

Appendix

Table of Contents

A Related Work	21
B Definition of Distribution Drift and Catastrophic Forgetting in Deep Learning	22
C Theoretical Results	22
C.1 Proof of Proposition 1	22
C.2 Proof of Proposition 2	24
C.3 Proof of Proposition 3	24
D Policy-based FAME Algorithm	27
E Experiments: Value-based Continual RL with Discrete Action Space	28
E.1 Details of Experimental Setup and Comparison Methods	28
E.2 More Experimental Results	29
E.3 Ablation Study	29
F Experiments: Policy-based Continual RL with Continuous Action Space	31
F.1 Details of Experimental Setup and Comparison Methods	31
F.2 More Experimental Results	31

A Related Work

Continual RL. Continual RL [30, 1] addresses the challenges in learning a sequence of decision-making tasks, particularly balancing the stability (i.e., mitigating catastrophic forgetting) and plasticity (i.e., rapid adaptation to a new environment). (1) *Catastrophic Forgetting*. The relay-based approach is commonly applied to mitigate forgetting. A replay-based recurrent methodology was initially proposed for task-agnostic agents [9]. RECALL [56] leverages adaptive normalization on approximate targets and policy distillation on old tasks to enhance generality and stability. Generative replay [12] was recently proposed using the diffusion model to memorize the high-return trajectory distribution of each encountered task. Moreover, biology-inspired methods include synaptic consolidation [27, 28] and sparse prompting [52]. Model expansion also serves as a promising direction to investigate [32, 22]. [22] builds the subspace of policies to consider the scalable continual RL, while pointing out the trade-off between the agent’s size and the performance of continual learning. [32] uses a growing policy neural network and applies the attention mechanism to integrate the knowledge from the previous policies and the current state to “self-compose” an internal policy. Despite the effectiveness of model expansion-based approaches, the primary concerns lie in their high memory and inference costs due to the leverage of previous policies with specific aggregated strategies. For instance, the attention module is adopted in [32], where the number of parameters grows *linearly* and the theoretical computational cost is *quadratic* with respect to the number of tasks. In contrast, our FAME approach only employs fast and meta learners with fixed model sizes and performs an incremental update for knowledge retention, avoiding the stability issue.

(2) *Knowledge Transfer and Loss of Plasticity*. The effectiveness of knowledge forward transfer determines the loss of plasticity, a reduced capability to rapidly adapt to a new environment [2, 16, 49, 45]. [51] improves the forward transfer and mitigates the loss of plasticity by selectively identifying the most relevant samples for the new task using learnable importance weights. The value function decomposing approach [5] is proposed to perform an interplay between fast and slow learning at various levels for value-based continual RL with a discrete action space to address the loss of plasticity. Parseval network [13] imposes orthogonality constraints to mitigate interference, while [34] employs parameter-free online convex optimization to retain plasticity. Recently, *negative*

transfer issue [4] was revealed in the adaptation to a new task due to the task dissimilarity, amplifying the loss of plasticity. As such, Reset and Distill (R&D) [4] explicitly resets the policy in each new environment to avoid the negative transfer. By contrast, it is not necessary that reset is always the best choice, especially when the task similarity often exists. Our proposed adaptive meta warm-up can selectively discriminate the most effective weight initialization and warm-up strategy.

Transfer, Multi-task, and Meta RL. The knowledge transfer phase in our approach is closely linked with transfer RL, where the accrued knowledge can be transferred through representation [39, 14], learned models [21, 18], and network weights [19] or experience [6, 47, 51]. Although the set of tasks should be learned simultaneously, multi-task RL also integrates the component of knowledge transfer, such as [10, 43, 46, 52, 38, 25]. As such, transfer RL has been an underpinning building block for both multi-task and continual RL, e.g., the explicit knowledge transfer in our dual-learning algorithm. As opposed to multi-task RL, the incremental or sequential learning nature of continual RL makes it more challenging to minimize catastrophic forgetting. However, our study shows that minimizing catastrophic forgetting can be, in principle, equivalent to the objectives in multi-task learning. Meta RL [20] can be seen as an extension or generalization of multi-task RL, with explicit mechanisms for fast adaptation and few-shot learning, and is also closely linked with continual RL. Continual Meta-Policy Search (CoMPS) [8] probes the setting of meta-training in an incremental fashion, extending meta-RL to a continual learning scenario. Fast and meta learners intermingle the key techniques of transfer, multi-tasking, and meta RL, illuminating their deep connections within our algorithmic framework. The interplay of knowledge transfer and knowledge integration via the coupled updates between fast and meta learners simultaneously tackles the involved challenges, exhibiting promising solutions to address continual RL.

B Definition of Distribution Drift and Catastrophic Forgetting in Deep Learning

We first introduce the concept of *drift* in the process of learning a parameterized function f from the source data distribution τ_S with the dataset \mathcal{D}_{τ_S} to the target data distribution τ_T with the dataset \mathcal{D}_{τ_T} . After learning f on the source dataset \mathcal{D}_{τ_S} , we obtain the estimated function \hat{f}_{τ_S} . Then we apply the same model architecture f on the target dataset \mathcal{D}_{τ_T} with any learning algorithms, and finally we evaluate the drift of the attained \hat{f}_{τ_T} via $\delta^{\tau_S \rightarrow \tau_T}$ defined as [15]:

$$\delta^{\tau_S \rightarrow \tau_T}(X^{\tau_S}) = \left(\hat{f}_{\tau_T}(x) - \hat{f}_{\tau_S}(x) \right)_{(x,y) \in \mathcal{D}_{\tau_S}} \quad (11)$$

Based on the definition of *drift*, we define the *vanilla catastrophic forgetting* $\Delta^{\tau_S \rightarrow \tau_T}$ as

$$\Delta^{\tau_S \rightarrow \tau_T}(X^{\tau_S}) = \|\delta^{\tau_S \rightarrow \tau_T}(X^{\tau_S})\|_2^2 = \sum_{(x,y) \in \mathcal{D}_{\tau_S}} \left(\hat{f}_{\tau_T}(x) - \hat{f}_{\tau_S}(x) \right)^2, \quad (12)$$

where the catastrophic forgetting can be further simplified as $\Delta^{\tau_S \rightarrow \tau_T} = \|\phi(X^{\tau_S})(\omega_{\tau_T}^* - \omega_{\tau_S}^*)\|_2^2$ in the Neural Tangent Kernel (NTK) regime [15, 26], allowing the proposal of new continual learning approaches. In deep learning, minimizing the catastrophic forgetting $\Delta^{\tau_S \rightarrow \tau_T}$ is equivalent to minimizing a weighted drift in terms of the prediction function \hat{f} with the weights determined by the dataset.

C Theoretical Results

C.1 Proof of Proposition 1

Proposition 1 [Incremental Q-Value-based Meta Learner Update] Consider d_Q to be ℓ_2 loss in Eq. (1) in Definition 2. Minimizing Q-value-based catastrophic forgetting in Eq. 3 is equivalent to:

$$Q_k^M = \arg \min_{\tilde{Q}_k^M} \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\left(Q_{k-1}^M - \tilde{Q}_k^M \right)^2 \right] + \mathbb{E}_{w_k^Q} \left[\left(Q_k - \tilde{Q}_k^M \right)^2 \right].$$

984 *Proof. Step 1: Optimality Condition.* Recap $w_i^Q(s, a) = \mu_i^{Q_i}(s)\pi_i^Q(a|s)$. We aim to minimize the
 985 Q-value-based catastrophic forgetting in Eq. (3):

$$Q_k^M = \arg \min_{\tilde{Q}_k^M} \sum_{i=1}^k \sum_{s,a} w_i^Q(s, a) \left(Q_i(s, a) - \tilde{Q}_k^M(s, a) \right)^2,$$

986 For each s and a , by taking the derivative to the objective in Eq. (3) regarding \tilde{Q}_k^M , the first-order
 987 optimality condition is

$$\sum_{i=1}^k w_i^Q(s, a) (Q_k^M(s, a) - Q_i(s, a)) = 0 \quad (13)$$

988 This leads to the two optimality conditions regarding Q_k^M and Q_{k-1}^M

$$Q_k^M(s, a) = \frac{\sum_{i=1}^k w_i^Q(s, a) Q_i(s, a)}{\sum_{j=1}^k w_j^Q(s, a)}, \quad Q_{k-1}^M(s, a) = \frac{\sum_{i=1}^{k-1} w_i^Q(s, a) Q_i(s, a)}{\sum_{j=1}^{k-1} w_j^Q(s, a)}.$$

989 **Step 2: Incremental Update Rule.** For brevity, we employ the expectation operation \mathbb{E}_w . Based on
 990 the two optimality conditions above, we can derive the following incremental update rule:

$$\begin{aligned} Q_k^M &= \arg \min_{\tilde{Q}_k^M} \sum_{i=1}^k \mathbb{E}_{w_i^Q} \left[\left(Q_i - \tilde{Q}_k^M \right)^2 \right] \\ &= \arg \min_{\tilde{Q}_k^M} \underbrace{\sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\left(Q_i - Q_{k-1}^M + Q_{k-1}^M - \tilde{Q}_k^M \right)^2 \right]}_{\textcircled{1}} + \mathbb{E}_{w_k^Q} \left[\left(Q_k - \tilde{Q}_k^M \right)^2 \right]. \end{aligned} \quad (14)$$

991

$$\begin{aligned} \textcircled{1} &= \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\left(Q_i - Q_{k-1}^M \right)^2 + \left(Q_{k-1}^M - \tilde{Q}_k^M \right)^2 + 2 \left(Q_i - Q_{k-1}^M \right) \left(Q_{k-1}^M - \tilde{Q}_k^M \right) \right] \\ &= \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\left(Q_i - Q_{k-1}^M \right)^2 + \left(Q_{k-1}^M - \tilde{Q}_k^M \right)^2 \right] + \\ &\quad 2 \sum_{s,a} \left(Q_{k-1}^M(s, a) - \tilde{Q}_k^M(s, a) \right) \left(\sum_{i=1}^{k-1} w_i^Q(s, a) \left(Q_i(s, a) - Q_{k-1}^M(s, a) \right) \right) \\ &\stackrel{(a)}{=} \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\left(Q_i - Q_{k-1}^M \right)^2 + \left(Q_{k-1}^M - \tilde{Q}_k^M \right)^2 \right] \\ &= C + \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\left(Q_{k-1}^M - \tilde{Q}_k^M \right)^2 \right], \end{aligned}$$

992 where (a) holds as $\sum_{i=1}^{k-1} w_i^Q(s, a) (Q_i(s, a) - Q_{k-1}^M(s, a)) = 0$ is the optimality condition for
 993 Q_{k-1}^M in Eq. (13) of Step 1. $C = \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\left(Q_i - Q_{k-1}^M \right)^2 \right]$ is a constant in terms of \tilde{Q}_k^M . Putting
 994 all together into Eq. 14, we have

$$Q_k^M = \arg \min_{\tilde{Q}_k^M} \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\left(Q_{k-1}^M - \tilde{Q}_k^M \right)^2 \right] + \mathbb{E}_{w_k^Q} \left[\left(Q_k - \tilde{Q}_k^M \right)^2 \right].$$

995

□

996 C.2 Proof of Proposition 2

997 **Proposition 2** [Incremental Softmax Q-Value-based Meta Learner Update] Denote $\pi_k^M(a|s) =$
 998 $\exp(Q_k^M(a|s)/\tau) / \sum_{a'} \exp(Q_k^M(a'|s)/\tau)$. After a softmax policy transformation, the Q-value-
 999 based meta learner incremental update is rewritten as

$$Q_k^M = \arg \min_{\tilde{Q}_k^M} \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\log \frac{\pi_{k-1}^M}{\tilde{\pi}_k^M} \right] + \mathbb{E}_{w_k^Q} \left[\log \frac{\pi_k^Q}{\tilde{\pi}_k^M} \right] = \arg \max_{\tilde{Q}_k^M} \sum_{i=1}^k \mathbb{E}_{w_i^Q} [\log \tilde{\pi}_k^M],$$

1000 *Proof.* We rely on the softmax transformation to transfer a meta Q function to a meta policy. As such,
 1001 the policy-based catastrophic forgetting in Eq. (5), when adapted from value-based continual RL and
 1002 equipped with KL divergence as d_π , can be expressed as

$$Q_k^M = \arg \min_{\tilde{Q}_k^M} \sum_{i=1}^k \sum_s \mu_i^{Q_i}(s) \left[\text{KL} \left(\pi_i^Q(\cdot|s) || \tilde{\pi}_k^M(\cdot|s) \right) \right]. \quad (15)$$

1003 where $\tilde{\pi}_k^M(a|s) = \exp(\tilde{Q}_k^M(a|s)/\tau) / \sum_{a'} \exp(\tilde{Q}_k^M(a'|s)/\tau)$. By the definition of the KL diver-
 1004 gence, we can rewrite the objective function in Eq. 15 as an incremental update rule:

$$\begin{aligned} Q_k^M &= \arg \min_{\tilde{Q}_k^M} \sum_{i=1}^k \sum_{s,a} \mu_i^{Q_i}(s) \pi_i^Q(a|s) \log \frac{\pi_i^Q(a|s)}{\tilde{\pi}_k^M(a|s)} \\ &= \arg \min_{\tilde{Q}_k^M} \sum_{i=1}^k \mathbb{E}_{w_i^Q} \left[\log \frac{\pi_i^Q(a|s)}{\tilde{\pi}_k^M(a|s)} \right] \\ &= \arg \min_{\tilde{Q}_k^M} \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\log \left(\frac{\pi_i^Q(a|s)}{\tilde{\pi}_k^M(a|s)} \frac{\pi_{k-1}^M(a|s)}{\pi_{k-1}^M(a|s)} \right) \right] + \mathbb{E}_{w_k^Q} \left[\log \frac{\pi_k^Q}{\tilde{\pi}_k^M} \right] \\ &= \arg \min_{\tilde{Q}_k^M} \left\{ \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\log \frac{\pi_{k-1}^M}{\tilde{\pi}_k^M} \right] + \mathbb{E}_{w_k^Q} \left[\log \frac{\pi_k^Q}{\tilde{\pi}_k^M} \right] \right\} + C \\ &= \arg \min_{\tilde{Q}_k^M} \left\{ \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\log \frac{\pi_{k-1}^M}{\tilde{\pi}_k^M} \right] + \mathbb{E}_{w_k^Q} \left[\log \frac{\pi_k^Q}{\tilde{\pi}_k^M} \right] \right\} \quad (16) \\ &= \arg \min_{\tilde{Q}_k^M} \left\{ \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\log \frac{1}{\tilde{\pi}_k^M} \right] + \mathbb{E}_{w_k^Q} \left[\log \frac{1}{\tilde{\pi}_k^M} \right] \right\} \\ &= \arg \max_{\tilde{Q}_k^M} \sum_{i=1}^k \mathbb{E}_{w_i^Q} [\log \tilde{\pi}_k^M], \quad (17) \end{aligned}$$

1005 where $C = \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\log \frac{\pi_i^Q}{\pi_{k-1}^M} \right]$ is a constant and is independent of \tilde{Q}_k^M . Although it may be
 1006 trivial to keep the form of Eq. (16), it emphasizes an incremental update rule of Q_k^M based on Q_{k-1}^M
 1007 (π_{k-1}^M) and Q_k (π_k^Q). Eventually, this minimization leads to an Maximum Likelihood estimation
 1008 regarding the meta learner Q_k^M in Eq. (17), on a mixture of state-action distribution of all encountered
 1009 environments up to k .

1010 □

1011 C.3 Proof of Proposition 3

1012 **Proposition 3** [Incremental Policy-based Meta Learner Update under Wasserstein Distance] Consider
 1013 d_π to be the squared 2-Wasserstein distance in Eq. (2) of Definition 2 and the policy is represented
 1014 as an independent (multivariate) Gaussian distribution over the action a . Minimizing policy-based

1015 catastrophic forgetting in Eq. (5) is equivalent to:

$$\pi_M^k = \arg \min_{\tilde{\pi}_k^M} \left\{ \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) W_2^2(\tilde{\pi}_k^M(\cdot|s), \pi_{k-1}^M(\cdot|s)) + \sum_s \mu_k^{\pi_k}(s) W_2^2(\tilde{\pi}_k^M(\cdot|s), \pi_k(\cdot|s)) \right\}.$$

1016 *Proof.* Recap the objective of the policy-based catastrophic forgetting based on Eq. (5) under squared
1017 2-Wasserstein distance:

$$\pi_k^M = \arg \min_{\tilde{\pi}_k^M} \sum_{i=1}^k \sum_s \mu_i^{\pi_i}(s) W_2^2(\pi_i(\cdot|s), \tilde{\pi}_k^M(\cdot|s)).$$

1018 where the squared 2-Wasserstein distance between two Gaussian distributions has close-form solution:

1019 $W_2^2(p, q) = \|\nu_p - \nu_q\|_2^2 + \text{tr} \left(\Sigma_p + \Sigma_q - 2 \left(\Sigma_p^{1/2} \Sigma_q \Sigma_p^{1/2} \right)^{1/2} \right)$ with the two Gaussian distributions
1020 denoted by $\mathcal{N}(\nu_p, \Sigma_p)$ and $\mathcal{N}(\nu_q, \Sigma_q)$. In particular, when the policy is represented as an independent
1021 (multivariate) Gaussian distribution across the action a , it implies that Σ_p and Σ_q are diagonal (i.e.,
1022 variables are independent), then the squared 2-Wasserstein distance can be further simplified as
1023 $W_2^2(p, q) = \|\nu_p - \nu_q\|_2^2 + \|\sigma_p - \sigma_q\|_2^2$, where σ_p and σ_q are the diagonal vector of Σ_p and Σ_q ,
1024 respectively. Then, the objective of the policy-based catastrophic forgetting based on Eq. (5) can be
1025 simplified as

$$\pi_k^M = \arg \min_{\tilde{\nu}_k^M, \tilde{\sigma}_k^M} \sum_{i=1}^k \sum_s \mu_i^{\pi_i}(s) (\|\nu_i(s) - \tilde{\nu}_k^M(s)\|_2^2 + \|\sigma_i(s) - \tilde{\sigma}_k^M(s)\|_2^2), \quad (18)$$

1026 where $\pi_i(\cdot|s)$ is represented as a (multivariate) Gaussian distribution $\mathcal{N}(\nu_i(s), \sigma_i(s))$ and π_M^k is
1027 represented as a (multivariate) Gaussian distribution $\mathcal{N}(\nu_k^M(s), \sigma_k^M(s))$.

1028 **Step 1: Optimality Condition.** For each s , we take the derivative of Eq. (18) in terms of $\tilde{\nu}_k^M$ and
1029 $\tilde{\sigma}_k^M$, respectively. Consequently, it arrives at the following optimality condition:

$$\sum_{i=1}^k \mu_i^{\pi_i}(s) (\nu_i(s) - \nu_k^M(s)) = 0 \quad (19)$$

$$\sum_{i=1}^k \mu_i^{\pi_i}(s) (\sigma_i(s) - \sigma_k^M(s)) = 0. \quad (20)$$

1030 **Step 2: Incremental Update.** We first rewrite Eq. (18) as

$$\begin{aligned} \pi_k^M = \arg \min_{\tilde{\nu}_k^M, \tilde{\sigma}_k^M} & \underbrace{\sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) \|\nu_i(s) - \tilde{\nu}_k^M(s)\|_2^2}_{\textcircled{1}} + \sum_s \mu_k^{\pi_k}(s) \|\nu_k(s) - \tilde{\nu}_k^M(s)\|_2^2 \\ & + \underbrace{\sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) \|\sigma_i(s) - \tilde{\sigma}_k^M(s)\|_2^2}_{\textcircled{2}} + \sum_s \mu_k^{\pi_k}(s) \|\sigma_k(s) - \tilde{\sigma}_k^M(s)\|_2^2. \end{aligned}$$

$$\begin{aligned}
\textcircled{1} &= \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) \|\nu_i(s) - \nu_{k-1}^M(s) + \nu_{k-1}^M(s) - \tilde{\nu}_k^M(s)\|_2^2 \\
&= \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) (\|\nu_i(s) - \nu_{k-1}^M(s)\|_2^2 + \|\nu_{k-1}^M(s) - \tilde{\nu}_k^M(s)\|_2^2) + \\
&\quad 2 \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) \langle \nu_i(s) - \nu_{k-1}^M(s), \nu_{k-1}^M(s) - \tilde{\nu}_k^M(s) \rangle \\
&= \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) (\|\nu_i(s) - \nu_{k-1}^M(s)\|_2^2 + \|\nu_{k-1}^M(s) - \tilde{\nu}_k^M(s)\|_2^2) + \\
&\quad 2 \sum_s \langle \sum_{i=1}^{k-1} \mu_i^{\pi_i}(s) (\nu_i(s) - \nu_{k-1}^M(s)), \nu_{k-1}^M(s) - \tilde{\nu}_k^M(s) \rangle \\
&\stackrel{(a)}{=} \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) (\|\nu_i(s) - \nu_{k-1}^M(s)\|_2^2 + \|\nu_{k-1}^M(s) - \tilde{\nu}_k^M(s)\|_2^2),
\end{aligned}$$

1032 where (a) holds due to the optimality condition $\sum_{i=1}^{k-1} \mu_i^{\pi_i}(s) (\nu_i(s) - \nu_{k-1}^M(s)) = 0$ we derived in
1033 Eq. (19) of Step 1. Similarly, we can show this simplification regarding the variance:

$$\begin{aligned}
\textcircled{2} &= \sum_{i=1}^{k-1} \sum_s \sigma_i^{\pi_i}(s) \|\sigma_i(s) - \sigma_{k-1}^M(s) + \sigma_{k-1}^M(s) - \tilde{\sigma}_k^M(s)\|_2^2 \\
&= \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) (\|\sigma_i(s) - \sigma_{k-1}^M(s)\|_2^2 + \|\sigma_{k-1}^M(s) - \tilde{\sigma}_k^M(s)\|_2^2) + \\
&\quad 2 \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) \langle \sigma_i(s) - \sigma_{k-1}^M(s), \sigma_{k-1}^M(s) - \tilde{\sigma}_k^M(s) \rangle \\
&= \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) (\|\sigma_i(s) - \sigma_{k-1}^M(s)\|_2^2 + \|\sigma_{k-1}^M(s) - \tilde{\sigma}_k^M(s)\|_2^2) + \\
&\quad 2 \sum_s \langle \sum_{i=1}^{k-1} \mu_i^{\pi_i}(s) (\sigma_i(s) - \sigma_{k-1}^M(s)), \sigma_{k-1}^M(s) - \tilde{\sigma}_k^M(s) \rangle \\
&\stackrel{(b)}{=} \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) (\|\sigma_i(s) - \sigma_{k-1}^M(s)\|_2^2 + \|\sigma_{k-1}^M(s) - \tilde{\sigma}_k^M(s)\|_2^2),
\end{aligned}$$

1034 where (b) holds due to the optimality condition $\sum_{i=1}^{k-1} \mu_i^{\pi_i}(s) (\sigma_i(s) - \sigma_{k-1}^M(s)) = 0$ we derived in
1035 Eq. (20) of Step 1. Putting all together, we have

$$\begin{aligned}
\pi_k^M &= \arg \min_{\tilde{\nu}_k^M, \tilde{\sigma}_k^M} \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) (\|\nu_i(s) - \nu_{k-1}^M(s)\|_2^2 + \|\nu_{k-1}^M(s) - \tilde{\nu}_k^M(s)\|_2^2) + \sum_s \mu_k^{\pi_k}(s) \|\nu_k(s) - \tilde{\nu}_k^M(s)\|_2^2 \\
&\quad + \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) (\|\sigma_i(s) - \sigma_{k-1}^M(s)\|_2^2 + \|\sigma_{k-1}^M(s) - \tilde{\sigma}_k^M(s)\|_2^2) + \sum_s \mu_k^{\pi_k}(s) \|\sigma_k(s) - \tilde{\sigma}_k^M(s)\|_2^2
\end{aligned}$$

1036 By removing the constant terms independent of $\tilde{\nu}_k^M$ and $\tilde{\sigma}_k^M$, we further have

$$\begin{aligned}
&= \arg \min_{\tilde{\nu}_k^M, \tilde{\sigma}_k^M} \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) \|\nu_{k-1}^M(s) - \tilde{\nu}_k^M(s)\|_2^2 + \sum_s \mu_k^{\pi_k}(s) \|\nu_k(s) - \tilde{\nu}_k^M(s)\|_2^2 \\
&+ \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) \|\sigma_{k-1}^M(s) - \tilde{\sigma}_k^M(s)\|_2^2 + \sum_s \mu_k^{\pi_k}(s) \|\sigma_k(s) - \tilde{\sigma}_k^M(s)\|_2^2 \\
&= \arg \min_{\tilde{\pi}_k^M} \left\{ \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) W_2^2(\tilde{\pi}_k^M(\cdot|s), \pi_{k-1}^M(\cdot|s)) + \sum_s \mu_k^{\pi_k}(s) W_2^2(\tilde{\pi}_k^M(\cdot|s), \pi_k(\cdot|s)) \right\}
\end{aligned}$$

1037

□

1038 D Policy-based FAME Algorithm

1039 We first denote the fast buffer as \mathcal{F} and π^0 as the initialized policy. As suggested in Algorithm 1,
1040 when the k -th environment arrives, we initialize the fast learner π_k via the adaptive meta warm-up
1041 among the preceding meta learner π_{k-1}^M , the preceding fast learner π_{k-1} and a random learner π^0
1042 (reset strategy) within L steps. The adaptive meta warm-up makes full use of previous information to
1043 perform an effective knowledge transfer. Once the k -th task ends, the knowledge integration phase
1044 starts, when the meta learner π_k^M is updated via Eq. (8) (FAME-KL) or via Eq. 9 (FAME-MD) on the
1045 data collected in the meta buffer \mathcal{M} . The meta learner incrementally incorporates the knowledge
1046 from π_k into π_{k-1}^M , leading to an updated meta learner π_k^M .

Algorithm 2 Policy-based FAME Update in the k -th Environment

- 1: **Initialize:** Fast Buffer \mathcal{F} , Meta Buffer \mathcal{M} , π_{k-1}^M , π_{k-1} , π^0 , Warm-Up Step L , Estimation Step N .
 - 2: # Knowledge Transfer: Adaptive Meta Warm-Up
 - 3: Initialize π_k in $\{\pi_{k-1}, \pi_k^M, \pi^0\}$ via Eq. (7) within L steps
 - 4: **for** $t = L$ to T **do**
 - 5: Observe S_t , take action A_t , receive R_t , observe S_{t+1}
 - 6: Store (S_t, A_t, R_t, S_{t+1}) in \mathcal{F}
 - 7: Update π_k
 - 8: **if** $t > T - N$ **then**
 - 9: Method 1 (FAME-KL): Store (S_t, A_t) in \mathcal{M} # To Estimate w_k
 - 10: Method 2 (FAME-WD): Store S_t in \mathcal{M} # To Estimate $\mu_k^{\pi_k}$
 - 11: **end if**
 - 12: **end for**
 - 13: Reset \mathcal{F}
 - 14: # Knowledge Integration: Minimize Catastrophic Forgetting
 - 15: Method 1 (FAME-KL): Update π_k^M via Eq. (8) on state-action pairs in \mathcal{M}
 - 16: Method 2 (FAME-WD): Update π_k^M via Eq. (9) on states in \mathcal{M}
-

1047 E Experiments: Value-based Continual RL with Discrete Action Space

1048 E.1 Details of Experimental Setup and Comparison Methods

1049 **Metric Calculation.** As average performance is the main metric in continual RL, we report the
 1050 results for each environment. For the evaluation of forgetting, we first calculate the metric scores for
 1051 each environment and then normalize them by their standard deviation across all methods in each
 1052 environment. This standard normalization mitigates the influence of different reward scales of each
 1053 game and allows us to average them across games to report a more comprehensive forgetting score.

1054 **Hyperparameters.** For our FAME approach, after doing the line search of the hyperparameter in
 1055 Section E.3, we choose the estimation step $N = 12000$ (i.e., the number of data to be stored in the
 1056 meta buffer \mathcal{M} in each task), the warm-up step $L = 50000$ (1% of the training steps in each task).
 1057 In the knowledge integration phase, we train the meta learner across 200 epochs from a 1×10^{-3}
 1058 learning rate with a decaying strategy. The learning rate for the fast learner is kept as 1×10^{-5} , the
 1059 same as the other variants of DQN baselines. Every time a new environment arrives, we clear the
 1060 fast buffer \mathcal{F} and reinitialize the parameters of all involved learners, except for DQN-Finetune. For
 1061 FAME, after the adaptive meta warm-up, we can automatically choose between a random initialization

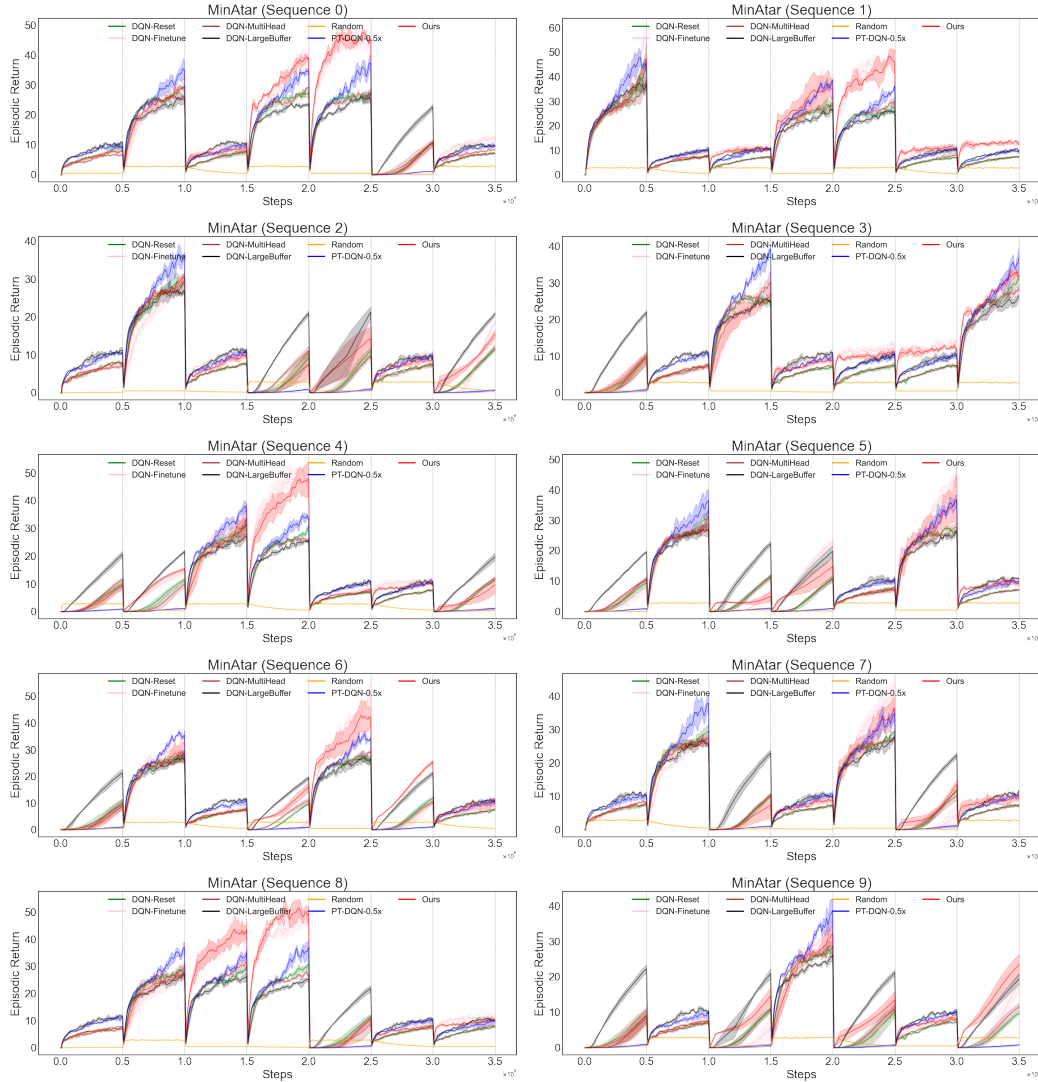


Figure 4: Learning curves of the fast learner in FAME on MinAtar Environments across 10 sequences of tasks.

and an initialization from the preceding fast learner with or without an additional behavior cloning regularization term for demonstration.

Sequences of Tasks. We randomly select 10 sequences of environments and then fix them for reproductivity. We run 3 seeds for each sequence of tasks.

1. ['breakout', 'spaceinvaders', 'breakout', 'spaceinvaders', 'spaceinvaders', 'freeway', 'breakout']
2. ['spaceinvaders', 'breakout', 'breakout', 'spaceinvaders', 'spaceinvaders', 'breakout', 'breakout']
3. ['breakout', 'spaceinvaders', 'breakout', 'freeway', 'freeway', 'breakout', 'freeway']
4. ['freeway', 'breakout', 'spaceinvaders', 'breakout', 'breakout', 'breakout', 'spaceinvaders']
5. ['freeway', 'freeway', 'spaceinvaders', 'spaceinvaders', 'breakout', 'breakout', 'freeway']
6. ['freeway', 'spaceinvaders', 'freeway', 'freeway', 'breakout', 'spaceinvaders', 'breakout']
7. ['freeway', 'spaceinvaders', 'breakout', 'freeway', 'spaceinvaders', 'freeway', 'breakout']
8. ['breakout', 'spaceinvaders', 'freeway', 'breakout', 'spaceinvaders', 'freeway', 'breakout']
9. ['breakout', 'spaceinvaders', 'spaceinvaders', 'spaceinvaders', 'freeway', 'breakout', 'breakout']
10. ['freeway', 'breakout', 'freeway', 'spaceinvaders', 'freeway', 'breakout', 'freeway']

E.2 More Experimental Results

Learning Curves of the Fast Learner. We provide the learning curves of all considered continual RL algorithms in the MinAtar environment across 10 sequences of tasks in Figure 4, demonstrating the favorable adaptation capability of the fast learner guided by the adaptive meta warm-up in each new environment.

E.3 Ablation Study

Regularization Hyperparameter λ in Behavior Cloning. Table 3 suggests that an overly large or small λ results in inferior performance in FT and Forgetting, although the average performance is still favorable. For example, the metric scores in FT and Forgetting are worst for FAME ($\lambda = 0.1$). In practice, we could choose $\lambda = 1.0$ to achieve the highest score in FT, or $\lambda = 5.0$ for the best forgetting score.

Table 3: Ablation Study of **Regularization Hyperparameter λ** on MinAtar on Average Performance (Avg. Perf), Forward Transfer (FT), and Forgetting. Results (Mean \pm SE) are averaged over 10 sequences, each with 3 seeds.

Method	Breakout	Ave. Perf \uparrow Spaceinvader	Freeway	FT \uparrow	Forgetting \downarrow
FAME ($\lambda=0.1$)	12.79 \pm 0.42	19.52 \pm 0.50	1.71 \pm 0.17	0.13 \pm 0.03	0.77 \pm 0.08
FAME ($\lambda=1.0$)	14.54 \pm 0.58	18.72 \pm 0.52	1.69 \pm 0.17	0.16 \pm 0.03	0.72 \pm 0.13
FAME ($\lambda=5.0$)	13.90 \pm 0.55	19.38 \pm 0.62	1.62 \pm 0.16	0.14 \pm 0.03	0.64 \pm 0.07
FAME ($\lambda=10.0$)	13.88 \pm 0.55	19.52 \pm 0.57	1.63 \pm 0.16	0.14 \pm 0.03	0.67 \pm 0.08

Warm-Up Step L . Table 4 shows the comprehensive metric scores of FAME in terms of the number of warm-up steps with the BC regularization. It is interesting to highlight that increasing the number of warm-up steps boosts the FT and average performance on certain games, such as Spaceinvade

1100 and Freeway. However, it worsens the general forgetting. To maintain the forgetting capability, it
 1101 is recommended to keep the warm-up until a specific phase of the fast learning, such as 5×10^4 or
 1102 20×10^4 training steps.

Table 4: Ablation Study of **Warm-Up Step** L on MinAtar on Average Performance (Avg. *Perf*), Forward Transfer (*FT*), and Forgetting. Results (Mean \pm SE) are averaged over 10 sequences, each with 3 seeds.

Method	Breakout	Ave. Perf \uparrow Spaceinvader	Freeway	FT \uparrow	Forgetting \downarrow
FAME ($L = 1 \times 10^4$)	13.34 ± 0.52	19.17 ± 0.68	1.64 ± 0.16	0.13 ± 0.03	0.74 ± 0.08
FAME ($L = 5 \times 10^4$)	14.54 ± 0.58	18.72 ± 0.52	1.69 ± 0.17	0.16 ± 0.03	0.72 ± 0.13
FAME ($L = 20 \times 10^4$)	13.28 ± 0.50	19.87 ± 0.66	1.65 ± 0.16	0.17 ± 0.03	0.72 ± 0.08
FAME ($L = 50 \times 10^4$)	11.83 ± 0.71	20.00 ± 0.96	1.87 ± 0.20	0.17 ± 0.03	0.78 ± 0.09

1103 **Weight Estimation Step** N . For a fixed size of the meta learner buffer \mathcal{M} , we vary the weight
 1104 estimation step N , which determines the collected data in a new environment used for knowledge
 1105 integration of the meta learner. Table 5 showcases that decreasing the weight estimation step
 1106 consistently worsens the performance across all metric scores. It is worthwhile to increase N in the
 1107 future to further enhance our performance, but this would require a larger buffer size of \mathcal{M} than the
 1108 one employed in other baselines. We leave the investigation of the performance of FAME with a
 1109 larger buffer size of \mathcal{M} as future work.

Table 5: Ablation Study of **Weight Estimation Step** N on MinAtar on Average Performance (Avg. *Perf*), Forward Transfer (*FT*), and Forgetting. Results (Mean \pm SE) are averaged over 10 sequences, each with 3 seeds.

Method	Breakout	Ave. Perf \uparrow Spaceinvader	Freeway	FT \uparrow	Forgetting \downarrow
FAME ($N=4000$)	11.95 ± 0.47	14.69 ± 0.55	1.54 ± 0.17	0.14 ± 0.03	0.93 ± 0.08
FAME ($N=8000$)	12.49 ± 0.37	17.99 ± 0.58	1.67 ± 0.17	0.15 ± 0.03	0.86 ± 0.08
FAME ($N=12000$)	14.54 ± 0.58	18.72 ± 0.52	1.69 ± 0.17	0.16 ± 0.03	0.72 ± 0.13

1110 **Policy Evaluation Step** n for Adaptive Meta Warm-Up. In Table 6, a proper range of the policy
 1111 evaluation step does not affect the metric scores significantly. As expected, a small policy evaluation
 1112 step may not sufficiently select the best warm-up strategy, thus decreasing FT. We found $n = 600$ is
 1113 sufficient to maintain a favorable FT, while managing the forgetting as well.

Table 6: Ablation Study of **Policy Evaluation Step** n on MinAtar on Average Performance (Avg. *Perf*), Forward Transfer (*FT*), and Forgetting. Results (Mean \pm SE) are averaged over 10 sequences, each with 3 seeds.

Method	Breakout	Ave. Perf \uparrow Spaceinvader	Freeway	FT \uparrow	Forgetting \downarrow
FAME ($n=300$)	13.02 ± 0.49	19.19 ± 0.65	1.42 ± 0.11	0.12 ± 0.03	0.69 ± 0.07
FAME ($n=600$)	14.54 ± 0.58	18.72 ± 0.52	1.69 ± 0.17	0.16 ± 0.03	0.72 ± 0.13
FAME ($n=1200$)	13.46 ± 0.63	19.39 ± 0.42	1.83 ± 0.20	0.16 ± 0.03	0.79 ± 0.08
FAME ($n=5000$)	13.09 ± 0.41	19.32 ± 0.52	1.84 ± 0.18	0.14 ± 0.03	0.76 ± 0.07

F Experiments: Policy-based Continual RL with Continuous Action Space

F.1 Details of Experimental Setup and Comparison Methods

Hyperparameter Details. We employ a replay buffer size of $1M$ for the fast learner and $0.1M$ for the meta learner. The buffer size for the meta learner should not be overly large as we expect the algorithm to develop the continual learning capability without replaying too much data in the past. When the new environment arrives, the fast learner starts the training guided by the adaptive meta warm-up (i.e., knowledge transfer). At the same time, the replay buffer for the fast learner is reset, which aims only to contain transitions from the current task. In contrast, the meta learner maintains a buffer that stores the most recent 1% of transitions (50 episodes) leveraged to update the fast learner for each task. After finishing a task, the meta learner is trained using $5000 \times k$ mini-batches, where k is the number of tasks so far.

Choice of Sequence of Tasks. We deploy three task sequences in Meta-World as [32], where the sequence of tasks is as follows:

- ['button-press-v2', 'plate-slide-back-side-v2', 'window-close-v2', 'plate-slide-side-v2', 'peg-unplug-side-v2', 'plate-slide-back-v2', 'coffee-button-v2', 'window-open-v2', 'handle-pull-side-v2', 'door-close-v2'],
- ['plate-slide-back-side-v2', 'soccer-v2', 'sweep-into-v2', 'handle-pull-side-v2', 'plate-slide-side-v2', 'peg-unplug-side-v2', 'door-lock-v2', 'reach-v2', 'plate-slide-back-v2', 'coffee-button-v2'],
- ['coffee-push-v2', 'button-press-v2', 'reach-v2', 'peg-unplug-side-v2', 'reach-wall-v2', 'door-close-v2', 'window-open-v2', 'handle-pull-side-v2', 'plate-slide-back-side-v2', 'soccer-v2'].

F.2 More Experimental Results

Metric Scores. Table 7 presents detailed results on Final Average Performance, Forward Transfer, and Forgetting. Compared to the baselines, FAME consistently achieves superior performance across all three metrics. Interestingly, the best Forgetting score does not always align with the best Average Performance or Forward Transfer. This discrepancy arises because the Forgetting metric is evaluated

Table 7: Results on Meta-World on Average Performance (*Ave. Perf*), Forward Transfer (*FT*), and Forgetting. Results are presented as averages and standard errors across 3 seeds.

Methods	Avg. Perf \uparrow	FT \uparrow	Forgetting \downarrow
Reset	0.067 ± 0.046	0.000 ± 0.000	0.740 ± 0.080
Finetune	0.033 ± 0.033	-0.326 ± 0.070	0.383 ± 0.088
Average	0.000 ± 0.000	-0.503 ± 0.061	0.100 ± 0.056
FAME-WD	0.867 ± 0.063	0.029 ± 0.038	0.033 ± 0.033
FAME-KL	0.933 ± 0.046	0.063 ± 0.033	0.007 ± 0.040
Methods	Avg. Perf \uparrow	FT \uparrow	Forgetting \downarrow
Reset	0.100 ± 0.056	0.000 ± 0.000	0.697 ± 0.079
Finetune	0.000 ± 0.000	-0.328 ± 0.076	0.290 ± 0.080
Average	0.000 ± 0.000	-0.448 ± 0.066	0.103 ± 0.056
FAME-WD	0.800 ± 0.074	0.044 ± 0.038	0.060 ± 0.055
FAME-KL	0.767 ± 0.079	0.012 ± 0.047	0.027 ± 0.051
Methods	Avg. Perf \uparrow	FT \uparrow	Forgetting \downarrow
Reset	0.100 ± 0.056	0.000 ± 0.000	0.603 ± 0.099
Finetune	0.000 ± 0.000	-0.235 ± 0.070	0.27 ± 0.077
Average	0.000 ± 0.000	-0.426 ± 0.058	0.007 ± 0.005
FAME-WD	0.667 ± 0.088	0.006 ± 0.059	-0.013 ± 0.052
FAME-KL	0.733 ± 0.082	0.111 ± 0.054	0.033 ± 0.034

1142 based on the performance of the fast learner itself. In some cases, the fast learner may adapt quickly
 1143 while the meta-learner forgets slightly; in other cases, the fast learner may perform only moderately,
 1144 but the meta-learner retains the acquired knowledge more effectively.

1145 **Evaluation of Performance Profile and Final Average Performance.** The performance profile [3,
 1146 17] provides a comprehensive view of the fast learner’s overall performance across the entire task
 1147 sequence. As shown in Fig. 5 (left), both FAME variants consistently outperform all baselines,
 1148 demonstrating the meta-learner’s ability to effectively consolidate knowledge over time and facilitate
 1149 transfer learning. Moreover, Fig. 5 (right) highlights the advantage of FAME across all previously
 1150 seen tasks. While baseline methods typically degrade as more tasks are introduced, FAME-KL and
 1151 FAME-MD achieve the highest average performance—indicating minimal catastrophic forgetting and
 1152 robust retention over time.

1153 **Learning Curves of the Fast Learner.** Figure 6 showcases that the fast learner in our FAME methods
 1154 achieves higher success rates across three sequences of tasks throughout training. Learning curves

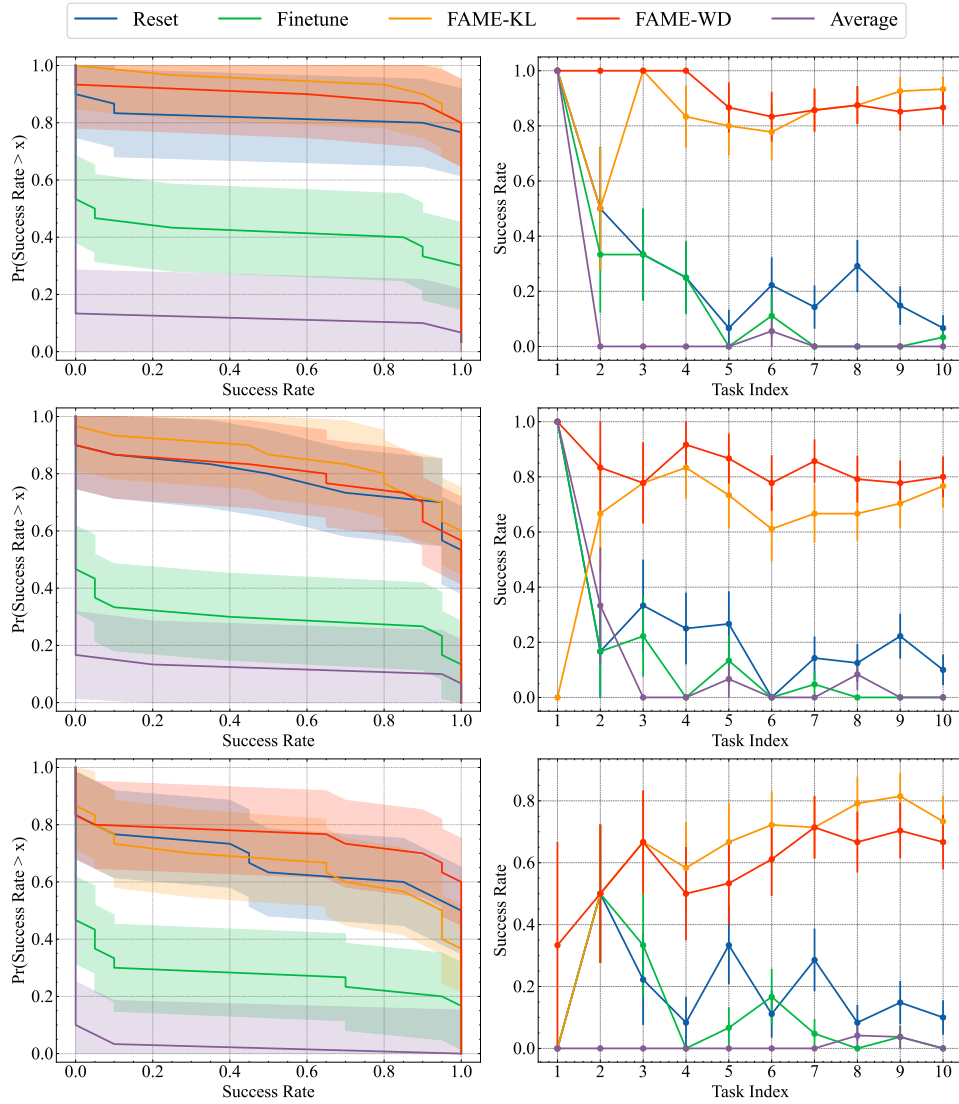


Figure 5: **(Left)** Performance profile of the fast learner across tasks, where the y-axis shows the proportion of tasks that achieve a success rate greater than or equal to the x-axis value. **(Right)** Average performance over time by evaluating the average success rates in the past tasks. **Each Row represents the result for one sequence of tasks.**

1155 are averaged over 3 seeds, and the shade region represents the standard error. This superiority implies
 1156 that the adaptive meta warm-up enhances the adaptation of the fast learner in each new environment.

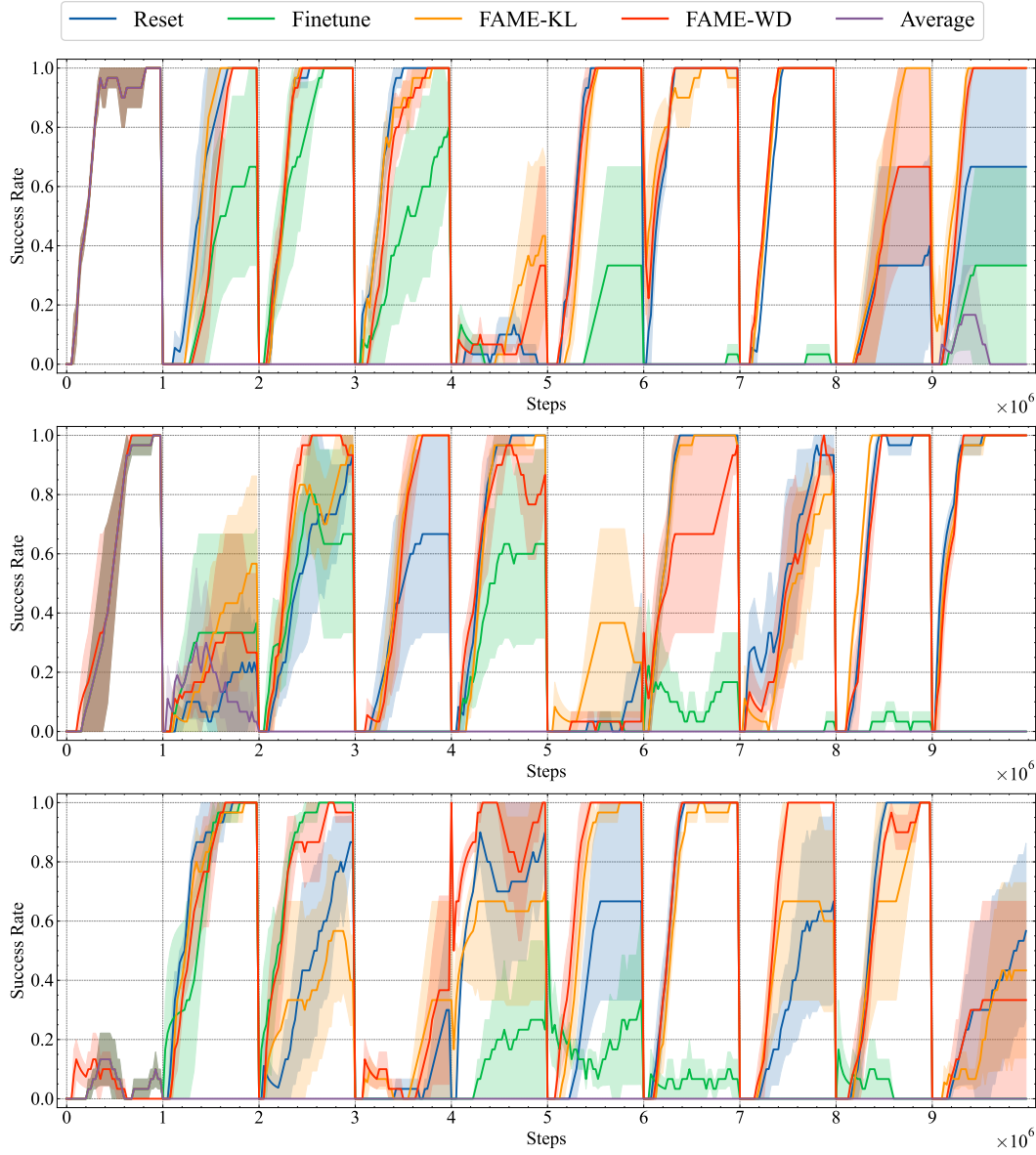


Figure 6: Learning curves of the fast learner on the Meta-World benchmark. The x-axis shows the total number of environment interactions, and the y-axis indicates the success rate. **Each Row represents the result for one sequence of tasks.**

1157