ClusterUCB：用于大模型的针对性微调的高效基于梯度的数据选择

Zige Wang[1], Qi Zhu, Fei Mi, Minghui Xu, Ruochun Jin[2], Wenjing Yang[2*]

[1]School of Computer Science, Peking University,
[2]College of Computer Science and Technology, National University of Defense Technology
Correspondence: wenjing.yang@nudt.edu.cn

Abstract

基于梯度的数据影响近似已经被用于在大语言模型的监督微调中选择有用的数据样本。然而，在整个微调过程中计算梯度需要过多的资源，在实践中难以实现。在本文中，我们提出了一种结合聚类和改进的上置信界（UCB）算法的高效梯度数据选择框架。基于具有相似梯度特征的数据样本拥有相似影响的直觉，我们首先对训练数据池进行聚类。然后，我们将聚类间的数据选择视为一个受限的计算预算分配问题，并将其视为多臂老虎机问题。我们利用改进的 UCB 算法来解决这个问题。具体而言，在迭代的采样过程中，记录历史数据影响信息，以直接估计每个聚类的分布，并采用冷启动来平衡探索与利用。在各种基准测试上的实验结果表明，我们提出的框架 ClusterUCB 可以在大大减少计算消耗的同时，实现与原始的基于梯度的数据选择方法相当的结果。

1 介绍

在大规模语言模型（LLMs）的监督微调（SFT）中，数据选择一直是一个具有挑战性的问题 (Wang et al., 2023b; Albalak et al., 2024)。一些研究人员提出使用数据影响近似 (Hampel, 1974) 来选择在训练过程中对目标损失优化影响最大的样本 (Charpiat et al., 2019; Pruthi et al., 2020; Xia et al., 2024; Wang et al., 2025)。在某个训练步骤的数据影响被近似为训练和目标验证数据样本梯度的内积或余弦相似度。然后，在每个短暂的周期之后计算一次步骤的数据影响近似，并在整个训练过程中进行汇总。

尽管已被证明有效，数据影响近似的计算消耗了许多计算资源，并且在计算预算受限时在实践中可能无法实现。为减少资源消耗，一种简单的方法是在较长的训练周期后计算所有数据样本的梯度。之前的研究表明 (Wang et al., 2025)，随着多次训练步骤的进行，基于梯度近似的数据影响很快会失去其指示性。因此，仅仅延长每两次梯度计算之间的间隔可能会导致质量较差的数据子集被选择。

另一种降低计算成本的方法是减少在单步数据影响近似计算中所需的数据样本数量，同时保持选择具有最高影响的数据样本的能力。从数据影响近似的推导中，我们得出一个直觉，即梯度相似的训练数据样本倾向于对同一目标损失优化产生相似的影响。因此，我们首先根据它们梯度的相似性在训练开始时对所有训练数据样本进行聚类。通过这种方式，具有高影响的数据样本往往集中在少数几个聚类中。通过选出更可能包含高影响数据样本的聚类，我们可以避免对大量低影响训练数据样本的计算。

在计算预算受限的情况下，我们将不同集群之间的数据选择框架视为一个计算预算分配问题，我们称之为“跨集群数据选择”。然而，挑战存在于每个集群的影响分布未知。为了应对这一挑战，我们将其视作一个多臂老虎机问题，每个集群作为一个臂，从该集群随机选取样本（不回采）的影响是抽取这个臂的奖励。然后，我们将最常用的上置信界（UCB）算法适应于跨集群数据选择问题。为了更好地适应我们的目标，我们记录历史奖励并直接估计每个臂的分布。引入冷启动以改善初始估计。通过将集群与修改后的 UCB 算法结合起来，我们提出了一个高效的基于梯度的数据选择框架 ClusterUCB，可应用于各种基于梯度的数据选择方法。为了验证 ClusterUCB 的有效性，我们在四个广泛使用的基准和两种最先进的基于梯度的数据选择方法 (Xia et al., 2024; Wang et al., 2025) 上进行了评估。实验结果表明，ClusterUCB 能够在大幅降低计算消耗的同时实现与原始基于梯度的数据选择方法相媲美的结果。

2 相关工作

大型语言模型的 SFT 数据选择 为大语言模型的监督微调选择合适的数据样本一直是研究界的一个热门话题 (Wang et al., 2023b; Albalak et al., 2024)。为了解决 SFT 数据选择的不同问题，提出了扩展的研究，如数据质量 (Zhou

et al., 2023; Cao et al., 2023; Lu et al., 2023a) 、多样性 (Lu et al., 2023b; Wan et al., 2023; Ding et al., 2023) 、复杂性 (He et al., 2024; Zhao et al., 2023) 等等。当一些工作试图通过适当的任务组合来改进多任务 SFT 时 (Dong et al., 2024; Kung et al., 2023) , 也有一些工作专注于单一任务的目标 SFT 中的数据选择问题 (Chen et al., 2024; Xia et al., 2024; Wang et al., 2025) 。在这些研究中，基于梯度的数据影响近似被用来评估目标 SFT 过程中单个数据样本的价值。例如，LESS (Xia et al., 2024) 将经典的数据影响近似调整为适用于 Adam 优化器和 LoRA (Hu et al., 2022) 训练。Dynamic (Wang et al., 2025) 指出在长期训练过程中选择的有效性递减，并提出动态更新选择的数据核心集。虽然有效，但这些提出的基于梯度的数据选择方法在大量数据样本的梯度计算中需要高计算消耗。我们的工作解决了这个限制，并提出了一种高效的基于梯度的数据选择框架，结合了聚类和修改的 UCB 算法。

个体数据影响 数据影响函数被提出用于评估数据样本对模型训练的影响 (Hampel, 1974) 。由于评估不同数据样本组合的成本太高，一些研究人员倾向于评估单个数据样本的影响，并将这些影响的总和视为数据子集的影响。在单个数据影响近似中有两个分支：一个是辅助模型学习和模拟 (Ilyas et al., 2022; Guu et al., 2023; Liu et al., 2024; Engstrom et al., 2024) ，另一个是基于梯度的训练动态近似 (Charpiat et al., 2019; Pruthi et al., 2020; Xia et al., 2024; Wang et al., 2025) 。在我们的工作中，我们提出了一个框架，以有效地在 LLM 的 SFT 数据选择中应用基于梯度的单个数据影响近似。

3 方法论

我们的工作重点是有效的数据选择，以对大型语言模型进行监督微调以适应目标任务。给定一个参数为 $\boldsymbol{\theta}$ 的预训练模型、训练损失 $\mathcal{L}(\cdot;\boldsymbol{\theta})$ 、训练数据池 $\boldsymbol{X}_{tr} = \{\boldsymbol{x}_{tr}^1, \boldsymbol{x}_{tr}^2, ..., \boldsymbol{x}_{tr}^N\}$ 、目标任务 T 以及代表目标任务的验证数据样本 $\boldsymbol{X}_v = \{\boldsymbol{x}_v^1, \boldsymbol{x}_v^2, ..., \boldsymbol{x}_v^M\}$ ，我们的目标是在有限的计算预算约束下，选择对训练过程针对任务 T 具有最大数据影响的训练数据子集。

在详细阐述我们提出的框架之前，我们首先介绍基于梯度的数据影响近似计算以及基于此的先前提出的数据选择方法。

在训练过程中的时间步 t 时，模型参数为 $\boldsymbol{\theta}^t$ 。考虑训练数据样本 $\boldsymbol{x}_{tr}^i$ 和验证数据样本 $\boldsymbol{x}_v^j$ ，$\boldsymbol{x}_{tr}^i$ 的单步影响 $\mathcal{I}^t$ 定义为在对 $\boldsymbol{x}_{tr}^i$ 进行一步训练后，$\boldsymbol{x}_v^j$ 的损失减少量。这个可以使用一阶泰勒展开近似：其中，$\nabla\mathcal{L}(\boldsymbol{x}_v^j;\boldsymbol{\theta}^t)$ 是关于 $\boldsymbol{\theta}^t$ 的 $\boldsymbol{x}_v^j$ 的梯度，$\langle\cdot,\cdot\rangle$ 是内积。

由于 Adam 优化器是 LLMs 的 SFT 中最常用的优化器，参数更新 $\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t$ 可以用 Adam 中的适应梯度 Γ 表示。然后，影响可以被近似为：

$$\mathcal{I}^t(\boldsymbol{x}_{tr}^i, \boldsymbol{x}_v^j) \approx -\eta^t \langle \nabla\mathcal{L}(\boldsymbol{x}_v^j;\boldsymbol{\theta}^t), \Gamma(\boldsymbol{x}_{tr}^i;\boldsymbol{\theta}^t)\rangle, \tag{1}$$

，其中 $\Gamma(\boldsymbol{x}_{tr};\boldsymbol{\theta}^t) = -\eta^t \frac{\boldsymbol{m}^t}{\sqrt{\boldsymbol{v}^t+\epsilon}}$ ，$\boldsymbol{m}^t$ 和 $\boldsymbol{v}^t$ 分别是历史梯度及其元素平方的移动平均值。

在之前的研究中 (Xia et al., 2024; Wang et al., 2025) ，数据影响近似被采用来为目标任务上的 SFT 选择最有利的训练数据样本。正如之前的研究所探讨的那样 (Wang et al., 2025) ，为了减轻选择长度偏差，在计算内积之前，公式 1 中的梯度被规范化，如下所示：

$$\widetilde{\mathcal{I}}^t(\boldsymbol{x}_{tr}^i, \boldsymbol{x}_v^j) \approx \langle \frac{\nabla\mathcal{L}(\boldsymbol{x}_v^j;\boldsymbol{\theta}^t)}{\|\nabla\mathcal{L}(\boldsymbol{x}_v^j;\boldsymbol{\theta}^t)\|}, \frac{\Gamma(\boldsymbol{x}_{tr}^i;\boldsymbol{\theta}^t)}{\|\Gamma(\boldsymbol{x}_{tr}^i;\boldsymbol{\theta}^t)\|}\rangle. \tag{2}$$

Wang et al. (2025) 指出，一步影响近似的指示在训练过程中有下降趋势。因此，他们建议在每个训练周期后动态地重新计算数据影响近似并更新所选数据子集。而 LESS (Xia et al., 2024) 则通过在一个随机选择的数据子集上使用模拟训练来获得多个模型检查点，并仅使用在所有检查点聚合的一步数据影响近似进行一次数据选择。

在每一次选择中，数据影响近似 $\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}^i, \boldsymbol{x}_v^j)$ 在所有验证数据样本上被汇总。具体来说，首先在每个子任务内对 $\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}^i, \boldsymbol{x}_v^j)$ 进行平均，然后在子任务中选择最大值作为验证数据集上的数据影响近似 $\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}^i)$ 。之后，选择数据影响近似值最高的前 $p\%$ 个训练数据样本，构成所选的数据子集。

为了减少基于梯度的数据选择中的计算消耗，一个简单的方法是减少用于计算一步数据影响近似的检查点数量，即延长每两次计算之间的间隔。然而，先前研究中所展示的选择有效性长期下降现象表明，过长的间隔会导致高度不准确的影响近似，这可能阻碍数据选择。因此，我们致力于减少在每次计算一步数据影响近似时需计算的梯度数量。

根据方程 2 ，单步数据影响近似 $\widetilde{\mathcal{I}}^t(\boldsymbol{x}_{tr}^i, \boldsymbol{x}_v^j)$ 是训练数据样本 $\boldsymbol{x}_{tr}^i$ 的梯度与验证数据样本 $\boldsymbol{x}_v^j$ 的梯度之间的余弦相似度。因此，一个直观的理解是，两个训练数据样本梯度具有较高余弦相似性的情况下，它们与同一验证数据样本梯度的余弦相似度程度将会相似。基于这一直觉，我们根据与预训练模型参数 $\boldsymbol{\theta}^0$ 的余弦相似度对所有训练数据样本的梯度进行聚类。我们的实验观察显示，聚类可以在模型训练过程中保持相对紧密，这一点在附录 B 中有讨论。

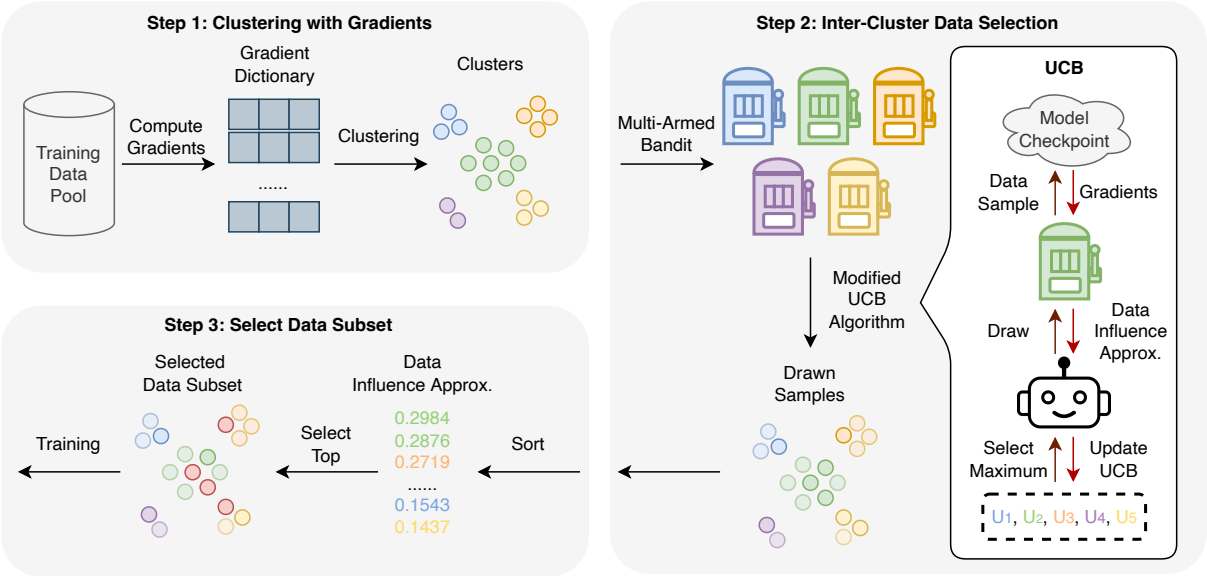


Figure 1: ClusterUCB 示例。步骤 1：我们首先计算所有训练数据样本的梯度，并根据它们的余弦相似度进行聚类。步骤 2：我们将簇间数据选择设定为一个多臂赌博机问题，将每个簇视作一个赌博臂。数据影响的估算被视为抽取的奖励。使用一个修改过的 UCB 算法来在有限的计算预算下进行样本抽取。步骤 3：从抽取的样本中，我们选择影响力最大的部分数据样本作为选定的数据子集。

然后，具有最高影响力的数据样本应该集中在少数几个簇中。将每个簇视为数据影响力上的一种分布，我们可以根据它们包含高影响力训练数据样本的概率将计算资源分配给这些簇。通过这种方式，我们可以避免对大量低影响力训练数据样本进行计算，从而节省大量计算资源。

3.1 使用 UCB 算法的簇间数据选择

获得训练数据样本的 k 个簇后，我们的下一步是通过在不同簇之间分配计算预算来最大化发现具有最高影响力的训练数据样本的整体概率，我们称之为受限计算预算的簇间数据选择问题：

$$
\begin{aligned}
B^* = &\arg\max \sum_{c=1}^{k} b_c P_{\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr})\sim \boldsymbol{P}_c}(\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}) \geq T), \\
s.t. &\sum_{c=1}^{k} b_c = \mathcal{B}, \\
&\forall c \in \{1, ..., k\}, 0 \leq b_c \leq |C_c|,
\end{aligned} \tag{3}
$$

其中 $B = \{b_1, b_2, ..., b_k\}$ 是计算预算 $\mathcal{B}$ 的分配，$\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr})$ 是对 $\boldsymbol{x}_{tr}$ 的数据影响估计，$\boldsymbol{P}_c$ 是包含在簇 C_c 中的数据影响的分布，而 T 是实际前 $p\%$ 个训练数据样本中影响最大的样本的最低影响。

一个挑战在于每个簇的分布 $\boldsymbol{P}_c$ 是未知的。因此，$\boldsymbol{P}_c$ 的估计需要通过簇间数据选择自发地进行。这个问题设置与著名的多臂老虎机问题 (Slivkins et al., 2019) 非常相似。具体来说，每个簇可以认为是一个具有未知分布的“臂”。一旦在每一轮绘制中选择了一个“臂”，将从相应的簇随机选择一个训练数据样本。然后，其数据影响近似值将被计算为这一轮的绘制奖励。通过这种方式，绘制轮次的数量就是消耗在数据影响近似计算上的计算预算。因此，优化公式 3 中的目标意味着最大化总的绘制奖励。

为了解决这个问题，我们将常用的上置信界 (UCB) 算法 (Auer et al., 2002) 进行适配。UCB 算法的核心思想是为每个在我们设定中对应于簇 C_c 的臂估计一个上置信界 U_c。在每一轮选择中 d，选择具有最大估计上置信界 U_d^* 的簇 C_d^*；然后，获取本次选择的奖励 C_d^*，并用其更新 U_d^*。通过重复这个过程，簇 C_d 的估计上置信界将更接近于其实际期望奖励的值 C_d。因此，期望奖励最高的簇 C^* 将被分配最多的计算预算，而期望奖励较低的簇将被分配较少的计算预算。

在经典的 UCB 算法 (Auer et al., 2002) 中，上置信界是每个臂均值估计置信区间的上界。由于我们在公式 3 中的目标比分布的均值更关注大于某一阈值的概率，我们记录所有历史抽取奖励，并用它们来估计每个簇的分布。由于实际的 T 也是未知的，直接估计 $P_{\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr})\sim \boldsymbol{P}_c}(\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}) \geq T)$ 可能具有挑战性。因此，我们比较每个簇的 $\hat{T}_c$，使得所有簇的 $P^c_{\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr})\sim \boldsymbol{P}_c}(\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}) \geq \hat{T}_c)$ 大致相等。在实践中，

我们使用历史抽取奖励估计的平均值 $\hat{\mu}_c$ 和标准差 $\hat{\sigma}_c$ 来计算 $\hat{T}_c$，并将其视为簇 C_c 的上置信界 U_c：

$$U_c = \hat{T}_c = \hat{\mu}_c + \beta * \hat{\sigma}_c, \tag{4}$$

其中 β 是一个在实践中设置为 1 的超参数。

在 UCB 算法的初始阶段，不足的历史抽奖回报可能会导致上置信界的估计不佳和目标优化中的巨大遗憾。因此，我们在 UCB 算法中应用冷启动，它将一小部分 $p_{cs}\%$ 计算预算根据集群大小分配给所有集群。在冷启动期间结束后，算法开始在每轮抽奖中选择具有最大估计上置信界的集群。

3.2 高效的基于梯度的数据选择框架

结合聚类和使用 UCB 算法的簇间数据选择，我们提出了一种高效的基于梯度的数据选择框架 ClusterUCB，如图 1 所示。具体而言，该框架首先根据其相对于预训练模型的梯度的余弦相似度对所有训练数据样本进行聚类。在每次数据选择时，将在预定义的计算预算下应用使用 UCB 算法的簇间数据选择。

由于在绘制过程中仍然存在遗憾，计算预算通常设定得比最后所需的训练数据样本数要大。在最后一步中，我们将通过簇间数据选择计算的影响近似值从高到低排序，并输出相应的训练数据样本的前若干个作为我们最终选定的数据子集。

4 实验

4.1 实验装置

基线 随机是用随机选取的数据子集训练模型。LESS (Xia et al., 2024) 使用随机选取数据的模拟训练来获取多个模型检查点，聚合所有训练数据样本关于这些检查点的一步数据影响近似值，并选择具有最高聚合影响近似值的前 $p\%$ 个训练数据样本作为最终选定数据子集。Dynamic (Wang et al., 2025) 直接使用一步数据影响近似值来选择前 $p\%$ 个训练数据样本，但选择过程将周期性地重复，从而通过模型训练过程动态更新数据子集。为了展示我们修改的 UCB 算法在相同计算预算下的有效性，我们还实现了两个基础版本 LESS-Rerank 和 Dynamic-Rerank，它们随机选择 $\mathcal{B}$ 个训练数据样本来计算其影响近似值，并从中选择前 $p\%$ 个高影响数据样本。

紧跟 LESS 和 Dynamic 的做法，我们使用 LLaMA-2-7B (Touvron et al., 2023) 作为我们的预训练模型，并将选择比例 $p\%$ 设置为 5 %。预训练模型使用 AdamW 优化器训练四个周期。学习率为 2e-5，并进行线性衰减。对于 Random，以随机方式选择 5 % 的训练数据样本来训练模型。对于 LESS 和 Dynamic，执行与其原始论文中描述的相同：在 LESS 中，使用 5 % 随机选取的训练数据样本进行模拟训练，以在每个周期结束后获得四个检查点；在 Dynamic 中，在每个周期开始时执行一次性数据选择，并在第一次一次性数据选择时进行 20 步预热。对于 ClusterUCB，我们也执行 20 步预热，并使用所得检查点来计算所有训练数据样本的梯度。然后，采用 K-means (Hartigan and Wong, 1979) 进行聚类。我们将 ClusterUCB 与 LESS 和 Dynamic 结合，分别形成两个变体，LESS-ClusterUCB 和 Dynamic-ClusterUCB。在 LESS-ClusterUCB 中，每次抽取的奖励是 LESS 中的聚合一次性数据影响近似。在 Dynamic-ClusterUCB 中，由于用于聚类和第一次一次性数据选择的梯度是相同且完整的，我们将第一训练周期保持与 Dynamic 相同，然后在接下来的三个一次性数据选择中应用我们提出的簇间数据选择。在我们的主要实验中，对于 LESS-ClusterUCB、Dynamic-ClusterUCB、LESS-Rerank 和 Dynamic-Rerank，簇的数量 k 为 150，冷启动比例 $p_{cs}\%$ 为 5 %，计算预算 $\mathcal{B}$ 为总训练数据样本数量的 20 %。所有梯度均使用 LoRA (Hu et al., 2022) 计算，并通过 Random Projection (Park et al., 2023) 投影到 8192 维向量。

数据集 训练数据池是 8 个常用数据集的混合：Flan-v2 (Longpre et al., 2023) 是一个从各种 NLP 数据集中转换而来的大型 SFT 数据集；CoT (Longpre et al., 2023) 是 Flan-v2 的一个包含思维链的子集；Dolly (Conover et al., 2023) 是由人工生成的高质量指令跟随数据集；Open Assistant v1 (Köpf et al., 2023) 是由人工和开源 LLM 生成的多轮聊天数据集；GPT4-Alpaca (Peng et al., 2023) 包含 Alpaca 数据集中的指令和由 GPT-4 重新生成的答案；ShareGPT (Chiang et al., 2023) 是一个质量混合的对话数据集；GSM8k train (Cobbe et al., 2021) 是一个小学水平的数学文字数据集；Code-Alpaca (Chaudhary, 2023) 是一个为开发模型的编码能力而设计的数据集。

评估基准和验证数据样本 我们采用四个常用的基准，覆盖 LLM 的一般、多语言、数学和编码能力。MMLU (Hendrycks et al., 2021) 是一个基于知识的多项选择问答基准，包括 57 个科目；TydiQA (Clark et al., 2020) 是一个多语言问答基准，包含九种语言；GSM8k (Cobbe et al., 2021) 是一个评估模型数学推理能力的数学推理基准；HumanEval (Chen et al., 2021) 是一个 Python 编码基准，评估模型的代码生

Methods	Budget	MMLU	TydiQA	GSM8k	HumanEval	Avg.	Δ
Random	-	45.3 (0.5)	48.5 (0.6)	19.7 (0.1)	16.5 (0.8)	32.5	-
LESS	100 %	47.0 (0.5)	53.2 (1.1)	27.3 (0.6)	17.7 (0.9)	36.3	-
LESS-Rerank	20 %	45.9 (0.2)	52.2 (0.3)	23.8 (0.1)	16.3 (0.3)	34.6	↓ 1.7
LESS-ClusterUCB	20 %	47.8 (0.8)	53.7 (1.3)	28.0 (1.0)	17.6 (0.7)	36.8	↑ 0.5
Dynamic	100 %	47.8 (0.3)	57.6 (0.8)	27.5 (1.2)	19.2 (0.4)	38.0	-
Dynamic-Rerank	20 %	46.7 (0.4)	54.9 (0.9)	26.7 (0.6)	17.9 (0.7)	36.6	↓ 1.4
Dynamic-ClusterUCB	20 %	47.9 (0.6)	57.4 (0.6)	27.4 (0.9)	17.7 (0.4)	37.6	↓ 0.4

Table 1: ClusterUCB 和基线在四个常用基准上的结果。所有实验都用三个随机种子重复。Δ 表示预算约束方法与其全预算对应方法之间的平均性能差异。加粗表示预算约束方法所实现的最佳结果。

Bgt	MMLU	TydiQA	GSM8k	HE
10 %	46.3 (1.1)	55.4 (0.5)	28.8 (0.5)	18.8 (0.6)
20 %	47.9 (0.6)	57.4 (0.6)	27.4 (0.9)	17.7 (0.4)
30 %	47.3 (0.2)	57.5 (0.5)	27.2 (1.4)	18.3(0.3)

Table 2: Dynamic-ClusterUCB 在不同计算预算下的结果。Bgt 和 HE 分别是 Budget 和 HumanEval 的缩写。

Methods	TydiQA	HumanEval
Random	64.5 (0.2)	38.4 (1.2)
LESS	66.8 (0.5)	36.9 (1.5)
LESS-ClusterUCB	66.7 (0.5)	38.1 (0.9)
Dynamic	67.3 (0.6)	40.0 (1.5)
Dynamic-ClusterUCB	67.8 (0.1)	40.9 (1.2)

Table 3: 使用 Qwen2.5-3B 作为预训练模型的 ClusterUCB 和基线的结果。

成能力。验证数据样本的选择和聚合对于所有方法都遵循 Dynamic：MMLU 和 TydiQA 的少量样本直接用作验证数据样本；从 GSM8k 和 HumanEval 中随机选择 50 个和 10 个测试数据样本作为验证数据样本。

4.2 主要结果

ClusterUCB 和基线在四个基准上表现的结果显示在表 1 中。所有基于梯度的方法在四个基准的平均性能上都远远优于随机，表明基于梯度的方法在选择适合的定向微调 LLM 数据子集方面是有效的。

在计算预算设置为 20 % 的情况下，LESS-ClusterUCB 和 Dynamic-ClusterUCB 在大多数基准测试中符合其全预算对应方案 LESS 和 Dynamic。这些结果表明，ClusterUCB 可以在保持性能的同时减少基于梯度的数据选择方法的计算消耗。虽然 Dynamic-ClusterUCB 在 HumanEval 上的性能相比 Dynamic 有所下降，但在第 4.3 节中，我们发现当计算预算减少到 10 % 时，它可以实现更好的结果。

在相同的计算预算下，LESS-ClusterUCB 和 Dynamic-ClusterUCB 在几乎所有基准上都优于 LESS-Rerank 和 Dynamic-Rerank，这进一步表明了 ClusterUCB 中使用的选择策略的有效性。

4.3 计算预算的影响

为了说明计算预算的影响，我们在 Dynamic-ClusterUCB 上进行实验，计算预算为 $\mathcal{B}$ =10 %、20 % 和 30 %。冷启动比率 p_{cs}% 为 5 %，簇的数量为 150，与主要实验中相同。表 2 中的结果显示了不同基准随着计算预算的变化模式。在 MMLU 和 TydiQA 上，当 $\mathcal{B}$ 只有 10 % 时，Dynamic-ClusterUCB 的表现较差。$\mathcal{B}$ = 20 % 就足够了，因为将 $\mathcal{B}$ 从 20 % 增加到 30 % 只带来了微小的性能提升。相反，在 GSM8k 和 HumanEval 上，较小的计算预算往往导致更高的准确率。原因可能是有助于提升数学和编码能力的训练数据样本分布在少数几个簇中。因此，一旦我们的 UCB 算法找到正确的臂，较小的计算预算就足以找到影响力高的数据样本。GSM8k 和 HumanEval 上随着计算预算增加而性能下降可能是由于数据选择和训练的随机性导致的。

4.4 在不同模型上的结果

为了在不同的模型架构和规模上进一步评估 ClusterUCB，我们在 TydiQA 和 HumanEval 基准上使用 Qwen2.5-3B (Team, 2024) 作为预训练模型进行实验。实现细节与我们的主要实验保持一致，除了所有模型都训练了三个 epoch。

结果如表 3 所示。与使用 LLaMA-2-7B 作为预训练模型的结果一致，LESS-ClusterUCB 和 Dynamic-ClusterUCB 在这两个基准上与其全预算对应的 LESS 和 Dynamic 相匹配，显示了 ClusterUCB 在不同模型架构和规模上的有效性。

我们分析了 ClusterUCB 中两个关键超参数的影响：簇的数量 k 和冷启动比例 p_{cs}% 。我们采用两个指标来评估所选超参数的优劣。一个是样本级召回率 R_s ，另一个是影响级召回率 R_{inf} ，分别如等式 5 和 6 所示，其中 D 是

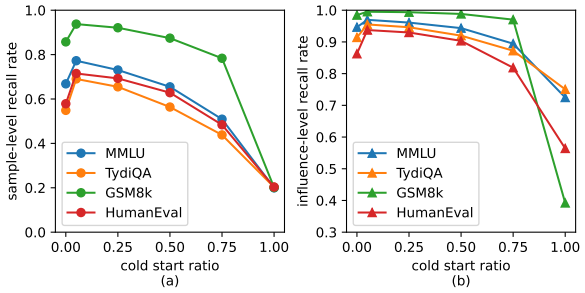


Figure 2: 不同冷启动比例下的样本级和影响级召回率。

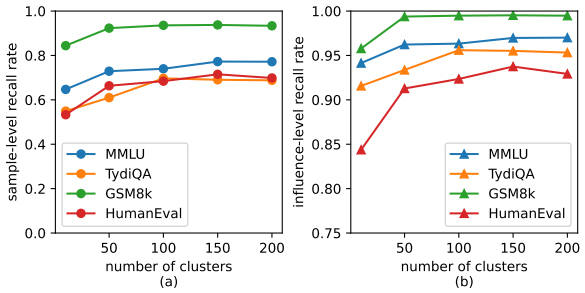


Figure 3: 样本级别和影响级别的召回率与不同数量的簇。

ClusterUCB 选择的最终数据子集，而 D_{gt} 是具有最高影响的实际训练数据样本的顶部部分。

$$R_s = \frac{|D \bigcap D_{gt}|}{|D_{gt}|} \tag{5}$$

$$R_{inf} = \frac{\sum_{\boldsymbol{x}_{tr}^i \in D} \widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}^i)}{\sum_{\boldsymbol{x}_{tr}^q \in D_{gt}} \widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}^q)}. \tag{6}$$

我们在经过预热训练后获得的模型检查点上计算 R_s 和 R_{inf} ，并且在本节中固定计算预算 $\mathcal{B}$ 为 20 %。

4.4.1 冷启动比率

设 $k = 150$，我们评估 $p_{cs}\% = \{$ 0 %，5 %，25 %，50 %，75 %，100 % $\}$，如图 2 所示。

当 $p_{cs}\% = 0$ % 时，所有任务的 R_s 和 R_{inf} 都很低。但是当 $p_{cs}\%$ 增加到 5 % 时，R_s 和 R_{inf} 显著提高。这表明在簇间数据选择的初始阶段，具有冷启动的充足历史奖励信息是必要的。

当 $p_{cs}\%$ 增加到 100 % 时，簇间数据选择退化为仅仅根据簇大小分配计算预算，而 R_s 和 R_{inf} 变得极低。这说明我们的 UCB 算法在簇间数据选择中是有效的。

随着 $p_{cs}\%$ 从 5 % 增加，R_s 和 R_{inf} 逐渐减少，表明仅有小部分计算预算应该用于冷启动。冷启动的比例还控制着 UCB 算法中探索与利用之间的权衡。这些结果表明，在我们的 UCB 算法中，探索是必要且重要的，但过多的探索可能会阻碍算法的性能。关于群集之间数据选择的分布的进一步讨论，请参见附录 C 。

4.4.2 聚类的数量

设定 $p_{cs}\% = 5$ % ，我们评估 $k = \{$ 10, 50, 100, 150, 200 $\}$ 。结果如图 3 所示。当簇的数量小到 10 时，R_s 和 R_{inf} 在所有任务中表现最差，这表明簇数量过少可能无法将训练数据样本完全划分为具有相似梯度的组。从 10 增加到 50 时，R_s 和 R_{inf} 显示出明显的改进。进一步增加簇的数量，改进变得不那么明显，且 R_s 和 R_{inf} 趋于稳定。它也表明，只要簇的数量不是太少，ClusterUCB 对簇数量不敏感。

4.5 不同上置信界评估指标的比较

在第 3.1 节中，我们将上置信界 U_c 评估为与相同概率对应的估计影响阈值 $\hat{T}_c$ 。另一种方法是直接估计 $P_{\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}) \sim \boldsymbol{P}_c}(\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}) \geq T)$ 。一种估计是奖励大于 T 的抽样比例，我们称之为 UCB-TH 。我们还可以将每个簇分布 $\boldsymbol{P}_c$ 视为一个高斯分布，其均值和标准偏差从历史奖励值 $\widetilde{\boldsymbol{P}}_c \sim \mathcal{N}(\hat{\mu}_c, \hat{\sigma}_c)$ 中估计，并计算 $P_{\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}) \sim \widetilde{\boldsymbol{P}}_c}(\widetilde{\mathcal{I}}(\boldsymbol{x}_{tr}) \geq T)$ 。我们称这种估计为 UCB-TN 。由于 T 是未知的，我们估计它为当前轮次中所有簇的所有历史奖励值的最高 $p/\mathcal{B}$ 部分中的最低影响。

保持所有超参数和计算预算与主要实验中相同，我们将 UCB-TH 和 UCB-TN 与主要实验中使用的评估指标（UCB-Beta）进行比较。我们还将它们与两个基线进行比较：Random-Draw，即每轮随机选择一个臂进行抽取，以及一个经典的 UCB 算法 UCB1 (Auer et al., 2002) 。

表 4 中的结果显示，UCB-Beta 和 UCB-TH 在所有任务中都取得了最好的结果，且前者在大多数任务中略优于后者。这表明在我们的设置中，UCB-Beta 和 UCB-TH 可能是等效的。UCB-TN 比 UCB-Beta 和 UCB-TH 差，表明使用高斯分布来拟合聚类分布可能不准确。虽然 UCB1 比 Random-Draw 表现更好，但远不及 UCB-Beta、UCB-TH 和 UCB-TN，这说明仅仅估计每个聚类分布的均值不能解决跨聚类的数据选择问题。

在本文中，我们旨在减少用于大型语言模型（LLM）梯度基础的 SFT 数据选择中的计算消耗。我们提出的框架首先对训练数据池进行聚类，基于这样的直觉：具有相似梯度的训练数据样本对目标损失优化的影响也相似。然后，我们将簇间数据选择构建为一个计算预算分配问题，这类似于多臂老虎机问题，并修改 UCB 算法来解决它。结合最先进的基于梯度的数据选择方法，实验结果表明，我们提出的框架可以匹配原始方法，同时大大减少计算消耗。

	Random-Draw		UCB1		UBC-TN		UCB-TH		UCB-Beta	
Tasks	R_s	R_{inf}	R_s	R_{inf}	R_s	R_{inf}	R_s	R_{inf}	R_s	R_{inf}
MMLU	20.14	72.72	26.24	76.76	73.96	96.40	77.00	96.96	77.24	96.97
TydiQA	19.38	74.74	23.12	77.21	59.08	92.96	68.96	95.56	69.03	95.52
GSM8k	25.84	45.87	59.31	90.05	90.64	99.13	93.12	99.47	93.75	99.52
HumanEval	22.63	58.68	24.72	61.01	55.96	86.95	71.61	93.83	71.50	93.75

Table 4: 不同上置信界评估指标的样本级和影响级召回率。粗体表示每项任务的最佳结果。

虽然我们提出的框架在节省计算资源方面已被证明是高效的，但也有必要考虑其未来可能改善的局限性。正如 Wang et al. (2025) 所说，基于梯度的数据选择方法仅考虑单一数据样本的影响，而忽略了所选数据子集内的相互影响。通过生成的聚类，跨簇数据选择可以将数据样本组作为抽臂奖励考虑，这是我们工作的下一步。此外，我们在本文中使用了最简单的聚类算法 K-means。更好的聚类方法也可能改善我们所提出框架的性能。

References

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A survey on data selection for language models. Trans. Mach. Learn. Res. , 2024.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. Machine learning , 47:235–256.

Yihan Cao, Yanbin Kang, and Lichao Sun. 2023. Instruction mining: High-quality instruction data selection for large language models. arXiv preprint arXiv:2307.06290v3 .

Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. 2019. Input similarity from the neural network perspective. Advances in Neural Information Processing Systems , 32.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374v2 .

Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2024. Skill-it! a data-driven skills framework for understanding and training language models. Advances in Neural Information Processing Systems , 36.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90 % * chatgpt quality.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. Transactions of the Association for Computational Linguistics , 8:454–470.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168v2 .

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023 , pages 3029–3051. Association for Computational Linguistics.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2024. How abilities in large language models are affected by supervised fine-tuning data composition. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024 , pages 177–198. Association for Computational Linguistics.

Logan Engstrom, Axel Feldmann, and Aleksander Madry. 2024. DsDm: Model-aware dataset selection with datamodels. In Proceedings of the 41st International Conference on Machine Learning , volume 235 of Proceedings of Machine Learning Research , pages 12491–12526. PMLR.

Kelvin Guu, Albert Webson, Ellie Pavlick, Lucas Dixon, Ian Tenney, and Tolga Bolukbasi. 2023. Simfluence: Modeling the influence of individual training examples by simulating training runs. arXiv preprint arXiv:2303.08114v1 .

Frank R Hampel. 1974. The influence curve and its role in robust estimation. Journal of the american statistical association , 69(346):383–393.

John A Hartigan and Manchek A Wong. 1979. Algorithm as 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics) , 28(1):100–108.

Qianyu He, Jie Zeng, Qianxi He, Jiaqing Liang, and Yanghua Xiao. 2024. From complex to simple: Enhancing multi-constraint complex instruction following ability of large language models. arXiv preprint arXiv:2404.15846v2 .

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. Proceedings of the International Conference on Learning Representations (ICLR) .

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022 .

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. 2022. Datamodels: Predicting predictions from training data. arXiv preprint arXiv:2202.00622v1 .

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations–democratizing large language model alignment. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 .

Po-Nien Kung, Fan Yin, Di Wu, Kai wei Chang, and Nanyun Peng. 2023. Active instruction tuning: Improving cross-task generalization by training on prompt sensitive tasks. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023 , pages 1813–1829. Association for Computational Linguistics.

Qingyi Liu, Yekun Chai, Shuohuan Wang, Yu Sun, Keze Wang, and Hua Wu. 2024. On training data influence of gpt models. arXiv preprint arXiv:2404.07840v3 .

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA , volume 202 of Proceedings of Machine Learning Research , pages 22631–22648. PMLR.

Jianqiao Lu, Wanjun Zhong, Wenyong Huang, Yufei Wang, Fei Mi, Baojun Wang, Weichao Wang, Lifeng Shang, and Qun Liu. 2023a. Self: Language-driven self-evolution for large language model. arXiv preprint arXiv:2310.00533v4 .

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023b. # instag: Instruction tagging for analyzing supervised fine-tuning of large language models. Preprint , arXiv:2308.07074.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. 2023. Trak: Attributing model behavior at scale. In International Conference on Machine Learning (ICML) .

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. arXiv preprint arXiv:2304.03277v1 .

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. Advances in Neural Information Processing Systems , 33:19920–19930.

Aleksandrs Slivkins and 1 others. 2019. Introduction to multi-armed bandits. Foundations and Trends® in Machine Learning , 12(1-2):1–286.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288v2 .

Fanqi Wan, Xinting Huang, Tao Yang, Xiaojun Quan, Wei Bi, and Shuming Shi. 2023. Explore-instruct: Enhancing domain-specific instruction coverage through active exploration. arXiv preprint arXiv:2310.09168 .

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Raghavi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, and 1 others. 2023a. How far can camels go? exploring the state of instruction tuning on open resources. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 .

Zige Wang, Wanjun Zhong, Yufei Wang, Qi Zhu, Fei Mi, Baojun Wang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023b. Data management for training large language models: A survey. arXiv preprint arXiv:2312.01700v3 .

Zige Wang, Qi Zhu, Fei Mi, Yasheng Wang, Haotian Wang, and Lifeng Shang. 2025. Dynamic data selection with normalized gradient-based influence approximation for targeted fine-tuning of llms. Available at SSRN 5206107 .

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. In Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024 . OpenReview.net.

Yingxiu Zhao, Bowen Yu, Binyuan Hui, Haiyang Yu, Fei Huang, Yongbin Li, and Nevin L Zhang. 2023. A preliminary study of the intrinsic relationship between complexity and alignment. arXiv preprint arXiv:2308.05696 .

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023 .

A 训练与评估

A.1 训练数据集

我们使用了与 (Wang et al., 2025) 相同的训练数据集。每个数据集中的实例数量在表 5 中展示。我们训练数据池中平均完成令牌的数量为 189.1。将选择比例设置为 5 %，选定的数据子集包含 20387 个实例。

根据 Xia et al. (2024) 和 (Wang et al., 2025) 的方法，我们在所有实验中采用了参数高效的微调方法 LoRA (Hu et al., 2022) 。LoRA 模块的秩为 128，α 的值为 512，并且所有注意力矩阵都应用了可学习的 LoRA 矩阵。在该配置下，LLaMA-2-7B 中有 134,217,728 个可训练参数，占原始参数的 1.95 %，而 Qwen2.5-3B 中有 58,982,400 个可训练参数，占原始参数的 1.88 %。我们所有的实验都是使用 8 块 Tesla V100 GPU 进行的。

Dataset	# Instance
Flan v2	99245
CoT	95557
Dolly	14865
Open Asisstant v1	54626
GPT4-Alpaca	52002
ShareGPT	63951
GSM8k train	7473
Code-Alpaca	20021
Total	407740

Table 5: 我们实验中使用的每个数据集中的实例数量。

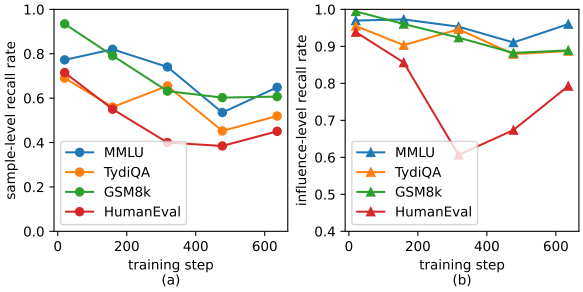


Figure 4: 训练过程中样本级和影响级召回率的变化。

A.2 评估细节

我们使用由 Open-Instruct (Wang et al., 2023a) 提供的评估代码工具包。在 MMLU 上，评估指标是模型完成中第一个标记与标准答案的精确匹配，我们在 5-shot 设置下进行评估，并对 57 个子任务取平均值；在 TydiQA 上，采用金数据段和 1-shot，性能评估为模型完成和标准答案的 F1 分数，并对九种语言取平均值；在 GSM8k 上，采用 8-shot，模型完成的最终数字被提取为最终答案，以与标准答案精确匹配；在 HumanEval 上，我们使用 pass@1 作为评估指标，并以温度 0.1 对每个指令采样 20 个完成。

B 训练过程中簇效果的下降

在我们提出的框架中，我们仅在训练开始时根据训练数据样本梯度的余弦相似性进行聚类。由于训练数据样本的梯度会随着模型权重的更新而变化，因此在训练过程中聚类是否仍然有效是一个关键问题。因此，我们进行实验来研究训练过程中聚类有效性变化的趋势。

我们再次使用公式 5 中的样本级召回率 R_s

和公式 6 中的影响级召回率 R_{inf} 来评估聚类的有效性。我们计算与在训练期间保存的检查点相关的 R_s 和 R_{inf}，采用 5 % 随机选择的数据样本。聚类数 k = 150，冷启动比例 $p_{cs}\%$ = 5 %，与我们的主要实验相同。结果如图 4 所示。

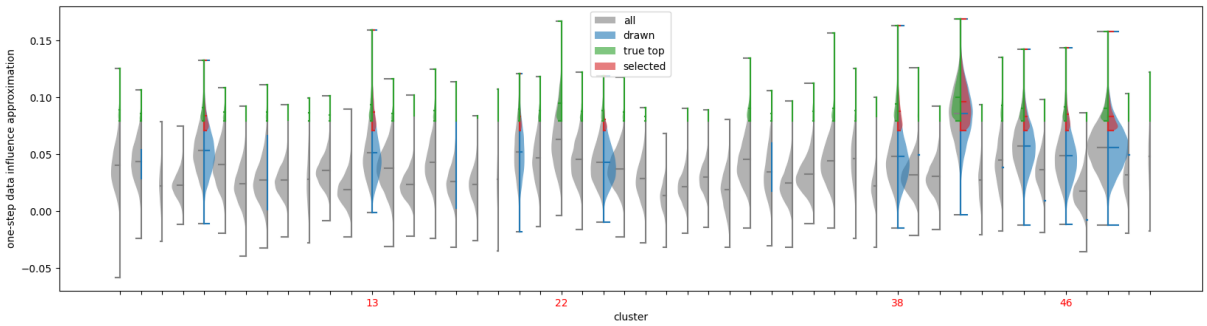
在大多数基准测试中，R_s 和 R_{inf} 随着训练步骤的增加呈下降趋势。这表明在某些训练步骤后更新聚类可能会导致更好的结果。然而，更新聚类也引入了额外的计算消耗。此外，R_{inf} 在训练的后期阶段仍然很高。因此，我们选择在实验中不更新聚类。尽管如此，根据表 1，我们仍然获得了与使用完整预算的方法相当的结果。

C 探索与利用簇间数据选择

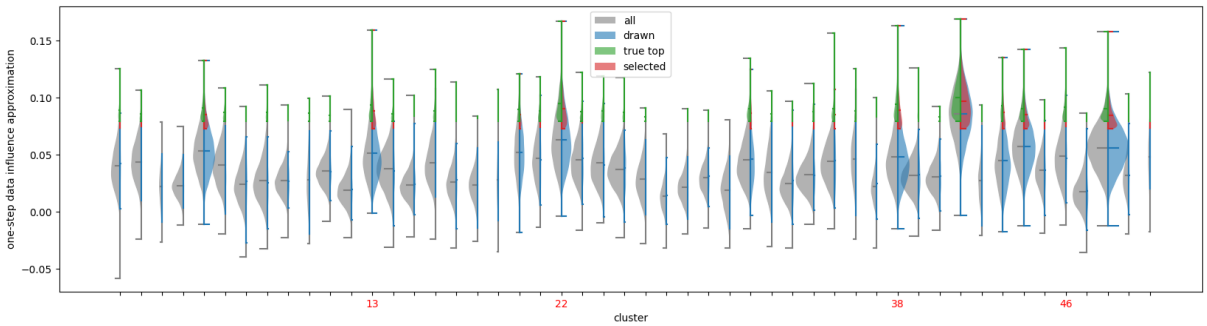
如在第 4.4.1 节中讨论的，冷启动比率 $p_{cs}\%$ 控制了我们 UCB 算法中探索与利用之间的权衡。为了进一步观察这种权衡的影响，我们绘制了每个集群中包含的总数据样本分布、在我们的 UCB 算法中提取的数据样本、具有最高数据影响近似的真实顶端数据样本，以及使用我们修改过的 UCB 算法选择的数据样本。为了简化，我们在 MMLU 基准上以 k = 50 的集群数量进行绘制。为了比较不同程度的探索和利用的效果，我们以冷启动比率 $p_{cs}\%$ = 0 %、5 % 和 50 % 进行绘制，如图 5a、5b 和 5c 所示。

当 $p_{cs}\%$ = 0 % 时，修改后的 UCB 算法不采用冷启动策略，并倾向于将大部分计算预算分配给开发。因此，图 5a 中绘制的数据样本分布更集中于几个簇，而没有覆盖许多具有高影响力数据样本的簇。给随机探索分配一个小预算，$p_{cs}\%$ = 5 %，在我们的 UCB 算法中，每个簇的评估更准确，从而更有可能找到具有更多高影响力数据样本的簇，例如图 5b 中的簇编号 22 和 46。继续增加 $p_{cs}\%$ 到 50 %，更多的预算用于探索，导致开发预算不足，修改后的 UCB 算法更可能错过高影响力的数据样本，即使它可以命中相应的簇，例如图 5c 中的簇编号 13 和 38。

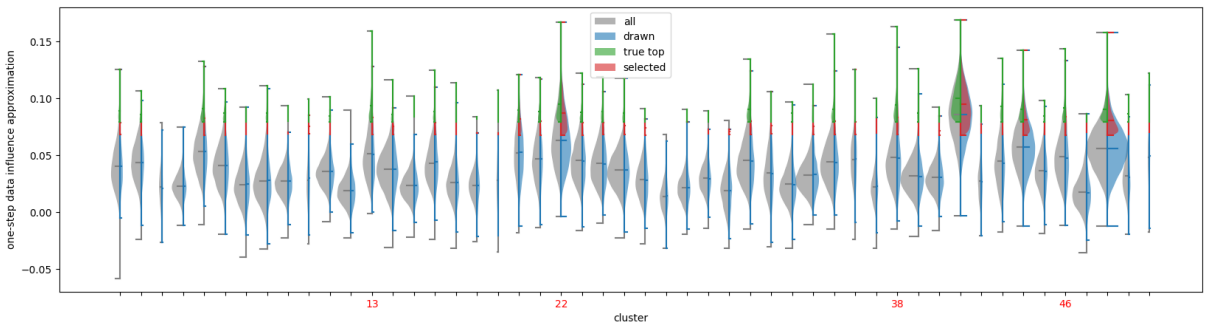
因此，跨簇数据选择的分布与我们在第 4.4.1 节中的实验结果一致，即在跨簇数据选择中，冷启动与随机探索是必要的，但在探索上花费过多预算可能会有害并导致更差的性能。



(a) 冷启动比率 = 0 %



(b) 冷启动比率 = 5 %



(c) 冷启动比率 = 50 %

Figure 5: 在 MMLU 基准测试中具有不同冷启动比例的各个集群的数据分布。每个小提琴图代表一个集群。对于每个集群，灰色部分表示该集群中包含的所有数据样本；蓝色部分表示我们在 UCB 算法中从该集群中抽取的数据样本；绿色部分是真实的具有最高影响力的数据样本在该集群中的顶端部分；红色部分是使用 ClusterUCB 从该集群中选择的数据样本。每个部分的宽度表示此部分中数据样本的数量。