梯度下降能模拟提示吗？

Eric Zhang **Leshem Choshen** **Jacob Andreas**
MIT CSAIL
{zeric,leshem,jda}@mit.edu

Abstract

将新信息纳入语言模型（LM）主要有两种方式：更改其提示或更改其参数，例如通过微调。参数更新不会对模型变更产生长期存储成本。然而，对于许多模型更新而言，提示方法要有效得多：被提示的模型可以单个示例中稳健地泛化，并进行逻辑推理，这些在标准微调下是不会发生的。是否可以通过修改使得微调做模仿提示效果？本文介绍了一种元训练 LM 的方法，使梯度更新模仿对新信息进行条件处理的效果。我们的方法使用基于梯度的元学习工具，但将 LM 的自己的提示预测作为目标，消除了对真实标签的需求。随后的梯度下降训练恢复了部分（有时是全部）被提示模型的表现——在“反转诅咒”任务上表现出改进，并在一个梯度更新后回答文本段落的问题。这些结果表明，通过适当的初始化，梯度下降可以出人意料地具有表现力。我们的研究结果为长上下文建模开辟了新的途径，并提供了对基于梯度学习泛化能力的见解。

1 介绍

高效且可靠地将新信息整合到语言模型（LMs）中的能力对于许多实际应用场景至关重要。现有的方法通常分为两类：基于上下文的方法（例如提示）和基于参数的方法（例如全微调或模型编辑）。虽然提示是灵活有效的，但它会增加推理时间和内存成本，并受制于 LMs 的上下文窗口大小。编辑方法尽管很有前景，但仍处于实验阶段，其泛化能力尚未得到充分理解。普通的微调在应用于个别新信息时往往无法产生连贯的模型更新。

我们能否通过构建一个从微调中学习的模型来结合这些不同更新方法的优点，就像它从上下文中学习一样？这将需要模型以权重的静态更新来表达由上下文引起的行为修改，这个更新由单一梯度决定。虽然有很多研究探讨了这个问题的相反方面（上下文中的学习方法能否被视为模拟梯度下降；Akyürek et al., 2022, von Oswald et al., 2022 等），但标准学习算法是否能模拟条件导致的更复杂预测这一问题则远未被理解。

我们使用受 MAML[Finn et al., 2017] 启发的元学习目标来探讨这个问题，寻找一种模型参数化方式，使得通过梯度步长引入新信息的方式与将其添加到上下文中的方式相同。在实验中，我们展示了以这种方式进行元训练的模型在各种任务上都有所改进，包括“逆转诅咒”知识编辑 [Berglund et al., 2023] 和基于段落的问题回答任务 [Rajpurkar et al., 2016]。

Preprint. Under review.

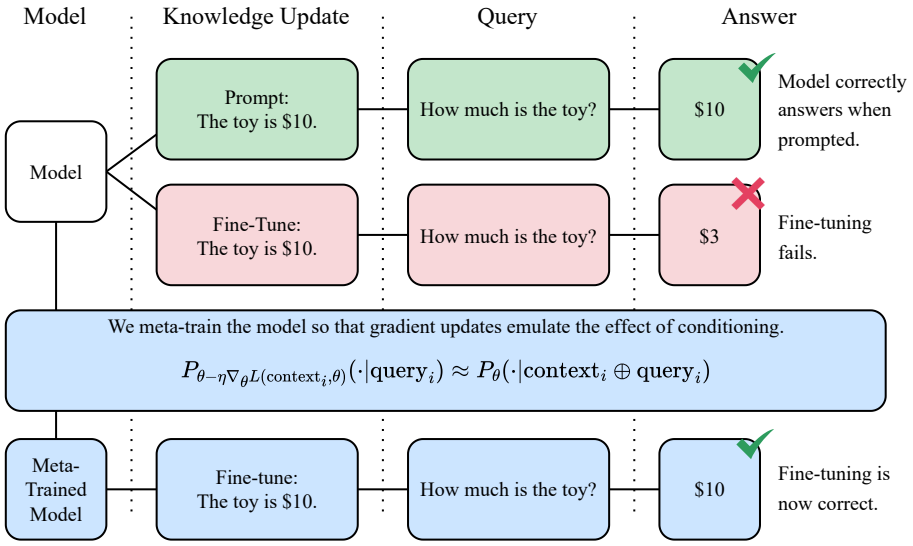

图 1: **元训练模型以通过微调来模拟条件设置**。在低样本或 1 样本情况下，微调可能会由于各种原因失败，包括无法泛化、无法处理反向措辞以及难以覆盖现有先验。另一方面，模型通常能够利用提示中呈现的信息。我们研究了一种元训练程序，以将提示的效果提炼到条件设置中，使模型可以通过微调来整合新信息。

2 相关工作

提示工程 语言模型展现出了遵循提示中的指令并进行上下文学习的惊人能力 [Brown et al., 2020]。因此，提示成为了当代研究的核心焦点，众多研究正在调查其行为特征和潜在机制 [Min et al., 2022, Xie et al., 2021, Olsson et al., 2022]。值得注意的是，近期的一些研究表明，变换器网络通过类似于梯度下降的机制来进行上下文学习 [von Oswald et al., 2022, Ahn et al., 2023, Dai et al., 2023]。这与我们所追求的目标相反：使模型上的梯度下降类似模型如何进行上下文学习。相关研究还探索了元学习方法，这些方法通过对多样化的基于提示的任务进行微调来增强上下文学习能力 [Min et al., 2021, Chen et al., 2021]。

知识表示 另一项研究调查了知识如何嵌入参数 [Mosbach et al., 2024] 中，如何在参数和上下文中解耦知识 [Neeman et al., 2023]，如何从参数中移除知识 [Zhang et al., 2024, Sha et al., 2024]，以及如何直接将新信息嵌入模型的参数而非通过输入上下文提供它，这通常被称为模型编辑 [Zhu et al., 2020, Cao et al., 2021, Meng et al., 2022a,b, Liu et al., 2022]，这些研究往往与模型可解释性的努力密切相关。

微调 细调模型通常会在新任务上的多个示例上提高下游性能。然而，当应用于使用单个新的知识片段更新模型时，细调存在许多限制，包括泛化能力差 [Zhu et al., 2020, Cao et al., 2021]、无法覆盖固有的先验知识 [Onoe et al., 2022]，以及有限的能力来建模训练数据的含义（例如以不同的表面形式呈现的信息 [Berglund et al., 2023] 或不同语言中的信息 [Ifergan et al., 2024]）。此外，细调大型模型的计算需求对于没有昂贵硬件的人来说可能是不可行的。参数高效的细调旨在通过更新少量参数来解决这些问题。一种广泛采用的方法是 LoRA，它将更新限制为低秩矩阵 [Hu et al., 2021]。对于该领域的更广泛的技术概述，请参见 Lialin et al. [2023] 的调查。

基于梯度的元学习 在大型语言模型被广泛采用之前，元学习研究的一条路线集中在基于梯度的技术上，如模型不可知的元学习（MAML）[Andrychowicz et al., 2016, Ravi and Larochelle, 2016, Li et al., 2017, Antoniou et al., 2018, Nichol et al., 2018, Rajeswaran et al., 2019]。MAML 及其扩展使用一个双层优化框架，通过梯度来学习一种初始化方法，使得在新任务中能够用最少的数据进行快速适应。这些方法启发了许多变体 [Nichol and Schulman, 2018, Choshen et al., 2022]，并因其在跨领域如计算机视觉和强化学习的少量样本学习场景中的有效性而被广泛研究 [Clavera et al., 2018, Liu et al., 2019, Sinitsin et al., 2020]。最近，一些工作集中在使用基于梯度的元学习来提高小型模型在不同类型任务上的性能 [Sinha et al., 2024]。

上下文蒸馏 由于提示通常优于微调，将提示信息注入模型参数的一种方法是在带有提示的模型生成的数据上对模型进行微调 [Wang et al., 2021, Askell et al., 2021, Choi et al., 2022, Akyürek et al., 2024]。现有工作已经探讨了各种策略来提高该技术的有效性，包括更好的采样策略和用于数据增强的手工制作提示。

3 方法论

3.1 MAML 评审

我们首先回顾基于梯度的元学习 [MAML; Finn et al., 2017] 作为我们的方法背景。假设我们有一个任务数据集，每个任务由一个演示对 (x_d, y_d) 和一个从相同分布中抽取的评估对 (x_e, y_e) 组成。我们假设存在一个通用损失函数 $L(x, y, \theta) \to \mathbb{R}$，它将（输入、输出）对和参数 θ 映射到一个损失值。MAML 的目标是找到一个模型初始化，使得在演示数据对上进行微调后能够有效地预测评估数据对。首先，我们定义初始化参数 θ_0 的更新，该更新学习了 θ'：

$$\theta' = \theta_0 - \eta \nabla_{\theta_0} L(x_d, y_d, \theta_0) \tag{1}$$

其中 η 是某个学习率。我们将此更新称为元学习中涉及的双层优化问题的**内部循环**。然后，该任务的元学习损失是我们评估对上的微调参数 θ' 的损失：

$$L_{\text{ML}}(x_e, y_e, \theta') = L(x_e, y_e, \theta - \eta \nabla_\theta \mathcal{L}(x_d, y_d, \theta)). \tag{2}$$

给定一组任务 $\mathcal{T}$，完整的元学习损失是：

$$\mathbb{E}_{((x_d,y_d),(x_e,y_e))\in\mathcal{T}}[L(x_e, y_e, \theta - \eta \nabla_\theta L(x_d, y_d, \theta))] \tag{3}$$

我们将其称为双层优化问题的**外循环**。优化方程 3 产生了一组初始参数 θ，在进行少量梯度下降后，这些参数在新任务上表现出良好的性能。

3.2 元学习从条件开始

上述过程依赖于一组带有标签的测试示例 x_e,y_e。但如果没有任何这样的标签，而是有一个自由形式的自然语言（上下文、查询）对的数据集 $\{(c_i, q_i)\}_N$—例如，(鲍勃在百思买工作。，鲍勃在电子产品商店工作吗？)。最后，假设我们有一个带有参数 θ 的现成语言模型。

已经确立的是，经过足够大规模训练的语言模型能够通过条件处理在 c_i 上回答类似 q_i 的问题。但是正如第 1 节中所指出的，这种条件操作会带来存储和处理成本，并且如果语言模型必须回答的问题数量很大，我们可能会更倾向于使用将新信息编码在参数而非输入数据中的语言模型。在这里，我们将描述如何利用基于梯度的元学习程序来实现这一效果。

令给定一个（上下文，问题）对的响应条件分布为：

$$P_\theta(\cdot \mid c_i \oplus q_i) \tag{4}$$

其中 $\oplus$ 表示连接。另外，设 $L(c_i, \theta)$ 是使用参数 θ 对上下文 c_i 进行下一个标记预测损失。在上下文 c_i 中对参数 θ 进行学习率 η 的微调后，我们得到了新的参数：

$$\theta' = \theta - \eta \nabla_\theta L(c_i, \theta). \tag{5}$$

此操作对应于 MAML 的内循环。我们希望找到 θ 使得模型在微调后的表现与条件化后的表现相同。因此，我们希望找到参数 θ^*，使得：

$$P_{\theta^*}(\cdot \mid c_i \oplus q_i) \approx P_{\theta^* - \eta \nabla_{\theta^*} L(c_i, \theta^*)}(\cdot \mid q_i) \quad \forall i. \tag{6}$$

同时，我们希望 θ^* 保持广泛的语言建模能力。令 $L_{\text{LM}}(\theta)$ 表示一个通用的语言建模损失，并令 $\mathrm{D}(P \,\|\, Q)$ 是概率分布 P 和 Q 之间的发散度量。然后我们可以将优化目标公式化为：

$$\arg\min_{\theta^*} \sum_i \mathrm{D}\left(P_{\theta^*}(\cdot \mid c_i \oplus q_i) \,\middle\|\, P_{\theta^* - \eta \nabla_{\theta^*} L(c_i, \theta^*)}(\cdot \mid q_i)\right) + \lambda L_{\text{LM}}(\theta^*) \tag{7}$$

对于某个权重 λ。这对应于 MAML 的外循环。

由于在我们的设置中，条件化几乎总是优于微调，我们不期望能够显著改进 $P_{\theta^*}(r_i | c_i \oplus q_i)$。因此，我们将预训练好的教师模型固定为初始模型 θ_B。然后，我们将使用 θ_B 的有条件分布视为“黄金”分布，并将等式 6 的左侧固定。此外，为了避免在计算分歧度量时从条件分布中抽样多标记连续项的计算开销，我们使用教师模型贪婪解码的条件概率的 KL 分歧。因此，我们的方法最终优化的是：

$$\arg\min_{\theta^*} \sum_i \sum_j \mathrm{KL}\left(P_{\theta_B}(\cdot \mid c_i \oplus q_i \oplus \hat{a}_{i,<j}) \,\middle\|\, P_{\theta^* - \eta \nabla_{\theta^*} L(c_i, \theta^*)}(\cdot \mid q_i \oplus \hat{a}_{i,<j})\right) + \lambda L_{\text{LM}}(\theta^*) \tag{8}$$

其中，$\hat{a}_i \sim P_{\theta_B}(\cdot \mid c_i \oplus q_i)$ 和 $\hat{a}_{i,<j}$ 表示 $\hat{a}$ 的前 j 个标记。

3.3 元学习与真实标签

为了建立性能的上限，我们的实验还评估了标准的 MAML 样式的训练，假设可以访问到 ground-truth 答案 a_i。（这可以被视为通过使用 gold-responses 而不是条件生成分布来模拟完全条件准确性。）在这种情况下，我们的目标是：

$$\arg\min_{\theta^*} \sum_i \log P_{\theta^* - \eta \nabla_{\theta^*} L(c_i, \theta^*)}(a_i \mid q_i) + \lambda L_{\text{LM}}(\theta^*) \tag{9}$$

其中我们使用 ground-truth 答案 a_i 作为标签。

3.4 元学习与 LoRA

为了减少训练所需的内存，我们还尝试将模型限制为低秩更新。我们检查了两种可能的实现方法：

1. 外循环中模型参数 θ^* 的低秩更新，以及使用相同的低秩适配器进行内部步骤。该实验评估是否可以通过元学习过程找到对下游微调有用的 LoRA 初始化。
2. 外层循环中对模型参数 θ^* 进行低秩更新，但在内层步骤进行全秩更新。本实验评估是否需要高秩（且可能更复杂）的更新来使模型更适合微调，或者简单的低秩更新就足够。

4 实验设置

4.1 数据

我们专注于四项任务：

1. **角色描述**（类似于逆转诅咒 [Berglund et al., 2023]）：该数据集包含三元组，分别为带有名称前描述的句子、描述的释义以及相同的名称。例如，一个三元组为（“第一个在火星上行走的人是埃文 · 阿姆斯特朗”，“历史书记录第一个到达火星的人是”，“埃文 · 阿姆斯特朗。”）。在学习了第一种描述后，模型应能够用相同的名字完成释义的描述。我们使用描述位于名称之前的句子，因为评估名字的补全比描述的补全更容易。由于原始逆转诅咒数据集中的示例数量有限，我们生成了一个包含 5000 个训练样本和 500 个测试样本的新数据集。

2. **逆向咒语** [Berglund et al., 2023] : 该数据集包含由以下三元组组成的：一个人名在其描述之前的句子、一个描述以及一个名字。例如，一个三元组是 (“埃文 · 阿姆斯特朗是第一个访问火星的人。”, “第一个在火星上行走的是”, “埃文 · 阿姆斯特朗.”)。这与角色描述数据集相同，只是模型学习的句子被反转了。从第一句话中学习后，模型应该能够用正确的名字完成描述。类似于角色描述任务，我们生成一个同样大小的新数据集。

3. **SQuAD** [Rajpurkar et al., 2016] : 我们使用标准的 SQUAD 数据集，该数据集包含上下文、问题和答案三元组。在学习了上下文之后，模型应该能够正确地用一个答案回答问题。为了避免污染，我们使用随数据集提供的划分。预处理后，我们有 82031 条训练记录和 10380 条测试记录。

4. **维基文本** [Merity et al., 2016] : 我们将 WikiText 数据集转换为下一个标记预测任务。我们随机将 WikiText 数据集拆分为三元组。例如，文本 “它改编自罗伯特 · 路易斯 · 史蒂文森于 1886 年创作的短篇小说《化身博士》。故事聚焦于伦敦受人尊敬的医生亨利 · 杰基尔,” 可以被拆分为（“它改编自罗伯特 · 路易斯 · 史蒂文森于 1886 年创作的短篇小说《化身博士》。故事聚焦于”, “the”, “受人尊敬的伦敦医生亨利 · 杰基尔”）。在学习了三元组的第一部分之后，模型应该使用三元组的第三部分来完成第二部分。为了避免污染，我们使用数据集自带的拆分。预处理后，我们有 343586 条训练记录和 758 条测试记录。

每个数据集是一组三元组（上下文，查询，响应）。我们可以用三种方式评估每个三元组：

1. **无上下文（NC）**: 我们忽略上下文并评估条件于查询下的响应的负对数似然（NLL）和准确性。

2. **提示（提示）**: 我们评估响应在提示中与查询拼接的上下文条件下的 NLL 和准确性。

3. **微调（FT）**: 我们对上下文进行微调，然后评估基于查询的响应的 NLL 和准确性。

如前所述，普遍观察到的情况是，对于大多数任务，提示准确性>微调准确性>零样本准确性。我们的目标是提高微调准确性以达到提示准确性的水平。

4.2 实现细节

我们使用固定学习率为 10^{-3} 的梯度下降进行内部步骤，因为我们发现它在有无元训练的情况下都能表现出色。我们在外部循环中使用 Adam[Kingma and Ba, 2014]，并利用保留数据调整每个任务的外部学习率。由于梯度方差较高，我们发现激进的梯度裁剪是有帮助的。我们的批量大小为 16，几乎填满了 80GB H100 的 VRAM，因为二阶优化问题需要与模型尺寸和批量大小的乘积成比例的 VRAM 来存储内部步骤后的适应参数。除非另有说明，我们会训练一个周期，因为我们观察到元学习在一个周期后开始显著过拟合。我们使用 Llama 3.2 1B 模型 [Dubey et al., 2024] 进行实验。大多数实验在一台 H100 上需要几个小时。

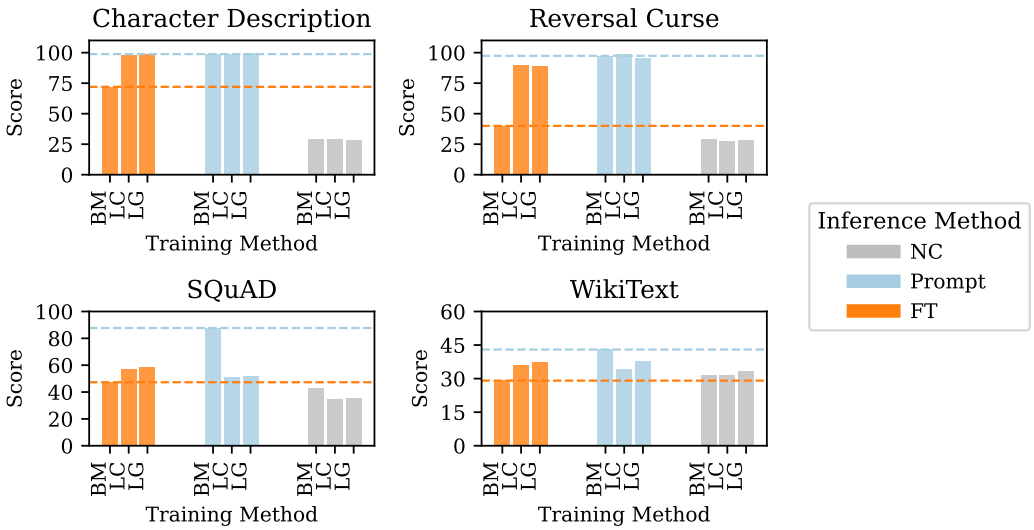

图 2: **模型在不同训练策略和任务上的表现**。我们评估了基础模型（BM）、从条件过程元学习后的模型（LC）以及从金标签元学习后的模型（LG）。我们在三种不同的设置下评估每个任务。在无上下文（NC）的设置中，要求模型在没有任何上下文的情况下执行任务。这是我们的方法性能的一个下限。在提示（Prompt）设置中，要求模型在提供上下文的情况下执行任务。这是我们的方法性能的一个上限。在微调（FT）设置中，在适当的上下文中进行微调后，要求模型执行任务。请注意，基础模型的原始微调准确率作为我们方法有效性的上限。基础模型使用上下文的准确性为 LC 和 LG 程序都设定了上限。所有准确率的标准误差在测试集上小于 ±2%。

4.3 微调准备

对于每个列出的数据集，我们首先在 Prompt 和 NC 配置下的数据集的小子集上对模型进行微调。这确保了语言模型已经见过数据集的格式，并将我们的方法的影响与仅通过微调获得的性能区分开来。

5 结果

5.1 梯度下降能模拟提示吗？

结果如图 2 所示。我们从一个普遍的观察开始，即基于条件的学习的结果与 oracle 元学习的结果极为接近。对于角色描述任务，我们发现我们的程序实现了非常高的准确率。这是一个简单的任务，因此它验证了元学习过程至少对这类受限的任务是可能实现的。有趣的是，在反转诅咒数据集上，准确性较低。即使经过元训练，模型通过梯度步骤学习反向方向也更加困难，尽管其性能显著优于仅进行微调的基础模型评估。对于 SQuAD 数据集，我们只能恢复到提示模型性能的大约四分之一。观察损失曲线，我们推测这是因为缺乏数据，因为直到我们用完用于训练的数据之前，损失一直在持续下降。对于 WikiText 数据集，我们发现能够恢复到提示和简单微调之间性能差距的一半左右。我们推测 WikiText 数据集的相对较好表现是由于较大的数据集规模。

表 1: **将 LoRA 应用于黄金元学习设置**。基础模型部分的两列给出了使用上下文和基础模型微调准确率的结果。完整元学习部分给出了对所有模型参数进行元学习后的微调准确率。LoRA 外循环部分给出仅在模型参数上执行秩-1 更新并完全内步微调后的准确率。LoRA 内步步骤部分给出了未经训练的 LoRA 适配器和经过元训练的 LoRA 适配器的微调准确率。所有结果都是基于从黄金响应中学习的设置。所有报告的数据在测试集上的标准误差都在 ±2 以内。

	基础模型		完整元数据	LoRA 外环	LoRA 内部步骤	
Dataset	Prompt	FT	FT	FT	Untrained FT	Meta FT
SQuAD	87.7	47.3	58.6	59.4	45.1	72.0
Wikitext	43.0	29.1	37.2	37.5	32.2	38.3

5.2 这种效果能否通过低秩更新实现？

为了研究 LoRA 的影响，我们使用 SQuAD 和 WikiText 数据集进行实验，因为字符描述和逆转诅咒数据集太容易学习了。由于如第 5.1 节所示，条件学习和从黄金响应中学习的结果非常相似，我们在表 1 中仅展示了从黄金响应中学习的结果，因为它稍微少一些噪声。

5.2.1 我们能否元学习一个适应良好的 LoRA 初始化？

如表 1 所示，我们发现 LoRA 内部 + 外部实现相较于全秩元学习和内部步骤更新表现非常好。性能在 SQuAD 数据集上尤其更好。我们推测，约束内部步骤的秩是一种强大的归纳偏差，只需要低秩更新即可注入更多知识 [Hu et al., 2021]。这具有正则化效果，对于我们之前观察到缺乏数据的 SQuAD 任务尤其有用。

5.2.2 为了改进全微调，我们需要什么样的秩更新？

我们研究元学习过程如何改变模型参数以提高学习效果。具体来说，我们提出的问题是需要什么样的秩更新来改进下游微调。

如表 1 所示，我们发现秩-1 更新能达到与全秩更新相当的性能。因此，这表明*秩 1 更新足以使模型表现出更好的微调性能。*

5.3 元训练的模型是否成功利用了上下文？

在检查元训练模型对 SQuAD 数据集的响应时，我们在图 4 中展示了模型正确回答的两种不同方式。

在第一个示例中，基础模型在相关上下文的微调前后均给出了错误的回答。元训练模型在对上下文进行微调前也给出了错误的回答，但在对上下文进行了微调后成功作出了正确的回答。在第二个示例中，元学习过程使得模型能够自动给出正确答案，即使没有针对相关上下文进行微调。

在第二个示例中，元学习并未发生，因为模型可以在不基于上下文进行梯度计算的情况下成功回答问题。为了验证这并不是改进的主要原因，我们在表 4 中统计了这种行为的频率。我们观察到，绝大多数的改进需要对上下文执行内部步骤。这一点支持了结论，即元学习过程不仅在教会模型猜测 SQuAD 的答案，而且是通过梯度步骤来利用上下文。

为了进一步验证我们的元训练模型实际上正在使用上下文，我们还检查了如果我们使用无关的上下文会发生什么。我们从相同的分布中随机采样一个上下文，在该上下文中对元训练模

表 2: **学习多个上下文**。我们评估了基础模型、原始元训练模型以及明确训练以同时处理多个上下文的元训练模型。模型被元训练来处理的上下文数量在括号中指定。所有评估均在 SQuAD 数据集上进行。我们看到，明确训练用于处理多个上下文的模型表现更好，尽管性能有所下降。所有结果均为从黄金响应设置中学习的结果。所有报告的结果在反复采样随机更新组后标准误差小于 ±3。

Model	1 Update	4 Updates	16 Updates
Base	47.3	43.6	42.6
Meta-Learn (1 ctx trained)	**57.2**	46.3	39.6
Meta-Learn (4 ctx trained)	55.1	**51.5**	45.6
Meta-Learn (16 ctx trained)	55.0	47.2	**46.6**

表 3: **跨数据集元学习**。我们测试维基文本元训练模型是否转移到 SQuAD 数据集。我们在两种设置下进行测试。在第一种顺序设置中，我们先在维基文本数据集上进行元学习，然后再在 SQuAD 数据集上进行微调。然后我们测试 SQuAD 数据集是否也获得元学习性能的提升。在第二种联合设置中，我们在维基文本集合上执行元学习的同时对 SQuAD 数据集进行微调。在这两种设置下，我们发现通过迁移学习在元学习性能方面只有轻微到可以忽略的改进。我们还测试了我们的程序之后维基文本元学习性能发生了什么变化，发现存在一定程度的遗忘现象。所有报告的数据都在测试集上具有 ±2 的标准误差。

Method	Evaluation	No Meta-Learning Acc.	In-Domain Acc.	Transfer Acc.
Sequential	SQuAD FT	47.3	58.6	47.8
Joint	SQuAD FT	47.3	58.6	48.0
Sequential	WikiText Retain	29.1	37.2	34.8
Joint	WikiText Retain	29.1	37.2	34.8

型进行微调，然后评估模型性能。我们在 SQuAD 和 WikiText 数据集上的评估结果见表 5。如表所示，无关的上下文会损害模型的性能。

5.4 元训练的模型能否同时保留多个上下文？

我们还研究了我们的元训练模型是否可以同时保留多个上下文。我们使用 SQuAD 数据集进行这项研究，因为 SQuAD 的上下文是非矛盾的，这使得保留多个上下文成为可能。

我们首先评估从 5.1（仅训练一次更新）开始的模型的朴素持续训练在多个上下文中的表现。我们还对两个模型进行元训练，通过在内部步骤中执行批量更新而不是单个示例的更新来显式处理多上下文更新。这另外具有减少 VRAM 使用的计算优势，因为这将我们必须实现的调整参数数量除以内部批次大小。

我们的结果如表 2 所示。元训练模型在对多个上下文进行微调时优于基础模型，尽管该方法的单个上下文版本性能有所下降。

5.5 元学习能否跨数据集迁移？

我们首先观察到，WikiText 预训练模型在 SQuAD 上的表现不佳。例如，对于问题“维多利亚和阿尔伯特博物馆计划在哪座苏格兰城市开设一个品牌的画廊?”与原始 Llama 模型直接回答“邓迪”不同，预训练模型开始了一段完整的叙述：“这个问题的答案是……”，这类似

于 WikiText 中的句子结构。因此，我们进行了两个实验以了解：（1）预训练模型是否能迁移到其他数据集上；（2）在下游任务中对预训练模型进行微调是否会引发灾难性的遗忘。

在第一次实验中，我们采用 WikiText 元训练模型并在 SQuAD 数据集上对其进行微调。然后我们感兴趣的是这个新模型是否保留了在 WikiText 数据集上的元学习性能以及它是否在 SQuAD 数据集上表现出增强的学习性能。在第二次实验中，我们在 SQuAD 微调任务和 WikiText 任务上联合训练模型。然后我们调查该模型是否在 SQuAD 数据集上表现出增强的学习性能以及其在 WikiText 元学习任务上的表现。对于这两个实验，我们都使用了原始微调时使用的 SQuAD 数据集的完全相同的子集。这防止了模型被进一步地在更多的 SQuAD 数据集上进行微调，从而不公平地提高性能。

两个实验的结果呈现在表 3 中。令人惊讶的是，我们发现两种方法给出了类似的结果：在两种情况下，WikiText 迁移学习任务很难推广到 SQuAD 任务上。更令人惊讶的是，我们还发现，在微调后 WikiText 上的表现损失与联合优化的表现成本相似。这表明两种类型的任务（微调与迁移学习）差异足够大，以至于很难同时对模型进行优化。

6 结论

在这篇论文中，我们提出了一种元训练语言模型的方法，使得梯度更新模拟了对新信息进行条件化的影响。我们发现这种方法能够在微调和提示之间部分地弥合差距，使模型能够通过基于梯度的更新实现一些“类似提示”的泛化。令人惊讶的是，我们发现在大多数情况下，秩 1 更新就足以提高模型的微调能力。尽管有这些有希望的结果，我们的方法仍然存在局限性。最值得注意的是，有限的计算资源阻止了我们在大型、多样化的数据集上进行扩展元训练。我们假设扩大元训练过程会带来跨任务和领域的更好泛化性能。此外，未来的工作可以研究更好地保留和组合多个更新的方法，这是实现稳健连续学习的关键能力。

参考文献

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *ArXiv*, abs/2306.00297, 2023. URL https://api.semanticscholar.org/CorpusID:258999480.

Afra Feyza Akyürek, Ekin Akyürek, Leshem Choshen, Derry Tanti Wijaya, and Jacob Andreas. Deductive closure training of language models for coherence, accuracy, and updatability. In *Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://api.semanticscholar.org/CorpusID:267028613.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *ArXiv*, abs/2211.15661, 2022. URL https://api.semanticscholar.org/CorpusID:254043800.

Marcin Andrychowicz, Misha Denil, Sergio Gomez Colmenarejo, Matthew W. Hoffman, David Pfau, Tom Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Neural Information Processing Systems*, 2016. URL https://api.semanticscholar.org/CorpusID:2928017.

Antreas Antoniou, Harrison Edwards, and Amos J. Storkey. How to train your maml. *ArXiv*, abs/1810.09502, 2018. URL https://api.semanticscholar.org/CorpusID:53036488.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Benjamin Mann, Nova Dassarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, John Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment. *ArXiv*, abs/2112.00861, 2021. URL https://api.semanticscholar.org/CorpusID:244799619.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: Llms trained on "a is b" fail to learn "b is a". *ArXiv*, abs/2309.12288, 2023. URL https://api.semanticscholar.org/CorpusID:262083829.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Ma teusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. URL https://api.semanticscholar.org/CorpusID:218971783.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://api.semanticscholar.org/CorpusID:233289412.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. Meta-learning via language model in-context tuning. *ArXiv*, abs/2110.07814, 2021. URL https://api.semanticscholar.org/CorpusID:239009828.

Eunbi Choi, Yongrae Jo, Joel Jang, and Minjoon Seo. Prompt injection: Parameterization of fixed inputs. *ArXiv*, abs/2206.11349, 2022. URL https://api.semanticscholar.org/CorpusID:249953762.

Leshem Choshen, Elad Venezian, Noam Slonim, and Yoav Katz. Fusing finetuned models for better pretraining. *arXiv preprint arXiv:2204.03044*, 2022.

Ignasi Clavera, Jonas Rothfuss, John Schulman, Yasuhiro Fujita, Tamim Asfour, and P. Abbeel. Model-based reinforcement learning via meta-policy optimization. In *Conference on Robot Learning*, 2018. URL https://api.semanticscholar.org/CorpusID:52282277.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. *ArXiv*, abs/2212.10559, 2023. URL https://api.semanticscholar.org/CorpusID:254877715.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aur'elien Rodriguez, Austen Gregerson, Ava Spataru, Bap tiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cris tian Cantón Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino,

Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriele Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Ken-591 neth Heafield, Kevin R. Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuen ley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Niko lay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Ro main Sauvestre, Ron nie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sa hana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Vir ginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit ney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Po-Yao (Bernie) Huang, Beth Loyd, Beto de Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco

Guzm' an, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory G. Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pe dro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models. *ArXiv*, abs/2407.21783, 2024. URL `https://api.semanticscholar.org/CorpusID:271571434`.

Chelsea Finn, P. Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017. URL `https://api.semanticscholar.org/CorpusID:6719686`.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. URL `https://api.semanticscholar.org/CorpusID:235458009`.

Maxim Ifergan, Leshem Choshen, Roee Aharoni, Idan Szpektor, and Omri Abend. Beneath the surface of consistency: Exploring cross-lingual knowledge representation sharing in llms. *arXiv preprint arXiv:2408.10646*, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL https://api.semanticscholar.org/CorpusID:6628106.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few shot learning. *ArXiv*, abs/1707.09835, 2017. URL https://api.semanticscholar.org/CorpusID:25316837.

Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *ArXiv*, abs/2303.15647, 2023. URL https://api.semanticscholar.org/CorpusID:257771591.

Hao Liu, Richard Socher, and Caiming Xiong. Taming maml: Efficient unbiased meta-reinforcement learning. In *International Conference on Machine Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:174800385.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *ArXiv*, abs/2205.05638, 2022. URL https://api.semanticscholar.org/CorpusID:248693283.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *Neural Information Processing Systems*, 2022a. URL https://api.semanticscholar.org/CorpusID:255825985.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *ArXiv*, abs/2210.07229, 2022b. URL https://api.semanticscholar.org/CorpusID:252873467.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843, 2016. URL https://api.semanticscholar.org/CorpusID:16299141.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *ArXiv*, abs/2110.15943, 2021. URL https://api.semanticscholar.org/CorpusID:240288835.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *ArXiv*, abs/2202.12837, 2022. URL https://api.semanticscholar.org/CorpusID:247155069.

Marius Mosbach, Vagrant Gautam, Tomás Vergara-Browne, Dietrich Klakow, and Mor Geva. From insights to actions: The impact of interpretability and analysis research on nlp. *arXiv preprint arXiv:2406.12618*, 2024.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. DisentQA: Disentangling parametric and contextual knowledge with counterfactual question answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the*

61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10056–10070, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.559. URL https://aclanthology.org/2023.acl-long.559/.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *ArXiv*, abs/1803.02999, 2018. URL https://api.semanticscholar.org/CorpusID:4587331.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova Dassarma, Tom Henighan, Benjamin Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, John Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom B. Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *ArXiv*, abs/2209.11895, 2022. URL https://api.semanticscholar.org/CorpusID:252532078.

Yasumasa Onoe, Michael J.Q. Zhang, Eunsol Choi, and Greg Durrett. Entity cloze by date: What lms know about unseen entities. In *NAACL-HLT*, 2022. URL https://api.semanticscholar.org/CorpusID:248525074.

Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Neural Information Processing Systems*, 2019. URL https://api.semanticscholar.org/CorpusID:202542766.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Conference on Empirical Methods in Natural Language Processing*, 2016. URL https://api.semanticscholar.org/CorpusID:11816014.

Sachin Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2016. URL https://api.semanticscholar.org/CorpusID:67413369.

A. Sha, Bernardo Pereira Nunes, and Armin Haller. "forgetting" in machine learning and beyond: A survey. *ArXiv*, abs/2405.20620, 2024. URL https://api.semanticscholar.org/CorpusId:270199793.

Sanchit Sinha, Yuguang Yue, Victor Soto, Mayank Kulkarni, Jianhua Lu, and Aidong Zhang. Maml-en-llm: Model agnostic meta-training of llms for improved in-context learning. *ArXiv*, abs/2405.11446, 2024. URL https://api.semanticscholar.org/CorpusID:269921787.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy V. Pyrkin, Sergei Popov, and Artem Babenko. Editable neural networks. *ArXiv*, abs/2004.00345, 2020. URL https://api.semanticscholar.org/CorpusID:213938729.

Johannes von Oswald, Eyvind Niklasson, E. Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:254685643.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. Towards zero-label language learning. *ArXiv*, abs/2109.09193, 2021. URL https://api.semanticscholar.org/CorpusID:237572306.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *ArXiv*, abs/2111.02080, 2021. URL https://api.semanticscholar.org/CorpusID:241035330.

Eric Zhang, Leshem Chosen, and Jacob Andreas. Unforgettable generalization in language models, 2024. URL https://arxiv.org/abs/2409.02228.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X. Yu, and Sanjiv Kumar. Modifying memories in transformer models. *ArXiv*, abs/2012.00363, 2020. URL https://api.semanticscholar.org/CorpusID:227238659.

A 技术附录和补充材料

表 4: **元训练对模型性能的影响和方式**。上面板展示了元训练如何使 SQuAD 答案变得更好的定性示例，无论是直接还是经过一个梯度步骤之后。下板块提供了正确响应在验证和训练分割中的定量分布，强调了上下文的影响。

元训练效果的定性示例

Question and Correct Answer	Base	Base Step	ML	ML Step
The V&A is looking to open a branded gallery in which city in Scotland? Answer: Dundee	Edinburgh	Edinburgh	Edinburgh	Dundee
The Rhine forms the border between Austria and what other country? Answer: Switzerland	Germany	Germany	Switzerland	Switzerland

来自元学习模型的正确响应分布

Split	Always Incorrect	Always Correct	Correct w/o Context	Only Correct w/ Context
Val	55.6%	28.0%	3.6%	12.8%
Train	29.9%	29.7%	5.8%	34.7%

表 5: **在不相关上下文上的微调**。我们比较了我们的元训练模型在正常上下文和从同一数据集中随机抽取的无关上下文中微调后的相关任务性能。正如预期，我们观察到性能有显著下降。我们还展示了基准模型的无上下文准确率以供对比。所有报告的数据在测试集上的标准误差都在 ±2 以内。

Dataset	No Context	Irrelevant Context	Normal Context
SQuAD	35.2	29.7	58.6
WikiText	33.1	.010	37.2