

TRANSFORMER-SQUARED: 自适应大型语言模型

孙琪^{1,2*}, 切廷爱德华多^{1*}, 唐宇津^{1*} ¹Sakana AI,
日本 ²东京科学研究所, 日本 {qisun,edo,yujintang}
@sakana.ai *贡献相同

摘要

自适应大型语言模型 (LLM) 旨在解决传统微调方法带来的挑战, 这些方法通常计算密集且在处理各种任务方面的能力是静态的。我们介绍了Transformer² (Transformer-Squared), 这是一种新颖的自适应框架, 它通过选择性地调整其权重矩阵的奇异分量来实时地使LLM适应未见的任务。在推理过程中, Transformer²采用两遍机制: 首先, 调度系统识别任务属性, 然后动态混合使用强化学习训练的任务特定“专家”向量, 以获得针对传入提示的目标行为。我们的方法始终优于LoRA等普遍方法, 参数更少, 效率更高。此外, Transformer²展示了在不同LLM架构和模态 (包括视觉语言任务) 上的多功能性。Transformer²代表了重大进步, 它提供了一种可扩展、高效的解决方案, 用于增强LLM的适应性和特定任务性能, 为真正动态、自组织的AI系统铺平了道路。我们提供了完整的源代码, 网址为<https://github.com/SakanaAI/self-adaptive-llms>。

1 引言

自适应大型语言模型 (LLM) 将代表人工智能的重大进步, 提供一个框架, 使模型能够实时适应不同的任务和动态环境。这一概念的灵感来自于长期以来神经网络修改自身权重以动态适应任务的想法 (Schmidhuber, 1993; Irie et al., 2022) 以及神经网络生成其他网络权重, 正如超网络和相关方法所普及的那样 (Ha et al., 2017; Stanley et al., 2009)。虽然组合性和可扩展性对于有效的适应至关重要, 但当前的 LLM 训练方法未能同时实现这两个特性。我们的研究旨在提出一个开创性的解决方案来实现这一愿景。

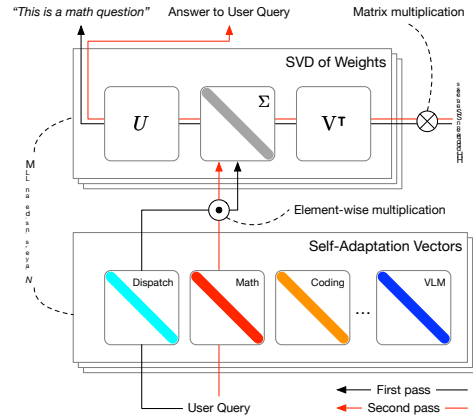


图1: Transformer²概述。在训练阶段, 我们调整权重矩阵奇异值的尺度以生成一组“专家”向量, 每个向量专门处理一种类型的任务。在推理阶段, 采用两遍过程, 第一遍应用特定任务的专家, 第二遍生成答案。

传统上, 大型语言模型的后期训练旨在通过一次大规模的训练来优化模型的多种能力。虽然这种“一次性”微调框架从简易性的角度来看是理想的, 但在实践中却很难实现。例如, 后期训练仍然非常耗费资源, 导致大量的计算成本和训练时间。此外, 还存在

在扩展数据广度时，往往会面临显著的性能权衡，这使得同时克服过拟合和任务干扰变得极具挑战性。

相比之下，自适应模型提供了一种更灵活、更高效的方法。与其试图一步训练一个适用于所有任务的大型语言模型，不如离线开发专家模块，并根据需要将其添加到基础大型语言模型中（Kang等人，2024）。这允许模型根据手头的任务动态地修改其行为，而无需不断地重新调整。除了具有独立组件的好处外，这种模块化还支持持续学习，使模型能够随着时间的推移添加新技能，而不会发生灾难性遗忘。此外，自适应大型语言模型反映了神经科学和计算生物学中一个已确立的原理，即大脑会根据手头的任务激活特定区域（Loose等人，2017），并根据不断变化的任务需求动态地重新配置其功能网络（Davison等人，2015）。

原则上，实现自适应LLM的第一步可以通过开发专门的专家模块来实现，每个模块都通过低秩自适应（LoRA）（Hu等人，2021）等技术进行微调（Kaplan等人，2020）。然后，这些专家模块可以根据任务需求在运行时动态组合，这个过程可以通过类似专家混合（MoE）的系统（Tianlong等人，2024）有效地管理。然而，要使这种方法既可扩展又具有组合性，需要解决几个挑战。首先，微调LLM以创建多个专家模块会显著增加需要训练的参数量。实际上，即使使用像LoRA这样的参数高效方法，这些模块的累积大小也会迅速增加，导致存储和计算需求增加。其次，这些专家模块往往容易过拟合，这种现象在使用较小的数据集或狭窄的任务领域进行训练时尤其普遍。第三，这些专家模块的灵活组合也带来了目前仍未解决的挑战，这些挑战构成了开放的研究问题。

为克服这些局限性，我们首先提出奇异值微调（SVF），这是一种新颖的、参数高效的微调（PEFT）方法，用于获得用于自适应的有效构建块。SVF通过提取和调整模型权重矩阵中的奇异值来工作。通过关注这种有原则的参数化，我们的方法减轻了过拟合的风险，大大降低了计算需求，并允许固有的组合性。我们展示了这些特性使我们能够通过狭窄的数据集上使用强化学习（RL）廉价地获得一组有效的特定领域“专家”向量 $\{v^*\}$ ，直接优化单个主题上的任务性能。

然后，我们介绍了完整的Transformer²（Transformer-Squared）框架，以利用自适应的基本原理增强大型语言模型。给定来自未知任务的提示，Transformer²包含一个两遍推理机制，我们在图1中进行了说明。在第一遍中，Transformer²执行模型并观察其测试时行为，收集相关信息以了解解决当前问题所需的技能。在第二遍中，我们的框架使用这些信息来组合可用的专家向量，并为大型语言模型的基础权重提供新的修改，这些修改专门针对其测试时条件。我们设计了三种不同的自适应策略，这些策略可以在Transformer²中使用，我们证明这些策略随着对测试时条件访问的增加而带来单调的性能提升。

我们通过广泛的实验，在各种大型语言模型和任务上评估了SVF和完整的Transformer²框架。首先，当在特定领域的数据集上进行训练时，我们证明SVF始终优于传统的有效微调策略，例如LoRA，同时参数数量也减少了几个数量级。然后，我们证明Transformer²能够进一步提升性能，即使在完全超出分布的应用（例如视觉问答）中也能有效地调整基础模型的权重。最后，我们分析了新框架的特性，验证了它随着对其当前测试时条件的更多访问而提供越来越多的好处，甚至允许跨模型架构循环利用预训练的SVF专家。总而言之，我们的主要技术贡献如下：

- Transformer²作为大型语言模型（LLM）的关键自适应框架的开发，提供了一个通用的蓝图，可以动态地调整大型语言模型从不断增长的预训练技能集中学习到的行为。
- 我们引入了SVF，这是一种新颖的PEFT方法，可在小型数据集上使用强化学习进行训练，产生具有内在组合性的紧凑型专家向量，所有这些关键特性对于我们的可扩展自适应框架都是必要的。

- 在Transformer²中实现了三种自适应策略，有效地调度具有针对不同需求和部署场景而设计的属性的SVF训练专家。

2 相关工作

自适应大型语言模型 我们将自适应大型语言模型定义为一组大型语言模型或一个独立的大型语言模型，它能够评估并修改其行为以响应其运行环境或内部状态的变化，而无需外部干预。这种动态调整与快速权重记忆等概念类似，后者使网络能够根据任务需求更新权重 (Schmidhuber, 1992; Gomez & Schmidhuber, 2005)，并将神经网络权重视为动态程序 (Schmidhuber, 2015)。最近，Panigrahi等人 (2023)提出了一种方法，其中一个较小的辅助转换器在较大的模型中动态更新，符合自适应行为的原则。

这种适应性可以从两个角度来探讨：宏观视角，多个大型语言模型协作和/或竞争；微观视角，内部适应性允许单个大型语言模型专门处理不同的任务。

Macroview: 从这个角度来看，系统将查询定向到具有特定领域专业知识的大型语言模型 (LLM)，优先考虑专家模型的输出，从而实现更高的准确性和特定任务的优化。这种特定任务的集成可以通过多种机制实现：多个大型语言模型扮演不同的角色并协调朝着共同目标努力 (Zhuge et al., 2023)，参与相互倾听和辩论 (Du et al., 2023)，或使用精心设计的提示构建 (Zhang et al., 2024) 来整合知识库和技能规划。自然地，集成中各个大型语言模型专业化和自适应能力的提高会增强集体性能。因此，在本文中，我们关注自适应大型语言模型的微观视角。

Microview: 从这个角度来看，MoE在大型语言模型中扮演着关键角色 (Tianlong等人, 2024)。在MoE系统中，输入被动态路由到包含特定领域知识的专业模块或层子集 (例如，MLP) (Rajbhandari等人, 2022; Fedus等人, 2022)。为了减少推理时间，研究人员引入了稀疏激活的MoE，其中每个token只选择专家子集 (Jiang等人, 2024; Qwen团队, 2024)。虽然可以将Transformer²松散地视为一种MoE，但存在两个主要区别。在上述系统中，自适应是通过token级别的路由实现的，而Transformer²则采用样本级别的模块选择策略。第二个区别在于专家模块的构建。在传统的MoE系统中，专家模块要么从头开始训练 (Fedus等人, 2022; Jiang等人, 2024)，要么是密集模型 (例如，升级循环) (Qwen团队, 2024; Zhu等人, 2024)，没有辅助损失来确保模块专业化。相比之下，Transformer²专门使用强化学习训练专家向量以获取特定领域知识，使其成为真正的专家。

诸如LoRA (Hu等人, 2021) 之类的低秩自适应PEFT方法通过冻结原始模型的参数并引入小的可训练低秩矩阵来进行特定任务的更新。它显著降低了计算和内存成本，同时提供了与完全微调相当的性能。受LoRA设计的启发，人们提出了各种改进 (Zhang等人, 2023; Kopiczko等人, 2023; Liu等人, 2024; Ba azy等人, 2024; Cetoli, 2024; ?)。Transformer²不依赖于低秩矩阵，而是缩放原始参数矩阵的跨越全秩空间的奇异向量。

用于LLM微调的SVD SVD越来越多地被用作LLM中PEFT的归纳偏置。例如，Wang等人 (2024) 分解权重矩阵，并使用与噪声或长尾信息相关的次要奇异分量来初始化用于LoRA微调的低秩矩阵。早期的工作提出使用诸如DCT系数之类的压缩形式来生成神经网络中的权重矩阵 (Koutnik等人, 2010)，这在内存受限的环境中提供了效率，这与我们的方法相呼应。类似地，SVD用于用前 r 个奇异向量 (对应于最高的奇异值) 来逼近原始权重矩阵。然后在截断的奇异值矩阵之上引入一个小型的可训练矩阵，以调整此前 r 个子空间内的幅度和方向 (Ba azy等人, 2024; Cetoli, 2024)。然而，这种方法的缺点是仅保留前几个奇异分量可能会导致重要信息的丢失，尤其是在奇异值分布不太倾斜的情况下。

与我们的工作最相似的研究是Lingam等人（2024年）的并发工作，他们引入了各种利用权重SVD的稀疏化方法。然而，它并非针对自适应LLM，也没有使用强化学习来提高学习效率。

三种方法

3.1 预备知识

奇异值分解 (SVD) 提供了矩阵乘法的基本视角。在神经网络的背景下，每个权重矩阵 $W \in \mathbb{R}^{n \times m}$ 可以分解成三个组成部分 $W = U\Sigma V^\top$ ，产生半正交矩阵 $U \in \mathbb{R}^{m \times r}$ 和 $V \in \mathbb{R}^{n \times r}$ 以及一个按降序排列的 r 奇异值向量（降序排列），这些奇异值排列在对角矩阵 $\Sigma \in \mathbb{R}^{r \times r}$ 中。由将 W 应用于 x 定义的线性运算，可以分解为独立项的总和，这些独立项源于将 V 中的每一列 v_i 映射到 U 中的对应列 u_i ，如 $y = \sum_{i=1}^r \sigma_i u_i v_i^\top x$ 所示。因此，由秩为 1 的矩阵 $u_i v_i^\top$ 表示的每个奇异分量独立地处理输入，为层的输出提供正交贡献，奇异值 σ_i 调节贡献的程度。

交叉熵方法 (CEM) 是一种用于重要性采样和优化的蒙特卡洛方法 (Rubinstein & Kroese, 2004)。该方法基于最小化两个概率分布 $D_{\text{KL}}(P\|Q)$ 之间KL散度的概念，其中 P 是目标分布， Q 是保持分布。其核心是，CEM重复地从 Q 生成一组样本，用性能函数评估这些样本，然后用性能最好的精英样本的特征更新分布 Q 。在大多数应用中使用的标准设置中， Q 设置为对角多元高斯分布，从而简化了问题，只需估计最新精英样本的经验均值和标准差，直到满足停止条件。我们在下面的Python伪代码中说明了一个完整的CEM步骤。

```
def cem_step(mu, sigma, num_elites, num_samples):
    samples = np.random.normal(loc=mu, scale=sigma, size=num_samples)
    scores = evaluate(samples)
    elites = samples[np.argsort(scores)[-num_elites:]]
    new_mu = np.mean(elites, axis=0)
    new_sigma = np.std(elites, axis=0)
    return (new_mu, new_sigma)
```

3.2 Transformer²

Transformer²的构建包含两个主要步骤，我们在图2中提供了说明性概述。首先，我们介绍奇异值微调 (SVF)，这是一种基于基础模型权重的SVD学习紧凑且 *compositional* 专业的向量的方法。然后，我们描述了Transformer²中的三种不同的适应策略，这些策略受到三个正交原则的启发，它们在推理过程中自适应地组合SVF训练的专家向量。我们阐述了SVF的特性如何与我们的适应策略高度互补，从而使Transformer²成为设计新型自适应LLM的有效且可扩展的框架。

奇异值微调是Transformer²中的一个关键组成部分。它为微调提供了极其高效的参数化方法，并为适应性提供了内在的组性。传统的微调技术通常旨在通过修改预训练模型的权重矩阵来增强其新能力。然而，在大规模Transformer中，由于预训练数据的广度和扩展的架构设计，这些权重已经是抽象知识的丰富存储库。事实上，正如许多之前的文献所证明的那样，解决许多下游任务所需的技能似乎已经存在于这些预训练模型中 (Sharma等人, 2023)。因此，有效的微调方法应该关注如何使这些潜在能力更具表达力，而不是寻求添加新功能。基于这些考虑，对于任何权重矩阵 W ，SVF学习一个简单的向量 $z \in \mathbb{R}^r$ ，它对 W 的每个奇异分量进行有针对性的独立修改，从而产生一个新的权重矩阵 $W' = U\Sigma'V^\top$ ，其中 $\Sigma' = \Sigma \otimes \text{diag}(z)$ 。这种基本的参数化方法具有多种优势：

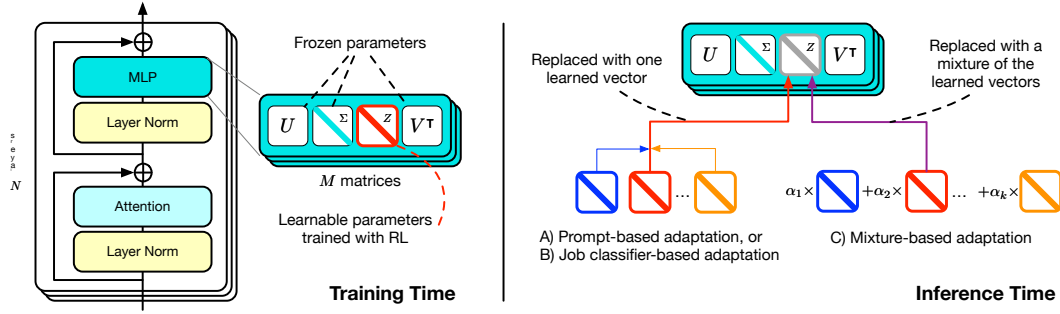


图2: 方法概述。左) 在训练时, 我们采用SVF和RL来学习缩放权重矩阵奇异值的“专家”向量 $\{v^*\}$ 。右) 在推理时, 我们提出三种不同的方法来自适应地选择/组合学习到的专家向量。

Negligible parameters: 仅为每个权重矩阵学习一个向量 z , 即使与专门为提高效率而设计的先前方法相比, 也能实现非常高效的微调, 优化参数的数量减少了几个数量级。例如, 广受欢迎的LoRA方法每个权重矩阵需要 $(m+n) \times r'$ 个可学习参数, 其中 r' 是一个超参数, 通常需要设置得足够大以保证表达能力。虽然最近的扩展, 例如LoRA-XS (Bazy等人, 2024年), 试图进一步提高效率, 但它们通常会引入限制性假设, 从而限制了在几种实际场景中的适用性 (参见附录C中的示例)。相比之下, 虽然SVF只需要 $r = \min(m, n)$ 个参数, 但我们通过实验证明, 它并没有表现出同样的缺点, 这要归功于它在一个高度有意义的空间中工作, 该空间由现代LLM权重中压缩的潜在表达能力提供。SVF仅缩放奇异值似乎会导致表达能力有限, 我们想指出, 以满秩方式影响权重矩阵的能力实际上比低秩方法提供了更多信息。

High compositionality: 将权重分解为独立的奇异分量, 使得学习到的 $\{v^*\}$ 向量具有高度的可组合性和可解释性, 为通过代数运算进行适应提供了许多可能性。相反, 基于LoRA的方法本质上缺乏这些特性。例如, 即使是在同一任务上学习的两个LoRA模型对每个 $\{v^*\}$ 都学习到完全相同的调整, 直接在其压缩的 $\{v^*\}$ 和 $\{v^*\}$ 矩阵之间进行插值也不太可能保留其任何原始行为, 因为它们可能收敛到无数个等效的参数排列。

Principled regularization: 仅修改现有奇异分量的幅度提供了一种有原则且有效的正则化形式。在实践中, 此属性使我们能够仅用数百个数据点对任意下游任务进行微调, 而无需担心严重崩溃或过拟合。

使用强化学习进行端到端优化。我们训练一组SVF向量 $\theta_z = \{z_1, \dots, z_{N \times M}\}$ 来微调由 θ_W 参数化的任意语言模型 π_{θ_W} , 直接针对任务性能进行强化学习优化。这里, $\theta_W = \{W_1, \dots, W_{N \times M}\}$ 是一组权重矩阵, 其中 N 是层数, M 是每层要微调的权重矩阵数量。我们使用开创性的REINFORCE算法 (Williams, 1992), 并根据其正确性 $r \in \{-1, 1\}$ 为提示 $x_i \in D$ 生成的每个答案 y_i (赋予单位奖励)。受强化学习优化大型语言模型相关应用的启发 (Ouyang et al., 2022), 我们通过添加一个KL惩罚项来正则化REINFORCE目标函数, 该惩罚项用于惩罚偏离原始模型行为的情况, 并由一个小系数 $\lambda \in \mathbb{R}^+$ 加权。因此, 我们的最终目标函数可以写成:

$$J(\theta_z) = \mathbb{E} [\log (\pi_{\theta_{W'}}(\hat{y}_i | x_i)) r(\hat{y}_i, y_i)] - \lambda D_{\text{KL}}(\pi_{\theta_{W'}} \| \pi_{\theta_W}), \quad (1)$$

其中我们使用 $\pi_{\theta_{W'}}$ 表示用 W' 替换原始权重矩阵 W 后得到的语言模型。虽然强化学习通常被认为比下一个token预测目标更不稳定, 但我们发现SVF的正则化特性避免了许多先前约束较少参数化的失败模式 (参见第4.3节)。因此, 有效地结合这些互补组件使我们能够避免依赖于昂贵的微调程序以及大型手工设计的数据集作为代理, 并直接端到端地最大化任务性能。

总的来说，使用强化学习的SVF对训练数据集的要求较低。例如，LoRA微调需要“解释性文本”来进行下一个token的预测，这对数据集提出了更高的要求（例如，想象一下在GSM 8K数据集上进行LoRA微调，其中没有推理文本，只有最终的数字）。这一优势使得SVF更加通用和有效。SVF可能面临的一个潜在问题是弱基模型导致的稀疏奖励，我们将在第5节中进一步讨论这个问题。

自适应是自然界中一种关键机制，它已成为现代系统设计中的核心指导原则（Kl s等人，2015年）。我们最初致力于自适应基础模型的工作重点是LLM的推理阶段，我们设计了一种简单的两遍自适应策略，该策略结合了 K 组使用SVF训练的基“专家”向量 $z^{1:K}$ ，以提供不同类型的功能（例如，编码、数学等）。功能与我们训练的数据集之间的映射可以在数据集的元数据中获取。在第一次推理过程中，给定一个任务或单个输入提示，Transformer²执行模型并观察其测试时行为以导出一个针对其测试时条件定制的新 z' 向量。然后在第二次推理过程中使用此自适应的 z' 向量，以使用新适应的权重提供实际响应。SVF训练的专家向量与自适应策略之间的交互确保了无缝集成，其中专家向量提供模块化功能，而自适应策略动态地确定和组合最合适的组合以解决输入任务。在这项初步工作中，我们提出了三种简单的方法来生成第一次推理过程中的向量 z' ，使用不同的方法和要求实现自适应。下面，我们概述了每种方法，并在附录A中提供了其他实现细节。

A) Prompt engineering: 我们最基本的方法涉及构建一个新的“适应”提示，我们用它来直接ask大型语言模型对输入提示进行分类。根据其响应，我们从用于预训练每个SVF专家的领域主题集中提取一个类别，因此，我们直接从 $z^{1:K}$ 中选择相应的 z' 。在我们的适应提示中，我们还明确提供了通用的“其他”类别选项，允许模型在没有专家提供适当能力的情况下使用其基础权重。我们在图3中展示了用于构建适应提示的格式。

B) Classification expert: 提示工程方法的直接扩展是使用专门的系统来处理任务识别。遵循自适应原则，我们应用SVF来微调基础LLM本身以处理此任务。特别是，我们从 K SVF训练任务中收集了一个数据集 $D = \{(x_{1,1}, 1), \dots, (x_{i,k}, k), \dots\}$ ，其中 $x_{i,k}$ 是来自 k 专家任务的第 i 个示例。每个元组 $(x_{i,k}, k)$ 然后形成一个示例，用于预训练另一个作业分类专家 z^c ，其学习方式与其他专家相同。在第一次推理过程中，我们只需加载 z^c ，旨在提高基础模型固有的任务分类能力，以选择更合适的 z' 来处理输入提示。

Analyze the given question and classify it into one of four categories: 'code', 'math', 'reasoning', or 'others'. Follow these guidelines:

1. Code: Questions asking for programming solutions...
2. Math: Questions involving mathematical calculations...
3. Reasoning: Questions requiring logical thinking...
4. Others: Questions not clearly fit into above categories...

Instructions:

- Consider the primary focus, skills, and knowledge required to answer the question.
- If a question spans multiple categories, choose the most dominant one.
- Provide your final classification within `\boxed{}` notation. Example: `\boxed{reasoning}`

Format your response as follows:
Classification: `\boxed{category}`

图3：基于提示的自适应。Transformer²使用的自适应提示将任务提示分类到预定义类别中。

C) Few-shot adaptation: 我们的第三种方法利用额外的任务信息，假设其测试时间条件超越单个提示的访问权限。我们的方法受到流行的少样本提示技术的启发，这些技术已被证明可以提供一致的性能改进，甚至允许LLM“在上下文中”学习推理之前完全未见过的任务（Brown，2020）。对于每个优化的 W ，我们的方法需要通过线性插值学习到的 K SVF向量来生成一个全新的 $z' = \sum_{k=1}^K \alpha_k z_k$ ，每个向量的权重为系数 α_k 。我们使用CEM搜索每个 α_k 的可能值，基于一组“少样本提示”的性能，这些提示专门从其余测试提示中剔除，并用于评估CEM的总体样本。如果多个总体样本在这些保留的提示上获得相同的得分，我们通过偏向在其自身生成的正确答案中具有最高平均对数似然的样本打破平局。至关重要的是，我们只需要对每个目标任务执行此过程一次，避免需要增加每个的长度

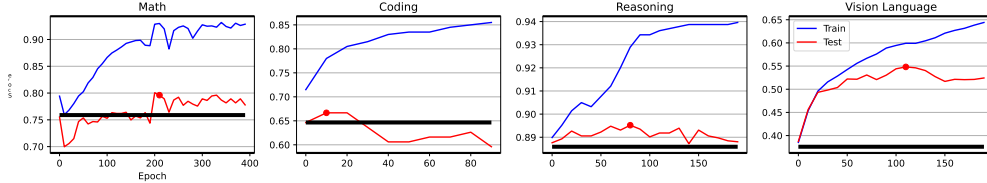


图4: SVF学习曲线。虚线表示LLAMA3-8B-INSTRUCT在每个任务测试集上的性能。SVF有效地微调以超越基线性能。虽然我们使用最佳验证分数来选择用于评估的检查点（用红点标记），但我们展示了整个训练曲线而没有提前停止，以展示SVF的学习能力。像编码和推理这样只有数百个训练样本的任务提前停止了。在我们的实验中，我们在每个epoch结束时更新参数。

问题提示，传统少样本提示的一个相关缺点。更多细节和对此最终方法的扩展讨论，请参见A.4节。

4 个实验

我们广泛评估了Transformer²在多个任务和模型上的性能，目的是：(1)评估SVF的效率和有效性；(2)通过三种提出的适应策略展示自适应性；(3)进行深入的分析 and 消融研究，旨在理解和解释我们新框架的特性。

4.1 实验装置

为了验证Transformer²的普适性，我们考虑了三个来自不同模型族和架构规模的预训练大型语言模型：LLAMA3-8B-INSTRUCT、MISTRAL-7B-INSTRUCT-V0.3和LLAMA3-70B-INSTRUCT。对于每个模型，我们获得三组SVF训练的 z 向量，以最大化其在GSM8K (Cobbe等人, 2021)、MBPP-pro (Austin等人, 2021) 和ARC-Easy (Clark等人, 2018) 上的性能。此外，我们还针对用作TextVQA (Singh等人, 2019) 语言主干的LLAMA3-8B-INSTRUCT训练了一组 z 向量，以评估SVF在视觉语言建模 (VLM) 领域的适用性。我们在图4中提供了SVF在每个任务上的主要学习曲线。最后，我们评估了完整的Transformer²自适应框架在四个未见任务上的性能：MATH (Hendrycks等人, 2021)、Humaneval (Chen等人, 2021)、ARC-Challenge (Clark等人, 2018) 和OKVQA (Marino等人, 2019)。在我们所有的自适应实验中，我们只考虑在纯语言环境中获得的专家，评估其即使在独特的视觉领域中的测试时适用性。更多细节和实验中使用的超参数总结，请参考附录A。

4.2 实验结果

SVF性能 我们在表1中提供了在每个考虑的任务上使用LLAMA3-8B-INSTRUCT、MISTRAL-7B-INSTRUCT-V0.3和LLAMA3-70B-INSTRUCT基础模型训练后的结果。值得注意的是，我们发现SVF在几乎所有任务和基础模型上都提供了相当且一致的性能提升。相反，LoRA专家产生的增益较小，甚至出现零星的性能下降。（这些LoRA专家是使用下一个token预测进行训练的。虽然我们也

表1: 微调结果。大型语言模型在数学、编码和推理测试集上的性能。（括号内为标准化分数）

Method	GSM8K	MBPP-Pro	ARC-Easy
LLAMA3-8B-INSTRUCT	75.89 (1.00)	64.65 (1.00)	88.59 (1.00)
+ LoRA	77.18 (1.02)	67.68 (1.05)	88.97 (1.00)
+ SVF (Ours)	79.15 (1.04)	66.67 (1.03)	89.56 (1.01)
MISTRAL-7B-INSTRUCT-V0.3	42.83 (1.00)	49.50 (1.00)	81.65 (1.00)
+ LoRA	44.66 (1.04)	51.52 (1.04)	81.19 (0.98)
+ SVF (Ours)	49.74 (1.16)	51.52 (1.04)	85.14 (1.04)
LLAMA3-70B-INSTRUCT	85.29 (1.00)	80.81 (1.00)	89.10 (1.00)
+ LoRA	77.26 (0.91)	68.69 (0.85)	88.55 (0.99)
+ SVF (Ours)	88.32 (1.04)	80.81 (1.00)	88.47 (0.99)

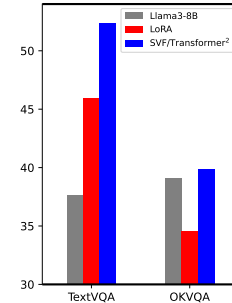


Figure 5: Results for the VLM domain.

表2: 在未见任务上的自适应。标准化分数在括号中。

Method	MATH	HumanEval	ARC-Challenge
LLAMA3-8B-INSTRUCT 3	24.54 (1.00)	60.98 (1.00)	80.63 (1.00)
+ LoRA	24.12 (0.98)	52.44 (0.86)	81.06 (1.01)
+ Transformer ² (Prompt)	25.22 (1.03)	61.59 (1.01)	81.74 (1.01)
+ Transformer ² (Cls-expert)	25.18 (1.03)	62.80 (1.03)	81.37 (1.01)
+ Transformer ² (Few-shot)	25.47 (1.04)	62.99 (1.03)	82.61 (1.02)
MISTRAL-7B-INSTRUCT-V0.3	13.02 (1.00)	43.29 (1.00)	71.76 (1.00)
+ LoRA	13.16 (1.01)	37.80 (0.87)	75.77 (1.06)
+ Transformer ² (Prompt)	11.86 (0.91)	43.90 (1.01)	72.35 (1.01)
+ Transformer ² (Cls-expert)	11.60 (0.89)	43.90 (1.01)	74.83 (1.04)
+ Transformer ² (Few-shot)	13.39 (1.03)	47.40 (1.09)	75.47 (1.05)
LLAMA3-70B-INSTRUCT	40.64 (1.00)	78.66 (1.00)	87.63 (1.00)
+ LoRA	25.40 (0.62)	73.78 (0.94)	83.70 (0.96)
+ Transformer ² (Prompt)	40.44 (1.00)	79.88 (1.02)	88.48 (1.01)

表4中使用强化学习训练的LoRA专家表明, 与SVF相比, 强化学习在LoRA中的效果较差。这种趋势也扩展到视觉语言领域, 使用SVF微调LLAMA3-LLAVA-NEXT-8B将基础模型的性能提升了39%以上(见图5)。为了确保公平比较, 我们在附录4.3中对我们的模型和LoRA基线进行了广泛的消融研究, 考虑了不同的架构和优化目标。由于其基本的参数化, 我们想指出, 训练SVF所需的资源要少得多, 训练参数不到我们LoRA实现的10%。

使用经过SVF训练的 z 向量, 我们评估了Transformer²在未见任务上的自适应能力。为了与LoRA进行公平比较, 我们使用所考虑训练任务的所有检查点记录此基线的性能, 并且仅报告其在每个测试任务中的最高性能。如表2所示, 我们所有的Transformer²自适应策略都证明了在所有任务中对LLAMA3-8B-INSTRUCT基础模型的改进, 并且在MISTRAL-7B-INSTRUCT-V0.3和LLAMA3-70B-INSTRUCT中至少有两个任务取得了改进。相比之下, 即使是最好的训练LoRA也只在ARC-Challenge任务上提供了微小的改进, 并且在MATH和HumanEval任务上仍然显著降低了性能。这种差异表明, LoRA的参数化和优化可能对过拟合特别敏感, 尤其是在使用较小的GSM8K和MBPP-Pro数据集进行训练时, 这些数据集提供与MATH和HumanEval最相关的信息。在图5中, 我们发现OKVQA任务中存在类似的两分法, LLAMA3-LLAVA-NEXT-8B VLM基础模型的性能只有在应用Transformer²后才得到改进。我们注意到, 在这种情况下, Transformer²也仅从GSM8K、MBPP-Pro和ARC-Easy的专家向量中进行自适应。因此, 这一结果进一步强调了自适应的高度灵活性, 即使对于完全基于语言的任务, 也能将压缩的知识转移到不相关的基于视觉的问题上。

比较这三种提出的适应策略, 我们发现了一个清晰的单调趋势——随着策略的复杂程度增加以及关于测试时间条件的额外信息增多, 自适应似乎越来越有效。特别是, 采用少量样本自适应的Transformer²几乎总是得分最高的模型, 在所有测试环境中都提供了显著的改进, 除了LLAMA3-70B-INSTRUCT @MATH, 由于GPU资源有限, 我们只对其中一半的层进行了SVF微调。这一趋势表明, 提供额外或不同类型的信息对我们的框架非常有益, 这表明Transformer²可以为基础模型提供新的方法, 使其在终身学习环境中持续改进性能。

表3报告了Transformer²的提示适应策略所需的推理时间, 其中分别列出了第一遍和第二遍解决整个问题集所花费的时间。注意, 第二遍推理时间是解决问题所花费的时间, 第一遍推理时间是自适应时间, 第一遍到第二遍推理时间的比率在括号中。虽然额外的推理过程似乎会使总运行时间加倍, 但这很重要

表3: Transformer²提示调整策略中针对整个问题集的二遍推理时间成本。括号中显示的是第一遍到第二遍推理时间的比率。

Task	1st (s)	2nd (s)
MATH	42.64 (13%)	321.19
HumanEval	2.76 (19%)	14.28
ARC-Challenge	13.40 (47%)	28.51

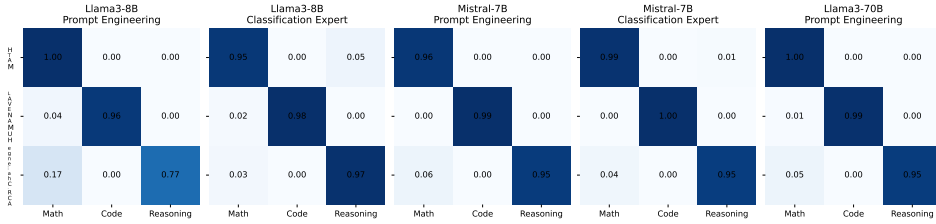


图6: 混淆矩阵。这些矩阵显示分类百分比，其中行代表任务类别（真实值），列表示预测类别。一些样本被错误分类为“其他”，这反映在总和不等于一的行中。

需要注意的是，推理时间主要取决于生成的标记数量。在我们的设置中，它是 $\mathcal{O}(n)$ ，其中 n 是输入的长度。ARC挑战的成本比率很大，因为它们是单选题，因此第二次传递的成本也是 $\mathcal{O}(n)$ 。在一般情况下，我们认为将此比率假设为更接近MATH和Humaneval的比率是合理的。有关改进CEM少样本适应方法效率的详细讨论，请参见附录D

4.3 分析

最后，我们分析并讨论了我们的适应策略的特性，我们在附录B中提供了扩展和进一步的讨论。

分析1: 任务调度准确性 图6给出了我们基于分类的适应策略的混淆矩阵。这些结果验证了我们基于分类的适应策略的有效性，该策略将每个提示与在相似领域接受过训练的专家匹配，这从对角线上的高值可以看出。此外，LLAMA3-8B-INSTRUCT和MISTRAL-7B-INSTRUCT-V0.3的结果也表明，使用分类专家始终比普通的提示工程提供更高的分类准确性。虽然这种差异可以解释相对自适应策略的更高性能，但我们也注意到，领域相似性可能不是确定每个提示或任务最佳专家的唯一相关指标。为此，我们认为在未来的工作中可以探索许多尚未探索的扩展，例如使用过去的专家表现或令牌级分析来进一步提高我们框架的可扩展性。

分析2: 训练任务适配贡献 图7显示了在所有未见的下游任务中，插值于我们通过CEM为LLAMA3-8B-INSTRUCT和MISTRAL-7B-INSTRUCT-V0.3学习的SVF向量之间的归一化自适应系数 a_k 。直观地，我们发现来自与未见任务具有相似主题的训练任务的专家向量往往是对生成的适应性权重贡献最大的。然而，我们观察到MATH任务是一个有趣的例外，因为从GSM8K训练中获得专家的 a_k 在两个模型中实际上都是三个中最低的。我们假设这反映了MATH中的数学竞赛问题与GSM8K中的小学问题性质的不同。事实上，MATH问题的难度不仅远超GSM8K，而且其大部分问题也主要依赖于逻辑推理，而像ARC这样的任务可能更贴合。此外，我们还注意到，不同的 z 向量似乎对Llama模型的适应性贡献更均匀。这种差异可能表明，由于其更高的基准性能，Llama模型不需要像Mistral那样依赖任何特定的技能集，并且可以从自适应中获得更全面的好处。需要注意的是，均匀地应用 a_k 并不是利用专家向量的通用解决方案。当我们查看不同的模型和任务组合时，这一点就变得很明显（例如，在LLAMA3-8B-INSTRUCT上对MATH任务均匀应用 a_k 仅达到24.47，而Transformer² (Few-shot)达到25.47）。

分析 3: 消融研究

Module sensitivity: 我们首先比较SVF应用于不同模块时的性能（参见试验1-3）。在一致条件下，单独的MLP和注意力更新都能提高性能，其中MLP更新带来的提升更为显著。同时更新这两种模块类型会带来更大的改进。

Objective function: 我们关注不同目标函数对性能的影响，并将强化学习目标函数与下一个token预测损失进行比较（参见试验2和4）。对于后者，我们使用官方GSM8K解决方案作为目标token进行指令微调。结果表明，强化学习带来了明显的性能提升，证明了其在特定任务微调中的有效性。相反，下一个token预测甚至会阻碍性能。这突出了强化学习处理缺乏详细解决方案的情况的能力，表明其在这种情况下具有优越性。

SVF vs LoRA: 最后，我们还使用 RL 目标评估了 LoRA（参见试验 2 和 5）。观察到显著的性能差异，这主要归因于 LoRA 训练过程的严重不稳定性。尽管探索了广泛的学习率，但 LoRA 的性能始终落后。更多说明，请参见附录中的图 9。

表4：消融研究。我们使用不同的设置在GSM8K训练集上微调LLAMA3-8B-INSTRUCT，并在测试集上给出结果，以及在MATH上的零样本迁移结果。

#	Method	Objective Function	Module	#Params (\downarrow)	GSM8K (\uparrow)	MATH (\uparrow)
0			LLAMA-3-8B-INSTRUCT		75.89 (1.00)	24.54 (1.00)
1	SVF	Policy gradient	MLP	0.39M	78.62 (1.04)	24.20 (0.99)
2	SVF	Policy gradient	attention	0.16M	76.19 (1.00)	24.20 (0.99)
3	SVF	Policy gradient	MLP + attention	0.58M	79.23 (1.04)	25.04 (1.04)
4	SVF	Next token pred	attention	0.16M	60.50 (0.80)	18.52 (0.75)
5	LoRA	Policy gradient	attention	6.82M	57.92 (0.76)	15.72 (0.64)
6	LoRA	Next token pred	attention	6.82M	77.18 (0.98)	24.12 (0.96)
7	LoRA	Next token pred	MLP + attention	35.13M	75.66 (0.96)	22.12 (0.91)

分析4：跨模型兼容性 最后，我们探索了将我们的自适应框架应用于*across different LLMs*的潜力。特别是，我们评估了在LLAMA3-8B-INSTRUCT上训练的SVF专家向量是否能使MISTRAL-7B-INSTRUCT-V0.3受益，以及我们能否在这两个模型的专家向量之间进行自适应。我们在表5中展示了我们的主要发现，并在附录B中提供了更详细的结果。令人惊讶的是，我们发现这两个模型之间发生了正迁移，在3个任务中的2个任务中都有明显的益处。我们注意到这些改进是由于SVF参数化的固有顺序造成的，因为在将*randomly shuffling*每个SVF向量应用于Mistral模型之前，都会一致地降低性能。

此操作导致每个任务的性能都显著下降。最后，通过使用从两个模型收集的SVF向量进行少样本适应，MISTRAL-7B-INSTRUCT-V0.3的性能在各方面都有所提高。我们观察到，这些增益甚至超过了表2中报告的，使用*all* SVF向量对MISTRAL-7B-INSTRUCT-V0.3进行ARC-Challenge任务适应所获得的最佳分数。虽然这些结果看起来很有前景，但我们注意到，通过我们简单的迁移方法发现的令人惊讶的兼容性可能与所考虑的两个大型语言模型的架构相似性有关。为此，是否可以将类似的迁移复制到不同规模的模型中仍然是一个悬而未决的研究问题，这可能会为解开和循环利用针对较新/较大模型的任务特定技能打开大门，这对民主化和可持续性具有重要意义。

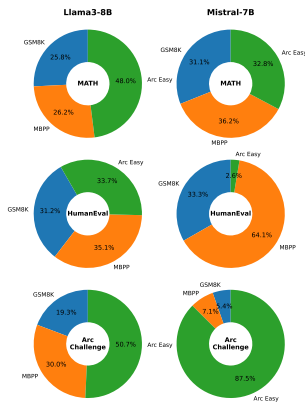


图7：学习到的权重 α_{k0}

5 结论

本文介绍了Transformer²，提供了一种实现自适应大型语言模型的新蓝图。在这个框架内，我们首先提出了SVF，它比之前的微调方法具有更好的性能，同时降低了成本，提高了组合性，并具有过拟合正则化功能——所有这些都是实现可扩展自适应的关键属性。利用一组SVF专家作为构建块，我们开发了三种有效的自适应策略，每种策略都具有独特的优势，并且随着测试时条件访问的增加，性能也单调递增。

虽然Transformer²展现出令人鼓舞的结果，但未来仍有令人兴奋的研究机会。一个局限性是SVF专家的能力与……的潜在成分相关联

表5: 跨模型 z 向量迁移。将LLAMA3-8B-INSTRUCT上训练的专家向量迁移到MISTRAL-7B-INSTRUCT-V0.3的结果, 使用了跨模型小样本适应方法。

Method <i>SVF training task</i>	MATH <i>GSM8K</i>	Humaneval <i>MBPP-pro</i>	ARC-Challenge <i>ARC-Easy</i>
MISTRAL-7B-INSTRUCT-V0.3	13.02 (1.00)	43.29 (1.00)	71.76 (1.00)
+ Llama SVF (ordered σ_i)	11.96 (0.92)	45.12 (1.04)	72.01 (1.00)
+ Llama SVF (shuffled σ_i)	10.52 (0.81)	40.24 (0.93)	70.82 (0.99)
+ Few-shot adaptation (cross-model)	12.65 (0.97)	46.75 (1.08)	75.64 (1.05)

基础模型。为了解决这个问题, 模型合并提供了一个有前景的方向 (Yu等人, 2024; Goddard等人, 2024; Akiba等人, 2024), 使专业模型能够组合成一个更强大的单一模型。此外, 虽然我们基于CEM的适应方法有效地平衡了性能和效率, 但扩展到大量专业领域可能会增加一次性计算成本。然而, 这种权衡可以通过改进的性能和增强的自适应能力的好处来抵消。模型合并和高效适应技术的进步已经产生了在公开排行榜上占据主导地位的模式, 这使得它们成为Transformer²强大基础模型的候选者, 并为自适应LLM开辟了新的可能性。

参考文献 秋庭拓哉, 新井誠, 唐宇津, 孙琦, David Ha. 模型合并配方的进化优化。
arXiv preprint arXiv:2403.13187, 2024年。Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, 等人。使用大型语言模型进行程序合成。*arXiv preprint arXiv:2108.07732*, 2021年。
 Klaudia Ba azy, Mohammadreza Banaei, Karl Aberer和Jacek Tabor。Lora-xs: 具有极少量参数的低秩自适应。*arXiv preprint arXiv:2405.17604*, 2024年。Tom B Brown。语言模型是少样本学习器。*arXiv preprint arXiv:2005.14165*, 2020年。Alberto Cetoli。使用奇异值分解微调LLM。Hugging Face 博客, 2024年6月。网址<https://huggingface.co/blog/fractalego/svd-training>。访问时间: 2024年7月1日。Mark Chen, Jerry Tworek, Heewoo Jun, 袁启明, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, 等人。评估在代码上训练的大型语言模型。*arXiv preprint arXiv:2107.03374*, 2021年。Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick和Oyvind Tafjord。你认为你解决了问答问题吗? 试试ARC, AI2推理挑战赛。*arXiv preprint arXiv:1803.05457*, 2018年。Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, 等人。训练验证器来解决数学文字题。*arXiv preprint arXiv:2110.14168*, 2021年。Elizabeth N Davison, Kimberly J Schlesinger, Danielle S Bassett, Mary-Ellen Lynall, Michael B Miller, Scott T Grafton和Jean M Carlson。跨任务状态的大脑网络适应性。*PLoS computational biology*, 11(1):e1004029, 2015年。杜逸伦, 李爽, Antonio Torralba, Joshua B Tenenbaum和Igor Mordatch。通过多智能体辩论改进语言模型的事实性和推理能力。*arXiv preprint arXiv:2305.14325*, 2023年。William Fedus, Barret Zoph和Noam Shazeer。Switch Transformers: 通过简单高效的稀疏性扩展到万亿参数模型。*Journal of Machine Learning Research*, 23(120):1–39, 2022年。Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vlad Karpukhin, Brian Benedict, Mark McQuade和Jacob Solawetz。Arcee的MergeKit: 一个用于合并大型语言模型的工具包。*arXiv preprint arXiv:2403.13257*, 2024年。Faustino Gomez和Jürgen Schmidhuber。进化模块化快速权重网络用于控制。在*International Conference on Artificial Neural Networks*, 第383-389页。Springer, 2005年。David Ha, Andrew M. Dai和Quoc V. Le。超网络。在*International Conference on Learning Representations*, 2017年。网址<https://openreview.net/forum?id=rkpACe1lx>。Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song和Jacob Steinhardt。使用数学数据集测量数学问题解决能力。*arXiv preprint arXiv:2103.03874*, 2021年。Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, 李元智, 王杉, 王璐和陈伟洲。Lora: 大型语言模型的低秩自适应。*arXiv preprint arXiv:2106.09685*, 2021年。Kazuki Irie, Imanol Schlag, Robert Csordás和Jürgen Schmidhuber。一个现代的自指权重矩阵, 它学习修改自身。在*International Conference on Machine Learning*, 第9660-9677页。PMLR, 2022年。

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, 等人。专家混合模型。 *arXiv preprint arXiv:2401.04088*, 2024年。Junmo Kang, Leonid Karlinsky, Hongyin Luo, Zhen Wang, Jacob Hansen, James Glass, David Cox, Rameswar Panda, Rogerio Feris和Alan Ritter。Self-MoE: 迈向具有自我专业化专家的组式大型语言模型。 *arXiv preprint arXiv:2406.12034*, 2024年。Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu和Dario Amodei。神经语言模型的缩放定律。 *arXiv preprint arXiv:2001.08361*, 2020年。Prakhar Kaushik, Ankit Vaidya, Alan Yuille, 等人。EigenLoRA: 回收训练的适配器以实现资源高效的适应和推理, 2025年。网址: <https://openreview.net/forum?id=KxGGZag9gW>。Verena K l s, Thomas G thel和Sabine Glesner。自适应系统设计中的自适应知识库。见 *2015 41st Euromicro Conference on Software Engineering and Advanced Applications*, 第472-478页, 2015年。doi: 10.1109/SEAA.2015.48。Dawid Jan Kopiczko, Tijmen Blankevoort和Yu ki Markus Asano。Vera: 基于向量的随机矩阵自适应。 *arXiv preprint arXiv:2310.11454*, 2023年。Jan Koutnik, Faustino Gomez和J rgen Schmidhuber。在压缩权重空间中进化神经网络。见 *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 第619-626页, 2010年。Vijay Lingam, Atula Tejaswi, Aditya Vavre, Aneesh Shetty, Gautham Krishna Gudur, Joydeep Ghosh, Alex Dimakis, Eunsol Choi, Aleksandar Bojchevski和Sujay Sanghavi。SVFT: 使用奇异向量进行参数高效微调。 *arXiv preprint arXiv:2405.19597*, 2024年。Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal和Colin A Raffel。少样本参数高效微调优于上下文学习且成本更低。 *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022年。Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Chen和Min-Hung Chen。Dora: 权重分解低秩自适应。 *arXiv preprint arXiv:2402.09353*, 2024年。Lasse S Loose, David Wisniewski, Marco Rusconi, Thomas Goschke和John-Dylan Haynes。额叶和顶叶皮层中与开关无关的任务表征。 *Journal of Neuroscience*, 37(33):8033–8042, 2017年。Kenneth Marino, Mohammad Rastegari, Ali Farhadi和Roohbeh Mottaghi。OK-VQA: 一个需要外部知识的视觉问答基准。见 *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, 第3195-3204页, 2019年。Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, 等人。训练语言模型以遵循带有用户反馈的指令。 *Advances in neural information processing systems*, 35: 27730–27744, 2022年。Abhishek Panigrahi, Sadhika Malladi, Mengzhou Xia和Sanjeev Arora。可训练的Transformer in Transformer。 *arXiv preprint arXiv:2307.01189*, 2023年。Qwen团队。Qwen1.5-MoE: 用1/3激活参数匹配7B模型性能, 2024年3月。网址: <https://qwenlm.github.io/blog/qwen-moe/>。博客文章。Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley和Yuxiong He。DeepSpeed-MoE: 推进混合专家推理和训练, 以支持下一代AI规模。见 *International conference on machine learning*, 第18332-18346页。PMLR, 2022年。

鲁文·Y·鲁宾斯坦和迪尔克·P·克罗塞。 *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning*, 第133卷。施普林格出版社, 2004年。Jürgen Schmidhuber。学习控制快速权重记忆: 动态循环网络的替代方法。 *Neural Computation*, 4(1):131–139, 1992。Jürgen Schmidhuber。一个“自指”权重矩阵。见 *ICANN'93: Proceedings of the International Conference on Artificial Neural Networks Amsterdam, The Netherlands 13–16 September 1993* 3, 第446–450页。施普林格出版社, 1993年。

Jürgen Schmidhuber。关于学习思考: 用于强化学习控制器和循环神经网络世界模型的新组合的算法信息论。 *arXiv preprint arXiv:1511.09249*, 2015。

Pratyusha Sharma, Jordan T Ash和Dipendra Misra。真相就在其中: 通过层选择性秩约简改进语言模型的推理能力。 *arXiv preprint arXiv:2312.13558*, 2023年。

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh和Marcus Rohrbach。迈向能够阅读的VQA模型。见 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 第8317–8326页, 2019年。

Kenneth O Stanley, David B D' Ambrosio和Jason Gauci。一种基于超立方体的用于进化大型神经网络的编码方法。 *Artificial life*, 15(2):185–212, 2009。

陈天龙, 程宇, 陈北迪, 张敏佳和班萨尔·莫希特。大型语言模型时代的专家混合模型: 一段新的旅程。ICML 2024 演示文稿, 2024年。国际机器学习大会 (ICML)。

王汉卿, 肖泽冠, 李亦霞, 王硕, 陈冠华, 陈云。Milora: 利用微小单一成分进行参数高效LLM微调。 *arXiv preprint arXiv:2406.09044*, 2024。

Ronald J Williams。用于连接主义强化学习的简单统计梯度下降算法。 *Machine learning*, 8:229–256, 1992。

于乐, 于 Bowen, 于海阳, 黄飞, 李永斌。语言模型是超级马里奥: 将同源模型的能力作为免费午餐吸收。见 *Forty-first International Conference on Machine Learning*, 2024年。

张策耀, 杨凯杰, 胡思怡, 王子豪, 李广河, 孙逸航, 张程, 张昭伟, 刘安吉, 朱松纯, 等。Proagent: 利用大型语言模型构建主动协作代理。载于 *Proceedings of the AAAI Conference on Artificial Intelligence*, 卷 38, 第 17591–17599 页, 2024年。

张清如, 陈敏硕, Alexander Bukharin, Nikos Karampatziakis, 何鹏程, 程宇, 陈伟洲, 赵tu o。Adalora: 自适应参数高效微调的预算分配。 *arXiv preprint arXiv:2303.10512*, 2023。

童舟, 曲晓晔, 董大泽, 阮嘉诚, 童景琦, 何聪慧, 程宇。Llama-moe: 基于Llama的持续预训练构建专家混合模型。 *arXiv preprint arXiv:2406.16554*, 2024。

朱明晨, 刘浩哲, 弗朗切斯科·法奇奥, 迪伦·R·阿什利, Róbert Csordás, 阿南德·戈帕拉克里希南, 阿卜杜拉·哈姆迪, 哈桑·阿贝德·阿尔·卡德尔·哈穆德, 文森特·赫尔曼, 入江一树, 等。基于自然语言的思维社会中的风暴。 *arXiv preprint arXiv:2305.17066*, 2023。

A 实现细节和超参数

A.1 SVF训练

我们通过使用一致的配方对所有考虑的训练任务和语言模型进行SVF微调训练，获得专家向量 z 作为Transformer²中的基础组件。我们将每个数据集划分为大小相等的训练集和验证集。然后，我们应用基于强化学习的方法，使用AdamW优化 θ_z ，学习率为 2×10^{-3} ，采用余弦衰减，批量大小为256，并进行梯度裁剪。我们采用提前停止策略，并根据验证性能选择最佳 λ (KL散度项系数)。对于LLAMA3-70B-INSTRUCT和视觉任务实验，我们将SVF应用于一半的层以减少内存使用，同时保持相当大的性能提升。在LLAMA3-8B-INSTRUCT在视觉语言任务上的训练过程中，我们应用一个小负奖励 (-0.1) 以提高训练稳定性。

A.2 LORA训练

我们遵循社区最佳实践进行LoRA微调，将其应用于查询和值投影层，学习率约为 5×10^{-5} 。我们设置了总共200次迭代，全局批量大小为256，以确保足够的训练。为了可行的LoRA指令训练，我们从官方来源收集所有训练任务 (GSM8K、MBPP、Arc-Easy、Text VQA) 的解决方案，并将它们添加到问题提示中。表8显示了用于LoRA微调的示例数学问题。尽管进行了大量的超参数调整，但我们经常观察到测试性能下降，正如所讨论的，这

Below is an instruction that describes a task. Write a response that appropriately completes the request.

Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Natalia sold $48/2 = <<48/2=24>>24$ clips in May. Natalia sold $48+24 = <<48+24=72>>72$ clips altogether in April and May. ##### 72

图8：示例问题及答案。用于LoRA指令微调的数学数据样本，蓝色文本为未掩码的解决方案。

这可以归因于训练样本数量少以及指令微调数据（特别是高度详细的思维过程）的潜在模型需求。

A.3 超参数

表6总结了我们在实验中使用的超参数。为了优化性能，我们对多个超参数进行了扫描，并根据验证结果选择了最有效的组合。对于SVF，我们的主要关注点是调整KL系数以增强训练稳定性。对于LoRA，我们专注于扫描学习率和最大梯度裁剪范数以识别最佳设置。

A.4 少样本自适应

如正文所述，我们的少样本适应方法需要为每个 W 生成一个全新的 $z' = \sum_{k=1}^K \alpha_k z_k$ ，方法是在学习到的SVF向量 K 之间进行线性插值，每个向量的权重为系数 $\alpha \in \mathbb{R}^K$ 。我们采用CEM来搜索 α_k ，其依据是在少样本提示上的性能，这些提示专门从其余测试提示中保留出来，并用于在每次迭代中获得精英集。如果多个样本解决方案在这些保留样本上获得相同的得分，我们通过选择在生成的正确答案的标记中具有最高平均对数似然的样本解决方案来打破平局。

在所有主要实验中，我们仅保留10个数据样本用于自适应，并执行最多100次CEM迭代。对于每种设置，我们都考虑逐层和逐向量自适应，后者策略的优势在于极大地简化了搜索（因为我们只有3个 $\{v^*\}$ 系数）。此外，我们还实验了跨不同任务的 $\{v^*\}$ 进行归一化（使得它们的和固定为1）或保持它们不受约束。由于缺乏验证集，我们仅报告在优化结束时，我们从这些测试配置中获得的最佳样本在每个任务剩余的未见样本上的性能。

表6: 用于SVF和LoRA训练的超参数。我们对不同方法中某些敏感的超参数进行了扫描, 以进行公平的比较。

SVF Hyperparameters	
Initial mean value of z	0.1
Initial variance value of z	1×10^{-3}
Global batch size	256
Learning rate	2×10^{-3}
Clip max norm	1×10^{-3}
KL coefficient λ	0.0, 0.1, 0.2, 0.3
LoRA Hyperparameters	
Rank	16
LoRA alpha	32
LoRA dropout	0.05
Global batch size	256
Learning rate	2×10^{-4} , 5×10^{-4} , 2×10^{-5} , 5×10^{-5} , 2×10^{-6} , 5×10^{-6} ,
Clip max norm	1×10^{-3} , 1.0

表7: 附加对比实验。(括号内为标准化分数)

Method	GSM8K	MBPP-Pro	ARC-Easy
LLAMA3-8B-INSTRUCT	75.89 (1.00)	64.65 (1.00)	88.59 (1.00)
+ IA3	78.01 (1.03)	67.68 (1.05)	89.10 (1.01)
+ DORA	78.09 (1.03)	64.65 (1.00)	89.14 (1.01)
+ SVF(Ours)	79.15 (1.04)	66.67 (1.03)	89.56 (1.01)
Method	MATH	Humaneval	ARC-Challenge
LLAMA3-8B-INSTRUCT	24.54 (1.00)	60.98 (1.00)	80.63 (1.00)
+ IA3	23.64 (0.96)	59.76 (0.98)	81.57 (1.01)
+ DORA	24.44 (0.99)	52.44 (0.86)	81.14 (1.01)
+ Transformer ² (Prompt)	25.22 (1.03)	61.59 (1.01)	81.74 (1.01)
+ Transformer ² (Cls-expert)	25.18 (1.03)	62.80 (1.03)	81.37 (1.01)
+ Transformer ² (Few-shot)	25.47 (1.04)	62.99 (1.03)	82.61 (1.02)

B 附加结果

B.1 基线与更多PEFT方法的比较

我们对更多参数高效微调方法进行了额外的比较研究, 包括IA3 (Liu等人, 2022) 和DORA (Liu等人, 2024)。

如表7所示, SVF仍然优于其他方法, 并显示出良好的泛化性能。

B.2 少样本数量的影响

我们研究了少样本适应中可用样本数量与下游性能之间的关系。我们的分析集中在LLAMA3-8B-INSTRUCT表现出最高基线性能的测试任务上, 以防止在基于CEM的搜索中出现无效信号。

。

如表8所示, $\{v^*\}$ 的益处显著

我们的少样本策略在只有3到5个测试样本的情况下就已显而易见。此外, 性能在超过10个样本后似乎趋于平稳, 这突显了我们基本的且固有的正则化SVF pa-

表8: Arc挑战任务中的少样本自适应缩放。性能随示例数量而变化。

Method	Transformer ²	IA ³ 100 steps	IA ³ 1000 steps
LLAMA3-8B-INSTRUCT	80.63 (1.00)	80.63 (1.00)	80.63 (1.00)
+ 3-shot adaptation	82.18 (1.02)	81.83 (1.01)	79.01 (0.98)
+ 5-shot adaptation	82.38 (1.02)	80.89 (1.00)	79.41 (0.98)
+ 10-shot adaptation	82.61 (1.02)	82.00 (1.02)	79.78 (0.99)
+ 20-shot adaptation	82.61 (1.02)	81.40 (1.01)	79.61 (0.99)

参数化有效地补充了自适应。这种效率能够优化数据的使用，从而增强对测试任务的理解。

为完整起见，我们还在IA³ (Liu等人, 2022)上进行了具有相同设置的实验，这是一种利用少量示例的另一种方法。所有实验均采用全批大小、 5×10^{-5} 的学习率以及100和1000个训练步骤进行。

我们的结果表明，在所有考虑的少量样本数量下，IA³在未见测试任务上的性能都逊于基于CEM的自适应方法。我们注意到，在我们的实验中，我们必须大幅限制优化步骤的数量，以避免IA³的50万个参数在少量样本上过拟合。然而，我们认为即使只有100步，过拟合可能仍在一定程度上发生，模型在这个极小的数据集上达到完美的训练精度也验证了这一点。这种基于微调的自适应方法的局限性突出了我们在Transformer²中基于CEM的自适应方法的优越泛化能力。

B.3 训练任务上的跨模型SVF迁移

我们在正文表5中提供了补充结果，其中我们分析了从GSM8K、MBPP-pro和ARC-Easy训练到我们考虑的测试任务的SVF跨模型迁移性能。表9显示了相同的迁移设置下的结果，这次评估的是在与获得LLAMA3-8B-INSTRUCT {v*}向量相同的训练任务上训练的MISTRAL-7B-INSTRUCT-V0.3。总的来说，我们观察到类似的趋势，尽管与原始模型相比改进不那么一致（仅在3个任务中的1个任务中），但性能仍然远高于随机洗牌的基线。这些结果进一步证实了SVF参数化的规范排序对于跨模型迁移至关重要，再次突出了其内在的自我适应能力。

表9：跨模型 z 向量迁移。将LLAMA3-8B-INSTRUCT上训练的SVF专家向量迁移到MISTRAL-7B-INSTRUCT-V0.3中各个训练任务的结果。

Method	GSM8K	MBPP-pro	ARC-Easy
MISTRAL-7B-INSTRUCT-V0.3	42.83 (1.00)	49.50 (1.00)	81.65 (1.00)
+ Llama SVF (ordered σ_i)	42.61 (0.99)	48.48 (0.98)	81.78 (1.00)
+ Llama SVF (shuffled σ_i)	41.93 (0.98)	46.34 (0.94)	80.81 (0.99)

B.4 LORA和策略梯度的训练曲线

图9给出了在GSM8K任务上进行LoRA训练的学习曲线。

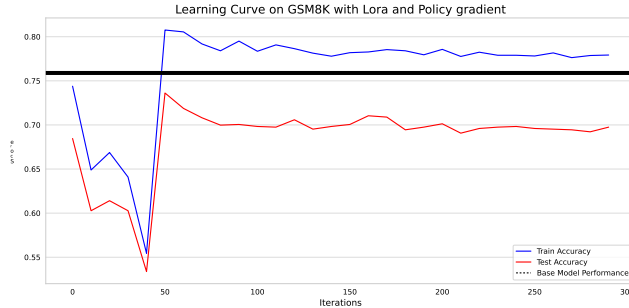


图9：使用策略梯度训练LoRA。虚线显示了LLAMA3-8B-INSTRUCT在测试集上的性能。LoRA在训练阶段开始时崩溃并未能恢复，导致测试性能下降。我们尝试了广泛的学习率（ 2×10^{-4} , 5×10^{-4} , ..., 2×10^{-2} , 5×10^{-2} ），所有学习曲线都与所示曲线相似。

C 在LLaMA3和Mistral上进行主成分分析

为了研究奇异值最大的奇异分量是否能够捕获权重矩阵的大部分信息，我们对LLAMA3-8B-INSTRUCT和MISTRAL-7B-INSTRUCT-V0.3中的权重矩阵进行了主成分分析（PCA）（参见图10和11）。在每个图中，我们绘制了权重矩阵 $W \in \mathbb{R}^{m \times n}$ 中每种类型的模块的所有层中前 r 个分量捕获的方差。

$$\text{ratio} = \frac{\sum_{i=1}^r \sigma_i}{\sum_{j=1}^{\min(m,n)} \sigma_j}$$

这里， σ 是权重矩阵 W 的有序（从大到小）奇异值。从图中可以很容易看出，当 $r =$ 为 256 时，这些顶部成分平均只捕获不到 50% 的方差。对于 MLP 层，这个比例甚至低于 20%。另一方面，LoRA-XS 或类似方法采用的秩远小于 256，导致更多的信息丢失，并限制了其主要依赖于这些 r 成分的建模能力。

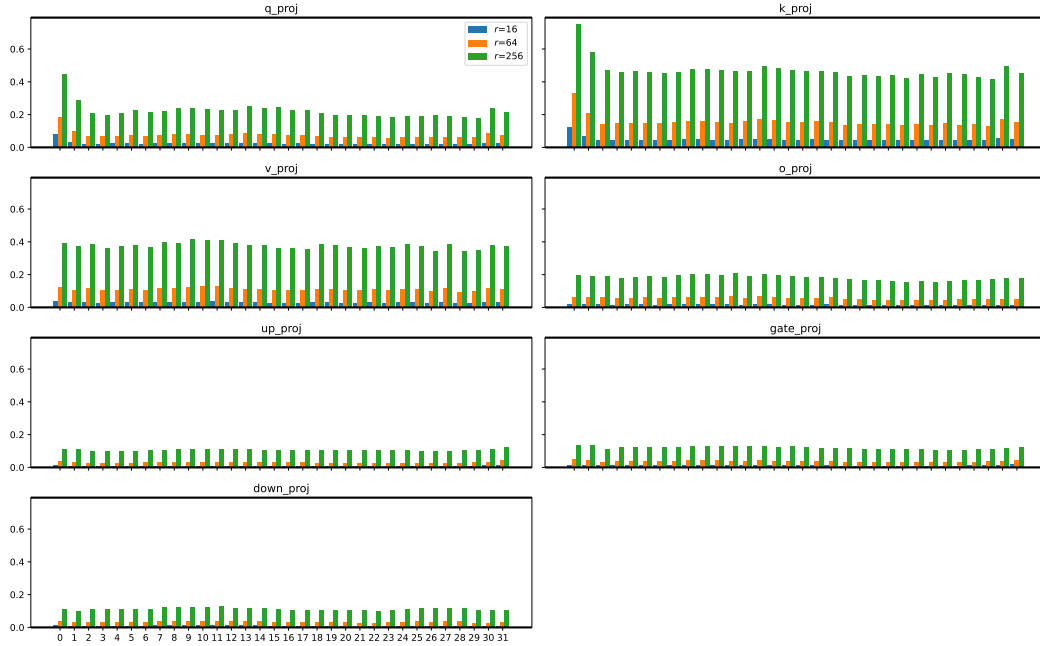


图10: LLaMA3-8B-INSTRUCT 的 PCA 分析。y 轴显示前 r 个奇异分量捕获的方差比例，x 轴显示层索引。除了 Query、Key 和 Value 投影矩阵外，较小的 r 值仅捕获参数矩阵奇异值中极小一部分的方差。

D 效率考量及改进

我们的基于CEM的自适应方法涉及对每个目标任务运行少量样本的推理（在我们的实验中最多10个）。在典型配置中，此过程相对高效：例如，我们的CEM-light方法（3次尝试，生成10次）在大约11分钟内完成了ARC挑战任务。如表10所示，

表10: 不同推理时间自适应预算下的3-shot和轻量级变体性能。

Method	ARC-Challenge
LLAMA3-8B-INSTRUCT	80.63 (1.00)
+ CEM 10-shot adaptation	82.61 (1.02)
+ CEM 3-shot (30% of prompts)	82.18 (1.02)
+ CEM light (3% of prompts)	82.08 (1.02)

这种更轻量级的设置将样本总数减少到原始设置的仅 3%，同时仍比基础模型提供了大幅度的性能提升。

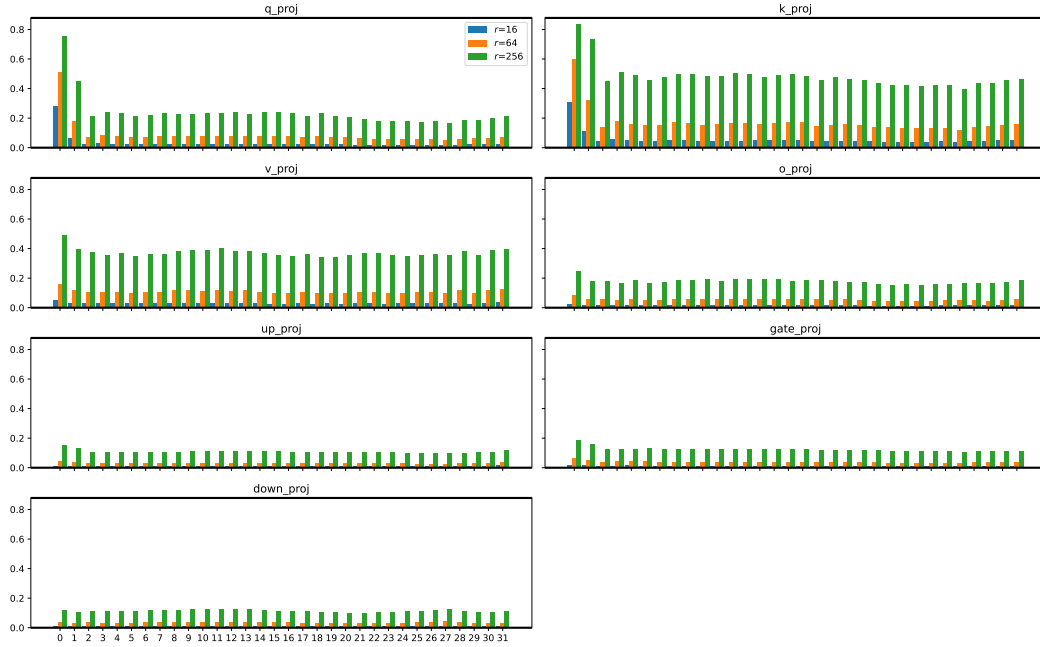


图11: MISTRAL-7B-INSTRUCT-V0.3 的 PCA 分析。y 轴显示前 r 个奇异分量捕获的方差比例, x 轴显示层索引。除了 Query、Key 和 Value 投影矩阵外, 较小的 r 值仅捕获参数矩阵中奇异值的一小部分方差。

我们承认, 基于CEM的适配需要在一次性搜索SVF-tune向量最优组合权重所花费的开销和性能之间进行权衡。增加少样本示例的数量或生成的数量可以提高性能, 但会增加额外的计算开销。然而, 重要的是要注意, 这种适配成本是每个任务的一次性开销。当应用于具有大量提示的任务时, 每个提示的成本会显著降低。

此外, 在实际场景中, 基于CEM的自适应方法比少样本提示方法具有更好的可扩展性, 后者需要增加每个提示的长度, 导致随着任务规模的增长, 可扩展性变得更差。相比之下, 我们的方法侧重于有效地确定最优专家向量组合, 并避免了重复的推理时间成本。然而, 我们注意到, 对于提示非常少的任务, 开销可能很大。因此, 对于这些特定设置, 其他自适应方法可能更合适。

我们还重点介绍了提高效率的两个直接方向:

1. 减少少样本示例的数量: 正如附录B.2中的消融研究所示, 即使在3-shot设置中也能看到显著的益处, 这只需要评估每个生成提示数量的30%。
2. 减少最大生成次数: 在探索的设置中, CEM参数往往会很快收敛, 在远低于100次的生成次数后就非常接近最终值。

最后, 在这项工作中, 我们只考虑了CEM, 因为它简单易行; 存在几种不同的进化算法, 在经验上显示出更好的效率和收敛特性, 我们希望在未来的研究中探索这些算法。