面向持续强化学习的有原则快速与元知识学习器

Ke Sun[1*]、**Hongming Zhang**[2*]、**Jun Jin**[1]、**Chao Gao**[3]、**Xi Chen**[4]、**Wulong Liu**[5]、**Linglong Kong**[1†]
[1]阿尔伯塔大学[2]中国科学院自动化研究所复杂系统认知与决策智能全国重点实验室[3]华为加拿大埃德蒙顿研究中心[4]华为诺亚方舟实验室[5]BetaInfinity{ksun6、jjin5、lkong}@ualberta.ca hongming.zhang@ia.ac.cn{chao.gao4、xi.chen4}@huawei.com wulong.liu@gmail.com

arXiv:2603.00903v1 [cs.LG] 1 Mar 2026

摘要

受人类学习与记忆系统的启发，尤其是海马体与大脑皮层之间的相互作用，本研究提出了一种由快速学习器和元学习器组成的双学习器框架，以解决持续强化学习（RL）问题。这两种学习器协同发挥各自不同但互补的作用：快速学习器侧重于知识迁移，而元学习器则确保知识整合。与通过最大化平均回报来共享知识的传统多任务强化学习方法不同，我们的元学习器通过显式地最小化灾难性遗忘，逐步整合新经验，从而支持快速学习器实现高效的累积知识迁移。为促进在新环境中的快速适应，我们引入了一种自适应元预热机制，有选择地利用以往知识。我们在多种基于像素和连续控制基准测试中开展实验，结果表明，与基线方法相比，我们提出的双学习器方法在持续学习方面表现更优。代码已发布于 https://github.com/datake/FAME。

1 引言

大多数深度强化学习（RL）算法（Sutton & Barto，2018年；Mnih 等人，2015；Schul- man 等人，2015；Haarnoja 等人，2018a；Schulman 等人，2017）都是针对单一任务设计的，在这类任务中，环境动态和奖励函数通常会随时间保持不变。相比之下，人类不断面对多样且不断变化的环境，并在一生中逐步学习解决新任务。要构建具备此类适应能力的通用智能体，就需要持续学习，即高效获取新知识，同时不遗忘先前学到的技能。在这一领域，持续强化学习（Khetarpal等人，2022年；Abel 等人，2023）作为一种重要范式应运而生，旨在平衡可塑性（快速适应新任务）与稳定性（保留既有知识）。理想的持续学习智能体能够将有用知识向前迁移，以加快在新环境中的学习，同时避免灾难性遗忘（Dohare 等人，2024）此前遇到的任务。

近期持续强化学习的研究涵盖了广泛的策略（Barreto 等人，2020年；Kessler 等人，2022年；Kaplanis 等人，2019年；Caccia 等人，2022年；Kaplanis 等人，2018年；Gaya 等人，2023年；Yang 等人，2023年；Wolczyk 等人，2022年；Anand & Precup, 2023；Wan 等人，2022年；Chandak 等人，2020年；Sun 等人，2025年；Liu 等人，2025年；Tang 等人，2025年）。尽管进展迅速，持续强化学习领域仍然较为零散：上述现有算法通常由启发式方法驱动，或基于不同的视角发展而来

* 同等贡献。
† 通讯作者。

（相关工作的详细比较见附录 A），但目前尚无原则性的方法来理解它们何时以及为何有效。缺乏理论基础使得人们难以判断知识迁移何时有益、如何减轻遗忘，以及如何设定明确的优化目标，从而阻碍了持续强化学习中算法的原则性开发。

为应对这些挑战，我们的研究通过以下两点为持续强化学习奠定新的基础：（1）定义马尔可夫决策过程差异，以正式量化环境相似性；（2）提出一种定量衡量灾难性遗忘的方法，适用于基于价值和基于策略的强化学习。在这些新基础和原则之上，我们提出了一种双学习器范式，模拟人类学习与记忆系统中海马体与皮层的交互（Kumaran 等人，2016），将持续强化学习分解为两个互补目标：知识迁移 和知识整合。具体而言，我们设置了两个各具特色又相辅相成的组成部分，即快速学习器和元学习器，分别对应人脑中海马体（快速学习器）和新皮层（元学习器）的功能角色。我们的方法还阐明了其与迁移和多任务强化学习问题之间更深层的联系（Schaul 等人，2015；Rusu 等人，2015；Parisotto 等人，2016；Rajendran 等人，2017；Teh 等人，2017；Bai 等人，2023）。

通过快速学习器进行知识迁移。 我们利用快速学习器，通过自适应地迁移元学习器中存储的先验知识，从新任务中快速获取知识。为避免潜在的负迁移问题，我们设计了一种自适应元预热 策略：在训练早期，或直接复制参数作为初始化，或加入行为克隆正则化，以引导探索。快速学习器在知识迁移中的作用类似于海马体。海马体迅速编码新的经验，并辨别现有知识的有效性；在新皮层的引导下，它专门负责快速适应新场景，以应对环境的即时变化或漂移。

通过元学习器进行知识整合。 快速学习器吸收新知识后，增量式知识整合会将新的经验纳入元学习器中存储的现有知识库。在此基础上，知识整合依据最小化灾难性遗忘 的原则，针对特定的散度指标进行增量更新。整合旧经验和新经验后，元学习器会增强自适应元预热，从而促进知识向下一个环境迁移。知识整合过程的作用类似于大脑皮层：它逐步将新知识整合、纳入并巩固到人脑现有的认知结构中，从而构建出泛化能力更强、更稳健且更稳定的决策系统。

贡献。 本研究的贡献可简要概括如下：

• 我们提出持续强化学习的新基础，包括MDP差异的定义以及灾难性遗忘的度量，为未来的算法创新奠定基础。

• 我们设计了一个双学习器系统，其中包含彼此不同但互补的快速学习器和元学习器，以执行知识迁移和知识整合。快速学习器与元学习器之间的相互作用模拟了大脑记忆系统中的海马-皮层对话。

• 我们提供了全面的实证研究，以验证我们的双学习器系统在离散和连续动作域中、以及在基于价值的和基于策略的强化学习算法上的效率。

2 问题设定与持续强化学习的新基础

问题设定。 令 $[K]$ 表示 $\{1, 2, ..., K\}$。我们考虑一个由 K 个任务组成的序列，其中每个任务 $k \in [K]$ 由马尔可夫决策过程（马尔可夫决策过程） $\mathcal{M}_k = \langle \mathcal{S}_k, \mathcal{A}_k, P_k, R_k, \gamma \rangle$建模。 $\mathcal{S}_k$和 $\mathcal{A}_k$ 表示状态和动作空间，P_k： $\mathcal{S}_k \times \mathcal{A}_k \to \mathcal{P}(\mathcal{S}_k)$ 为转移动态，R_k： $\mathcal{S}_k \times \mathcal{A}_k \to \mathbb{R}$ 为奖励函数，γ为折扣因子。我们将动作价值函数 $Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{i=0}^{\infty} \gamma^i R_{t+i+1} \mid S_t = s, A_t = a \right]$ 定义为在给定状态 s、动作 a和策略 π时的函数。遵循持续强化学习中的常见做法（Wolczyk 等人，2022；Khetarpal等人，2022；Malagon 等人，2024），我们采用三项假设：（1）相同的状态和动作空间，（2）已知任务边界，即半持续强化学习（Anand & Precup, 2023），以及（3）训练预算对应适中的模型规模和可接受的计算成本。一个“最优”策略可通过平衡对新任务的适应与先验知识保留，在所有任务上实现良好泛化。

基础 1：MDP 距离。 持续强化学习中的理论分析需要一种环境之间的定量相似性度量，以判断知识迁移何时有益或有害，并评估新任务对先前已学任务的干扰程度。理想的 MDP 距离应同时考虑马尔可夫决策过程之间奖励函数与转移动态的变化。为此，我们利用两个由马尔可夫决策过程确定的最优 Q 函数或任务特定最优策略之间的距离，来量化定义 1 中的 MDP 距离。

定义 1. *(MDP* 距离*)* 对于两个有限马尔可夫决策过程：$MDP_1 = (\mathcal{S}, \mathcal{A}, R_1, P_1, \gamma)$ 和 $MDP_2 = (\mathcal{S}, \mathcal{A}, R_2, P_2, \gamma)$，将它们的最优$Q$函数分别记为 Q_1^*和 Q_2^*，最优策略分别记为π_1^* 和 π_2^*。在某些散度或距离 d_Q 和 d_π下，基于Q值的MDP距离和基于策略的MDP距离定义为 $d_Q(Q_1^*, Q_2^*)$ 和$d_\pi(\pi_1^*, \pi_2^*)$，例如 ℓ_2 损失或 KL 散度。

基础 2：灾难性遗忘。 我们对持续强化学习中灾难性遗忘的定义，受到深度学习文献中所量化的分布漂移 和灾难性遗忘 的启发（Doan 等人，2021），相关内容在附录B中简要回顾。基于定义 1 中对 MDP 差异的定义，我们在定义 2 中引入两个 MDP 之间的灾难性遗忘。记 μ_k^π 和 μ_k^Q 为状态访问分布，其中策略 π 或在 Q上由 $\pi^Q(\cdot|s) = \arg\max_a Q(s,a)$定义的贪心策略 π^Q，在由 K 个任务组成的序列中与第 k个环境交互。

定义 2。 （两个环境间的灾难性遗忘）将 Q_{k-1}、Q_k 和 π_{k-1}、π_k 分别表示依次在第 $k-1$ 个和第 k 个环境上训练强化学习算法后得到的 Q 函数和策略。灾难性遗忘记为 CF，其定义为

$$CF(Q_{k-1}, Q_k) = \sum_{s,a} \mu_{k-1}^{Q_{k-1}}(s)\pi^{Q_{k-1}}(a|s) d_Q\left(Q_{k-1}(s,a), Q_k(s,a)\right), \tag{1}$$

$$CF(\pi_{k-1}, \pi_k) = \sum_{s} \mu_{k-1}^{\pi_{k-1}}(s) d_\pi\left(\pi_k(\cdot|s), \pi_{k-1}(\cdot|s)\right). \tag{2}$$

对于每个 s 和 a，权重 $\mu_{k-1}^{Q_{k-1}}(s)\,\pi^{Q_{k-1}}(a|s)$ 和 $\mu_{k-1}^{\pi_{k-1}}(s)$ 刻画了在衡量 Q 函数与策略之间差异时的相对重要性。关键在于，我们使用先前的策略 π_{k-1} $(\pi^{Q_{k-1}})$ 而非当前策略 π_k (π^{Q_k}) 来评估这些权重，因为过去的策略更能反映旧任务中最重要的状态和动作。相反，如果我们使用 π_k (π^{Q_k}) 进行权重评估，那么先前
重要的状态–动作对可能会被忽略，因为新策略可能不再访问它们。

3 FAME：有原则的快速与Meta知识持续强化学习

所提出的 FAME 方法适用于基于价值的强化学习和基于策略的强化学习。在基于价值的强化学习中，我们将学习任务 k 后更新的快速学习器记为 Q_k，随后由元学习器 Q_k^M整合先前元学习器 Q_{k-1}^M 和 Q_k中的知识。在基于策略的强化学习中，我们将学习任务 k 后的快速学习器记为 π_k，随后由元学习器 π_k^M整合先前元学习器 π_{k-1}^M 和 π_k中的知识。FAME 中快速学习器与元学习器的耦合更新过程如图1所示。原则上，快速学习器在元学习器的指导下快速学习新任务

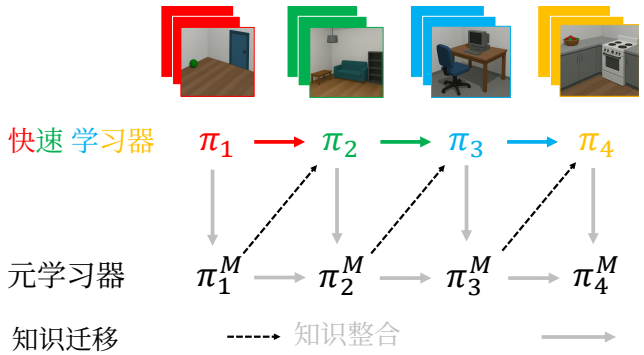


图1：FAME 示意图。在基于价值的持续强化学习中，快速学习器可相应地记为 $\{Q_k\}_{k=1}^K$，而非 $\{\pi_k\}_{k=1}^K$。

通过所提出的自适应元预热。同时，元学习器通过知识整合，巩固前一个元学习器和当前快速学习器的经验，以最大限度地减少灾难性遗忘。

3.1 基于价值的离散动作空间持续强化学习

3.1.1 知识整合：灾难性遗忘最小化原则

快速学习器 Q_k 完成在第 k个环境中的训练后，知识整合阶段便开始。在此阶段，通过结合前一个元学习器 Q_{k-1}^M 编码的先验知识与快速学习器 Q_k获取的新知识，更新元学习器 Q_k^M以整合信息。与旨在最大化平均奖励的经典多任务强化学习不同，我们的元学习器经过专门设计，旨在最小化定义 2 中定义的灾难性遗忘。

基于 Q 值的灾难性遗忘及其局限性。我们将式 1 中定义的基于 Q 值的灾难性遗忘扩展到由 K 个任务组成的序列。在第 k个环境中，最优元Q值函数 Q_k^M通过求解以下目标函数得到，并且是该目标函数的最小化解：

$$Q_k^M = \arg\min_{\widetilde{Q}_k^M} \sum_{i=1}^{k} \sum_{s,a} \mu_i^{Q_i}(s)\pi^{Q_i}(a|s) \left(Q_i(s,a) - \widetilde{Q}_k^M(s,a)\right)^2, \tag{3}$$

其中， $\mu_i^{Q_i}$表示贪心策略 π^{Q_i} （即$\pi^{Q_i}(\cdot|s) = \arg\max_a Q_i(s, a)$）与第 i个环境交互时的状态访问分布。直观而言，通过求解式 3 得到的直接最小化解Q_k^M是 $\{Q_i\}_{i=1}^k$的加权平均。然而，通过存储所有先前的 Q 函数来实现持续学习能力，无法随着任务数量增加而扩展，而这是持续强化学习的一项关键要求。事实上，我们可以将上述目标函数重写为前一个元学习器 Q_{k-1}^M 与快速学习器 Q_k之间的增量更新规则，具体内容见附录 C.1。尽管如此，基于 Q 值的灾难性遗忘有一个根本局限：它主要适用于 Q 值尺度相似的不同环境，例如转移动态不同但奖励函数相同的环境。由于新到来的环境尚不可知，Q 值的尺度可能难以学习，因为它不一定有界，而且可能相当不稳定（Rusu 等人，2015）。与低奖励任务相比，先前已充分学习且奖励较高的任务在知识整合过程中往往更加显著（Zhang 等人，2023）。因此，式 2 中基于策略的灾难性遗忘定义比式 1 中基于 Q 值的定义适用范围更广，是更可取的替代方案。此外，与价值函数相比，策略本身可能具有较低的方差，从而有助于提升性能和稳定性（Greensmith 等人，2004）。

基于策略的灾难性遗忘。即使在基于价值的持续强化学习中，为了增量更新元学习器，最好也采用基于式 2 的灾难性遗忘的基于策略的定义。与式 3 中基于 Q 值的灾难性遗忘类似，通过求解以下目标函数，在第 k 个环境中的最优元策略 π_k^M 是该目标函数的最小化解：

$$\pi_k^M = \arg\min_{\widetilde{\pi}_k^M} \sum_{i=1}^{k} \sum_{s} \mu_i^{\pi_i}(s) d_\pi \left(\pi_i(\cdot|s), \widetilde{\pi}_k^M(\cdot|s)\right). \tag{4}$$

基于价值的持续强化学习的增量式 Softmax 元学习器更新。对每个 $i \in [K]$，定义权重函数 $w_i^Q(s, a) = \mu_i^{Q_i}(s)\pi^{Q_i}(a|s)$。对于任意可测函数 $f(s, a)$ 和满足 $\sum_s, {}_a w(s, a) = 1$ 以及 $w(s, a) \geq 0$ 的权重函数 w，对每个 s 和 a，我们定义$\mathbb{E}_w[f] = \sum_{s,a} w(s, a)f(s, a)$。采用分类表示后，Q 值可以转换为 Softmax（玻尔兹曼）策略，从而使基于价值的持续强化学习能够最小化式 4 中定义的基于策略的灾难性遗忘目标。具体而言，给定温度参数 τ，我们将 $\pi^{Q_i}(a|s) = \exp(Q_i(a|s)/\tau) / \sum_{a'} \exp(Q_i(a'|s)/\tau)$ 记作。通过使用 KL 散度作为 d_π，我们在命题 1 中推导出简洁的元学习器更新规则。

命题 1 (基于增量 Softmax Q 值的元学习器更新)。 记作 $\widetilde{\pi}_k^M(a|s) = \exp\left(\widetilde{Q}_k^M(a|s)/\tau\right) / \sum_{a'} \exp\left(\widetilde{Q}_k^M(a'|s)/\tau\right)$。经过 *Softmax* 策略变换后，基于 *Q* 值的元学习器增量更新可写为

$$Q_k^M = \arg\min_{\widetilde{Q}_k^M} \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q}\left[\log \frac{\pi_{k-1}^M}{\widetilde{\pi}_k^M}\right] + \mathbb{E}_{w_k^Q}\left[\log \frac{\pi^{Q_k}}{\widetilde{\pi}_k^M}\right] = \arg\max_{\widetilde{Q}_k^M} \sum_{i=1}^{k} \mathbb{E}_{w_i^Q}\left[\log \widetilde{\pi}_k^M\right]. \tag{5}$$

命题 1 的证明很直接，因此为完整起见，将其推迟至附录 C.2。关键的是，通过将元学习器 Q_k^M拟合到已遇到环境中的状态-动作分布混合，最小化公式 5 中基于策略的灾难性遗忘，等价于最大似然估计量（MLE）。我们强调，公式 5 中最终简化后的目标与 Q_{k-1}^M 和 Q_k 无关；然而，原则上，知识整合采用增量更新规则的形式，我们将在第 3.2.1 节具体说明。

3.1.2 通过自适应元预热进行知识迁移

挑战。 有效的知识迁移需要在能够利用已有知识的前提下，迅速适应新环境。然而，常用的微调方法在任务相似时效果良好，但可能导致持续强化学习中频繁出现的负迁移问题（Ahn 等人，2025；Wolczyk 等人，2022）。负迁移是可塑性丧失的一个关键因素（Dohare 等人，2024），由于两个任务之间存在差异，会导致性能下降。从头训练（即重置）很容易实现，可用于规避负迁移（Chen 等人，2024；Ahn 等人，2025）。然而，这种简单的预热缺乏灵活性，无法充分利用积累的知识来加快对新任务的适应。

通过一对多假设检验实现自适应元预热。 当新任务到来时，常见的做法是使用元学习器的参数初始化快速学习器。然而，当新环境与以往环境存在显著差异时，过去任务中获得的知识和技能可能会产生误导。例如，当新信息与先前经验相矛盾时，人们显然更容易做出错误决策或采取次优动作。为协调可能相互冲突的目标，我们提出一种自适应元预热方法，从**先前的元学习器、随机学习器（即重置）和先前的快速学习器（即微调）**中选择最有效的预热策略。形式上，自适应元预热被表述为一种基于与新环境早期交互期间策略评估的一对多假设检验。当第 k个任务到来时，我们可以使用三种类型的预热学习器，包括先前的快速学习器 Q_{k-1}、经过 Q_{k-1}^M 的 Softmax 变换的元策略 π_{k-1}^M，以及与策略 π^0关联的随机 Q 函数 Q^0。对这三种预热学习器进行评估，可得到由 $V_k^f = \mathbb{E}_{\pi_{k-1}}[R]$、$V_k^M = \mathbb{E}_{\pi_{k-1}^M}[R]$和 $V_k^r = \mathbb{E}_{\pi^0}[R]$定义的价值函数。对于每个到来的任务 k，带有复合原假设的一对多假设检验表示为

$$H_0 : V_k^M \leq \max\left\{V_k^f, V_k^r\right\} \quad \text{vs.} \quad H_1 : V_k^M > \max\left\{V_k^f, V_k^r\right\}. \tag{6}$$

当零假设 H_0 无法被拒绝时，我们进一步通过常见的参数假设检验（例如 t 检验）比较 V_k^f和 V_k^r。在大多数场景中，根据经验排序选择最佳预热策略通常表现良好。然而，在安全关键场景（例如自动驾驶）中，对用于策略评估的自适应收集的数据集进行严格的统计检验至关重要，可通过自助法或任意时刻有效推断（Ramdas 等人，2023年）来实现。

通过行为克隆正则化进行元预热。 一旦我们拒绝 H_0，即可进行元预热。然而，直接通过元策略 π_{k-1}^M 初始化快速学习器 Q_k 并不可行，因为在命题 1 的更新下，元学习器现在表示为策略而非 Q 函数。解决这种策略到价值的迁移不匹配的一种简单有效方法，是在早期训练阶段施加行为克隆（BC）正则化，此时元策略π_k^M作为专家用于数据收集和早期探索。具体而言，Q_k 是 BC 正则化损失 $L(Q_k) = L_0(Q_k) + \lambda \mathbb{E}_s\left[\mathrm{KL}(\pi_{k-1}^M(\cdot|s)||\pi^{Q_k}(\cdot|s))\right]$的极小化元，其中 $L_0(Q_k)$ 是用于更新 Q_k 的原始损失，例如深度 Q 网络中的均方误差或 Huber 损失（Mnih 等人，2015）。

3.1.3 算法：基于价值的 FAME

元缓冲区 $\mathcal{M}$ 中的知识整合。在每个环境中更新快速学习器的最后 N 步，我们还会将状态-动作对存储在元学习器的缓冲区 $\mathcal{M}$中，用于近似 w_i^Q 在 $i \in [k]$中的值（见公式 5）。请注意，存储的状态-动作对仅占每个任务训练数据集的一小部分（在我们的实验中约为 1% 或 2%），因此元缓冲区 $\mathcal{M}$的规模适中。元缓冲区的规模适中且固定，这在实际应用中至关重要，因为在持续强化学习中，我们不应存储过多的历史数据。

算法 1 在第 k个环境中的基于价值的 FAME 更新

1：初始化：快速缓冲区 $\mathcal{F}$、元缓冲区 $\mathcal{M}$、 Q_{k-1}^{M}、 Q_{k-1}、 Q^{0}、预热步数 L、估计步数N。2：# 知识迁移：自适应元预热 3：在 L 步内，通过公式 6 在 $\{Q_{k-1}, Q_{k}^{M}, Q^{0}\}$ 中初始化 Q_{k} 4：对于 $t = L$到 T，执行 5：观察 S_t，采取动作 A_t，接收 R_t，观察 S_{t+1} 6：将（S_t， A_t， R_t， S_{t+1}）存储到 $\mathcal{F}$中 7：更新 Q_k 8：如果 $t > T - N$，则 9：将（S_t， A_t）存储到 $\mathcal{M}$ 中 # 用于估计 w_k^Q 10：结束如果 11：结束循环 12：重置 $\mathcal{F}$ 13：# 知识整合：最大限度减少灾难性遗忘 14：使用 $\mathcal{M}$中的状态-动作对，通过公式 5 更新 Q_k^M

算法。 将快速学习器的缓冲区记为 $\mathcal{F}$，将 Q^0 记为随机初始化的 Q 函数。我们将 T 记为每个环境中的时间步数。如算法 1 所示，当第 k个环境到来时，我们在最初的 L步内，通过自适应元预热策略，在前一个元学习器 Q_{k-1}^{M}、前一个快速学习器 Q_{k-1} 和随机学习器Q^0 （即重置）之间，为快速学习器 Q_k 进行热启动。自适应元预热充分利用先前的信息，执行自适应知识迁移。第 k个任务结束后，知识整合阶段随即开始，元学习器 Q_k^M根据元缓冲区 $\mathcal{M}$中收集的数据，通过公式 5 进行更新。原则上，元学习器 Q_k^M通过增量更新规则，将在 Q_k 中获取的知识整合到 Q_{k-1}^{M} 中。

3.2 连续动作空间下基于策略的持续强化学习

3.2.1 知识整合：灾难性遗忘最小化原则

与命题 1 中采用增量式 softmax 元学习器更新的基于价值的持续强化学习不同，在基于策略的持续强化学习中，我们直接针对参数化策略函数最小化式 4 中的基于策略的灾难性遗忘。具体的增量更新规则取决于**所选的** d_π **以及我们在**连续动作空间中表示策略的方式。接下来，我们将介绍两种基于策略的持续强化学习方法，分别采用正向 KL 散度和 Wasserstein 距离。

方法 1（FAME-KL）：正向 KL 散度下的策略蒸馏。我们表明，基于策略的知识整合可归结为一种策略蒸馏形式。与基于价值的持续强化学习中的命题 1 类似，在可访问的概率策略下，采用正向 KL 散度实例化式 4 中的基于策略的知识整合，可得到如下更新规则：

$$\pi_k^M = \arg\max_{\widetilde{\pi}_k^M} \sum_{i=1}^{k} \mathbb{E}_{w_i} \left[\log \widetilde{\pi}_k^M \right], \tag{7}$$

其中，我们记 $w_i(s,a) = \mu_i^{\pi_i}(s)\pi_i(a|s)$ 为第 i 个环境中的基于策略的稳态动作分布。重要的是，上述基于策略的知识整合目标与策略蒸馏中使用的知识蒸馏更新（Rusu 等人，2015年）以及典型的多任务强化学习表述（Teh 等人，2017年）相一致，从而建立了一个引人关注的联系。

方法 2（FAME-WD）：基于 Wasserstein 距离（WD）的知识整合。尽管 KL 散度因计算简单而常被采用，但在比较更复杂的策略分布时，Wasserstein 距离更为合适。与仅衡量分布间逐点差异的 KL 散度不同，Wasserstein 距离根据定义会显式考虑数据空间的底层几何结构，例如策略输出所在的概率空间，因此能够更充分地刻画分布差异（Panaretos 与 Zemel，2019；Arjovsky 等人，2017）。在命题 2 中，我们推导了基于 Wasserstein 距离的策略

增量更新规则。具体而言，当策略函数表示为（多元）高斯分布时（这在许多基于策略的算法中很常见），便可得到闭式增量更新规则，从而使元学习器能够高效地整合知识。证明见附录 C.3。

命题 2（Wasserstein 距离下基于策略的元学习器增量更新）。 令 d_π 表示定义 *2* 的公式 *2* 中记作 W_2^2的平方 *2-Wasserstein* 距离。策略表示为关于动作 a的独立（多元）高斯分布。最小化公式 *4* 中基于策略的灾难性遗忘等价于：

$$\pi_k^M = \arg\min_{\widetilde{\pi}_k^M} \left\{ \sum_{i=1}^{k-1} \sum_s \mu_i^{\pi_i}(s) W_2^2 \left(\widetilde{\pi}_k^M(\cdot|s), \pi_{k-1}^M(\cdot|s) \right) + \sum_s \mu_k^{\pi_k}(s) W_2^2 \left(\widetilde{\pi}_k^M(\cdot|s), \pi_k(\cdot|s) \right) \right\}. \tag{8}$$

3.2.2 通过自适应元预热进行知识迁移

对于基于策略的强化学习中的自适应元预热，我们在前 L 步进行策略评估，并在新任务到达时执行公式 6 中的一对多假设检验，这与基于价值的持续强化学习类似。在快速策略 π_{k-1}、元策略 π_{k-1}^M 和随机策略 π^0 中确定表现最佳的预热策略后，我们直接初始化快速策略。与采用第 3.1.2 节介绍的基于价值的持续强化学习中使用的行为克隆正则化相比，将参数初始化用作元预热策略更便于部署。

算法。 由于基于策略的 FAME 算法的描述与算法 1 非常相似，我们将其与计算成本和实践指南的讨论一并放在附录 D 中。此外，我们还在附录 E 中对 FAME 框架进行了总体讨论。

4 实验

本节中，我们将在来自多个环境和领域的一系列任务上验证 FAME 方法，包括第 4.1 节中具有离散动作空间的基于像素的任务，以及第 4.2 节中具有连续动作空间的控制问题。我们的核心假设是，FAME 中快速学习器与元学习器之间的知识迁移和知识整合相互作用，既有利于前向迁移（即可塑性），也有利于缓解灾难性遗忘（即提升稳定性）。

评估指标。 我们采用持续强化学习中的标准指标（Wolczyk 等人，2021年；2022年）来评估平均性能、衡量稳定性的遗忘，以及量化可塑性的前向迁移 。令 $p_i(t)$ 表示在任务 i 中，于时间 t 使用策略且具备$t \in [K \cdot T]$时的成功率或平均回报，其中 K 为环境数量， T为每个任务中的总时间步数。 $p_i(t)$ 是任务特定的：对于基于像素的任务， $p_i(t) \in \mathbb{R}$ ；对于控制任务， $p_i(t) \in [0,1]$ 。

• **平均性能**。 平均性能是在时间 t 的策略上，对所有 K个任务通过 $P_K(t) = \frac{1}{K}\sum_{i=1}^{K} p_i(t)$进行评估的。默认情况下，当 $t = K \times T$时，平均性能根据最终策略计算。对于FAME，该指标在元学习器上计算。

• **前向迁移（FT）**： 前向迁移定义为所考虑算法的训练曲线与基线之间的归一化面积。即，$\mathrm{FT} = \frac{1}{K}\sum_{i=1}^{K} \mathrm{FTr}_i$， 其中

$$\mathrm{FTr}_i = \frac{\mathrm{AUC}_i - \mathrm{AUC}_i^b}{1 - \mathrm{AUC}_i^b}, \quad \mathrm{AUC}_i = \frac{1}{\Delta}\int_{(i-1)\cdot\Delta}^{i\cdot\Delta} p_i(t)\mathrm{d}t, \quad \mathrm{AUC}_i^b = \frac{1}{\Delta}\int_{(i-1)\Delta}^{i\Delta} p_i^b(t)\mathrm{d}t. \tag{9}$$

T为在基于像素的任务中评估该指标，我们首先对每个任务中的 $p_i(t)$进行归一化，以确保曲线下面积 $\in [0,1]$ 然后我们计算前向迁移的归一化指标。

• **遗忘（F）**： 遗忘是某一任务结束时的策略与整个任务序列结束后的策略之间的性能差异。即， $F = \frac{1}{K}\sum_{i=1}^{K} F_i$， 其中 $F_i = p_i(i \cdot T) - p_i(K \cdot T)$。

实验设置。**(1)** 对于基于像素的任务，我们在 MinAtar 和 Atari 游戏上开展实验（Young & Tian, 2019；Bellemare 等人，2013）。MinAtar 是一种常用的持续强化学习基准（Anand & Precup, 2023；Tang 等人，2025），计算需求相对较低，因此我们能够遍历一系列超参数，报告基于 30 个随机种子计算的统计结果，并探究我们所提方法的机制及优势。

我们在 Breakout、Freeway 和 Space Invaders 游戏中使用深度 Q 网络（Mnih 等人，2015），总共运行 350 万步，每 50 万步随机选择三个游戏之一，即每个任务序列包含 7 个任务。我们按照 Malagon 等人（2024）的方法研究两个 Atari 游戏序列，其中包括 *ALE/SpaceInvaders-v5* 环境的 10 种游戏模式和 *ALE/Freeway-v5* 环境的 7 种游戏模式。我们采用近端策略优化（PPO）（Schulman 等人，2017），每个任务运行 $1M$ 时间步。**(2)**对于机械臂操作任务，我们采用 Meta-World（Yu 等人，2020），这是一个标准且成熟的基准，常用于持续强化学习（Malagon 等人，2024；Chung 等人，2024；Ahn 等人，2025）。请注意，传统的 Continual World（Wolczyk 等人，2021）基于 Meta-World 构建，并采用特定的任务序列（例如 CW10 或 CW20）。相比之下，我们按照 Chung 等人（2024）的方法，在 Meta-World 中评估 3 个随机选取的任务序列（见附录 G.1），从而实现更灵活、更稳健的评估。我们采用软演员-评论家（SAC）算法（Zhang 等人，2020；Haarnoja 等人，2018a），每个任务运行 $1M$ 时间步，每个序列包含 10 个任务。

4.1 离散动作空间中的基于像素的环境

对比方法。(1) 对于 MinAtar，我们遵循 Anand & Precup, 2023，将 FAME 方法与 DQN（重置）、DQN-微调（微调）、采用大缓冲区的深度 Q 网络（大缓冲区）、能够识别任务身份的多头深度 Q 网络（多头）以及 PT-DQN（Anand & Precup, 2023）进行比较。FAME 方法中的快速学习器和元学习器均采用相同的深度 Q 网络架构。(2) 对于 Atari 游戏，我们加入 PackNet（Mallya 和 Lazebnik，2018）和 ProgressiveNet（Rusu 等，2016）作为基线方法。FAME 中的快速学习器和元学习器均采用相同的近端策略优化架构。除微调方法外，每当出现新环境时，我们都会重置所有基线方法的参数。相比之下，FAME 会在快速学习器、随机初始化以及采用行为克隆正则化的初始学习之间进行自适应元预热，详见第 3.1.2 节。实验设置和超参数的更多细节见附录 F.1。

表 1：在 **MinAtar** 上的主要结果，包括平均性能（平均性能）、前向迁移（前向迁移）和遗忘。结果（均值 ± 标准误）是对 10 个序列取平均，每个序列包含 3 个随机种子。↑表示正向指标（越大越好），而 ↓为负向指标（越小越好）。重置是评估前向迁移的基线。遗忘按每个任务的标准差进行归一化。

方法	**平均性能** ↑			**FT** ↑	**遗忘** ↓
	Breakout	Spaceinvader	Freeway		
重置	6.51 ± 1.67	3.29 ± 3.09	0.74 ± 0.38	0.00 ± 0.00	1.31 ± 0.23
微调	10.62 ± 2.75	4.95 ± 2.92	0.89 ± 0.49	0.13 ± 0.03	1.26 ± 0.32
多头	6.85 ± 1.76	3.26 ± 2.99	0.94 ± 0.42	-0.01 ± 0.00	1.25 ± 0.22
大缓冲区	10.71 ± 2.84	3.24 ± 2.91	1.16 ± 0.59	**0.16 ± 0.02**	1.65 ± 0.33
PT-DQN	0.39 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.07 ± 0.02	1.64 ± 0.02
FAME	**14.54 ± 0.58**	**18.72 ± 0.52**	**1.69 ± 0.17**	**0.16 ± 0.03**	**0.72 ± 0.13**

主要结果：MinAtar。 表 1 汇总了所有方法的指标分数，结果表明，FAME 在提升知识迁移和保留全部知识以减轻灾难性遗忘方面，始终优于其他基线方法。值得注意的是，在平均性能方面，除 PT-DQN 外，FAME 在所有算法中最为稳定，波动最小；PT-DQN 中的永久价值函数（即元学习器的对应部分）来自 (Anand & Precup, 2023)，其保留知识的能力有限，因此平均性能几乎为零。就前向迁移而言，LargeBuffer 的表现与 FAME 相近，因为存储更多过去的知识也有助于适应已知环境。关于 λ、预热步数 L和估计步数 N 的详细敏感性分析见附录 F.2。

知识整合性能。 图 2（左）展示了所有方法在每个任务结束时的平均性能，反映了灾难性遗忘的趋势。结果表明，在大多数情况下，FAME 在整个训练过程中都取得了最高的平均性能，验证了元学习器通过知识整合保留信息的有效性。

自适应元预热在知识迁移中的性能。图 2（右）展示了智能体遇到不同类型的新到达环境时的预热选择比例，揭示了自适应机制。具体而言，如果智能体此前已在$\mathcal{M}$ 中存储了有关新到达环境的相关数据，则有 95.1% 的概率会选择元预热。当

出现与智能体已有知识相悖的新任务时，自适应元预热中更常选择随机初始化。附录 F.3 给出了所有算法的学习曲线。

主要结果：Atari 游戏。我们进一步在两个 Atari 游戏序列上，将 FAME 与更多基线方法进行比较：*ALE/SpaceInvaders-v5*和*ALE/Freeway-v5*。表 2 展示了 FAME 在平均性能和前向迁移方面均优于所有基线方法。虽然微调也能从前向迁移中获益，尤其是在 SpaceInvader 上，其前向迁移效果与 FAME 相当，但我们推测，这一优势源于相似的底层环境动

表 2：Atari 游戏的主要结果，包括平均性能（平均性能）和前向迁移（前向迁移）。结果（均值 ± 标准误）为 3 个随机种子的平均值。由于 PackNet 和 ProgressiveNet 会保存过去的模型参数，因此遗忘指标被省略；它们的遗忘值为零。重置是前向迁移的基线。

方法	Freeway		SpaceInvader	
	平均性能 ↑	FT ↑	平均性能 ↑	FT ↑
重置	0.16 ± 0.18	0.00	0.10 ± 0.22	0.00
微调	0.21 ± 0.17	0.53	0.61 ± 0.41	**0.65**
ProgressiveNet	0.39 ± 0.25	0.21	0.61 ± 0.03	0.06
PackNet	0.41 ± 0.24	0.18	0.47 ± 0.06	0.17
FAME	**0.90 ± 0.12**	**0.68**	**0.96 ± 0.02**	0.63

y力学和目标 j中，尽管游戏模式各不相同，但每个任务序列都是如此。我们还在附录 F.3 中提供了所有所考虑算法的学习曲线。

4.2 连续动作空间中的机器人操作任务

对比方法。(1) 重置；(2) 微调；(3) 平均值：在评估评论家损失时，我们对所有历史任务的时序差分（TD）目标取平均；(4) PackNet；(5) FAME-KL：我们采用式 7 中基于 KL 散度的知识整合（方法 1）；(6) FAME-WD：我们采用式 8 中基于 Wasserstein 距离的知识整合（方法 2）。所有方法均采用与标准软演员-评论家（SAC）相同的网络架构。在自适应元预热中，我们针对随机策略、前一快速策略和元策略进行策略评估，每种策略评估 10 个回合。随后，我们用表现最佳的策略初始化快速策略。评估过程中收集的数据也会存储在快速学习器的回放缓冲区 $\mathcal{F}$ 中，且不会产生额外的环境交互成本。关于 FAME方法 (5) 和 (6) 的更多实验细节见附录 G.1。

主要结果。如表3所示，FAME-KL 和 FAME-WD 的表现均显著优于大多数基线方法。PackNet 通过存储过去的模型参数（的掩码），并预先获知任务标识符及其数量，实现了零遗忘；但在实际场景中，这种做法不太实用。相比之下，FAME 出色的前向迁移表明，

表3：**Meta-World** 上的主要结果。在平均性能（平均性能）、前向迁移（前向迁移）和遗忘方面的结果，均为 3 个序列的平均值。结果以 10 个随机种子的平均值和标准误表示。

方法	平均性能 ↑	FT ↑	遗忘 ↓
重置	0.093 ± 0.017	0.000 ± 0.000	0.710 ± 0.030
微调	0.037 ± 0.011	-0.265 ± 0.028	0.427 ± 0.033
平均值	0.013 ± 0.007	-0.530 ± 0.024	0.070 ± 0.022
PackNet	0.491 ± 0.025	-0.194 ± 0.018	**0.000 ± 0.000**
FAME-KL	0.733 ± 0.026	**0.022 ± 0.015**	0.073 ± 0.019
FAME-WD	**0.767 ± 0.024**	-0.003 ± 0.014	0.023 ± 0.015

自适应元预热通过利用元学习器的先验知识，促使快速学习器适应新环境。此外，最高的平均性能和几乎最低的

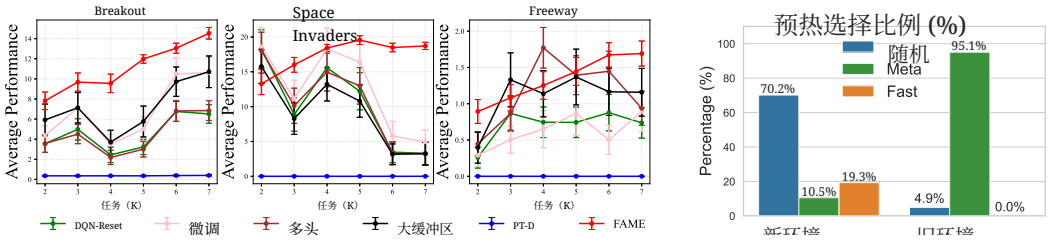


图 2：（左）在**MinAtar**上，策略在 10 个序列中各项任务上的平均性能。结果取 3 个随机种子的平均值。每个数据点处的竖线表示标准误。（右）当新到达的环境是此前遇到过的或全新的环境时，三种预热策略的选择比例。

我们 FAME 方法的遗忘表明，元学习器通过在知识整合中进行增量更新来巩固所有过往知识。

知识迁移性能。为了全面验证 FAME 中自适应元预热带来的知识迁移收益，我们展示了性能概况（Agarwal 等，2021年），该概况反映了快速学习器在整个任务序列中的整体性能。图 3（左）表明，两种 FAME 方法均优于所有基线，证实元学习器有效地

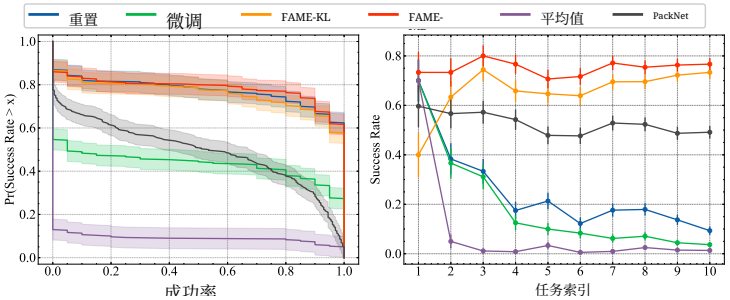


图 3：（左）快速学习器在各任务上的性能概况，其中纵轴表示成功率大于或等于横轴数值的任务比例。（右）通过评估 10 个随机种子在历史任务上的平均成功率，得出的平均性能。

con巩固知识并增强知识迁移。学习曲线和性能概况 -
文件 每个序列的结果也见附录 G.2。

知识整合性能。 为体现 FAME 容易发生灾难性遗忘的倾向，我们还展示了元学习器在每个任务结束时的平均性能，并对 3 个序列取平均值。如图 3（右）所示，在所有已遇到的任务中，FAME-KL 和 FAME-MD 随时间推移均保持着最高的平均性能。

5 讨论与结论

本文为持续强化学习奠定了基础，并提出了一种新颖的双学习器算法，通过快速学习器和元知识学习器的耦合更新来实现知识迁移与整合。这里值得重申两点。(1) 自适应地选择实用的先验知识（例如，通过假设检验）对于克服负迁移问题至关重要。(2) 基于现有多任务学习目标推导增量更新规则，是连接持续强化学习与多任务强化学习的必要条件。

局限性与未来工作。本研究使用元学习器来保留所有知识。另一种可行方案是学习有效的潜在表示，它不仅能提炼所有知识，还能进行高效推理，从而指导对新环境的适应。除了所提出的自适应元预热之外，未来还可以探索更多知识迁移技术，例如上下文嵌入。最后，将我们的算法扩展到完整的持续强化学习场景（无需获知任务边界，可通过开发在线一对多假设检验实现），或放宽状态和动作空间相同的假设（Hu 等，2025），对实践者也很有价值。

伦理声明。 本研究聚焦于持续强化学习的理论与算法方面，不涉及人类受试者、个人数据或敏感应用。我们承认，部署于现实系统中的持续学习器若缺乏谨慎的控制与监测，可能因遗忘或迁移错位而表现出意外行为。然而，本研究仍纯粹面向理论与算法，因此我们预计本研究不会引发任何直接的伦理问题。

致谢

Linglong Kong 部分获得了加拿大 CIFAR 人工智能讲席计划、阿尔伯塔机器智能研究所（AMII）、加拿大自然科学与工程研究理事会（NSERC）以及 NSERC 加拿大研究讲席计划的资助。Hongming Zhang 获得了中国科学院自动化研究所复杂系统认知与决策智能全国重点实验室的资助（项目编号：E5SPFZ0112）。我们也感谢审稿人和领域主席提出的所有建设性建议和意见。

参考文献

David Abel、Yuu Jinnai、Sophie Yue Guo、George Konidaris 和 Michael Littman。《终身强化学习中的策略与价值迁移》。载于国际机器学习会议，第 20–29 页。PMLR，2018年。

David Abel、Andr´e Barreto、Benjamin Van Roy、Doina Precup、Hado van Hasselt 和 Satinder Singh。持续强化学习的定义。神经信息处理系统进展，2023年。

Rishabh Agarwal、Max Schwarzer、Pablo Samuel Castro、Aaron C Courville 和 Marc Bellemare。统计悬崖边缘的深度强化学习。神经信息处理系统进展，34：29304–29320，2021年。

Hongjoon Ahn、Jinu Hyeon、Youngmin Oh、Bosun Hwang 和 Taesup Moon。持续强化学习中负迁移的普遍程度：分析与一个简单的基线。载于第十三届国际学习表征会议，2025年。

Nishanth Anand 和 Doina Precup。持续强化学习中的预测与控制。神经信息处理系统进展，2023年。

Marcin Andrychowicz、Filip Wolski、Alex Ray、Jonas Schneider、Rachel Fong、Peter Welinder、Bob McGrew、Josh Tobin、OpenAI 的 Pieter Abbeel 和 Wojciech Zaremba。事后经验回放。神经信息处理系统进展，30，2017年。

Mart载于 Arjovsky、Soumith Chintala 和 L´eon Bottou 的《Wasserstein 生成对抗网络》国际机器学习会议，第 214–223 页。PMLR，2017年。

Fengshuo Bai、Hongming Zhang、Tianyang Tao、Zhiheng Wu、Yanna Wang 和 Bo Xu。PICOR：采用策略校正的多任务深度强化学习。载于*AAAI* 人工智能会议论文集，第 37 卷，第 6728–6736 页，2023年。

Andr´e Barreto、Shaobo Hou、Diana Borsa、David Silver 和 Doina Precup。基于广义策略更新的快速强化学习。美国国家科学院院刊，117 (48):30079–30087，2020年。

Marc G Bellemare、Yavar Naddaf、Joel Veness 和 Michael Bowling。街机学习环境：面向通用智能体的评估平台。《人工智能研究杂志》，47：253–279，2013年。

Glen Berseth、Zhiwei Zhang、Grace Zhang、Chelsea Finn 和 Sergey Levine。CoMPS：持续元策略搜索。国际学习表征会议，2022年。

Massimo Caccia、Jonas Mueller、Taesup Kim、Laurent Charlin 和 Rasool Fakoor。任务无关的持续强化学习：为简单基线正名。*arXiv* 预印本 *arXiv:2205.14495*，2022年。

Rich Caruana。多任务学习。机器学习，28：41–75，1997年。

Yash Chandak、Georgios Theocharous、Shiv Shankar、Martha White、Sridhar Mahadevan 和 Philip Thomas。在非平稳马尔可夫决策过程中面向未来进行优化。见国际机器学习会议，第1414–1425 页。PMLR，2020年。

Feng Chen、Fuguang Han、Cong Guan、Lei Yuan、Zhilong Zhang、Yang Yu 和 Zongzhang Zhang。通过基于扩散的轨迹回放实现稳定的持续强化学习。*ICLR 2024*年决策制定生成模型研讨会，2024年。

Wesley Chung、Lynn Cherif、Doina Precup 和 David Meger。面向持续强化学习的帕塞瓦尔正则化。神经信息处理系统进展，37：127937–127967，2024年。

Coline Devin、Abhishek Gupta、Trevor Darrell、Pieter Abbeel 和 Sergey Levine。学习用于多任务和多机器人迁移的模块化神经网络策略。载于*2017*年*IEEE*国际机器人与自动化会议（*ICRA*），第 2169–2176 页。IEEE，2017年。

Thang Doan、Mehdi Abbana Bennani、Bogdan Mazoure、Guillaume Rabusseau 和 Pierre Alquier。通过 NTK 重叠矩阵对灾难性遗忘进行理论分析。载于人工智能与统计国际会议，第 1072–1080 页。PMLR，2021年。Shibhansh Dohare、J Fernando Hernandez-Garcia、Qingfeng Lan、Parash Rahman、A Rupam Mahmood 和 Richard S Sutton。深度持续学习中的可塑性丧失。《自然》，632(8026)：768–774，2024年。Elizabeth D Dolan 和 Jorge J Mor´e。利用性能剖面对优化软件进行基准测试。数学规划，91：201–213，2002年。Benjamin Eysenbach、Swapnil Asawa、S hreyas Chaudhari、Sergey Levine 和 Ruslan Salakhutdinov。动力学外强化学习：使用领域分类器进行迁移训练。国际学习表征会议（*ICLR*），2021年。Fernando Fern´andez 和 Manuela Veloso。在强化学习智能体中进行概率策略复用。载于第五届自主智能体与多智能体系统国际联合会议论文集，第 720–727 页，2006年。Norman Ferns 和 Doina Precup。双模拟度量是最优价值函数。载于*UAI*，第 210–219 页，2014年。Chelsea Finn 和 Sergey Levine。用于机器人运动规划的深度视觉预见。载于*2017*年*IEEE*国际机器人与自动化会议（*ICRA*），第 2786–2793 页。IEEE，2017年。Chelsea Finn、Pieter Abbeel 和 Sergey Levine。用于深度网络快速适应的模型无关元学习。载于国际机器学习会议，第 1126–1135 页。PMLR，2017年。Jean-Baptiste Gaya、Thang Doan、Lucas Caccia、Laure Soulier、Ludovic Denoyer 和 Roberta Raileanu。构建策略子空间以实现可扩展的持续学习。2023年。Evan Greensmith、Peter L Bartlett 和 Jonathan Baxter。强化学习中梯度估计的方差缩减技术。机器学习研究期刊，5（11月）：1471–1530，2004年。Tuomas Haarnoja、Aurick Zhou、Pieter Abbeel 和 Sergey Levine。软演员-评论家：使用随机演员进行离策略最大熵深度强化学习。载于国际机器学习会议，第 1861–1870 页。PMLR，2018a年。Tuomas Haarnoja、Aurick Zhou、Kristian Hartikainen、George Tucker、Sehoon Ha、Jie Tan、Vikash Kumar、Henry Zhu、Abhishek Gupta、Pieter Abbeel 等。软演员-评论家算法及其应用。*arXiv* 预印本 *arXiv:1812.05905*，2018b年。Karol Hausman、Jost Tobias Springenberg、Ziyu Wang、Nicolas Heess 和 Martin Riedmiller。学习可迁移机器人技能的嵌入空间。载于国际学习表征会议，2018年。Jifeng Hu、Sili Huang、Li Shen、Zhejian Yang、Shengchao Hu、Shisong Tang、Hechang Chen、Yi Chang、Dacheng Tao 和 Lichao Sun。通过在对齐空间中选择性激活权重，应对持续离线强化学习。神经信息处理系统进展，2025年。Arthur Jacot、Franck Gabriel 和 Cl´ement Hongler。神经切线核：神经网络中的收敛与泛化。神经信息处理系统进展，31，2018年。Christos Kaplanis、Murray Shanahan 和 Claudia Clopath。利用复杂突触进行持续强化学习。载于国际机器学习会议，第 2497–2506 页。PMLR，2018年。Christos Kaplanis、Murray Shanahan 和 Claudia Clopath。持续强化学习的策略巩固。国际机器学习会议，2019年。Samuel Kessler、Jack Parker-Holder、Philip Ball、Stefan Zohren 和 Stephen J Roberts。同一状态，不同任务：无干扰的持续强化学习。载于*AAAI*人工智能会议论文集，第 36 卷，第 7143–7151 页，2022年。

Khimya Khetarpal、Matthew Riemer、Irina Rish 和 Doina Precup。《持续强化学习：回顾与展望》。《人工智能研究杂志》，75:1401–1476，2022年。Dharshan Ku maran、Demis Hassabis 和 James L McClelland。智能体需要什么样的学习系统？更新后的互补学习系统理论。《认知科学趋势》，20 (7):512–534，2016年。Zichen Liu、Guoji Fu、Chao Du、Wee Sun Lee 和 Min Lin。通过在线世界模型规划实现持续强化学习。国际机器学习会议，2025年。Mikel Malagon、Josu Ceberio 和 Jose A Lozano。用于可扩展持续强化学习的自组合策略。载于第四十一届国际机器学习会议，2024年。Arun Mallya 和 Svetlana Lazebnik。Packnet：通过迭代剪枝将多个任务纳入单个网络。载于*IEEE*计算机视觉与模式识别会议论文集，第7765–7773页，2018年。Volodymyr Mnih、Koray Kavukcuoglu、David Silver、Andrei A Rusu、Joel Veness、Marc G Belle- mare、Alex Graves、Martin Riedmiller、Andreas K Fidjeland、Georg Ostrovski 等。通过深度强化学习实现人类水平的控制。《自然》，518(7540):529–533，2015年。Aneesh Muppidi、Zhiyu Zhang 和 Heng Yang。Fast trac：用于终身强化学习的无参数优化器。神经信息处理系统进展，37:51169–51195，2024年。Victor M Panaretos 和 Yoav Zemel。Wasserstein距离的统计学方面。《统计学及其应用年度评论》，6(1):405–431，2019年。Emilio Parisotto、Jimmy Lei Ba 和 Ruslan Salakhutdinov。Actor-mimic：深度多任务与迁移强化学习。国际学习表征会议，2016年。Janarthanan Rajendran、Aravind Srinivas、Mitesh M Khapra、P Prasanna 和 Balaraman Ravindran。关注、适应与迁移：用于从同一领域多个来源进行自适应迁移的注意力深度架构。*arXiv*预印本 *arXiv:1510.02879*，2017年。Aaditya Ramdas、Peter Gr¨unwald、Vladimir Vovk 和 Glenn Shafer。博弈论统计学与安全的任意时刻有效推断。《统计科学》，38(4):576–601，2023年。Andrei A Rusu、Sergio Gomez Colmenarejo、Caglar Gulcehre、Guillaume Desjardins、James Kirk- patrick、Razvan Pascanu、Volodymyr Mnih、Koray Kavukcuoglu 和 Raia Hadsell。策略蒸馏。*arXiv*预印本 *arXiv:1511.06295*，2015年。Andrei A Rusu、Neil C Rabinowitz、Guillaume Desjardins、Hubert Soyer、James Kirkpatrick、Koray Kavukcuoglu、Razvan Pascanu 和 Raia Hadsell。渐进式神经网络。*arXiv*预印本 *arXiv:1606.04671*，2016年。Tom Schaul、Daniel Horgan、Karol Gregor 和 David Silver。通用价值函数逼近器。载于国际机器学习会议，第1312–1320页。PMLR，2015年。John Schulman、Sergey Levine、Pieter Abbeel、Michael Jordan 和 Philipp Moritz。信赖域策略优化。载于国际机器学习会议，第1889–1897页。PMLR，2015年。John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。近端策略优化算法。*arXiv*预印本 *arXiv:1707.06347*，2017年。Shagun Sodhani、Amy Zhang 和 Joelle Pineau。基于上下文表征的多任务强化学习。载于国际机器学习会议，第9767–9779页。PMLR，2021年。Yixiang Sun、Haotian Fu、Michael Littman 和 George Konidaris。用于持续基于模型的强化学习的知识保持。国际机器学习会议，2025年。

Richard S Sutton 和 Andrew G Barto。强化学习：导论。麻省理工学院出版社，2018年。

Hongyao Tang、Johan Obando-Ceron、Pablo Samuel Castro、Aaron Courville 和 Glen Berseth。通过降低变化率缓解持续强化学习中的可塑性丧失。国际机器学习会议，2025年。

Yunzhe Tao、Sahika Genc、Jonathan Chung、Tao Sun 和 Sunil Mallya。Repaint：深度强化学习中的知识迁移。载于国际机器学习会议，第10141–10152页。PMLR，2021年。

Yee Teh、Victor Bapst、Wojciech M Czarnecki、John Quan、James Kirkpatrick、Raia Hadsell、Nicolas Heess 和 Razvan Pascanu。Distral：稳健的多任务强化学习。神经信息处理系统进展，30，2017年。

Andrea Tirinzoni、Andrea Sessa、Matteo Pirotta 和 Marcello Restelli。强化学习中样本的重要性加权迁移。载于国际机器学习会议，第4936–4945页。PMLR，2018年。

Yi Wan、Ali Rahimi-Kalahroudi、Janarthanan Rajendran、Ida Momennejad、Sarath Chandar 和 Harm H Van Seijen。迈向评估基于模型的强化学习方法的适应性。见国际机器学习会议，第22536–22561页。PMLR，2022年。

Maciej Wolczyk、Michał Zajkac、Razvan Pascanu、Łukasz Kuci´nski 和 Piotr Miło´s。Continual World：面向持续强化学习的机器人基准测试。神经信息处理系统进展，34:28496–28510，2021年。

Maciej Wolczyk、Michal Zajac、Razvan Pascanu、Lukasz Kucinski 和 Piotr Milos。厘清持续强化学习中的迁移。神经信息处理系统进展，35:6304–6317，2022年。

Annie Xie 和 Chelsea Finn。通过保留经验实现终身机器人强化学习。载于终身学习智能体会议，第 838–855 页。PMLR，2022年。

Yijun Yang、Tianyi Zhou、Jing Jiang、Guodong Long 和 Yuhui Shi。通过稀疏提示在元策略网络中实现持续任务分配。2023年。

Kenny Young 和 Tian Tian。MinAtar：一个受 Atari 启发的测试平台，用于全面且可复现的强化学习实验。*arXiv* 预印本 *arXiv:1903.03176*，2019年。

Tianhe Yu、Deirdre Quillen、Zhanpeng He、Ryan Julian、Karol Hausman、Chelsea Finn 和 Sergey Levine。Meta-World：多任务和元强化学习的基准与评估。载于机器人学习会议，第 1094–1100 页。PMLR，2020年。

Hongming Zhang、于天扬和黄瑞通。将深度 Q 网络与演员-评论家相结合。载于《深度强化学习：基础、研究与应用》，第 213–245 页。施普林格，2020年。

Hongming Zhang、任同正、肖晨君、戴尔·舒尔曼斯和戴博。面向部分可观测强化学习的可证明表示与高效规划。载于第*41*届国际机器学习大会论文集，第59759–59782页，2024年。

Hongming Zhang、肖晨君、Chao Gao、王涵、Bo Xu和Martin M¨uller。先利用回放记忆，再探索环境：通过经验马尔可夫决策过程迭代增强强化学习。神经信息处理系统进展，37：85658–85692，2024年。

张天天、沈泽华、林子川、袁博、王学谦、修丽和叶德恒。回放增强的持续强化学习。机器学习研究汇刊，2023年。

附录

目录

A 相关工作 16 B 预备知识：深度学习中分布漂移与灾难性遗忘的定义 17 C 理论结果 17C.1 命题 3 中的基于 Q 值的元学习器增量更新 . . . . . 17 C.2 命题 1 的证明.19 C.3 命题 2 的证明.19**D 基于策略的 FAME 算法 22 E FAME 框架讨论 23 F 实验：离散动作空间下基于价值的持续强化学习 24**F.1 实验设置与对比方法的细节.24 F.2 MINATAR 消融研究.25 F.3 可塑性能力的学习曲线分析.27**G 实验：连续动作空间下基于策略的持续强化学习 29**G.1 实验设置与对比方法的细节.29 G.2 更多实验结果与细节.30**H 大语言模型（LLM）的使用 30 I 补充结果 33**I.1 Meta-World 中三个任务序列上的更多基线方法 . . . . . 33 I.2 附加任务序列：CW10.36

相关工作

MDP 距离的度量与灾难性遗忘。 尽管视角不同，双模拟度量（Ferns & Precup, 2014）提供了更多证据，支持这样一种策略：最优价值函数可以自然地用于衡量状态相似性。

持续强化学习。 持续强化学习（Khetarpal 等人，2022年；Abel 等人，2023年）旨在应对学习一系列决策任务所面临的挑战，尤其是如何平衡稳定性（即缓解灾难性遗忘）与可塑性（即快速适应新环境）。下文根据研究重点将现有文献分为两类，并详细比较这些工作与本文的贡献。

• **灾难性遗忘**。 **(1)** 基于回放的方法常用于缓解遗忘。最初，有研究为任务无关智能体提出了基于回放的循环方法（Caccia 等人，2022年）。RECALL（Zhang 等人，2023年）利用自适应归一化处理近似目标，并通过对旧任务进行策略蒸馏来增强泛化能力和稳定性。近期提出的生成式回放（Chen 等人，2024年）使用扩散模型，记忆每个已遇任务的高回报轨迹分布。Sun 等人（2025年）在基于模型的强化学习框架中持续学习生成模型，以便在不存储过去数据的情况下进行经验回放。同样，Liu 等人（2025年）学习在线世界模型，并通过模型预测控制进行规划和行动，以构建统一的世界动力学，从而解决灾难性遗忘问题。 **(2)** 第二类是从不同角度出发的基于正则化的方法。较早的工作在历史策略上研究了行为克隆（Wolczyk 等人，2022年）。受复杂突触的启发，Kaplanis 等人（2018年；2019年）提出策略巩固策略，在记住智能体不同时间尺度上的策略的同时，利用其自身的历史经验对当前策略进行正则化。稀疏提示（Yang 等人，2023年）通过字典学习施加正则化，生成作为提示的稀疏掩码，并从元策略网络中为每项任务提取一个子网络。 **(3)** 模型扩展也是一个值得探索的方向（Mallya 和 Lazebnik, 2018年；Malagon 等人，2024年；Gaya 等人，2023年）。Gaya 等人（2023年）构建策略子空间以实现可扩展的持续强化学习，同时指出智能体规模与持续学习性能之间存在权衡。Malagon 等人（2024年）采用不断扩大的策略神经网络，并利用注意力机制整合先前策略和当前状态中的知识，从而“自组合”出一个内部策略。尽管基于模型扩展的方法效果显著，但由于依赖先前策略并采用特定的聚合策略，其主要问题在于较高的内存和推理成本。例如，Malagon 等人（2024年）采用了注意力模块，其参数数量随任务数量线性增长，而理论计算成本则呈二次增长 。**比较：** 我们的 FAME 方法采用双学习器策略，可视为处于三种主流范式的交汇处。在知识整合方面，缓解灾难性遗忘需要从元缓冲区进行经验回放。在知识迁移方面，如果在自适应元预热框架中选择元学习器作为最佳学习器，则可以采用行为克隆正则化来引导探索。尽管采用双学习器策略，FAME 仍使快速学习器和元学习器的模型规模保持固定，以避免稳定性问题，这一点有别于其他模型扩展方法。

• **知识迁移与可塑性丧失**。知识前向迁移的有效性决定了可塑性丧失的程度；可塑性丧失是指快速适应新环境的能力下降（Abel 等人，2018；Dohare 等人，2024；Wolczyk 等人，2022；Tao 等人，2021）。在持续强化学习文献中，大量研究从不同角度主要关注知识迁移和可塑性能力。Xie 和 Finn（2022）通过可学习的重要性权重，有选择地识别与新任务最相关的样本，从而改善前向迁移并减轻可塑性丧失。价值函数分解方法（Anand & Precup, 2023）旨在离散动作空间下的基于价值的持续强化学习中，在不同层级实现快慢学习的相互作用，以应对可塑性丧失。从优化的角度来看，Parseval 网络（Chung 等人，2024）施加正交约束以减轻干扰，而 Muppidi 等人（2024）采用无参数的在线凸优化来保持可塑性。Tang 等人（2025）近期通过神经切线核（NTK）矩阵建立了可塑性与变化率之间的联系。Ahn 等人（2025）发现，由于任务差异性，适应新任务时会出现负迁移问题，从而加剧可塑性丧失。因此，重置与蒸馏（R&D）（Ahn 等人，2025）明确

会在每个新环境中重置策略，以避免负迁移。**比较：**就知识迁移而言，重置并非总是最佳选择，尤其是在任务往往具有相似性的情况下。通过利用一对多假设检验，我们提出了自适应元预热方法，有选择地甄别最有效的权重初始化和预热策略，以促进知识迁移并减少可塑性丧失。

迁移、多任务与元强化学习。 我们方法中的知识迁移阶段与迁移强化学习密切相关，其中积累的知识可以通过表示（Rusu 等，2016；Devin 等，2017）、学习到的模型（Finn & Levine，2017；Eysenbach 等人，2021）、网络权重（Fern´andez & Veloso，2006）或经验（Andrychowicz 等人，2017；Tirinzoni 等人，2018；Xie & Finn，2022）进行迁移。尽管任务集合应当同时进行学习，多任务强化学习同样整合了知识迁移这一组成部分，例如（Caruana，1997年；Sodhani 等人，2021年；Teh 等人，2017年；Yang 等人，2023年；Rusu 等人，2015；Hausman 等人，2018）。因此，迁移强化学习一直是多任务强化学习和持续强化学习的基础构件，例如我们双学习器算法中的显式知识迁移。与多任务强化学习不同，持续强化学习具有增量式或序列式学习特性，因此更难以最小化灾难性遗忘。然而，我们的研究表明，从原理上讲，最小化灾难性遗忘可以等价于多任务学习中的目标。元强化学习（Finn 等人，2017）可以被视为多任务强化学习的扩展或泛化，其中包含快速适应和少样本学习的显式机制，并且也与持续强化学习密切相关。持续元策略搜索（CoMPS）（Berseth 等人，2022年）以增量方式探索元训练设置，将元强化学习扩展至持续学习场景。**比较：**我们 FAME 方法所采用的快速学习器和元学习器交织融合了迁移、多任务和元强化学习的关键技术，阐明了它们在我们算法框架中的深层联系。通过快速学习器与元学习器之间的耦合更新，知识迁移与知识整合的相互作用同时应对了其中的挑战，为解决持续强化学习问题展现出颇具前景的方案。

B 预备知识：深度学习中分布漂移与灾难性遗忘的定义

我们首先介绍在学习参数化函数 f 的过程中，从源数据分布 τ_S 及数据集 $\mathcal{D}_{\tau_S}$ 到目标数据分布 τ_T 及数据集$\mathcal{D}_{\tau_T}$时的漂移 概念。在源数据集 $\mathcal{D}_{\tau_S}$ 上学习 f 之后，我们得到估计函数 f_{τ_S}。然后，我们使用任意学习算法将相同的模型架构 f 应用于目标数据集 $\mathcal{D}_{\tau_T}$，并且最后通过 $\delta^{\tau_S\to\tau_T}$ 评估所得 f_{τ_T} 的漂移，其定义如下（Doan等人，2021）：

$$\delta^{\tau_S\to\tau_T}\left(X^{\tau_S}\right)=\left(\widehat{f}_{\tau_T}(x)-\widehat{f}_{\tau_S}(x)\right)_{(x,y)\in\mathcal{D}_{\tau_S}} \tag{10}$$

基于漂移的定义，我们将普通灾难性遗忘 $\Delta^{\tau_S\to\tau_T}$ 定义为

$$\Delta^{\tau_S\to\tau_T}\left(X^{\tau_S}\right)=\left\|\delta^{\tau_S\to\tau_T}\left(X^{\tau_S}\right)\right\|_2^2=\sum_{(x,y)\in\mathcal{D}_{\tau_S}}\left(\widehat{f}_{\tau_T}(x)-\widehat{f}_{\tau_S}(x)\right)^2, \tag{11}$$

其中，灾难性遗忘在神经切线核（NTK）机制下可进一步简化为 $\Delta^{\tau_S\to\tau_T}=\left\|\phi\left(X^{\tau_S}\right)\left(\omega_{\tau_T}^*-\omega_{\tau_S}^*\right)\right\|_2^2$（Doan等人，2021；Jacot 等人，2018年），从而能够提出新的持续学习方法。在深度学习中，最小化灾难性遗忘 $\Delta^{\tau_S\to\tau_T}$ 等价于就预测函数 f而言最小化加权漂移，其权重由数据集决定。深度学习中的初步结果在很大程度上启发我们探索持续强化学习的基础。

C 理论结果

C.1 命题 3 中基于增量 Q 值的元学习器更新

在命题 3 中，我们给出了元学习器的一种高效增量更新规则，以尽量减少我们在定义 2 中定义的基于 Q 值的有原则灾难性遗忘。

命题 3 (基于 Q 值的元学习器增量更新)。令 d_Q 为定义 *2* 中公式 *1* 的 ℓ_2 损失。最小化公式 *3* 中的基于 *Q* 值的灾难性遗忘等价于:

$$Q_k^M = \arg\min_{\widetilde{Q}_k^M} \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\left(Q_{k-1}^M - \widetilde{Q}_k^M \right)^2 \right] + \mathbb{E}_{w_k^Q} \left[\left(Q_k - \widetilde{Q}_k^M \right)^2 \right]. \tag{12}$$

证明。 **步骤 1：最优性条件**。 回顾 $w_i^Q(s,a) = \mu_i^{Q_i}(s)\pi^{Q_i}(a|s)$。我们的目标是最小化式 3 中定义的基于 Q 值的灾难性遗忘:

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

对于每个 s 和 a，对式 3 中的目标函数关于 $\widetilde{Q}_k^M$ 求导，一阶最优性条件为

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

通过改写关于 Q_k^M和 Q_{k-1}^M的两个最优性条件，可得

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

步骤 2：增量更新规则。 为简洁起见，我们采用期望运算 $\mathbb{E}_w$。根据上述两个最优性条件，可以推导出以下增量更新规则:

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

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

其中，(a) 满足条件 $\sum_{i=1}^{k-1} w_i^Q(s,a) \left(Q_i(s,a) - Q_{k-1}^M(s,a) \right) = 0$，这是步骤 1 的 Eq. 13 中关于 Q_{k-1}^M 的最优性条件。 $C = \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q} \left[\left(Q_i - Q_{k-1}^M \right)^2 \right]$ 是关于 $\widetilde{Q}_k^M$ 的常数。将所有内容代入 Eq. 14，可得

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

□

C.2 命题 1 的证明

命题 1 [基于增量 Softmax Q 值的元学习器更新] 令 $\pi_k^M(a|s) = \exp\left(Q_k^M(a|s)/\tau\right) / \sum_{a'} \exp\left(Q_k^M(a'|s)/\tau\right)$。经过 Softmax 策略变换后，基于 Q 值的元学习器的增量更新可改写为

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

证明。 我们依靠 Softmax 变换将元 Q 函数迁移为元策略。因此，将基于价值的持续强化学习中的基于策略的灾难性遗忘适配到式 4，并以 KL 散度作为 d_π，可表示为

$$Q_k^M = \arg\min_{\widetilde{Q}_k^M} \sum_{i=1}^{k} \sum_{s} \mu_i^{Q_i}(s)\left[\mathrm{KL}\left(\pi^{Q_i}(\cdot|s)||\widetilde{\pi}_k^M(\cdot|s)\right)\right]. \tag{15}$$

其中 $\widetilde{\pi}_k^M(a|s) = \exp\left(\widetilde{Q}_k^M(a|s)/\tau\right) / \sum_{a'} \exp\left(\widetilde{Q}_k^M(a'|s)/\tau\right)$。根据 KL 散度的定义，我们可以将式 15 中的目标函数改写为增量更新规则：

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

其中 $C = \sum_{i=1}^{k-1} \mathbb{E}_{w_i^Q}\left[\log \frac{\pi^{Q_i}}{\pi_{k-1}^M}\right]$ 是一个常数，且与 $\widetilde{Q}_k^M$ 无关。尽管保留式 16 的形式可能显得多余，但它强调了一个基于 $Q_{k-1}^M(\pi_{k-1}^M)$ 和 $Q_k(\pi^{Q_k})$ 的 Q_k^M 增量更新规则。最终，这一最小化过程使元学习器 $\widetilde{Q}_k^M$ 在式 17 中对截至 k 已遇到的所有环境的状态-动作分布混合进行最大似然估计。

□

C.3 命题 2 的证明

命题 2 [Wasserstein 距离下基于策略的元学习器增量更新] 设 d_π 为定义 2 的式 2 中的平方 2-Wasserstein 距离，且策略表示为关于动作 a的独立（多元）高斯分布。最小化基于策略的

式 4 中的灾难性遗忘等价于:

$$\pi_{\mathrm{M}}^{k}=\underset{\widetilde{\pi}_{k}^{M}}{\arg\min}\left\{\sum_{i=1}^{k-1}\sum_{s}\mu_{i}^{\pi_{i}}(s)W_{2}^{2}\left(\widetilde{\pi}_{k}^{M}(\cdot|s),\pi_{k-1}^{M}(\cdot|s)\right)+\sum_{s}\mu_{k}^{\pi_{k}}(s)W_{2}^{2}\left(\widetilde{\pi}_{k}^{M}(\cdot|s),\pi_{k}(\cdot|s)\right)\right\}.$$

证明。 根据平方 2-Wasserstein 距离，回顾式 4 中基于策略的灾难性遗忘目标:

$$\pi_{k}^{M}=\underset{\widetilde{\pi}_{k}^{M}}{\arg\min}\sum_{i=1}^{k}\sum_{s}\mu_{i}^{\pi_{i}}(s)W_{2}^{2}\left(\pi_{i}(\cdot|s),\widetilde{\pi}_{k}^{M}(\cdot|s)\right).$$

两个高斯分布 p和 q 之间的平方 2-Wasserstein 距离有闭式解:

$$W_{2}^{2}(p,q)=\|\nu_{p}-\nu_{q}\|_{2}^{2}+\mathrm{tr}\left(\Sigma_{p}+\Sigma_{q}-2\left(\Sigma_{q}^{1/2}\Sigma_{p}\Sigma_{q}^{1/2}\right)^{1/2}\right),\tag{18}$$

其中，两个高斯分布分别表示为 $\mathcal{N}(\nu_p,\ \Sigma_p)$ 和 $\mathcal{N}(\nu_q,\ \Sigma_q)$。具体而言，当策略表示为在动作 a上相互独立的（多元）高斯分布时，这意味着 Σ_p 和 Σ_q 是对角的（即变量相互独立），那么式 18 中的平方 2-Wasserstein 距离可进一步简化为

$$W_{2}^{2}(p,q)=\|\nu_{p}-\nu_{q}\|_{2}^{2}+\|\sigma_{p}-\sigma_{q}\|_{2}^{2},\tag{19}$$

其中， σ_p 和 σ_q 分别是 Σ_p 和 Σ_q的对角向量。于是，根据式 4，基于策略的灾难性遗忘目标可简化为

$$\pi_{k}^{M}=\underset{\widetilde{\nu}_{k}^{M},\widetilde{\sigma}_{k}^{M}}{\arg\min}\sum_{i=1}^{k}\sum_{s}\mu_{i}^{\pi_{i}}(s)\left(\|\nu_{i}(s)-\widetilde{\nu}_{k}^{M}(s)\|_{2}^{2}+\|\sigma_{i}(s)-\widetilde{\sigma}_{k}^{M}(s)\|_{2}^{2}\right),\tag{20}$$

其中， π_i（$\cdot|s$）表示为一个（多元）高斯分布 $\mathcal{N}(\nu_i(s)\ \sigma_i^2(s))$，其中 ν_i（s）和 σ_i^2（s）分别是均值（向量）和方差（的对角向量）。类似地， π_M^k 表示为一个（多元）高斯分布 $\mathcal{N}(\nu_k^M(s),\ (\sigma_k^M(s))^2)$。

步骤 1：最优性条件。对于每个 s，我们分别对关于 $\widetilde{\nu}_k^M$和$\widetilde{\sigma}_k^M$的式 20 求导。因此，得到以下最优性条件:

$$\sum_{i=1}^{k}\mu_{i}^{\pi_{i}}(s)\left(\nu_{i}(s)-\nu_{k}^{M}(s)\right)=0\tag{21}$$

$$\sum_{i=1}^{k}\mu_{i}^{\pi_{i}}(s)\left(\sigma_{i}(s)-\sigma_{k}^{M}(s)\right)=0.\tag{22}$$

步骤 2：增量更新。我们首先将公式 20 改写为

$$\pi_{k}^{M}=\underset{\widetilde{\nu}_{k}^{M},\widetilde{\sigma}_{k}^{M}}{\arg\min}\underbrace{\sum_{i=1}^{k-1}\sum_{s}\mu_{i}^{\pi_{i}}(s)\|\nu_{i}(s)-\widetilde{\nu}_{k}^{M}(s)\|_{2}^{2}}_{①}+\sum_{s}\mu_{k}^{\pi_{k}}(s)\|\nu_{k}(s)-\widetilde{\nu}_{k}^{M}(s)\|_{2}^{2}$$
$$+\underbrace{\sum_{i=1}^{k-1}\sum_{s}\mu_{i}^{\pi_{i}}(s)\|\sigma_{i}(s)-\widetilde{\sigma}_{k}^{M}(s)\|_{2}^{2}}_{②}+\sum_{s}\mu_{k}^{\pi_{k}}(s)\|\sigma_{k}(s)-\widetilde{\sigma}_{k}^{M}(s)\|_{2}^{2}.$$

$$
\begin{aligned}
① &= \sum_{i=1}^{k-1}\sum_{s}\mu_i^{\pi_i}(s)\|\nu_i(s)-\nu_{k-1}^M(s)+\nu_{k-1}^M(s)-\widetilde{\nu}_k^M(s)\|_2^2 \\
&= \sum_{i=1}^{k-1}\sum_{s}\mu_i^{\pi_i}(s)\left(\|\nu_i(s)-\nu_{k-1}^M(s)\|_2^2+\|\nu_{k-1}^M(s)-\widetilde{\nu}_k^M(s)\|_2^2\right)+ \\
&\qquad 2\sum_{i=1}^{k-1}\sum_{s}\mu_i^{\pi_i}(s)\langle\nu_i(s)-\nu_{k-1}^M(s),\nu_{k-1}^M(s)-\widetilde{\nu}_k^M(s)\rangle \\
&= \sum_{i=1}^{k-1}\sum_{s}\mu_i^{\pi_i}(s)\left(\|\nu_i(s)-\nu_{k-1}^M(s)\|_2^2+\|\nu_{k-1}^M(s)-\widetilde{\nu}_k^M(s)\|_2^2\right)+ \\
&\qquad 2\sum_{s}\langle\sum_{i=1}^{k-1}\mu_i^{\pi_i}(s)\left(\nu_i(s)-\nu_{k-1}^M(s)\right),\nu_{k-1}^M(s)-\widetilde{\nu}_k^M(s)\rangle \\
&\overset{(a)}{=} \sum_{i=1}^{k-1}\sum_{s}\mu_i^{\pi_i}(s)\left(\|\nu_i(s)-\nu_{k-1}^M(s)\|_2^2+\|\nu_{k-1}^M(s)-\widetilde{\nu}_k^M(s)\|_2^2\right),
\end{aligned}
$$

其中，(a) 成立，这是由于我们在步骤 1 的公式 21 中推导出的最优性条件 $\sum_{i=1}^{k-1}\mu_i^{\pi_i}(s)\left(\nu_i(s)-\nu_{k-1}^M(s)\right)=0$。类似地，我们可以对方差作如下简化：

$$
\begin{aligned}
② &= \sum_{i=1}^{k-1}\sum_{s}\sigma_i^{\pi_i}(s)\|\sigma_i(s)-\sigma_{k-1}^M(s)+\sigma_{k-1}^M(s)-\widetilde{\sigma}_k^M(s)\|_2^2 \\
&= \sum_{i=1}^{k-1}\sum_{s}\mu_i^{\pi_i}(s)\left(\|\sigma_i(s)-\sigma_{k-1}^M(s)\|_2^2+\|\sigma_{k-1}^M(s)-\widetilde{\sigma}_k^M(s)\|_2^2\right)+ \\
&\qquad 2\sum_{i=1}^{k-1}\sum_{s}\mu_i^{\pi_i}(s)\langle\sigma_i(s)-\sigma_{k-1}^M(s),\sigma_{k-1}^M(s)-\widetilde{\sigma}_k^M(s)\rangle \\
&= \sum_{i=1}^{k-1}\sum_{s}\mu_i^{\pi_i}(s)\left(\|\sigma_i(s)-\sigma_{k-1}^M(s)\|_2^2+\|\sigma_{k-1}^M(s)-\widetilde{\sigma}_k^M(s)\|_2^2\right)+ \\
&\qquad 2\sum_{s}\langle\sum_{i=1}^{k-1}\mu_i^{\pi_i}(s)\left(\sigma_i(s)-\sigma_{k-1}^M(s)\right),\sigma_{k-1}^M(s)-\widetilde{\sigma}_k^M(s)\rangle \\
&\overset{(b)}{=} \sum_{i=1}^{k-1}\sum_{s}\mu_i^{\pi_i}(s)\left(\|\sigma_i(s)-\sigma_{k-1}^M(s)\|_2^2+\|\sigma_{k-1}^M(s)-\widetilde{\sigma}_k^M(s)\|_2^2\right),
\end{aligned}
$$

其中，(b) 成立，这是由于我们在步骤 1 的式（22）中推导出的最优性条件 $\sum_{i=1}^{k-1}\mu_i^{\pi_i}(s)\left(\sigma_i(s)-\sigma_{k-1}^M(s)\right)=0$。综上可得

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

去除与 $\widetilde{\nu}_k^M$ 和 $\widetilde{\sigma}_k^M$ 无关的常数项后，我们进一步得到

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

由此，在平方 2-Wasserstein 距离下进行增量式基于策略的元学习器更新。

□

D 基于策略的 FAME 算法

算法描述。我们首先将快速缓冲区记为 $\mathcal{F}$，并将 π^0 作为初始化策略。如算法 1 所示，当第 k 个环境到来时，我们在 L 步内，通过在先前的元学习器 π_{k-1}^M、先前的快速学习器 π_{k-1} 和随机学习器 π^0（重置策略）之间进行自适应元预热，来初始化快速学习器 π_k。自适应元预热充分利用先前的信息，以实现有效的知识迁移。当第 k 个任务结束后，知识整合阶段随即开始，此时根据式 7（FAME-KL）或式 8（FAME-MD），利用元缓冲区 $\mathcal{M}$ 中收集的数据更新元学习器 π_k^M。元学习器将来自 π_k 的知识逐步整合到 π_{k-1}^M 中，从而得到更新后的元学习器 π_k^M。

算法 2 第 k个环境中的基于策略的 FAME 更新

1：初始化：快速缓冲区 $\mathcal{F}$、元缓冲区 $\mathcal{M}$、 π_{k-1}^M、 π_{k-1}、 π^0、预热步数 L、估计步数N。2：# 知识迁移：自适应元预热 3：在经过 L 步后，通过公式 6 在 $\{\pi_{k-1}, \pi_k^M, \pi^0\}$ 中初始化 π_k 4：对于 $t = L$到 T，执行 5：观测 S_t，采取动作 A_t，接收 R_t，观测 S_{t+1} 6：将 (S_t, A_t, R_t, S_{t+1})存入 $\mathcal{F}$ 7：更新 π_k 8：如果 $t > T - N$，则 9：方法 1（FAME-KL）：将(S_t, A_t)存入 $\mathcal{M}$ # 用于估计 w_k 10：方法 2（FAME-WD）：将 S_t 存入 $\mathcal{M}$ # 用于估计 $\mu_k^{\pi_k}$ 11：结束 if 12：结束 for 13：重置 $\mathcal{F}$ 14：# 知识整合：尽量减少灾难性遗忘 15：方法 1（FAME-KL）：根据公式 7，使用 $\mathcal{M}$中的状态-动作对更新 π_k^M 16：方法 2（FAME-WD）：根据公式 8，使用 $\mathcal{M}$中的状态更新 π_k^M

计算成本、性能变化与实践指南。使用标准高斯策略参数化时，FAME-WD 增加的计算开销与 FAME-KL 相比微不足道；这种参数化在连续动作空间的强化学习中很常见。如公式 19 所示，Wasserstein 距离的简化形式会产生与策略网络相同的输出，因此其计算成本与 KL 散度相当。我们还要指出，尽管 Wasserstein 距离更能捕捉数据几何结构，但高斯策略可能会限制 Wasserstein 距离的优势。因此，FAME-WD 不一定优于 FAME-KL。在复杂环境中，任务涉及显著的分布偏移以及随机性的变化，此时捕捉分布几何结构至关重要。我们认为，这正是 FAME-WD 可能优于 FAME-KL 的情形。

E FAME 框架讨论

在元缓冲区中存储过去轨迹的可扩展性。如基于价值的 FAME 算法所述，在每个环境中的训练结束时，我们会存储一部分状态–动作对，以估计权重 w_k。值得注意的是，所存储的状态–动作对仅占每个任务训练数据总量的一小部分，在我们的实验中约为 1–2%。因此，在所考察的基准测试中，我们的 FAME 方法几乎不会带来可扩展性开销。为确保公平比较，我们将元缓冲区维持在与标准强化学习所用回放缓冲区相同的大小，而我们的 FAME 方法在所有考察的基准测试中仍表现出更优性能。不过，如附录 F.2 的表 6 所示，扩大元缓冲区、纳入更多过去轨迹通常会提升性能。此外，元学习更新采用监督学习的方式，即使训练集在元缓冲区中更大，其成本仍远低于在线强化学习训练。总之，FAME 以极低的可扩展性顾虑实现了强劲性能，凸显了其有效应对持续强化学习关键挑战的潜力。

确定策略的原则 L 用于策略评估。关于预热步数 L 的选择，最优的 L 确实取决于任务。由于在总交互步数固定的情况下，增加 L 会减少在线学习可用的样本数，因此最优的 L 需要在选择最有效的先验知识与保留充足数据用于在线适应之间取得平衡。不过，在 Meta-World 实验中，预热评估被简单地固定为 10 个回合，因为这一选择已能带来良好性能，并显著优于其他基线方法。在基于价值的持续强化学习中，延长行为克隆正则化中的预热时长 L 并不一定会提升整体学习效果。当通过一对多假设检验选择元学习器来预热快速学习器时，结论是，元学习器提供的先验知识比随机学习器或先前的快速学习器能提供更好的初始化，并且能够加快训练过程，尤其是在训练早期。然而，随着学习的推进，快速学习器会逐渐适应新环境，其性能也可能超过元学习器。在这种情况下，持续进行行为克隆正则化往往会对快速学习器造成过度约束，从而阻碍适应并导致最终性能下降（例如，表 5 中的非单调效应）。

确定 N 的原则性策略。关于存储的轨迹数量 N，其选择主要取决于计算和内存约束（即元缓冲区的大小）。如附录 F.2 的表 6 所示，增加 N 通常能提升性能，但也会带来更高的内存成本（增大元缓冲区）和计算成本（在知识整合阶段，元学习器需要在更大的轨迹数据集上进行更新）。然而，如前文所述，FAME 的性能出色，且几乎不受可扩展性问题影响；只要内存预算允许，我们便可在实践中增加 N，以追求更优的性能。

FAME 的空间复杂度和计算复杂度。(1) 在空间复杂度方面，FAME 的双学习器系统需要额外存储一份快速学习器（Reset 和 Finetune 等基线方法中的普通学习器）的副本，并额外配置一个与快速学习器的缓冲区大小相同的元回放缓冲区。由于总样本内存固定，且不受未来任务数量的影响，因此这种设计具有可扩展性。(2) 在计算成本方面，为了公平比较，我们让智能体与环境进行相同数量的交互步数。换言之，策略评估的代价是减少策略优化的总更新步数。主要的额外计算成本来自元学习器的更新，而这一过程通过监督学习的方式高效完成。例如，我们发现，在 MinAtar 中，一个任务结束后，更新元学习器在 200 个轮次中仅需几分钟。这一额外计算开销相较于强化学习算法在线训练的总体计算成本微不足道。

F 实验：具有离散动作空间的基于价值的持续强化学习

F.1 实验设置与对比方法的详细信息

指标计算。由于平均性能是持续强化学习中的主要指标，我们报告了各个环境的结果。为了评估遗忘，我们首先计算每个环境的指标分数，然后根据各方法在该环境中的标准差对其进行标准化。这种标准化减轻了不同游戏奖励尺度的影响，并使我们能够对各游戏的结果取平均值，从而报告更全面的遗忘分数。

超参数：MinAtar。我们的实现改编自（Anand & Precup, 2023）发布的代码。对于 FAME 方法，在第 F.2 节对超参数进行线搜索后，我们选择估计步数 $N = 12000$，即每个任务中存储在大小为 100000 的元缓冲区中的数据量 $\mathcal{M}$；选择用于一对多假设检验的策略评估步数 $n = 600$；以及预热步数 $L = 50000$（每个任务训练步数的 10%）。请注意，我们使用L 表示采用行为克隆正则化进行学习时的预热步数，同时引入另一种记号 n表示自适应预热的策略评估步数。每个任务的整个训练过程中，存储数据的比例为 $12000/500\text{k} = 2.4\%$。在知识整合阶段，我们以 1×10^{-3} 的学习率，采用衰减策略训练元学习器 200 轮。快速学习器的学习率保持为 1×10^{-5}，与其他 DQN 基线方法相同。每当有新环境到来时，我们都会清空快速缓冲区 $\mathcal{F}$，并重新初始化所有相关学习器的参数，DQN-微调除外。对于 FAME，在自适应元预热之后，我们可以自动选择随机初始化，或从前一个快速学习器初始化；还可以选择是否添加行为克隆正则化项作为示范。对于所有任务，我们均选择命题 1 中的 $\tau = 1$。

任务序列：MinAtar。我们随机选取 10 个环境序列，并将其固定以确保可复现性。每个任务序列运行 3 个随机种子。

1. [‘Breakout’, ‘Space Invaders’, ‘Breakout’, ‘Space Invaders’, ‘Space Invaders’, ‘Freeway’, ‘Breakout’]2. [‘Space Invaders’, ‘Breakout’, ‘Breakout’, ‘Space Invaders’, ‘Space Invaders’, ‘Breakout’, ‘Breakout’]3. [‘Breakout’, ‘Space Invaders’, ‘Breakout’, ‘Freeway’, ‘Freeway’, ‘Breakout’, ‘Freeway’]4. [‘Freeway’, ‘Breakout’, ‘Space Invaders’, ‘Breakout’, ‘Breakout’, ‘Breakout’, ‘Space Invaders’]5. [‘Freeway’, ‘Freeway’, ‘Space Invaders’, ‘Space Invaders’, ‘Breakout’, ‘Breakout’, ‘Freeway’]6. [‘Freeway’, ‘Space Invaders’, ‘Freeway’, ‘Freeway’, ‘Breakout’, ‘Space Invaders’, ‘Breakout’]7. [‘Freeway’, ‘Space Invaders’, ‘Breakout’, ‘Freeway’, ‘Space Invaders’, ‘Freeway’, ‘Breakout’]8. [‘Breakout’, ‘Space Invaders’, ‘Freeway’, ‘Breakout’, ‘Space Invaders’, ‘Freeway’, ‘Breakout’]9. [‘Breakout’, ‘Space Invaders’, ‘Space Invaders’, ‘Space Invaders’, ‘Freeway’, ‘Breakout’, ‘Breakout’]10. [‘Freeway’, ‘Breakout’, ‘Freeway’, ‘Space Invaders’, ‘Freeway’, ‘Breakout’, ‘Freeway’]

备注：重复任务的选择。任务的随机排列顺序对于阐明我们的自适应元预热策略的机制尤其有用。在实际应用中，任务并非严格地只出现一次；新到达的任务具有不确定性，可能是完全未知的任务，也可能是再次遇到的、具有相似结构的任务——尤其是在动态、周期性或季节性场景中。例如，家用机器人在完成本周的清洁任务后，接下来几周仍需要再次清扫地面。尽管序列的随机排列相比完全不重复的场景略 less challenging，但评估不同类型的新任务对于一般的持续强化学习场景同样具有意义。

超参数：Atari 游戏。我们的实现基于 Malagon 等人，2024年发布的代码进行了调整。对于基于近端策略优化的 FAME 方法，我们选择估计步数 $N = 20000$，即每个任务中存储在元缓冲区 $\mathcal{M}$中的数据量，缓冲区大小为 200000；选择用于一对多假设检验的策略评估步数 $n = 1200$，以及预热步数 $L = 50000$（每个任务训练步数的 5%）。每个任务中存储数据量占全部训练数据的比例为 $20000/1M = 2\%$。在知识整合阶段，我们使用 2.5×10^{-4} 学习率训练元学习器 200 个训练轮次。快速学习器的学习率同样为 2.5×10^{-4}。在元预热中，我们采用 $\lambda = 1.0$ 进行行为克隆正则化。我们在所有任务中选择命题 1 中的 $\tau = 1$ 。

任务序列：Atari 游戏。我们沿用相关设定，并根据（Malagon 等人，2024）调整了实现方式。(1) 对于 *ALE/SpaceInvaders-v5* 环境，我们设置了 10 个任务，每个任务对应游戏的一种游戏模式。所有任务的目标相同：在太空侵略者抵达地球之前将其击落。观测由 210 × 160 帧的 RGB 图像构成，动作包括：不做任何操作、开火、向右移动、向左移动、向右移动并开火，以及向左移动并开火。10 种模式的详细说明如下：

• **模式 0：** 这是默认设置。玩家有三条生命数，击毁太空侵略者可获得奖励（击中后排的侵略者可获得更多奖励）。
• **模式 1：** 护盾会在屏幕上来回移动，而非停留在固定位置。因此，利用护盾进行防护变得不可靠。

• **模式 2：** 入侵者投下的激光炸弹在屏幕上向下落时会曲折移动，因此更难预测它们会落在哪里。

• **模式 3：** 此任务结合了模式 1 和模式 2。
• **模式 4：** 与模式 0 相同，但激光炸弹下落速度快得多。
• **模式 5：** 与模式 1 相同，但激光炸弹下落速度快得多。
• **模式 6：** 与模式 2 相同，但激光炸弹下落速度快得多。
• **模式 7：** 与模式 3 相同，但激光炸弹下落速度快得多。
• **模式 8：** 与模式 0 相同，但入侵者会有几帧变得不可见。
• **模式 9：** 与模式 1 相同，但入侵者会有几帧变得不可见

对于 *ALE/Freeway-v5*环境，目标是引导一只鸡穿过交通繁忙的道路。以下是 7 种模式的详细说明：

• **模式 0：** 这是默认设置。
• **模式 1：** 交通更加繁忙，车辆速度也更快；最靠近中心的上车道上有卡车。卡车车身更长，因此更难躲避。
• **模式 2：** 卡车的行驶速度比先前模式中最快的车辆还快，而且交通更加繁忙。
• **模式 3：** 所有车道上都有卡车，并且在部分车道中，卡车的行驶速度与模式 2 中一样快。
• **模式 4：** 交通状况与先前模式相似，没有卡车，但车辆速度会随机增大或减小。

• **模式 5：** 与模式 1 相同，但车辆速度会随机变化，并且部分车辆会以两辆或三辆为一组、彼此非常靠近地行驶。
• **模式 6：** 与前一模式相同，但交通更为繁忙。

F.2 MinAtar 中的消融研究

正则化超参数 λ 在行为克隆中。 表 4 表明，过大或过小的 λ 会导致前向迁移和遗忘方面的性能较差，尽管平均性能仍然较好。例如，FAME（$\lambda = 0.1$）在前向迁移和遗忘上的指标分数最差。在实践中，我们可以选择 $\lambda = 1.0$ 以在前向迁移上取得最高分数，或选择 $\lambda = 5.0$ 以获得最佳遗忘分数。

预热步数 L 在行为克隆正则化下。表 5 展示了 FAME 在行为克隆正则化下，不同预热步数对应的综合指标分数。值得注意的是，增加预热步数可以提升某些游戏中的前向迁移和平均性能，

表 4：MinAtar 上正则化超参数 λ 对平均性能（平均性能）、前向迁移（前向迁移）和遗忘的消融研究。结果为 10 个序列（每个序列使用 3 个随机种子）的平均值。

方法	平均性能 ↑ Breakout	Spaceinvader	Freeway	FT ↑	遗忘 ↓
FAME (λ=0.1)	12.79 ± 0.42	**19.52 ± 0.50**	**1.71 ± 0.17**	0.13 ± 0.03	0.77 ± 0.08
FAME (λ=1.0)	**14.54 ± 0.58**	18.72 ± 0.52	1.69 ± 0.17	**0.16 ± 0.03**	0.72 ± 0.13
FAME (λ=5.0)	13.90 ± 0.55	19.38 ± 0.62	1.62 ± 0.16	0.14 ± 0.03	**0.64 ± 0.07**
FAME (λ=10.0)	13.88 ± 0.55	**19.52 ± 0.57**	1.63 ± 0.16	0.14 ± 0.03	0.67 ± 0.08

例如 Spaceinvade 和 Freeway。然而，这会加剧整体遗忘。为了保持遗忘能力，建议将预热阶段持续到快速学习的特定阶段，例如 5×10^4 或 20×10^4 训练步数。

表 5：**预热步数** L 对 MinAtar 平均性能（平均性能）、前向迁移（前向迁移）和遗忘的消融研究。结果为 10 个序列的平均值，每个序列包含 3 个随机种子。

方法	平均性能 ↑ Breakout	Spaceinvader	Freeway	FT ↑	遗忘 ↓
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

权重估计步数 N。 对于固定大小的元学习器缓冲区 $\mathcal{M}$，我们改变权重估计步数 N，该步数决定了在新环境中收集的、用于元学习器知识整合的数据。表 6 表明，减小权重估计步数会在所有指标分数上持续降低性能。未来增大 N 以进一步提升我们的性能是值得的，但这需要比其他基线方法所采用的更大的 $\mathcal{M}$ 缓冲区大小。我们将对更大缓冲区大小 $\mathcal{M}$ 下 FAME 性能的研究留作未来工作。

表 6：MinAtar 上**权重估计步数** N 的消融研究，涉及平均性能（平均性能）、前向迁移（前向迁移）和遗忘。结果在 10 个序列上取平均，每个序列使用 3 个随机种子。

方法	平均性能 ↑ Breakout	Spaceinvader	Freeway	FT ↑	遗忘 ↓
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**

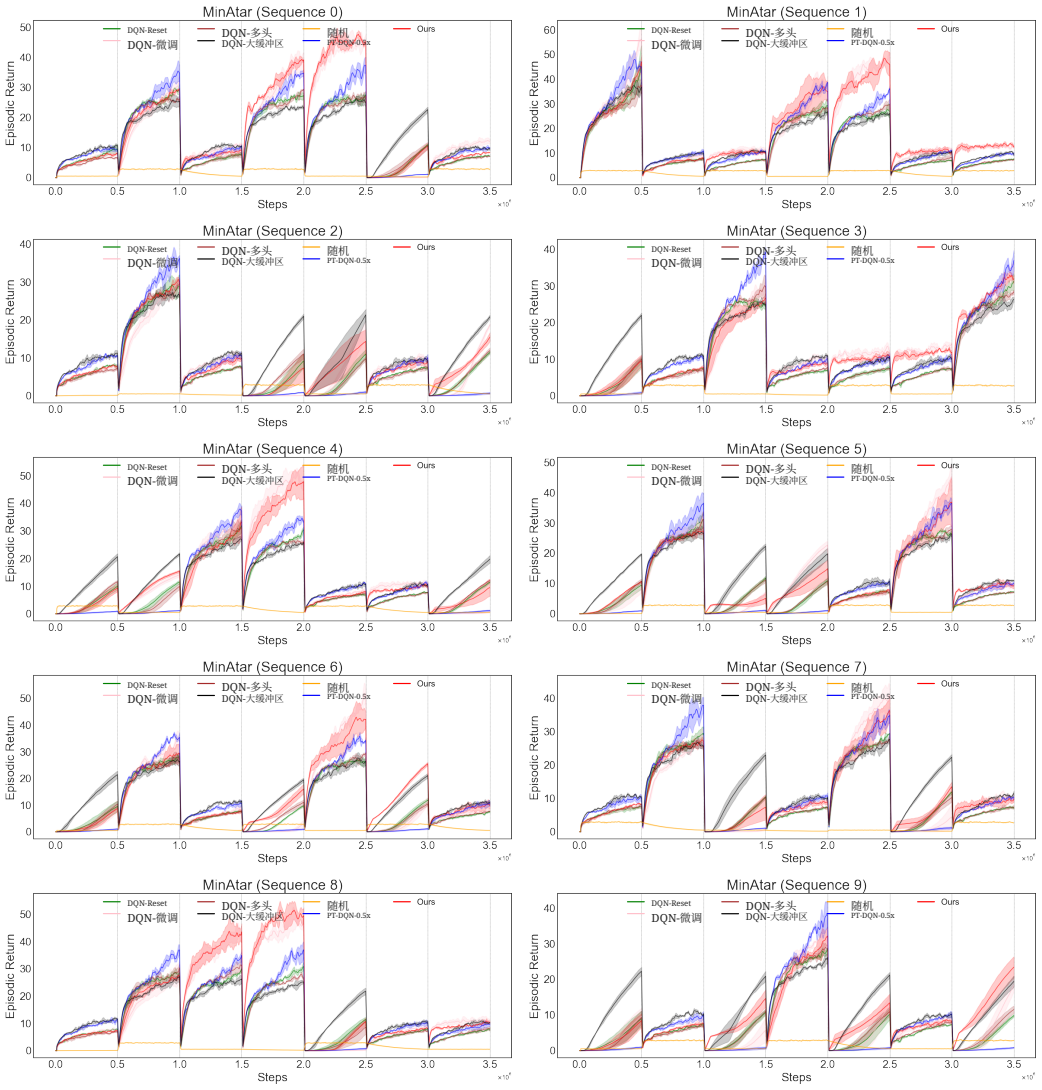
用于策略评估步数 n**的自适应元预热（位于行为克隆正则化之前），步数为** L。 表7显示，适当范围内的策略评估步数不会显著影响指标分数。正如预期，策略评估步数过少可能无法充分选出最佳预热策略，从而降低前向迁移。我们发现，$n = 600$ 足以保持良好的前向迁移，同时也能控制遗忘。

表7：MinAtar 上策略评估步数的消融研究 n，评估其对平均性能（平均性能）、前向迁移（前向迁移）和遗忘的影响。结果为 10 个序列（每个序列使用 3 个随机种子）的平均值。

方法	平均性能 ↑ Breakout	Spaceinvader	Freeway	FT ↑	遗忘 ↓
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

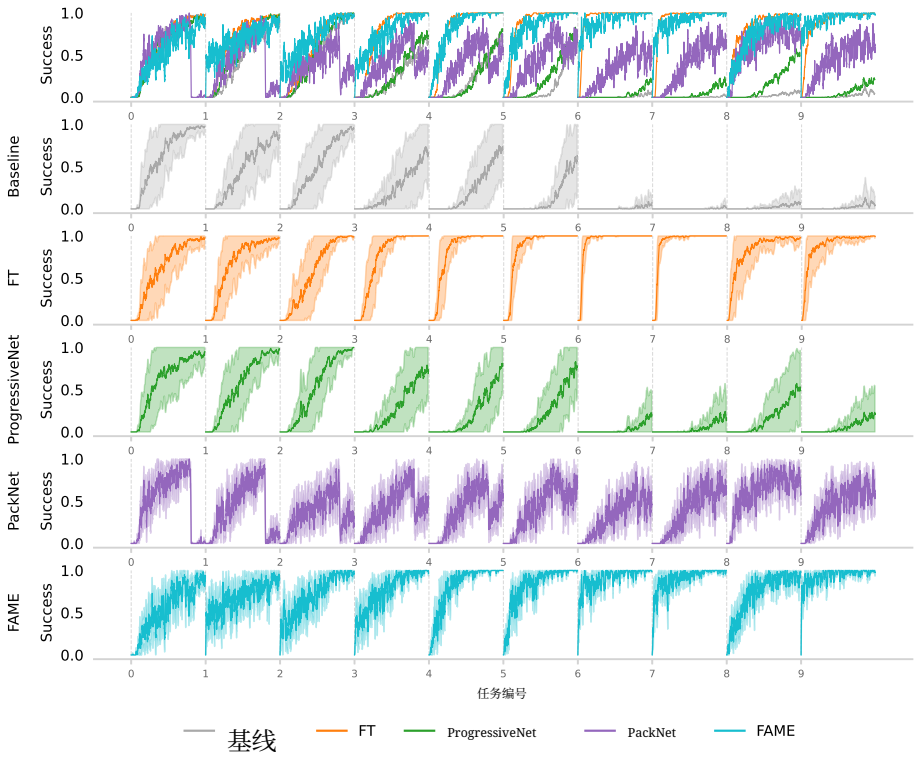
F.3 可塑性能力的学习曲线分析

快速学习器的学习曲线：MinAtar。 图4展示了在MinAtar环境中，所有纳入考量的持续强化学习算法在10个任务序列上的学习曲线，表明在每个新环境中，由自适应元预热引导的快速学习器具有良好的适应能力。



Fig图4：FAME中的快速学习器在MinAtar环境下跨越10个任务序列的学习曲线 s.

快速学习器的学习曲线：Atari 游戏。我们在图5和图6中给出了所有所考虑的持续强化学习算法在两个 Atari环境中的学习曲线。这些结果展现了在自适应元预热的引导下，快速学习器在每个新环境中出色的适应能力。



Fig图5：FAME 中快速学习器在 SpaceIvader环境上的学习曲线，取 3 个随机种子的平均值 s.

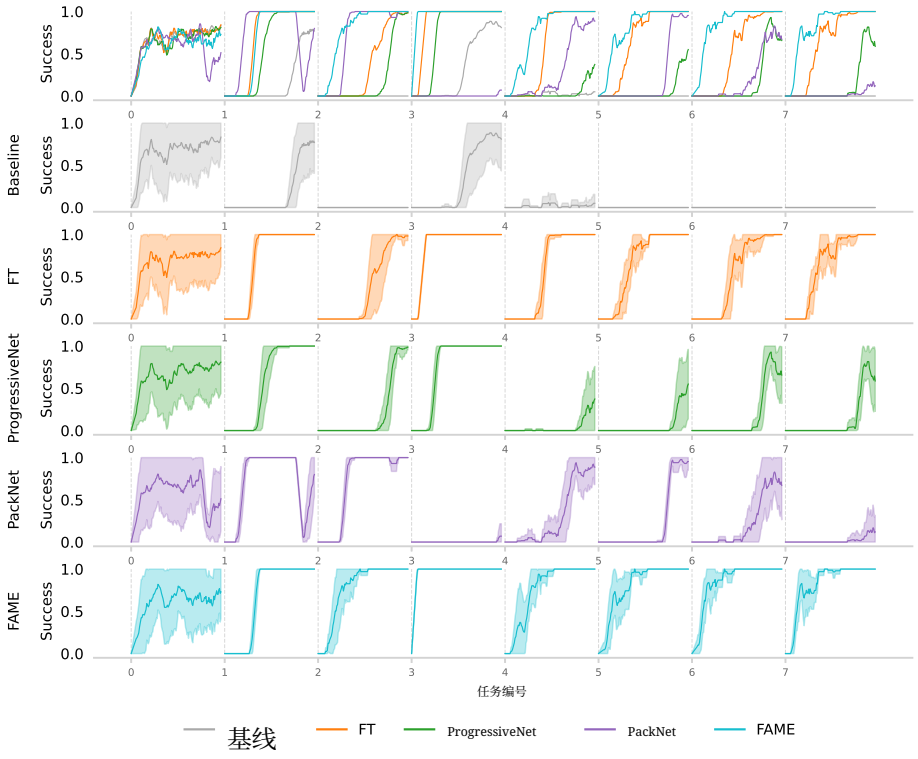


图6：FAME 中快速学习器在 Freeway环境上的学习曲线，取 3 个随机种子的平均值。

G 实验：基于策略的连续动作空间持续学习

G.1 实验设置与对比方法详情

超参数详情。我们的实现改编自已发布的代码（Haarnoja 等人，2018年b；Chung 等人，2024；Zhang 等人，2024年a；b；Malagon 等人，2024）。我们为快速学习器和元学习器分别采用大小为 $1M$ 和 $0.1M$ 的回放缓冲区。元学习器的缓冲区不应过大，因为我们希望算法在不过多重放过去数据的情况下发展出持续学习能力。当新环境到来时，快速学习器在自适应元预热（即知识迁移）的指导下开始训练。同时，快速学习器的回放缓冲区会被重置，其目的仅是包含来自当前任务的转移样本。相比之下，元学习器维护一个缓冲区，存储最近 1% 的转移样本（50 个回合），用于为每个任务更新快速学习器。完成一个任务后，元学习器使用 $5000 \times k$ 个小批量进行训练，其中 k 是截至目前的任务数量。

任务序列。我们从（Chung 等人，2024）中随机选择了 3 个任务序列：

- [‘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’]。

表 8：Meta-World 上的平均性能（平均性能）、前向迁移（前向迁移）和遗忘**在每个任务序列上的**结果。结果以 10 个随机种子上的平均值和标准误表示。从上到下，每个表格分别代表一个任务序列。最佳结果以粗体标出，次佳结果加下划线。↑表示正向指标（越大越好），而↓表示负向指标（越小越好）。重置 是评估前向迁移的基线。

方法	平均性能 ↑	FT ↑	遗忘 ↓
重置	0.090 ± 0.029	0.000 ± 0.000	0.800 ± 0.040
微调	0.070 ± 0.026	-0.294 ± 0.039	0.480 ± 0.050
平均值	0.020 ± 0.014	-0.584 ± 0.035	0.100 ± 0.030
PackNet	0.703 ± 0.041	-0.111 ± 0.028	**0.000 ± 0.000**
FAME-WD	**0.87 ± 0.034**	0.004 ± 0.022	0.010 ± 0.017
FAME-KL	0.86 ± 0.035	**0.042 ± 0.019**	0.050 ± 0.026

方法	平均性能 ↑	FT ↑	遗忘 ↓
重置	0.110 ± 0.031	0.000 ± 0.000	0.680 ± 0.047
微调	0.040 ± 0.020	-0.252 ± 0.045	0.440 ± 0.052
平均值	0.000 ± 0.000	-0.496 ± 0.039	0.110 ± 0.031
PackNet	0.413 ± 0.041	-0.249 ± 0.031	**0.000 ± 0.000**
FAME-WD	**0.750 ± 0.044**	**0.008 ± 0.024**	0.040 ± 0.032
FAME-KL	0.680 ± 0.047	0.004 ± 0.029	0.120 ± 0.036

方法	平均性能 ↑	FT ↑	遗忘 ↓
重置	0.080 ± 0.027	0.000 ± 0.000	0.650 ± 0.052
微调	0.000 ± 0.000	-0.25 ± 0.036	0.360 ± 0.048
平均值	0.020 ± 0.014	-0.509 ± 0.036	0.000 ± 0.002
PackNet	0.358 ± 0.043	-0.222 ± 0.032	**0.000 ± 0.000**
FAME-WD	**0.680 ± 0.047**	-0.020 ± 0.028	0.020 ± 0.032
FAME-KL	0.660 ± 0.048	**0.022 ± 0.028**	0.050 ± 0.030

表 9：Meta-World 上三**个任务序列的平均**平均性能（平均性能）、前向迁移（前向迁移）和遗忘结果。结果以 10 个随机种子上的平均值和标准误表示。最佳结果以粗体标出，次佳结果加下划线。↑ 表示正向指标（越大越好），而↓表示负向指标（越小越好）。重置 是评估前向迁移的基线。

方法	平均性能↑	FT↑	遗忘↓
重置	0.093 ± 0.017	0.000 ± 0.000	0.710 ± 0.030
微调	0.037 ± 0.011	-0.265 ± 0.028	0.427 ± 0.033
平均值	0.013 ± 0.007	-0.530 ± 0.024	0.070 ± 0.022
PackNet	0.491 ± 0.025	-0.194 ± 0.018	**0.000 ± 0.000**
FAME-WD	**0.767 ± 0.024**	-0.003 ± 0.014	0.023 ± 0.015
FAME-KL	0.733 ± 0.026	**0.022 ± 0.015**	0.073 ± 0.019

G.2 更多实验结果与细节

指标分数。 表 8 展示了每个任务序列在平均性能（平均性能）、前向迁移（前向迁移）和遗忘方面的详细结果。表 9 进一步展示了三个序列的平均分数。总之，与基线方法相比，FAME 变体在平均性能和前向迁移方面表现更优。对于遗忘指标：由于 PackNet 保留了先前的策略，其遗忘分数为 0。相比之下，Average 未能学到有意义的知识，整体表现不佳，因此也就没有什么可“遗忘”的。因此，最佳遗忘分数并不总是与最佳平均性能相一致。

性能概况与最终平均性能评估。性能概况（Agarwal 等人，2021年；Dolan & Mor´e, 2002）全面展示了快速学习器在整个任务序列中的总体表现。如图 7（左）所示，FAME 的两种变体始终优于所有基线方法，展现了元学习器随时间有效整合知识并促进迁移学习的能力。此外，图 7（右）凸显了 FAME 在所有先前见过的任务上的优势。随着引入更多任务，基线方法的表现通常会下降，而 FAME-KL 和 FAME-MD 则取得了最高的平均性能——这表明它们几乎没有发生灾难性遗忘，并且能够随着时间推移稳健地保留知识。值得注意的是，图 7 最后一行展示了三个任务序列的平均性能概况和平均性能。

快速学习器的学习曲线。图 8 展示了在训练过程中，我们的 FAME 方法中的快速学习器在三个任务序列上均取得了更高的成功率。前三行展示了三个任务序列的学习曲线，最后一行则展示它们的平均学习曲线。学习曲线是对 10 个随机种子的结果取平均得到的，阴影区域表示标准误。这一优势表明，自适应元预热增强了快速学习器在每个新环境中的适应能力。

H 大语言模型（LLM）的使用

本研究仅使用大语言模型（LLM）进行少量语言润色和编辑，不涉及理论、方法或内容的开发。

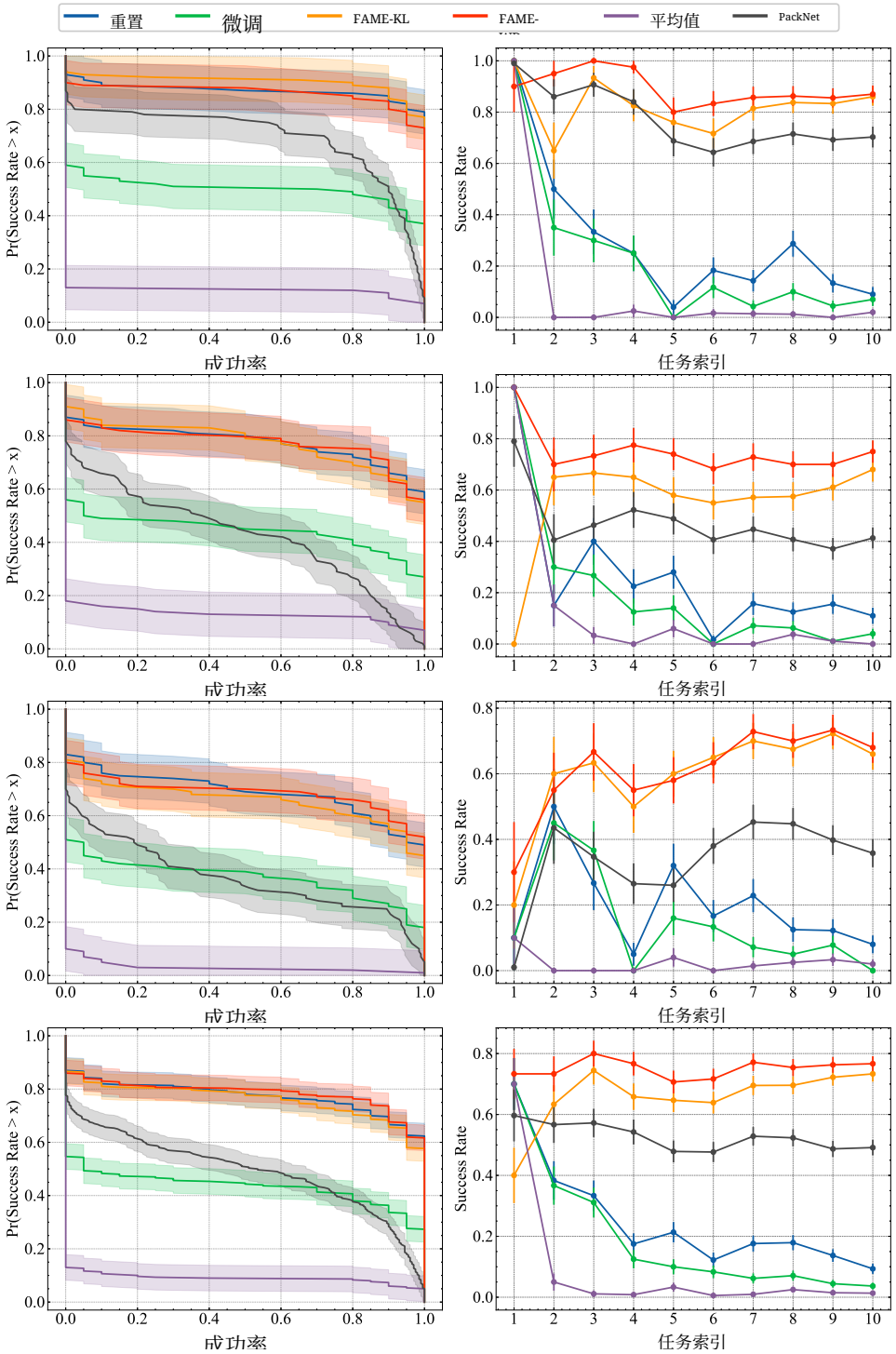


图 7：（左）快速学习器在各任务上的性能概况，其中纵轴表示成功率大于或等于横轴所示数值的任务比例。（右）通过评估历史任务的平均成功率，展示随时间变化的平均性能。**每一行表示一个任务序列的结果。最后一行展示三个序列的平均性能。**

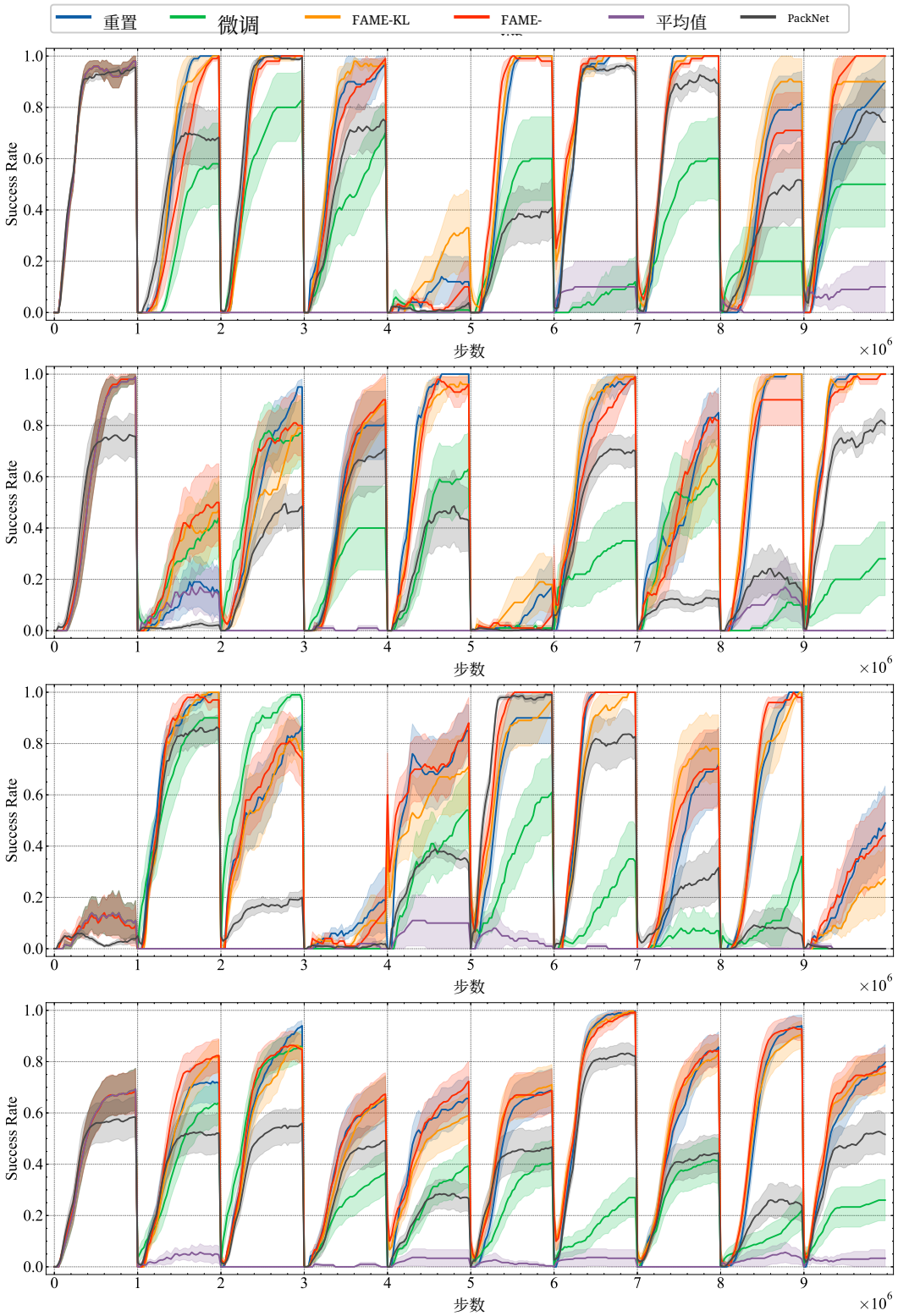


图 8：快速学习器在 Meta-World 基准上的学习曲线。横轴表示环境交互总次数，纵轴表示成功率。**每一行代表一个任务序列的结果。最后一行显示三个序列的平均性能。**

I 补充结果

在本研究的 camera-ready 阶段，我们还增加了更多基线方法：ProgressiveNet（Rusu et al., 2016）和 CompoNet（Malagon et al., 2024年），以及来自 Meta-World 环境中 CW10 的一个新任务序列。实证结果同样表明，我们的 FAME 算法具有优越性。

I.1 META-WORLD 中三个任务序列上的更多基线方法

表 10：Meta-World 上关于平均性能（平均性能）、前向迁移（前向迁移）和遗忘**在增加更多基线方法后的每个任务序列上**的结果。结果以 10 个随机种子上的平均值和标准误呈现。每张表自上而下分别对应各个任务序列。最佳结果以粗体突出显示，次佳结果以下划线标出。↑表示正向指标（越大越好），而↓表示负向指标（越小越好）。重置 是用于评估前向迁移的基线。

方法	平均性能 ↑	FT ↑	遗忘 ↓
重置	0.090 ± 0.029	0.000 ± 0.000	0.800 ± 0.040
微调	0.070 ± 0.026	-0.294 ± 0.039	0.480 ± 0.050
平均值	0.020 ± 0.014	-0.584 ± 0.035	0.100 ± 0.030
PackNet	0.703 ± 0.041	-0.111 ± 0.028	**0.000 ± 0.000**
ProgressiveNet	0.760 ± 0.038	-0.063 ± 0.031	**0.000 ± 0.000**
CompoNet	0.803 ± 0.036	-0.007 ± 0.022	**0.000 ± 0.000**
FAME-WD	**0.87 ± 0.034**	0.004 ± 0.022	0.010 ± 0.017
FAME-KL	0.86 ± 0.035	**0.042 ± 0.019**	0.050 ± 0.026

方法	平均性能 ↑	FT ↑	遗忘 ↓
重置	0.110 ± 0.031	0.000 ± 0.000	0.680 ± 0.047
微调	0.040 ± 0.020	-0.252 ± 0.045	0.440 ± 0.052
平均值	0.000 ± 0.000	-0.496 ± 0.039	0.110 ± 0.031
PackNet	0.413 ± 0.041	-0.249 ± 0.031	**0.000 ± 0.000**
ProgressiveNet	0.604 ± 0.042	-0.133 ± 0.03	**0.000 ± 0.000**
CompoNet	0.566 ± 0.041	-0.125 ± 0.027	**0.000 ± 0.000**
FAME-WD	**0.750 ± 0.044**	**0.008 ± 0.024**	0.040 ± 0.032
FAME-KL	0.680 ± 0.047	0.004 ± 0.029	0.120 ± 0.036

方法	平均性能 ↑	FT ↑	遗忘 ↓
重置	0.080 ± 0.027	0.000 ± 0.000	0.650 ± 0.052
微调	0.000 ± 0.000	-0.25 ± 0.036	0.360 ± 0.048
平均值	0.020 ± 0.014	-0.509 ± 0.036	0.000 ± 0.002
PackNet	0.358 ± 0.043	-0.222 ± 0.032	**0.000 ± 0.000**
ProgressiveNet	0.494 ± 0.044	-0.137 ± 0.031	**0.000 ± 0.000**
CompoNet	0.468 ± 0.045	-0.136 ± 0.037	**0.000 ± 0.000**
FAME-WD	**0.680 ± 0.047**	-0.020 ± 0.028	0.020 ± 0.032
FAME-KL	0.660 ± 0.048	**0.022 ± 0.028**	0.050 ± 0.030

表11：在 Meta-World 上，对包含更多基线方法的三个序列取平均后的结果，指标包括平均性能（平均性能）、前向迁移（前向迁移）和遗忘。结果以 10 个随机种子的平均值和标准误表示。最佳结果以粗体标出，次佳结果加下划线。↑表示正向指标（越大越好），而↓表示负向指标（越小越好）。重置 是用于评估前向迁移的基线。

方法	平均性能 ↑	FT ↑	遗忘 ↓
重置	0.093 ± 0.017	0.000 ± 0.000	0.710 ± 0.030
微调	0.037 ± 0.011	-0.265 ± 0.028	0.427 ± 0.033
平均值	0.013 ± 0.007	-0.530 ± 0.024	0.070 ± 0.022
PackNet	0.491 ± 0.025	-0.194 ± 0.018	**0.000 ± 0.000**
ProgressiveNet	0.619 ± 0.025	-0.111 ± 0.016	**0.000 ± 0.000**
CompoNet	0.612 ± 0.025	-0.089 ± 0.016	**0.000 ± 0.000**
FAME-WD	**0.767 ± 0.024**	-0.003 ± 0.014	0.023 ± 0.015
FAME-KL	0.733 ± 0.026	**0.022 ± 0.015**	0.073 ± 0.019

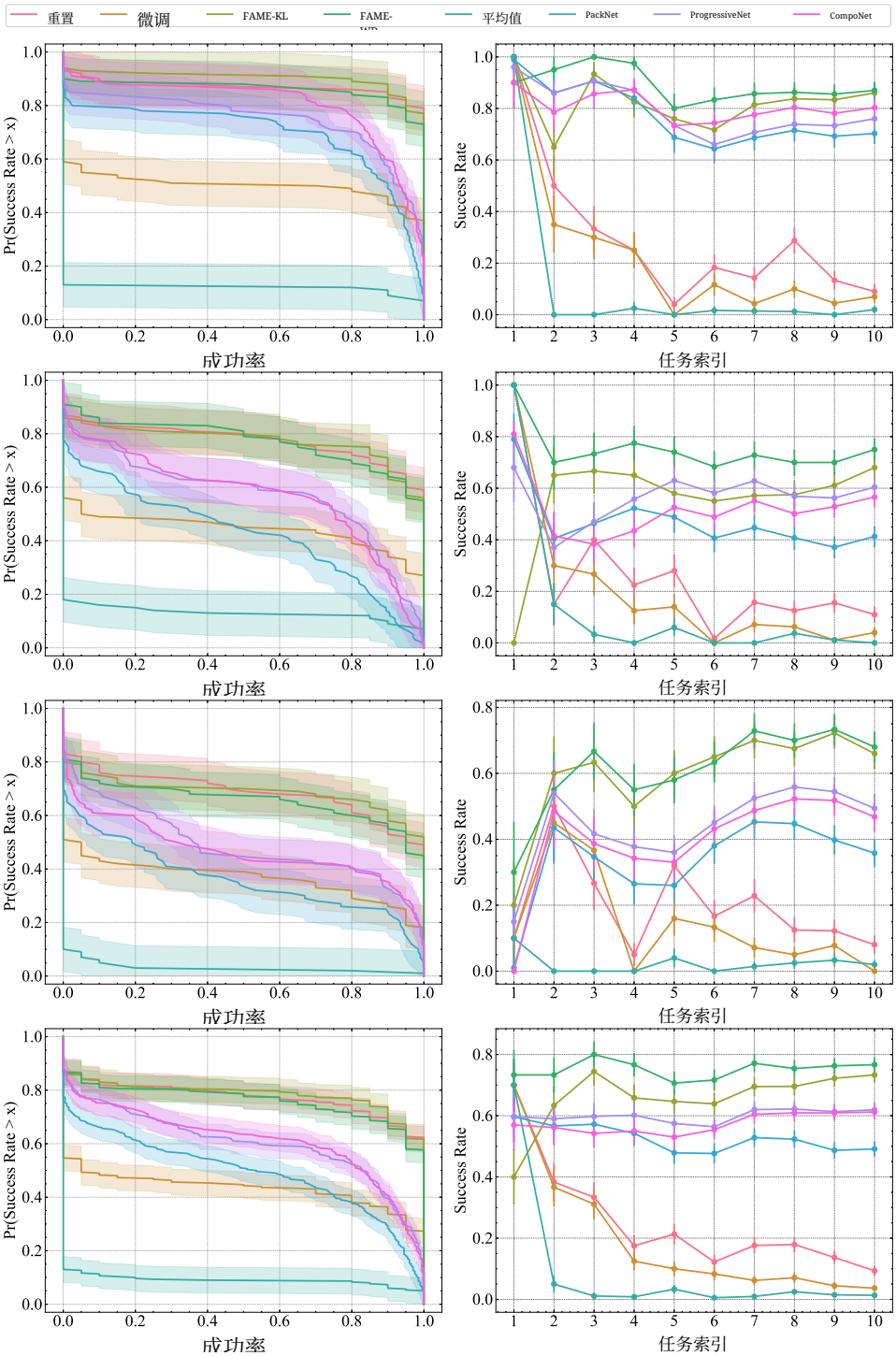


图 9：（左）快速学习器在**包含更多基线方法时**各任务上的性能概况，其中纵轴表示成功率大于或等于横轴数值的任务比例。（右）通过评估历史任务的平均成功率，展示随时间变化的平均性能。**每一行代表一个任务序列的结果。最后一行展示三个序列的平均性能**。

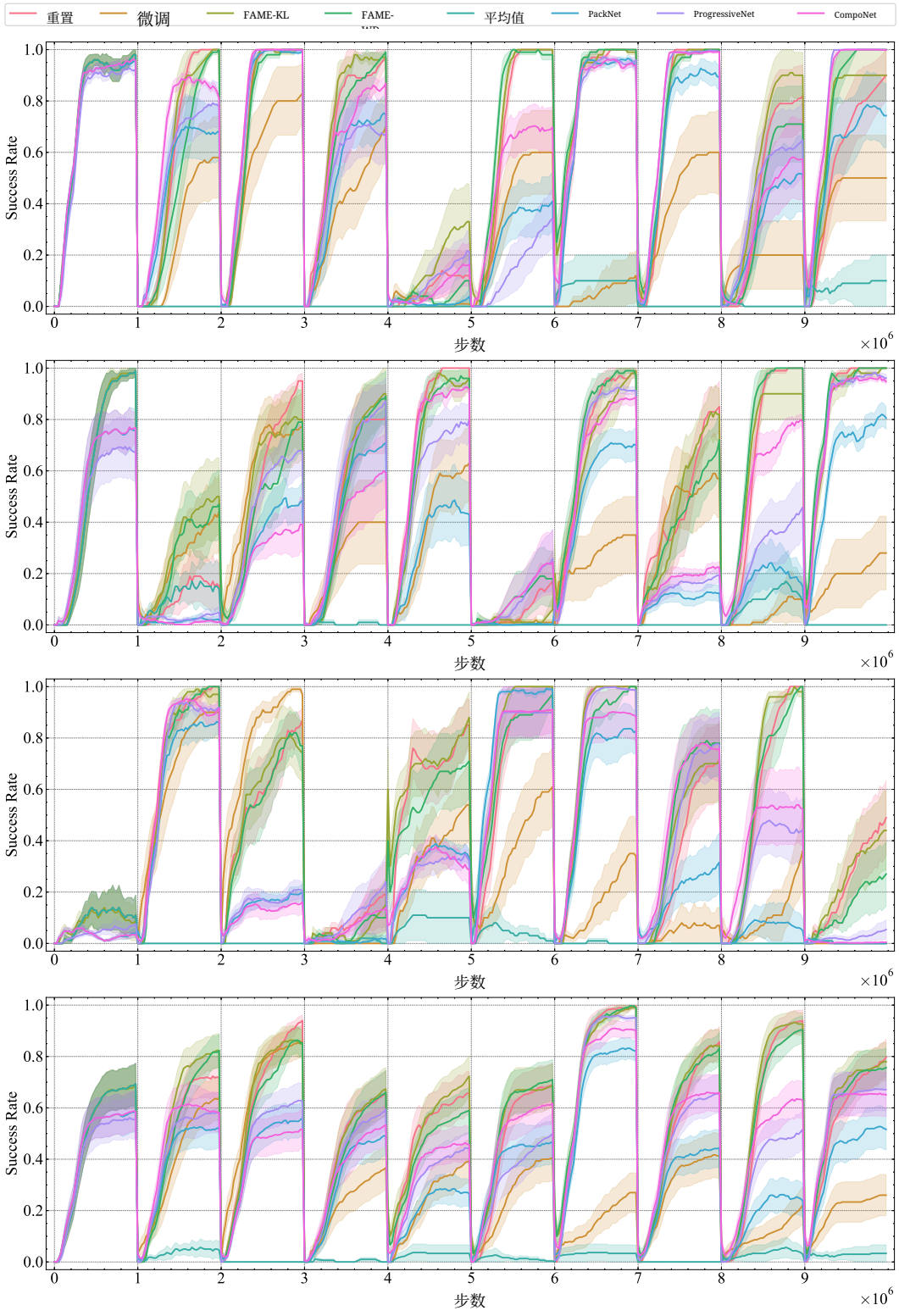


图10：快速学习器在 Meta-World 基准上的学习曲线，**包含更多基线方法**。横轴表示环境交互总次数，纵轴表示成功率。**每一行表示一个任务序列的结果**。**最后一行展示三个序列的平均性能**。

I.2 任务序列补充：CW10

由于时间有限，PackNet、ProgressiveNet 和 CompoNet 在 CW10 上的基线方法使用 3 个随机种子运行，而所有其他基线方法均使用 10 个随机种子运行。

表 12：CW10 任务序列（Wolczyk 等人，2021年）的结果，包括平均性能（平均性能）、前向迁移（前向迁移）和遗忘。结果以 10 个随机种子上的平均值和标准误呈现。最佳结果以粗体突出显示，次佳结果加下划线。 ↑表示正向指标（越大越好），而↓表示负向指标（越小越好）。重置 是评估前向迁移的基线。

方法	平均性能 ↑	FT ↑	遗忘 ↓
重置	0.020 ± 0.014	0.000 ± 0.000	0.370 ± 0.049
微调	0.020 ± 0.014	-0.123 ± 0.028	0.240 ± 0.047
平均值	0.010 ± 0.01	-0.288 ± 0.039	0.010 ± 0.017
PackNet	0.292 ± 0.071	-0.086 ± 0.019	**0.000 ± 0.000**
ProgressiveNet	0.325 ± 0.074	-0.034 ± 0.016	**0.000 ± 0.000**
CompoNet	0.208 ± 0.063	-0.129 ± 0.029	**0.000 ± 0.000**
FAME-WD	0.340 ± 0.048	-0.003 ± 0.016	0.030 ± 0.017
FAME-KL	**0.370 ± 0.049**	**0.015 ± 0.017**	0.060 ± 0.024

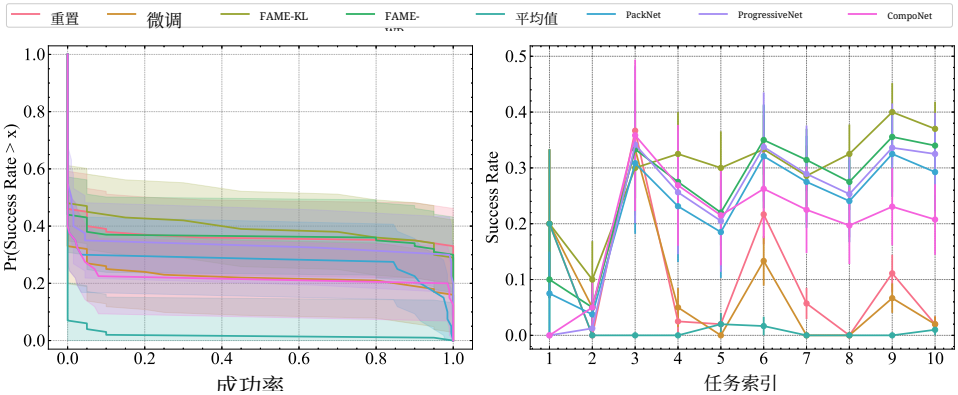


图11：CW10（Wolczyk 等人，2021年）任务序列上的结果。**（左）**快速学习器在各任务上的性能概况，其中纵轴表示成功率大于或等于横轴数值的任务比例。**（右）**通过评估历史任务的平均成功率，展示随时间变化的平均性能。

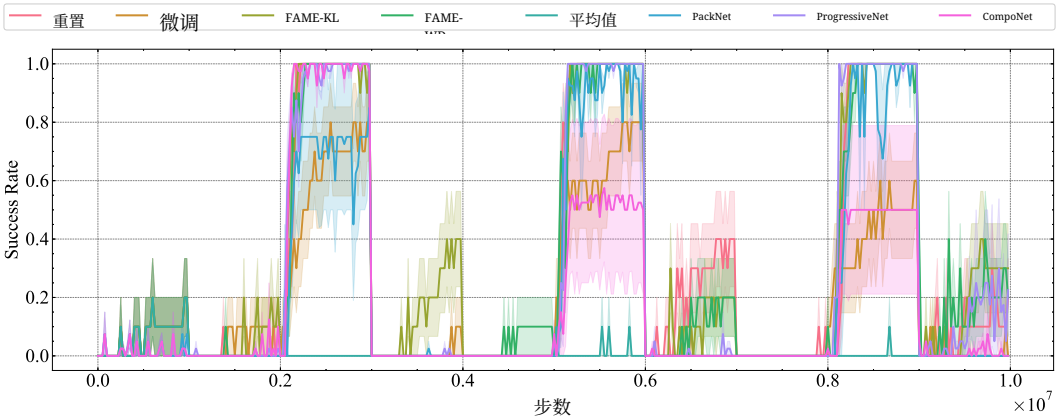


图12：快速学习器在 CW10（Wolczyk 等人，2021年）任务序列上的学习曲线。横轴表示环境交互总次数，纵轴表示成功率。**每一行代表一个任务序列的结果。最后一行展示三个序列的平均性能。**