arXiv:2302.00674v4 [cs.LG] 3 Oct 2023

通过探索和利用辅助数据改进少样本泛化

Alon Albalak
加州大学圣塔芭芭拉分校
alon_albalak@ucsb.edu

Colin Raffel
多伦多大学 Vector Institute
craffel@gmail.com

William YangWang加州大学圣塔芭芭拉分校
william@cs.ucsb.edu

摘要

少样本学习在许多现实应用中都很有价值，但如何在不对少量带标签数据点过拟合的情况下，学习出具有泛化能力的模型，仍是一项挑战。在本研究中，我们聚焦于使用辅助数据进行少样本学习（FLAD），这是一种训练范式，假设在少样本学习期间可以访问辅助数据，以期提升泛化能力。先前研究提出了自动混合辅助数据与目标数据的方法，但这些方法通常会随辅助数据集数量线性增长（甚至更糟），限制了其实用性。本研究将 FLAD 与多臂老虎机设定中的核心难题——探索-利用困境——联系起来，并推导出计算复杂度不受辅助数据集数量影响的算法，使我们能够处理比以往方法多 $100\times$的辅助数据集。我们提出了两种算法——EXP3-FLAD 和 UCB1-FLAD，并将它们与先前仅进行探索或 仅进行利用的 FLAD方法进行比较，发现将探索与 利用相结合至关重要。通过广泛实验，我们发现，我们的方法比所有现有 FLAD方法的表现高出 4%，并首次实现了参数量为 30 亿、性能超过参数量为 1750 亿的 GPT-3 的语言模型。总的来说，我们的研究表明，为 FLAD 探索更优、更高效的混合策略，或许能为大幅提升少样本学习的泛化能力提供一条可行路径。我们的所有代码均可在 github.com/alon-albalak/FLAD 获取。

1 引言

少样本学习在许多方面都颇具吸引力：它有望节省成本和时间，而且在某些情况下，出于隐私方面的考虑或问题本身的性质，数据根本无法获取。然而，少样本学习也是一种颇具挑战性的设置，需要在学习特征空间和标签空间的结构与防止有限训练样本导致过拟合之间取得微妙的平衡 [1, 2, 3]。在少样本设置下提升模型泛化能力的一种方法是利用辅助数据进行**少**样本**学**习**辅**助**数**据（FLAD），即使用额外的辅助数据集来提升模型在目标少样本任务上的泛化能力 [4, 5, 6, 7]。

然而，FLAD方法也带来了自身的挑战，包括算法复杂度和计算复杂度的增加。具体而言，在训练期间引入辅助数据会带来大量设计选择（例如，如何以及何时使用辅助数据进行训练）。由于搜索空间在组合意义上极其庞大，手动设计如何利用大量辅助数据进行训练的课程并不可行；而基于启发式方法（例如，选择与目标小样本数据集来自相同领域或任务的数据）手动挑选辅助数据，可能导致次优结果 [8]。正如迁移学习 [8, 9, 10],、元学习 [11, 12], 、多任务学习 [13, 14, 15, 16], 和辅助学习文献 [4, 17] 所展示的那样，将这类选择交由算法处理可以获得更好的解决方案。然而，先前的辅助学习算法通常假设只有 1-3 个相关的辅助数据集。

第37届神经信息处理系统大会（NeurIPS 2023）。

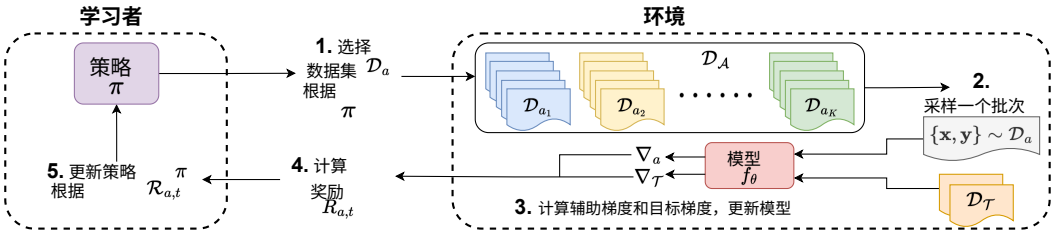

图1：**利用辅助数据进行少样本学习（FLAD）并将其建模为多臂老虎机问题的概览**。左侧是学习者，它定义了一个策略 π，用于决定从哪个辅助数据集中进行采样。右侧是环境，其中包含辅助数据集集合 $\mathcal{D}_\mathcal{A}$、目标数据集 $\mathcal{D}_\mathcal{T}$和模型 f_θ。在每一轮 t中，都会执行以下五个步骤，详见第3.1节：**1.** 学习者根据其策略 π选择一个辅助数据集 $\mathcal{D}_a$ 。**2.** 环境采样一个批次 $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_a$。**3.** 模型 f_θ 计算采样批次（∇_a）和目标数据集（$\nabla_\mathcal{T}$）的梯度，然后更新参数 θ。**4.** 根据 ∇_a 和 $\nabla_\mathcal{T}$计算奖励 $\mathcal{R}_{a,\ t}$ 。**5.** 学习者根据 $\mathcal{R}_{a,\ t}$更新 π 。

可用，并设计计算复杂度随辅助数据集数量线性（或更快）增长的算法 [18, 8], ，这促使我们在辅助数据集数量增加时寻找更高效的方法。

为克服先前研究的挑战，我们希望设计一种FLAD算法，使其**(1)** 不对可用的辅助数据做任何先验假设（是否域内、是否任务内、质量、数量等），**(2)** 能够很好地扩展到数量众多的辅助数据集，并且**(3)** 仅增加极少的内存和计算开销。我们从多臂老虎机（MAB）设置中的核心问题——探索-利用权衡 [19, 20]——获得启发，设计出满足上述目标的算法。我们将辅助数据集集合对应到多臂老虎机的臂，并通过设计三种高效的基于梯度的奖励信号，对经典的EXP3 [21]和UCB 1 [22]算法进行定制，使其适用于FLAD框架。将基于MAB的算法与高效的基于梯度的奖励相结合后，我们能够扩展到比以往方法更多的 100× 个辅助数据集。图1对我们如何将FLAD建模为多臂老虎机问题进行了基本说明。

为实证验证我们的方法，我们聚焦于语言模型的少样本训练，并利用 P3 [23], 这一现成资源，其中包含数百个辅助语言数据集。我们在与 T0 语言模型相同的留出任务上评估了方法 [16] ，结果表明，使用相同的辅助数据集集合时，我们的算法比直接微调的 T0 绝对提升了 5.6%（EXP3-FLAD）和 5.7%（UCB1-FLAD）。此外，将 P3 中所有可用数据集纳入其中（即不仅限于用于训练 T0 的数据集），提升幅度可分别增至 9.1% 和 9.2%。最后，我们将使用我们的方法训练的模型与最先进的少样本方法进行比较，发现尽管其中一个模型使用了大量未标注的目标数据集样本，我们的方法仍将性能提升了 >3%。此外，据我们所知，我们的方法首次实现了一个 30 亿参数模型，能够通过少样本上下文学习超越 175B 的 GPT-3。

总而言之，我们的主要贡献如下：

- 我们将 FLAD 与多臂老虎机环境联系起来，并通过设计两种算法 EXP3-FLAD 和 UCB1-FLAD，以及三个简单且高效（在空间和计算复杂度方面）的奖励函数，着重研究探索-利用权衡。
- 我们通过实证验证了这些方法能够提升预训练语言模型的少样本性能，并表明仅采用探索或利用的策略会导致次优性能。
- 我们开展了案例研究，以更好地理解奖励函数的动态特性，以及它们与大语言模型训练动态特性之间的相互作用。

2 相关工作

长期以来，许多研究发现，将辅助数据与目标数据结合使用能够取得成功 [4, 24, 6, 25, 26, 5, 18, 7, 27, 28, 8]。一些研究探索了加入辅助学习目标，以助力目标任务的学习 [24, 26, 25, 5, 17]。与我们的工作更为相似的是

通过引入目标数据之外的额外数据来源来进行辅助学习的方法 [4, 6, 18, 7, 27, 28, 8]。与本研究关注的少样本设置不同，以往研究探讨的是目标数据量较大的情况下的辅助学习。例如，Chen 等人 [18]和 Verboven 等人 [7]假设可以获取 10,000 个有标签目标样本；Ivison 等人 [28]和 Lin 等人 [27]假设可以获取数千个无标签目标样本；Du 等人 [6]和 Albalak 等人 [8] 则假设可以获取数百个有标签目标样本。此外，许多研究辅助学习的先前工作仅考虑了包含 1–3 个辅助数据集的设置 [6, 18, 7, 8]。例如，Verboven 等人 [7] 提出了一种任务加权方法，该方法需要求解一个方程组；当辅助任务数量较多时，该方程组会变成欠定方程组，因此该方法仅适用于单个辅助任务。此外，Chen 等人 [18] 只对 3 个辅助任务进行了实验，因为他们的方法需要为每个源任务学习一个目标感知分类器，其计算量按 $O(|\mathcal{A}||\mathcal{T}|)$ 的规模增长，其中 $|\mathcal{A}|$ 是辅助任务的数量，$|\mathcal{T}|$ 是目标任务的数量，因此难以扩展到大量源任务和目标任务。在本研究中，我们致力于通过将辅助数据集的数量提升至比以往工作高出数个数量级，在目标样本极少（20–70 个样本）的情况下改进辅助学习。与以往工作不同，提升学习过程的规模时，效率是本研究的核心考量。

数据选择研究的是一个相似（但有所不同）的问题，其目标是从单个大型数据集中有选择地使用一个子集，而不是从辅助数据集中选择数据。近期关于数据选择的研究发现，智能的数据选择能够显著提升模型性能 [29, 30, 31, 32]。

3 用于辅助数据少样本学习的多臂老虎机

本节首先定义辅助数据少样本学习（**FLAD**）设置。随后，我们将 FLAD 表述为一个多臂老虎机（**多臂老虎机**）问题，如图1所示。接下来，我们定义便于高效计算且适用于 FLAD 的奖励函数。最后，我们介绍对两种常用多臂老虎机算法的改造：EXP3-FLAD 和 UCB1-FLAD。

3.1 设置

FLAD 问题设置。辅助数据少样本学习（FLAD）适用于以下设置：假设可以访问一组大型辅助数据集 $\mathcal{D}_{\mathcal{A}}$，其中，对于所有 $a \in \mathcal{A}$，$\mathcal{D}_a$ 是一个独立的辅助数据集。给定属于目标数据集 $\mathcal{D}_{\mathcal{T}}$ 的少量数据，FLAD 的目标是找到模型 f_θ 的参数 θ，使其仅利用可用数据 $\mathcal{D}_{\mathcal{T}} \cup \mathcal{D}_{\mathcal{A}}$，便能在 $\mathcal{D}_{\mathcal{T}}$ 所对应的未知分布上取得良好性能。

将 FLAD 表述为多臂老虎机。在这项工作中，我们采用多臂老虎机（MAB）设定，将 FLAD 表述为马尔可夫决策过程 [33]，并定义学习者和环境，如图1所示。学习者由一个策略 π 构成，该策略为所有$\mathcal{D}_a \in \mathcal{D}_{\mathcal{A}}$定义选择策略。环境由目标数据集 $\mathcal{D}_{\mathcal{T}}$、辅助数据集 $\mathcal{D}_{\mathcal{A}}$和模型$f_\theta$构成。在这一表述中，学习者与环境进行交互，共持续 N 轮次。每一轮 t中，学习者从环境的 $|\mathcal{A}|$ 数据集中选择一个 $\mathcal{D}_a \in \mathcal{D}_{\mathcal{A}}$。随后，环境采样一个批次 $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_a$，并使用任务适用的损失函数计算相对于 θ 的梯度，表示为$\nabla_a = \nabla_\theta \mathcal{L}(f_\theta, \mathbf{x}, \mathbf{y})$。然后，环境计算目标梯度 $\nabla_{\mathcal{T}} = \nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_{\mathcal{T}})$，并据此更新模型参数 $\nabla_{\mathcal{T}} + \nabla_a$。最后，学习者使用基于梯度的奖励$\mathcal{R}_{a,\,t}(\nabla_a, \nabla_{\mathcal{T}})$来更新其策略 π。有关多臂老虎机的更多细节，请参阅附录A和 Lattimore & Szepesvári [34] 。

奖励函数的设计。我们设计奖励函数 $\mathcal{R}$ 时会考虑我们的设计目标。为了确保算法仅增加极少的内存和计算开销，我们考虑使用模型本身固有的信息以及正在优化的损失来计算奖励，而不依赖外部模型或指标（例如准确率或 BLEU）。在本研究中，我们借鉴先前的工作，提出三种基于梯度的奖励函数：**梯度对齐** [6, 24, 35], **梯度幅值相似性** [36, 37],及其聚合形式。形式上，在第 t 轮，令 ∇_a 为辅助批次的梯度，$\nabla_{\mathcal{T}}$ 为目标数据集的梯度。**梯度对齐**定义为 $\mathcal{R}^{GA}_{a,\,t} = \frac{\nabla_a \cdot \nabla_{\mathcal{T}}}{\|\nabla_a\|_2 \|\nabla_{\mathcal{T}}\|_2}$，即采样的辅助数据集批次与整个目标数据集的梯度之间的余弦相似度。**梯度幅值相似性**定义为 $\mathcal{R}^{GMS}_{a,\,t} = \frac{2\|\nabla_a\|_2 \|\nabla_{\mathcal{T}}\|_2}{\|\nabla_a\|_2^2 + \|\nabla_{\mathcal{T}}\|_2^2}$ 因此，当两个梯度的幅值相等时，该值为 1；随着幅值差异增大，该值趋近于零。在

除了个体奖励函数外，我们还考虑聚合奖励。为确保聚合奖励不会被任何一个个体奖励主导，我们将 $\mathcal{R}^{GA} \in [0,1]$ 归一化到与 $\mathcal{R}^{GMS}$ 相同的范围，并将二者之和定义为聚合奖励：$\mathcal{R}_a^{AGG}{}_{,\,t} = {}^{1+\mathcal{R}_a^{GA}}{}_{.\,2}{}^{t} + \mathcal{R}_a^{GMS}{}_{,\,t}$ · 我们将在第 6 节进一步讨论奖励函数的设计。

3.2 调整 EXP3 算法。

EXP3 背景 我们的首个算法 EXP3-FLAD 基于 EXP3算法 [21]（“探索与利用的指数权重算法”）。EXP3 面向对抗性多臂老虎机设定，该设定假设奖励生成过程由对手控制；对手可以访问学习者策略 π，并在每次选择之前，为每个臂确定奖励序列 $(R_{a,\,t})_{t=1}^{N}$ [38]。由于深度神经网络的损失景观高度非凸，且我们使用基于随机梯度下降的优化方法，因此考虑对抗式多臂老虎机的表述。这些因素意味着我们无法保证奖励是平稳、独立的，或遵循任何特定分布（例如高斯分布）。关于对抗式多臂老虎机的更多详情见附录A和 [21]。

在 EXP3-FLAD 中，学习者根据各臂经验确定的重要性加权奖励，按照吉布斯分布选择臂 [39]。为支持探索，我们将吉布斯分布与均匀分布混合 [21]。形式化地，设 $\mathcal{E}_t$ 为第 t轮的探索率，并回顾 $K = |\mathcal{A}|$ 为辅助数据集的数量，则 π 定义了选择给定臂 $a \in \mathcal{A}$的概率，即吉布斯分布与均匀分布的线性组合$\pi_t(a) = (1 - K\mathcal{E}_t)\,\exp(\varepsilon_{t-1}\hat{}R_a)/\sum_{a'}\exp(\varepsilon_{t-1}\hat{}R_{a'}) + \mathcal{E}_t$ 其中，$\hat{}R_{a,\,t}$ 是重要性加权奖励 $\hat{}R_{a,\,t} = \hat{}R_{a,\,t-1} + {}^{R_a}{}_{,\,\pi_{t-1}}{}^{t}{}_{a}()$ · 我们希望学习者在训练早期比后期进行更多探索，因此采用递减的探索率 $\mathcal{E}_t = \min\left\{\frac{1}{K},\ \sqrt{\ln\frac{K}{K\cdot t}}\right\}$，这是 Seldin等人 [39]提出的做法。使用重要性加权估计奖励，可以补偿不太可能被选中的动作所获得的奖励，从而保证每个动作的期望估计奖励等于实际奖励。EXP3-FLAD 的设计目标是在最坏情况下近乎最优，但由于探索率的存在，它仍会以 $\mathcal{E}_t$的比率选择“较差”的动作。EXP3-FLAD 的探索机制与重要性加权相结合，使策略能够应对非平稳奖励生成过程。

3.3 调整 UCB1 算法

UCB1 背景。 虽然 EXP3-FLAD 适用于奖励高度随机且非平稳的无约束场景，但在受约束的场景中，其他算法可能表现更好。其中一种算法是上置信界（UCB1）算法 [22], ，最初设计用于在平稳且服从正态分布的奖励函数中达到最优。尽管如此，UCB1 的变体已被证明在多种场景中有效，例如涉及非平稳、次高斯或重尾分布的场景 [40, 41]。UCB1 算法及其变体基于霍夫丁不等式 [42]为每个臂分配一个称为上置信界的值，并遵循不确定性下的乐观原则，这意味着以较高概率分配给每个臂的上置信界会高估其未知的平均奖励。

在 UCB1-FLAD 中，学习者会根据各臂的上置信界贪心地选择臂。UCB1 最初是为平稳的奖励生成过程而设计的，因此，为了适应非平稳性，我们在估计给定臂的平均奖励时加入了指数移动平均。形式化地，设 $R_{a,\,t}$ 为在第 t轮观察到的臂 a 的奖励，则估计平均奖励计算为 $R_a = (1-\beta)R_a + \beta R_{a,\,t}$ 其中 β 为平滑因子。随后，我们将上置信界定义为 $UCB_{a,\,t} = \hat{}R_a + \sqrt{2\ln {}_{n_a}{}^{t}}$. 在原始的多臂老虎机环境中，与环境的所有交互都在线上进行；而 FLAD 的独特之处在于，学习者可以在训练前与辅助数据进行交互。为利用这一点，我们不再用单个小批量初始化估计奖励，而是使用更多数据来初始化，以更准确地近似真实的数据集梯度。具体来说，我们针对每个辅助数据集计算梯度$\nabla_a = \nabla_\theta \mathcal{L}(f_\theta,\ \mathbf{x},\ \mathbf{y})$，其中 $\{\mathbf{x},\ \mathbf{y}\}$ 中的样本数可以远多于一个小批量，最多可达到整个数据集的规模。实际操作中，我们使用 1,000 个样本，在单个图形处理器上计算耗时 ~ 2 分钟。

算法 EXP3-FLAD 和 UCB1-FLAD 算法如图1所示。在每一轮中，两种方法都会先选择一个辅助数据集 $\mathcal{D}_a$。EXP3-FLAD 首先计算当前探索率 $\mathcal{E}_t$，并根据由 $\pi_t(\mathcal{A})$定义的分布对 $\mathcal{D}_a$ 进行采样；而 UCB1-FLAD 则贪心地选择 $\mathcal{D}_{a^*}$，即对应于上置信界最大的臂，$a^* = \arg\max_{a \in \mathcal{A}} UCB_{a,\ t}$。接下来，对这两种方法，环境都会从所选数据集中采样一个批次，$\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_a$，并计算梯度 $\nabla_a = \nabla_\theta \mathcal{L}(f_\theta, \mathbf{x}, \mathbf{y})$。设模型更新之间的轮数为 G，则上述步骤将重复 G次，随后环境计算目标数据集的梯度 $\nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_\mathcal{T})$，并根据$\nabla_\mathcal{T} + \sum_a \nabla_a$更新模型。最后，EXP3-FLAD 使用观测到的奖励计算每个辅助批次的重要性加权奖励，而 UCB1-FLAD 则计算平滑估计平均奖励。伪代码见附录B。

4 实验设置

模型。 在我们的实验中，我们使用 T5 系列预训练语言模型中的编码器-解码器 Transformer 模型 [43]。具体来说，我们对经过语言模型适配的 T5（T5-LM）和 T0 进行实验。T5-LM 模型以 C4 数据集 [43] 为基础，采用前缀语言建模目标 [44]对 T5.1.1 模型进一步训练了 100,000 步（相当于 100B 个 token）。T0 模型以 T5-LM 为初始模型，并按照 Sanh 等人 [16]的描述，在由提示数据集组成的多任务混合数据上进一步训练。我们以 T5-LM XL（下文称为 **T5-XL**）和 **T0-3B** 作为基础模型，重复每项实验。这两个模型采用相同的架构，均有 28.5 亿个参数；我们使用了来自 Hugging Face Transformers [45]）的模型检查点。

目标数据集。 我们从 Hugging Face Datasets[1], 获取所有数据集，并通过应用用于训练 T0 的提示公共池（P3） [23] 中的提示模板，将其转换为文本到文本格式。为了评估我们的少样本方法，我们使用与 T0 相同的留出数据集，涵盖四种不同任务：**句子补全**（COPA [46], HellaSwag [47], Story Cloze [48]）、**自然语言推理**（ANLI [49], CB [50], RTE [51]）、**共指消解**（WSC [52], Winogrande [53]）和**词义消歧**（WiC [54]）。对于每个数据集，我们从其训练数据中随机抽取五个少样本划分，每个划分包含与以往工作相同数量的训练样例，即 20 到 70 [55, 56]。此外，为了进行真正的少样本学习 [57]，我们将每个划分均分为训练集和验证集（例如，HellaSwag 中有 10 个训练样例和 10 个验证样例）。只有 ANLI 数据集提供了公开的测试集，因此对于所有其他数据集，我们都在原始验证集上评估模型（该验证集未用于少样本训练或验证）。

辅助数据集。 我们使用两组辅助数据来比较各方法的性能，并且从不将任何目标数据集纳入辅助数据。首先，我们使用 T0 多任务训练所用的数据集集合（以下简称 T0Mix），其中包含 35 个独特的数据集，涵盖问答、情感分析、主题分类、摘要生成、释义检测和结构到文本。其次，我们使用 P3 [23] 中的所有数据集（它是 T0Mix 的超集），并通过筛除与任何目标数据集重叠的数据集来避免数据泄漏，最终得到 260 个可用数据集（列表见附录 H）。对于每个辅助数据集，我们最多使用其中 10,000 个示例。

基线方法。 我们将提出的方法与几种 FLAD 和非 FLAD 基线方法进行比较。**仅目标数据**（非 FLAD）直接在目标数据集上微调基础模型（即不使用辅助数据）。**仅探索** [8] 是一种常用的 FLAD 方法，通过均等混合辅助数据集，同时使用辅助数据和目标数据进行训练。该方法最初在 [8], 中被称为 Multitask；我们将其称为仅探索，因为它相当于持续探索辅助数据，却从不利用辅助数据与目标数据之间关系的知识。**仅利用** 在训练前计算梯度对齐度（如 UCB1），并根据对齐度上的吉布斯分布混合辅助数据集来对模型进行多任务训练（与 EXP3 中的做法类似）。因此，该算法利用训练前确定的关系，却从不进行探索。仅探索和仅利用都会混合目标数据与辅助数据，辅助数据的比例为最高辅助数据采样概率的 M 倍。例如，对于使用 $M = 5$ 和 $\mathcal{D}_\mathcal{A} =$ T0Mix 的仅探索方法，采样辅助数据集 $\mathcal{D}_a \in \mathcal{D}_\mathcal{A}$ 的概率为 $1/35$，采样目标数据集的概率为 $5/35$。**损失缩放** [6] 是一种与 EXP3 和 UCB1 类似的 FLAD 方法；主要区别在于，它根据梯度对齐度缩放辅助批次损失，而不是修改采样概率。Du 等人 [6] 最初提出使用梯度对齐度来

[1]https://huggingface.co/数据集

训练方法 \ 基础模型 / 辅助数据	T5-XL		T0-3B	
	T0Mix	*P3*	*T0Mix*	*P3*
仅目标数据	52.82		56.44	
损失缩放 [6] (GA)	53.22	55.19	59.47	60.66
损失缩放 [6] (GMS)	55.98	56.40	60.47	60.70
仅探索 [8]	59.18	60.64	61.17	62.77
仅利用 [8]	59.79	60.49	60.87	62.87
EXP3-FLAD ($\mathcal{R}^{GA}$)	61.50	64.07	62.87	65.98
UCB1-FLAD ($\mathcal{R}^{GA}$)	62.01	65.52	62.35	66.29
EXP3-FLAD ($\mathcal{R}^{GMS}$)	61.72	65.57	62.78	65.51
UCB1-FLAD ($\mathcal{R}^{GMS}$)	61.67	65.21	62.85	66.00
EXP3-FLAD ($\mathcal{R}^{AGG}$)	62.05	65.47	62.84	**66.84**
UCB1-FLAD ($\mathcal{R}^{AGG}$)	**62.08**	**65.63**	**62.93**	66.29

表1：**主要结果**。每个单元格展示了使用指定训练方法（左列）和辅助数据（第二行）训练基础模型（顶行）的得分；该得分是在11个目标数据集和5个随机种子上取平均所得（每个单元格是55次实验的平均值）。仅目标数据方法不使用辅助数据。**粗体**得分表示给定基础模型和辅助数据集下（按列比较）平均值最高的得分；带下划线的得分表示 Wilcoxon秩和检验未能发现其与最高得分存在显著差异（$p > 0.05$）。扩展结果见附录D。

使用梯度对齐度（**损失缩放**（GA）），但我们还提出了一个根据梯度幅值相似性缩放损失的版本（**损失缩放**（GMS））。

训练细节。对于仅目标数据基线，我们使用范围为 $\{1e\text{-}4、\ 3e\text{-}4\}$ 的学习率。对于所有其他方法，我们始终使用 $1e$-4 的学习率。对于仅目标数据、仅探索和仅利用基线方法，我们使用范围为 $\{32, 128\}$ 的批次大小。对于损失缩放、EXP3-FLAD 和 UCB1-FLAD，我们使用包含8个样本的小批次，并令 G 处于 $\{4, 16\}$ 范围内，以匹配所有方法的批次大小。对于仅探索和仅利用方法，我们使用 $M \in \{1, 5, 10\}$ 的目标数据集混合比例。所有实验均使用 Adafactor 优化器 [58]，并通过基于验证集的早停来选择模型检查点。在初步实验中，我们考察了使用来自不同模型分区的梯度计算奖励的效果：完整模型、仅编码器、仅解码器，以及输出词汇矩阵（语言建模头）的权重。我们发现，使用语言建模头的参数能获得最佳性能，且仅占完整模型参数的2.3%，显著降低了内存消耗。对于 UCB1-FLAD，我们在初步实验中发现平滑因子 $\beta = 0.9$ 效果良好，并使用每个辅助数据集中的1,000个样本初始化辅助数据集梯度对齐度。更多实现细节见附录C

实验流程。 FLAD 实验流程包括针对每个目标数据集训练一个专门的模型。对于每种提出的方法和基线，我们都会在 11 个目标数据集上分别训练并评估模型。我们使用 5 个随机种子重复训练和评估，并将汇总结果列于表1。每个单元格显示了全部 55 次实验（11 个目标数据集、5 个随机种子）的平均准确率。对于两种模型和辅助数据集，我们都会针对每种训练方法执行这一实验流程。未汇总的结果见附录D。

5 发现与分析

在表1中，我们比较了基于多臂老虎机的方法（EXP3-FLAD 和 UCB1-FLAD）与相应基线方法在 11 个目标数据集上的实证结果（扩展结果见附录D）。对于每种基础模型与辅助数据组合（每一列），EXP3-FLAD 和 UCB1-FLAD 的表现均优于所有基线方法。事实上，我们发现对于每一项任务，我们的方法始终与基线方法表现相当或更好。这表明，与此前的 FLAD 方法相比，我们基于多臂老虎机的方法显著提升了少样本泛化能力。为了进行公平比较，确保每种方法使用相同数量的数据，我们将使用 T0 和 T0Mix 的仅目标数据方法（56.44），与使用 T5 和 T0Mix 的 FLAD 方法及基线方法（左列）进行比较。比较结果清楚地表明，损失缩放的表现实际上比多任务训练后直接微调低 0.5–3.2%。不过，我们发现其余 FLAD 方法均带来了提升（绝对提升在 2.7–5.6% 之间）。我们发现性能

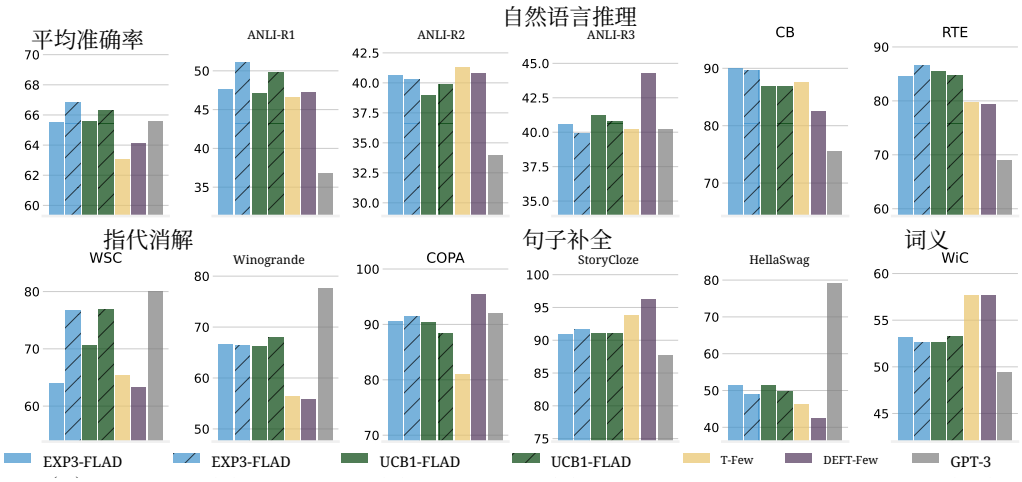

图 2：**使用 在 P3 上训练的最先进少样本方法与 FLAD 方法的比较**。T-Few 的分数来自 [56]。DEFT-Few 的分数来自 [28]。GPT-3 的分数来自 [55]，并采用少样本上下文学习。所有模型使用相同数量的少样本示例，且（GPT-3 除外）均有 3B 个参数。

在三种奖励函数下，EXP3-FLAD 与 UCB1-FLAD 的表现差异较小。总体而言，$\mathcal{R}^{AGG}$ 的表现最佳，但我们采用双侧 Wilcoxon秩和检验来检验平均得分之间的差异是否显著，发现其他奖励函数通常与其没有显著差异（$p > 0.05$）。

优先采样的重要性。损失缩放最初是为仅使用单个辅助数据集而提出的，因此事先并不清楚它如何应对更多的数据。此外，Du 等人 [6] 有意选择了与目标相关的辅助数据集，而在我们的设定中，我们不作此类假设。我们发现，我们的方法平均比损失缩放方法高出 6.3%。图 3（以及附录 E 中的图 5）显示，在训练过程中，大多数数据集的梯度对齐度和梯度幅值相似度会趋近于 0，导致损失缩放产生的梯度更新非常小。更重要的是，对于损失缩放而言，与目标任务相关的辅助数据出现的频率低于基于多臂老虎机的方法。比较损失缩放方法在使用较少（T0Mix）与较多（P3）辅助数据时的性能差异，即可看出这一点。我们发现，使用 T5 时，损失缩放（GA）最多提升 2%；而使用 T0 时，损失缩放（GMS）的提升最少仅为 0.2%。相比之下，在考虑从 T0Mix 增加到 P3 的相同数据增量时，EXP3-FLAD 和 UCB1-FLAD 有显著提升，幅度达 2.6–4 %。

探索 与 利用的重要性。有趣的是，我们原本预期仅利用方法会优于仅探索方法，因为前者利用了目标任务与辅助任务之间的关系信息，但结果发现两种方法在统计上没有差异（双侧Wilcoxon秩和检验的结果为 $p > 0.05$）。此外，在比较利用更多辅助数据的能力时（即从 T0Mix 扩展到全部 P3），我们发现仅探索和仅利用方法的提升幅度很小，只有 0.7%–2%。另一方面，EXP3-FLAD 和 UCB1-FLAD 则提升显著，幅度达到 2.6%–4%，这凸显了探索与 利用两者的重要性，尤其是在处理大量辅助数据时。

FLAD 比非 FLAD 方法具有更好的泛化能力。接下来，我们将使用 $\mathcal{R}^{AGG}$ 在 P3 上训练的最佳模型，与最先进的少样本方法进行性能比较：T-Few、DEFT-Few 和 GPT-3。T-Few [56] 是 T0-3B 模型的一个变体，它通过多任务预训练参数高效的 $(IA)^3$ 模块，然后对 $(IA)^3$ 模块进行仅目标微调。DEFT-Few [28]是 T5-XL 模型的一个变体，它使用检索到的辅助数据进行多任务训练。它首先使用 1000 个未标注的目标数据集样本，在 P3 中的 500 个最近邻样本上训练 T5-XL 模型，然后使用 Liu 等人 [56]提出的 $(IA)^3$模块进行少样本仅目标微调。最后，我们还将其与 1750 亿参数的版本进行比较

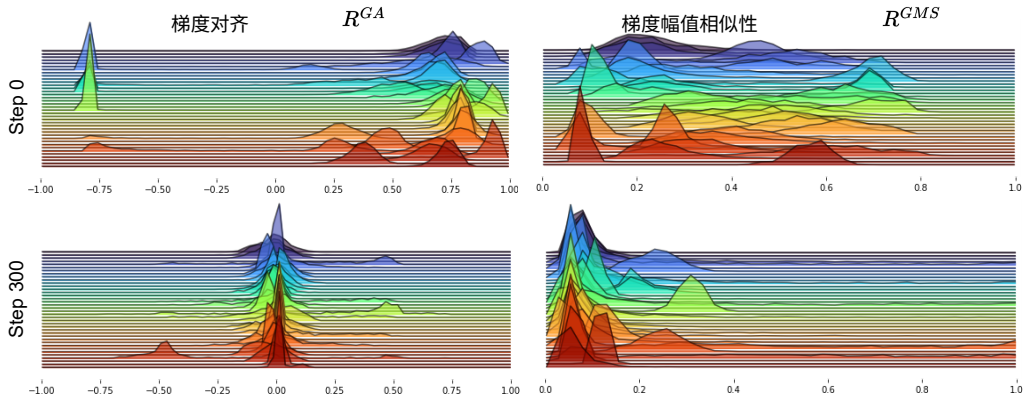

图 3：T5-XL 模型以 T0Mix 作为辅助数据集、WSC [52] 作为目标数据集时， R^{GA} 和 R^{GMS} 在训练前（第 0 步）及经过 300 次梯度更新后的奖励分布。每个象限显示了全部 35 个辅助数据集的奖励分布直方图。到第 300 步时，大多数辅助数据集提供的奖励为 0，而仅有少数剩余的“有益”数据集提供正奖励。每经过 100 次梯度更新的结果见附录 E 中的图 5。

GPT-3 [55], , 该版本使用上下文学习。我们发现，平均而言，使用基于 FLAD 的方法训练的模型优于所有其他方法；据我们所知，尽管参数量比 GPT-3 少 62.5 倍，我们的方法仍首次实现了一个在该数据集混合上表现优于 GPT-3 的 30 亿参数模型（此前最小的模型有 110 亿参数）。此外，我们发现基于 FLAD 的方法在不同数据集上表现稳健，在 $8/11$个数据集上取得最佳或次佳表现，且从未表现最差。使用特定任务模块使 T-Few 和 DEFT-Few 相较于仅目标微调取得了显著提升，避免模型陷入较差的局部极小值。然而，这些结果表明，在使用相同数据的情况下，同时利用辅助数据和目标数据进行微调，能够提升少样本泛化能力，为改善性能提供了互补途径。

研究奖励生成过程。在第 3.2 节中，我们提到，由于损失景观高度非凸，且采用基于随机梯度下降的优化技术，因此无法确保奖励生成过程是平稳的、在不同辅助数据集之间相互独立，或服从正态分布。为了深入了解奖励生成过程，我们使用 T0Mix 中全部 35 个辅助数据集的 5,000 个样本，以及小样本目标数据集 WSC [52]中的 32 个样本，考察每种奖励的分布。每隔 100 步的直方图见附录 E，图 3 则展示了一个简化版本。图 3 左侧表明，对于 $\mathcal{R}^{GA}$，几乎每个数据集都产生高斯奖励分布，少数则呈现多峰分布。值得注意的是，WikiBio [59]（深橙色）在 0.25 和 -0.75 处出现峰值。有趣的是， $\mathcal{R}^{GA}$ 会使不同数据集产生两极分化的奖励，且在 -0.75 到 0.25 之间的分布密度极低。相比之下，图 3 右侧显示， $\mathcal{R}^{GMS}$的分布中非高斯分布更多，且相比 $\mathcal{R}^{GA}$分布更平坦。值得注意的是，我们观察到 $\mathcal{R}^{GA}$ 产生的奖励分布更平稳，因为几乎所有数据集（30/35）的分布在仅经过 100 步后就迅速收敛至 0。尽管 $\mathcal{R}^{GMS}$ 的大多数分布也收敛至 0，但收敛速度较慢，耗时近 500 步。

探究训练动态。为了更好地理解我们提出的方法的训练动态，我们使用 T0Mix 对 T5-XL 进行案例研究，并且 $\mathcal{R}^{GA}$ 找到了两个数据集，在这两个数据集上，一种算法的表现都显著优于另一种（详细信息和图表见附录 F）。首先，我们研究 RTE，在该数据集上 UCB1-FLAD 的表现优于 EXP3-FLAD。我们计算了从各辅助数据集中采样的样本的经验分布，发现 EXP3-FLAD 几乎从所有数据集中均匀采样，而 UCB1-FLAD 则形成了一个在 2.5% 和 3.25% 处出现峰值的双峰采样分布（相对差异为 30%）。EXP3-FLAD 分布的均匀性有悖直觉，因为我们确实发现它在累积估计奖励方面实现了辅助任务之间的区分（如图 7 所示），但这并未使采样概率空间中的采样分布也出现区分。此外，我们还发现，即使在 EXP3-FLAD 表现优于 UCB1-FLAD 的 COPA 上，EXP3-FLAD 仍然能够实现

累积估计奖励之间的良好区分，但其抽样分布呈单峰，而 UCB1-FLAD 的分布则不像在 RTE 上那样呈现明显的双峰。经验采样分布的差异很可能源于 UCB1-FLAD 的贪婪策略与 EXP3-FLAD 的随机策略之间的差异。根据经验，我们发现 EXP3-FLAD 极少会给某个辅助数据集分配 $< 1\%$ 的概率，导致在数千轮的过程中出现许多“糟糕”的批次。另一方面，UCB1-FLAD 的乐观策略花在探索上的时间要少得多，因此采样到“糟糕”批次的频率也会低得多。

缩放的影响 $|\mathcal{A}|$ 为了评估我们提出的方法的可扩展性，我们通过改变 $|\mathcal{A}| \in \{35, 75, 125, 175, 225, 260\}$ 的规模开展了一项研究。对于每个 $|\mathcal{A}|$ 值，我们始终纳入 T0Mix 中的 35 个数据集，并从 P3 中随机选择额外的辅助数据集，直至达到所需的 $|\mathcal{A}|$。该研究在与主研究相同的 11 个目标数据集上进行，使用 T0基础模型和 $\mathcal{R}^{AGG}$ 奖励。实验使用三个随机种子重复进行。图 4 展示了 11 个目标数据集上的均值，以及不同随机种子之间的标准差。

我们发现，EXP3-FLAD 和 UCB1-FLAD 的表现都从 $|\mathcal{A}|$ =35 到 75 急剧提升。此外，我们观察到性能提升一直持续到 $|\mathcal{A}| = 260$ 的最大值；当数据集数量从 35 增至 75 时，EXP3-FLAD 的准确率最终提高了 2.54，UCB1-FLAD 提高了 3.12；当 $|\mathcal{A}|$ 从 75 增至 260 时，EXP3-FLAD 和 UCB1-FLAD 的准确率又分别提高了 1.54 和 0.47。

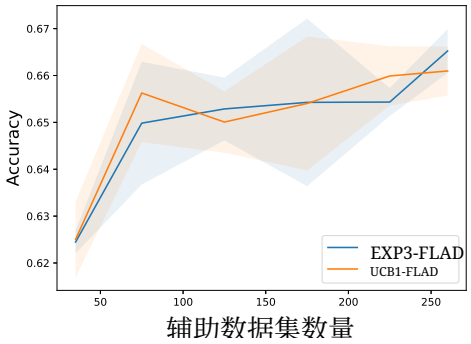

图 4：**缩放 $|\mathcal{A}|$ 对目标任务性能的影响**。每条线表示在 11 个数据集和三个随机种子上的平均得分，阴影区域表示均值上下一个标准差的范围。

6 讨论

奖励函数的讨论。在 FLAD 中，我们希望优先使用与目标任务具有相似解空间的辅助数据集进行训练，同时避免对小样本目标数据过拟合；我们的奖励函数正是为此设计的。为了更好地理解聚合奖励的奖励信号 $\mathcal{R}^{AGG}$，我们考察了四种奖励组合：$\mathcal{R}^{GA}$ 和 $\mathcal{R}^{GMS}$ 均低、$\mathcal{R}^{GA}$ 高但 $\mathcal{R}^{GMS}$ 低、$\mathcal{R}^{GA}$ 低但 $\mathcal{R}^{GMS}$ 高，以及 $\mathcal{R}^{GA}$ 和 $\mathcal{R}^{GMS}$ 均高。当两项奖励都较高时，我们可以认为辅助梯度是有用的。然而，当一项奖励较高而另一项较低时，就难以得出结论，因为单独较高的 $\mathcal{R}^{GA}$ 意味着辅助梯度会朝正确方向更新权重，但较低的 $\mathcal{R}^{GMS}$ 则可能意味着我们大幅过冲或欠冲目标，其中过冲的危害可能远大于欠冲。如果 $\mathcal{R}^{GA}$ 和 $\mathcal{R}^{GMS}$ 都较小，我们就知道辅助梯度会使我们偏离目标解空间，但不知道它的幅度比目标梯度大还是小很多。在训练开始时，我们无法判断目标梯度还是辅助梯度的幅度更大；但随着训练推进，辅助梯度幅度大于目标梯度的可能性会显著增加。因此，当 $\mathcal{R}^{GA}$ 和 $\mathcal{R}^{GMS}$ 都较低时，我们很可能会被拉离当前解很远。

本研究使用基于训练集的奖励，但也可以使用基于验证集的奖励。基于验证的奖励的一个缺点是需要频繁计算验证分数，从而增加计算复杂度。此外，我们关注少样本设置，并使用基于验证集的早停。如果使用基于验证的奖励，那么为了防止过拟合，我们需要将数据进一步划分为 3 个子集：训练集、提前停止验证集和奖励验证集。

基线方法的选择。对于辅助数据集 $|\mathcal{A}|$ 和目标数据集 $|\mathcal{T}|$的数量，我们的方法以及作为比较对象的基线方法，其计算复杂度为 $O(|\mathcal{T}|)$，且与 $|\mathcal{A}|$ 无关。对于我们的模型和这些基线方法，我们训练的模型在每个目标数据集上需要 ~ 6 GPU小时。如果考虑一种计算量随 $|\mathcal{A}|$ 线性增长的基线方法，

$O(|\mathcal{A}||\mathcal{T}|)$（例如 [18]），若没有大量硬件，这些实验将无法进行：使用 *T0Mix* 训练这样的模型，在单个目标数据集上需要 *200* 多 *GPU*小时（超过 *8 GPU*天）；使用全部 P3 时，则需要 1500 多 GPU小时（超过 *2 GPU*个月）。

为何不纳入理论保证。多臂老虎机算法的设计通常会附带遗憾界的理论证明，但在本研究中我们省略了这类分析。尽管我们可以为算法的遗憾界提供保证，但这些保证并无实际意义，因为它们针对的是奖励，而非留出数据集上的准确率（这才是我们真正关心的指标）。

FLAD 与少样本学习和多任务学习有何关系？ 少样本学习和 FLAD 都关注如何利用目标任务中有限数量的示例，优化模型在单个目标任务上的性能。在少样本学习中，模型仅获得目标任务数据 $\mathcal{D}_{\mathcal{T}}$，且没有辅助数据。实际上，对于少样本学习而言，$\mathcal{D}_{\mathcal{A}}$ 是空集。相比之下，在 FLAD 设置中 $|\mathcal{D}_{\mathcal{A}}| > 1$。根据本研究的发现，我们强烈建议从业者在有辅助数据可用时加以利用。
多任务学习关注的是同时优化模型在多个目标数据集上的性能。这与此处介绍的 FLAD 方法直接相悖，后者旨在优化模型在单个目标任务上的表现。不过，可以扩展我们基于多臂老虎机的方法，通过汇总多个奖励，同时优化多个目标任务。我们认为这是一个值得开展的有趣后续研究方向。

局限性。 FLAD 设置中的一个隐含假设（本研究及所有先前研究均如此）是，至少存在一些对目标任务有用的辅助数据。然而，我们的方法与 FLAD 设置下先前研究的主要区别之一在于，先前研究强烈假设所有辅助数据都有用，因此必须由人工挑选合适的辅助数据集。另一方面，我们的方法允许只有一小部分辅助数据有用——我们提出的算法会探索并找出有用的辅助数据集，然后加以利用。

7 结论

回顾一下引言中提出的算法设计要求：我们的算法应**(1)**不对可用的辅助数据作任何先验假设，**(2)** 能够随着辅助数据集数量的增加良好扩展，并**(3)** 尽可能减少内存和计算开销。**(1)** 在设计算法时，我们有意将问题表述为一个多臂老虎机问题。一般而言，多臂老虎机算法不对奖励质量作任何假设；尤其是，EXP3 甚至假设辅助数据集在返回奖励时会起对抗作用。**(2)** 如前所述，我们算法的单轮计算复杂度与辅助数据集的数量无关。**(3)** 最后，除了常规训练计算之外，我们的方法只增加了极少的计算开销。我们用于奖励函数的每个梯度也都用于更新模型，不会增加额外计算。唯一的计算开销是计算梯度对齐度（三次向量点积、两次标量平方根运算和两次标量乘法）或幅度相似度（四次向量点积、两次标量平方根运算、三次标量乘法和一次标量加法）。此外，我们的方法还会增加少量内存开销，用于存储模型更新之间的梯度。我们的奖励仅考虑语言建模头的梯度；实际上，每个辅助梯度需要 0.25Gb 的存储空间，因此空间复杂度略高于标准微调。

本研究提出的方法证明，在少样本设置下，同时在辅助数据集和目标数据集上训练、通过探索和利用辅助数据持续更新信念，以及将 FLAD 建模为多臂老虎机问题，均能取得成效。我们进一步表明，只要满足我们的设计要求，就能将 FLAD 扩展到数百个辅助数据集，并且表现优于传统的少样本微调和上下文学习方法。尽管本文提出的算法满足了我们的设计要求，本研究的发现仍可为未来工作提供参考，从多个方面进一步改进这些方法，例如改进奖励函数和降低空间复杂度。

致谢

本材料基于一项研究，该研究部分由谷歌提供的无限制赠款资助。本研究获得了美国国家科学基金会项目 #2048122 的支持。文中所表达的观点仅代表作者观点，并不反映美国政府的官方政策或立场。此外，我们感谢 Robert N. Noyce Trust 通过 Noyce Initiative 向加利福尼亚大学慷慨捐赠。

我们还要感谢 Dheeraj Baby 在 multi-臂老虎机设置方面提供的深刻建议，以及 Lucio Dery 和 Jiachen Li 对本文早期草稿提出的宝贵意见。

参考文献

[1] Sachin Ravi 和 Hugo Larochelle。将优化视为少样本学习的模型。发表于国际学习表征会议，2017年。[2] Yaqing Wang、Quanming Yao、James T. Kwok 和 Lionel M. Ni。从少量示例中泛化：少样本学习综述。*ACM*计算综述，53(3)，2020年6月。[3] Archit Parnami 和 Minwoo Lee。从少量示例中学习：少样本学习方法概述，2022年。[4] Pengcheng Wu 和 Thomas G. Dietterich。通过使用辅助数据源训练来提升 SVM 准确率。发表于第二十一届国际机器学习会议论文集，ICML '04，第110页，美国纽约州纽约，2004年。计算机协会。[5] Reza Esfandiarpoor、Amy Pu、Mohsen Hajabdollahi 和 Stephen H. Bach。扩展少样本学习：利用现有资源处理新任务，2020年。[6] Yunshu Du、Wojciech M. Czarnecki、Siddhant M. Jayakumar、Mehrdad Farajtabar、Razvan Pascanu 和 Balaji Lakshminarayanan。利用梯度相似性调整辅助损失，2020年。[7] Sam Verboven、Muhammad Hafeez Chaudhary、Jeroen Berrevoets、Vincent Ginis 和 Wouter Verbeke。Hydalearn。应用智能，2022年7月。[8] Alon Albalak、Yi-Lin Tuan、Pegah Jandaghi、Connor Pryor、Luke Yoffe、Deepak Ramachandran、Lise Getoor、Jay Pujara 和 William Yang Wang。FETA：开放领域对话中少样本任务迁移的基准。发表于*2022*年自然语言处理经验方法会议论文集，第10936–10953页，阿联酋阿布扎比，2022年12月。计算语言学协会。[9] Tu Vu、Tong Wang、Tsendsuren Munkhdalai、Alessandro Sordoni、Adam Trischler、Andrew Mattarella-Micke、Subhransu Maji 和 Mohit Iyyer。探索并预测自然语言处理任务间的可迁移性。发表于*2020*年自然语言处理经验方法会议论文集（*EMNLP*），第7882–7926页，线上，2020年11月。计算语言学协会。[10] Yada Pruksachatkun、Jason Phang、Haokun Liu、Phu Mon Htut、Xiaoyi Zhang、Richard Yuanzhe Pang、Clara Vania、Katharina Kann 和 Samuel R. Bowman。使用预训练语言模型进行中间任务迁移学习：何时有效，为什么有效？发表于第*58*届计算语言学协会年会论文集，第5231–5247页，线上，2020年7月。计算语言学协会。[11] Sebastian Thrun 和 Lorien Pratt（编）。学会学习。Kluwer Academic Publishers，美国，1998年。[12] Trapit Bansal、Rishikesh Jha、Tsendsuren Munkhdalai 和 Andrew McCallum。面向少样本自然语言分类任务的自监督元学习。发表于*2020*年自然语言处理经验方法会议论文集（*EMNLP*），第522–534页，线上，2020年11月。计算语言学协会。

[13] Sen Wu、Hongyang R. Zhang 和 Christopher Ré。理解并改进多任务学习中的信息传递。发表于国际学习表征会议，2020年。[14] Armen Aghajanyan、Anchit Gupta、Akshat Shrivastava、Xilun Chen、Luke Zettlemoyer 和 Sonal Gupta。Muppet：通过预微调获得大规模多任务表征。发表于*2021*年自然语言处理经验方法会议论文集，第5799–5811页，线上及多米尼加共和国蓬塔卡纳，2021年11月。计算语言学协会。[15] Vamsi Aribandi、Yi Tay、Tal Schuster、Jinfeng Rao、Huaixiu Steven Zheng、Sanket Vaibhav Mehta、Honglei Zhuang、Vinh Q. Tran、Dara Bahri、Jianmo Ni、Jai Gupta、Kai Hui、Sebastian Ruder 和 Donald Metzler。Ext5：迈向迁移学习的极端多任务扩展。发表于国际学习表征会议，2022年。[16] Victor Sanh、Albert Webson、Colin Raffel、Stephen Bach、Lintang Sutawika、Zaid Alyafeai、Antoine Chaffin、Arnaud Stiegler、Arun Raja、Manan Dey、M Saiful Bari、Canwen Xu、Urmish Thakker、Shanya Sharma Sharma、Eliza Szczechla、Taewoon Kim、Gunjan Chhablani、Nihal Nayak、Debajyoti Datta、Jonathan Chang、Mike Tian-Jian Ji ang、Han Wang、Matteo Manica、Sheng Shen、Zheng Xin Yong、Harshit Pandey、Rachel Bawden、Thomas Wang、Trishala Neeraj、Jos Rozen、Abheesht Sharma、Andrea Santilli、Thibault Fevry、Jason Alan Fries、Ryan Teehan、Teven Le Scao、Stella Biderman、Leo Gao、Thomas Wolf 和 Alexander M Rush。多任务提示训练实现零样本任务泛化。发表于国际学习表征会议，2022年。[17] Lucio M. Dery、Paul Michel、Mikhail Khodak、Graham Neubig 和 Ameet Talwalkar。AANG：自动化辅助学习。发表于第十一届国际学习表征会议，2023年。[18] Shuxiao Chen、Koby Crammer、Hangfeng He、Dan Roth 和 Weijie J Su。面向跨任务学习的加权训练。发表于国际学习表征会议，2022年。[19] William G Macready 和 David H Wolpert。多臂赌博机问题与探索/利用权衡。*IEEE*进化计算学报，2(1)：2–22，1998年。[20] Alex Simpkins、Raymond De Callafon 和 Emanuel Todorov。探索与利用之间的最优权衡。发表于*2008*年美国控制会议，第33–38页。IEEE，2008年。[21] Peter Auer、Nicolo Cesa-Bianchi、Yoav Freund 和 Robert E Schapire。非随机多臂赌博机问题。*SIAM*计算杂志，32(1)：48–77，2002年。[22] Peter Auer、Nicolo Cesa-Bianchi 和 Paul Fischer。多臂赌博机问题的有限时间分析。机器学习，47(2)：235–256，2002年。[23] Stephen Bach、Victor Sanh、Zheng Xin Yong、Albert Webson、Colin Raffel、Nihal V. Nayak、Abheesht Sharma、Taewoon Kim、M Saiful Bari、Thibault Fevry、Zaid Alyafeai、Manan Dey、Andrea Santilli、Zhiqing Sun、Srulik Ben-david、Canwen Xu、Gunjan Chhablani、Han Wang、Jason Fries、Maged Al-shaibani、Shanya Sharma、Urmish Thakker、Khalid Almubarak、Xiangru Tang、Dragomir Radev、Mike Tian-jian Jiang 和 Alexander Rush。PromptSource：用于自然语言提示的集成开发环境和资源库。发表于第*60*届计算语言学协会年会论文集：系统演示，第93–104页，爱尔兰都柏林，2022年5月。计算语言学协会。[24] Xingyu Lin、Harjatin Baweja、George Kantor 和 David Held。面向强化学习的自适应辅助任务加权。载于 H. Wallach、H. Larochelle、A. Beygelzimer、F. d'Alché-Buc、E. Fox 和 R. Garnett（编）的神经信息处理系统进展，第32卷。Curran Associates, Inc.，2019年。[25] Lucio M. Dery、Paul Michel、Ameet S. Talwalkar 和 Graham Neubig。我们应该进行预训练吗？论采用面向终端任务的训练作为替代方案。*ArXiv*，abs/2109.07437，2021年。[26] Aviv Navon、Idan Achituve、Haggai Maron、Gal Chechik 和 Ethan Fetaya。通过隐式微分进行辅助学习。发表于国际学习表征会议，2021年。

[27] Bill Yuchen Lin、Kangmin Tan、Chris Miller、Beiwen Tian 和 Xiang Ren。通过检索增强实现无监督跨任务泛化。*ArXiv*，abs/2204.07937，2022。[28] Hamish Ivison、Noah A. Smith、Hannaneh Hajishirzi 和 Pradeep Dasigi。利用跨任务最近邻进行数据高效微调，2022。[29] Sören Mindermann、Muhammed Razzak、Winnie Xu、Andreas Kirsch、Mrinank Sharma、Adrien Morisot、Aidan N. Gomez、Sebastian Farquhar、Janina Brauner 和 Yarin Gal。优先训练那些可学习、值得学习且尚未学会的样本。载于国际机器学习会议，2021。[30] Shoaib Ahmed Siddiqui、Nitarshan Rajkumar、Tegan Maharaj、David Krueger 和 Sara Hooker。元数据考古：利用训练动态挖掘数据子集。*ArXiv*，abs/2209.10015，2022。[31] Ben Sorscher、Robert Geirhos、Shashank Shekhar、Surya Ganguli 和 Ari S. Morcos。超越神经网络缩放定律：通过数据剪枝打破幂律缩放。载于 Alice H. Oh、Alekh Agarwal、Danielle Belgrave 和 Kyunghyun Cho 编，神经信息处理系统进展，2022。[32] Amro Abbas、Kushal Tirumala、Dániel Simig、Surya Ganguli 和 Ari S. Morcos。Semdedup：通过语义去重实现网络规模的数据高效学习，2023。[33] RICHARD BELLMAN。马尔可夫决策过程。《数学与力学杂志》，6(5):679–684，1957。[34] Tor Lattimore 和 Csaba Szepesvári。《*Bandit Algorithms*》。剑桥大学出版社，2020。[35] Zirui Wang、Yulia Tsvetkov、Orhan Firat 和 Yuan Cao。梯度疫苗：研究并改进大规模多语言模型中的多任务优化。载于国际学习表征会议，2021。[36] Wufeng Xue、Lei Zhang、Xuanqin Mou 和 Alan C. Bovik。梯度幅值相似性偏差：一种高效的感知图像质量指数。《*IEEE*图像处理汇刊》，23(2):684–695，2014。[37] Tianhe Yu、Saurabh Kumar、Abhishek Gupta、Sergey Levine、Karol Hausman 和 Chelsea Finn。面向多任务学习的梯度手术。*arXiv*预印本 *arXiv:2001.06782*，2020。[38] Peter Auer、Nicolo Cesa-Bianchi、Yoav Freund 和 Robert E Schapire。在作弊赌场赌博：对抗式多臂老虎机问题。载于*IEEE*第*36*届年度计算机科学基础会议论文集，第322–331页。IEEE，1995。[39] Yevgeny Seldin、Csaba Szepesvári、Peter Auer 和 Yasin Abbasi-Yadkori。随机环境中EXP3算法性能的评估与分析。载于 Marc Peter Deisenroth、Csaba Szepesvári 和 Jan Peters 编，第十届欧洲强化学习研讨会论文集，第24卷《机器学习研究论文集》，第103–116页，爱丁堡，苏格兰，2013年6月30日至7月1日。PMLR。[40] Aurélien Garivier 和 Eric Moulines。关于切换老虎机问题的上置信界策略。载于 Jyrki Kivinen、Csaba Szepesvári、Esko Ukkonen 和 Thomas Zeugmann 编，算法学习理论，第174–188页，柏林，海德堡，2011。Springer Berlin Heidelberg。[41] Lai Wei 和 Vaibhav Srivastava。非平稳随机多臂老虎机：UCB策略与极小极大遗憾，2021。[42] Wassily Hoeffding。有界随机变量之和的概率不等式。《美国统计学会杂志》，58(301):13–30，1963。[43] Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li 和 Peter J. Liu。利用统一的文本到文本转换器探索迁移学习的极限。《机器学习研究杂志》，21(1)，2020年6月。

[44] Brian Lester、Rami Al-Rfou 和 Noah Constant。参数高效提示调优的规模效应。载于*2021*年自然语言处理实证方法会议论文集，第3045–3059页，线上及多米尼加共和国蓬塔卡纳，2021年11月。计算语言学协会。

[45] Thomas Wolf、Lysandre Debut、Victor Sanh、Julien Chaumond、Clement Delangue、Anthony Moi、Pierric Cistac、Tim Rault、Remi Louf、Morgan Funtowicz、Joe Davison、Sam Shleifer、Patrick von Platen、Clara Ma、Yacine Jernite、Julien Plu、Canwen Xu、Teven Le Scao、Sylvain Gugger、Mariama Drame、Quentin Lhoest 和 Alexander Rush。Transformers：最先进的自然语言处理。在*2020*年自然语言处理实证方法会议论文集：系统演示中，第38–45页，线上，2020年10月。计算语言学协会。

[46] Andrew Gordon、Zornitsa Kozareva 和 Melissa Roemmele。SemEval-2012任务7：合理替代方案：常识因果推理评估。在**SEM 2012*：首届词汇与计算语义联合会议——第*1*卷：主会议及共享任务论文集，以及第*2*卷：第六届国际语义评测研讨会论文集（*SemEval 2012*）中，第394–398页，加拿大蒙特利尔，2012年6月7日至8日。计算语言学协会。

[47] Rowan Zellers、Ari Holtzman、Yonatan Bisk、Ali Farhadi 和 Yejin Choi。HellaSwag：机器真的能续写你的句子吗？载于第*57*届计算语言学协会年会论文集，第4791–4800页，意大利佛罗伦萨，2019年7月。计算语言学协会。

[48] Rishi Sharma、James Allen、Omid Bakhshandeh 和 Nasrin Mostafazadeh。应对故事完形测试中的故事结尾偏差。载于第*56*届计算语言学协会年会论文集（第*2*卷：短篇论文），第752–757页，澳大利亚墨尔本，2018年7月。计算语言学协会。

[49] Yixin Nie、Adina Williams、Emily Dinan、Mohit Bansal、Jason Weston 和 Douwe Kiela。对抗性自然语言推理：自然语言理解的新基准。载于第*58*届计算语言学协会年会论文集，第4885–4901页，线上，2020年。计算语言学协会。

[50] Marie-Catherine de Marneffe、Mandy Simons 和 Judith Tonhauser。承诺库：研究自然发生的话语中的投射。载于*Sinn und Bedeutung*，2019年。

[51] Ido Dagan、Oren Glickman 和 Bernardo Magnini。PASCAL文本蕴含识别挑战。载于 Joaquin Quiñonero-Candela、Ido Dagan、Bernardo Magnini 和 Florence d'Alché Buc 编，机器学习挑战：评估预测不确定性、视觉对象分类和文本蕴含识别，第177–190页，柏林、海德堡，2006年。Springer Berlin Heidelberg。

[52] Hector Levesque、Ernest Davis 和 Leora Morgenstern。Winograd模式挑战。载于第十三届知识表示与推理原理国际会议，2012年。

[53] Keisuke Sakaguchi、Ronan Bras、Chandra Bhagavatula 和 Choi Yejin。Winogrande：大规模对抗性Winograd模式挑战。*AAAI*人工智能会议论文集，34：8732–8740，2020年4月。

[54] Mohammad Taher Pilehvar 和 Jose Camacho-Collados。WiC：用于评估语境敏感的意义表示的词语语境数据集。载于《*2019*年北美计算语言学协会分会年会论文集：人类语言技术》，第*1*卷（长篇与短篇论文），第1267–1273页，明尼阿波利斯，明尼苏达州，2019年6月。计算语言学协会。

[55] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、JaredDKaplan、PrafullaDhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell、Sandhini Agarwal、Ariel Herbert-Voss、Gretchen Krueger、Tom Henighan、Rewon Child、Aditya Ramesh、Daniel Ziegler、Jeffrey Wu、Clemens Winter、Chris Hesse、Mark Chen、Eric Sigler、Mateusz Litwin、Scott

Gray、BenjaminChess、Jack Clark、ChristopherBerner、SamMcCandlish、AlecRadford、Ilya Sutskever 和 Dario Amodei。《语言模型是少样本学习者》。载于 H. Larochelle、M. Ranzato、R. Hadsell、M.F. Balcan 和 H. Lin 编，《神经信息处理系统进展》，第 33 卷，第 1877–1901 页。Curran Associates, Inc.，2020 年。

[56] Haokun Liu、Derek Tam、Muqeeth Mohammed、Jay Mohta、Tenghao Huang、Mohit Bansal 和 Colin Raffel。少样本参数高效微调比上下文学习效果更好、成本更低。载于 Alice H. Oh、Alekh Agarwal、Danielle Belgrave 和 Kyunghyun Cho 编，神经信息处理系统进展，2022 年。

[57] Ethan Perez、Douwe Kiela 和 Kyunghyun Cho。利用语言模型实现真正的少样本学习。载于 A. Beygelzimer、Y. Dauphin、P. Liang 和 J. Wortman Vaughan 编，神经信息处理系统进展，2021年。[58] Noam Shazeer 和 Mitchell Stern。Adafactor：具有次线性内存开销的自适应学习率。载于 Jennifer Dy 和 Andreas Krause 编，第*35*届国际机器学习大会论文集，第80卷《机器学习研究论文集》，第4596–4604页。PMLR，2018年7月10日至15日。[59] Rémi Lebret、David Grangier 和 Michael Auli。利用结构化数据生成文本，并应用于传记领域。*CoRR*，abs/1603.07771，2016年。

一种多臂老虎机

多臂老虎机（**多臂老虎机**）是机器学习中的一个问题：学习者遵循策略 π，在 N 轮次中与环境交互。每一轮 t，学习者会选择环境的一个 K 臂， $a \in \mathcal{A}$ 其中 $K = |\mathcal{A}|$，随后环境会提供奖励 R_t。未被选择的臂所对应的奖励不会被观察到。学习者的目标是采用一种策略π，选择能够在 N 轮次中带来最大累积奖励的行动， $R = \sum_{t=1}^{N} R_t$。在这项工作中，我们假设有限 K，并且每个臂的底层奖励分布可能具有多种属性（例如随机性或平稳性），具体取决于确切场景，因此会产生不同的最优策略 [34]。

对抗式多臂老虎机。 对抗性多臂老虎机设定假设奖励生成过程由对手控制。此假设能够对非平稳且高度随机的奖励信号进行建模。稍后我们将说明，为何我们的 FLAD 表述符合这一设定。在此设定下，假设对手可以访问学习者策略 π，并在每次行动之前确定每个臂的奖励序列 $(R_{a,\ t})_{t=1}^{N}$ [38]。每一轮中， π 确定一个动作分布 p（$\mathcal{A}$），并从该分布中抽取一个动作 $a \sim p$（$\mathcal{A}$）。更多详情请参见 Lattimore & Szepesvári [34]。

EXP3算法。 EXP3算法（“探索与利用的指数加权算法”）通过根据臂的经验重要性加权奖励 [21]所确定的吉布斯分布来选择臂，从而解决对抗式多臂老虎机问题。为实现探索，EXP3将吉布斯分布与均匀分布混合。

形式上，设探索率为 $\gamma \in (0, 1]$。在第 t 轮， π 将选择给定臂 $a \in \mathcal{A}$ 的概率定义为 Gibbs 分布和均匀分布的线性组合

$$p_t(a) = (1-\gamma)\frac{\exp(\gamma \hat{R}_{a,t-1}/K)}{\sum_{a'} \exp(\gamma \hat{R}_{a',t-1}/K)} + \frac{\gamma}{K} \tag{1}$$

ˆ 其中，重要性加权奖励$R_{a,\ t}$ 的计算方式为

$$\hat{R}_{a,t} = \hat{R}_{a,t-1} + \frac{R_{a,t}}{p_{t-1}(a)} \tag{2}$$

且 $R_{a,\ t}$ 表示观测到的奖励。所有未选择的臂 $a' \neq a$ 的重要性加权奖励保持不变；$R_{a',\ t} = R_{a',\ t-1}$。

从算法角度来看，EXP3 在每一轮执行以下步骤：首先，计算采样分布 p_t，并从该分布中抽取一个臂。然后观测到奖励 $R_{a,\ t}$，算法 ˆ 更新所选臂的重要性加权奖励$R_{a,\ t}$。

通俗地说，使用重要性加权估计奖励可以补偿较不可能被选中的动作所获得的奖励，从而保证每个动作的期望估计奖励等于实际奖励。EXP3 的设计旨在最坏情况下接近最优，但由于探索率的影响，它会以 γ/K 的频率选择“较差”的动作。EXP3 的探索结合重要性加权，使该策略能够应对非平稳奖励生成过程。

UCB1 算法。虽然对抗性设定几乎不对奖励生成过程作任何假设，因此在几乎所有情况下都能保持其性能保证，但在受到限制的环境中，其他方法可能表现更好。本节假设奖励生成过程服从平稳高斯分布。解决这种多臂老虎机环境的一种常见策略是上置信界（UCB1）算法，该算法根据霍夫丁不等式 [22] 为每个臂分配一个称为上置信界的值。UCB1 算法基于不确定性下的乐观原则，这意味着以较高概率分配给每个臂的上置信界会高于未知的平均奖励。

ˆ 形式化地，设臂 a 被执行 n_a 次后的估计平均奖励为 R_a，真实平均奖励为 R_a，则

$$\mathbb{P}\left(R_a \geq \hat{R}_a + \sqrt{\frac{2\ln(1/\delta)}{n_a}}\right) \leq \delta \quad \forall \delta \in (0, 1)$$

该结论由霍夫丁不等式推导得出（遵循 Lattimore & Szepesvári [34]的公式 7.1），其中δ为置信水平，用于量化对该臂的确信程度。在本文中，我们令 $\delta = 1/t$，其中 t 为已执行的轮次数，使置信界随轮次增加而缩小。因此，我们将第 t 轮时臂 a 的上置信界定义为

$$UCB_{a,t} = \begin{cases} \infty, & \text{if } n_a = 0 \\ \hat{R}_a + \sqrt{\frac{2 \ln t}{n_a}}, & \text{otherwise} \end{cases} \tag{3}$$

从算法角度来看，UCB1 在每一轮都会执行以下步骤。首先，UCB1策略选择上置信界最大的臂，$a^* = \arg\max_{a \in \mathcal{A}} UCB_{a,\ t}$。接着，观察到奖励 $R_{a^*,\ t}$，然后算法更新 R_{a^*}（a^*的估计平均奖励）以及所有 a的上置信界。简而言之，该算法表明，如果臂满足以下任一条件，学习者就应更频繁地选择它：1. 期望奖励较高，R；或 2. n_a 较小，因为该臂尚未得到充分探索。

B 伪代码

我们在此给出所提出的两种算法的伪代码。算法 1 包含 EXP3-FLAD 的伪代码，算法 2 包含 UCB1-FLAD 的伪代码。

算法 1EXP3-FLAD

要求： $\mathcal{D}_{\mathcal{A}}$，$\mathcal{D}_{\mathcal{T}}$：辅助数据集和目标数据集
要求： f_θ：参数化模型
要求： G：梯度累积步数
初始化：$K = |\mathcal{A}|$；$\mathcal{E}_0 = \frac{1}{K}$；
$\forall a \in \mathcal{A}$：$\nabla_a = 0$,$\hat{R}_a = 1$
循环 $t = 1, 2, \ldots, N$ **do**
$\mathcal{E}_t = \text{最小值}\left\{\frac{1}{K}, \sqrt{\frac{\ln K}{K \cdot t}}\right\}$
$\forall a \in \mathcal{A}$：$\pi(a) \leftarrow (1 - K\mathcal{E}_t) \frac{\exp(\mathcal{E}_{t-1} \hat{R}_a)}{\sum_{a'} \exp(\mathcal{E}_{t-1} R_{a'})} + \mathcal{E}_t$
样本 $a \sim \pi(\mathcal{A})$ 和批次 $\{\mathbf{x},\ \mathbf{y}\} \sim \mathcal{D}_a$
$\nabla_a \leftarrow \nabla_a + \nabla_\theta \mathcal{L}\left(f_\theta, \mathbf{x}, \mathbf{y}\right)$
如果 $t \pmod G \equiv 0$ **那么**
$\nabla_{\mathcal{T}} \leftarrow \nabla_\theta \mathcal{L}(f_\theta,\ \mathcal{D}_{\mathcal{T}})$
更新相对于$\nabla_{\mathcal{T}} + \sum$的模型参数 $_a \nabla_a$
对于所有 $\{a \in \mathcal{A} | \nabla_a \neq 0\}$ **执行**
$\hat{R}_a \leftarrow R_a + \frac{R_{a,t}}{\pi(a)}$
$\nabla_a \leftarrow 0$
结束循环
结束条件判断
结束循环

算法 2UCB1-FLAD

要求： $\mathcal{D}_{\mathcal{A}}, \mathcal{D}_{\mathcal{T}}$：辅助数据集和目标数据集
要求： f_θ：参数化模型
要求： G：梯度累积步数
要求： β：平滑因子
初始化：
$\forall a \in \mathcal{A}: n_a = 1$,
$\hat{R}_a = \cos(\nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_{\mathcal{T}}) \nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_a))$
对于 $t = 1, 2, \ldots, N$ **do**
$a^* = \text{argmax}_{a \in \mathcal{A}} \hat{R}_a + \sqrt{\frac{2 \ln t}{n_a}}$
样本批次 $\{\mathbf{x}, \mathbf{y}\} \sim \mathcal{D}_{a^*}$
$\nabla_{a^*} \leftarrow \nabla_{a^*} + \nabla_\theta \mathcal{L}(f_\theta, \mathbf{x}, \mathbf{y})$
$n_{a^*} \leftarrow n_{a^*} + 1$
如果 $t \pmod G \equiv 0$ **则**
$\nabla_{\mathcal{T}} \leftarrow \nabla_\theta \mathcal{L}(f_\theta, \mathcal{D}_{\mathcal{T}})$
根据 $\nabla_{\mathcal{T}} + \sum_a \nabla_a$ 更新模型参数
对于所有 $\{a \in \mathcal{A} | \nabla_a \neq 0\}$ **执行**
$\hat{R}_a \leftarrow (1-\beta)\hat{R}_a + \beta R_{a,t}$
$\nabla_a \leftarrow 0$
结束循环
结束条件判断
结束循环

C 训练细节

我们在 40GB A100 上训练所有模型（FLAD 和非 FLAD）。

在所有实验中，我们都采用基于验证集的早停，最多训练 10,000 个梯度更新步数。实践中我们发现，早停会使更新次数显著少于 10,000 次：直接微调通常为 50–150 次，其他方法则为 1–2,000 次。

For 对于 UCB1-FLAD 中的平滑因子 β，我们进行了初步实验，使用的值为 $\{0.99, 0.9, 0.75, 0.5\}$，并发现 0.9 在各数据集上表现良好。所有报告的分数均使用 $\beta = 0.9$。在初步实验中，我们考虑了使用来自多个模型分区的梯度计算奖励：完整模型、仅编码器、仅解码器以及语言建模头（词元分类器）。我们发现，使用语言模型头的参数能取得最佳性能，其次依次是仅解码器、仅编码器和完整模型的梯度。最佳方法与最差方法之间的相对性能差异为 $\sim 3\%$。请注意，当梯度累积因子为 G 时，我们的算法在任意时刻最多只需存储 $G+1$ 个梯度。因此，使用语言模型头不仅能提升性能，还能节省内存。对于我们使用的模型，语言模型头仅包含完整模型参数的 2.3%。

D 完整结果

仅目标微调、仅探索、仅利用、EXP3-FLAD 和 UCB1-FLAD 的完整实验结果见下一页。

Tab表 2：主要实验的详细结果，包括直接微调、仅探索 y,
expl和仅利用的基线方法，以及我们提出的方法 EXP3-FLAD 和 UCB1-FLAD。

E 探测奖励生成过程。

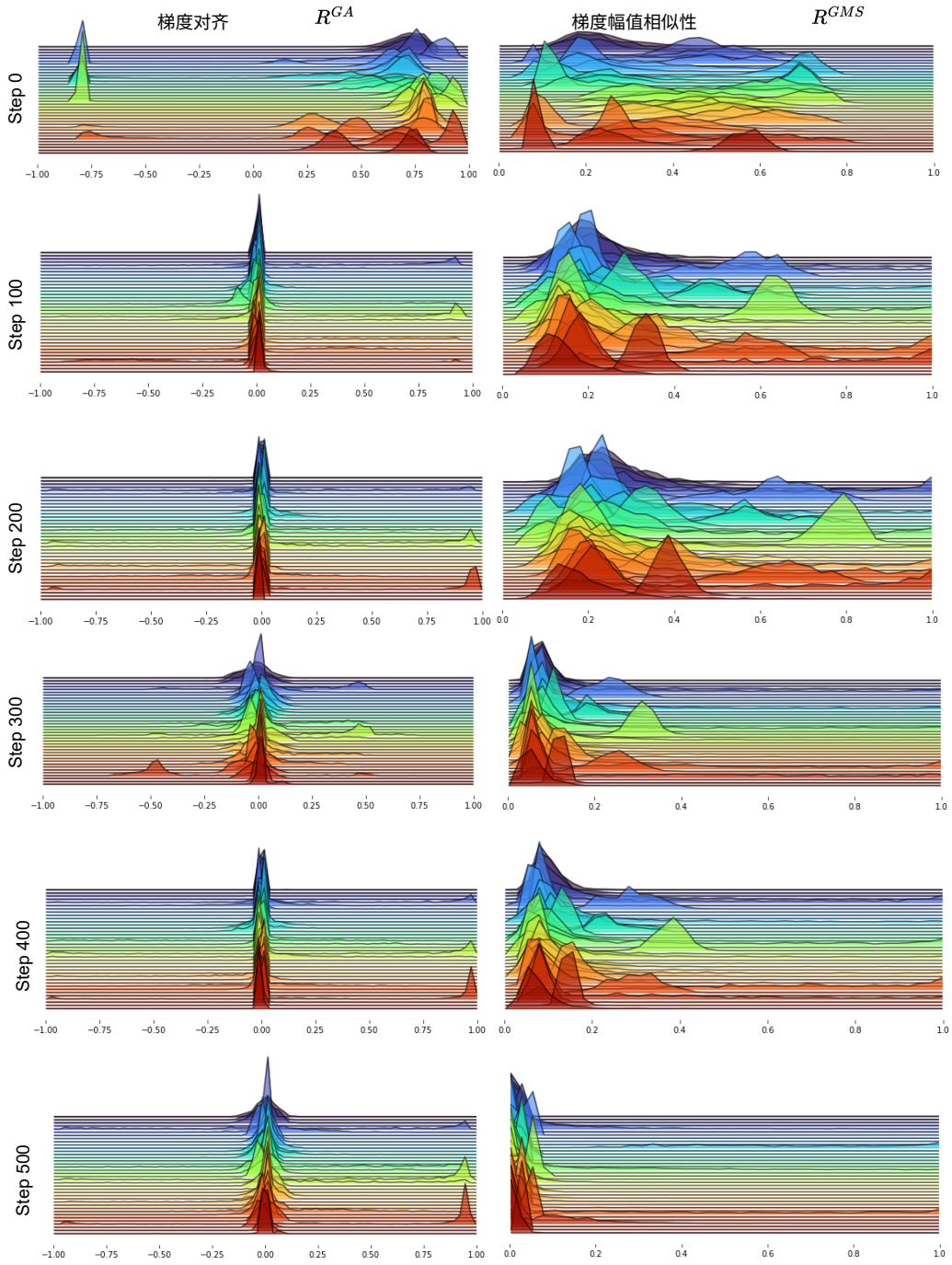

图 5：训练前以及此后每经过 100 次梯度更新时， $\mathcal{R}^{GA}$ 和 $\mathcal{R}^{GMS}$ 的**奖励分布**。我们使用 T0Mix 辅助数据集上的 T5-XL 模型进行探测，并以 WSC [52] 作为目标数据集。

F EXP3-FLAD 与 UCB1-FLAD 的训练动态

接下来的 4 页将通过案例研究，分析使用 T0Mix 作为辅助数据训练 T5-XL 时 EXP3-FLAD 和 UCB1-FLAD 的训练动态。首先，我们找出 EXP3-FLAD 和 UCB1-FLAD 相较于基线 FLAD 方法有显著提升的数据集，以及 EXP3-FLAD 或 UCB1-FLAD 明显优于另一种方法的数据集。符合我们要求的两个数据集是 RTE 和 COPA。

我们发现 UCB1-FLAD 在 RTE 上优于 EXP3-FLAD，并在图 6（UCB1）和图 7（EXP3）中展示了它们各自的训练动态。

我们发现 EXP3-FLAD 在 COPA 上优于 UCB1-FLAD，并在图 8（UCB1）和图 9（EXP3）中展示了它们各自的训练动态。

我们在每幅图的图注中提供了详细信息和要点。对于 EXP3-FLAD 图，我们展示了累积估计奖励、经验梯度对齐、由策略决定的瞬时抽样分布，以及由每个数据集所见样本数占全部已见样本数的比例决定的经验采样分布图。对于 UCB1-FLAD 图，我们展示了上置信指数、估计梯度对齐和经验采样分布图。

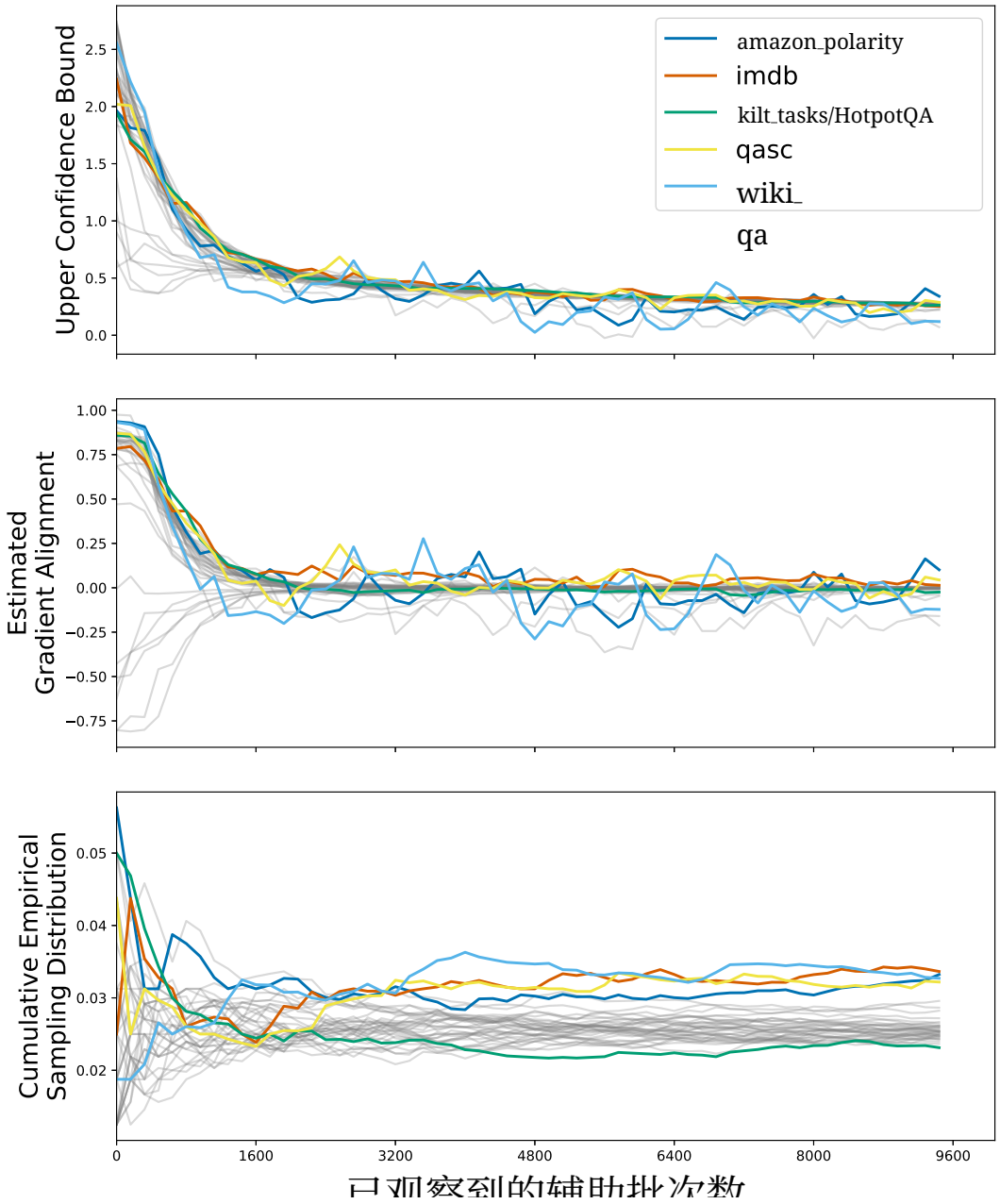

图 6：UCB1-FLAD 的训练动态案例研究，以 RTE 为目标数据集、T0Mix 为辅助数据，其中 UCB1-FLAD 的表现优于 EXP3-FLAD。彩色线条表示具有有趣特性的部分辅助数据集，其余数据集以灰色显示。我们发现，尽管 wiki_qa 的估计梯度对齐降至 0 以下（中图），UCB1 并未因此在之后停止从中采样；在 3200 至 4800 个批次之间，它成为上置信界最大的辅助数据集（上图）。类似地，我们看到 UCB1 会在 wiki_qa、amazon_polarity 和 qasc 之间交替选择梯度对齐度和上置信界较高的数据集。kilt_tasks/hotpotqa 在训练前的梯度对齐度很高，但由于其上置信界较低，UCB1 很少从中采样。这是基于迁移学习的方法的一个失败案例。有趣的是，UCB1 从未估计出 imdb 的梯度为负，并且在训练过程中逐渐越来越频繁地从中采样。

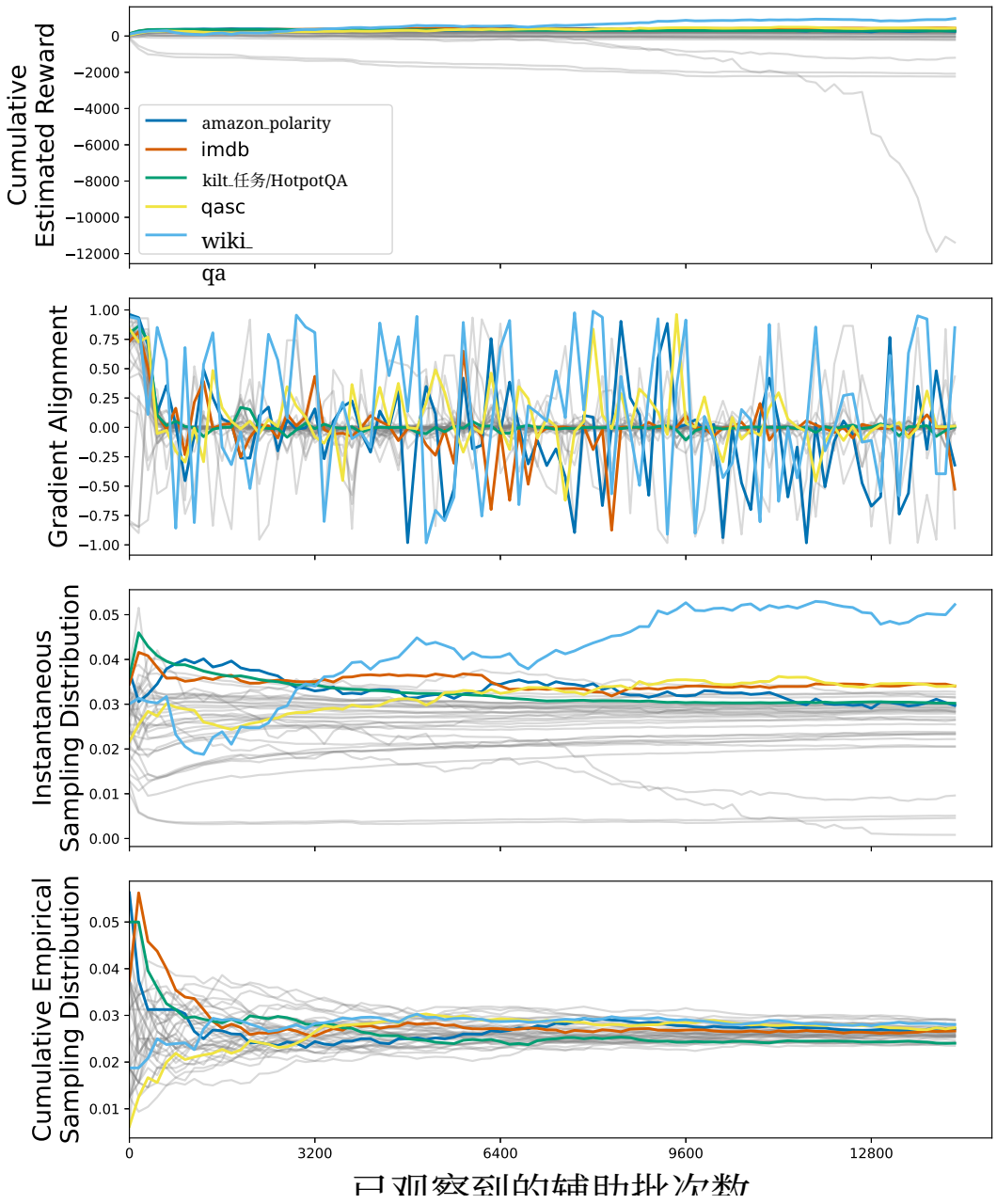

图 7：EXP3-FLAD 的训练动态案例研究：以 RTE 为目标数据集、T0Mix 为辅助数据，其中 UCB1-FLAD 的表现优于 EXP3-FLAD。彩色线条表示具有有趣特性的部分辅助数据集，其余数据集以灰色显示。我们发现，对于 EXP3-FLAD，梯度对齐信号尤其嘈杂，这可能导致其在 RTE 上的表现略差。所有五个突出显示的辅助数据集都具有较高的瞬时抽样概率，但在整个训练过程中，经验采样分布在全部辅助数据集上的分布非常集中；相比之下，UCB1 能够找到更好的区分度。

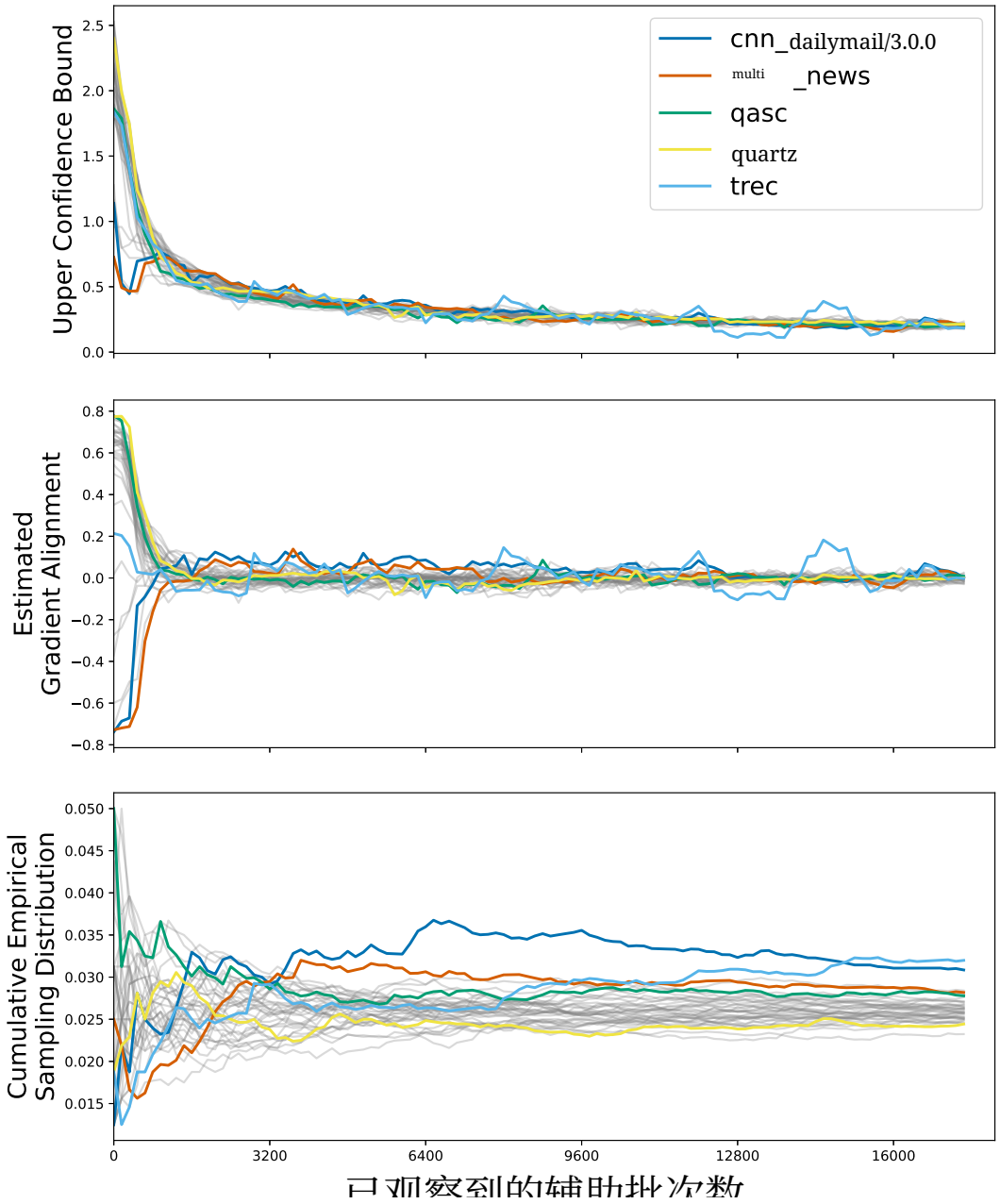

图 8：UCB1-FLAD 的训练动态案例研究，以 COPA 作为目标数据集、T0Mix 作为辅助数据，其中 EXP3-FLAD 的表现优于 UCB1-FLAD。彩色线条表示具有有趣特性的辅助数据集样本，其余数据集以灰色显示。我们发现，尽管 qasc 和 quartz 起初具有非常高的梯度对齐度，但它们很快就降至负对齐度（中间图，绿色和黄色）。最终，我们发现该算法从 qasc 中采样的数量远多于从 quartz 中采样的数量（底部图）。有趣的是，我们发现，尽管 cnn_dailymail 和 multi_news 起初的梯度对齐度都非常负，但它们很快就成为与目标任务对齐度最高的数据集（中间图，蓝色和红色）。我们发现，上置信指数最高（顶部图）且采样百分比最大（底部图）的三个辅助数据集是 cnn_dailymail、multi_news 和 trec，尽管在训练开始前，这些数据集都被认为与目标任务不相似。

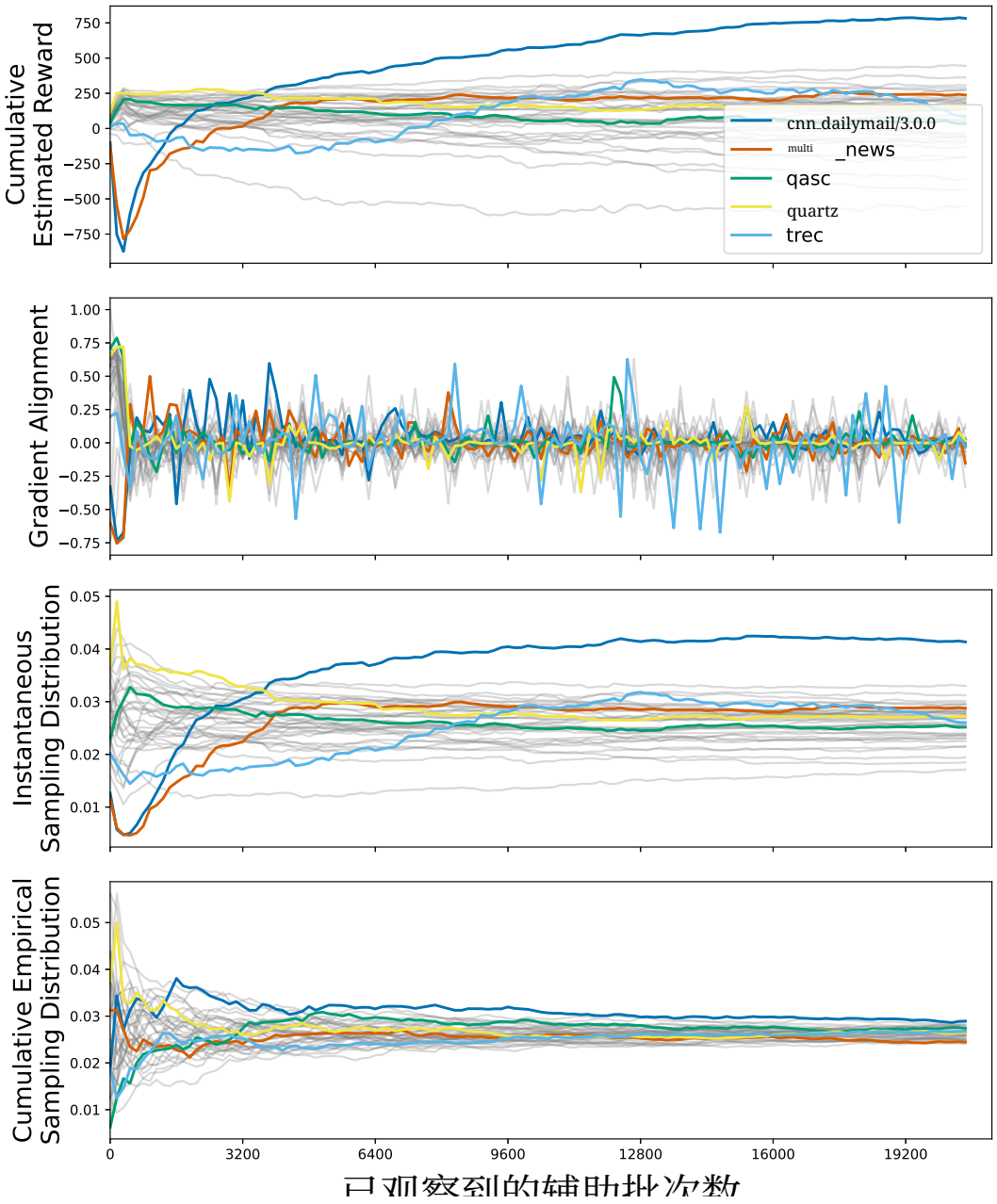

图 9：EXP3-FLAD 的训练动态：以 COPA 为目标数据集、T0Mix 为辅助数据的案例研究，此时 EXP3-FLAD 的表现优于 UCB1-FLAD。彩色线条表示具有有趣属性的部分辅助数据集，其余数据集以灰色显示。这是一个展现重要性加权估计奖励重要性的精彩示例。我们看到，cnn_dailymail 和 multi_news 的对齐度起初都非常负，但当对齐度转为正值后，EXP3 会迅速更新其估计奖励。与 RTE 类似，我们看到 EXP3 从未在经验采样分布中造成显著差异，这可能是 UCB1 整体表现优于 EXP3 的原因之一。与 RTE 相比，我们发现梯度对齐度的变化小得多，最大对齐度接近 0.5，最小对齐度接近 -0.5。而在 RTE 中，对齐度经常接近 1.0 和 -1.0。

	仅探索	仅利用	EXP3-FLAD ($\mathcal{R}^{GA}$)	UCB1-FLAD ($\mathcal{R}^{GA}$)
$\|\mathcal{A}\| = 35$ (T0Mix)	570.9	549.1	769.1	700.0
$\|\mathcal{A}\| = 260$ (P3)	863.6	692.7	832.7	794.5
% 增幅	51.3%	26.2%	8.3%	13.5%

表3：使用训练方法（列）和一组辅助数据集（行）使 T0-3B 收敛所需的训练迭代次数。收敛所需的迭代次数取 11 个目标数据集和 5 个随机种子的平均值，因此每个单元格汇总了 55 次实验。

G 缩放 $|\mathcal{A}|$ 对收敛时间的影响

正如我们在本文中所述，我们方法的单轮计算复杂度与辅助数据集的数量无关。然而，目前尚不清楚多臂老虎机的计算复杂度是否会通过其探索率而依赖于辅助数据集的数量。因此，单次训练运行的计算复杂度可能会受到辅助数据集数量（$|\mathcal{A}|$）的影响，但由于训练和大语言模型所涉及的复杂且随机的过程，无法明确刻画这种关系。

为了更好地了解增加 $|\mathcal{A}|$ 对模型收敛时间的实证影响，我们研究了不同FLAD算法收敛所需的迭代次数。表3显示，从 $|\mathcal{A}| = 35$增加到260时，所有方法都需要更长的训练时间才能收敛。我们发现，与基线方法相比，对于规模较小的辅助数据集集合，我们的基于多臂老虎机的方法需要更多步骤，但将 $|\mathcal{A}|$ 增加近10倍时，训练我们的方法所需的额外步骤数仅略有增加（$\sim 10\%$）。相比之下，随着辅助数据集数量增加，仅探索和仅利用方法的扩展性都差得多。值得注意的是，对于P3，探索类方法所需的训练迭代次数比T0Mix多出50%以上，此时其收敛时间已超过任一基于多臂老虎机的方法。

H 辅助数据集

这里列出了用于训练模型以完成ANLI目标任务的P3 [23] 所用的完整辅助数据集清单。由于需要对测试集进行去污染，其他目标数据集所用的辅助数据集略有不同，但总体上相同。数据集按其在HuggingFace Datasets[2]中的名称列出。

Zaid/quac_扩展，缩略词_识别，ade_语料库_v2/Ade_语料库_v2_分类，ade_语料库_v2/Ade_语料库_v2_药物_ade_关系，ade_语料库_v2/Ade_语料库_v2_药物_剂量_关系，对抗_问答/对抗QA，adversarial_qa/dbert，adversarial_qa/dbidaf，adversarial_qa/droberta，aeslc，ag_新闻，ai2_arc/ARC-Challenge，ai2_arc/ARC-Easy，amazon_情感极性，ama- zon_评论_multi/en，amazon_us_评论/Wireless_v1_00，ambig_qa/light，app_评论，aqua_rat/raw，art，asset/ratings，asset/simplification，banking77，billsum，bing_冠状病毒_查询_集，biosses，blbooksgenre/title_genre_分类，blended_skill_talk，cbt/CN，cbt/NE，cbt/P，cbt/V，cbt/raw，cc_新闻，circa，气候_fever，cnn_dailymail/3.0.0，codah/codah，codah/fold_0，codah/fold_1，codah/fold_2，codah/fold_3，codah/fold_4，code_x_glue_tc_文本_转_代码，com- mon_gen，常识_qa，conv_ai，conv_ai_2，conv_ai_3，cord19/metadata，cos_e/v1.0，cos_e/v1.11，cosmos_qa，covid_qa_castorini，craffel/openai_lambada，craigslist_特价商品，crows_pairs，dbpedia_14，discofuse/discofuse-sport，discofuse/discofuse-wikipedia，discov- ery/discovery，docred，dream，drop，duorc/ParaphraseRC，duorc/SelfRC，e2e_nlg_清理版，ecthr_cases/alleged-violation-prediction，emo，emotion，扩展_web_nlg/en，esnli，ev- idence_infer_treatment/1.1，evidence_infer_treatment/2.0，fever/v1.0，fever/v2.0，finan- cial_phrasebank/sentences_allagree，freebase_qa，生成_评论_enth，gigaword，glue/ax，glue/cola，glue/mnli，glue/mnli_matched，glue/mnli_mismatched，glue/mrpc，glue/qnli，glue/qqp，glue/rte，glue/sst2，glue/stsb，glue/wnli，google_wellformed_query，great_code，guardian_authorship/cross_genre_1，guardian_authorship/cross_topic_1，guardian_authorship/cross_topic_4，guardian_authorship/cross_topic_7，gutenberg_time，hans，仇恨_speech18，head_qa/en，健康_事实，hlgd，hotpot_qa/distractor，hotpot_qa/fullwiki，

[2]https://huggingface.co/datasets

humicroedit/subtask-1，humicroedit/subtask-2，hyperpartisan_新闻_检测/byarticle，hyperparti- san_新闻_检测/bypublisher，IMDb，JFLEG，KELM，kilt_任务/HotpotQA，kilt_任务/自然问题，LAMA/T-REx，LAMBADA，LIAR，LIMIT，数学_数据集/代数__线性_1d，数学_数据集/代数__线性_1d_组合，数学_数据集/代数__线性_2d，数学_数据集/代数__线性_2d_组合，数学_问答，mc_taco，MDD/task1_问答，MDD/task2_推荐，MDD/task3_问答推荐，MEDAL，医学_问题_对，meta_woz/对话，MOCHA，电影_理由，multi_新闻，multi_自然语言推理，multi_x_科学_摘要，MWSC，NarrativeQA，ncbi_疾病，神经_代码_搜索/评估_数据集，NewsPop，nlu_评估_数据，自然问题_open，numer_sense，onestop_英语，openai_humaneval，OpenBookQA/additional，open- bookqa/main，PAWS-X/en，PAWS/labeled_final，PAWS/labeled_swap，PAWS/unlabeled_final，PIQA，poem_情感，PubMed_qa/pqa_labeled，qa_srl，qa_zre，qasc，QED，QuAC，QuAIL，QuaRel，quartz，Quora，Quoref，RACE/all，RACE/high，RACE/middle，riddle_sense，ROPES，烂_番茄，SAMSum，SCAN/addprim_jump，SCAN/addprim_turn_left，SCAN/filler_num0，SCAN/filler_num1，SCAN/filler_num2，SCAN/filler_num3，SCAN/length，SCAN/simple，SCAN/template_around_right，SCAN/template_jump_around_right，SCAN/template_opposite_right，SCAN/template_right，SciCite，科学_论文/arxiv，科学_论文/pubmed，SciQ，SciTail/snli_format，SciTail/tsv_format，SciTLDR/摘要，SelQA/答案_选择_分析，sem_eval_2010_任务_8，sem_eval_2014_任务_1，sent_comp，SICK，sms_垃圾邮件，snips_内置_意图_，SNLI，social_i_qa，species_800，SQuAD，squad_adversarial/AddSent，squad_v2，SQuADShifts/amazon，SQuADShifts/new_wiki，SQuADShifts/nyt，SST/default，stsb_multi_mt/en，SubjQA/books，SubjQA/electronics，SubjQA/grocery，SubjQA/movies，SubjQA/restaurants，SubjQA/tripadvisor，super_glue/axb，super_glue/axg，super_glue/BoolQ，su- per_glue/multirc，super_glue/record，SWAG/regular，tab_fact/tab_fact，tmu_gfm_数据集，trec，trivia_qa/unfiltered，Turk，tweet_eval/emoji，tweet_eval/emotion，tweet_eval/hate，tweet_eval/irony，tweet_eval/offensive，tweet_eval/sentiment，tweet_eval/stance_abortion，tweet_eval/stance_atheism，tweet_eval/stance_climate，tweet_eval/stance_feminist，tweet_eval/stance_hillary，ty- diqa/primary_任务，tydiqa/secondary_任务，web_问题，wiki_bio，wiki_hop/masked，wiki_hop/original，wiki_qa，wiki_split，wino_bias/type1_anti，wino_bias/type1_pro，wino_bias/type2_anti，wino_bias/type2_pro，winograd_wsc/wsc273，winograd_wsc/wsc285，WIQA，XNLI/en，XQuAD/xquad.en，xquad_r/en，XSum，yahoo_answers_qa，yahoo_answers_topics，yelp_情感极性，yelp_评论_全文，ZEST