带任务边界检测的在线元学习算法设计

Daouda Sow[1], SenLin[2], YingbinLiang[1], JunshanZhang[3]1俄亥俄州立大学电子与计算机工程系2休斯敦大学计算机科学系3加州大学戴维斯分校电子与计算机工程系
sow.53@osu.edu, slin50@central.uh.edu, liang889@osu.edu, jazh@ucdavis.edu

arXiv:2302.00857v2 [cs.LG] 5 Aug 2024

在线元学习近年来兴起，融合了批量元学习与在线学习，以实现终身学习方式下对新任务的快速适应。然而，现有大多数方法都局限于这样一种设定：在线任务的分布保持固定，且任务边界已知。本文放宽了这些假设，并提出一种适用于非平稳环境的全新任务无关在线元学习算法。具体而言，我们首先基于经验观察，提出两种简单而有效的任务切换和分布偏移检测机制，它们构成了算法中更精细在线模型更新的关键基础：任务切换检测机制能够复用当前任务可用的最佳模型；分布偏移检测机制则对元模型更新加以区分，从而保留对分布内任务的知识，并快速学习适用于分布外任务的新知识。具体来说，我们的在线元模型更新仅基于当前数据，从而无需像现有大多数方法那样存储先前数据。我们进一步证明，在温和条件下，该算法能够实现次线性任务平均遗憾。在三个不同基准测试上的实证研究清楚地表明，与相关基线方法相比，我们的算法具有显著优势。

1. 引言

人类智能的两个关键方面，是快速学习复杂任务的能力，以及持续更新知识库以便更快地学习未来任务的能力。元学习 [1–3]和在线学习 [4–6] 是旨在赋予学习智能体这些能力的两个主要研究方向。具体而言，元学习旨在通过基于相关任务的知识构建模型参数的先验，促进对未见过的新任务进行快速学习；而在线学习研究的是任务数据依次揭示给学习智能体的问题。为实现以终身学习方式快速适应新任务的能力，在线元学习 [3, 7, 8] 近年来受到广泛关注。考虑在线任务逐一到来的情形，在线元学习的目标是持续更新元先验，使智能体在遇到更多任务后，能够据此更快地学习新任务。

在线元学习中，智能体通常维护两个独立模型，即用于捕捉各任务共同底层知识的元模型 ，以及用于解决当前任务的在线任务模型 。现有大多数在线元学习研究 [3, 9] 采用“重置”策略：利用当前数据从元模型快速适配在线任务模型，更新元模型，并在下一个任务开始时将在线任务模型重置为更新后的元模型。当任务边界已知且任务分布保持平稳时，这一策略通常效果良好。然而，在许多真实世界数据流中，智能体无法直接观测到任务边界 [7, 10, 11], ，而且任务分布在在线学习阶段可能发生动态变化。因此，本研究旨在解决这类更贴近现实场景下的在线元学习问题。

首届简约性与学习会议（CPAL 2024）。

毋庸置疑，在非平稳环境中，如何在未知任务边界的情况下高效解决在线元学习问题并非易事，原因在于以下几个关键问题：（1）如何更新元模型和在线任务模型？ 显然，新数据到来时采用“重置”策略并不可取，因为如果新数据与先前数据属于同一任务，那么从先前的任务模型进行适配更为合适。另一方面，对于分布内（IND）任务和分布外（OOD）任务，元模型的更新方式应有所区别：前者应保留当前知识，后者则应快速学习新知识。（2）如何让系统轻量化，以实现快速在线学习？在线元学习的特性使其无法采用复杂的学习算法，因为智能体应能在通常无法访问先前数据的情况下快速适应不同任务。此外，考虑到环境可能在在线学习期间快速变化，应对环境非平稳性不应显著增加计算成本。

我们的主要贡献可概括如下。

(1) (新颖的算法设计) 我们提出了一种适用于非平稳环境的新型在线元学习算法，无需预先知道任务边界，能够妥善解决上述问题。具体而言，受经验观察的启发，我们首先提出了两种简单而有效的机制，分别利用分类损失检测任务切换，并利用亥姆霍兹自由能检测分布偏移[12], 。基于这些检测机制，我们的算法对在线模型更新进行了更细致的处理，从而带来以下优势：**(1) (任务知识复用)** 检测任务切换使算法能够复用每个任务可用的最佳模型，避免像大多数先前研究那样在每一步都“重置”为元模型；**(2) (审慎的元模型更新)** 检测分布偏移使算法能够以这样的方式更新元模型：针对分布外任务，新知识能够快速习得；针对分布内任务，则能保留先前知识；**(3) (高效的内存使用)** 我们的算法不会复用或存储任何先前数据，而是在每个在线回合中仅基于当前数据更新元模型，这显然不同于在线元学习领域的大多数现有研究 [8, 10, 13]。

(2) (大量实验) 我们在三个不同的标准在线元学习基准测试上进行了大量实验。实验结果表明，我们的算法在所有基准测试上都显著优于相关基线方法。消融研究也验证了所提出检测机制的有效性。

(3) (理论分析) 我们在考虑任务边界检测的情况下，对所提出的算法进行了遗憾分析，并证明在温和条件下可以实现次线性任务平均遗憾。具体而言，我们的结果揭示了任务相似性对标准在线元学习性能的影响（假设任务边界已知），与任务边界检测存在不确定性时的性能之间的权衡。也就是说，任务越相似，随着时间推移任务变化越少，因此性能越好；但任务切换也越难检测。

相关工作：元学习。 元学习也称为学会学习， [3, 14, 15]是一种强大的工具，可利用从相关任务中获得的过往经验，快速为新的、未见过的任务学习出良好的任务特定模型。作为推动近年来元学习取得成功的先驱方法，模型无关元学习（MAML） [3] 旨在寻找良好的元初始化，使得从该元初始化出发进行一步或几步梯度下降后，能够为新任务得到良好的任务特定模型。已有研究提出了 MAML 的若干变体 [16–23]。其他方法主要包括基于模型的方法 [2, 24–26] 和基于度量空间的方法 [1, 14, 27, 28]。

在线学习。在在线学习中， [4, 6, 29], 代价函数会依次揭示给智能体，而智能体必须在看到每个代价之前选择一个动作。最受研究的方法之一是跟随领导者（FTL）， [4], 它在每一步利用此前见过的所有损失函数更新参数。为提高稳定性，人们还提出了 FTL 的正则化版本 [30, 31]。从计算资源的角度看，我们的方法与在线梯度下降（OGD） [32]有相似之处：在线梯度下降每一轮仅利用已揭示的损失执行一次梯度下降。然而，传统在线学习方法无法有效利用过往经验，也不针对零样本性能进行优化

，也不进行任何适应。在这项工作中，我们研究在线元学习问题，其目标是随着智能体不断接触更多任务，优化其在未来任务上的快速适应能力。

在线元学习。 在线元学习最早于 [13]提出。开创性方法[8, 13, 33] 采用类似 FTL 的设计思路，需要存储先前任务，因此内存需求会线性增长。正则化跟随领先者（FTRL） [31] 方法也被扩展到在线元学习场景中 [34, 35], ，从而降低了内存需求。 [9] 提出了一种内存高效的方法，将先前任务的经验概括为一个状态向量。然而，这些方法都需要了解任务边界，并在每个在线回合开始时将任务模型“重置”为元模型 [3, 36]。与 [9], 类似，我们的算法也避免了内存的线性扩展。但与其方法不同，我们不需要了解任务边界，并且考虑了动态环境。 [10] 通过始终从先前状态开始更新在线模型，缓解了“重置”问题；然而，这种方法需要存储先前模型，而且性能有限，尤其是在相邻任务差异很大的动态环境中。

上述方法均未考虑动态环境下的在线元学习问题：在这种环境中，任务分布会随时间发生显著变化，且任务边界未知。 [11] 是首个在动态环境中对所提出算法进行实证评估的工作，但未提出快速学习分布外任务所需知识的方法。相比之下，我们以保留分布内知识的方式更新元表示，同时不断提升对分布外任务的快速适应能力。

2. 背景与问题表述

背景。 我们首先简要介绍在线元学习的一些相关概念。

基于 *MAML* 的元学习。 元学习（ [3, 14, 15], 又称学会学习）旨在借助相似任务中的知识，通过有限样本快速学习新任务。更具体地说，其目标是基于一组任务 $\{\mathcal{T}_i\}_{i=1}^M$ 学习元模型，这些任务取自某个未知分布 $\mathbb{P}(\mathcal{T})$；基于该元模型，可以快速获得适用于来自同一平稳分布 $\mathbb{P}(\mathcal{T})$的新任务的任务特定模型。以 MAML 为例，其目标是学习模型初始化 θ，使得从 θ 出发进行一次或几次梯度下降后，便能为新任务 $\mathcal{T} \sim \mathbb{P}(\mathcal{T})$得到一个良好的模型；为此，需要利用训练任务 $\{\mathcal{T}_i\}_{i=1}^M$求解以下问题：

$$\theta := \arg\min_\theta \frac{1}{M} \sum_{i=1}^M f_i\left(U_i(\theta)\right) \tag{1}$$

ˆ ˆ 其中任务模型 $\phi_i = U_i(\theta) = \theta - \alpha \nabla f_i(\theta)$， f_i 和 f_i 分别是任务 $\mathcal{T}_i$的训练损失和测试损失。

在线学习。 在一般的在线学习问题中，损失函数会依次揭示给学习智能体：在每一步 t，智能体首先选择一个动作 θ_t，然后产生代价 $f_t(\theta_t)$。智能体的目标是选择一系列动作，使以下**静态** 遗憾最小化

$$\mathrm{R}(T) = \sum_{t=1}^T f_t\left(\theta_t\right) - \min_\theta \sum_{t=1}^T f_t\left(\theta\right), \tag{2}$$

也就是说，智能体的预测结果 $\{f_t(\theta_t)\}_{t=1}^T$ 与事后看来最佳**静态** 模型的表现之间的差距。成功的智能体能够实现遗憾 R(T)，其增长速度关于 T是次线性的。在线学习是一个研究成熟的领域，更多信息请参阅 [37] 。

在线元学习。 作为在线学习与元学习的结合，在线元学习[7, 8, 13] 旨在实现以下两个特性：**(i)** 快速适应当前任务（元学习方面）；**(ii)** 随着看到更多任务，学会更快地适应（在线学习方面）。具体而言，智能体观察从 $\mathbb{P}(\mathcal{T})$中采样的任务流 $\mathcal{S} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T\}$，其中任务逐一揭示。对于每个任务 $\mathcal{T}_t$，智能体都可以访问用于任务特定适应的支持集 S_t 以及用于评估的查询集 Q_t。这里的目标是选择一系列元模型$\{\mathbf{w}_t\}$，使以下遗憾的增长速度为次线性

$$\mathrm{R}_{\mathrm{meta}}(T) = \sum_{t=1}^T f_t\left(U_t\left(\theta_t\right)\right) - \min_\theta \sum_{t=1}^T f_t\left(U_t(\theta)\right) \tag{3}$$

其中， U_t 是依赖于支持集 S_t 的任务适应函数，而代价函数 f_t 则使用查询集 Q_t 上的适应后的参数 $U_t(\mathbf{w}_t)$ 进行评估。直观而言，智能体旨在学习一个更好的元模型，使其在观察到更多任务后，能够为未来的任务生成更好的任务模型。

非平稳环境中的在线元学习。与大多数在线元学习研究 [7–10, 13, 33]，不同，在这项工作中，我们考虑一种更贴近现实的情形：

预训练元模型。 在许多实际应用中，有大量数据可用于预训练；如果不具备关于当前任务的任何基础知识，就将智能体部署到复杂的动态环境中是不现实的 [11]。因此，按照 [11], 中的相同思路，我们假设存在一组训练任务 $\{\mathcal{T}_i^0\}_{i=1}^M$，这些任务 $\mathbb{P}^0(\mathcal{T})$来自某个未知分布。按照元学习中的标准做法，每个预训练任务 $\mathcal{T}_i^0$都有一个支持数据集 S_i^0和一个查询数据集 Q_i^0。在本文中，我们在训练任务上采用 MAML，以学习一个预训练元模型。

未知的任务边界。 在在线元学习阶段，我们假设任务边界是未知的，即所谓的任务无关设置 [11], ；具体而言，智能体不知道时刻 t新到达的数据属于先前的任务还是新任务。为刻画任务边界的不确定性，我们假设在任意时刻 t，新数据以概率 $p \in (0, 1)$ 属于先前的任务，或以概率 $1-p$属于新任务。

非平稳的任务分布。在在线元学习阶段，智能体可能会遇到从其他分布而非预训练分布中采样的新任务 $\mathbb{P}^0(\mathcal{T})$。为捕捉任务分布中的这种非平稳性，我们假设在线学习期间每当有新任务到来时，它要么以概率 $\eta \in (0, 1)$ 从 $\mathbb{P}^0(\mathcal{T})$ 中采样，要么以概率 $1-\eta$ 从一个新的（相对于 $\mathbb{P}^0(\mathcal{T})$ 而言）分布中采样。请注意，我们不限制在线学习期间可能遇到的新分布数量，并且任务分布可能会再次出现。

3. 分布偏移下的拟议算法

为解决上述非平稳环境下的在线元学习问题，我们接下来提出一种简单但有效的算法，称为分布偏移下的在线元学习（LEEDS）。该算法通过检测任务切换和分布偏移来辅助快速在线学习。与大多数在线元学习研究 [10, 11] 一样，我们在在线学习阶段维护两个独立模型： θ 用作元模型， ϕ 用作在线任务模型。

任务切换与分布偏移的检测：为了在在线学习中快速学习新任务，检测机制不能过于复杂，而必须高效且具有较高的检测准确率。为此，我们提出了两种不同的方法，分别用于检测任务切换和分布偏移，它们协同工作，成为 LEEDS 的关键组成部分。

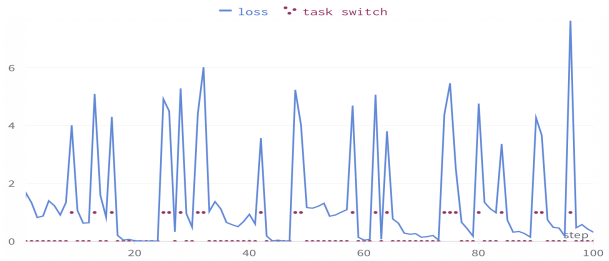
任务切换的检测。为了理解任务切换时的学习行为，我们使用新到达的数据评估前一个任务模型的分类损失，即在时刻 t 的 $\mathcal{L}(\phi_{t-1}; S_t)$，其中 $\mathcal{L}$ 是损失函数， ϕ_{t-1} 是时刻 $t-1$ 的前一个在线模型， S_t 是当前支持集。图1左侧的图展示了在线小样本图像识别问题上的实证结果。如图所示，当智能体接收到来自同一任务的更多数据时，损失值会持续下降；但每当有新任务到来时，损失值就会突然上升。这显然是合理的，因为针对前一个任务学习得到的在线模型已不再适用于新任务。受这一实证观察启发，我们采用一种基于 $\mathcal{L}(\phi_{t-1}; S_t)$ 值的简单机制来检测任务边界：当损失超过预设阈值时，即判定发生了任务切换。如第5节后文所示，这种简单机制确实相当有效，在各种在线元学习问题上都取得了较高的检测准确率。

分布偏移检测。为了高效判断新任务是分布内还是分布外，即是否从预训练任务分布中采样得到，我们考虑一种基于能量的分布外检测机制，其二元分类器 $C_\tau(\cdot; \theta)$ 定义如下

$$C_\tau(\mathbf{x}; \theta) = \begin{cases} 1 & \text{if } -\mathrm{E}(\mathbf{x}; \theta) \le \tau \\ 0 & \text{if } -\mathrm{E}(\mathbf{x}; \theta) > \tau \end{cases} \tag{4}$$

其中，$\mathrm{E}(\mathbf{x}; \theta) = -\delta \log \sum_{k=1}^K \exp\left(-g_k(\mathbf{x}; \theta)/\delta\right)$对应于输入$\mathbf{x}$的亥姆霍兹自由能， $g_k(\cdot; \theta)$是预测模型输出的第 k个分量，而 δ 是温度参数。超参数 τ 是一个可通过预训练任务设置的阈值。如 [12],所示

负能量与输入$\mathbf{x}$ 在模型 θ下的似然呈线性关系，因此可作为区分分布内任务和分布外任务的有效得分。



方法	内存
Ours	$\mathcal{O}(1)$
[10]	$\mathcal{O}(T)$
[11]	$\mathcal{O}(\frac{1}{1-p})$
[9]	$\mathcal{O}(1)$
[13]	$\mathcal{O}(T)$

图 1：**左图：**使用 MAML 预训练的元模型部署于在线学习时在线损失的变化。红点为 0 表示该时刻没有发生任务切换，为 1 则表示该时刻发生了任务切换。**右表：**不同方法的内存需求对比。 T 表示在线轮次， $p \in (0, 1)$表示非平稳性水平。

元参数和在线参数的更新：基于这两种检测方案，接下来的问题是如何更新元模型和任务模型，以便在动态环境中快速适应。

如果没有这类检测机制，以往的在线元学习算法通常会使用支持集，根据元模型适应任务模型，使用查询集评估适应后的模型，并在在线数据流过程中接收到新数据时，将任务模型重置为元模型，然后重复这一过程。然而，这种“重置”方案在现实场景中可能并非最优。例如，如果新接收的数据与之前的数据属于同一任务，智能体应当从先前适应后的参数开始更新任务模型，而不是从元模型开始。

相比之下，本文简单但有效的检测机制能够更优雅地处理在线学习过程中的知识更新：

（1）如果在时间 t 发生任务切换，即 $\mathcal{L}(\phi_{t-1}; S_t) > \ell$，其中 ℓ 是阈值，那么通常从元模型进行适应会优于从前一任务的任务模型进行适应。因此，我们首先使用新数据从元模型获得在线任务模型 ϕ_t：

$$\phi_t = \theta_{\text{adapt}} = \theta_{t-1} - \alpha_1 \nabla_{\theta_{t-1}} \mathcal{L}(\theta; S_t),$$

然后，无论是否发生分布偏移，都更新元模型，以便将新任务的知识纳入元模型：

$$\theta_t = \theta_{t-1} - \alpha_2 \nabla_\theta \mathcal{L}(\theta_{\text{adapt}}; Q_t).$$

(2) 如果没有发生任务切换，即 $\mathcal{L}(\phi_{t-1}; S_t) \leq \ell$，我们将使用新数据，基于前一个任务模型继续更新任务模型，这不同于文献中的“重置”方案：

$$\phi_t = \phi_{t-1} - \alpha_1 \nabla_{\phi_{t-1}} \mathcal{L}(\phi_{t-1}; S_t).$$

为了加快新领域的知识学习，我们进一步区分分布内（IND）任务和分布外（OOD）任务的元模型更新方式。具体而言，如果当前任务是分布内任务，我们只会在该任务开始时更新一次元模型。也就是说，在同一任务中不会进一步更新元模型。与此截然相反，如果当前任务是分布外任务，每当收到该任务的新数据时，我们都会继续更新元模型，如下所示：

$$\theta_{\text{adapt}} = \theta_{t-1} - \alpha_1 \nabla_\theta \mathcal{L}(\theta_{t-1}; S_t), \quad \theta_t = \theta_{t-1} - \alpha_2 \nabla_\theta \mathcal{L}(\theta_{\text{adapt}}; Q_t).$$

LEEDS 的详细信息见算法 1。请注意，如后文实验结果所示，LEEDS 对阈值参数 ℓ 和 τ 确实具有稳健性；此外，我们还在附录中说明了如何更有效地设置阈值的启发式方法。

节省内存：所提出的 LEEDS 算法的一个重要特点是，元模型的更新仅基于当前数据。相比之下，此前的大多数研究都会将先前数据存储在内存中，以便更新元模型。各种方法的内存需求对比

算法 1 分布偏移下的在线元学习（LEEDS）

输入： 动态数据流 $\mathcal{S}$、预训练分布 $\mathbb{P}^0(\mathcal{T})$、步长 α_1 和 α_2、阈值 ℓ 和 τ。
使用模型无关元学习方法，在从 $\mathbb{P}^0(\mathcal{T})$中抽取的任务上执行预训练阶段
当 数据流 $\mathcal{S}$ 处于活动状态时 **执行**
  $D_t \longleftarrow \mathcal{S}$ // 从在线数据流接收当前数据 S_t， Q_t
  S_t， $Q_t \longleftarrow D_t$ // 将数据拆分为支持集和查询集
  如果 $\mathcal{L}(\phi_{t-1}; S_t) \leq \ell$ （即不切换）**那么**
    $\phi_t = \phi_{t-1} - \alpha_1 \nabla_{\phi_{t-1}} \mathcal{L}(\phi_{t-1}; S_t)$ // 从之前的在线模型开始进行适配
    在查询集 Q_t上评估 ϕ_t
    如果 $C_\tau(S_t;\ \theta_{t-1})$（即协变量偏移）**那么**
      $\theta_{\text{adapt}} = \theta_{t-1} - \alpha_1 \nabla_\theta \mathcal{L}(\theta_{t-1}; S_t)$, $\theta_t = \theta_{t-1} - \alpha_2 \nabla_\theta \mathcal{L}(\theta_{\text{adapt}};\ Q_t)$ // 更新元模型
    结束 if
  else
    $\theta_{\text{adapt}} = \theta_{t-1} - \alpha_1 \nabla_{\theta_{t-1}} \mathcal{L}(\theta_{t-1}; S_t)$ // 从元模型出发，使用支持集进行适应
    设置 $\phi_t = \theta_{\text{adapt}}$ 并在查询集 Q_t上评估 ϕ_t
    $\theta_t = \theta_{t-1} - \alpha_2 \nabla_\theta \mathcal{L}(\theta_{\text{adapt}}; Q_t)$ // 更新元模型
  结束 if
结束 while

汇总于图1右侧的表格中。正如后文实验所示，LEEDS 无需存储任何先前数据，性能便显著优于相关基线方法。这一令人鼓舞的结果表明，除了强调算法复杂性之外，研究群体也可以从对更简单设计的审慎探索中有所借鉴。

4. 理论结果

在存在分布偏移的在线元学习中，显然，公式（3）中的静态 比较器不足以刻画非平稳性，因为我们不能指望所有任务分布都使用同一个元模型。因此，我们参照 [34]考虑任务平均遗憾（TAR）

$$\mathbf{R}_{\text{avg}} = \frac{1}{T} \sum_{t=1}^{T} \left(\sum_{k=1}^{K_t} f_t^k \left(\phi_t^k \right) - \sum_{k=1}^{K_t} f_t^k \left(\phi_t^* \right) \right) \tag{5}$$

其中， T 是任务数量， K_t 是任务 t中的步数， ϕ_t^* 是每个任务 t对应的动态比较器。为简单起见，我们将任务 t 的第 k步中的任务内损失函数（在查询集上评估）记为 f_t^k。如 [34], 所示，由于动态比较器 $\{\phi_t^*\}_t$ 可能会导致每个任务 t都产生恒定损失，因此不能指望 TAR 随 T 增加而下降。不过，求平均时考虑的并非总步数 $\sum_{t=1}^{T} K_t$，而仅仅是任务数量 T。因此，此处的目标是实现关于 K_t呈次线性的 TAR。需要注意的是，在许多实际的在线元学习问题中，任务内损失函数 $\{f_t^t\}_{k=1}^{K_t}$ 通常是非对抗性的。例如，在少样本在线图像分类问题中， $\{f_t^t\}_{k=1}^{K_t}$对应于同一任务在不同查询集上对同一分类损失函数进行的不同评估。接下来我们将证明，即使考虑任务边界的检测错误，非对抗性情形下仍可实现恒定的 TAR。

令比较器 ϕ_t^*为每个损失函数 $f_t^k, k = 1, ..., K_t$的极小化器。为正式刻画在线元学习中任务内损失函数 $\{f_t^t\}_{k=1}^{K_t}$ 的非对抗性，我们作出以下假设。

假设4.1. 比较器 ϕ_t^* 是任务 t中第 k 步适应映射 U_t^k的不动点，即 $U_t^k(\phi_t^*) = \phi_t^*$，对 $k = 1, ..., K_t$均成立。

假设4.2. 适应映射 U_t^k是压缩映射，即存在某个 ρ_t^k，使得对所有 ϕ_1, ϕ_2， $\left\| U_t^k(\phi_1) - U_t^k(\phi_2) \right\| \leq \rho_t^k \left\| \phi_1 - \phi_2 \right\|$。

假设4.1刻画了动态比较器 ϕ_t^*的性质。显然，若存在一个任务模型 ϕ ，能够在从任务 t的数据分布中采样的所有数据点上实现零损失，则假设4.1成立。例如，在过参数化情形下，

总能训练出一个模型，以近乎为零的损失完美预测所有训练数据点。在在线元学习中，假设4.2很容易满足 [13]。例如，当函数ˆ f_t^k是β-光滑且 μ-强凸的，并且步长 α 选取在(0， $\frac{2}{\beta}$)内时，一步梯度下降映射，即 $U_t^k(\phi) = \phi - \alpha\nabla\hat{} f_t^k(\phi)$，满足假设4.2。定义 $\rho = \max_{t,\ k} \rho_t^k$。

假设 4.3。 每个函数 f_t^k都是 L-光滑的。

请注意，我们不对任务内损失函数 f_t^k的凸性作出任何假设。对于在线元学习过程中的任意步骤 r（其中每个任务 t 包含 K_t 个步骤），令 P_r 表示当前数据分布，令 ϕ_r 表示适应后得到的任务模型。为分析本文提出的检测机制的错误概率，我们对单数据损失函数 ℓ作出如下假设：

假设 4.4。 令 $\ell(\cdot, \xi)$表示数据点 ξ 上的损失，令 $0 \le \ell(\cdot, \xi) \le M$。我们假设存在常数 $\ell_m \le \ell_p$，使得对于任意步骤 r和 s，以下条件成立：(1) $E_{\xi\sim P_r}\ell(\phi_r, \xi) \le \ell_m$；(2) 当 $P_r \ne P_s$时， $E_{\xi\sim P_r}\ell(\phi_s, \xi) \ge \ell_p$。

假设 4.4 描述了以下两种模型在某个数据分布下的期望损失对比：1）在该分布上适应的模型；2）在另一个分布上适应的模型。这实质上意味着，从期望上看，适应后的损失小于另一个任务模型的损失，这对于应用基于阈值的检测方案至关重要。

在每个任务 t结束后，使用在线梯度下降 [32] 根据元损失 $\left\|\phi_t^0 - \phi_t^*\right\|^2$ 更新元模型，我们可以得到以下定理，用以刻画任务边界检测不确定性下的期望 TAR。

定理 4.5。 假设假设 *4.1*、*4.2*、*4.3* 和 *4.4* 均成立。设 $R = \sum_{t=1}^{T} K_t$ 为在线轮次总数， $S = c$对数 R 为每一步用于适应的数据点数量，其中 c 为某个正常数。则期望 *TAR* 的上界为

$$\mathbb{E}[\mathbf{R}_{\text{avg}}] \le \mathcal{O}\left(\frac{\sigma_*^2 + \frac{\log T}{T}}{1-\rho^2} + R^{-\left(\frac{c(\ell_p-\ell_m)^2}{2M^2} - 2\right)}\right),$$

其中，期望是关于任务边界检测不确定性取的。 $\sigma_*^2 = \frac{1}{T}\sum_{t=1}^{T}\left\|\phi_t^* - \phi^*\right\|^2$ 和$\phi^* = \frac{1}{T}\sum_{t=1}^{T}\phi_t^*$分别表示比较器 $\{\phi_t^*\}_{t=1}^T$的方差和均值。特别地，通过选择 $c > \frac{4M^2}{(\ell_p-\ell_m)^2}$，可以实现恒定的期望 *TAR*。

直观而言，定理 4.5 上界中的第一项刻画了已知任务边界时在线元学习的 TAR，而第二项则表征了任务边界检测不确定性的影响。如定理 4.5 所示，随着任务变得更加相似，第一项会减小，因为方差 σ_*^2 较小；而第二项可能增大，因为常数 ℓ_p 和 ℓ_m 变得更加接近（也就是说，任务越相似，就越难检测任务切换）。因此，定理 4.5 揭示了任务相似性对标准在线元学习性能的影响与任务边界检测不确定性下的性能之间的权衡。在实践中，最优动作 ϕ_t^* 通常无法用于元更新。因此，我们的算法采用了类似 MAML 的替代更新方式，该方式已被证明有助于学习元参数。正如下一节的各项实验所展示的，这类更新确实可以作为一种良好的实用替代方案。

5. 实验

实验设置。 我们在三个标准的在线元学习基准测试上考察 LEEDS 的性能，并将其与多种相关基线算法进行比较。这三个基准测试分别是 Omniglot-MNIST-FashionMNIST、Tiered-ImageNet 和 Synbols。具体而言，我们先在一个领域对元模型进行预训练，再将其部署到一个任务可能来自新领域的动态环境中。我们使用整个在线学习阶段所获得的测试损失的平均值来评估所有算法。为考察非平稳程度对学习性能的影响，我们进一步考虑环境非平稳性的两种不同情况：中等平稳的情况，即不切换到新任务的概率设为 $p = 0.9$；以及低平稳的情况，即 $p = 0.75$。我们不考虑 p 非常小的情况，因为在这种情况下，仅假设每一轮都会发生任务切换的算法应该表现良好。

除非另有说明，我们在 10000 个回合上比较各算法。关于数据集和基线方法的更多详情，请参见附录 C。附录 B 还提供了不同方法更全面的结果和训练曲线。

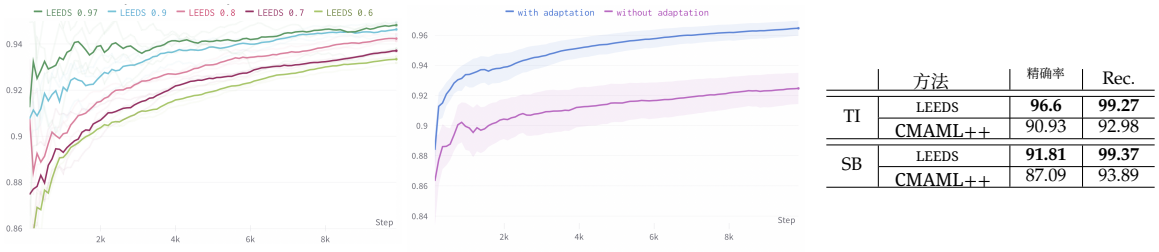
在所有实验中，每当需要揭示新任务时，该任务将以 0.5 的概率从预训练域中抽取，或以 0.5 的概率从某个分布外领域中抽取。对于 Tiered-ImageNet，由于相对于预训练任务分布而言，只有 ood2 真正属于分布外，我们将 ood2 的抽样概率提高到 0.5，这与我们所有实验中针对分布内任务和分布外任务所采用的 50% – 50% 协议一致。有关实验设置的更多细节，包括神经网络架构和超参数搜索，将在附录 D 中介绍。

表1：在不同非平稳性水平下，Omniglot-MNIST-FashionMNIST 基准上 10000 个在线回合的平均准确率。“预训练”域：**Omniglot**；“ood1”域：**MNIST**；“ood2”域：**FashionMNIST**。与其他基线方法相比，我们的 LEEDS 算法在分布外领域中的优势更加显著。有关数据集的更多细节，请参见附录 C。

方法	非平稳性水平 $p = 0$ .9			非平稳性水平 $p = 0$ .75		
	预训练	ood1	ood2	预训练	ood1	ood2
LEEDS	**99.39** ±0.09	**96.44** ±0.11	**82.87** ±0.19	**98.97** ±0.10	**95.68** ±0.12	**81.49** ±0.22
CMAML++	98.78 ±0.12	92.52 ±0.19	76.16 ±0.28	97.39 ±0.11	89.07 ±0.20	73.35 ±0.35
CMAML	89.79 ±0.54	84.06 ±0.80	69.70 ±0.63	75.51 ±0.94	70.41 ±1.22	58.58 ±1.27
FOML	89.20 ±0.61	70.84 ±0.76	64.83 ±0.74	81.68 ±0.59	59.24 ±0.78	58.07 ±0.77
MAML	95.07 ±0.10	62.02 ±0.14	54.67 ±0.17	95.51 ±0.11	62.31 ±0.13	54.83 ±0.13
ANIL	96.54 ±0.12	42.14 ±0.16	40.12 ±0.13	96.88 ±0.11	42.08 ±0.11	40.00 ±0.13
MetaOGD	84.05 ±1.66	73.73 ±1.39	60.03 ±1.60	85.67 ±1.57	75.09 ±1.43	60.51 ±1.65
BGD	63.58 ±2.25	46.12 ±2.10	44.89 ±1.41	23.86 ±2.36	17.97 ±2.17	19.81 ±1.63
MetaBGD	77.73 ±1.26	59.11 ±0.84	54.67 ±0.82	43.95 ±1.31	27.87 ±0.94	30.14 ±0.98

5.1. 主要结果

Omniglot-MNIST-FashionMNIST（OMF）上的结果。表1展示了在非平稳程度为 $p = 0.9$ 和 $p = 0.75$时各对比方法的在线评估。对于每种设置，我们分别报告预训练域上的在线准确率以及其他两个分布外域上的在线准确率，以展示我们的方法如何在不断提升分布外域表现的同时，记住预训练任务。显然，在这两种设置下，我们的方法 LEEDS 的表现都优于所有其他基线算法。具体而言，在分布内域上，所有使用 MAML 进行预训练的方法表现相近，但 LEEDS 和 CMAML++ 表现更优，因为它们能够检测任务边界。然而，在分布外域上，我们的算法显著优于所有其他基线方法，包括 CMAML++。这归功于关键的分布外适应模块，它使 LEEDS 能够根据任务分布动态调整元模型。有趣的是，对比 MAML 和 ANIL 的表现，可以深入了解在非平稳环境中重复利用预训练表示的局限性。事实上，不调整其内层表示的 ANIL 基线，在分布外域上的表现不如 MAML，但在预训练域上取得了相近的结果。此外，结果也凸显了近期提出的 FOML [10] 方法的一些局限性，其表现低于其他有竞争力的基线方法。这是因为 FOML 要求各任务不能互斥，而这对于我们实验中考虑的标准少样本基准测试而言，未必成立。



	方法	精确率	Rec.
TI	LEEDS	**96.6**	**99.27**
	CMAML++	90.93	92.98
SB	LEEDS	**91.81**	**99.37**
	CMAML++	87.09	93.89

图2：**左：** 不同 p 下的 LEEDS。**中：** 有域适应和无域适应的 LEEDS。**右：** 在**Tiered-ImageNet (TI)** 和 **Synbols (SB)**上的任务边界检测。

Tiered-ImageNet（TI）和 Synbols（SB）上的结果。 我们在表3中报告了这两个基准测试在所有领域和分布外领域上的在线准确率。由于预训练任务的分布与 Tiered-ImageNet 基准测试中的分布外任务相似，

表 2：在不同非平稳性水平下，分别在 **Tiered-ImageNet** 基准和**Synbols** 基准上进行 20000 次和 10000 次在线回合时的平均准确率。不同领域是原始 Tiered-ImageNet 数据集的不同划分（有关这些划分的获取方式，请参阅附录 C 中的实验设置）。在 **Synbols** 中，“预训练”领域对应于 Synbols 数据集中的 3 个不同字母表，“ood1”对应于 Synbols 数据集中一个新的（相对于预训练所用字母表而言）字母表，而“ood2”包含字体分类任务。包含方差的完整表格见附录 B。

方法	**Tiered-ImageNet**				**Synbols**			
	非平稳性 $p = 0.9$		非平稳性 $p = 0.75$		非平稳性 $p = 0.9$		非平稳性 $p = 0.75$	
	all	ood	all	ood	all	ood	all	ood
LEEDS	**66.07**	**67.43**	**64.52**	**65.80**	**85.12**	**67.48**	**82.22**	**63.68**
CMAML++	63.83	63.75	61.28	61.96	81.14	62.39	79.74	60.70
FOML	35.90	35.87	32.02	31.61	46.40	41.73	37.46	34.13
MAML	62.37	61.00	62.54	60.88	76.25	42.70	74.87	43.84
ANIL	59.78	57.61	59.57	57.38	64.58	34.51	72.69	35.66
MetaOGD	57.01	57.32	56.80	56.94	72.04	46.69	67.93	42.66
BGD	40.95	41.44	35.48	35.97	25.63	25.61	27.53	27.17
MetaBGD	49.21	50.01	44.58	45.30	53.74	42.25	40.79	34.63

因此，MAML 等方法能够取得合理的表现。事实上，在较低的非平稳程度（$p = 0.75$）下，MAML 能够超越更复杂的 CMAML++ 基线。然而，在两种非平稳程度下以及两个基准测试中，我们的算法仍然取得了最佳性能。需要注意的是，在较大的**TI** 数据集情形下，FOML 算法会存储所有此前见过的任务，并在大约 6500 个在线回合后耗尽内存。由于分布外任务与分布内任务在**TI**基准测试中具有相似性，ANIL 学习到的静态表示对所有领域都很有用。

5.2. 消融研究

任务边界检测。 图2右侧的表格给出了我们的方法和 CMAML++ 所采用的任务切换检测方案的精确率和召回率。我们的检测方案在所有指标上均优于 CMAML++ 。这是因为，CMAML++ 中的检测方案基于比较连续损失，这可能导致过度检测任务边界，尤其是在在线算法首次发现任务时，该任务的损失过高的情况下。

域适应模块的重要性。 我们研究了分布偏移检测模块的重要性，该模块使 LEEDS 算法能够针对分布内任务和分布外任务采用不同方式更新元模型。图2展示了我们的算法在有无分布偏移检测模块时的性能。使用该模块后，算法性能显著提升（提升了 $\sim 4.3\%$）。这表明，这种简单的机制能够有效提升在线学习性能，使智能体在从分布外数据中学习更多内容的同时，也能记住预训练知识。

对任务切换频率的敏感性。图 2 展示了我们的算法在任务切换概率 p 取不同值时的性能。随着 p 增大，性能也随之提升，这表明我们的 LEEDS 算法能够成功复用先前的任务知识，从而提高性能。

6. 结论

本文研究了在未知任务边界的非平稳环境中的在线元学习问题。为解决其中的问题，我们提出了 LEEDS，以高效更新元模型和在线任务模型。具体而言，LEEDS 基于两种简单而有效的任务切换和分布偏移检测机制，能够高效复用可用的最佳任务模型，而无需重置为元模型；同时区分针对分布内任务和分布外任务的元模型更新，从而在保留预训练分布旧知识的同时，快速学习新分布中的新知识。具体而言，LEEDS 中的元模型更新仅基于当前数据，无需存储先前数据。大量实验表明，在多个基准测试中，LEEDS 的表现优于相关基线方法。

致谢

本研究部分经费由美国国家科学基金会资助，资助项目编号为 DMS- 2134145、ECCS-2113860 和 CNS-2112471。

参考文献

[1] Gregory Koch、Richard Zemel、Ruslan Salakhutdinov 等。用于单样本图像识别的孪生神经网络。载于*ICML*深度学习研讨会，第2卷，第0页。里尔，2015年。[2] Sachin Ravi 和 Hugo Larochelle。将优化作为小样本学习的模型。载于国际学习表征会议（*ICLR*），2016年。[3] 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论文集（*ICML*），第1126–1135页，2017年。[4] James Hannan。重复博弈中贝叶斯风险的近似。博弈论文集，3(2):97–139，1957年。[5] Shai Shalev-Shwartz 和 Yoram Singer。在线学习：理论、算法与应用。2007年。[6] Nicolo Cesa-Bianchi 和 Gábor Lugosi。预测、学习与博弈。剑桥大学出版社，2006年。[7] James Harrison、Apoorva Sharma、Chelsea Finn 和 Marco Pavone。无任务的连续元学习。神经信息处理系统进展，33:17571–17581，2020年。[8] Huaxiu Yao、Yingbo Zhou、Mehrdad Mahdavi、Zhenhui Jessie Li、Richard Socher 和 Caiming Xiong。在线结构化元学习。神经信息处理系统进展，33:6779–6790，2020年。[9] Durmus Alp Emre Acar、Ruizhao Zhu 和 Venkatesh Saligrama。内存高效的在线元学习。载于国际机器学习会议，第32–42页。机器学习研究论文集，2021年。[10] Jathushan Rajasegaran、Chesea Finn 和 Sergey Levine。无任务边界的完全在线元学习。*arXiv*预印本 *arXiv:2202.00263*，2022年。[11] Massimo Caccia、Pau Rodríguez López、Oleksiy Ostapenko、Fabrice Normandin、Min Lin、Lucas Page-Caccia、Issam H. Laradji、Irina Rish、Alexandre Lacoste、David Vázquez 和 Laurent Charlin。在线快速适应与知识积累（osaka）：一种持续学习的新方法。载于*NeurIPS*，2020年。[12] Weitang Liu、Xiaoyun Wang、John Owens 和 Yixuan Li。基于能量的分布外检测。神经信息处理系统进展，33:21464–21475，2020年。[13] Chelsea Finn、Aravind Rajeswaran、Sham Kakade 和 Sergey Levine。在线元学习。载于国际机器学习会议，第1920–1930页。机器学习研究论文集，2019年。[14] Oriol Vinyals、Charles Blundell、Timothy Lillicrap 和 Daan Wierstra。用于单样本学习的匹配网络。载于神经信息处理系统进展（*NIPS*），2016年。[15] Zhenguo Li、Fengwei Zhou、Fei Chen 和 Hang Li。Meta-SGD：为小样本学习而快速学会学习。*arXiv*预印本 *arXiv:1707.09835*，2017年。[16] Chelsea Finn 和 Sergey Levine。元学习与普适性：深度表征和梯度下降可以近似任意学习算法。*arXiv*，abs/1710.11622，2018年。

[17] Chelsea Finn、Kelvin Xu 和 Sergey Levine。概率模型无关元学习。神经信息处理系统进展，31，2018年。[18] Aniruddh Raghu、Maithra Raghu、Samy Bengio 和 Oriol Vinyals。快速学习还是特征复用？探索 MAML 有效性的成因。国际学习表征会议（*ICLR*），2019年。[19] Aravind Rajeswaran、Chelsea Finn、Sham M Kakade 和 Sergey Levine。使用隐式梯度进行元学习。载于神经信息处理系统进展（*NeurIPS*），第113–124页，2019年。[20] Alex Nichol 和 John Schulman。Reptile：一种可扩展的元学习算法。*arXiv*：学习，2018年。[21] Alex Nichol、Joshua Achiam 和 John Schulman。一阶元学习算法。*arXiv*预印本 *arXiv:1803.02999*，2018年。[22] Fei Mi、Minlie Huang、Jiyong Zhang 和 Boi Faltings。面向任务导向型对话系统的低资源自然语言生成元学习。*arXiv*预印本 *arXiv:1905.05644*，2019年。[23] Pan Zhou、Xiaotong Yuan、Huan Xu、Shuicheng Yan 和 Jiashi Feng。通过小批量近端更新实现高效元学习。神经信息处理系统进展，32，2019年。[24] Adam Santoro、Sergey Bartunov、Matthew Botvinick、Daan Wierstra 和 Timothy Lillicrap。使用记忆增强神经网络进行元学习。载于国际机器学习会议，第1842–1850页。机器学习研究论文集，2016年。[25] Luca Bertinetto、Joao F Henriques、Philip Torr 和 Andrea Vedaldi。使用可微闭式求解器进行元学习。载于国际学习表征会议（*ICLR*），2018年。[26] Tsendsuren Munkhdalai 和 Hong Yu。元网络。载于国际机器学习会议，第2554–2563页。机器学习研究论文集，2017年。[27] Jake Snell、Kevin Swersky 和 Richard Zemel。用于小样本学习的原型网络。载于神经信息处理系统进展（*NIPS*），2017年。[28] Flood Sung、Yongxin Yang、Li Zhang、Tao Xiang、Philip HS Torr 和 Timothy M Hospedales。学会比较：用于小样本学习的关系网络。载于*IEEE*计算机视觉与模式识别会议论文集，第1199–1208页，2018年。[29] Elad Hazan、Amit Agarwal 和 Satyen Kale。用于在线凸优化的对数遗憾算法。机器学习，69(2):169–192，2007年。[30] Jacob D Abernethy、Elad Hazan 和 Alexander Rakhlin。暗中竞争：一种高效的赌博机线性优化算法。2009年。[31] Shai Shalev-Shwartz 等。在线学习与在线凸优化。基础与趋势® 机器学习，4(2):107–194，2012年。[32] Martin Zinkevich。在线凸规划与广义无穷小梯度上升。载于第*20*届国际机器学习会议论文集（*ICML-03*），第928–936页，2003年。[33] Tianhe Yu、Xinyang Geng、Chelsea Finn 和 Sergey Levine。面向在线元学习的可变样本数适应。*arXiv*预印本 *arXiv:2012.07769*，2020年。[34] Maria-Florina Balcan、Mikhail Khodak 和 Ameet Talwalkar。基于梯度的元学习的可证明保证。载于 K. Kamalika Chaudhuri 和 Ruslan Salakhutdinov（编），第*36*届国际机器学习会议论文集，机器学习研究论文集第97卷，第424–433页。PMLR，2019年6月9日至15日。网址：https://proceedings.mlr.press/v97/balcan19a.html。

[35] Mikhail Khodak、Maria-Florina F Balcan 和 Ameet S Talwalkar。自适应基于梯度的元学习方法。神经信息处理系统进展，32，2019年。[36] Giulia Denevi、Dimitris Stamos、Carlo Ciliberto 和 Massimiliano Pontil。在线中的在线元学习。神经信息处理系统进展，32，2019年。[37] Elad Hazan 等人。在线凸优化导论。优化基础与趋势® 优化，2(3-4):157–325，2016年。[38] Michael McCloskey 和 Neal J Cohen。联结主义网络中的灾难性干扰：序列学习问题。载于学习与动机心理学，第24卷，第109–165页。Elsevier，1989年。[39] Roger Ratcliff。识别记忆的联结主义模型：学习与遗忘函数所施加的限制。心理学评论，97(2):285，1990年。[40] David Lopez-Paz 和 Marc'Aurelio Ranzato。用于持续学习的梯度情景记忆。神经信息处理系统进展，30，2017年。[41] Arslan Chaudhry、Marc'Aurelio Ranzato、Marcus Rohrbach 和 Mohamed Elhoseiny。借助 A-GEM 实现高效终身学习。*arXiv* 预印本 *arXiv:1812.00420*，2018年。[42] Matthew Riemer、Ignacio Cases、Robert Ajemian、Miao Liu、Irina Rish、Yuhai Tu 和 Gerald Tesauro。通过最大化迁移并最小化干扰，实现不遗忘的学会学习。*arXiv* 预印本 *arXiv:1810.11910*，2018年。[43] James Kirkpatrick、Razvan Pascanu、Neil Rabinowitz、Joel Veness、Guillaume Desjardins、Andrei A Rusu、Kieran Milan、John Quan、Tiago Ramalho、Agnieszka Grabska-Barwinska 等人。克服神经网络中的灾难性遗忘。美国国家科学院院刊，114(13):3521–3526，2017年。[44] Rahaf Aljundi、Francesca Babiloni、Mohamed Elhoseiny、Marcus Rohrbach 和 Tinne Tuytelaars。记忆感知突触：学习该忘记什么（以及不该忘记什么）。载于欧洲计算机视觉会议（*ECCV*）论文集，第139–154页，2018年。[45] 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。[46] Jaehong Yoon、Eunho Yang、Jeongtae Lee 和 Sung Ju Hwang。利用动态可扩展网络进行终身学习。载于国际学习表征会议，2018年。[47] Syed Shakib Sarwar、Aayush Ankit 和 Kaushik Roy。利用部分网络共享，在深度卷积神经网络中进行增量学习。*IEEE Access*，8:4615–4628，2019年。[48] Chen Zeno、Itay Golan、Elad Hoffer 和 Daniel Soudry。使用在线变分贝叶斯进行与任务无关的持续学习。*arXiv* 预印本 *arXiv:1803.10123*，2018年。

补充材料

W我们提供正文中省略的详细内容。各节安排如下 :

- 附录 B：我们提供了更多实验结果，包括所有设置下阈值的敏感性分析和在线评估曲线。

- 附录 C：我们进一步介绍数据集和基线方法的详细信息。
- 附录 D：我们进一步介绍实验规范，并讨论用于设定阈值的启发式方法。

- 附录 F：我们给出定理 4.5 的证明。

A. 关于持续学习的更多相关工作

持续学习。 持续学习（CL，又称终身学习）旨在克服从一系列非平稳数据分布中学习时出现的“灾难性遗忘” [38, 39] 。现有方法包括基于回放的方法 [40–42], 基于正则化的方法 [43, 44], 和基于扩展的方法[45–47]。例如，基于回放的方法会存储先前任务数据的一个子集，并重新使用这些数据进行经验“回放”，以避免遗忘。然而，传统的持续学习方法会在所有已见任务上评估最终模型，以衡量遗忘程度。在这项工作中，我们关注在线元学习，并通过适应后的平均在线性能（例如准确率）来评估模型，因为这能更好地反映模型快速适应新在线任务 [11]的能力。尽管我们并未专门关注如何避免遗忘，但我们以一种既能保留分布内领域知识、又能提升模型对分布外领域快速适应能力的方式更新元模型，这一点已在我们的多项实验中得到验证。

B. 更多实验结果

B.1. 准确率 Tiered-ImageNet 和 Synbols 数据集的方差准确率表 s

方法	非平稳性 $p = 0.9$		非平稳性 $p = 0.75$	
	所有领域	分布外领域	所有领域	分布外领域
LEEDS	**66.07** ±0.24	**67.43** ±0.38	**64.52** ±0.17	**65.80** ±0.31
CMAML++	63.83 ±0.27	63.75 ±0.55	61.28 ±0.23	61.96 ±0.41
FOML	35.90 ±0.56	35.87 ±0.83	32.02 ±0.42	31.61 ±0.69
MAML	62.37 ±0.46	61.00 ±0.72	62.54 ±0.37	60.88 ±0.65
ANIL	59.78 ±0.21	57.61 ±0.38	59.57 ±0.22	57.38 ±0.36
MetaOGD	57.01 ±0.28	57.32 ±0.66	56.80 ±0.25	56.94 ±0.42
BGD	40.95 ±0.85	41.44 ±1.15	35.48 ±0.76	35.97 ±1.09
MetaBGD	49.21 ±1.05	50.01 ±1.25	44.58 ±1.12	45.30 ±1.20

表3：在不同非平稳性水平下，Tiered-ImageNet 基准上的 20000 个在线回合的平均准确率。不同领域是原始 Tiered-ImageNet 数据集的不同划分（关于这些划分的具体方式，请参阅第 5 节的实验设置）。

B.2. 对阈值 ℓ 和 τ 以及温度 δ的敏感性

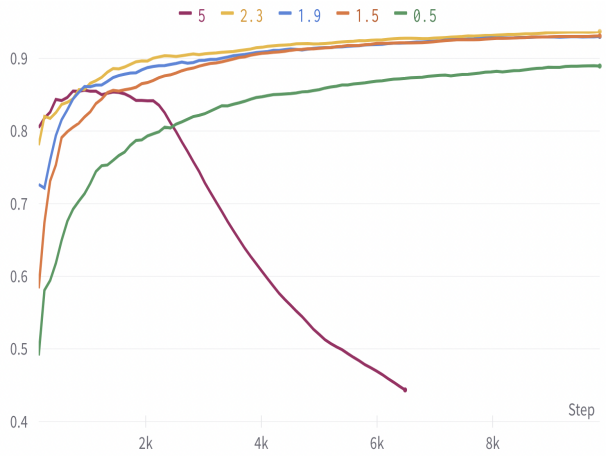
图 3 和图 4 展示了我们的 LEEDS 算法对基于能量的检测模块中阈值 ℓ 和τ 以及温度参数 δ 的敏感性。如图 3 所示，当阈值 ℓ 过小时，算法往往会过度检测任务切换（如表中 $\ell = 0.5$ 的召回率较低所示），这会导致算法性能下降，因为

方法	非平稳性 $p = 0.9$		非平稳性 $p = 0.75$	
	所有领域	分布外领域	所有领域	分布外领域
LEEDS	**85.12** ±0.91	**67.48** ±0.97	**82.22** ±0.32	**63.68** ±0.36
CMAML++	81.14 ±1.05	62.39 ±1.00	79.74 ±1.07	60.70 ±1.12
FOML	46.40 ±0.61	41.73 ±0.73	37.46 ±0.27	34.13 ±0.31
MAML	76.25 ±0.63	42.70 ±0.68	74.87 ±0.42	43.84 ±0.45
ANIL	64.58 ±0.32	34.51 ±0.54	72.69 ±0.30	35.66 ±0.49
MetaOGD	72.04 ±0.67	46.69 ±0.77	67.93 ±0.59	42.66 ±0.62
BGD	25.63 ±0.07	25.61 ±0.09	27.53 ±0.08	27.17 ±0.11
MetaBGD	53.74 ±0.41	42.25 ±0.52	40.79 ±0.23	34.63 ±0.33

表 4：在不同非平稳程度下，Synbols 基准上 10000 个在线回合的平均准确率。“预训练”域对应于 Synbols 数据集中的 3 种不同字母表；“ood1”对应于 Synbols 数据集中一种新的字母表（相对于预训练所用的字母表）；“ood2”包含字体分类任务。

任务知识未能得到有效复用。另一方面，当 ℓ 过大时，较高的误检率（例如，$\ell = 5$的精确率较低所示）会导致算法主要使用当前任务支持数据对在线任务模型 ϕ_t 进行微调。正如预期，这会因不同任务的对抗性而导致一种失效模式（算法发散）。我们发现，ℓ 在范围[1.5 到 2.3] 内时，算法性能最佳。

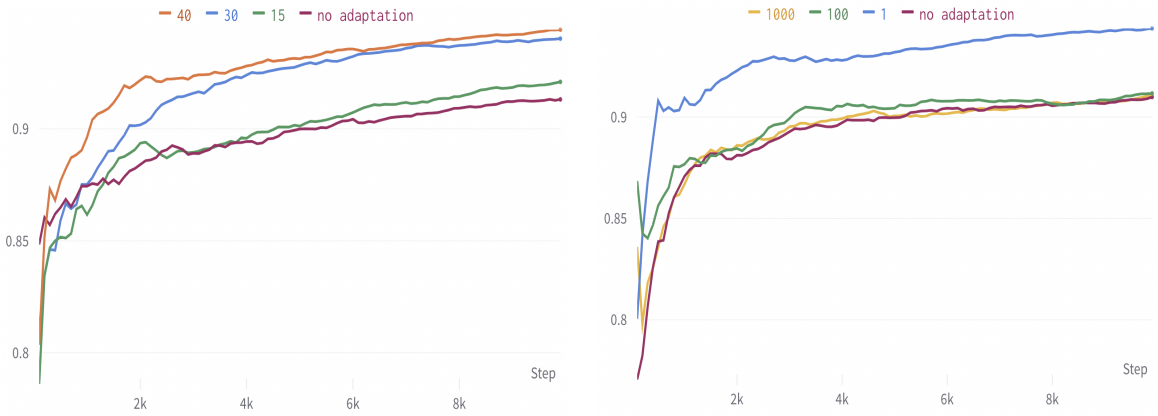
图 4 (a) 显示，较大的 τ 值会使元模型在每一步都进行更新（即使对于预训练任务分布也是如此），但这并不能显著提升性能。这表明，针对分布内和分布外任务提出的不同元更新方案具有优势：它避免了对预训练任务进行不必要的频繁元更新，从而能够更合理地使用计算预算。较小的 τ 值（例如 $\tau = 15$）往往会将所有任务分布都判定为预训练任务分布，这相当于取消了算法中的域适应组件。我们还发现，在能量表达式中直接设置温度 $\delta = 1$ 可获得最佳性能，而较大的 δ 值也会使基于能量的检测模块失效（图 4 (b)）。这实际上与 [12], 中的发现一致，该发现也建议设置 $\delta = 1$。



ℓ	精确率	召回率
$\ell = 0.5$	99.9	68.0
$\ell = 1.5$	99.9	98.4
$\ell = 1.9$	99.9	99.3
$\ell = 2.3$	99.9	99.7
$\ell = 5$	38.13	99.9

图 3：LEEDS 在阈值 ℓ 取不同值时的性能。**左图：**在线学习过程中在所有已遇到领域上的性能。**右表：**不同 ℓ 值下的任务边界检测。实验在 Omniglot-MNIST-FashionMNIST 基准上进行。

图5-8展示了所有设置下不同方法的在线评估曲线。不同图表进一步表明，我们的 LEEDS 算法在分布外领域优于其他基线算法，同时也保留了在预训练任务上的性能。



(a) 对能量阈值 τ的敏感性　　(b) 对温度尺度的敏感性

图4：LEEDS 在以下情况下的性能：(a) 不同能量阈值 τ 和 (b) 不同温度尺度 δ。对于两幅图，我们报告在线学习期间所有遇到的领域上的性能。实验在 Omniglot-MNIST-FashionMNIST 基准上进行。

当分布外任务与预训练任务相距甚远时，在在线学习阶段不调整元参数的方法（例如 MAML 和 ANIL）的性能会大幅下降（如表1中的**Omniglot-MNIST-FashionMNIST**所示）。对于分布外任务与预训练任务较为接近的设置（例如**Tiered-ImageNet** 数据集），MAML 的表现可与 C MAML++相近。

即使在预训练域上，LEEDS 仍表现出色，这尤其表明，重复利用任务知识有益于在线元学习；与之相反，通常的做法是在每一步都“重置”为元参数。

比较图10中的两幅图可以看出，当分布外域与预训练域相距较远时，我们的域适应模块优势更加显著；与 Omniglot 预训练域相比，FashionMNIST 分布外域便是如此。

C. F关于数据集和基线方法的进一步描述

C.1. 数据集

我们在以下基准测试上研究动态在线元学习：

Omniglot-MNIST-FashionMNIST 数据集。 对于此数据集，我们考虑 10-way 5-shot 分类任务。我们在 Omniglot 数据集的一个子集上预训练元模型，然后将其部署到在线学习环境中，在该环境中，任务从完整的 Omniglot 数据集或某个分布外数据集（即 MNIST 或 FashionMNIST 数据集）中采样。

Tiered-ImageNet 数据集。 我们针对该数据集考虑 5-way 5-shot 分类任务。遵循[cite paper], , 我们将原始 Tiered-ImageNet 数据集划分为预训练域和分布外领域。具体而言，我们使用原始训练集中的前 200 个类别进行预训练，将剩余类别作为第一个分布外域（ood1），并将原始测试类别设为第二个分布外域（ood2）。在线学习阶段，我们将完整训练集设为分布内域，并将 ood1 和 ood2 设为分布外领域。我们在 20000 个在线学习回合上评估算法。

Synbols 数据集。 我们考虑该数据集中的 4-way 4-shot 分类任务。元模型在来自 3 种不同字母表的字符上进行预训练，并部署于来自新字母表（ood1）的字符上。我们还将字体分类任务作为额外的分布外任务（ood2）进行考察。

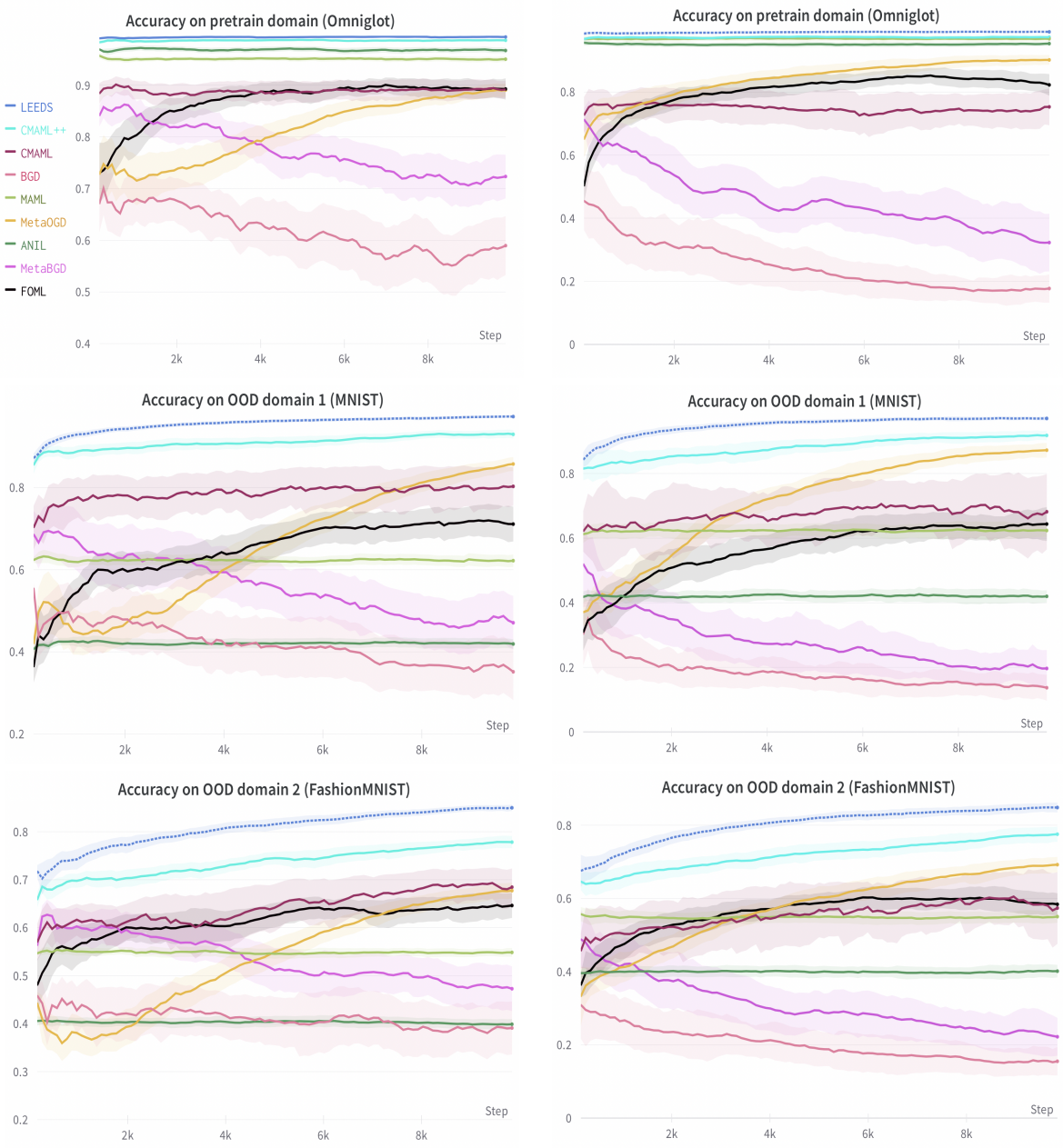


图5：在**Omniglot-MNIST-FashionMNIST** 基准中，在线学习阶段在每个遇到的领域上的在线评估。第一列对应非平稳性水平$p = 0.9$。第二列中 $p = 0.75$。LEEDS 是唯一一种既能保留预训练知识，又能显著提升分布外领域表现的方法。图例仅显示在第一个图中。

C.2. 基线方法

W我们将算法与以下在线元学习基线方法进行比较 g.

(1) C-MAML [11] 和 C-MAML++： 持续学习 MAML 方法（C-MAML）使用 MAML 预训练元模型，并采用基于任务边界检测的在线学习策略。由于 C-MAML 不会在单独的查询集上评估任务模型，为了进行公平比较，我们对其进行调整，使其能够这样做，并将由此得到的算法称为 C-MAML++。

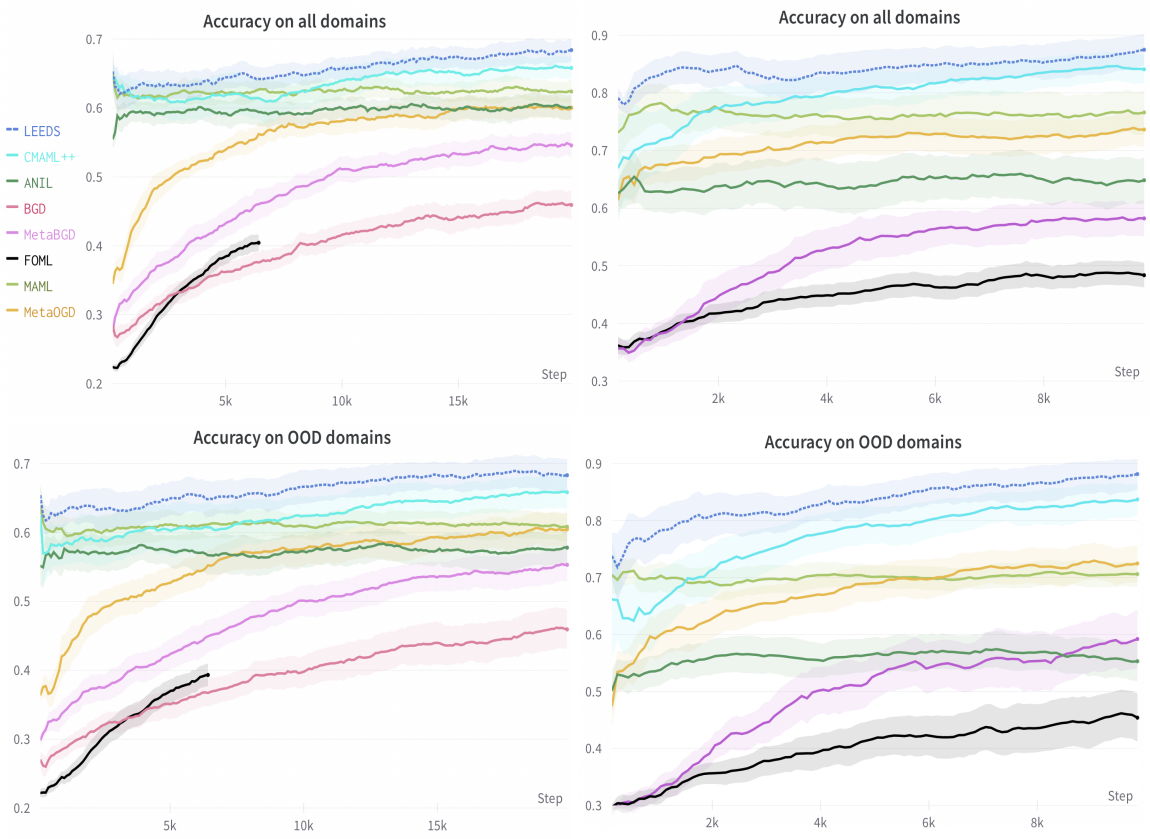


图 6：在 p =0.9 条件下，**Tiered-ImageNet (TI)** 和**Synbols (SB)** 基准测试的在线评估结果。第一列对应**TI**，第二列对应**SB**。更多结果，包括各个领域中的表现以及不同 p 条件下的表现，请参见附录 B。图例仅显示在第一个图中。

(2) FOML [10]： 完全在线元学习方法使用最新的在线数据更新在线参数，并维持一个并行的元训练过程，以指导受元模型正则化的在线更新。

(3) MAML [3] 和 ANIL [18]： MAML 基线由离线预训练阶段和在线部署阶段组成。在离线预训练期间，该方法会学习一个供所有任务使用的通用元初始化，并且在在线阶段不会更新该元初始化。任务模型从元初始化出发，利用支持集进行适配，并在查询集上进行评估。ANIL 与 MAML 类似，但只进行部分参数适配，即针对每个任务仅适配最后一层。

(4) MetaBGD [11] 和 BGD [48]： 基线方法 MetaBGD 在在线学习期间结合了 MAML 和贝叶斯梯度下降方法。

(5) MetaOGD [32]： 元在线梯度下降方法在每一步中，都会使用基于当前任务数据计算的、类似 MAML 的元目标函数来更新元模型。

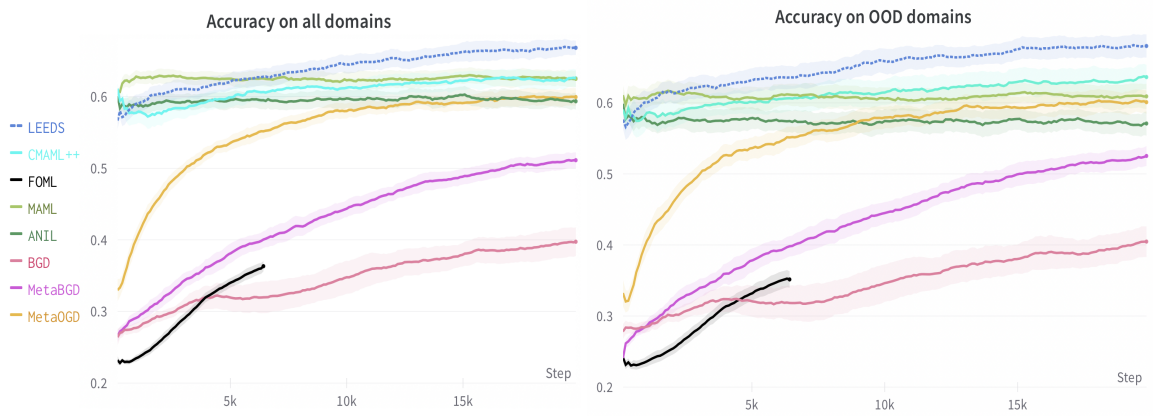


图 7：在 **Tiered-ImageNet(TI)** 基准上，针对 $\mathbf{p} = 0.75$ 进行的在线评估。**左：** 在线学习期间，在所有已遇到领域上的准确率。**右：** 在线学习期间，在所有已遇到的分布外领域上的准确率。我们在 16GB 图形处理器内存预算下比较所有基线方法；由于内存需求呈线性增长，FOML 在此基准上运行时内存不足。

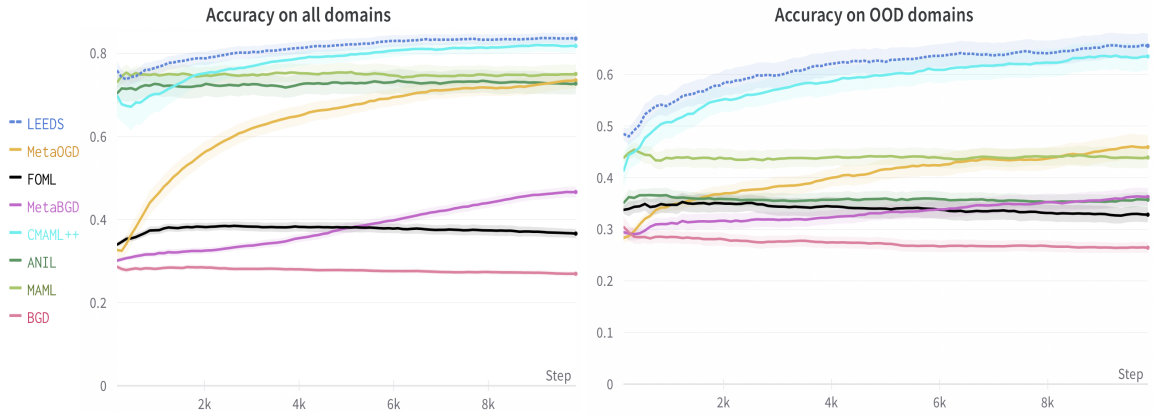


图 8：在**Synbols (SB)** 基准上，针对 $\mathbf{p} = 0.75$ 进行的在线评估。**左：** 在线学习期间，在所有已遇到领域上的准确率。**右：** 在线学习期间，在所有已遇到的分布外领域上的准确率。

D. 进一步的实验细节与超参数搜索

D.1. 进一步的实验规范

在所有实验中，我们都考虑分类任务。我们使用预测结果与真实标签之间的交叉熵损失来训练所有模型。我们采用小样本学习文献中广泛使用的相同卷积神经网络（CNN）架构 [3, 11, 13], ，其中包含四个卷积块，后接一个线性分类层。每个卷积块由一层3×3 卷积层以及后续的批归一化、ReLU 和 2×2 最大池化层组成。对于 **Omniglot-MNIST-FashionMNIST** 基准，我们在每个卷积层中使用 64 个滤波器，并将灰度图像下采样至 28×28 空间分辨率，使分类前的特征向量维度达到 64。对于 **Tiered-ImageNet** 和 **Synbols** 数据集，输入分别为$3 \times 64 \times 64$ 和 $3 \times 32 \times 32$ RGB图像；每个卷积层使用 64 个滤波器时，得到的特征向量维度分别为 1024 和 256。

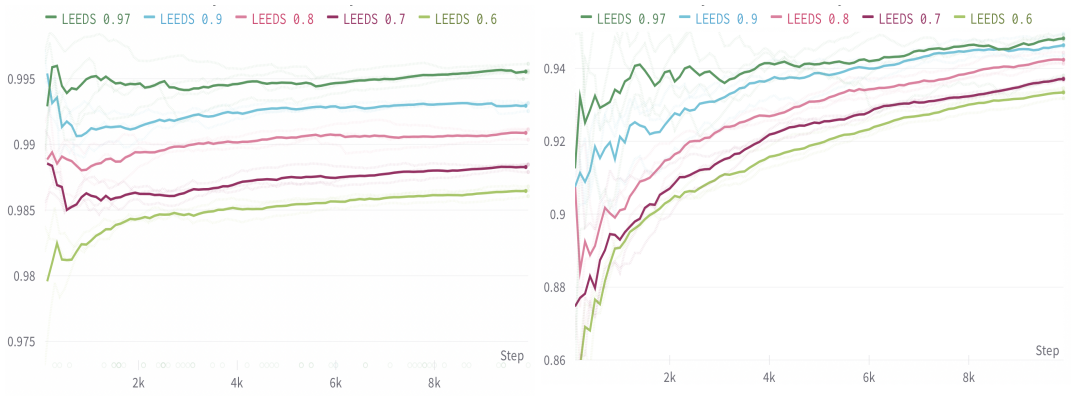


图9：我们的 LEEDS 算法在不同 p下的性能。**左图：**在预训练领域中的评估。**右图：**在所有领域中的评估。

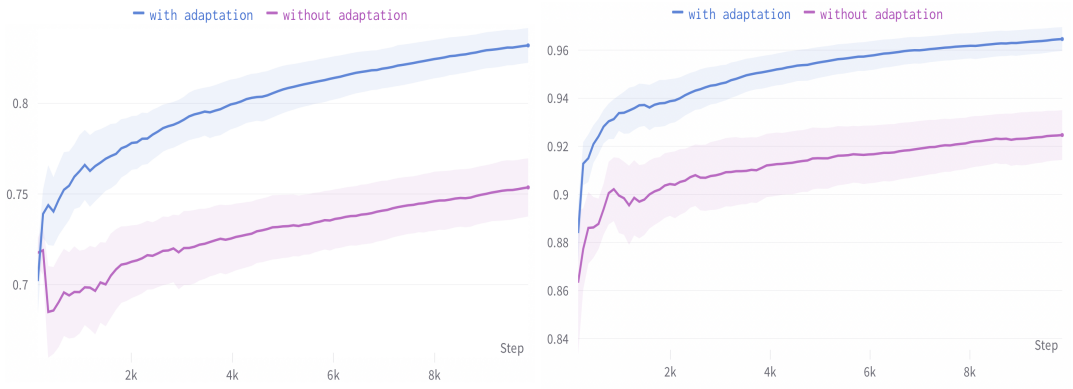


图10：LEEDS 在有无基于能量的域适应模块时的性能。**左图：**在分布外域 **FashionMINIST**中的评估。**右图：**在分布外域**MNIST**中的评估。

D.2. 更多实现细节

FOML [10] 方法的实现尚未发布，因此我们使用自己实现的该算法进行比较。对于所有其他基线方法，我们使用了其公开可用的实现。我们提交的补充材料中提供了 LEEDS 算法及所有其他基线方法的代码。所有代码均已在 Python 3.6 和 PyTorch 1.2 下测试。

例如，若要在 $p = 0.9$下使用我们针对 Omniglot-MNIST-FashionMNIST 数据集获得的最佳超参数运行 LEEDS 算法，可以运行以下命令：

python main . py ––算法需要 ––使用 _最佳 1

实验设置（例如要使用的数据集）可以在 configurations.yaml 文件中更改。我们在单个 NVIDIA Tesla P100 GPU 上运行所有方法。除 FOML 外，所有对比算法都能在 16GB 图形处理器内存上运行。对于 Tiered-ImageNet 数据集，FOML 至少需要 32GB 内存才能运行 12000 个在线回合。

E. 设置阈值的启发式方法

对于能量阈值 τ，我们遵循 Liu 等（2020）中的策略，即使用预训练任务设置阈值 τ。具体而言，我们设置 τ，使得 95% 的预训练输入都能被正确

检测为预训练数据。在标准的小样本学习实验设置中，每个任务的真实标签通常是随机选择的。发生任务切换时，从先前任务中学到的特定任务模型通常不再适用于新任务，其学习性能可能与随机模型相近。基于这一点，我们发现，选择阈值 ℓ 初始值的一种有效启发式方法是采用在随机模型上评估得到的损失值。例如，对于 10 类分类任务，该值为 $\ell_r = -\log(1/10) = 2.3$。

F. 定理 4.5 的证明

回顾任务平均遗憾:

$$\mathbf{R}_{\text{avg}} = \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{k=1}^{K_t} f_t^k\left(\phi_t^k\right) - \sum_{k=1}^{K_t} f_t^k\left(\phi_t^*\right)\right).$$

我们考虑这样一种情形：在每一轮中，智能体都可以在使用损失函数 f_t^k [13] 进行评估之前执行某种任务特定适应。令 U_t^k 为定义每轮适应过程的映射。例如，一种常见的映射函数 U_t^k 是一步梯度下降映射： $U_t^k(\phi) = \phi - \alpha\nabla\hat{} f_t^k(\phi)$，其中 α 是学习率，$\hat{} f_t^k$ 是损失函数 f_t^k 的近似，例如在任务 t 的一个小型支持集上计算得到。

将 $\hat{} f_r(\phi_{r-1}) = \frac{1}{|\mathcal{S}_r|}\sum_{\xi\in\mathcal{S}_r}\ell(\phi_{r-1},\ \xi) :=\hat{} f_{r,\ r-1}$ 记为第 r 轮的适应损失，其中 $\ell(\cdot,\ \xi)$ 是数据点 ξ 上的分类损失。我们首先给出检测错误的上界。

总　错误概率为 r 时，基于阈值的检测方案的总错误概率为　　y:

$$P_{\text{error}} = P\left(P_r \neq P_{r-1}\right) P\left(\hat{f}_{r,r-1} < \tau \,\middle|\, P_r \neq P_{r-1}\right) + P\left(P_r = P_{r-1}\right) P\left(\hat{f}_{r,r-1} > \tau \,\middle|\, P_r = P_{r-1}\right) \tag{6}$$

其中，第一项表示任务边界漏检的概率，第二项表示底层任务未发生变化时的误报概率。

根据假设 4.4，对于从 P_r 中独立同分布抽取的数据分布 P_r 和支持集 $\mathcal{S}_r$，由霍夫丁不等式可得:

$$P\left(\frac{1}{|\mathcal{S}_r|}\sum_{\xi\in\mathcal{S}_r}\ell(\phi_{r-1},\xi) - E_{\xi\sim P_r}\,\ell(\phi_{r-1},\xi) < -\epsilon\right) \leq \exp\left(\frac{-2|\mathcal{S}_r|\epsilon^2}{M^2}\right)$$

因此，

$$\begin{aligned}
&P\left(\hat{f}_{r,r-1} - \ell_p < -\epsilon \,\middle|\, P_r \neq P_{r-1}\right)\\
\leq &P\left(\frac{1}{|\mathcal{S}_r|}\sum_{\xi\in\mathcal{S}_r}\ell(\phi_{r-1},\xi) - E_{\xi\sim P_r}\,\ell(\phi_{r-1},\xi) < -\epsilon \,\middle|\, P_r \neq P_{r-1}\right)\\
\leq &\exp\left(\frac{-2|\mathcal{S}_r|\epsilon^2}{M^2}\right).
\end{aligned}$$

通过设置 $\epsilon = \frac{\ell_p - \ell_m}{2}$，我们可以得到

$$P\left(\hat{f}_{r,r-1} < \frac{\ell_m + \ell_p}{2} \,\middle|\, P_r \neq P_{r-1}\right) \leq \exp\left(\frac{-|\mathcal{S}_r|(\ell_p - \ell_m)^2}{2M^2}\right).$$

因此，设置阈值 $\tau = \frac{\ell_m+\ell_p}{2}$ 可得——

$$P\left(\hat{f}_{r,r-1} < \tau \,\middle|\, P_r \neq P_{r-1}\right) \leq \exp\left(\frac{-|\mathcal{S}_r|(\ell_p-\ell_m)^2}{2M^2}\right). \tag{7}$$

利用霍夫丁不等式的另一侧，我们有：

$$\begin{aligned}
&P\left(\hat{f}_{r,r-1} - \ell_m > \epsilon \,\middle|\, P_r = P_{r-1}\right) \\
\leq &P\left(\frac{1}{|\mathcal{S}_r|}\sum_{\xi\in\mathcal{S}_r}\ell(\phi_{r-1},\xi) - E_{\xi\sim P_r}\ell(\phi_{r-1},\xi) > \epsilon \,\middle|\, P_r = P_{r-1}\right) \\
\leq &P\left(\frac{1}{|\mathcal{S}_r|}\sum_{\xi\in\mathcal{S}_r}\ell(\phi_{r-1},\xi) - E_{\xi\sim P_r}\ell(\phi_{r-1},\xi) > \epsilon\right) \\
\leq &\exp\left(\frac{-2|\mathcal{S}_r|\epsilon^2}{M^2}\right).
\end{aligned}$$

因此，我们有：

$$P\left(\hat{f}_{r,r-1} > \frac{\ell_m+\ell_p}{2} \,\middle|\, P_r = P_{r-1}\right) \leq \exp\left(\frac{-|\mathcal{S}_r|(\ell_p-\ell_m)^2}{2M^2}\right),$$

因为 $\epsilon = \frac{\ell_p-\ell_m}{2}$——因此，我们得到

$$P\left(\hat{f}_{r,r-1} > \tau \,\middle|\, P_r = P_{r-1}\right) \leq \exp\left(\frac{-|\mathcal{S}_r|(\ell_p-\ell_m)^2}{2M^2}\right). \tag{8}$$

结合公式 (6)、(7) 和 (8)，并利用 $P(P_r \neq P_{r-1}) = 1-P(P_r = P_{r-1})$这一事实，可以得到当阈值设为 $\tau = \frac{\ell_m+\ell_p}{2}$ 时错误概率的以下上界：——

$$P_{\text{error}} \leq \exp\left(\frac{-|\mathcal{S}_r|(\ell_p-\ell_m)^2}{2M^2}\right). \tag{9}$$

根据假设 4.3，对于任意 ϕ，我们有：

$$f_t^k(\phi_t^k) \leq f_t^k(\phi) + \left\langle \nabla f_t^k(\phi), \phi_t^k - \phi \right\rangle + \frac{L}{2}\left\|\phi_t^k - \phi\right\|^2.$$

因此，将 ϕ 设为比较器 ϕ_t^*，并利用假设4.1，我们得到 ：

$$f_t^k(\phi_t^k) - f_t^k(\phi_t^*) \leq \frac{L}{2}\left\|\phi_t^k - \phi_t^*\right\|^2.$$

因此，对 $k = 1, ..., K_t$ 求和可得

$$\sum_{k=1}^{K_t} f_t^k(\phi_t^k) - \sum_{k=1}^{K_t} f_t^k(\phi_t^*) \leq \frac{L}{2}\sum_{k=1}^{K_t}\left\|\phi_t^k - \phi_t^*\right\|^2. \tag{10}$$

接下来，我们对项 $\sum_{k=1}^{K_t}\left\|\phi_t^k - \phi_t^*\right\|^2$ 给出上界。由此可得

$$\begin{aligned}
\left\|\phi_t^k - \phi_t^*\right\|^2 &= \left\|U_t^k(\phi_t^{k-1}) - U_t^k(\phi_t^*)\right\|^2 \\
&\leq \rho^2\left\|\phi_t^{k-1} - \phi_t^*\right\|^2 \\
&\leq \rho^{2k}\left\|\phi_t^0 - \phi_t^*\right\|^2,
\end{aligned} \tag{11}$$

其中，第一个不等式用到了假设4.2。因此，结合公式（10）和（11）可得

$$\begin{aligned}
\sum_{k=1}^{K_t} f_t^k(\phi_t^k) - \sum_{k=1}^{K_t} f_t^k(\phi_t^*) &\leq \frac{L}{2}\sum_{k=1}^{K_t}\rho^{2k}\left\|\phi_t^0 - \phi_t^*\right\|^2 \\
&\leq \frac{L}{2(1-\rho^2)}\left\|\phi_t^0 - \phi_t^*\right\|^2.
\end{aligned} \tag{12}$$

因此，对 $t=1,...,T$ 求和公式（12）可得

$$\sum_{t=1}^{T}\left(\sum_{k=1}^{K_t} f_t^k\left(\phi_t^k\right)-\sum_{k=1}^{K_t} f_t^k\left(\phi_t^*\right)\right) \leq \frac{L}{2(1-\rho^2)}\sum_{t=1}^{T}\left\|\phi_t^0-\phi_t^*\right\|^2.$$

因此，任务平均遗憾的上界如下

$$\mathbf{R}_{\text{avg}} \leq \frac{L}{2(1-\rho^2)T}\sum_{t=1}^{T}\left\|\phi_t^0-\phi_t^*\right\|^2. \tag{13}$$

.

接下来，我们证明项 $\frac{1}{T}\sum_{t=1}^{T}\left\|\phi_t^0-\phi_t^*\right\|^2$ 随着 $T\to\infty$ 收敛。我们有，

$$\frac{1}{T}\sum_{t=1}^{T}\left\|\phi_t^0-\phi_t^*\right\|^2=\frac{1}{T}\sum_{t=1}^{T}\left(\left\|\phi_t^0-\phi_t^*\right\|^2-\left\|\phi_t^*-\phi^*\right\|^2\right)+\frac{1}{T}\sum_{t=1}^{T}\left\|\phi_t^*-\phi^*\right\|^2, \tag{14}$$

其中，$\phi_*=\frac{1}{T}\sum_{t=1}^{T}\phi_t^*$ 是动态比较器的均值。注意，ϕ^* 是求和项 $\sum_{t=1}^{T}\left\|\phi_t^*-\phi^*\right\|^2$ 的极小值。因此，如果我们使用诸如 OGD [32] 这样的算法，通过损失 $\left\|\phi_t^0-\phi_t^*\right\|^2$ 更新初始化值 ϕ_t^0，那么公式（14）右侧第一项中的求和对应于 OGD 在一系列强凸函数上的遗憾；众所周知，该遗憾为 $\mathcal{O}$ (log T)。第二项 $\frac{1}{T}\sum_{t=1}^{T}\left\|\phi_t^*-\phi^*\right\|^2=\sigma_*^2$ 对应于动态比较器的方差。因此，结合公式（13）和公式（14），我们得到

$$\mathbf{R}_{\text{avg}}=\mathcal{O}\left(\frac{\sigma_*^2+\frac{\log T}{T}}{1-\rho^2}\right). \tag{15}$$

根据公式（9）运行 $R=\sum_{t=1}^{T}K_t$ 轮可知，以至少 $1-R\exp\left(\frac{-S(\ell_p-\ell_m)^2}{2M^2}\right)$ 的概率，任务平均遗憾的上界如公式（15）所示。因此，对于不具备精确任务边界检测的算法，其期望任务平均遗憾的上界为

$$\mathbb{E}\,\mathbf{R}_{\text{avg}} \leq \mathcal{O}\left(\frac{\sigma_*^2+\frac{\log T}{T}}{1-\rho^2}+R^2\exp\left(\frac{-S(\ell_p-\ell_m)^2}{2M^2}\right)\right) \leq \mathcal{O}\left(\frac{\sigma_*^2+\frac{\log T}{T}}{1-\rho^2}+R^{-\left(\frac{c(\ell_p-\ell_m)^2}{2M^2}-2\right)}\right), \tag{16}$$

其中，第二个不等式通过选择大小为 $S=c\log R$ 的支持集得到，由此完成定理 4.5 的证明。进一步选择 $c>\frac{4M^2}{(\ell_p-\ell_m)^2}$ 可确保期望任务平均遗憾为常数。