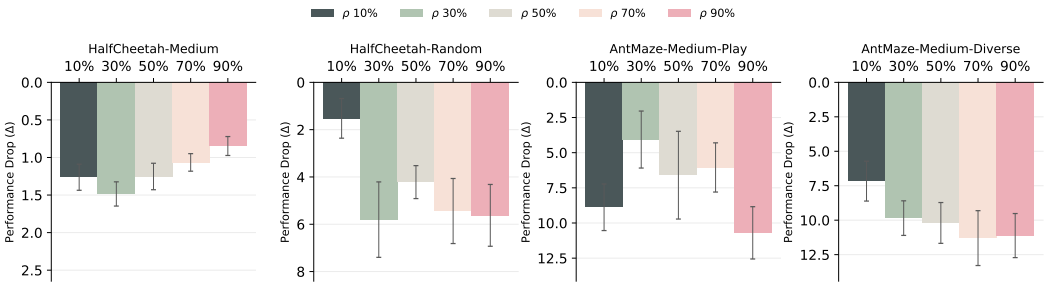


Figure 12: Online performance with and without unfreezing the parameter subset after distillation.

A final design question is whether the parameter subset frozen during distillation should remain frozen during online fine-tuning. The *online-freeze* variant retains the same mask throughout the online phase, whereas the default REFIT configuration removes the mask and updates the entire policy. As shown in Figure 12, retaining the mask consistently limits online improvement and yields lower asymptotic performance. Unfreezing is therefore necessary to recover the full optimization capacity of the student policy during online adaptation.

F Supplementary Plasticity Diagnostics

F.1 Definitions of Additional Plasticity Metrics

Here we introduce the two other metrics we used in the experiment:

1. *Weight Norm*: The magnitude of network parameters provides a complementary diagnostic of network plasticity (Lyle et al., 2024). We track the L2 norm of all actor parameters:

$$\|\boldsymbol{\theta}\|_{\text{weight}} = \sqrt{\sum_{i=1}^{P} \theta_i^2}, \tag{4}$$

where $\boldsymbol{\theta} \in \mathbb{R}^P$ contains all actor parameters, including weights, biases, and policy standard-deviation parameters. We interpret changes in this norm alongside FAU and SRank, rather than as direct evidence of plasticity loss or restoration.

2. *SRank*: SRank measures the effective dimensionality of learned representations (Kumar et al., 2020a). For an actor hidden layer, let $F \in \mathbb{R}^{d\times m}$ denote the matrix of post-ReLU activations for d sampled states and hidden width m, and let $\sigma_1(F) \geq \cdots \geq \sigma_q(F)$ be its singular values, where $q = \min(d, m)$. We define SRank as the minimum number of leading singular values required to retain a fraction $1 - \epsilon$ of the total singular-value mass:

$$\text{SRank}_\epsilon(F) = \min\left\{ k : \frac{\sum_{j=1}^{k} \sigma_j(F)}{\sum_{j=1}^{q} \sigma_j(F)} \geq 1 - \epsilon \right\}, \qquad \epsilon = 0.01. \tag{5}$$

Unlike an absolute singular-value threshold, this relative criterion is invariant to a global rescaling of F and therefore separates changes in spectral shape from changes in feature magnitude. We compute SRank separately for each actor hidden layer using 10 mini-batches of $d = 256$ states sampled uniformly from the replay buffer, average the resulting values across batches and layers, and record the metric every 5000 policy updates. Because $d = 256$, the reported SRank is upper bounded by 256. A higher value indicates that the actor representation distributes its spectral mass across more directions, providing a complementary measure of representational capacity rather than a direct measure of performance.

F.2 Additional Diagnostic Results

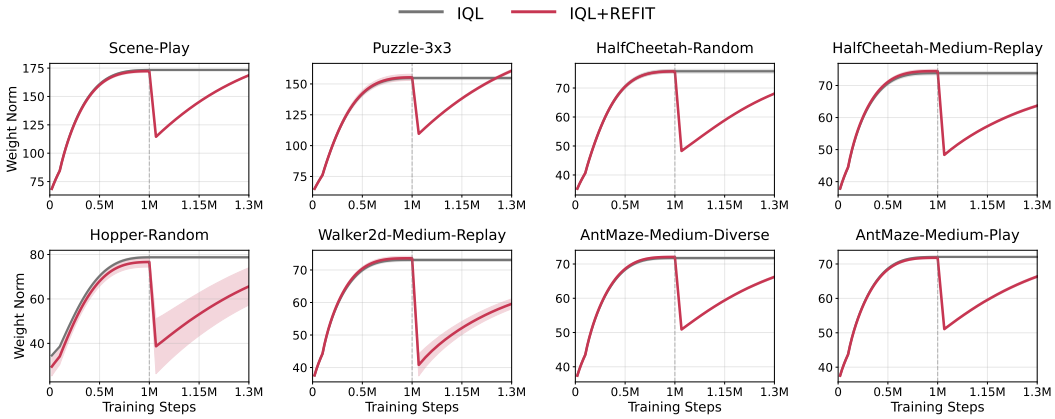


Figure 13: Weight Norm of REFIT and baseline throughout the training process.

Weight Norm Figure 13 depicts the evolution of the actor weight norm throughout training. The baseline maintains a relatively high weight norm across the O2O transition, whereas REFIT produces a sharp reduction at the start of online fine-tuning, consistent with transferring the offline policy behavior into a freshly initialized student. These results show that REFIT changes the parameter scale at the transition and, together with the FAU and SRank diagnostics, provide complementary evidence for its plasticity-restoring effect.

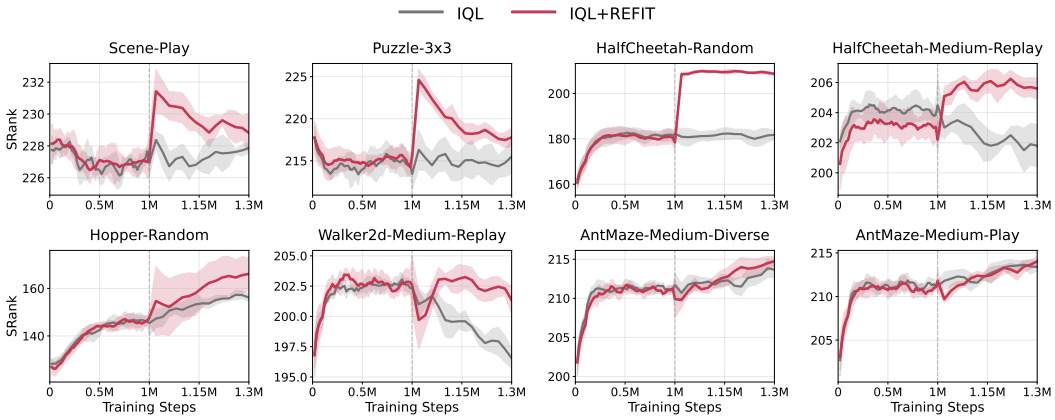


Figure 14: SRank of REFIT and baseline throughout the training process.

SRank Finally, we evaluate the representational richness using SRank, where a higher value signifies a more diverse and robust feature space (Gulcehre et al., 2022; Lyle et al., 2023). As shown in Figure 14, transferring directly from O2O learning often causes the baseline representations to suffer from representational stagnation, characterized by a stabilized yet severely bounded SRank that fails to expand when encountering new online environments. REFIT increases SRank sharply on several tasks and is comparable on others; the relationship is task-dependent. This preservation of higher-rank representations suggests that the network retains sufficient plasticity and adaptive capacity to capture the novel dynamics encountered during online exploration.

G LIMITATIONS AND DISCUSSION

While REFIT consistently improves aggregate performance and outperforms the baselines and other plug-in methods on the majority of environments, its primary contribution lies in addressing O2O adaptation from the perspective of network plasticity. This perspective is largely orthogonal to existing methods that target other bottlenecks, such as inaccurate value estimation (Shin et al., 2025). A more systematic understanding of how these factors interact, as well as how complementary mechanisms can be jointly incorporated into O2O RL, remains an interesting direction for future work.