

让专家各司其职：面向稀疏架构大语言模型的专家专用微调

Zihan Wang^{12*}, Deli Chen¹, Damai Dai¹, Runxin Xu¹, Zhuoshu Li¹, Y. Wu¹

¹DeepSeek AI

²Northwestern University

{zw, victorchen}@deepseek.com

摘要

参数高效微调 (PEFT) 对于在资源受限条件下定制大语言模型 (LLMs) 至关重要。尽管针对稠密架构 LLMs 的 PEFT 方法已层出不穷，但面向稀疏架构 LLMs 的 PEFT 研究仍显不足。本文研究了具有混合专家 (MoE) 架构 LLMs 的 PEFT 方法，主要贡献如下：(1) 我们探究了定制任务中激活专家的分散程度，发现特定任务的路由分布往往高度集中，而不同任务间激活专家的分布差异显著。(2) 我们提出了 **Expert-Specialized Fine-Tuning** (专家专用微调，简称 ESFT)，该方法仅微调与下游任务最相关的专家，同时冻结其他专家与模块；实验结果表明，该方法不仅提升了微调效率，其性能也匹敌甚至超越了全参数微调。(3) 我们进一步分析了 MoE 架构对专家专用微调的影响。我们发现，具有更细粒度专家的 MoE 模型在筛选与下游任务最相关的专家组合方面更具优势，从而同时提升了训练效率与效果。我们的代码已开源至 <https://github.com/deepseek-ai/ESFT>。

1 引言

随着大语言模型 (LLMs) 参数规模的持续扩大 (????)，参数高效微调 (PEFT) 方法 (?) 在将预训练 LLMs 适配至下游定制任务中的作用日益凸显。然而，现有的 PEFT 工作 (如低秩自适应 LoRA 和 P-Tuning (??)) 主要

集中于稠密架构 LLMs，针对稀疏架构 LLMs 的研究仍明显不足。

本文聚焦于探索混合专家 (MoE) LLMs (??) 内的 PEFT 技术，如 §3.1 所述。与所有任务均由相同参数处理的稠密模型不同，在 MoE 架构中，不同任务由不同的激活专家进行处理 (??)。观察表明，专家系统中的任务专用性是 MoE LLM 性能的关键 (?)。我们在 §3.2 中进一步阐明了这种专用性：由同一任务数据激活的专家分布集中，而不同任务激活的专家差异显著，这表明 MoE 模型利用专用的专家组合来处理不同任务。受此启发，我们提出了专家专用微调 (ESFT)，如 §3.3 所示。ESFT 仅微调与任务亲和力最高的专家，同时冻结其他专家与模块的参数。

ESFT 的主要优势体现在两个方面：(1) **保持专家专用性**：ESFT 避免了全参数微调中专用性下降的问题 (在全参微调中，不擅长该任务的专家也会更新参数)。§5.1 的实验结果表明，与全参数微调相比，ESFT 在下游任务中能达到相当甚至更优的性能，并在通用任务中更好地保持了性能。(2) **节省计算资源**：ESFT 仅训练选定专家的参数，如 §5.2 所示，与全参数微调相比，其存储开销最多降低 90%，训练时间最多缩短 30%。

此外，我们深入探讨了 ESFT 方法的工作机制。我们在 §6.1 中分析了专家选择过程，并展示了 ESFT 如何有效利用专用专家，仅需选择 5-15% 的专家即可在不同任务中取得

*本工作完成于 DeepSeek 实习期间。

优异性能。我们在 §6.2 中探究了 ESFT 在不同计算约束下的效率，展示了其相较于 LoRA 等其他 PEFT 方法在高效利用训练资源方面的能力。§6.3 中的研究分析了模型中共享与非共享参数对专用性能与通用性能的影响，指出在 ESFT 中应优先选择性训练非共享参数。通过 §6.4 中的消融实验，我们强调了专家相关性分数与细粒度专家划分架构的重要性。

2 相关工作

2.1 面向稠密架构 LLMs 的参数高效微调

参数高效微调 (?) 的目标是高效地为下游任务定制 LLMs，而现有研究主要集中于稠密架构 LLMs。稠密模型的 PEFT 方法通常可分为三类：(1) **添加新参数**：此类方法固定现有模型参数，仅对少量新增参数进行微调。Adapter (????) 和 Soft Prompt (????) 是该类方法的两个典型代表。(2) **选择现有参数**：此类方法仅微调现有参数中的一小部分，同时保持大部分其他参数固定。根据可训练参数空间是否连续，这些方法通常可分为结构化训练 (????) 与非结构化训练 (????)。(3) **应用低秩自适应**：LoRA (??) 是一种广泛使用的 PEFT 方法，它将原始权重矩阵分解为低秩分量。后续工作 (????) 对原始 LoRA 方法引入了诸多改进。然而，针对稀疏模型的 PEFT 研究仍然匮乏。在本文中，我们基于专家与下游任务的亲和力来选择并微调部分专家，这构成了稀疏 MoE 架构独有的选择维度。

2.2 粗粒度与细粒度 MoE LLMs

与稠密 LLMs (如 LLaMA 系列, ??) 相比, MoE LLMs (如 Mixtral 系列, ??) 能够在增加模型规模的同时节省训练与推理成本。根据专家的粒度, 现有的大型 MoE 模型通常可分为两类: 粗粒度与细粒度专家 LLMs。大多数现有的 MoE LLMs (?????) 采用粗粒度专家, 专家数量非常有限。例如, Mixtral MoE 系列 (??) 和 Grok-V1 (?) 中仅激活 8 个专家中的 2 个。因此, 单个专家必须同时从不同

领域的任务中学习复杂的模式。为解决这一问题, DeepSeek MoE (?) 引入了细粒度专家划分。在 DeepSeek-V2 (?) 中, 专家数量多达 162 个, 每次激活 8 个专家 (DeepSeek-V2-Lite 中激活 66 个专家中的 8 个)。专家的细粒度划分确保了专家之间的高度专用性。此外, 专用的专家系统使得能够筛选出与任务最相关的专家进行高效微调。

3 方法

3.1 预备知识: Transformer 中的混合专家

Transformer 中的混合专家 (MoE) 用 MoE 层替换了前馈神经网络 (FFNs)。每个 MoE 层由多个在结构上与 FFN 相同的专家组成。基于亲和力分数, Token 被分配给并交由最相关的一组专家子集进行处理, 从而确保 MoE 层的计算效率。第 l 个 MoE 层中第 t 个 Token 的输出隐藏状态 \mathbf{h}_t^l 计算如下:

$$\mathbf{h}_t^l = \sum_{i=1}^N \left(g_{i,t} \text{FFN}_i^n(\mathbf{u}_t^l) \right) + \mathbf{u}_t^l, \quad (1)$$

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \text{TopK}(\{s_{j,t} | 1 \leq j \leq N\}, K), \\ 0, & \text{otherwise,} \end{cases}, \quad (2)$$

$$s_{i,t} = \text{Softmax}_i \left(\mathbf{u}_t^{lT} \mathbf{e}_i^l \right), \quad (3)$$

其中 N 表示专家总数, $\text{FFN}_i(\cdot)$ 是第 i 个专家 FFN, $g_{i,t}$ 表示第 i 个专家的门控值, $s_{i,t}$ 表示 token 与专家之间的亲和力, $\text{TopK}(\cdot, K)$ 表示在第 t 个 token 与所有 N 个专家计算出的亲和力分数中, 包含 K 个最高分数的集合, \mathbf{e}_i^l 是第 l 层中第 i 个专家的中心点。

最近, DeepSeekMoE (?) 通过多项技术对 MoE 架构进行了改进, 包括: (1) 细粒度分割, 将每个专家分割为多个较小的专家, 并保持处理每个 token 的专家比例不变, 从而在维持相同计算成本的同时实现不同知识类型的专业化。(2) 共享专家隔离, 利用处理所有 token 的共享专家来捕捉通用知识, 减少参数

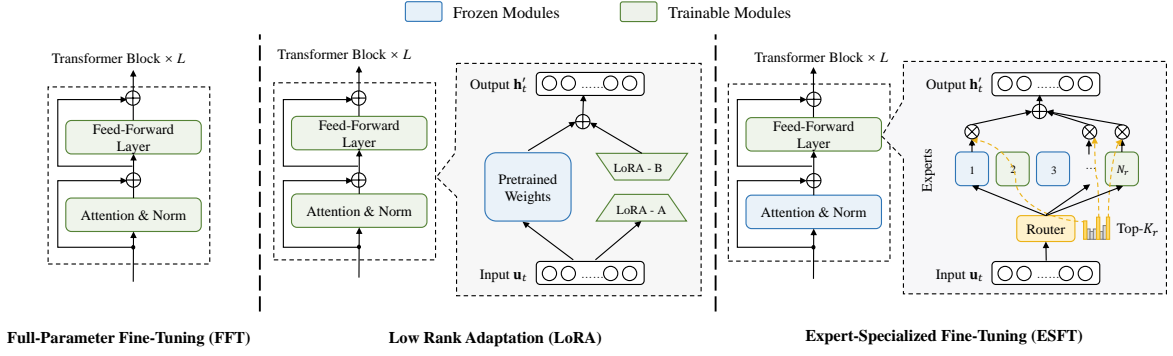


图 1: 专家专用微调 (ESFT) 与其他微调方法的对比。FFT 训练所有参数。LoRA 将预训练权重与低秩矩阵结合以降低训练成本。ESFT 仅训练混合专家 (MoE) 架构中的一部分专家, 从而优化效率与任务专用性。

冗余并提升效率。DeepSeekMoE 中 MoE 层的输出为:

$$\mathbf{h}_t^l = \sum_{i=1}^{K_s} \text{FFN}_i^s(\mathbf{u}_t^l) + \sum_{i=1}^N (g_{i,t} \text{FFN}_i^n(\mathbf{u}_t^l)) + \mathbf{u}_t^l, \quad (4)$$

$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \text{TopK}(\{s_{j,t} | 1 \leq j \leq N\}, K - K_s), \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

其中 K_s 为共享专家的数量, FFN_i^s 和 FFN_i^n 分别表示共享专家和非共享专家。每个专家被分割为 m 个专家, 与粗粒度架构相比, N 和 K 也相应扩大了 m 倍。

3.2 探测 MoE 模型中面向特定任务的专家专业化

尽管 MoE 大语言模型取得了显著成功, 但其底层机制的清晰理解仍然难以准确把握。我们开展了探针实验, 以探究非共享专家在不同任务中的利用方式。如 §4.1 所述, 这些任务涵盖数学与代码等通用领域, 以及意图识别、摘要生成、法律判决预测和翻译等专业领域。这些实验从以下两个方面揭示了 MoE 模型中的专家专业化现象:

专家路由集中于同一任务 我们研究了归一化门控值的分布, 即每个专家的所有专家-token 门控值之和除以所有专家的总和。图 2 展示了该分布, 其中专家按其归一化值从高到

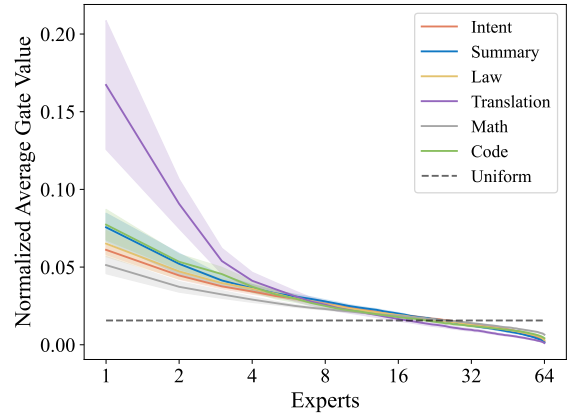


图 2: 特定任务的 Top 专家分布。阴影区域表示跨层的方差。该图显示少数专家处理了大多数门控值, 凸显了不同任务的专家专业化现象。

低排序。该图显示, 一小部分专家处理了绝大多数的门控值, 表明模型针对特定任务进行了集中的专家分配。

活跃专家在不同任务间差异显著 我们研究了专家在不同任务间的联合分布。图 3 展示了每个任务两个独立数据样本的共享 Top-6 专家热力图, 该结果已跨层平均。这反映了同一任务内部或不同任务之间所用专家的重叠程度。非对角线值接近 0, 对角线值接近 6, 表明同一任务使用相似的专家, 而不同任务使用不同的专家集合。

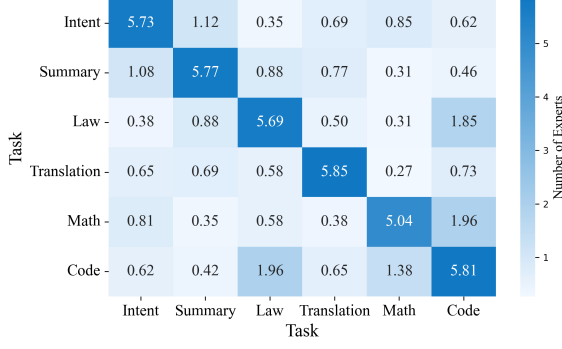


图 3: 跨任务共享的 Top-6 路由专家的平均数量。数值已按层平均, 表明同一任务使用的专家集合具有一致性, 而不同任务之间则存在明显差异。

3.3 专家专业化微调 (ESFT)

高度专业化的专家系统表明, 不同的专家可以为特定任务进行优化。受此启发, 我们提出了面向 MoE 大语言模型定制的专家专业化微调 (Expert-Specialized Fine-Tuning, ESFT) 方法。该方法针对下游任务选择性地微调最相关的专家, 以提升计算效率并保持专家的专业性。图 1 展示了我们的方法与现有方法之间的差异。下面, 我们将逐步介绍我们的方法。

数据采样 我们从训练数据 $D = \{(x_i, y_i)\}_{i=1}^N$ 中随机采样一个子集 $D_s = \{(x_i, y_i)\}_{i=1}^{N_s}$ 用于专家选择, 其中 x_i 和 y_i 分别表示输入和标签。经验表明, 由 32 个拼接样本组成的子集 (每个样本的固定长度为 $L = 4096$) 已足够稳健, 能够为特定任务筛选出最相关的专家。我们在附录 C 中详细阐述了这一结论。

专家相关性评分 我们提出了两种基于专家与样本 token 亲和度来计算专家与任务相关性的方法, 分别定义为平均门控评分 (average gate score) 和 token 选择比例 (token selection ratio)。这两种方法均可评估各专家与下游任务的相关性, 可根据特定任务的实验表现进行选择。

平均门控评分 (ESFT-Gate) 该评分计算专家 e_i 对采样数据中所有 token 的平均亲

和度。其定义如下:

$$g_i^l = \frac{1}{N_s} \sum_{j=1}^{N_s} \frac{1}{L_j} \sum_{k=1}^{L_j} g_{i,k}^l, \quad (6)$$

其中 L_j 是采样数据 D_s 中输入序列 x_j 的长度。

Token 选择比例 (ESFT-Token) 该评分计算专家 e_i 被选中的 token 所占的比例。其定义如下:

$$r_i^l = \frac{1}{N_s} \sum_{j=1}^{N_s} \frac{1}{L_j} \sum_{k=1}^{L_j} \frac{\mathbb{1}(g_{i,k}^l > 0)}{K}, \quad (7)$$

其中 $\mathbb{1}(g_{i,k}^l > 0)$ 是一个指示函数, 当门控评分 $g_{i,k}^l$ 为正时等于 1, 否则为 0。 K 表示每个 token 所选专家的数量。

专家选择与微调 对于每个 MoE 层 l , 我们根据其相关性评分选择一部分专家进行微调。我们定义一个阈值 $p \in (0, 1]$ 作为超参数, 用于控制所选子集中包含的总相关性评分的比例。对于每一层 l , 我们选择一组得分最高的专家 E_s^l , 使其累积相关性评分超过阈值 p , 满足:

$$\sum_{i \in E_s^l} R_i^l \geq p, \quad (8)$$

其中 R_i^l 是第 l 层中专家 i 的相关性评分 (即 r_i^l 或 g_i^l)。在训练和推理过程中, token 可以被分配给任意专家。然而, 每一层中只有被选中的专家 E_s^l 可以被更新; 其他专家和模块保持冻结状态。

4 实验设置

4.1 主要评估

我们在两种常见的大语言模型 (LLM) 定制场景下评估了我们的 ESFT 方法: (1) 提升模型在某个领域中的**特定能力**, 该领域模型可能已具备不错的性能; (2) 使模型适应可能**狭窄但不熟悉的专项任务**。

4.1.1 模型增强任务

我们选择了两个特定领域任务，即数学 (Math) 和代码 (Code)，以评估我们的方法如何增强模型的现有能力。这两个领域在当前的大语言模型研究中备受关注且适合进行评估，因为许多预训练模型在此表现尚可，但通过进一步训练仍有巨大的提升空间。我们通过性能提升来评估我们方法的有效性。

在数学领域，我们使用 MetaMathQA (?) 进行训练，并使用 GSM8K (?) 和 MATH (?) 进行评估。在代码领域，我们在庞大的 evol-codealpaca 数据集 (?) 的 Python 子集上训练模型，以模拟更集中的大语言模型定制场景，并在 HumanEval (?) 和 MBPP (?) 上评估其性能。

4.1.2 模型适配任务

我们选择了四个专项任务，以评估我们的方法如何促进语言模型适应不熟悉的下游任务。这些任务涵盖了多种能力，大多数模型在训练后能够出色完成，但在未经训练时则表现不佳：(1) BDCI-21 智能人机交互 NLU 挑战赛¹中的文本转 JSON 意图识别任务，要求将文本指令转换为家用电器的 JSON 格式。(2) BDCI-21 摘要挑战赛²中的文本摘要任务，用于总结客服通话记录。(3) BDCI-21 法律事件预测挑战赛³中的法律判决预测任务，其中将“案情描述”和“判决结果”重新用作法律判决预测任务。(4) ChrEn 数据集 (?) 中的低资源翻译任务，将少数民族语言切罗基语 (Cherokee) 翻译为英语。任务示例见附录 A。

为衡量模型性能，对于文本转 JSON 任务，我们计算模型输出与参考答案的精确匹配率；对于其他任务，我们使用 GPT-4 根据参考答案对模型输出进行 0 到 10 分的评分⁴。所有评估均采用少样本 (few-shot) 示例。

¹<https://www.datafountain.cn/competitions/511>

²<https://www.datafountain.cn/competitions/536>

³<https://www.datafountain.cn/competitions/540>

⁴我们使用的具体版本为 gpt-4-1106-preview。评估指令见附录 G。

4.2 通用能力评估

我们选取了广泛的基准测试，以评估模型在新任务训练后其通用能力得以保留的程度。这些基准测试包括 MMLU (?)、TriviaQA (?)、HellaSwag (?)、ARC-Challenge (?)、IFEval (?)、CEval (?) 和 CLUEWSC (?)，涵盖了自然语言理解、问答、指令遵循和常识推理等多个领域的全面模型能力评估。

4.3 骨干模型与训练设置

我们在所有实验中均采用 DeepSeek-V2-Lite (?) 的骨干架构。该模型在每个 Transformer 层中均包含一组细粒度的 66 个专家。这使得它在开展本研究时独特地契合我们的方法，因为我们的方法能够从专家专业化中获益。我们在一个精心构建的对齐数据集上训练该模型，该数据集排除了数学和代码数据，并将训练得到的检查点作为后续实验的原始模型 (vanilla model)。该对齐阶段能够在激活模型跨领域能力的同时，将数学/代码能力保持在基础水平，从而更好地验证我们的方法在这两个领域的性能提升。

我们采用两个基线方法：全参数微调 (FFT) 和低秩自适应 (LoRA, ?)。对于 LoRA，我们在训练时为除词元嵌入 (token embeddings) 和语言建模头 (language modeling head) 之外的所有参数添加低秩矩阵。对于所有方法，我们保持任务特定数据与对齐数据 1:1 的比例，我们发现这对于 FFT 和 LoRA 保留对齐阶段获得的通用能力非常有效。然而，对于我们的 ESFT 方法，不采用这种数据混合策略甚至能更好地保持通用能力。我们在附录 F 中对此进行了详细说明。所有实验均在 HFAI 集群⁵上进行，该集群包含 2 个节点，每个节点配备 8 块 Nvidia A100 PCIe GPU。

在超参数设置方面，所有方法在训练时均使用 32 的批次大小和 4096 的序列长度。对于每个任务，我们将最大训练步数设置为 500，并每 100 步评估一次模型。基于在 {1e-5,

⁵<https://doc.hfai.high-flyer.cn/index.html>

	数学能力		代码能力		专项任务				平均
	MATH	GSM8K	Humaneval	MBPP	意图识别	摘要	法律	翻译	
原始模型	19.6	55.9	<u>42.1</u>	<u>44.6</u>	16.8	58.6	17.1	14.5	33.6
FFT	23.4	66.4	<u>42.1</u>	42.2	78.8	69.4	<u>47.0</u>	38.4	51.0
LoRA	20.6	58.9	39.6	44.8	67.8	64.7	39.7	23.1	44.9
ESFT-Token (本文)	22.6	66.0	41.5	42.6	75.6	65.4	45.7	<u>36.2</u>	49.4
ESFT-Gate (本文)	<u>23.2</u>	64.9	43.3	41.8	78.6	<u>65.8</u>	49.1	35.2	<u>50.2</u>

表 1: 不同方法与任务的主要性能对比。最佳或接近最佳的结果以**粗体**显示，次佳结果以下划线标出。我们的方法 ESFT 在多样化任务中提供了强劲的性能平衡，与 FFT 相当并超越了 LoRA，尤其在专项任务领域表现突出。

	CLUEWSC	TriviaQA	IFEval	MMLU	CEval	HellaSwag	ARC	平均
原始模型	81.5	67.7	42.5	57.5	59.9	74.0	53.7	62.4
FFT	80.9 ± 1.1	65.9 ± 0.7	34.2 ± 4.1	55.5 ± 1.0	58.8 ± 0.9	67.9 ± 3.8	48.4 ± 2.4	58.8 ± 1.3
LoRA	74.3 ± 7.7	63.4 ± 5.4	38.7 ± 2.5	55.5 ± 1.2	57.0 ± 1.5	72.8 ± 1.9	51.8 ± 2.3	59.1 ± 2.5
ESFT-Token	80.9 ± 0.9	66.7 ± 1.8	40.7 ± 1.3	57.1 ± 0.5	59.6 ± 0.8	72.3 ± 3.6	52.9 ± 1.5	61.5 ± 1.1
ESFT-Gate	81.4 ± 1.1	66.5 ± 2.3	40.2 ± 1.5	57.0 ± 0.4	59.5 ± 0.8	68.2 ± 9.9	51.5 ± 3.1	60.6 ± 2.3

表 2: 不同方法与任务在通用能力上的性能对比。各任务的性能为所有训练实验的平均值，后附跨任务的标准差。最佳或接近最佳的结果以**粗体**显示。我们的方法 ESFT 在所有任务中均稳定取得了良好性能。

3e-5, 1e-4, 3e-4} 范围内的超参数搜索，我们将 FFT、LoRA 和 ESFT 的学习率分别设置为 3e-5、1e-4 和 1e-5。遵循 ?，我们将 LoRA 的秩 (rank) 设置为 8，缩放因子 (scaling) 设置为 2。阈值 p 分别设置为 ESFT-Gate 的 0.1 和 ESFT-Token 的 0.2。§6.2 展示了我们如何确定 ESFT 的阈值。

5 实验结果

5.1 基准性能结果

表 1 与表 2 的结果揭示了以下几点结论。与基础模型相比，所有方法均能提升模型在定制化任务中的性能，但可能会导致模型在通用任务上的性能下降。总体而言，模型在适配任务中的性能提升幅度高于在增强任务中的提升幅度。

在定制化能力评估方面，ESFT 显著优

于 LoRA，且与 FFT 具有竞争力。如表 1 所示，ESFT-Token 和 ESFT-Gate 在 Math 等模型增强任务中取得了接近最优的结果，且 ESFT-Gate 在 Humaneval 任务中达到了最佳性能。ESFT 在模型适配任务中同样表现出色，其中 ESFT-Gate 在 4 项任务中的 3 项取得了接近最优的性能。值得注意的是，ESFT-Gate 的平均分为 50.2，与 FFT 的 51.0 相比具有竞争力，略优于 ESFT-Token 的 49.4，并显著超越了 LoRA 的 44.9。这表明，寻找与任务相关的专家能够高效地适配模型，从而实现高效的定制化。

在通用能力评估方面，ESFT 通过表现出更小的性能下降，始终优于 FFT 和 LoRA。如表 2 所示，ESFT-token 的表现优于 ESFT-gate，平均分分别为 61.5 和 60.6。结果表明，在 TriviaQA 和 IFEval 等任务中，模型表现

出良好的性能保持，超越了 FFT 的 58.8 和 LoRA 的 59.1。这两种方法在性能保持方面均优于 LoRA 和 FFT，凸显了它们在维持通用任务性能方面的有效性⁶。§6.3 中的分析表明，FFT 和 LoRA 在通用任务上的此类性能下降可能是由于训练共享参数所致。

5.2 计算效率结果

图 6 的结果表明，ESFT 在训练时间和存储空间需求方面具有多项优势：

训练时间 ESFT-Token 和 ESFT-Gate 的平均训练时间分别为 19.8 分钟和 20.9 分钟。FFT 方法耗时显著更长，达到 28.5 分钟。尽管 LoRA 实现了更短的训练时间（16.5 分钟），但我们的方法与其相对接近。

存储空间 ESFT-Token 和 ESFT-Gate 训练参数的平均存储空间分别为 2.57 GB 和 3.20 GB，而 FFT 则需要高达 28.6 GB 的存储空间。尽管 LoRA 所需的存储空间更少，ESFT 在下游任务性能上显著优于 LoRA。

综上所述，ESFT 在训练时间和存储空间方面均表现出优异的性能，显著优于 FFT。此外，如表 3 所示，与 FFT 相比，ESFT 所需的可训练参数少得多，从而降低了 GPU 内存占用。这些优势表明，ESFT 在语言模型定制化和适配方面兼具高效性与有效性。

6 分析

在本节中，我们在 §6.1 中探讨了 ESFT 的专家选择过程，并在 §6.2 中展示了 ESFT 和 LoRA 在不同计算约束下的性能表现。我们在 §6.3 中分析了训练共享与非共享参数的影响，并在 §6.4 中进行了消融实验，以验证我们提出的专家相关性分数以及细粒度专家模型结构的重要性。

⁶我们在附录 H 中进一步研究了在专用任务上训练的模型在 Math 和 Code 任务上的性能。FFT 和 LoRA 表现出更为严重的性能下降，而 ESFT 仅出现微小的性能降幅。

6.1 ESFT 有效利用专业化专家

我们分析了 ESFT 在不同任务和层中训练的专家数量，以了解其专家选择过程。结果如图 4 所示。

从结果中，我们得出以下几点观察：(1) 跨层每个任务使用的专家平均数量在 66 个中占 2 到 15 个，表明与 FFT 相比，ESFT 的可训练参数可减少 75%-95%。(2) ESFT-Token 通常使用更少的专家，同时能更好地保持通用性能，在 Math、Intent 和 Law 等任务上与 ESFT-Gate 表现相当。(3) 专家数量因任务而异，Math 和 Translation 等更专业化的任务使用的专家更少；我们的方法在这些任务上的性能超越 LoRA 的幅度最大，表明该方法尤其适合更专业化的任务。(4) 对于大多数任务，中间层选择的专家较少，表明专家分布在这些层中更为集中。

6.2 ESFT 高效利用训练资源

ESFT 和 LoRA 均包含一个训练效率超参数 (ESFT 为 p , LoRA 为 rank)。增大该值会增加计算资源消耗，并可能提升性能。为了探究 ESFT 和 LoRA 在不同效率设置下的表现，我们在数学任务上评估了基准性能。我们将 LoRA 的 rank 设置为 ≤ 512 ，因为更高的值会导致可训练参数多于 FFT。图 6 展示了在不同训练效率设置下的专项能力与通用能力。

从结果中我们可以得出以下结论：(1) 所有三种方法在训练效率与性能之间均存在权衡。在达到某一临界点之前，增加训练参数 (ESFT 的 p 和 LoRA 的 rank) 能够提升性能。(2) ESFT-Token 和 ESFT-Gate 在任何设置下均优于 LoRA，展现出更高的专项能力和更稳定的通用能力。(3) ESFT-Token 在 $p=0.5$ 时专项和通用能力均达到峰值，而 ESFT-Gate 在专项能力上于 $p=0.3$ 达到峰值，在通用能力上于 $p=0.1$ 达到峰值。(4) ESFT-Token 和 ESFT-Gate 的性能分别在 $p=0.2$ 和 $p=0.1$ 时趋于饱和，这表明大多数专家的选择可能与任

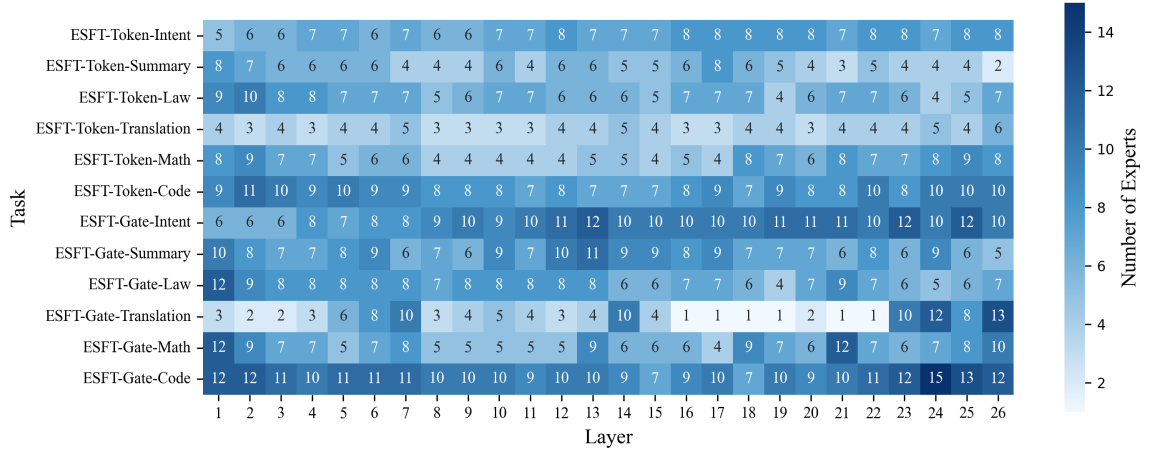


图 4: ESFT 在不同层和任务中训练的专家数量。较早计算的层编号较小。大多数任务和层仅训练 5-15% 的专家，证明了 ESFT 在选择任务相关专家方面的有效性。

非共享专家	共享专家	非专家参数	可训练参数	专业能力	通用能力	平均
ALL	✓	✓	15.7B	51.0	58.8	54.9
Relevant	✓	×	1.85B	49.8	60.7	55.3
Relevant	×	×	1.4B	49.4	61.5	55.4
×	✓	×	450M	47.4	61.2	54.3
×	✓	✓	1.3B	49.0	60.0	54.5
Relevant	✓	✓	2.7B	50.8	60.3	55.6
×	×	×	-	33.8	62.4	48.1

表 3: 基于是否训练共享或非共享参数的不同模型配置对比。结果包含可训练参数数量以及专业能力和通用能力的性能表现。除不训练设置外，最优或接近最优的结果以**粗体**显示。

务性能的相关性较低。我们在附录 E 中对此进行了深入探讨。

6.3 选择性训练非共享参数是 ESFT 的关键

在我们提出的 ESFT 方法中，我们仅微调非共享专家的一个子集。本节详细讨论了该方法中可能也会训练共享参数的几种变体。变量基于以下因素：

- 是否训练 **所有**非共享专家，还是仅训练其中 **与任务相关**的子集（我们使用 Token 选择比例并设置 $p=0.2$ ）。
- 是否训练共享专家。
- 是否训练其他参数，包括门控、注意力层和嵌入层。

结果如表 3 所示。我们报告了所有任务上的平均可训练参数数量、专项能力与通用能力的性能及其平均值。所有基准的详细数据见附录 D。从结果中，我们可以得出以下几点结论：

专项性能随可训练参数的增加而提升。可训练参数数量从 450M 到 15.7B 的排序与专项能力从 47.4 到 51.0 的排序高度一致。这表明增加可训练参数能有效提升专项性能。

通用性能随可训练共享参数的增加而下降。无论是否训练相关的非共享专家，当我们训练共享专家和/或非专家参数时，通用性能分别从 61.5 降至 60.3，或从 62.4 降至 60.0。当训练完整的非共享专家集合时，通用性能进一步从 60.3 降至 58.8。这表明与训练非共享参数相比，训练共享参数更容易导致在下游任

	数学能力		代码能力		专项任务				平均
	MATH	GSM8K	Humaneval	MBPP	意图	摘要	法律	翻译	
ESFT-Token	22.6	66.0	41.5	42.6	75.6	65.4	45.7	36.2	49.4
Δ of rand	-1.0	-3.7	-2.5	0.2	-2.6	-1.7	1.3	-13.5	-2.8
ESFT-Gate	23.2	64.9	43.3	41.8	78.6	65.8	49.1	35.2	50.2
Δ of rand	-1.7	-3.2	-4.3	1.6	-5.0	0.3	-2.9	-20.4	-4.4

表 4: 原始专家与随机专家的性能对比。将高亲和力专家替换为随机专家会显著损害模型在不同任务上的性能。

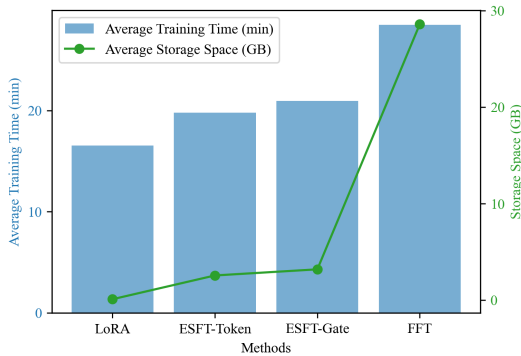


图 5: 计算效率结果。蓝色柱状图表示训练时间, 绿色折线表示存储空间。ESFT 在训练时间和存储空间方面均表现出较高的效率。

务上过拟合以及在通用任务上发生遗忘。

训练与任务相关的非共享专家具有最高优先级。 训练相关专家至少能达到 55.3 的性能, 而其他设置即使使用高达 15.7B 的参数, 最高也只能达到 54.9。因此, 微调这些专家对于模型定制化具有极高的优先级。

基于这些结论, 我们提出两种主要的训练策略:

- 1. 优先专项能力:** 训练所有共享参数和与任务相关的非共享专家, 以最大化提升专项性能。
- 2. 平衡专项能力、通用能力与计算效率:** 仅训练与任务相关的非共享专家, 在最小化参数成本的同时最大化保持通用能力。

6.4 ESFT 中关键模块的分析

在本节中, 我们分析并证明了本方法的有效性主要归功于两个模块: (1) 我们提出的专家相关性评分函数, 以及 (2) MoE 模型架构的细粒度专家划分。

专家相关性评分函数 在本工作中, 我们提出平均门控评分 (Average Gate Score) 和 Token 选择率 (Token Selection Ratio) 作为专家相关性评分函数, 以筛选适用于不同任务的相关专家。为验证其有效性, 我们在保持每层激活专家数量不变的前提下, 将这些函数筛选出的专家替换为随机专家。表 4 的结果表明, 将相关专家替换为随机专家会显著降低任务性能, 从而证明了我们所提相关性评分函数的有效性。

MoE 模型的细粒度专家划分 我们采用细粒度划分的 DeepSeek-V2 模型作为主干网络。为验证该细粒度划分的有效性, 我们使用贪婪搜索 (详见附录 B) 对专家进行分组, 以模拟粗粒度划分。同一组内的专家共享平均亲和力评分。我们通过为每个 token 固定选择 1/8 的专家来保持计算成本不变。图 7 中数学领域的实验结果表明, 随着分组规模的增大, 本方法的性能下降幅度比 FFT 更为严重, 同时训练成本 (即可训练专家数量) 上升。这些发现表明, 本方法乃至有效的 LLM 定制, 都高度依赖于具有更专业化专家的细粒度划分 LLM 架构。

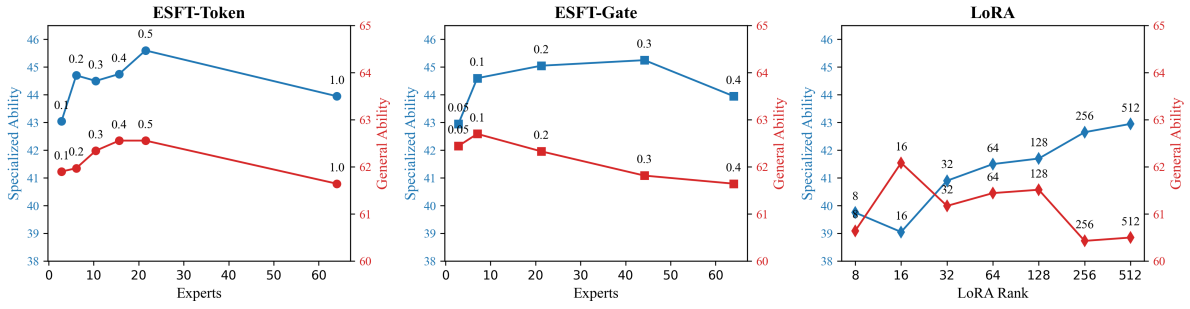


图 6: 在 Math 任务上, 三种方法在不同训练效率设置下的对比。横轴表示 ESFT 每层的平均可训练专家数量以及 LoRA 的 rank, 用于指示训练参数的比例。纵轴表示专业能力和通用能力。曲线上的标记点表示 p 或 rank 的值。在专业能力和通用能力方面, ESFT 始终优于 LoRA。

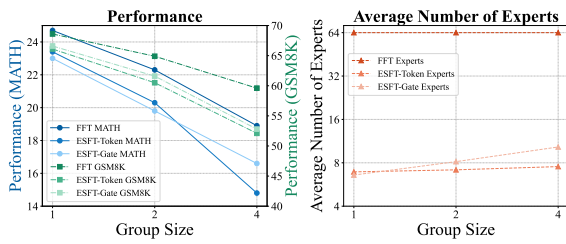


图 7: 分组专家实验的结果。随着专家粒度变粗, ESFT 的性能下降比 FFT 更为严重。

拟方法, 即将多组专家绑定在一起, 以比较粗粒度与细粒度 MoE 方法。

7 结论

在本工作中, 我们研究了针对具有混合专家 (MoE) 架构的稀疏大语言模型的参数高效微调方法。我们首先观察到, 不同领域的任务由不同的专家组合处理。随后, 我们提出使用两项指标——平均门控评分和 Token 选择率——为下游任务筛选最相关的专家。实验结果表明, 我们的方法在显著降低训练成本的同时, 其性能可媲美甚至超越全参数微调。进一步的分析证实, 本方法增强了 MoE 架构内部专家系统的专业化程度。

局限性

首先, 受限与其他细粒度 MoE 模型的可获取性, 我们的方法仅在 DeepSeek-V2-Lite MoE 模型上进行了测试。将该模型得出的结论推广至其他场景时, 仍需进一步验证。此外, 由于缺乏在参数层面和结构上对齐且具有不同专家粒度的 MoE 模型, 我们采用了一种模

附录

A 专用任务示例

表 5 以提示词和对应参考回答的形式，展示了各专用任务的示例，包括意图识别、文本摘要、法律判决预测和低资源翻译。

B 专家分组策略

为了将专家分组并模拟粗粒度的混合专家 (Mixture-of-Experts) Transformer 模型，我们计算专家相似度，并使用贪心搜索算法通过最大化组内相似度对专家进行分组。

我们从对齐数据集中采样数据（包含 32 个样本，每个样本的序列长度为 4096），以计算专家之间的相似度。我们将所有专家对的共现矩阵初始化为零矩阵。对于在某个 token 的 Top-6 专家选择中同时出现的每一对专家，我们在矩阵中将它们的得分加 1。遍历完数据集后，我们利用矩阵中第 i 行和第 j 行向量之间的余弦相似度，计算专家 i 与专家 j 之间的相似度。

为了通过贪心搜索获得专家分组策略，我们从每层 66 个专家中的 64 个非共享专家中，计算所有可能的 K 专家组（其中 K 为组大小，取 2 或 4）的平均组内相似度（即组内所有专家两两之间的平均相似度）。随后，我们选择得分最高的 K 专家组。对于剩余未选中的专家，我们重复此过程，直到所有专家都被选中并完成分组。

C 专家亲和力样本规模分析

为了评估识别任务最相关专家所需的数据量，我们为六个任务中的每一个从训练集中独立采样两组数据，并计算这两组数据之间共享的 Top-6 专家。结果如图 8 所示。当样本规模达到 2^{17} （即 32 个序列长度为 4096 的样本）时，所有任务在两个样本之间均表现出较高数量的共享专家。这表明该样本规模已足够大，能够选出对任务最相关的专家。

非共享	共享	非专家	CLUEWSC	TriviaQA	IFEval	MMLU	CEval	HellaSwag	ARC	平均
全部	✓	✓	80.9 ± 2.2	65.9 ± 1.5	34.2 ± 8.1	55.5 ± 1.9	58.8 ± 1.7	67.9 ± 7.4	48.4 ± 4.7	58.8 ± 2.5
相关	✓	×	80.9 ± 2.1	66.1 ± 4.4	42.4 ± 3.0	56.8 ± 1.0	58.9 ± 1.6	67.8 ± 20.4	52.1 ± 5.7	60.7 ± 4.4
相关	×	×	80.9 ± 1.8	66.7 ± 3.5	40.7 ± 2.6	57.1 ± 1.0	59.6 ± 1.5	72.3 ± 7.0	52.9 ± 3.0	61.5 ± 2.3
×	✓	×	81.1 ± 3.4	66.7 ± 4.2	41.2 ± 1.6	56.9 ± 1.2	58.9 ± 1.6	71.3 ± 14.1	52.6 ± 5.6	61.2 ± 3.3
×	✓	✓	79.5 ± 4.4	65.8 ± 5.0	41.4 ± 3.2	56.2 ± 1.6	58.6 ± 1.7	67.5 ± 20.7	51.2 ± 4.1	60.0 ± 4.4
相关	✓	✓	80.4 ± 4.1	66.3 ± 4.1	41.1 ± 5.0	56.7 ± 1.2	59.0 ± 1.9	67.5 ± 20.3	51.5 ± 4.6	60.3 ± 4.6
×	×	×	81.5	67.7	42.5	57.5	59.9	74.0	53.7	62.4

表 6: 基于是否训练共享或非共享参数, 不同方法在通用任务上的性能表现。各任务的性能为所有训练实验的平均值, 后附跨任务的标准差。最佳或接近最佳的结果以**粗体**显示。

非共享	共享	非专家	数学能力		代码能力		专业化任务				平均
			MATH	GSM8K	Humaneval	MBPP	意图识别	摘要	法律	翻译	
全部	✓	✓	23.4	66.4	42.1	42.2	78.8	69.4	47.0	38.4	51.0
相关	✓	×	23.8	65.7	40.2	43.8	80.4	67.3	42.4	35.1	49.8
相关	×	×	22.6	66.0	41.5	42.6	75.6	65.4	45.7	36.2	49.4
×	✓	×	22.7	64.5	37.2	44.0	73.6	68.3	42.7	26.0	47.4
×	✓	✓	23.4	66.6	41.5	44.4	81.0	66.7	39.0	29.5	49.0
相关	✓	✓	24.8	66.0	42.1	43.2	82.2	69.5	46.4	32.2	50.8
×	×	×	19.6	55.9	42.1	44.6	16.8	58.6	17.1	14.5	33.6

表 7: 基于是否训练共享或非共享参数, 不同方法在专业化任务上的性能表现。最佳或接近最佳的结果以**粗体**显示。

F 混合对齐数据对训练的影响

在训练过程中, 我们为所有方法采用下游任务数据与对齐数据 1:1 的混合比例, 以更好地保持通用任务性能。该人工设定的比例保持恒定, 以避免为每个任务微调比例所带来的显著额外成本。

在本节中, 我们展示了不同方法和任务之间的性能对比, 以揭示训练过程中混合对齐数据的影响。表 9 展示了在下游专业任务上的性能, 表 10 展示了在通用任务上的性能。

结果表明, FFT 和 LoRA 受益于对齐数据的加入, 在提升通用任务性能的同时, 仅使下游任务性能略有下降。相反, 我们的 ESFT 方法并未表现出同样的优势。具体而言, 混合对齐数据并未带来通用任务或下游任务的性能提升。这一发现表明, ESFT 本身即具备适

应下游任务的能力, 即使不添加对齐数据, 也不会导致通用任务性能的显著下降。这凸显了 ESFT 在多样化任务设置中的鲁棒性与适应性。

G 专业任务的评估说明

表 11 展示了评估专业任务 (包括文本摘要、法律判决预测和低资源翻译) 的详细标准。每个任务都包含将预测答案与参考答案进行对比评估的具体说明, 重点关注内容准确性、完整性、相关性和一致性等方面。

H 将 Math 与 Code 作为通用任务进行评估

我们研究了在适配任务 (即 Intent、Summary、Law、Translation) 上训练的模型在

	Math Ability		Code Ability		Average
	MATH	GSM8K	HumanEval	MBPP	
Vanilla Model	19.6	55.9	42.1	44.6	40.5
FFT	15.1 ± 0.3	40.3 ± 5.3	30.2 ± 4.4	40.6 ± 3.9	31.5 ± 2.5
LoRA	11.8 ± 0.6	36.1 ± 4.4	27.9 ± 2.3	36.6 ± 2.6	28.1 ± 2.0
ESFT-Token	19.4 ± 0.8	55.2 ± 0.7	39.5 ± 1.0	44.8 ± 0.8	39.7 ± 0.4
ESFT-Gate	19.5 ± 0.3	55.1 ± 1.3	39.3 ± 1.3	45.3 ± 0.6	39.8 ± 0.6

表 8: 在专项任务上训练的不同方法的数学与代码能力性能对比。最佳或接近最佳的结果以**粗体**显示。与 FFT 和 LoRA 相比, ESFT 能显著更好地保持性能。

	Math Ability		Code Ability		Specialized Tasks				Average
	MATH	GSM8K	HumanEval	MBPP	Intent	Service	Law	Translation	
FFT	26.1	70.4	51.2	42.6	78.8	72.8	45.6	34.4	52.7
+ mix data	-2.7	-4.0	-9.1	-0.4	0.0	-3.4	1.4	4.0	-1.7
LoRA	21.8	57.8	42.1	42.6	78.2	66.4	46.0	21.8	47.1
+ mix data	-1.2	1.1	-2.5	2.2	-10.4	-1.7	-6.3	1.3	-2.2
ESFT-Token	25.2	64.8	42.1	43.8	78.0	67.4	47.2	31.9	50.0
+ mix data	-2.6	1.2	-0.6	-1.2	-2.4	-2.0	-1.5	4.3	-0.6
ESFT-Gate	24.1	64.9	42.1	44.6	77.2	68.4	43.6	32.8	49.7
+ mix data	-0.9	0.0	0.0	-2.8	1.4	-2.6	0.9	2.4	0.5

表 9: 不同方法在下游任务上的性能对比 (包含与不包含对齐阶段数据混合的情况)。结果表明, 混合对齐数据会导致大多数方法的性能出现轻微下降。

Math 和 Code 任务上的表现, 因为如果未在这些领域进行专门训练, 它们能够反映模型的通用能力。我们报告了仅在下游任务数据上进行训练的设置下的结果。表 8 中的结果表明, FFT 和 LoRA 会导致 Math 和 Code 领域的性能显著下降, 平均性能下降分别为 9.0 和 12.4。值得注意的是, 与 FFT 和 LoRA 相比, 我们的 ESFT 方法能更好地保持性能, 平均性能下降不到 1。

	CLUEWSC	TriviaQA	IFEval	MMLU	CEval	HellaSwag	ARC	Average
Vanilla Model	81.5	67.7	42.5	57.5	59.9	74.0	53.7	62.4
FFT	76.8 ± 1.7	62.4 ± 10	28.4 ± 5.1	55.5 ± 1.1	58.4 ± 0.4	74.6 ± 3.2	53.6 ± 3.1	58.5 ± 2.5
+ mix data	4.1	3.5	5.8	0.0	0.4	-6.7	-5.2	0.3
LoRA	60.2 ± 27	61.2 ± 4.0	33.4 ± 6.1	52.3 ± 3.3	55.3 ± 2.3	71.5 ± 2.5	50.7 ± 2.2	55.0 ± 4.6
+ mix data	14.1	2.2	5.3	3.2	1.7	1.3	1.1	4.1
ESFT-Token	80.0 ± 2.5	67.5 ± 0.3	41.9 ± 0.8	57.3 ± 0.2	60.2 ± 0.5	74.5 ± 0.7	54.9 ± 0.7	62.3 ± 0.5
+ mix data	0.9	-0.8	-1.2	-0.2	-0.6	-2.2	-2.0	-0.8
ESFT-Gate	80.2 ± 1.6	67.6 ± 0.3	40.8 ± 2.4	57.3 ± 0.3	59.9 ± 0.4	74.3 ± 0.9	55.1 ± 0.9	62.2 ± 0.5
+ mix data	1.2	-1.1	-0.6	-0.3	-0.4	-6.1	-3.6	-1.6

表 10: 不同方法在通用任务上的性能对比 (包含与不包含对齐数据混合的情况)。结果表明, 混合对齐数据能提升 FFT 和 LoRA 在通用任务上的表现, 但对我们的 ESFT 方法无效。这展示了 ESFT 能够直接适应下游任务, 且在通用任务上仅产生极小的性能损失。

Task	Evaluation Instruction
Summary	<p>请你进行以下电话总结内容的评分。请依据以下标准综合考量，以确定预测答案与标准答案之间的一致性程度。满分为 10 分，根据预测答案的准确性、完整性和相关性来逐项扣分。请先给每一项打分并给出总分，再给出打分理由。总分为 10 分减去每一项扣除分数之和，最低可扣到 0 分。请以“内容准确性扣 x 分，详细程度/完整性扣 x 分，...，总分是：x 分”为开头。</p> <p>1. 内容准确性：- 预测答案是否准确反映了客户问题或投诉的核心要点。- 是否有任何关键信息被错误陈述或误解。2. 详细程度/完整性：- 预测答案中包含的细节是否充分，能否覆盖标准答案中所有重要点。- 对于任何遗漏的关键信息，应相应减分。3. 内容冗余度：- 预测答案是否简洁明了，和标准答案风格一致，不存在冗余信息。- 如果预测答案过长或与标准答案风格不一致，需相应减分。4. 行动指令正确性：- 预测答案对后续处理的建议或请求是否与标准答案相符。- 如果处理建议发生改变或丢失，需相应减分。预测答案：{prediction} 参考答案：{ground_truth}</p>
Law	<p>请你进行以下法案判决预测内容的评分。请依据以下标准综合考量，以确定预测答案与标准答案之间的一致性程度。满分为 10 分，根据预测答案的准确性、完整性和相关性来逐项扣分。请先给每一项打分并给出总分，再给出打分理由。总分为 10 分减去每一项扣除分数之和，最低可扣到 0 分。请以“相关性扣 x 分，完整性扣 x 分，...，总分是：x 分”为开头。1. 相关性：预测答案与标准答案的相关程度是最重要的评分标准。如果预测的判决情况与标准答案完全一致，即所有事实和结果都被精确复制或以不同但等效的方式表述，则应给予高分。若只有部分一致或存在偏差，则根据一致的程度适当扣分。如果没有预测判决内容，扣 10 分。2. 完整性：评估预测答案是否涵盖了所有标准答案中提到的关键点，包括但不限于当事人、具体金额、责任判定、费用承担等。如果遗漏重要信息，则应相应扣分。3. 准确性：检查预测答案中提及的细节、数字、日期和法律依据是否与标准答案保持一致。任何错误信息均需扣分，并且严重错误应该导致更多的扣分。4. 客观性与专业性：预测答案应客观反映法案内容并使用恰当的法律术语。主观臆断或非专业表达需酌情扣分。预测答案：{prediction} 参考答案：{ground_truth}</p>
Translation	<p>你是机器翻译领域的专家。请根据以下标准，以 10 分为满分对预测答案相对于标准答案进行评分：内容准确性：预测答案是否准确反映了参考答案的关键要点？详细程度/完整性：预测答案是否涵盖了标准答案中的所有重要要点？内容冗余度：预测答案是否简洁明了，且与标准答案的风格保持一致？请按照以下格式回复：“内容准确性 x 分，详细程度/完整性 x 分，...，总分：x 分”。总分为所有分项得分的平均值。无需给出评分理由。预测答案：{prediction} 参考答案：{ground_truth}</p>

表 11: 模型性能评估的任务指令。占位符 {prediction} 和 {ground_truth} 分别代表模型预测结果和参考答案。

