SUF: Stabilized Unconstrained Fine-Tuning for Offline-to-Online Reinforcement Learning

Jiaheng Feng[1], **Mingxiao Feng**[1], **Haolin Song**[1], **Wengang Zhou**[1,2], **Houqiang Li**[1,2]

[1]EEIS Department, University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{fengjiaheng, fmxustc, hlsong}@mail.ustc.edu.cn, {zhwg, lihq}@ustc.edu.cn

Abstract

Offline-to-online reinforcement learning (RL) provides a promising solution to improving suboptimal offline pre-trained policies through online fine-tuning. However, one efficient method, unconstrained fine-tuning, often suffers from severe policy collapse due to excessive distribution shift. To ensure stability, existing methods retain offline constraints and employ additional techniques during fine-tuning, which hurts efficiency. In this work, we introduce a novel perspective: eliminating the policy collapse without imposing constraints. We observe that such policy collapse arises from the mismatch between unconstrained fine-tuning and the conventional RL training framework. To this end, we propose Stabilized Unconstrained Fine-tuning (SUF), a streamlined framework that benefits from the efficiency of unconstrained fine-tuning while ensuring stability by modifying the Update-To-Data ratio. With just a few lines of code adjustments, SUF demonstrates remarkable adaptability to diverse backbones and superior performance over state-of-the-art baselines.

Introduction

Reinforcement learning (RL) has demonstrated great success across various tasks, including board games (Silver et al. 2017) and video games (Mnih et al. 2015). However, online RL demands extensive environmental interactions initialized from a random policy (Sutton and Barto 2018), which may be impractical in realistic scenarios for expense or safety concerns (Nair et al. 2020; Zheng et al. 2023; Zhang, Xu, and Yu 2022). Offline RL provides a practical solution by learning from a pre-collected dataset (Levine et al. 2020). However, the performance of offline policy is heavily limited by the quality and state-action space coverage of the dataset (Jin, Yang, and Wang 2021). Recent studies (Zheng et al. 2023; Zhang, Xu, and Yu 2022) investigate offline-to-online RL as a promising solution to improving suboptimal offline policies through online fine-tuning.

Unfortunately, ensuring both efficient and stable fine-tuning remains a main challenge in offline-to-online RL, which can be attributed to the inherent conflict between exploring for improvements and exploiting for maintaining the learned behaviors (Mark et al. 2022; Guo et al. 2023). In this paper, we term this challenge as the *efficiency-stability dilemma* in offline-to-online RL. The optimal fine-tuning approach is to remove offline constraints in the online phase (Wu et al. 2022a), thereby facilitating the exploration and promoting efficient improvements. However, previous studies (Nair et al. 2020; Zhang, Xu, and Yu 2022) have observed severe policy collapse at the initial stage of unconstrained fine-tuning, which is usually unacceptable in practice (Beeson and Montana 2022). As even short-lived policy collapse can lead to immature or dangerous actions, resulting in irreversible damage to the environment or the agent. To the best of our knowledge, the problem of policy collapse during unconstrained fine-tuning has not been well addressed.

To skirt around such policy collapse, previous methods retain part (Zheng et al. 2023; Guo et al. 2023; Beeson and Montana 2022) or whole (Mark et al. 2022) of offline constraints, carefully adjust constraint strength for different tasks (Zhao et al. 2022) or introduce alternative constraints (Luo et al. 2022; Li et al. 2023). However, existing constraints hinder the exploration of the environment and the efficiency of fine-tuning. Simultaneously, many complex techniques have been imposed in offline-to-online RL, including density estimation network (Lee et al. 2022; Guo et al. 2023), ensembled networks (Mark et al. 2022; Zheng et al. 2023; Zhao et al. 2022), and model-based method (Mao et al. 2022), resulting in increased complexity and limited adaptability.

In this work, we introduce a novel perspective to circumvent the *efficiency-stability dilemma*: eliminating policy collapse without imposing constraints. We note that such policy collapse arises from the mismatch between unconstrained fine-tuning and conventional RL training framework. Typically, the update-to-data (UTD) ratio (the number of parameter updates per environment step) is set to 1 for both the value network (critic) and policy network (actor). However, the exploratory unconstrained objective leads to significant distribution shift, which exceeds the ability of value fitting, resulting in excessive value bias on out-of-distribution (OOD) data. Simultaneously, such frequent policy updates exacerbate the risk of policy being misguided. To this end, we introduce an effective framework for unconstrained fine-tuning, which involves *increasing* the Critic UTD to expedite the fitting of the value network and *decreasing* the Actor UTD to improve the accuracy of policy updates. However, previous studies (Chen et al. 2020; Li et al. 2022) in

online RL have observed that increased UTD can lead to inferior performance. Despite (Chen et al. 2020; Wu et al. 2022b) address this problem by employing ensembled networks, which in turn introduces increased parameters and complexity. Fortunately, we observe that in offline-to-online RL, the accessible dataset and the pre-trained policy initialization enable agents to benefit from high UTD learning without employing ensemble. Consequently, we exclude ensemble from our framework to reduce complexity.

Our contributions can be summarized as follows:

- We introduce a novel perspective to tackle the *efficiency-stability dilemma* in offline-to-online RL: eliminating the policy collapse without imposing any constraints.
- We point out that the conventional RL training framework is inapplicable in unconstrained offline-to-online RL. As a solution, we propose **S**tabilized **U**nconstrained **F**ine-tuning (SUF) framework, which effectively ensures stability during unconstrained fine-tuning.
- SUF can be easily implemented and widely adapted to diverse offline RL backbones. Experimental results demonstrate its superiority over state-of-the-art (SOTA) baselines across various environments and datasets.

Related Work

Offline-to-Online RL

Offline-to-online RL aims to improve suboptimal offline policies through online fine-tuning. Many previous studies (Lee et al. 2022; Mark et al. 2022; Beeson and Montana 2022; Zhao et al. 2022; Nakamoto et al. 2023; Yu and Zhang 2023) focus on fine-tuning based on specific offline RL backbones. However, practical scenarios may involve agents pre-trained by various offline RL algorithms, highlighting the necessity for developing a generic offline-to-online RL framework. Recent studies place a growing emphasis on adaptability. PEX (Zhang, Xu, and Yu 2022) freezes the pre-trained policy and initializes a random policy to enhance exploration. PROTO (Li et al. 2023) gradually evolves the regularization term to relax the constraint strength. From a data-centric perspective, APL (Zheng et al. 2023) and SUNE (Guo et al. 2023) impose constraints exclusively on data from offline datasets and data with high uncertainty, respectively. In contrast, SUF operates without imposing any constraints, thereby ensuring efficient fine-tuning.

UTD in RL

In online RL, recent studies focus on the utilization methodology of UTD, driven by its potential for improved efficiency. REDQ (Chen et al. 2020) and its variant (Wu et al. 2022b) employ ensembled value networks to mitigate the value bias caused by increased UTD. (Dorka, Welschehold, and Burgard 2023) addresses model overfitting by employing dynamic UTD. (Li et al. 2022) investigates the factors contributing to inferior performance in high UTD learning. However, in offline-to-online RL, previous studies (Zhao et al. 2022; Mark et al. 2022; Zheng et al. 2023) primarily introduce powerful online backbones (e.g., REDQ) to constrained fine-tuning. In contrast, SUF modifies UTD to address policy collapse during unconstrained fine-tuning. We further supplement the studies of UTD under the setting of offline-to-online RL and reasonably exclude ensemble from our framework, effectively reducing the complexity.

Preliminaries and Background

Problem Definition

We follow the standard RL paradigm, which can be modeled as a Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, state transition function $\mathcal{P}(s'|s,a)$, reward function $r(s,a)$ and discount factor $\gamma \in [0,1)$. The objective of RL is to find a policy $\pi(a|s)$ that maximizes the discounted return $R = \sum_{t=0}^{\infty} \gamma^t r(s,a)$. The state-action value function $Q_\pi(s,a) = \mathbb{E}_\pi[R|s,a]$ represents the expected discounted return after performing the action a in state s and following the policy π.

Off-Policy RL

We mainly focus on off-policy RL methods (Fujimoto, Hoof, and Meger 2018; Haarnoja et al. 2018) for online fine-tuning, because of their efficient exploitation of historical data from replay buffer $\mathcal{B}$. Off-policy RL methods typically employ temporal difference learning to iteratively update the value network $Q_\theta(s,a)$ (i.e., critic, parameterized by θ), and improve the policy network $\pi_\phi(a|s)$ (i.e., actor, parameterized by ϕ) through value maximization:

$$Q_\theta(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}, a' \sim \pi_\phi(\cdot|s')} \left[Q_\theta\left(s', a'\right) \right], \quad (1)$$

$$\max_\phi \mathbb{E}_{s \sim \mathcal{B}, a \sim \pi_\phi(\cdot|s)} \left[Q_\theta\left(s, a\right) \right]. \quad (2)$$

With access to the environment, off-policy RL methods typically prioritize optimistic exploration of the environment instead of imposing conservative constraints on the agent. For example, TD3 (Fujimoto, Hoof, and Meger 2018) uses random noise to augment the output actions, and SAC (Haarnoja et al. 2018) introduces an additional maximum entropy objective to enhance exploration further. However, online RL demands extensive environmental interactions initialized from a random policy, posing challenges in many realistic scenarios (Nair et al. 2020; Zheng et al. 2023; Zhang, Xu, and Yu 2022).

Offline RL

Offline RL aims to learn policy from a fixed dataset $\mathcal{D}$ without environmental interactions. To mitigate the well-known extrapolation error in value networks for OOD actions (Fujimoto, Meger, and Precup 2019; Kumar et al. 2020), offline RL methods typically constrain the policy to perform actions close to the dataset through policy constraint (Fujimoto, Meger, and Precup 2019; Kumar et al. 2019; Fujimoto and Gu 2021), value regularization (Kumar et al. 2020; An et al. 2021), in-sample learning (Kostrikov, Nair, and Levine 2021; Garg et al. 2022; Xiao et al. 2022), etc. As discussed in (Guo et al. 2023), most model-free offline RL methods can be briefly summarized by introducing an additional regularizer $\mathcal{R}_{\text{reg}}$ to the online RL objective:

$$\mathcal{J}_{\text{offline}} = \mathcal{J}_{\text{online}} + \lambda \mathcal{R}_{\text{reg}}, \quad (3)$$

where λ is a trade-off coefficient.

Method

In this section, we first analyze the *efficiency-stability dilemma* in offline-to-online RL and the reasons behind policy collapse during unconstrained fine-tuning. Then, we explain our motivation for tackling these problems. Further, we supplement the studies of UTD under the setting of offline-to-online RL. Finally, we present our SUF framework and the detailed algorithm for its implementation.

Efficiency-Stability Dilemma in Offline-to-Online RL

Offline RL prioritizes the effective exploitation of existing datasets, while online RL prioritizes the efficient exploration of new actions. This indicates the inherent challenge that offline-to-online RL demands a proper bridge between the conflicting objectives of offline pre-training and online fine-tuning.

To illustrate this challenge, we investigate two typical methods in offline-to-online RL: constrained fine-tuning and unconstrained fine-tuning. Specifically, we perform 1 million pre-training steps on `walker2d-medium` using IQL (Kostrikov, Nair, and Levine 2021), an advanced offline RL method. In the online phase, we use IQL for constrained fine-tuning and SAC for unconstrained fine-tuning, respectively. Figure 1(a-b) presents the learning curves of normalized return and the density histogram of action distance at 50000 steps for each method. The action distance $\mathbb{E}_{(s,a)\sim\mathcal{D},\hat{a}\sim\pi_\phi(\cdot|s)}\left[\|\hat{a}-a\|_2^2\right]$ illustrates the offset between the policy-induced data distribution and the dataset.

As shown in Figure 1(a-b), the constrained method restricts the policy to perform actions close to the dataset, which limits the exploration of new actions and leads to stable yet inefficient improvements. In contrast, the unconstrained objective encourages policy to explore a broader action space outside the dataset, leading to efficient improvements yet severe policy collapse at the initial fine-tuning stage. In this paper, we term this phenomenon as the *efficiency-stability dilemma* in offline-to-online RL.

To skirt around such policy collapse, previous methods retain constraints and introduce complex techniques, resulting in inefficiency and complexity. A natural question thus arises: *Can policy collapse be eliminated without imposing any constraints?* It motivates us to develop a stabilized unconstrained fine-tuning framework, thereby circumventing this *dilemma*.

Eliminating Policy Collapse without Imposing Constraints

To investigate the underlying reasons behind policy collapse during unconstrained fine-tuning, we evaluate the quality of value estimates on data newly collected in the online phase. Specifically, we randomly sample 1000 state-action pairs from online replay buffer $\mathcal{B}$ every 5000 environment steps. For each pair, we obtain the value estimate Q_θ through the value network and the true value R through the Monte Carlo method. Then we compute the normalized bias $\left|\left(Q_\theta(s,a)-R(s,a)\right)/\mathbb{E}_{(s,a)\sim\mathcal{B}}\left[R(s,a)\right]\right|$ between them. Following (Chen et al. 2020), we present the average normalized bias (ANB) in Figure 1(c) and the standard deviation of normalized bias (SNB) in Figure 1(d). Among them, SNB quantifies the uniformity of bias across various state-action pairs. As discussed in (Chen et al. 2020), non-uniform bias can severely impair policy learning by significantly changing the action selection.

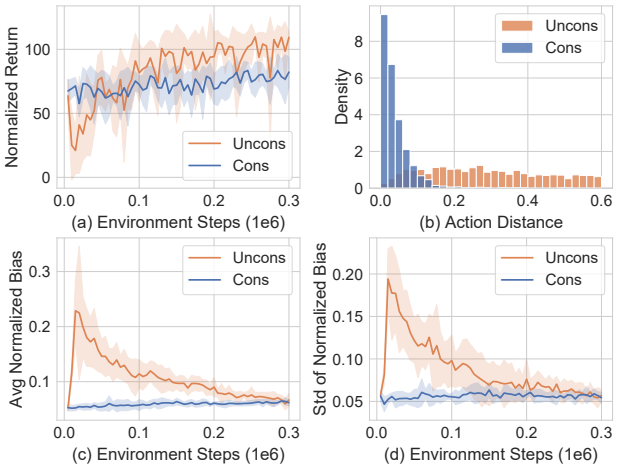

Figure 1: Metrics for unconstrained fine-tuning (denoted as `Uncons`) and constrained fine-tuning (denoted as `Cons`) in `walker2d-medium`. (a) Learning curves of normalized return. (b) Density histograms of distance between actions from the policy and actions from the dataset. (c-d) Average and standard deviation of normalized bias in value estimates.

Regarding the unconstrained method, Figure 1(b) has illustrated its distribution shift between the policy-induced data and the offline dataset. However, the pre-trained value network struggles to promptly provide accurate estimates for such OOD data, resulting in excessive ANB and SNB at the initial fine-tuning stage, as shown in Figure 1(c-d). Further, the value bias propagates across a broader state-action space due to bootstrapping and the generalization of neural networks, which severely misguides the policy and undermines the learned behavior. In turn, the deterioration of policy exacerbates the distribution shift, generating a vicious circle. Encouragingly, this circle is surmountable. Figure 1 demonstrates that the increasing accuracy of value estimates in the unconstrained method enables more effective policy improvements than the constrained method. Briefly, we can attribute the policy collapse during unconstrained fine-tuning to the following two sequential problems:

- **Problem 1**: Value network underfitting on OOD data, leading to estimation bias.
- **Problem 2**: Policy misguidance from value bias.

We note that these problems arise from the conventional RL training framework, where the value network and policy network are updated once per environment step (i.e., both Critic UTD and Actor UTD are set to 1). Due to the abrupt and significant distribution shift in unconstrained fine-tuning, this setting will severely hinder the fitting of value networks on OOD data, lead to excessive value bias,

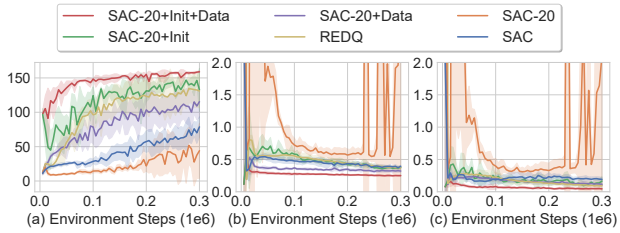

Figure 2: (a) Learning curves of normalized return, (b) ANB, and (c) SNB in `Ant`. `+Data` denotes access to `ant-random` dataset. `+Init` denotes initialization with IQL agent pre-trained on `ant-medium` dataset.

and exacerbate the risk of policy being misguided. Based on these insights, we propose a unified framework for unconstrained fine-tuning, involving the following solutions:

- **Solution 1**: Increase the Critic UTD to expedite the fitting of the value network, addressing Problem 1.
- **Solution 2**: Decrease the Actor UTD to improve the accuracy of policy updates, addressing Problem 2.

Note that the complete solution of Problem 1 inherently solves Problem 2. However, our experiments in Hyperparameter Analysis indicate the challenge of further mitigating policy collapse by solely increasing the Critic UTD. In such instances, decreasing the Actor UTD proves a more effective solution. In Ablation Study, we will verify and explain the effectiveness of Solution 1 and Solution 2, respectively.

Rethinking the Necessity of Ensemble

In Solution 1, we increase the Critic UTD to address Problem 1. However, previous studies (Chen et al. 2020; Li et al. 2022) have observed that directly increasing UTD in online RL can severely impair the value estimates on some tasks. To verify this issue, we present the ANB and SNB of vanilla SAC and its variants in Figure 2(b-c), computed in the same way as the results in Figure 1(c-d). As expected, with a Critic UTD of 20, `SAC-20` exhibits significant value bias, resulting in inferior performance compared with the vanilla SAC. Despite REDQ (Chen et al. 2020) mitigates this value bias by employing 10 ensembled value networks, which in turn introduces increased parameters and complexity. We note that the settings of offline-to-online RL differ from online RL, particularly in terms of:

- Access to the offline dataset.
- Initialization through a pre-trained agent.

Such properties of offline-to-online RL prompt us to rethink the necessity of ensemble in our framework. To this end, we investigate their impacts on high UTD learning, respectively.

Impact of Offline Data We allow `SAC-20` to access the offline dataset. Herein, we use the `random` dataset (about 1 million in size) to isolate the impact of data quality. As shown in Figure 2, `SAC-20+Data` effectively mitigates the value bias observed in `SAC-20`, leading to relatively superior performance.

Impact of Offline Initialization Further, we initialize `SAC-20` through an agent pre-trained on `medium` dataset for 1 million steps using IQL, denoted as `SAC-20+Init`. Herein, we disabled its access to the dataset to isolate the impact of offline data. Figure 2 demonstrates that `SAC-20+Init` exhibits lower bias and outperforms `SAC-20` significantly. Remarkably, the combination of offline data and offline initialization `SAC-20+Init+Data` exhibits the lowest bias and the best performance.

The aforementioned experiments demonstrate that the increased data or improved policy initialization can mitigate the value bias in high UTD learning. Herein, we offer possible explanations for such observations.

Online RL necessitates data collection from scratch. As discussed in previous studies (Nikishin et al. 2022; Li et al. 2022; D'Oro et al. 2022), performing large updates on the limited low-quality data severely impairs the value estimates, which is termed as primacy bias (Nikishin et al. 2022) or statistical overfitting (Li et al. 2022) in online RL. In contrast, offline-to-online RL harnesses broader data from datasets, thereby facilitating a smoother fitting of the value network. Simultaneously, the initialization through a pre-trained policy generates higher quality data at the initial training stage. These properties effectively mitigate the value bias in online RL, enabling agents to benefit from high UTD learning without employing ensemble. The aim of this work is to offer a streamlined framework for offline-to-online RL, thus we exclude ensemble to reduce complexity, Nonetheless, the practical implementation allows the combination of our framework with various techniques, including ensemble, to further improve the performance.

We believe the significance of investigating the properties of UTD in offline-to-online RL and simplifying our framework accordingly, which have been largely overlooked in previous studies (Zhao et al. 2022; Mark et al. 2022; Zheng et al. 2023). In the next section, we will demonstrate the superiority of the proposed framework over constrained methods with ensemble.

SUF Framework for Offline-to-Online RL

We name our framework as **S**tabilized **U**nconstrained **F**ine-tuning (SUF) and summarize it in Algorithm 1. SUF entails (1) removing the offline constraints before transitioning to the online phase and (2) modifying the default UTD of actor and critic. These can be easily implemented by making minor code adjustments to the original algorithm, which are underlined in Algorithm 1. Note that SUF does not change the existing pre-training process, enabling seamless integration with diverse offline RL backbones.

Experiments

In this section, we conduct experiments to answer the following questions: (1) Can SUF stabilize unconstrained fine-tuning by eliminating policy collapse? (2) Can SUF outperform SOTA baselines when combined with diverse offline RL backbones, including IQL, TD3-BC, and CQL? (3) What are the contributions of each component in SUF? (4) What are the impacts of different hyperparameters on SUF?

Algorithm 1: SUF pseudo-code

Input: Offline RL algorithm $\mathcal{F}_{\text{offline}}$, offline dataset $\mathcal{D}$, pre-trained value network Q_θ, pre-trained policy network π_ϕ, total fine-tuning steps T, Critic UTD G_c, Actor UTD G_a.
Initialize: Remove the constraints of $\mathcal{F}_{\text{offline}}$, named $\mathcal{F}_{\text{online}}$. Initialize online replay buffer $\mathcal{B} \leftarrow \varnothing$.

for $t = 0$ **to** T **do**
  $a \sim \pi(\cdot|s), s' \sim \mathcal{P}(\cdot|s, a)$
  $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s, a, r(s, a), s')\}$
  for G_c updates **do**
    Sample a mini-batch B from $\mathcal{D} \cup \mathcal{B}$
    Update Q_θ using B according to $\mathcal{F}_{\text{online}}$
  end for
  if $t \bmod (1/G_a) == 0$ **then**
    Update π_ϕ using B according to $\mathcal{F}_{\text{online}}$
  end if
end for

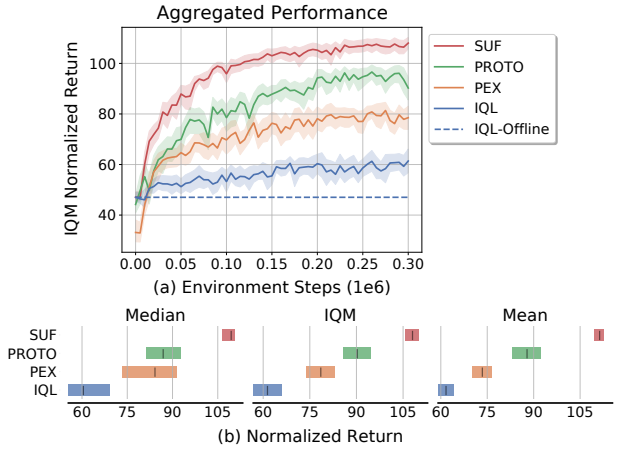

Figure 3: (a) Aggregated learning curves of interquartile mean (IQM) normalized return and (b) aggregated metrics over 5 seeds on 12 MuJoCo tasks from D4RL, with pointwise 95% stratified bootstrap confidence intervals, following the method by (Agarwal et al. 2021). Dashed line `IQL-Offline` shows the initial pre-training performance.

Experimental Setup

Tasks We consider all MuJoCo (Todorov, Erez, and Tassa 2012) environments from the public D4RL (Fu et al. 2020) benchmark: `Halfcheetah`, `Hopper`, `Walker2d`, and `Ant`. To investigate effective improvements for various suboptimal policies, we use three suboptimal dataset types: `random`, `medium`, and `medium-replay` with the latest `v2` version, following the compared baselines (Zhang, Xu, and Yu 2022; Li et al. 2023; Guo et al. 2023).

Baselines We compare SUF with the latest generic frameworks in offline-to-online RL: **(1) PEX** (Zhang, Xu, and Yu 2022) freezes the pre-training policy and introduces policy expansion to enhance exploration. **(2) PROTO** (Li et al. 2023) gradually evolves the regularization term to relax the constraint strength. **(3) APL** (Zheng et al. 2023) leverages the distinct advantages of offline and online data for adaptive constraints. **(4) SUNG** (Guo et al. 2023) controls constraints based on the uncertainty of data quantified by a VAE density estimator.

Offline RL Backbones To demonstrate the adaptability of SUF, we instantiate it on diverse offline RL backbones, including **(1) IQL** (Kostrikov, Nair, and Levine 2021): an in-sample learning-based method, **(2) TD3-BC** (Fujimoto and Gu 2021): a policy constraint-based method, and **(3) CQL** (Kumar et al. 2020): a value regularization-based method.

Settings We compare SUF-IQL with PEX and PROTO, as they are built upon IQL and an enhanced version of IQL (Garg et al. 2022), respectively. PEX and PROTO are implemented on the author-provided codes without additional hyperparameter adjustments. For IQL-based methods, we perform 1 million update steps for offline pre-training and then 0.3 million environment steps for online fine-tuning. We compare SUF-TD3-BC and SUF-CQL with APL and SUNE, as they are built upon both TD3-BC and CQL. Since APL and SUNE have not been open-sourced until now, we directly use their final returns reported in (Guo et al. 2023). For SUF-TD3-BC and SUF-CQL, we perform 1 million pre-training steps and 0.1 million fine-tuning steps to ensure consistency with the reported returns in (Guo et al. 2023).

Performance Comparison

We present the aggregated performance for all 12 tasks in Figures 3, and the final return on different backbones in Table 1. Moreover, we present the results at 1 million steps to evaluate the asymptotic performance in Table 2. The return values have been normalized following (Fu et al. 2020), where 0 and 100 represent the performances of random and expert policy, respectively. All learning curves for each task are provided in Supplementary Material.

For IQL-based methods, all baselines improve the initial pre-trained performance to some extent, as shown in Figure 3(a). Among them, retaining the same offline constraints, IQL exhibits inefficient improvement. PEX enhances exploration by initializing a random policy, resulting in higher final performance but lower initial performance than IQL. PROTO gradually relaxes the constraint strength through iterative policy regularization, leading to relatively effective improvement. In contrast, in a completely unconstrained way, SUF significantly outperforms the competitive baselines in efficiency. Simultaneously, SUF eliminates policy collapse during unconstrained fine-tuning, ensuring stable improvement. Figure 3(b) and Table 2 highlight SUF's superiority across various statistical metrics and in asymptotic performance, respectively.

Further, Table 1 demonstrates SUF's adaptability. When combined with different offline RL backbones, SUF consistently outperforms SOTA baselines. Note that APL and SUNE introduce complex techniques to constrained fine-tuning, resulting in increased parameters and complexity. Specifically, APL pre-trains 5 ensembled value networks and then fine-tunes them in the online phase, and SUNE

Tasks	IQL-based (0.3 million)			TD3-BC-based (0.1 million)			CQL-based (0.1 million)		
	PEX	PROTO	SUF	APL	SUNE	SUF	APL	SUNE	SUF
halfcheetah-r	60±7	89±4	**90±2**	70±5	**77±2**	59±7	68±10	69±9	**70±4**
halfcheetah-m	70±4	90±3	**97±2**	**81±2**	**81±3**	**81±4**	45±39	**80±1**	78±1
halfcheetah-mr	54±1	76±3	**85±2**	72±1	70±3	**74±3**	**79±1**	76±2	74±3
hopper-r	25±13	44±25	**105±3**	27±14	39±15	**90±19**	42±22	44±12	**100±3**
hopper-m	96±7	80±35	**109±2**	77±24	102±6	**105±2**	103±3	**104±1**	87±18
hopper-mr	96±12	85±19	**110±2**	101±10	101±7	**106±2**	97±10	102±9	**104±6**
walker2d-r	12±2	23±13	**77±5**	14±4	14±5	**66±24**	6±2	15±6	**41±18**
walker2d-m	89±19	77±33	**123±2**	98±14	**114±2**	111±3	75±26	86±13	**115±14**
walker2d-mr	88±12	107±6	**117±4**	108±4	**109±2**	107±2	103±19	108±4	**113±3**
ant-r	80±23	93±19	**116±19**	-	-	85±2	-	-	58±8
ant-m	105±14	146±3	**156±1**	-	-	137±8	-	-	142±3
ant-mr	105±13	144±4	**151±1**	-	-	135±7	-	-	136±1
Total	880	1054	**1336**	648	707	**799**	618	684	**782**

Table 1: Comparison of the average normalized return of each method based on different offline RL backbones. ± captures the standard deviation over 5 seeds. The highest-performing returns are in bold. r = `random`, m = `medium`, mr = `medium-replay`. Note that APL and SUNE do not report their performance for `ant`, thus we excluded these three tasks from the computation for the total return of SUF-TD3-BC and SUF-CQL.

	IQL-Offline	IQL	PEX	PROTO	SUF
Total	588	808	1032	1217	**1381**

Table 2: Total final return for all 12 tasks at 1 million steps.

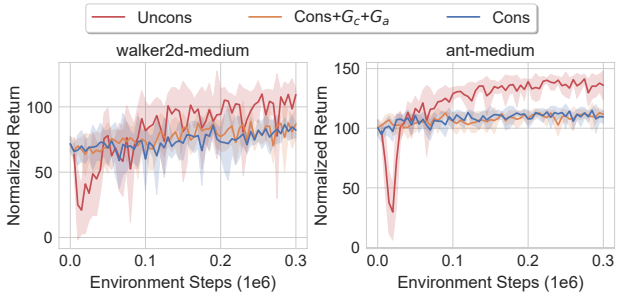

Figure 4: The impact of unconstrained fine-tuning on SUF in `walker2d-medium` (left) and `ant-medium` (right). `Cons` denotes constrained fine-tuning (i.e., vanilla `IQL`).

trains an additional VAE density estimator. In contrast, SUF neither changes the existing pre-training process nor introduces extra training components. Our results indicate the unnecessity of many complex techniques employed in offline-to-online RL.

Ablation Study

In this subsection, we investigate the contributions of each component in SUF: Unconstrained fine-tuning (denoted as `Uncons`), modified Critic UTD (denoted as G_c, set to 20) and Actor UTD (denoted as G_a, set to $1/4$). To this end, we evaluate SUF-IQL and its variants on two typical tasks: `walker2d-medium` and `ant-medium`.

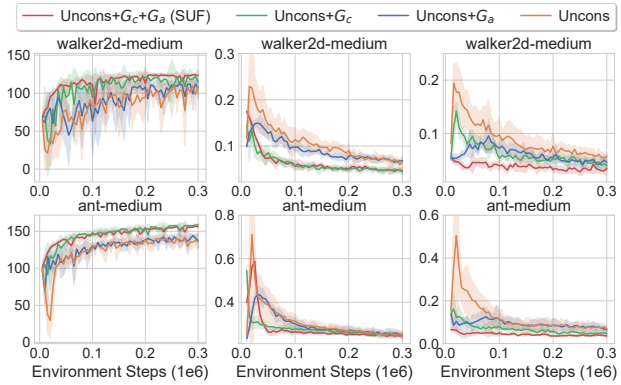

Figure 5: The impact of modifying UTD on SUF. From left to right, we present the learning curves of normalized return, ANB, and SNB on `walker2d-medium` (top) and `ant-medium` (bottom).

Unconstrained Fine-Tuning Figure 4 demonstrates the criticality of unconstrained fine-tuning in SUF. Due to the restricted exploration depicted in Figure 1(b), constrained method exhibits limited efficiency during fine-tuning, even with the same modifications to G_c and G_a as in SUF. Note that SUF is designed for stabilizing unconstrained fine-tuning, rendering the modifications of G_c and G_a ineffective in constrained method. However, despite enabling effective improvement, unconstrained fine-tuning still suffers from severe policy collapse at the initial stage if G_c and G_a are not appropriately modified.

Modifications of UTD To illustrate how SUF mitigates policy collapse while improving efficiency, we present the ANB and SNB in Figure 5, computed in the same way as the results in Figure 1(c-d). Under the premise of unconstrained

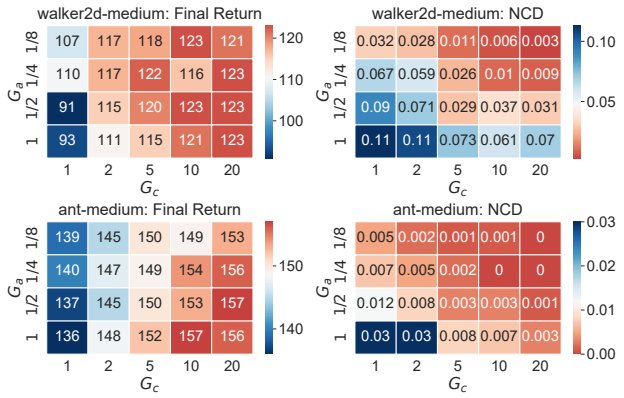

Figure 6: Heat maps depicting the final return (left) and NCD (right) across different values of Critic UTD G_c and Actor UTD G_a on `walker2d-medium` (top) and `ant-medium` (bottom). Lower NCD is better.

fine-tuning, the conventional UTD settings ($G_c = G_a = 1$) severely hinder the fitting of value networks on OOD data, resulting in excessive value bias. Figure 5 demonstrates that both increasing G_c and decreasing G_a eliminate the policy collapse to some extent. Specifically, increasing G_c expedites the fitting of the value network, leading to reduced value bias on OOD data and improved efficiency. Decreasing G_a improves the accuracy of policy updates and significantly reduces the SNB of the value network.

Ultimately, the synergistic integration of unconstrained fine-tuning and UTD modifications enables SUF to almost eliminate the policy collapse completely during unconstrained fine-tuning, ensuring both efficiency and stability.

Hyperparameter Analysis

In this subsection, we analyze the impact of hyperparameters in SUF on the efficiency and stability of fine-tuning. SUF introduces two hyperparameters: Critic UTD G_c and Actor UTD G_a. To quantify the degree of policy collapse, we define the ***N**ormalized **C**umulative Performance **D**rop* (**NCD**) as:

$$\frac{1}{T+1}\sum_{t=0}^{T} I\left(R(0) > R(t)\right)\frac{R(0)-R(t)}{R(0)}, \qquad (4)$$

where $R(\cdot)$ is a function representing the performance of policy across environment steps t. $R(0)$ represents the initial pre-training performance. T is the total fine-tuning steps. $I(\cdot)$ is an indicator function, which takes the value 1 when the condition is satisfied and 0 otherwise.

We visualize the heat maps of the final return and NCD across different values of G_c and G_a, which reflect the efficiency and stability of fine-tuning, respectively. As shown in Figure 6, increasing G_c proves efficient and stable, but too high G_c brings little gain. Decreasing G_a reduces NCD significantly, but too low G_a may hurt efficiency. In practice, we recommend setting G_c and G_a based on specific scenario requirements. For example, selecting a relatively low G_a for stability-focused tasks and avoiding setting G_a too low for efficiency-focused tasks. In this work, we consistently set $G_c = 20$ and $G_c = 1/4$ across diverse backbones, environments, and datasets throughout fine-tuning for simplicity.

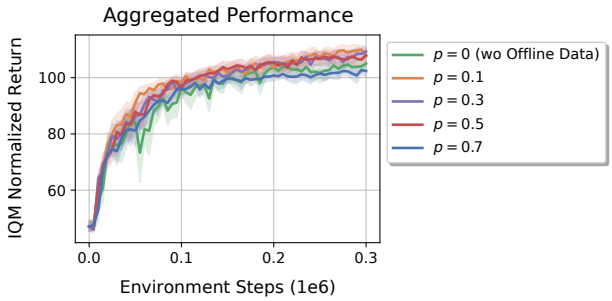

Figure 7: Aggregated performance over all 12 tasks with different sampling ratios of offline data. `p=0` denotes no access to offline datasets.

Sensitivity Analysis

In this subsection, we investigate how the sampling ratio of offline data (denoted as p) affects SUF. During online fine-tuning, data for parameter updating are sampled from the offline dataset $\mathcal{D}$ following a Bernoulli distribution with probability p, and sampled from the online replay buffer $\mathcal{B}$ with probability $1 - p$.

Figure 7 demonstrates the robustness of SUF across a wide range of p, even in the absence of offline data (`p=0`). The success of SUF highlights the importance of accurate value estimates on OOD data generated during the online phase. However, higher p (e.g., 0.7) leads to increased updates on data from offline datasets, resulting in relatively inferior performance. In this work, we set $p = 0.5$ to maintain consistency with the compared baselines (Zhang, Xu, and Yu 2022; Li et al. 2023; Zheng et al. 2023).

Conclusion and Future Work

In this paper, we circumvent the *efficiency-stability dilemma* in offline-to-online RL through a novel perspective: eliminating the policy collapse without imposing constraints. We point to the mismatch between unconstrained fine-tuning and conventional RL training framework as the culprit for such policy collapse. To this end, we propose SUF, a streamlined framework for unconstrained fine-tuning that ensures both efficiency and stability. We conduct comprehensive experiments and ablation studies, demonstrating the remarkable adaptability and superior performance of SUF.

However, for simplicity, we maintain the same UTD configuration for SUF across various environments and datasets, which may not be optimal. In addition, considering that policy collapse typically arises at the initial stage of unconstrained fine-tuning, employing dynamic UTD (e.g., annealing) appears to be more reasonable. In the future, a promising and interesting solution is to explore an adaptive UTD framework for both actor and critic, based on the properties of distinct offline datasets and pre-trained policies.

Acknowledgments

This work was supported in part by National Key R&D Program of China under contract 2022ZD0119802, and in part by National Natural Science Foundation of China under Contract 61836011. It was also supported by the GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC, and the Supercomputing Center of the USTC.

References

Agarwal, R.; Schwarzer, M.; Castro, P. S.; Courville, A. C.; and Bellemare, M. 2021. Deep Reinforcement Learning at the Edge of the Statistical Precipice. *Advances in Neural Information Processing Systems*, 34: 29304–29320.

An, G.; Moon, S.; Kim, J.-H.; and Song, H. O. 2021. Uncertainty-Based Offline Reinforcement Learning with Diversified Q-Ensemble. *Advances in Neural Information Processing Systems*, 34: 7436–7447.

Beeson, A.; and Montana, G. 2022. Improving TD3-BC: Relaxed Policy Constraint for Offline Learning and Stable Online Fine-Tuning. In *Offline Reinforcement Learning Workshop NeurIPS*.

Chen, X.; Wang, C.; Zhou, Z.; and Ross, K. W. 2020. Randomized Ensembled Double Q-Learning: Learning Fast Without a Model. In *International Conference on Learning Representations*.

Dorka, N.; Welschehold, T.; and Burgard, W. 2023. Dynamic Update-to-Data Ratio: Minimizing World Model Overfitting. In *International Conference on Learning Representations*.

D'Oro, P.; Schwarzer, M.; Nikishin, E.; Bacon, P.-L.; Bellemare, M. G.; and Courville, A. 2022. Sample-Efficient Reinforcement Learning by Breaking the Replay Ratio Barrier. In *International Conference on Learning Representations*.

Fu, J.; Kumar, A.; Nachum, O.; Tucker, G.; and Levine, S. 2020. D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv:2004.07219.

Fujimoto, S.; and Gu, S. S. 2021. A Minimalist Approach to Offline Reinforcement Learning. *Advances in Neural Information Processing Systems*, 34: 20132–20145.

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *International Conference on Machine Learning*, 1587–1596. PMLR.

Fujimoto, S.; Meger, D.; and Precup, D. 2019. Off-Policy Deep Reinforcement Learning without Exploration. In *International Conference on Machine Learning*, 2052–2062. PMLR.

Garg, D.; Hejna, J.; Geist, M.; and Ermon, S. 2022. Extreme Q-Learning: MaxEnt RL without Entropy. In *International Conference on Learning Representations*.

Guo, S.; Sun, Y.; Hu, J.; Huang, S.; Chen, H.; Piao, H.; Sun, L.; and Chang, Y. 2023. A Simple Unified Uncertainty-Guided Framework for Offline-to-Online Reinforcement Learning. arXiv:2306.07541.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning*, 1861–1870. PMLR.

Jin, Y.; Yang, Z.; and Wang, Z. 2021. Is Pessimism Provably Efficient for Offline RL? In *International Conference on Machine Learning*, 5084–5096. PMLR.

Kostrikov, I.; Nair, A.; and Levine, S. 2021. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*.

Kumar, A.; Fu, J.; Soh, M.; Tucker, G.; and Levine, S. 2019. Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction. *Advances in Neural Information Processing Systems*, 32.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative Q-Learning for Offline Reinforcement Learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191.

Lee, S.; Seo, Y.; Lee, K.; Abbeel, P.; and Shin, J. 2022. Offline-to-Online Reinforcement Learning via Balanced Replay and Pessimistic Q-Ensemble. In *Conference on Robot Learning*, 1702–1712. PMLR.

Levine, S.; Kumar, A.; Tucker, G.; and Fu, J. 2020. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. arXiv:2005.01643.

Li, J.; Hu, X.; Xu, H.; Liu, J.; Zhan, X.; and Zhang, Y.-Q. 2023. PROTO: Iterative Policy Regularized Offline-to-Online Reinforcement Learning. arXiv:2305.15669.

Li, Q.; Kumar, A.; Kostrikov, I.; and Levine, S. 2022. Efficient Deep Reinforcement Learning Requires Regulating Overfitting. In *International Conference on Learning Representations*.

Luo, Y.; Kay, J.; Grefenstette, E.; and Deisenroth, M. P. 2022. Finetuning from Offline Reinforcement Learning: Challenges, Trade-offs and Practical Solutions. In *Multi-Disciplinary Conference on Reinforcement Learning and Decision Making*.

Mao, Y.; Wang, C.; Wang, B.; and Zhang, C. 2022. MOORe: Model-Based Offline-to-Online Reinforcement Learning. arXiv:2201.10070.

Mark, M. S.; Ghadirzadeh, A.; Chen, X.; and Finn, C. 2022. Fine-Tuning Offline Policies with Optimistic Action Selection. In *Deep Reinforcement Learning Workshop NeurIPS*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-Level Control through Deep Reinforcement Learning. *Nature*, 518(7540): 529–533.

Nair, A.; Gupta, A.; Dalal, M.; and Levine, S. 2020. AWAC: Accelerating Online Reinforcement Learning with Offline Datasets. arXiv:2006.09359.

Nakamoto, M.; Zhai, Y.; Singh, A.; Ma, Y.; Finn, C.; Kumar, A.; and Levine, S. 2023. Cal-QL: Calibrated Offline RL Pre-Training for Efficient Online Fine-Tuning. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*.

Nikishin, E.; Schwarzer, M.; D'Oro, P.; Bacon, P.-L.; and Courville, A. 2022. The Primacy Bias in Deep Reinforcement Learning. In *International Conference on Machine Learning*, 16828–16847. PMLR.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the Game of Go without Human Knowledge. *Nature*, 550(7676): 354–359.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement Learning: An Introduction*. MIT press.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. MuJoCo: A Physics Engine for Model-Based Control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5026–5033. IEEE.

Wu, J.; Wu, H.; Qiu, Z.; Wang, J.; and Long, M. 2022a. Supported Policy Optimization for Offline Reinforcement Learning. *Advances in Neural Information Processing Systems*, 35: 31278–31291.

Wu, Y.; Chen, X.; Wang, C.; Zhang, Y.; and Ross, K. W. 2022b. Aggressive Q-Learning with Ensembles: Achieving Both High Sample Efficiency and High Asymptotic Performance. In *Deep Reinforcement Learning Workshop NeurIPS*.

Xiao, C.; Wang, H.; Pan, Y.; White, A.; and White, M. 2022. The In-Sample Softmax for Offline Reinforcement Learning. In *International Conference on Learning Representations*.

Yu, Z.; and Zhang, X. 2023. Actor-Critic Alignment for Offline-to-Online Reinforcement Learning. In *International Conference on Machine Learning*, 40452–40474. PMLR.

Zhang, H.; Xu, W.; and Yu, H. 2022. Policy Expansion for Bridging Offline-to-Online Reinforcement Learning. In *International Conference on Learning Representations*.

Zhao, Y.; Boney, R.; Ilin, A.; Kannala, J.; and Pajarinen, J. 2022. Adaptive Behavior Cloning Regularization for Stable Offline-to-Online Reinforcement Learning. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

Zheng, H.; Luo, X.; Wei, P.; Song, X.; Li, D.; and Jiang, J. 2023. Adaptive Policy Learning for Offline-to-Online Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*.