仅使用
前向传递微调语言模型

Sadhika Malladi* **Tianyu Gao*** **Eshaan Nichani** **Alex Damian**

Jason D. Lee **Danqi Chen** **Sanjeev Arora**

Princeton University
{smalladi, tianyug, eshnich, ad27, jasonlee, danqic, arora}@princeton.edu

Abstract

微调语言模型 (LM) 已在各种下游任务中取得成功，但随着 LM 规模的增长，反向传播需要大量内存。零阶 (ZO) 方法原则上可以仅使用两次前向传递来估计梯度，但理论上在优化大型模型时速度极其缓慢。在这项工作中，我们提出了一种内存高效的零阶优化器 (**MeZO**)，采用经典的 ZO-SGD 方法进行就地操作，从而使用与推理相同的内存占用来微调 LM。例如，使用单个 A100 80GB GPU，MeZO 可以训练一个 300 亿参数的模型，而使用反向传播进行微调在相同预算下只能训练一个 2.7B 的 LM。我们跨模型类型（屏蔽和自回归 LM）、模型规模（高达 66B）和下游任务（分类、多项选择和生成）进行综合实验。我们的结果表明 (1) MeZO 显着优于上下文学习和线性探测；(2) MeZO 实现了与跨多个任务的反向传播微调相当的性能，最多减少 12 × 内存；(3) MeZO 兼容全参数和参数高效调优技术，如 LoRA 和前缀调优；(4) MeZO 可以有效地优化不可微分的目标（例如，最大化精度或 F1）。我们用理论见解支持我们的经验发现，强调充分的预训练和任务提示如何使 MeZO 能够微调大型模型，尽管经典的 ZO 分析表明并非如此。[2]

1 介绍

微调预训练语言模型 (LM) 一直是解决许多语言任务的主要方法 [18]，适应专门领域 [29]，或结合人类指令和偏好 [51]。然而，随着 LM 的扩大 [8, 50]，计算反向传播的梯度需要大量内存——在我们的测试中，推理所需的内存高达 12× ——因为它需要在前向传播期间缓存激活，在前向传播期间缓存梯度向后传递，并且，在 Adam [35] 的情况下，还存储梯度历史 （请参阅 Section 3.4 了解详细分析）。因此，虽然可以在单个 Nvidia A100 GPU（具有 80GB 内存）上使用 300 亿 (30B) 参数 LM 运行推理，但使用 Adam 的反向传播仅适用于 2.7B LM。参数有效的微调方法 (PEFT [30, 40, 37]) 只更新一小部分网络参数，但仍需要缓存许多激活，因为调优参数分散在整个模型中。在我们的测试中，微调具有全参数或 PEFT 的 OPT-13B 模型分别需要比推理多 12 × 和 6× 的内存。

上下文学习 (ICL [8]) 允许通过单次推理解决许多任务，在此期间模型在其上下文中处理标记示例（演示），然后输出对测试示例的预测。虽然这允许模型快速适应特定用例，但当前模型允许有限的上下文大小（因此，有限的演示）并且性能对演示的格式和选择很敏感 [43, 47]。ICL 也常常比中型模型的微调表现更差 [8]。此外，使用 ICL 进行推理的成本更高，因为它总是需要在上下文中进行演示，从而增加了输入长度。考虑替代标准反向传播的另一个原因是它不能包含不可微分的标准，这些标准在根据人类偏好分数或设定的安全标准微调 LMs 中很受欢迎 [63, 51]。通常，这些调整涉及从人类反馈 (RLHF [11]) 中强化学习，这是昂贵的。

*Equal contribution and corresponding authors.

[2]我们的代码可在 https://github.com/princeton-nlp/MeZO 获得。

Preprint. Under review.

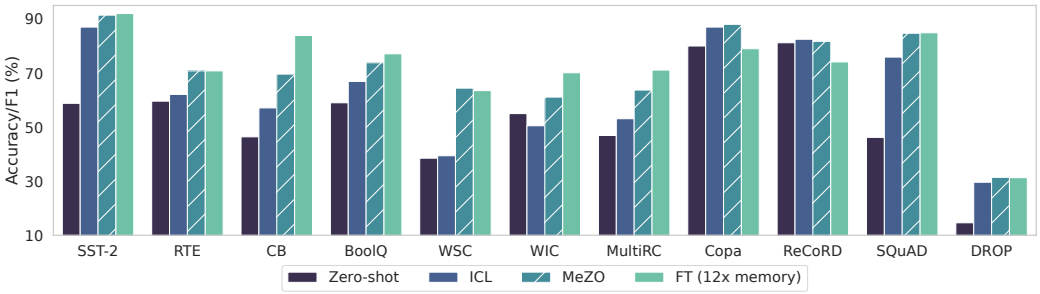

Figure 1: OPT-13B 使用零样本、上下文学习 (ICL)、MeZO（我们报告 MeZO / MeZO (LoRA)/ MeZO （前缀）中最好的）和 Adam (FT) 微调的结果。MeZO 展示了优于零样本和 ICL 的结果，并且在 11 个任务中的 7 个上与 FT（在 1 % 内）表现相当，尽管只使用了 1/12 的内存。详细数字请参见表 1 ，内存分析请参见图 3 。

经典的零阶优化方法 (ZO-SGD [62]) 仅使用损失值的差异来估计梯度。因此，原则上，该方法可以仅通过前向传递来更新神经网络，尽管简单的实现仍然会使内存开销和经典下限加倍 [49, 21] 表明收敛速度随模型大小线性减慢。因此，ZO 方法已应用于深度学习设置中以查找对抗性示例或调整输入嵌入 [65, 64] 但不是直接优化大型模型（请参阅 Liu et al. [44] 进行调查）。

在这项工作中，我们提出了一种内存高效的零阶优化器 (MeZO)，它采用经典的 ZO-SGD 算法并将其内存消耗降低到与推理相同的水平。我们应用 MeZO 来微调大型 LM，并表明，无论是在经验上还是理论上，MeZO 可以成功优化具有数十亿参数的 LM。具体来说，我们的贡献是：

1. 在 MeZO 中，我们调整了 ZO-SGD 算法 [62] 和许多变体，以在几乎没有内存开销的情况下在任意大的模型上就地运行（参见算法 1 和 Section 2 ）。
2. 我们对模型类型（掩蔽 LM 和自回归 LM）、模型规模（从 350M 到 66B）和下游任务（分类、多项选择和生成）进行综合实验。MeZO 始终证明优于零次、ICL 和线性探测。此外，使用 RoBERTa-large，MeZO 在 5 个% 差距内实现了接近标准微调的性能；使用 OPT-13B，MeZO 在 11 项任务中的 7 项上优于或与微调性能相当，尽管需要大约 12× 更少的内存（Figure 1 和 Section 3 ）。
3. 我们在 Section 3 中展示了 MeZO 与全参数调整和 PEFT（例如，LoRA [30] 和前缀调整 [40] ）的兼容性。
4. 进一步的探索表明 MeZO 可以优化不可微分的目标，例如准确性或 F1 分数，同时仍然只需要与推理 (Section 3.3) 相同的内存。
5. 我们的理论表明，充分的预训练可确保 MeZO 的每步优化率 (Theorem 1) 和全局收敛率 (Lemma 3) 取决于景观的特定条件数（即局部有效等级，参见 Assumption 1 ）而不是数字的参数。这个结果与现有的 ZO 下界形成鲜明对比 [49, 21] 表明收敛速度可以与参数数量 (Section 4) 成比例地减慢。

2 零阶优化

长期以来，人们一直在凸目标和强凸目标的背景下研究零阶 (ZO) 优化器。下面首先介绍一个经典的 ZO 梯度估计器 SPSA(Definition 1 [62]) 和对应的 SGD 算法 ZO-SGD (Definition 2)。然后我们描述 MeZO ，我们的就地实现需要与 Section 2.1 和 Algorithm 1 中的推理相同的内存。我们强调 SPSA 也可以用于更复杂的优化器，例如 Adam，我们也为这些算法提供内存高效实现 (Section 2.2)。

考虑一个带标签的数据集 $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i\in[|\mathcal{D}|]}$ 和一个大小为 B 的小批量 $\mathcal{B} \subset \mathcal{D}$ ，我们让 $\mathcal{L}(\boldsymbol{\theta}; \mathcal{B})$ 表示小批量的损失。我们在此设置中引入了经典的 ZO 梯度估计。

Definition 1 (Simultaneous Perturbation Stochastic Approximation or SPSA [62])**.** *给定一个带有参数 $\boldsymbol{\theta} \in \mathbb{R}^d$ 和损失函数 $\mathcal{L}$ 的模型，SPSA 将小批量 $\mathcal{B}$ 上的梯度估计为*

$$\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B}) = \frac{\mathcal{L}(\boldsymbol{\theta} + \epsilon\boldsymbol{z}; \mathcal{B}) - \mathcal{L}(\boldsymbol{\theta} - \epsilon\boldsymbol{z}; \mathcal{B})}{2\epsilon}\boldsymbol{z} \approx \boldsymbol{z}\boldsymbol{z}^\top \nabla\mathcal{L}(\boldsymbol{\theta}; \mathcal{B}) \tag{1}$$

其中 $\boldsymbol{z} \in \mathbb{R}^d$ 与 $\boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}_d)$ 和 ϵ 是扰动尺度。n -SPSA 梯度估计平均 $\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B})$ 超过 n 随机采样的 $\boldsymbol{z}$ 。

Algorithm 1: MeZO

Require : parameters $\boldsymbol{\theta} \in \mathbb{R}^d$, loss $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$, step budget T , perturbation scale ϵ , batch size B learning rate schedule $\{\eta_t\}$

for $t = 1, ..., T$ **do**
  Sample batch $\mathcal{B} \subset \mathcal{D}$ and random seed s
  $\boldsymbol{\theta} \leftarrow$ `PerturbParameters(` $\boldsymbol{\theta}, \epsilon, s$`)`
  $\ell_+ \leftarrow \mathcal{L}(\boldsymbol{\theta}; \mathcal{B})$
  $\boldsymbol{\theta} \leftarrow$ `PerturbParameters(` $\boldsymbol{\theta}, -2\epsilon, s$`)`
  $\ell_- \leftarrow \mathcal{L}(\boldsymbol{\theta}; \mathcal{B})$
  $\boldsymbol{\theta} \leftarrow$ `PerturbParameters(` $\boldsymbol{\theta}, \epsilon, s$`)` ▷ `Reset parameters before descent`
  `projected_grad` $\leftarrow (\ell_+ - \ell_-)/(2\epsilon)$
  Reset random number generator with seed s ▷ `For sampling` z
  for $\theta_i \in \boldsymbol{\theta}$ **do**
    $z \sim \mathcal{N}(0, 1)$
    $\theta_i \leftarrow \theta_i - \eta_t *$ `projected_grad` $* z$
  end
end

Subroutine `PerturbParameters(` $\boldsymbol{\theta}$, ϵ , s`)`
  Reset random number generator with seed s ▷ `For sampling` z
  for $\theta_i \in \boldsymbol{\theta}$ **do**
    $z \sim \mathcal{N}(0, 1)$
    $\theta_i \leftarrow \theta_i + \epsilon z$ ▷ `Modify parameters in place`
  end
  return $\boldsymbol{\theta}$

SPSA 只需要两次前向通过模型来计算梯度估计（对于 n -SPSA，每个估计需要 $2n$ 次前向通过）。在训练过程中，n 可以被视为一个超参数并遵循一个时间表 [6, 9] ，尽管在粗略的实验 (Appendix A) 中，$n = 1$ 是最有效的。我们默认使用 $n = 1$ 。众所周知，该估计可用于替换任何优化器（例如 SGD）中的反向传播梯度。

Definition 2 (ZO-SGD). *ZO-SGD 是一个学习率为 η 的优化器，它将参数更新为 $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B}_t)$ ，其中 $\mathcal{B}_t$ 是时间 t 的小批量，$\hat{\nabla}\mathcal{L}$ 是 SPSA 梯度估计。*

2.1 内存高效 ZO-SGD (MeZO)

香草 ZO-SGD 算法消耗两倍的推理内存，因为它需要存储 $\boldsymbol{z} \in \mathbb{R}^d$ 。我们提出了 ZO-SGD 的内存高效实现，称为 **MeZO** ，如 Algorithm 1 中所示。在每一步，我们首先对随机种子 s 进行采样，然后对于 $\boldsymbol{z}$ 在 Algorithm 1 中的四次使用中的每一次，我们将随机数生成器重置为 s 并重新采样 $\boldsymbol{z}$ 的相关条目。使用此就地实现，MeZO 的内存占用量相当于推理内存成本。

我们注意到 Algorithm 1 描述了分别扰动每个参数，这对于大型模型来说可能很耗时。在实践中，我们可以通过扰动整个权重矩阵而不是独立地扰动每个标量来节省时间。这会产生与最大权重矩阵一样大的额外内存成本；通常，这是词嵌入矩阵（例如，OPT-66B 为 0.86GB）。

2.2 MeZO 扩展

MeZO 还可以与其他基于梯度的优化器结合使用，包括带有动量的 SGD 或 Adam。虽然天真的实现需要额外的内存来存储梯度矩估计，MeZO -momentum 和 MeZO -Adam 通过使用保存的传递损失和 $\boldsymbol{z}$ 重新计算梯度的移动平均值来减轻这种开销（有关完整讨论，请参阅 Appendix B ）。

我们还注意到，SPSA 梯度估计的所有坐标都具有相同的尺度，但深度 Transformer 的每一层都可以具有不同尺度的梯度 [42, 44] 。因此，我们从分层自适应优化器 [75, 76] 中汲取灵感来设计几个 MeZO 变体。粗略的实验表明，这些算法并没有更有效（就前向传播而言），但我们仍然将它们作为更复杂目标的潜在优化器。请参见 Appendix B 。

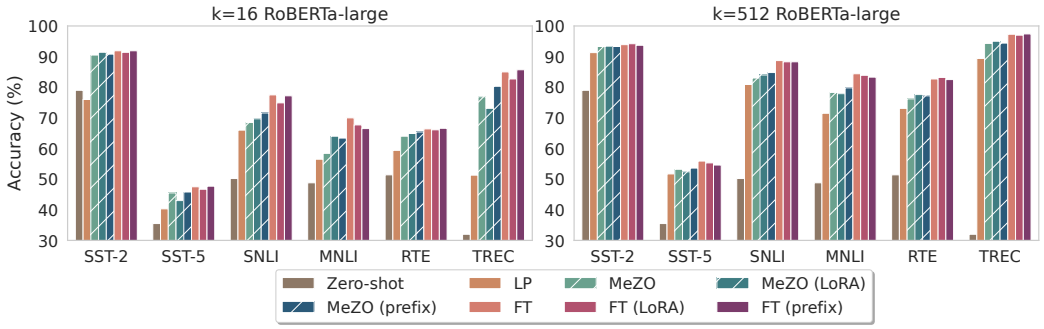

Figure 2: RoBERTa-large 上的实验。我们报告了零样本、线性探测 (LP) 和 MeZO 以及具有完整参数、LoRA 和前缀调整的微调 (FT)。MeZO 优于零样本和 LP，并且接近 FT（$k = 512$ 的 5 % 以内），内存少得多。Table 16 中的详细数字。

3 实验

初步实验（Appendix A ）表明 ZO 仅在使用提示时有效 [8, 60, 23] 。下面的所有实验都使用 Appendix D.2 中详述的提示。所有使用反向传播 (FT) 进行微调的实验都遵循惯例并使用 Adam，尽管我们还在 Appendix E 中展示了带有 SGD 的 FT。

我们在带有提示的少镜头和多镜头设置中对中型掩蔽 LM（RoBERTa-large，350M [46] ）和大型自回归 LM（OPT-13B、30B、66B [78] ）进行综合实验。我们还探索了全参数调整和 PEFT，包括 LoRA [30] 和前缀调整 [40] （详见附录 D.5 ）。我们将 MeZO 与零样本、上下文学习 (ICL)、线性探测 (LP) 和 Adam (FT) 微调进行比较。MeZO 使用的内存比 FT 少得多，但需要更多的训练步骤。

我们首先展示 MeZO 在模型类型、大小和任务类型方面显着优于零样本、ICL 和 LP。此外，MeZO 在许多任务上的性能与 FT 相当，同时显着降低了内存成本，例如 OPT-13B 上的 12 × 。进一步的实验表明，MeZO 可以优化不可微分的目标，例如准确性和 F1 分数 (Section 3.3)。我们比较了 Figures 3 and 4 中 ICL、FT、LP 和 MeZO 的内存消耗。

3.1 中型掩码语言模型

我们使用 RoBERTa-large 在情感分类、自然语言推理和主题分类任务上进行实验。我们按照 [23, 48] 研究少镜头和多镜头设置，为 $k = 16$ 和 $k = 512$ 的每个类采样 k 个示例（详见 Appendix D ）。我们在下面总结了 Figure 2 和 Table 16 的结果。

MeZO 的效果明显优于零样本、线性探测和其他内存等效方法。 在所有六项不同的任务中，MeZO 可以优化预训练模型并始终如一地表现优于零样本和线性探测。我们还针对多项任务展示了 MeZO 可以胜过另一种 ZO 算法 BBTv2 [64] ，最高可达 11% 绝对值 (Appendix E.4)。[3]

有了足够的数据，MeZO 可实现与 FT 相当的性能（高达 5 个% 差距）。 MeZO 在 $k = 16$ 上实现了接近微调的性能，一些任务只有 2 个% 差距。当使用 $k = 512$ 数据时，MeZO 和 FT 之间的差距在所有任务中进一步缩小到 5 % 以内。

MeZO 适用于全参数调优和 PEFT。 全参数调优 (MeZO) 和 PEFT（具有 LoRA 和前缀调优的 MeZO ）实现了相当的性能，而 MeZO（前缀）有时优于 MeZO 。我们还在 Appendix E.3 中表明这三个变体以相似的速率收敛，这与我们在 Section 4 中的理论一致，这表明 MeZO 以独立于被优化参数数量的速率收敛。

我们在附录 E.1 中展示了更多 FT（FT with SGD）和 MeZO 变体的额外结果。我们看到 (1) ZO-Adam 有时优于 ZO-SGD 但在任务之间并不一致；(2) LP 然后 MeZO ，按照微调的建议 [36] ，有时可以提高性能。

[3] BBTv2 对 # 参数敏感，只能训练向下投影的前缀而不是完整模型。

Task Task type	SST-2	RTE	CB	BoolQ	WSC	WIC	MultiRC	COPA	ReCoRD	SQuAD	DROP
	——— classification ———							– multiple choice –		— generation —	
Zero-shot	58.8	59.6	46.4	59.0	38.5	55.0	46.9	80.0	81.2	46.2	14.6
ICL	87.0	62.1	57.1	66.9	39.4	50.5	53.1	87.0	**82.5**	75.9	29.6
LP	**93.4**	68.6	67.9	59.3	63.5	60.2	63.5	55.0	27.1	3.7	11.1
MeZO	91.4	66.1	67.9	67.6	63.5	**61.1**	60.1	**88.0**	81.7	**84.7**	30.9
MeZO (LoRA)	89.6	67.9	66.1	**73.8**	**64.4**	59.7	61.5	87.0	81.4	83.8	**31.4**
MeZO (prefix)	90.7	**70.8**	**69.6**	73.1	57.7	59.9	**63.7**	84.0	81.2	84.2	28.9
FT (12x memory)	92.0	70.8	83.9	77.1	63.5	70.1	71.1	79.0	74.1	84.9	31.3

Table 1: OPT-13B 上的实验（有 1,000 个例子）。ICL：情境学习；LP：线性探测；FT：与 Adam 进行全面微调。MeZO 全面优于零样本、ICL 和 LP，并在 11 项任务中的 7 项上实现了与 FT 相当（在 1 % 以内）或更好的性能。

Task	SST-2	RTE	BoolQ	WSC	WIC	SQuAD
30B zero-shot	56.7	52.0	39.1	38.5	50.2	46.5
30B ICL	81.9	66.8	66.2	56.7	51.3	78.0
30B MeZO/MeZO (prefix)	**90.6**	**72.6**	**73.5**	**63.5**	**59.1**	**85.2**
66B zero-shot	57.5	**67.2**	66.8	43.3	50.6	48.1
66B ICL	89.3	65.3	62.8	52.9	54.9	81.3
66B MeZO/MeZO (prefix)	**93.6**	66.4	**73.7**	**63.5**	**58.9**	**85.0**

Table 2: OPT-30B 和 OPT-66B 上的实验（有 1,000 个例子）。我们报告 MeZO 和 MeZO （前缀）中最好的。有关更多结果，请参阅 Appendix E.2 。我们在大多数任务中看到 MeZO 有效优化了多达 66B 的模型，并且优于零样本和 ICL。

3.2 大型自回归语言模型

凭借 RoBERTa-large 的可喜成果，我们将 MeZO 扩展到 OPT 系列 [78]，规模为 13B（表 1 ）、30B 和 66B（Table 2 ）。我们选择 SuperGLUE [69] 任务 [4] （包括分类和多选）和生成任务。我们分别为每个数据集随机抽取 1000、500 和 1000 个示例进行训练、验证和测试。详情请参考附录 D 。从表 1 中的主要结果，我们得出以下观察结果。

MeZO 优于内存等效方法并接近微调结果。 我们看到，在 13B 参数规模上，MeZO 及其 PEFT 变体几乎在所有任务中都优于零样本、ICL 和 LP。与 FT 相比，后者的内存多 12 × (Section 3.4)，MeZO 在 11 个任务中的 7 个上实现了相当（在 1 % 以内）或更好的性能。

MeZO 在分类、多项选择和生成任务中表现出强大的性能。 我们在生成任务上调查了 MeZO , 这些任务被认为比分类或多项选择任务更复杂。我们对两个问答数据集 SQuAD [57] 和 DROP [20] 进行评估。我们使用 teacher-forcing 进行训练，使用贪婪解码进行推理（Appendix D 中的详细信息）。

表 1 显示，在所有生成任务上，MeZO 优于零样本、ICL 和 LP，并实现了与 FT 相当的性能。考虑到微调 LM 的许多应用——包括指令调优或域适应——目标生成任务，我们的结果强调了 MeZO 作为一种内存高效技术的潜力，可以优化大型 LM 以实现现实和令人兴奋的应用。

MeZO 可扩展至 660 亿个参数模型。 我们在表 2 中展示了 MeZO 在更大模型（高达 66B）上的功效。虽然在这种规模上直接微调模型的成本非常高 (Section 3.4)，MeZO 可以有效地优化这些模型并优于零样本和 ICL。

3.3 以不可区分的目标进行训练

我们通过初始实验证明了 MeZO 优化不可微分目标的功效。准确性和 F1 用作各自的目标（Appendix D.6 中的详细信息）。表 3 表明具有精度/F1 的 MeZO 成功优化了 LM，其性能优于零射击。尽管最小化交叉熵会带来更强的性能，但这些初步发现强调了应用 MeZO 优化不可微目标的潜力，而无需明确的可微代理，例如人类偏好 [51] 。

[4] 我们还包括 SST-2，这是我们用于开发的简单情感分类任务。

Model	RoBERTa-large (350M)				OPT-13B
Task	**SST-2**	**SST-5**	**SNLI**	**TREC**	**SQuAD**
Zero-shot	79.0	35.5	50.2	32.0	46.2
Cross entropy (FT)	93.9	55.9	88.7	97.3	84.2
Cross entropy (MeZO)	93.3	53.2	83.0	94.3	84.7
Accuracy/F1 (MeZO)	92.7	48.9	82.7	68.6	78.5

Table 3: MeZO 具有不可微分的目标。对于分类（$k = 512$），我们使用具有全参数的 MeZO 并优化精度；对于 SQuAD（1,000 个示例），我们使用 MeZO （前缀）和 F1。

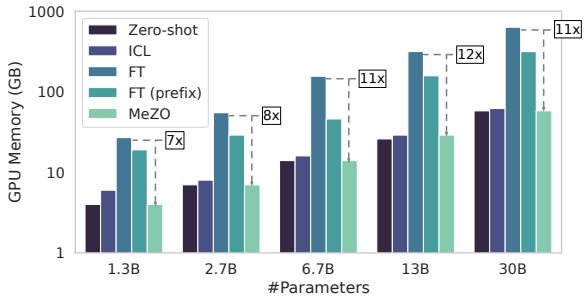

Figure 3: MultiRC 上不同 OPT 模型和调整方法的 GPU 内存消耗（平均每个示例 400 个标记）。

Hardware	Largest OPT that can fit		
	FT	**FT-prefix**	**MeZO**
1 × A100 (80GB)	2.7B	6.7B	30B
2 × A100 (160GB)	6.7B	13B	66B
4 × A100 (320GB)	13B	30B	66B
8 × A100 (640GB)	30B	66B	175B †

Figure 4: 可以使用特定硬件和算法进行调整的最大 OPT 模型。†：未经实际测试的预测结果。

3.4 内存使用情况

在本节中，我们将分析零样本、ICL、FT、FT（前缀）和 MeZO 的内存使用情况。我们在 MultiRC（平均 # 令牌 = 400）上使用 Nvidia A100 GPU（80GB 内存）测试各种尺寸的 OPT 模型，并报告峰值 GPU 内存消耗（Appendix D.7 中的详细信息）。

如图 3（详细数字请参考 Appendix E.5）所示，MeZO 表现出与零样本相同的内存消耗，同时与标准 FT 相比，内存节省高达 12 倍，与 FT（前缀）相比，内存节省高达 6 倍。这种优势可以在固定的硬件预算内训练更大的模型，如图 4 所示。具体来说，使用单个 A100 GPU，MeZO 允许调整比 FT 可行的模型大 11 倍的模型。

上述测量取决于所使用的基础设施和包。在 Appendix C 中，我们比较了 MeZO 和反向传播的理论时间-内存权衡。我们发现 MeZO 总是比反向传播更节省内存，而且通常更省时。上面的两种内存分析模式也没有考虑最近在使转换器的内存效率更高方面取得的进展，例如梯度检查点 [10]、FlashAttention [14] 和量化训练 [17]。我们将调查 MeZO 如何使用这些方法留到未来的工作中。

4 理论

我们的理论分析强调了为什么 MeZO 可以优化大型 LM，尽管有许多经典结果 [49, 31, 56, 1] 表明在训练如此多的参数时优化应该非常慢。在本节中，我们展示了当损失情况表现出有利条件（Assumption 1）时，我们可以得出与参数数量无关的收敛速度。我们表明，损失每步减少的速度与参数维度 d（Theorem 1）无关，并且在更强的条件下，算法在时间上收敛，与 d（Lemma 3）无关。总而言之，这些结果表明 MeZO 在微调时并不比 SGD 慢得多。[5] 为了便于说明，我们假设 $\boldsymbol{z}$ 是从半径为 $\sqrt{d}$ 的球体中采样的，在 Appendix F.2 中，我们推导出实验中使用的一般高斯 $\boldsymbol{z}$ 的速率。

我们遵循 SGD 的经典分析，并用 SPSA 代替小批量梯度估计。考虑小批量 SGD 更新 $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta \nabla \mathcal{L}(\boldsymbol{\theta}; \mathcal{B}_t)$，其中 $\mathcal{B}_t$ 是从 $\mathcal{D}^B$ 统一抽取的小批量。至关重要的是，SGD 小批量梯度估计是无偏的。

Definition 3 (Unbiased Gradient Estimate)**.** 如果 $\mathbb{E}[\boldsymbol{g}(\boldsymbol{\theta}, \mathcal{B})] = \nabla \mathcal{L}(\boldsymbol{\theta})$，则称任何小批量梯度估计 $\boldsymbol{g}(\boldsymbol{\theta}, \mathcal{B})$ 是无偏的。

[5] Section 3 使用标准的 Adam for FT 选择；我们在 Appendix E.1 中提供 SGD 实验。

4.1 每步分析

经典下降引理使用泰勒展开来研究 SGD 如何减少每个优化步骤的损失。它强调，当梯度协方差很大时，每个优化步骤损失的最大可能减少量很小，从而导致优化速度变慢。

Lemma 1 (Descent Lemma)**.** *设 $\mathcal{L}(\boldsymbol{\theta})$ 为 ℓ -平滑。*[6] *对于任何无偏梯度估计 $\boldsymbol{g}(\boldsymbol{\theta}, \mathcal{B})$ ，*

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] - \mathcal{L}(\boldsymbol{\theta}_t) \leq -\eta \left\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\right\|^2 + \frac{1}{2}\eta^2 \ell \cdot \mathbb{E}[\|\boldsymbol{g}(\boldsymbol{\theta}, \mathcal{B}_t)\|^2]. \tag{2}$$

下降引理突出了梯度范数的重要性，我们在下面为 MeZO 推导了它。

Lemma 2. *让 $\mathcal{B}$ 成为大小为 B 的随机小批量。那么，MeZO 的梯度范数为*

$$\mathbb{E}_x\left[\left\|\hat{\nabla} \mathcal{L}(\boldsymbol{\theta}; \mathcal{B})\right\|^2\right] = \frac{d+n-1}{n} \mathbb{E}\left[\left\|\nabla \mathcal{L}(\boldsymbol{\theta}; \mathcal{B})\right\|^2\right].$$

其中 n 是 n -SPSA (Definition 1) 中采样的 $\boldsymbol{z}$ 的数量，d 是参数的数量。

因此，在 $n \ll d$ 的通常情况下，MeZO 的梯度范数比 SGD 大得多。[7] 下降引理也表明，为了保证损失减少，需要选择学习率作为

$$\eta \leq \frac{2\left\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\right\|^2}{\ell \cdot \mathbb{E}[\|\boldsymbol{g}(\boldsymbol{\theta}, \mathcal{B})\|^2]} \quad \xRightarrow{\text{Lemma 2}} \quad \eta_{\text{ZO}} = \frac{n}{d+n-1}\eta_{\text{SGD}} \tag{3}$$

，其中 η_{ZO} 和 η_{SGD} 分别是 MeZO 和 SGD 的最大允许学习率。因此我们看到，在没有任何进一步假设的情况下，MeZO 可以通过将最大允许学习率降低 d 的一个因子来减慢优化速度。此外，MeZO 减少了每一步可以获得的损失减少，因此，也将收敛速度减慢了 d 倍。

令人惊讶的是，我们的实验表明 MeZO 可以快速优化具有数十亿个参数的预训练模型，并且通过 PEFT 技术减少调整参数的数量并不会显着加速优化 (Appendix E.3)。我们将这些现象归因于损失的 Hessian 矩阵表现出较小的局部有效秩。在合理大小的数据集上直接测量大型 LM 的 Hessian 矩阵的有效秩是非常昂贵的。然而，许多先前的工作表明，由 SGD 训练的深度神经网络损失的 Hessian 具有非常低的有效秩 [52, 53, 24, 74, 73, 59] 。特别是，大部分频谱集中在 0 周围，只有少量异常值，这些异常值的数量是有效等级的上限。此外，之前的工作 [2, 39] 已经证明 LM 微调可以发生在非常低维的子空间（< 200 参数）中，这进一步支持了以下假设。我们将以下有效排名的假设形式化。特别是，我们需要当前迭代周围邻域中 Hessian 矩阵的上限，以具有最多 r 的有效等级。

Assumption 1 (Local r -effective rank)**.** *让 $G(\boldsymbol{\theta}_t) = \max_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}} \left\|\nabla \mathcal{L}(\boldsymbol{\theta}_t; \{(\boldsymbol{x}, \boldsymbol{y})\})\right\|$ 。存在矩阵 $\boldsymbol{H}(\boldsymbol{\theta}_t)$ 使得：*

1. *对于所有 $\boldsymbol{\theta}$ 这样的 $\|\boldsymbol{\theta} - \boldsymbol{\theta}_t\| \leq \eta d G(\boldsymbol{\theta}_t)$ ，我们有 $\nabla^2 \mathcal{L}(\boldsymbol{\theta}) \preceq \boldsymbol{H}(\boldsymbol{\theta}_t)$ 。*
2. *$\boldsymbol{H}(\boldsymbol{\theta}_t)$ 的有效等级，即 $\operatorname{tr}(\boldsymbol{H}(\boldsymbol{\theta}_t)) / \left\|\boldsymbol{H}(\boldsymbol{\theta}_t)\right\|_{op}$ ，最多为 r 。*

在此假设下，我们表明 ZO-SGD 的收敛速度不依赖于参数的数量。相反，减速因子仅取决于 Hessian 矩阵的有效秩。

Theorem 1 (Dimension-Free Rate)**.** *假设损失表现出局部 r 有效等级 (Assumption 1)。如果 $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_{ZO} \hat{\nabla} \mathcal{L}(\boldsymbol{\theta}_t; \mathcal{B})$ 是 ZO-SGD 的单步，使用 n -SPSA 估计和大小为 B 的小批量，则存在 $\gamma = \Theta(r/n)$ 使得预期损失减少可以被限定为*

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] - \mathcal{L}(\boldsymbol{\theta}_t) \leq -\eta_{ZO} \left\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\right\|^2 + \frac{1}{2}\eta_{ZO}^2 \ell \cdot \gamma \cdot \mathbb{E}[\|\nabla \mathcal{L}(\boldsymbol{\theta}; \mathcal{B})\|^2] \tag{4}$$

通过应用 Equation (3) ，我们可以直接与 SGD 下降引理进行比较。

Corollary 1. *选择学习率 $\eta_{ZO} = \gamma^{-1} \cdot \eta_{SGD}$, ZO-SGD 的损失减少为*

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] - \mathcal{L}(\boldsymbol{\theta}_t) \leq \frac{1}{\gamma} \cdot \left[-\eta_{SGD} \left\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\right\|^2 + \frac{1}{2}\eta_{SGD}^2 \ell \cdot \mathbb{E}[\|\nabla \mathcal{L}(\boldsymbol{\theta}; \mathcal{B})\|^2]\right]. \tag{5}$$

在这里我们看到，与 SGD 相比，ZO-SGD 的减速因子与局部有效等级 r 成比例，我们认为它比参数数量 d 小得多。上面的分析重点是 ZO-SGD 和 SGD 在每一步减少了多少 loss。下面，我们展示了在关于损失情况的更强假设下，我们可以获得 ZO-SGD 算法收敛到最佳值的速度。

[6] 这对于标准交叉熵目标是满足的。

[7] 我们所有的实验都使用 $n = 1$ 。

4.2 全局收敛分析

我们表明，在对损失情况的更强假设下，全局收敛速度也会减慢一个与局部有效等级成正比的因子。我们假设景观服从经典的 PL 不等式：梯度范数随着迭代的次优性呈二次方增长。

Definition 4 (PL Inequality)**.** 让 $\mathcal{L}^* = \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$ 。损失 $\mathcal{L}$ 是 μ *-PL* 如果对于所有 $\boldsymbol{\theta}$ 和 $\frac{1}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 \geq \mu(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*)$.

PL 不等式不如假设优化表现出类似内核的动态那样强，但它确保景观适合分析 [33] 。除了 PL 不等式之外，我们还假设梯度协方差的轨迹是有界的，因此噪声不会过分破坏轨迹。

Definition 5 (Gradient Covariance)**.** 大小为 B 的小批量的 *SGD* 梯度估计具有协方差 $\boldsymbol{\Sigma}(\boldsymbol{\theta}) = B(\mathbb{E}\left[\nabla\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\nabla\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^\top\right] - \nabla\mathcal{L}(\boldsymbol{\theta})\nabla\mathcal{L}(\boldsymbol{\theta})^\top)$ 。

正如我们在 Appendix F.1 中展示的那样，此假设适用于常见的损失函数，例如几种设置的平方损失或二元交叉熵（例如，内核行为 [48] ）。有了这两个假设，我们表明 ZO-SGD 的减速与有效等级 r 成比例，而不是参数维度。

Lemma 3 (Global Convergence of ZO-SGD)**.** 设 $\mathcal{L}(\boldsymbol{\theta})$ 为 μ *-PL* 并设 α 存在，使得所有 $\boldsymbol{\theta}$ 都为 $\mathrm{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta})) \leq \alpha(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*)$ 。然后

$$t = \mathcal{O}\left(\left(\frac{r}{n}+1\right)\cdot\underbrace{\left(\frac{\ell}{\mu}+\frac{\ell\alpha}{\mu^2 B}\right)\log\frac{\mathcal{L}(\boldsymbol{\theta}_0)-\mathcal{L}^*}{\epsilon}}_{\textit{SGD rate (Lemma 4)}}\right)$$

ZO-SGD 的迭代我们有 $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] \leq \mathcal{L}^* + \epsilon$.

ěsection 相关工作 ěsubsection 零阶优化 在强凸和凸设置中为 ZO-SGD 推导出许多经典下界 ěcitep jamieson2012query,agarwal2012information,raginsky2011information,duchi2015optimal,shamir2017optimal,nemirovskij1983 以及非凸 ěcitep wang2020zeroth 。这些边界通常取决于参数 d 的数量。最近，ěcitep wang2018stochastic,balasubramanian2018zeroth,cai2022zoro 表明，如果梯度具有低维结构，则查询复杂度与内在维度成线性关系，与参数数量成对数关系，尽管估计至少有 ¥$Omega(sd$¥$logd)$ 内存成本。在本文中，我们调整 ěcitep spall1992 多元 以提高内存效率，我们发现它可以成功地微调大型 LM。

ZO-SGD 的许多变体已经被提出。抽样计划 ěcitep bollapragada2018 自适应 和其他方差减少方法 ěcitep ji2019improved,liu2018zeroth 可以添加到 ězosgd 。ZO 在深度学习中特别突出的应用是分布式方法 ěcitep tang2019 分布式，hajinezhad2018 渐变 和黑盒对抗样本生成 ěcitep cai2021zerothorder,liu2018signsgd,chen2017zoo,liu2020primer 。ěcitep ye2018hessian,balasubramanian2022zeroth 使用 Hessian 的 ZO 估计进一步增强重要方向的优化。此外，还有一些 ZO 方法可以在不估计梯度的情况下进行优化 ěcitep golovin2020gradientless,mania2018simple,hinton2022forwardforward 。

ěsubsection 内存高效的反向传播 ělabel 秒：内存 _ 高效 _backprop 已经提出了几种算法来通过稀疏梯度 ěěcitep sun17meprop,wei2017minimal 、近似雅可比矩阵 ěěcitep abdel2008low，choromanski2017blackbox 和二次采样计算图 ěěcitep oktay2020 随机化，adelman2021 更快来有效地近似反向传播。然而，这些方法可能会为深度网络产生较大的近似误差。梯度检查点 ěcitep chen2016 培训 通过重新计算一些激活来降低反向传播的内存成本，但代价是速度明显变慢。FlashAttentioněcitep dao2022flash 注意 还通过重新计算注意力矩阵来降低内存成本。ěcitet dettmers2022gptint,dettmers2022bit 探索大型 LM 的权重和优化器状态的量化，这会导致训练和推理中的内存减少。

ěsubsection 大型语言模型的无梯度适配 BBT 和 BBTv2ěěcitep sun2022black,sun2022bbtv2 使用进化算法实现无梯度优化；然而，由于其对高维的敏感性，BBT 仅限于优化前缀的低维投影，并且他们专注于 RoBERTa-large size models 和 few-shot settings。LM 的“黑盒调整”中的其他工作侧重于在不更新模型的情况下优化离散提示，通过强化学习 ěcitep chai2022clip,deng2022rl 提示,diao2022black 、集成 ěcitep hou2022 提示提升 或迭代搜索 ěcitep 普拉萨德 2022grips 。

5 结论

我们已经证明，MeZO 可以跨许多任务和规模有效地优化大型 LM。进一步的实验表明，MeZO 可以优化不可微分的目标，而反向传播通常无法做到这一点。我们的理论说明了为什么 MeZO 在调整数十亿个参数时并没有灾难性地慢。作为一个限制，MeZO 采取了许多步骤来实现强大的性能。在这项工作中，我们没有探索将 MeZO 与其他内存高效方法相结合，例如 FlashAttention [14] 和量化 [16] 。我们希望将来对此进行调查。

我们很高兴探索 MeZO 在许多有前途的领域的适用性，包括但不限于：修剪、蒸馏、显着性、可解释性和用于微调的数据集选择。鉴于最近在调整大型 LM 以适应人类反馈方面取得的进展，不可微分目标是一个特别令人兴奋的领域。对这些有效的梯度估计如何影响不同应用程序的性能进行理论分析也很有趣。

6

致谢 感谢 Kaifeng Lyu、Abhishek Panigrahi、Nikunj Saunshi 和 Mengzhou Xia 提供的有用反馈。SA 和 SM 由 NSR、ONR、SRC 和西蒙斯基金会资助。JDL、AD 和 EN 承认 ARO 在 MURI 奖 W911NF-11-1-0304、斯隆研究奖学金、NSF CCF 2002272、NSF IIS 2107304、NSF CIF 2212262、ONR 青年研究者奖和 NSF 职业奖 2144994 下的支持. 这项工作也得到了国家科学基金会 (IIS-2211779) 的部分资助。

References

[1] Alekh Agarwal, Peter L. Bartlett, Pradeep Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, May 2012.

[2] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, 2021.

[3] Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, et al. Promptsource: An integrated development environment and repository for natural language prompts. *arXiv preprint arXiv:2202.01279*, 2022.

[4] Roy Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second PASCAL recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

[5] Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth PASCAL recognizing textual entailment challenge. In *TAC*, 2009.

[6] Raghu Bollapragada, Richard Byrd, and Jorge Nocedal. Adaptive sampling strategies for stochastic optimization. *SIAM Journal on Optimization*, 28(4):3312–3343, 2018.

[7] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in neural information processing systems*, volume 33, pages 1877–1901, 2020.

[9] HanQin Cai, Daniel McKenzie, Wotao Yin, and Zhenliang Zhang. Zeroth-order regularized optimization (zoro): Approximately sparse gradients and adaptive sampling. *SIAM Journal on Optimization*, 32(2):687–714, 2022.

[10] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

[11] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in neural information processing systems*, volume 30, 2017.

[12] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, 2019.

[13] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In *the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*, 2005.

[14] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359, 2022.

[15] Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *Sinn und Bedeutung*, volume 23, pages 107–124, 2019.

[16] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, 2022.

[17] Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit optimizers via block-wise quantization. In *International Conference on Learning Representations*, 2022.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[19] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*, 2022.

[20] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, 2019.

[21] John C. Duchi, Michael I. Jordan, Martin J. Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

[22] FairScale authors. Fairscale: A general purpose modular pytorch library for high performance and large scale training, 2021.

[23] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, 2021.

[24] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pages 2232–2241, 2019.

[25] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In *the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007.

[26] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[27] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008.

[28] José Grimm, Loïc Pottier, and Nicole Rostaing-Schmidt. *Optimal time and minimum space-time product for reversing a certain class of programs*. PhD thesis, INRIA, 1996.

[29] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, 2020.

[30] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[31] Kevin G Jamieson, Robert Nowak, and Ben Recht. Query complexity of derivative-free optimization. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

[32] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, 2013.

[33] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition, 2020.

[34] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, 2018.

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[36] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

[37] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021.

[38] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

[39] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

[40] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.

[41] Zhiyuan Li, Sadhika Malladi, and Sanjeev Arora. On the validity of modeling SGD with stochastic differential equations (SDEs). In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[42] Zhiyuan Li, Srinadh Bhojanapalli, Manzil Zaheer, Sashank Reddi, and Sanjiv Kumar. Robust training of neural networks using scale invariant architectures. In *International Conference on Machine Learning*, pages 12656–12684, 2022.

[43] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, 2022.

[44] Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5747–5763, 2020.

[45] Sijia Liu, Bhavya Kailkhura, Pin-Yu Chen, Paishun Ting, Shiyu Chang, and Lisa Amini. Zeroth-order stochastic variance reduction for nonconvex optimization. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[46] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[47] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, 2022.

[48] Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. *arXiv preprint arXiv:2210.05643*, 2022.

[49] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

[50] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[51] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[52] Vardan Papyan. The full spectrum of deepnet hessians at scale: Dynamics with sgd training and sample size. *arXiv preprint arXiv:1811.07062*, 2018.

[53] Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra. *Journal of Machine Learning Research*, 21(252):1–64, 2020.

[54] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.

[55] Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, 2019.

[56] Maxim Raginsky and Alexander Rakhlin. Information-based complexity, feedback and dynamics in convex programming. *IEEE Transactions on Information Theory*, 57(10):7036–7056, 2011.

[57] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

[58] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. 2011.

[59] Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

[60] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, 2021.

[61] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

[62] J.C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

[63] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021, 2020.

[64] Tianxiang Sun, Zhengfu He, Hong Qian, Yunhua Zhou, Xuanjing Huang, and Xipeng Qiu. BBTv2: Towards a gradient-free future with large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3916–3930, 2022.

[65] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855, 2022.

[66] Zhiwei Tang, Dmitry Rybin, and Tsung-Hui Chang. Zeroth-order optimization meets human feedback: Provable learning via ranking oracles, 2023.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

[68] Ellen M Voorhees and Dawn M Tice. Building a question answering test collection. In *the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000.

[69] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in neural information processing systems*, volume 32, 2019.

[70] Chong Wang, Xi Chen, Alexander J Smola, and Eric P Xing. Variance reduction for stochastic gradient optimization. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[71] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

[72] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.

[73] Yikai Wu, Xingyu Zhu, Chenwei Wu, Annie Wang, and Rong Ge. Dissecting hessian: Understanding common structure of hessian in neural networks. *arXiv preprint arXiv:2010.04261*, 2020.

[74] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pages 581–590, 2020.

[75] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[76] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

[77] Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*, 2018.

[78] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

A 算法消融

我们执行多次消融以选择最佳算法。按照 ZO 文献中的标准，我们认为主要的计算成本是前向传递的次数。在我们的例子中，前向传递的数量可能受到所采用的梯度步数、梯度累积的任何使用以及使用更多噪声样本以减少梯度估计的方差的影响。

我们观察到 MeZO 的性能随着步数的增加而单调提高，并且没有出现任何过拟合的情况。因此，在执行算法消融时，我们可以专注于不同算法的效率，而无需考虑隐式偏差。这也反映在我们的理论分析中。为了减轻计算负荷，我们固定了 $10,000$ 的前向传递数量，并比较了 RoBERTa-large 的许多不同算法在跨越情感分析、蕴含和主题分类的较小任务集上的效果：SST-2、SNLI 和 TREC。我们强调 $10,000$ 是一个小预算，仅用作将这些 ZO 算法相互比较的设置。我们发现在训练期间使用线性递减的学习率计划，就像在 [46] 中通过反向传播进行微调所做的那样，对 MeZO 没有帮助或伤害。同样，使用学习率预热会在这三个任务上产生相同的结果。为简单起见，我们对以下所有实验使用不预热的恒定学习率计划。我们使用 $k = 16$ 进行少量实验，并对 5 个种子的结果进行平均。

Experiment	Hyperparameters	Values
MeZO	Batch size	$\{16, 64\}$ ×
	Learning rate	$\{1e{-}5, 1e{-}6, 1e{-}7\}$ ×
	ϵ	$\{1e{-}3, 1e{-}5\}$ ×
	Weight Decay	$\{0, 0.1\}$

Table 4: 我们的消融实验中使用的超参数网格。为简单起见，我们使用恒定的学习率计划。

A.1 提示

我们研究添加提示是否对 MeZO 优化网络的能力至关重要。我们使用来自 Gao et al. [23] 的提示。Malladi et al. [48] 声称提示使优化轨迹表现良好，但我们注意到当前论文考虑了 RoBERTa-large 和大型自回归模型，而之前的工作仅研究了 RoBERTa-base。我们注意到内核行为与我们在 Section 4 中的理论设置之间的相似性。MeZO 成功执行了据报告在 Malladi et al. [48] 中未表现出内核行为的任务，因此我们调查是否需要提示。

	SST-2	SNLI	TREC
Prompt	89.6 (1.2)	65.1 (6.2)	66.7 (6.2)
No Prompt	51.9 (2.9)	34.8 (2.1)	19.5 (9.0)

Table 5: 使用 MeZO 在有提示和无提示的情况下微调模型的实验。

两个实验都遵循 Table 4 中的网格，但我们还扩展了网格以包括 $1e-4$ 的学习率，用于无提示情况。作为这些实验的结果，我们将以下所有实验的设置固定为基于提示的微调。

A.2 样品时间表

可以在第 t 步对 n_t 噪声向量进行采样，并使用 n_t -SPSA 计算梯度估计。 Bollapragada et al. [6], Cai et al. [9] 中提出了类似的想法。我们研究了消融设置中线性增加和恒定采样计划的影响。线性增加计划的直觉是优化器在接近最小值时可能需要更高的保真度梯度。增加 z 的数量可以通过减小梯度方差来加速优化，但这样做也会增加每个优化步骤所需的前向传递次数，因此需要权衡取舍。我们注意到，增加 z 的数量应该伴随着学习率的比例缩放，类似于 [26] 中提出的线性缩放规则（理论证明可以遵循 SDE 技术 [41] ）。Table 6 在一个时间表中没有显示出相对于另一个时间表的一致优势，并且它表明在 n -SPSA 中增加 n 同时固定允许的前向传递数量最多只能带来边际收益。

B MeZO 变体

将一阶优化的思想转移到增强 ZO 算法的历史悠久。下面，我们重点介绍 MeZO 的几个变体，它们没有实现与 Algorithm 1 中介绍的算法一样高的性能。

n	Schedule	SST-2	SNLI	TREC
$n=1$	Constant	89.6 (1.2)	65.1 (6.2)	66.7 (6.2)
$n=4$	Constant	89.5 (1.1)	68.6 (3.2)	62.3 (5.6)
$n=4$	Linear	89.6 (1.4)	65.3 (6.4)	66.1 (5.5)
$n=16$	Constant	90.4 (0.7)	67.0 (3.4)	62.8 (6.3)
$n=16$	Linear	88.9 (1.2)	62.8 (5.9)	64.2 (5.3)

Table 6: 使用 MeZO 和 n 的不同时间表的实验。我们根据 $\boldsymbol{z}$ 的采样数按比例调整学习率。

B.1 内存高效的 n-SPSA

我们强调了 MeZO 如何在 Algorithm 2 中为 $n>1$ 高效地执行 n -SPSA (Definition 1)。特别是，如果对 n $\boldsymbol{z}$ 向量进行采样并对投影梯度进行平均，我们需要存储 $2n$ 额外的标量：随机种子和投影梯度。关于扰动单个权重与整个权重矩阵的相同警告在这里仍然适用（参见 Section 2 ）。

Algorithm 2: MeZO 与 $n>1$

Require : parameters $\boldsymbol{\theta}\in\mathbb{R}^d$, loss $\mathcal{L}:\mathbb{R}^d\to\mathbb{R}$, step budget T , perturbation scale ϵ , batch size B learning rate schedule $\{\eta_t\}$, n for n -SPSA estimate (Definition 1)

for $t=1,...,T$ **do**
  `seeds, projected_grads` $\leftarrow$ `[]` ▷ `Will each contain` n `scalars`
  for $j=1,...,n$ **do**
    Sample batch $\mathcal{B}\subset\mathcal{D}^B$ and random seed s
    $\boldsymbol{\theta}\leftarrow$ `PerturbParameters(` $\boldsymbol{\theta},\epsilon,s$`)`
    $\ell_+\leftarrow\mathcal{L}(\boldsymbol{\theta};\mathcal{B})$
    $\boldsymbol{\theta}\leftarrow$ `PerturbParameters(` $\boldsymbol{\theta},-2\epsilon,s$`)`
    $\ell_-\leftarrow\mathcal{L}(\boldsymbol{\theta};\mathcal{B})$
    $\boldsymbol{\theta}\leftarrow$ `PerturbParameters(` $\boldsymbol{\theta},\epsilon,s$`)` ▷ `Reset parameters`
    `projected_grad` $\leftarrow(\ell_+-\ell_-)/(2\epsilon)$
    `projected_grads[j]` $\leftarrow$ `projected_grad`
    `seeds[j]` $\leftarrow s$
  end
  for $j=1,...,n$ **do**
    Reset random number generator with seed `seeds[j]`
    for $\theta_i\in\boldsymbol{\theta}$ **do**
      $z\sim\mathcal{N}(0,1)$
      $\theta_i\leftarrow\theta_i-(\eta_t/n)*$`projected_grads[j]`$*z$ ▷ `Avg grad for` $\boldsymbol{z}_1,...,\boldsymbol{z}_n$
    end
  end
end

Subroutine `PerturbParameters(` $\boldsymbol{\theta}$ `,` ϵ `,` s`)`
  Reset random number generator with seed s ▷ `For sampling` z
  for $\theta_i\in\boldsymbol{\theta}$ **do**
    $z\sim\mathcal{N}(0,1)$
    $\theta_i\leftarrow\theta_i+\epsilon z$ ▷ `Modify parameters in place`
  end
  return $\boldsymbol{\theta}$

B.2 使用梯度历史增强 MeZO

n -SPSA 算法仅提供梯度估计，随后可用于代替任何基于梯度的优化器中的梯度。许多流行的优化器，例如具有动量的 Adam 和 SGD，需要存储一些关于梯度的历史信息（例如，移动平均线）。此要求导致此类算法需要 2× 或 3× SGD 所需的内存。

然而，MeZO 的一个优点是可以在每一步重新计算梯度历史，而不需要太多额外的内存。参考 Algorithm 1，注意梯度只需要 `projected_grad` 和随机种子 s 用于计算扰动 $\boldsymbol{z}$。`projected_grad` 可以从两个扰动损失 ℓ_1 和 ℓ_2 重新计算，因此我们只需要每步存储 3 个标量来重现梯度历史（即，在训练期间最多 $3T$ 个标量）。这大大减少了使用 Adam 或动量而不是普通 SGD 通常需要的额外内存开销。

Table 16 说明 MeZO -Adam 有时可以提高 MeZO 的性能，尽管每个梯度步骤都需要额外的计算（但不需要额外的前向传递）。我们将其留给未来的工作来调查 MeZO -Adam 何时可能比 MeZO 更有用。

Experiment	Hyperparameters	Values
MeZO-Adam	Batch size	64
	Learning rate	$\{1e{-}6, 1e{-}5, 1e{-}4, 5e{-}4, 1e{-}3\}$
	ϵ	$1e{-}3$
	Weight Decay	0

Table 7: 用于 MeZO -Adam 的超参数网格。为简单起见，我们使用恒定的学习率计划。

B.3 修改 MeZO 的方差

我们在 Section 4 中的理论概述了一个众所周知的事实，即随机梯度估计的方差会影响优化率。ZO 方法可以与标准方差减少技术相结合，以可能提高优化速度。例如，Liu et al. [45] 设计了一个方差减少的 ZO 算法，类似于 SVRG [32]，以提高收敛速度。下面，我们展示了几种减少方差的方法（例如，使用梯度范数）可以以内存高效的方式实现。但是，在控制前向传递（即函数查询）的总预算时，这些方法的性能不如 MeZO 。尽管如此，我们还是展示了它们以证明 MeZO 可以轻松适应，我们建议这些方法可能对优化更复杂的目标有用。

首先，我们定义了一个通用的 SPSA 估计，它具有相同的期望值（即真实梯度）但具有缩放方差。

Definition 6 (Variance-Modified SPSA)**.** *给定矩阵 $D = \mathrm{diag}(\boldsymbol{d})$，方差修改 SPSA 计算*

$$\tilde{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B}) = \frac{\mathcal{L}(\boldsymbol{\theta} + \epsilon(\boldsymbol{d}^{-1} \odot \boldsymbol{z});\mathcal{B}) - \mathcal{L}(\boldsymbol{\theta} - \epsilon(\boldsymbol{d}^{-1} \odot \boldsymbol{z});\mathcal{B})}{2\epsilon}(\boldsymbol{d} \odot \boldsymbol{z})$$

，其中 $\boldsymbol{d} \in \mathbb{R}^d$ 具有非零条目，$\boldsymbol{d}^{-1}$ 表示坐标倒数。

上述 SPSA 变体是梯度的无偏估计量，因为 $\mathbb{E}[\tilde{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})] = \mathbb{E}[D^{-1}\boldsymbol{z}\boldsymbol{z}^\top D\nabla\mathcal{L}(\boldsymbol{\theta};\mathcal{B})] = \mathbb{E}[\nabla\mathcal{L}(\boldsymbol{\theta};\mathcal{B})]$。我们将从经典方法（即“控制变量”）中汲取灵感，并选择 $\boldsymbol{d}$ 作为具有梯度范数或参数范数的块向量 [70]。为了选择参数组，我们按层拆分模型，保持嵌入和头部分离（即 RoBERTa-large 有 $24 + 2 = 26$ 参数组）。无需消耗额外内存即可直接测量参数范数。我们可以在不执行反向传播的情况下测量梯度范数，如下所示。

Proposition 1 (ZO Estimate of Gradient Norm of ℓ th Layer)**.** *将 $\boldsymbol{z}_\ell$ 定义为 $z \sim \mathcal{N}(0,1)$ 在每个坐标对应于 ℓ th layer 和 0 中的参数。然后，我们可以估计损失 w.r.t. 梯度的范数。ℓ 层 $\nabla_{\boldsymbol{\theta}_\ell}$ 作为*

$$\|\nabla_{\boldsymbol{\theta}_\ell}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\|_2 \approx \left|\frac{\mathcal{L}(\boldsymbol{\theta} + \epsilon\boldsymbol{z}_\ell;\mathcal{B}) - \mathcal{L}(\boldsymbol{\theta} - \epsilon\boldsymbol{z}_\ell;\mathcal{B})}{2\epsilon}\right|$$

与 SPSA 一样，增加 ℓ 的每个值的采样 $\boldsymbol{z}_\ell$ 的数量并对结果进行平均会减少估计的方差。此估计的基本原理是对于任何矢量 $\boldsymbol{v}$，$\mathbb{E}_{\boldsymbol{z}}[(\langle\boldsymbol{v},\boldsymbol{z}\rangle)^2] = \|\boldsymbol{v}\|_2^2$ 为高斯 $\boldsymbol{z}$。很明显，这个估计可以以内存高效的方式计算，尽管它需要 $2L$ 前向传递来计算 L 参数组的梯度范数。

我们在下面展示了修改方差的实验结果。我们遵循消融设置并使用 $10,000$ 步 (Appendix A) 的固定预算。通常，使用梯度范数来减少方差会大大损害性能 (Table 8)。如果我们“作弊”并允许通过网络进行一次反向传播来估计梯度范数，那么我们会发现使用梯度范数减少方差不会显着损害或帮助性能。使用参数范数修改方差，类似于分层自适应速率方法，不会显着影响 MeZO (Table 9) 的性能。

我们的观察是，通过将 $\boldsymbol{d}$ 设置为梯度范数来降低方差不会改善优化。该经验结果与 Section 4 中的说明一致，即直接方差分析（产生对参数 d 数量的依赖性）并不是研究使用 MeZO 进行微调时优化率的最佳视角。我们在 Theorem 1 和 Lemma 3 中的有效排名视图可能是微调动态的更好表征。我们留待未来的工作去探索这些方法是否对其他更复杂的目标有用。

Recompute $\boldsymbol{d}$	ZO estimate of $\boldsymbol{d}$	SST-2	SNLI	TREC
Baseline MeZO (Algorithm 1)		89.6 (1.2)	65.1 (6.2)	66.7 (6.2)
✗	✗	89.7 (0.8)	65.2 (5.2)	64.3 (6.4)
✗	✓	87.0 (2.5)	49.6 (9.2)	32.6 (7.7)
✓	✓	79.0 (10.3)	48.9 (2.2)	38.7 (7.5)

Table 8: 使用 $\boldsymbol{d}$ 作为梯度范数修改 MeZO 方差的实验（参见 Definition 6 ）。我们有时会在每个纪元开始时重新计算 $\boldsymbol{d}$ 或使用 Proposition 1 来估计 $\boldsymbol{d}$ 而无需反向传播。

Recompute $\boldsymbol{d}$	SST-2	SNLI	TREC
Baseline MeZO (Algorithm 1)	89.6 (1.2)	65.1 (6.2)	66.7 (6.2)
✗	89.2 (2.1)	65.4 (4.2)	64.8 (5.6)
✓	88.2 (4.7)	65.2 (4.0)	64.7 (5.5)

Table 9: 使用 $\boldsymbol{d}$ 作为参数范数修改 MeZO 方差的实验（参见 Definition 6 ）。我们有时会在每个纪元开始时重新计算 $\boldsymbol{d}$ 。

B.4 修改 MeZO 的期望值

上面的实验表明，修改 MeZO 的方差并不能始终如一地加速其收敛。然而，对 Definition 6 的简单修改也允许我们改变对 MeZO 的期望。这可用于有效地估计基于坐标的归一化梯度优化器更新（例如 Adam）。

Definition 7 (Expectation-Modified SPSA)**.** *给定矩阵 $D = \mathrm{diag}(\boldsymbol{d})$ ，方差修改 SPSA 计算*

$$\tilde{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B}) = \frac{\mathcal{L}(\boldsymbol{\theta} + \epsilon(\boldsymbol{d}^{-1} \odot \boldsymbol{z});\mathcal{B}) - \mathcal{L}(\boldsymbol{\theta} - \epsilon(\boldsymbol{d}^{-1} \odot \boldsymbol{z});\mathcal{B})}{2\epsilon}\boldsymbol{z}$$

其中 $\boldsymbol{d} \in \mathbb{R}^d$ 。

现在，我们看到 $\tilde{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B}) = \mathbb{E}[D^{-1}\boldsymbol{z}\boldsymbol{z}^\top\nabla\mathcal{L}(\boldsymbol{\theta};\mathcal{B})]$ 因此 SPSA 估计不再是 $\nabla\mathcal{L}(\boldsymbol{\theta})$ 的无偏估计。例如，如果我们选择 $\boldsymbol{d}$ 作为梯度范数，那么 SPSA 可以估计归一化梯度。Tang et al. [66] 中的并行工作给出了归一化梯度的另一个 ZO 估计，同时假设只能访问输入的排名（而不是我们设置中可用的嘈杂函数评估）。我们发现估计归一化梯度的效果不如直接估计梯度 (Table 10)。无论如何，我们提出这个算法是为了强调任何对梯度的坐标操作都可以以节省内存的方式应用。

Method	SST-2	SNLI	TREC
Baseline MeZO (Algorithm 1)	89.6 (1.2)	65.1 (6.2)	66.7 (6.2)
Estimate of normalized gradient (Definition 7)	88.0 (1.2)	60.0 (2.4)	44.0 (14.0)

Table 10: 使用 $\boldsymbol{d}$ 作为梯度范数修改 MeZO 的期望的实验（参见 Definition 7 ）。我们使用梯度范数 (Proposition 1) 的 ZO 估计。

C 内存分析

反向传播的计算内存权衡分析起来很复杂。Griewank and Walther [27] 提供了对该问题的严格理论处理。我们凭经验测量了常用大型语言模型的不同方法的内存消耗，但在这里我们希望提供不同梯度估计算法的更严格的比较，独立于用于实现它们的软件。下面，我们总结了一些关键点，可以帮助读者理解 MeZO 计算内存权衡与反向传播的比较。

给定一个网络，执行反向传播的第一步是将模型分解为易于区分的块。我们注意到这种分解不是唯一的。对于每个块，可以选择在前向传递期间缓存结果输出（从而消耗内存），或者在需要时重新计算输出（从而消耗计算）。以下命题改编自 Griewank and Walther [27] 中的规则 21，体现了这种权衡。

Proposition 2 (Time-Memory Tradeoff for Backpropagation, Griewank and Walther [27])**.** *考虑包含 N 位的网络。对于任何时间-内存权衡超参数 $c = O(1)$ ，存在一个反向传播算法，该算法在时间 $O(cN)$ 中运行并消耗与 $O(N^{1/c})$ 成比例的内存。*

Grimm et al. [28] 还为内存时间产品提供了明确的界限。请注意，流行的梯度检查点 [10] 方法允许以有限的精度调整 c（即，不能总是进一步拆分可微块并观察节省）。 Chen et al. [10] 中的实验选择 $c = 2$ 来实现 $O(\sqrt{N})$ 内存同时消耗 $O(2N)$ 计算。在极端情况下，梯度检查点允许使用 $O(N \log N)$ 计算和 $O(\log N)$ 内存。

MeZO 始终消耗 $2N$ 计算和 $O(1)$ 内存，因此它在与梯度检查点相同的内存成本下具有更高的计算效率。我们在 Section 2 中的阐述讨论了我们可以一起扰动参数组以节省时间，同时消耗额外的内存。但是，我们在这里不考虑该变体，因为它位于计算内存帕累托曲线中间的某个位置，我们无法推断反向传播会做什么。特别是，MeZO 可以以不同于反向传播的方式拆分组，因为 MeZO 不要求每个参数组都易于区分，因此很难沿着整个帕累托曲线比较这两种算法。

我们还比较了 $c = 1$ 情况下的反向传播（即，在前向传播过程中存储所有内容）。存储所有内容时，反向传播会消耗 $O(N)$ 时间和 $O(N)$ 内存。因此，在权衡的这一端，SPSA 比反向传播消耗更多的时间和更少的内存。

与梯度检查点不同，MeZO 仅计算梯度的近似值。这种近似仅对提示微调有用，因此不如梯度检查点广泛有用。还有其他方法可以用比梯度检查点更少的内存消耗来近似梯度（请参阅相关工作部分），尽管尚不清楚这些算法的内存消耗与 MeZO 相比如何。

D 实验设置

D.1 数据集

对于 RoBERTa-large，我们考虑分类数据集：SST-2 [61]、SST-5 [61]、TREC [68]、MNLI [71]、SNLI [7] 和 RTE [13, 4, 25, 5]。我们按照 Malladi et al. [48] 将测试集限制为 $1,000$ 示例以进行快速迭代。对于训练和验证，我们有两个设置：$k = 16$ 和 $k = 512$，这意味着我们每个类有 16 或 512 个示例用于训练和验证。

对于 OPT 实验，我们考虑 SuperGLUE 数据集 collection [69]，包括：BoolQ [12]、CB [15]、COPA [58]、MultiRC [34]、ReCoRD [77]、RTE [13, 4, 25, 5]、WiC [55] 和 WSC [38]。我们还包括 SST-2 [61] 和两个问答 (QA) 数据集，SQuAD [57] 和 DROP [20]。我们随机抽取 1,000 个样本进行训练，500 个样本进行验证，1,000 个样本进行测试。

D.2 提示

表 11 显示了我们微调 RoBERTa-large 的一组下游任务和提示，这些任务和提示改编自 [23]。

Dataset	C	Type	Prompt	Label words
SST-2	2	sentiment cls.	$< S_1 >$ It was `[MASK]` .	{ great, terrible }
SST-5	5	sentiment cls.	$< S_1 >$ It was `[MASK]` .	{ great, good, okay, bad, terrible }
TREC	6	topic cls.	`[MASK]` : $< S_1 >$	{ Description, Expression, Entity, Human, Location, Number }
MNLI	3	NLI	$< S_1 >$? `[MASK]` , $< S_2 >$	{ Yes, Maybe, No }
SNLI	3	NLI	$< S_1 >$? `[MASK]` , $< S_2 >$	{ Yes, Maybe, No }
RTE	2	NLI	$< S_1 >$? `[MASK]` , $< S_2 >$	{ Yes, No }

Table 11: 我们在 RoBERTa 大型实验（Table 16 和 Figure 2）中使用的数据集的提示。提示改编自 [23]，包括一个模板和一组可以填充 `[MASK]` 标记的标签词。 $< S_1 >$ 和 $< S_2 >$ 指的是第一个和第二个（如果有的话）输入语句。

表 12 演示了我们用于 OPT 的提示。请注意，在 OPT 实验中，我们有三种类型的任务：分类、多项选择和问题回答。提示从 GPT-3 [8] 和 PromptSource 中采纳，稍有改动 [3]。

D.3 超参数

我们使用 Table 13 中的超参数在 RoBERTa-large（Table 16 和 Figure 2）上进行 MeZO 实验。Appendix A 中的实验通知网格；特别是，ϵ 的选择似乎不会显着影响性能，并且使用更大的批处理大小始终会产生更快的优化。我们将 Table 14 中的超参数用于 OPT 上的 MeZO 实验。

Dataset	Type	Prompt
SST-2	cls.	`<text>` It was terrible / great
RTE	cls.	`<premise>` Does this mean that " `<hypothesis>` " is true? Yes or No? Yes / No
CB	cls.	Suppose `<premise>` Can we infer that " `<hypothesis>` "? Yes, No, or Maybe? Yes / No / Maybe
BoolQ	cls.	`<passage>` `<question>` ? Yes / No
WSC	cls.	`<text>` In the previous sentence, does the pronoun " `<span2>` " refer to `<span1>` ? Yes or No? Yes / No

WIC	cls.	Does the word " `<word>` " have the same meaning in these two sentences? Yes, No?
		`<sent1>`
		`<sent2>`
		Yes / No

MultiRC	cls.	`<paragraph>` Question: `<question>` I found this answer " `<answer>` ". Is that correct? Yes or No? Yes / No
COPA	mch.	`<premise>` so/because `<candidate>`
ReCoRD	mch.	`<passage>` `<query>.replace("@placeholder", <candidate>)`
SQuAD	QA	Title: `<title>` Context: `<context>` Question: `<question>` Answer:
DROP	QA	Passage: `<context>` Question: `<question>` Answer:

Table 12: 我们在 OPT 实验中使用的数据集的提示。共有三种类型的任务：分类 (cls.)、多项选择 (mch.) 和问答 (QA)。Prompts 是从 GPT-3 [8] 和 PromptSource [3] 中采纳的，稍有改动。`<text>` 表示来自数据集的输入，Yes 表示标签词。对于多项选择任务的推理，我们在提示中放入不同的候选者并计算每个候选者的平均对数似然，并选择得分最高的候选者。对于 QA 任务的推理，我们使用贪婪解码来生成答案。

关于学习率调度和提前停止，我们使用线性学习调度进行所有反向传播实验的微调和所有 MeZO 实验的恒定学习率。对于 RoBERTa 实验，我们每 1/10 的总训练步骤在验证集上评估模型并保存最佳验证检查点。所有 FT 实验都使用 1K 步，MeZO 实验使用 100 K 步。对于 OPT 实验，我们每 1/5 的总训练步骤在验证集上评估模型并保存最佳验证检查点。所有 FT 实验训练 5 个时期，所有 MeZO 实验都使用 20 K 步。请注意，FT 实验大多在 5 个时期内收敛，但我们观察到 MeZO 的性能仍然可以通过更多的训练步骤来提高。

D.4 建模和实施

对于 RoBERTa 实验，我们遵循 [23] 用于屏蔽语言模型的基于提示的微调范例。有关更多详细信息，请参阅原始论文。

在 OPT 实验中，对于分类任务，我们训练类似于 [23] 的模型，即取标签词对应的 logits，对其应用交叉熵损失；对于多项选择任务和生成任务（QA），我们只保留正确的候选人并使用教师强制训练正确的例子。我们只保留候选部分的代币损失，排除提示部分。

对于分类和多项选择任务的 OPT 推理，我们使用该模型获得所有候选词/标签词的平均对数似然（按标记），并预测平均对数似然最高的词。对于生成任务，我们使用贪心解码来生成答案。

对于上下文学习，我们在上下文中使用了 32 个示例。我们也尝试在上下文中填充尽可能多的示例，但不会提高性能，有时会导致不稳定的结果。因此我们保留了 32 个示例的结果。

对于分类任务的线性探测，我们采用输出特征并使用 `scipy` 包来训练线性分类器。对于多项选择任务和生成任务，我们发现这会导致较差的结果，因为输出空间是整个词汇表；相

Experiment	Hyperparameters	Values
MeZO	Batch size	64
	Learning rate	$\{1e{-}7, 1e{-}6, 1e{-}5\}$
	ϵ	$1e{-}3$
	Weight Decay	0
MeZO (prefix)	Batch size	64
	Learning rate	$\{1e{-}2, 5e{-}3, 1e{-}3\}$
	ϵ	$1e{-}1$
	Weight Decay	0
	# prefix tokens	5
MeZO (LoRA)	Batch size	64
	Learning rate	$\{1e{-}5, 5e{-}5, 1e{-}4\}$
	ϵ	$1e{-}3$
	Weight Decay	0.1
	(r, α)	$(8, 16)$
FT with Adam	Batch size ($k = 16$)	$\{2, 4, 8\}$
	Batch size ($k = 512$)	$\{8, 16, 32\}$
	Learning Rates	$\{1e{-}5, 3e{-}5, 5e{-}5\}$
	Weight Decay	0
FT with SGD	Batch size ($k = 16$)	$\{2, 4, 8\}$
	Batch size ($k = 512$)	$\{8, 16, 32\}$
	Learning Rates	$\{1e{-}4, 5e{-}4, 1e{-}3, 5e{-}3, 1e{-}2\}$
	Weight Decay	0
FT (prefix)	Batch size	$\{8, 16, 32\}$
	Learning Rates	$\{1e{-}2, 3e{-}2, 5e{-}2\}$
	Weight Decay	0
	# prefix tokens	5
FT (LoRA)	Batch size	$\{4, 8, 16\}$
	Learning Rates	$\{1e{-}4, 3e{-}4, 5e{-}4\}$
	(r, α)	$(8, 16)$

Table 13: 用于 RoBERTa 大型实验的超参数网格。MeZO 采用恒定学习率调度，FT 采用线性调度。所有 FT 实验都使用 1K 步，MeZO 实验使用 100 K 步。我们每 1/10 个总训练步骤检查一次验证性能。

反，我们进行头部调整，除了 LM 投影头之外，整个模型都是固定的。我们使用 8 的批量大小和 $\{1e{-}4\ 5e{-}4\}$ 的学习率，并为 5 个时期训练头部。

对于 30B 和 66B OPT 模型的实验，我们主要遵循 OPT 超参数，只是我们不评估中间验证性能并直接使用最后一个检查点进行评估，因为大型模型的中间检查点的存储成本很高。

D.5 参数高效微调

为每个下游任务微调和存储大型语言模型的副本非常昂贵。参数高效微调 (PEFT) 技术缓解了这个问题：PEFT 不是调整所有模型参数，而是仅调整少量附加参数（通常少于 1 % ）并且通常可以获得相当或更好的性能 [40, 37, 19] 。ZO 优化器与 PEFT 方法兼容，因为 ZO 可以对模型参数的任何子集进行操作。我们对以下两种常见的 PEFT 方法感兴趣，专为 transformers 设计 [67] 。

LoRA [30] 在微调期间向线性层添加可调低秩增量。假设线性层在使用 $\boldsymbol{W} \in \mathbb{R}^{m \times n}$ 进行预训练期间执行 $\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$ 。微调时，LoRA 引入了两个较小的矩阵 $\boldsymbol{A} \in \mathbb{R}^{m \times r}$ 和 $\boldsymbol{B} \in \mathbb{R}^{r \times n}$ 使得 $r \ll \min(m, n)$ 。然后将线性层计算为

$$\left(\boldsymbol{W} + \frac{\alpha}{r}\boldsymbol{A}\boldsymbol{B}\right)\boldsymbol{x} + \boldsymbol{b} \tag{6}$$

，其中 r 和 α 是超参数。$\boldsymbol{A}$ 和 $\boldsymbol{B}$ 在下游任务上接受训练，而 $\boldsymbol{W}$ 冻结在其预训练值。在 transformers 中，对线性层的这种修改应用于每个注意力层的查询和值操作。根据经验，r 可以非常小，因此微调期间可训练的参数数量很少。我们选择 $r = 8$ 和 $\alpha = 16$ 。

Experiment	Hyperparameters	Values
MeZO	Batch size	16
	Learning rate	$\{1e{-}6, 1e{-}7\}$
	ϵ	$1e{-}3$
MeZO (prefix)	Batch size	16
	Learning rate	$\{1e{-}2, 1e{-}3\}$
	ϵ	$1e{-}1$
	# prefix tokens	5
MeZO (LoRA)	Batch size	16
	Learning rate	$\{1e{-}4, 5e{-}5\}$
	ϵ	$1e{-}2$
	(r, α)	$(8, 16)$
FT with Adam	Batch size	8
	Learning Rates	$\{1e{-}5, 5e{-}5, 8e{-}5\}$

Table 14: 用于 OPT 实验的超参数网格。所有权重衰减都设置为 0 。FT 使用 5 个时期和线性计划学习率，MeZO 使用 20 K 步和恒定学习率。我们检查验证性能并每 1/5 总训练步骤保存最佳检查点。

Prefix-tuning [40] 在每一层添加 m 可调表示的前缀，并冻结模型的其余部分。这些表示被添加为新的键和值，并在注意力操作期间被视为额外的上下文。我们通过从词汇表中随机抽取标记并将它们传递给 LLM 以在不同的注意力层获取它们的键和值来初始化这些可调表示。我们发现这对于使 MeZO 的前缀调整稳定至关重要，并且此技巧还可以提高反向传播前缀调整的性能，如 Table 15 所示。我们还在 [40] 中尝试了重新参数化技巧，这对 MeZO 训练没有帮助。在我们的实验中，我们发现 $m = 5$ 足以在大多数任务上实现良好的性能。

我们还表明 MeZO 与参数有效的微调方法兼容，例如前缀调整和 LoRA。令人惊讶的是，当调整少得多的参数时，MeZO 的性能并没有显着提高，正如人们可能从经典分析中所期望的那样（参见 Section 4 ）。因此，我们在 Section 4 中的理论分析表明，ZO-SGD 的收敛速度不依赖于微调期间的参数维度。

Task	SST-2	SST-5	SNLI	MNLI	RTE	TREC
Type	—— sentiment ——		—— natural language inference ——			— topic —
FT (prefix, random init)	90.7 (1.7)	47.2 (2.0)	70.7 (2.8)	62.6 (3.3)	63.5 (4.4)	83.4 (4.7)
FT (prefix, real act init)	91.9 (1.0)	47.7 (1.1)	77.2 (1.3)	66.5 (2.5)	66.6 (2.0)	85.7 (1.3)

Table 15: 前缀调整消融。我们比较随机初始化的前缀和真实的单词激活前缀。使用真实词激活显着优于随机初始化。

D.6 以不可区分的目标进行训练

最大化 RoBERTa 大型模型精度的实验均使用与 Table 13 中的 MeZO 相同的网格进行。

对于以 F1 为目标的 SQuAD 上的 OPT 实验，我们使用批量大小 16 。对于 MeZO ，我们使用 $\{1e{-}6, 5e{-}6, 1e{-}5\}$ 和 $\epsilon = 1e{-}3$ 的学习率。对于 MeZO（前缀），我们使用 $\{1e{-}1, 5e{-}2, 1e{-}2\}$ 和 $\epsilon = 1e{-}1$ 的学习率。

D.7 内存分析

在内存分析中，我们使用 Huggingface 的 `transformers` [72] 包的标准实现。我们没有打开任何高级内存节省选项，例如梯度检查点。我们将每个设备的批量大小设置为 1 ，以测试使用特定优化算法运行模型的最低硬件要求。对于多 GPU 反向传播，我们使用 PyTorch 提供的完全分片数据并行 (FSDP) [22] [54] 。对于多 GPU MeZO ，我们使用 `transformers` 大型模型的多 GPU 推理。我们使用 Nvidia 的 `nvidia-smi` 命令来监控 GPU 内存使用情况。如果 GPU 至少 100 步没有出现内存不足错误，我们称运行“成功”。

E 更多实验结果

E.1 RoBERTa-大型实验

Table 16 包含对应于 Figure 2 的详细数字，还报告了 MeZO -Adam 的表现。

Task	SST-2	SST-5	SNLI	MNLI	RTE	TREC
Type	—— sentiment ——		—— natural language inference ——			— topic —
Zero-shot	79.0	35.5	50.2	48.8	51.4	32.0
Gradient-free methods: $k = 16$						
LP	76.0 (2.8)	40.3 (1.9)	66.0 (2.7)	56.5 (2.5)	59.4 (5.3)	51.3 (5.5)
MeZO	90.5 (1.2)	45.5 (2.0)	68.5 (3.9)	58.7 (2.5)	64.0 (3.3)	76.9 (2.7)
MeZO (LoRA)	91.4 (0.9)	43.0 (1.6)	69.7 (6.0)	64.0 (2.5)	64.9 (3.6)	73.1 (6.5)
MeZO (prefix)	90.8 (1.7)	45.8 (2.0)	71.6 (2.5)	63.4 (1.8)	65.4 (3.9)	80.3 (3.6)
MeZO-Adam	90.4 (1.4)	45.4 (1.5)	74.1 (2.7)	64.3 (0.8) †	59.2 (11.1) †	78.3 (1.4)
Gradient-based methods: $k = 16$						
FT	91.9 (1.8)	47.5 (1.9)	77.5 (2.6)	70.0 (2.3)	66.4 (7.2)	85.0 (2.5)
FT (LoRA)	91.4 (1.7)	46.7 (1.1)	74.9 (4.3)	67.7 (1.4)	66.1 (3.5)	82.7 (4.1)
FT (prefix)	91.9 (1.0)	47.7 (1.1)	77.2 (1.3)	66.5 (2.5)	66.6 (2.0)	85.7 (1.3)
Gradient-free methods: $k = 512$						
LP	91.3 (0.5)	51.7 (0.5)	80.9 (1.0)	71.5 (1.1)	73.1 (1.5)	89.4 (0.5)
MeZO	93.3 (0.7)	53.2 (1.4)	83.0 (1.0)	78.3 (0.5)	78.6 (2.0)	94.3 (1.3)
MeZO (LoRA)	93.4 (0.4)	52.4 (0.8)	84.0 (0.8)	77.9 (0.6)	77.6 (1.3)	95.0 (0.7)
MeZO (prefix)	93.3 (0.1)	53.6 (0.5)	84.8 (1.1)	79.8 (1.2)	77.2 (0.8)	94.4 (0.7)
MeZO-Adam	93.3 (0.6)	53.9 (0.8)	85.3 (0.8)	79.6 (0.4)	79.2 (1.2)	95.1 (0.3)
Gradient-based methods: $k = 512$						
FT	93.9 (0.7)	55.9 (0.9)	88.7 (0.8)	84.4 (0.8)	82.7 (1.4)	97.3 (0.2)
FT (LoRA)	94.2 (0.2)	55.3 (0.7)	88.3 (0.5)	83.9 (0.6)	83.2 (1.3)	97.0 (0.3)
FT (prefix)	93.7 (0.3)	54.6 (0.7)	88.3 (0.7)	83.3 (0.5)	82.5 (0.8)	97.4 (0.2)

Table 16: 在 RoBERTa-large（350M 参数）上进行实验。LP：线性探测；ZO 、ZO (LoRA) 和 ZO (prefix)：我们的内存高效 ZO-SGD (Section 2.1) 分别具有全参数调整、LoRA 和前缀调整；FT：与亚当进行微调。所有报告的数字都是平均准确度（标准偏差）。所有实验都使用提示 (Appendix D.2)。ZO 的性能大大优于零射击和 LP，并以更少的内存成本接近 FT 性能。

LP- MeZO 我们还将 MeZO 与执行线性探测进行比较，然后通过 MeZO 执行微调，遵循类似的微调建议 Kumar et al. [36] 。我们使用 Table 13 中描述的 MeZO 网格。请注意，与 Table 16 中报告的检查点不同，此处使用的线性探测检查点提前停止。我们通过限制迭代次数（从 5000 到 1000 ）和增加 `scipy` 求解器中的收敛容差（从 1e−4 到 0.01 ）来启发式地实现提前停止。对一些设置的实验表明，LP- MeZO 有时可以在不增加内存消耗的情况下提高性能（参见 Table 17 ）。然而，有时，首先进行线性探测会严重损害性能。

Task	SST-2	SST-5	SNLI	TREC
Zero-shot	79.0	35.5	50.2	32.0
FT	91.9 (1.8)	47.5 (1.9)	77.5 (2.6)	85.0 (2.5)
MeZO	90.5 (1.2)	45.5 (2.0)	68.5 (3.9)	76.9 (2.7)
LP-MeZO	91.4 (1.4)	41.9 (3.3)	70.7 (3.4)	54.0 (4.5)

Table 17: 如前所述，在使用 MeZO 进行微调之前执行线性探测 [36] 有时可以在不增加内存开销的情况下提高性能。我们使用 $k = 16$ 进行这些实验。

E.2 OPT 实验

Table 18 提供了 OPT-30B 和 OPT-66B 的完整结果，以及详细的 MeZO 编号。

Task	SST-2	RTE	BoolQ	WSC	WIC	SQuAD
30B zero-shot	56.7	52.0	39.1	38.5	50.2	46.5
30B ICL	81.9	66.8	66.2	56.7	51.3	78.0
30B MeZO	90.6	66.4	67.2	63.5	56.3	85.2
30B MeZO (prefix)	87.5	72.6	73.5	55.8	59.1	83.9
66B zero-shot	57.5	**67.2**	66.8	43.3	50.6	48.1
66B ICL	89.3	65.3	62.8	52.9	54.9	81.3
66B MeZO	91.2	65.7	72.7	63.5	58.9	*
66B MeZO (prefix)	93.6	66.4	73.7	57.7	58.6	85.0

Table 18: OPT-30B 和 OPT-66B 上的实验（有 1,000 个例子）。*：MeZO 需要进一步调整才能成功优化。

E.3 MeZO 与全参数和 PEFT 的收敛

我们证明了收敛速度图中前 5,000 步的 SST-2 和 SNLI 上的 MeZO 、MeZO (LoRA) 和 MeZO（前缀） 5 。我们看到尽管他们优化的参数数量不同，MeZO 在全参数和 PEFT 上展示了相似的训练速度。这与我们在 Section 4 中的理论一致，表明 MeZO 的优化速度与参数数量无关。

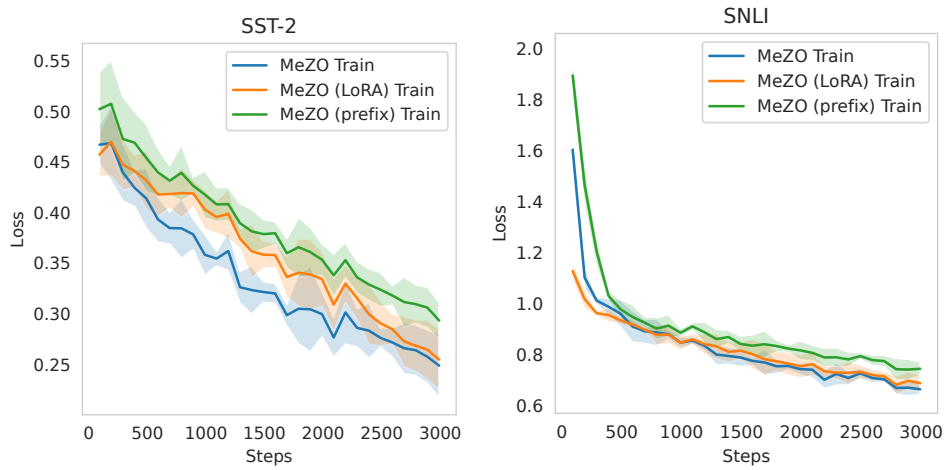

Figure 5: 当调整较少的参数时，MeZO 不会优化得更快，这与我们在 Section 4 中的理论一致。

E.4 ZO vs BBTv2

我们在表 19 中的相互评估任务上将 ZO 与 BBTv2 [64] 进行了比较。ZO 明显优于 BBTv2。此外，BBTv2 仅限于在低维空间中进行优化，需要进行前缀调整和向下投影以减少优化参数的数量。BBTv2 还采用了一种迭代方案，一次只优化一层。相比之下，ZO 适用于全参数调整和 PEFT，如我们的实验 (Section 3) 和理论 (Section 4) 所示。

Task Task type	SST-2 —— sentiment ——	SNLI – natural language inference –	RTE
Zero-shot	79.0	50.2	51.4
BBTv2	90.3 (1.7)	57.3 (2.3)	56.7 (3.3)
MeZO	90.5 (1.2)	68.5 (3.9)	64.0 (3.3)
MeZO (LoRA)	91.4 (0.9)	69.7 (6.0)	64.9 (3.6)
MeZO (prefix)	90.8 (1.7)	71.6 (2.5)	65.4 (3.9)

Table 19: ZO vs BBTv2 与 RoBERTa-large。BBTv2 性能来自 Sun et al. [64] 。

E.5 内存分析

我们显示了内存分析结果的详细数字 Table 20 ，也对应于 Figure 3 。有关我们如何分析内存使用情况，请参阅 Appendix D.7 。

Method	**Zero-shot / MeZO**	**ICL**	**Prefix FT**	**Full-parameter FT**
1.3B	1xA100 (4GB)	1xA100 (6GB)	1xA100 (19GB)	1xA100 (27GB)
2.7B	1xA100 (7GB)	1xA100 (8GB)	1xA100 (29GB)	1xA100 (55GB)
6.7B	1xA100 (14GB)	1xA100 (16GB)	1xA100 (46GB)	2xA100 (156GB)
13B	1xA100 (26GB)	1xA100 (29GB)	2xA100 (158GB)	4xA100 (316GB)
30B	1xA100 (58GB)	1xA100 (62GB)	4xA100 (315GB)	8xA100 (633GB)
66B	2xA100 (128GB)	2xA100 (134GB)	8xA100	16xA100

Table 20: MultiRC (avg # tokens=400) 数据集的内存使用情况。

F 校样

Proof of Lemma 2. 我们首先注意到在 $\epsilon \to 0$ 极限中，我们有

$$\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B}) = \frac{1}{Bn}\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{B}}\sum_{i\in[n]} \boldsymbol{z}_i\boldsymbol{z}_i^T\nabla\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x},\boldsymbol{y})\}).$$

对批量 $\mathcal{B}$ 和 $\boldsymbol{z}_i$ 进行期望，我们得到 $\mathbb{E}[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})] = \nabla\mathcal{L}(\boldsymbol{\theta})$ ，因此 $\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})$ 是梯度的无偏估计量。计算第二个时刻，我们得到

$$\begin{aligned}&\mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T\right]\\ &\quad= \frac{1}{B^2n^2}\sum_{(\boldsymbol{x}_1,\boldsymbol{y}_1),(\boldsymbol{x}_2,\boldsymbol{y}_2)\in\mathcal{B}}\sum_{i,j\in[n]}\mathbb{E}\left[(\boldsymbol{z}_i\boldsymbol{z}_i^T\nabla\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_1,\boldsymbol{y}_1)\}))(\boldsymbol{z}_j\boldsymbol{z}_j^T\nabla\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_2,\boldsymbol{y}_2)\}))^T\right]\end{aligned}$$

令 $\boldsymbol{u},\boldsymbol{v}$ 为两个任意向量。我们有那个

$$\mathbb{E}_{\boldsymbol{z}_i,\boldsymbol{z}_j}[\boldsymbol{z}_i\boldsymbol{z}_i^T\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{z}_j\boldsymbol{z}_j^T] = \boldsymbol{u}\boldsymbol{v}^T$$

当 $i\neq j$ 和

$$\begin{aligned}\mathbb{E}_{\boldsymbol{z}_i}[\boldsymbol{z}_i\boldsymbol{z}_i^T\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{z}_i\boldsymbol{z}_i^T] &= \mathbb{E}_{\boldsymbol{z}}[\boldsymbol{z}^{\otimes 4}](\boldsymbol{u},\boldsymbol{v})\\ &= \frac{3d}{d+2}\mathrm{Sym}(\boldsymbol{I}^{\otimes 2})(\boldsymbol{u},\boldsymbol{v})\\ &= \frac{d}{d+2}\cdot\boldsymbol{u}^T\boldsymbol{v}\cdot\boldsymbol{I} + \frac{2d}{d+2}\cdot\boldsymbol{u}\boldsymbol{v}^T.\end{aligned}$$

因此

$$\begin{aligned}&\mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T\right]\\ &= \frac{1}{B^2}\sum_{(\boldsymbol{x}_1,\boldsymbol{y}_1),(\boldsymbol{x}_2,\boldsymbol{y}_2)\in\mathcal{B}}\left(\frac{n-1}{n}+\frac{2d}{n(d+2)}\right)\mathbb{E}\left[\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_1,\boldsymbol{y}_1)\})\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_2,\boldsymbol{y}_2)\})^T\right]\\ &\quad+\frac{d}{n(d+2)}\cdot\mathbb{E}\left[\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_1,\boldsymbol{y}_1)\})^T\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_2,\boldsymbol{y}_2)\})\right]\boldsymbol{I}.\end{aligned}$$

接下来，请注意当 $(\boldsymbol{x}_1,\boldsymbol{y}_1)\neq(\boldsymbol{x}_2,\boldsymbol{y}_2)$ 时，我们有

$$\mathbb{E}\left[\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_1,\boldsymbol{y}_1)\})\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_2,\boldsymbol{y}_2)\})^T\right] = \nabla\mathcal{L}(\boldsymbol{\theta})\nabla\mathcal{L}(\boldsymbol{\theta})^T,$$

和 $(\boldsymbol{x}_1,\boldsymbol{y}_1)=(\boldsymbol{x}_2,\boldsymbol{y}_2)$ 我们有

$$\mathbb{E}\left[\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_1,\boldsymbol{y}_1)\})\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_2,\boldsymbol{y}_2)\})^T\right] = \nabla\mathcal{L}(\boldsymbol{\theta})\nabla\mathcal{L}(\boldsymbol{\theta})^T + \boldsymbol{\Sigma}_{MB}(\boldsymbol{\theta}).$$

因此

$$\frac{1}{B^2}\sum_{(\boldsymbol{x}_1,\boldsymbol{y}_1),(\boldsymbol{x}_2,\boldsymbol{y}_2)\in\mathcal{B}}\mathbb{E}\left[\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_1,\boldsymbol{y}_1)\})\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_2,\boldsymbol{y}_2)\})^T\right] = \nabla\mathcal{L}(\boldsymbol{\theta})\nabla\mathcal{L}(\boldsymbol{\theta})^T + \frac{1}{B}\boldsymbol{\Sigma}(\boldsymbol{\theta}),$$

并插入这个产量

$$\begin{aligned}\mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T\right] &= \left(1+\frac{d-2}{n(d+2)}\right)\cdot\left(\nabla\mathcal{L}(\boldsymbol{\theta})\nabla\mathcal{L}(\boldsymbol{\theta})^T+\frac{1}{B}\boldsymbol{\Sigma}(\boldsymbol{\theta})\right)\\ &\quad+\frac{d}{n(d+2)}\boldsymbol{I}\cdot\left(\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2+\frac{1}{B}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}))\right).\end{aligned}\tag{7}$$

最后，我们有

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\right\|^2\right] &= \left(1+\frac{d^2+d-2}{n(d+2)}\right)\cdot\left(\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2+\frac{1}{B}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}))\right)\\ &= \frac{d+n-1}{n}\cdot\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\|^2\right].\end{aligned}$$

□

Proof of Theorem 1. 根据带余数的泰勒定理，我们有

$$\mathcal{L}(\boldsymbol{\theta}_{t+1}) = \mathcal{L}(\boldsymbol{\theta}_t) + \nabla\mathcal{L}(\boldsymbol{\theta}_t)^T(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t) + \int_0^1 \lambda(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)^T\nabla^2\mathcal{L}(\lambda\boldsymbol{\theta}_{t+1} + (1-\lambda)\boldsymbol{\theta}_t)(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)^T d\lambda$$

接下来，请注意

$$\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\| = \eta\left\|\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\right\| \le \eta\sqrt{d}\cdot\frac{1}{Bn}\sum\left|\boldsymbol{z}_i^T\nabla\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x},\boldsymbol{y})\})\right| \le \eta d G(\boldsymbol{\theta}_t).$$

因此 $\|\lambda\boldsymbol{\theta}_{t+1} + (1-\lambda)\boldsymbol{\theta}_t - \boldsymbol{\theta}_t\| \le \eta d G(\boldsymbol{\theta}_t)$ 。假设我们有上限 $\nabla^2\mathcal{L}(\lambda\boldsymbol{\theta}_{t+1}+(1-\lambda)\boldsymbol{\theta}_t) \preceq \boldsymbol{H}(\boldsymbol{\theta}_t)$ ，因此

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}_{t+1}) &\le \mathcal{L}(\boldsymbol{\theta}_t) + \nabla\mathcal{L}(\boldsymbol{\theta}_t)^T(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t) + (\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)^T\boldsymbol{H}(\boldsymbol{\theta}_t)(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t)\\ &= \mathcal{L}(\boldsymbol{\theta}_t) - \eta\nabla\mathcal{L}(\boldsymbol{\theta}_t)^T\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B}) + \frac{1}{2}\eta^2\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B})^T\boldsymbol{H}(\boldsymbol{\theta}_t)\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B}).\end{aligned}$$

取关于 $\boldsymbol{\theta}_t$ 的条件期望并代入 (9) ，我们的 ZO 估计 $\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B})$ 的协方差公式，产生

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] &\le \mathcal{L}(\boldsymbol{\theta}_t) - \eta\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{\eta^2}{2}\left\langle\boldsymbol{H}(\boldsymbol{\theta}_t), \mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T\right]\right\rangle\\ &= \mathcal{L}(\boldsymbol{\theta}_t) - \eta\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{\eta^2}{2}\cdot\frac{d}{n(d+2)}\left(\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{1}{B}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t))\right)\operatorname{tr}(\boldsymbol{H}(\boldsymbol{\theta}_t))\\ &\quad + \frac{\eta^2}{2}\left(1 + \frac{d-2}{n(d+2)}\right)\left(\nabla\mathcal{L}(\boldsymbol{\theta}_t)^T\boldsymbol{H}(\boldsymbol{\theta}_t)\nabla\mathcal{L}(\boldsymbol{\theta}_t) + \frac{1}{B}\langle\boldsymbol{\Sigma}(\boldsymbol{\theta}_t), \boldsymbol{H}(\boldsymbol{\theta}_t)\rangle\right)\end{aligned}$$

根据假设，Hessian 上界 $\boldsymbol{H}(\boldsymbol{\theta}_t)$ 满足 $\|\boldsymbol{H}(\boldsymbol{\theta}_t)\|_{op} \le \ell$ 和 $\operatorname{tr}(\boldsymbol{H}(\boldsymbol{\theta}_t)) \le \ell r$ 。因此

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] &\le \mathcal{L}(\boldsymbol{\theta}_t) - \eta\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{\eta^2\ell}{2}\cdot\left(\frac{dr+d-2}{n(d+2)} + 1\right)\cdot\left(\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{1}{B}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t))\right)\\ &= \mathcal{L}(\boldsymbol{\theta}_t) - \eta\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{\eta^2\ell}{2}\cdot\left(\frac{dr+d-2}{n(d+2)} + 1\right)\cdot\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B})\|^2\right],\end{aligned}$$

根据需要。 □

F.1 全球融合的证明

Lemma 4. *设 $\mathcal{L}(\boldsymbol{\theta})$ 为 μ -PL 并设 α 存在，使得所有 $\boldsymbol{\theta}$ 都为 $\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta})) \le \alpha(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*)$ 。然后*

$$t = O\left(\left(\frac{\ell}{\mu} + \frac{\ell\alpha}{\mu^2 B}\right)\log\frac{\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*}{\epsilon}\right)$$

SGD 的迭代我们有 $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] \le \mathcal{L}^* + \epsilon$.

Proof of Lemma 4. SGD 收益率的下降引理

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] - \mathcal{L}(\boldsymbol{\theta}_t) \le -\eta\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{1}{2}\eta^2\ell\cdot\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B})\|^2].$$

插入 $\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B})\|^2] = \|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{1}{B}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t))$ 并选择学习率 $\eta \le \frac{1}{\ell}$ 产生

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] \le \mathcal{L}(\boldsymbol{\theta}_t) - \frac{\eta}{2}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{\eta^2\ell}{2B}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t))$$

由于 $\mathcal{L}$ 是 μ -PL，我们得到

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] \le \mathcal{L}(\boldsymbol{\theta}_t) - \eta\mu(\mathcal{L}(\boldsymbol{\theta}_t) - \mathcal{L}^*) + \frac{\eta^2\ell}{2B}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t)).$$

自 $\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t)) \leq \alpha(\mathcal{L}(\boldsymbol{\theta}_t) - \mathcal{L}^*)$ 以来，我们有

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] \leq \mathcal{L}(\boldsymbol{\theta}_t) - \eta\mu(\mathcal{L}(\boldsymbol{\theta}_t) - \mathcal{L}^*) + \frac{\eta^2\ell\alpha}{2B}(\mathcal{L}(\boldsymbol{\theta}_t) - \mathcal{L}^*).$$

总而言之，

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1})] - \mathcal{L}^* \leq \left(1 - \eta\mu + \frac{\eta^2\ell\alpha}{2B}\right)(\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] - \mathcal{L}^*)$$

选择 $\eta = \min(\frac{1}{\ell}, \frac{\mu B}{\ell\alpha})$ ，我们得到

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1})] - \mathcal{L}^* \leq \left(1 - \min(\frac{\mu}{2\ell}, \frac{\mu^2 B}{2\ell\alpha})\right)(\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] - \mathcal{L}^*).$$

因此我们在之后用 $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] - \mathcal{L}^* \leq \epsilon$ 达成了解决方案

$$t = \max\left(\frac{2\ell}{\mu}, \frac{2\ell\alpha}{\mu^2 B}\right)\log\left(\frac{\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*}{\epsilon}\right) = O\left(\left(\frac{\ell}{\mu} + \frac{\ell\alpha}{\mu^2 B}\right)\log\frac{\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*}{\epsilon}\right)$$

次迭代。 □

Proof of Lemma 3. 通过 Corollary 1 ，具有 $\eta_{\text{ZO}} = \gamma^{-1}\eta_{\text{SGD}}$ 的 ZO-SGD 收益率

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] - \mathcal{L}(\boldsymbol{\theta}_t) \leq \frac{1}{\gamma} \cdot \left[-\eta_{\text{SGD}} \left\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\right\|^2 + \frac{1}{2}\eta_{\text{SGD}}^2\ell \cdot \mathbb{E}[\left\|\nabla\mathcal{L}(\boldsymbol{\theta}; \mathcal{B})\right\|^2]\right].$$

与 SGD 的证明一样，选择 $\eta_{\text{SGD}} \leq \frac{1}{\ell}$ 产生

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] - \mathcal{L}(\boldsymbol{\theta}_t) \leq \gamma^{-1} \cdot \left[-\frac{\eta_{\text{SGD}}}{2} \left\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\right\|^2 + \frac{\eta_{\text{SGD}}^2\ell}{2B}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t))\right].$$

因此，在 μ -PL 和 $\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t)) \leq \alpha(\mathcal{L}(\boldsymbol{\theta}_t) - \mathcal{L}^*)$ 假设下，我们获得

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1})] - \mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] \leq \gamma^{-1} \cdot \left[-\eta_{\text{SGD}}\mu + \frac{\eta_{\text{SGD}}^2\ell\alpha}{2B}\right] \cdot (\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] - \mathcal{L}^*)$$

$$\implies \mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1})] - \mathcal{L}^* \leq \left(1 - \gamma^{-1}\left(\eta_{\text{SGD}}\mu - \frac{\eta_{\text{SGD}}^2\ell\alpha}{2B}\right)\right)(\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] - \mathcal{L}^*).$$

选择 $\eta_{\text{SGD}} = \min(\frac{1}{\ell}, \frac{\mu B}{\ell\alpha})$ 产量

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1})] - \mathcal{L}^* \leq \left(1 - \gamma^{-1} \cdot \min(\frac{\mu}{2\ell}, \frac{\mu^2 B}{2\ell\alpha})\right)(\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] - \mathcal{L}^*).$$

因此我们在之后用 $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] - \mathcal{L}^* \leq \epsilon$ 得出了一个解决方案

$$t = \gamma \cdot \max\left(\frac{2\ell}{\mu}, \frac{2\ell\alpha}{\mu^2 B}\right)\log\left(\frac{\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*}{\epsilon}\right) = \mathcal{O}\left(\left(\frac{r}{n} + 1\right) \cdot \left(\frac{\ell}{\mu} + \frac{\ell\alpha}{\mu^2 B}\right)\log\frac{\mathcal{L}(\boldsymbol{\theta}_0) - \mathcal{L}^*}{\epsilon}\right)$$

次迭代。 □

F.1.1 假设验证

我们表明 $\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t)) \leq \alpha(\mathcal{L}(\boldsymbol{\theta}_t) - \mathcal{L}^*)$ 假设适用于某些损失。

首先，考虑用平方损失优化模型 $f(\boldsymbol{x}; \boldsymbol{\theta})$ ，使得

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i \in [N]}(f(\boldsymbol{x}_i; \boldsymbol{\theta}) - \boldsymbol{y}_i)^2.$$

然后有那个

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \frac{2}{N}\sum_{i \in [N]}(f(\boldsymbol{x}_i; \boldsymbol{\theta}) - \boldsymbol{y}_i)^2 \nabla f(\boldsymbol{x}_i; \boldsymbol{\theta}) \nabla f(\boldsymbol{x}_i; \boldsymbol{\theta})^T - \nabla\mathcal{L}(\boldsymbol{\theta}) \nabla\mathcal{L}(\boldsymbol{\theta})^T.$$

因此

$$\begin{aligned}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta})) &\leq \frac{2}{N}\sum_{i\in[N]}\left(f(\boldsymbol{x}_i;\boldsymbol{\theta})-\boldsymbol{y}_i\right)^2\left\|\nabla f(\boldsymbol{x}_i;\boldsymbol{\theta})\right\|^2 \\ &\leq 2\mathcal{L}(\boldsymbol{\theta})\sum_{i\in[N]}\left\|\nabla f(\boldsymbol{x}_i;\boldsymbol{\theta})\right\|^2.\end{aligned}$$

假设数据可以被插值，即 $\mathcal{L}^*=0$ 。如果函数是 L -Lipschitz，即 $\|\nabla f(\boldsymbol{x};\boldsymbol{\theta})\|\leq L$ ，则条件适用于 $\alpha=2NL^2$ 。如果我们处于内核状态，即某些特征图 ϕ 的 $f(\boldsymbol{x}_i;\boldsymbol{\theta})=\phi(\boldsymbol{x}_i)^T\boldsymbol{\theta}$ ，那么

$$\nabla^2\mathcal{L}(\boldsymbol{\theta})=\frac{2}{N}\sum_{i\in[N]}f(\boldsymbol{x}_i;\boldsymbol{\theta})\nabla f(\boldsymbol{x}_i;\boldsymbol{\theta})^T.$$

因此

$$\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}))\leq N\operatorname{tr}(\nabla^2\mathcal{L}(\boldsymbol{\theta}))\cdot\mathcal{L}(\boldsymbol{\theta})\leq N\ell r\cdot\mathcal{L}(\boldsymbol{\theta}).$$

所以条件适用于 $\alpha=N\ell r$ 。

接下来，考虑交叉熵损失函数，即

$$\mathcal{L}(\boldsymbol{\theta})=\frac{1}{N}\sum_{i\in[N]}\exp(-y_i f(\boldsymbol{x}_i;\boldsymbol{\theta})).$$

然后有那个

$$\boldsymbol{\Sigma}(\boldsymbol{\theta})=\frac{1}{N}\sum_{i\in[N]}\exp(-2y_i f(\boldsymbol{x}_i;\boldsymbol{\theta}))y_i^2\nabla f(\boldsymbol{x}_i;\boldsymbol{\theta})\nabla f(\boldsymbol{x}_i;\boldsymbol{\theta})^T-\mathcal{L}(\boldsymbol{\theta})\mathcal{L}(\boldsymbol{\theta})^T,$$

假设目标 y_i 在 $[-1,1]$ 中有界（这对于二进制分类任务是正确的），并且 $\mathcal{L}^*=0$ （如果 $|f(\boldsymbol{x};\boldsymbol{\theta})|$ 可以发送到 ∞ 则可以实现）我们有

$$\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}))\leq\frac{1}{N}\sum_{i\in[N]}\exp(-2y_i f(\boldsymbol{x}_i;\boldsymbol{\theta}))\left\|\nabla f(\boldsymbol{x}_i;\boldsymbol{\theta})\right\|^2.$$

在内核机制中，$f(\boldsymbol{x}_i;\boldsymbol{\theta})=\phi(\boldsymbol{x}_i)^T\boldsymbol{\theta}$ ，因此

$$\nabla^2\mathcal{L}(\boldsymbol{\theta})=\frac{1}{N}\sum_{i\in[N]}\exp(-y_i f(\boldsymbol{x}_i;\boldsymbol{\theta}))\nabla f(\boldsymbol{x}_i;\boldsymbol{\theta})\nabla f(\boldsymbol{x}_i;\boldsymbol{\theta})^T.$$

因此

$$\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}))\leq N\operatorname{tr}(\nabla^2\mathcal{L}(\boldsymbol{\theta}))\cdot\mathcal{L}(\boldsymbol{\theta})\leq N\ell r\cdot\mathcal{L}(\boldsymbol{\theta}).$$

因此该条件也适用于 $\alpha=N\ell r$ 。

F.2 高斯扰动的证明

当绘制 $\boldsymbol{z}\,\mathcal{N}(0,\boldsymbol{I})$ 时，第一个引理计算协方差估计值 $\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})$ 的二阶矩。

Lemma 5. *让 $\boldsymbol{z}_i\sim\mathcal{N}(0,\boldsymbol{I})$ i.i.d. 然后*

$$\begin{aligned}\mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T\right]=&\left(1+\frac{1}{n}\right)\cdot\left(\nabla\mathcal{L}(\boldsymbol{\theta})\nabla\mathcal{L}(\boldsymbol{\theta})^T+\frac{1}{B}\boldsymbol{\Sigma}_{MB}(\boldsymbol{\theta})\right)\\&+\frac{1}{n}\boldsymbol{I}\cdot\left(\left\|\nabla\mathcal{L}(\boldsymbol{\theta})\right\|^2+\frac{1}{B}\operatorname{tr}(\boldsymbol{\Sigma}_{MB}(\boldsymbol{\theta}))\right).\end{aligned}\tag{8}$$

Proof. 正如在 Lemma 2 的证明中一样，我们在 $\epsilon\to0$ 极限中有它

$$\begin{aligned}&\mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T\right]\\&\quad=\frac{1}{B^2n^2}\sum_{(\boldsymbol{x}_1,\boldsymbol{y}_1),(\boldsymbol{x}_2,\boldsymbol{y}_2)\in\mathcal{B}}\sum_{i,j\in[n]}\mathbb{E}\left[(\boldsymbol{z}_i\boldsymbol{z}_i^T\nabla\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_1,\boldsymbol{y}_1)\}))(\boldsymbol{z}_j\boldsymbol{z}_j^T\nabla\mathcal{L}(\boldsymbol{\theta};\{(\boldsymbol{x}_2,\boldsymbol{y}_2)\}))^T\right]\end{aligned}$$

对于向量 $\boldsymbol{u}, \boldsymbol{v}$ ，我们有

$$\mathbb{E}_{\boldsymbol{z}_i, \boldsymbol{z}_j}[\boldsymbol{z}_i \boldsymbol{z}_i^T \boldsymbol{u}\boldsymbol{v}^T \boldsymbol{z}_j \boldsymbol{z}_j^T] = \boldsymbol{u}\boldsymbol{v}^T$$

当 $i \neq j$ 和

$$\mathbb{E}_{\boldsymbol{z}_i}[\boldsymbol{z}_i \boldsymbol{z}_i^T \boldsymbol{u}\boldsymbol{v}^T \boldsymbol{z}_i \boldsymbol{z}_i^T] = \mathbb{E}_{\boldsymbol{z}}[\boldsymbol{z}^{\otimes 4}](\boldsymbol{u}, \boldsymbol{v}) = 3\mathrm{Sym}(\boldsymbol{I}^{\otimes 2})(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T\boldsymbol{v} \cdot \boldsymbol{I} + 2\boldsymbol{u}\boldsymbol{v}^T.$$

因此

$$\begin{aligned}
&\mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B})^T\right] \\
&= \frac{1}{B^2} \sum_{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2) \in \mathcal{B}} \left(\frac{n-1}{n} + \frac{2}{n}\right) \mathbb{E}\left[\mathcal{L}(\boldsymbol{\theta}; \{(\boldsymbol{x}_1, \boldsymbol{y}_1)\})\mathcal{L}(\boldsymbol{\theta}; \{(\boldsymbol{x}_2, \boldsymbol{y}_2)\})^T\right] \\
&\quad + \frac{1}{n} \cdot \mathbb{E}\left[\mathcal{L}(\boldsymbol{\theta}; \{(\boldsymbol{x}_1, \boldsymbol{y}_1)\})^T \mathcal{L}(\boldsymbol{\theta}; \{(\boldsymbol{x}_2, \boldsymbol{y}_2)\})\right] \boldsymbol{I}.
\end{aligned}$$

在 Lemma 2 的证明中我们证明了

$$\frac{1}{B^2} \sum_{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2) \in \mathcal{B}} \mathbb{E}\left[\mathcal{L}(\boldsymbol{\theta}; \{(\boldsymbol{x}_1, \boldsymbol{y}_1)\})\mathcal{L}(\boldsymbol{\theta}; \{(\boldsymbol{x}_2, \boldsymbol{y}_2)\})^T\right] = \nabla\mathcal{L}(\boldsymbol{\theta})\nabla\mathcal{L}(\boldsymbol{\theta})^T + \frac{1}{B}\boldsymbol{\Sigma}(\boldsymbol{\theta}).$$

插入这个产量

$$\begin{aligned}
\mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B})^T\right] = &\left(\frac{n+1}{n}\right) \cdot \left(\nabla\mathcal{L}(\boldsymbol{\theta})\nabla\mathcal{L}(\boldsymbol{\theta})^T + \frac{1}{B}\boldsymbol{\Sigma}(\boldsymbol{\theta})\right) \\
&+ \frac{1}{n}\boldsymbol{I} \cdot \left(\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 + \frac{1}{B}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}))\right).
\end{aligned} \tag{9}$$

□

在 $\boldsymbol{z}_i$ 是高斯分布的情况下，我们可以证明 Theorem 1 的模拟。一个挑战是 $\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|$ 不再有界；相反，我们 r -local effective rank 假设仅以高概率成立，因此为了限制预期的损失减少，我们必须控制 $\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\|$ 较大的概率。

考虑以下局部 r 有效等级假设的修改版本，其中 Hessian 矩阵的上限是在半径是 Assumption 1 中球的两倍大的球上测量的。

Assumption 2 (Local r -effective rank, Gaussian)**.** 让 $G(\boldsymbol{\theta}_t) = \max_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}} \|\nabla\mathcal{L}(\boldsymbol{\theta}_t; \{(\boldsymbol{x}, \boldsymbol{y})\})\|$ 。存在矩阵 $\boldsymbol{H}(\boldsymbol{\theta}_t)$ 使得：

1. 对于所有 $\boldsymbol{\theta}$ 这样的 $\|\boldsymbol{\theta} - \boldsymbol{\theta}_t\| \leq 2\eta d G(\boldsymbol{\theta}_t)$ ，我们有 $\nabla^2\mathcal{L}(\boldsymbol{\theta}) \preceq \boldsymbol{H}(\boldsymbol{\theta}_t)$ 。
2. $\boldsymbol{H}(\boldsymbol{\theta}_t)$ 的有效等级，即 $\operatorname{tr}(\boldsymbol{H}(\boldsymbol{\theta}_t))/\|\boldsymbol{H}(\boldsymbol{\theta}_t)\|_{op}$ ，最多为 r 。

Theorem 2 (Dimension-Free Rate, Gaussian $\boldsymbol{z}$)**.** 假设损失表现出局部 r 有效等级 *(Assumption 2)*。如果 $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_{ZO}\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_t; \mathcal{B})$ 是 *ZO-SGD* 的单步，使用 n *-SPSA* 估计和大小为 B 的小批量，则存在 $\gamma = \Theta(r/n)$ 使得预期损失减少可以被限定为

$$\begin{aligned}
&\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] - \mathcal{L}(\boldsymbol{\theta}_t) \\
&\leq -\eta_{ZO}\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{1}{2}\eta_{ZO}^2 \ell \cdot \gamma \cdot \mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{\theta}_t; \mathcal{B})\|^2] + \eta_{ZO}^2 \ell G(\boldsymbol{\theta}_t)^2 \exp(-\Omega(nd)).
\end{aligned}$$

Proof of Theorem 2. 让 $\mathcal{A}$ 成为 $\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\| \leq 2\eta d G(\boldsymbol{\theta}_t)$ 的事件。在 $\mathcal{A}$ 上，我们有

$$\mathcal{L}(\boldsymbol{\theta}_{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}_t) - \eta\nabla\mathcal{L}(\boldsymbol{\theta}_t)^T\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B}) + \frac{1}{2}\eta^2\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_t; \mathcal{B})^T\boldsymbol{H}(\boldsymbol{\theta})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_t; \mathcal{B}).$$

同样，由于 $\mathcal{L}$ 是 ℓ -平滑的，我们有

$$\mathcal{L}(\boldsymbol{\theta}_{t+1}) \leq \mathcal{L}(\boldsymbol{\theta}_t) - \eta\nabla\mathcal{L}(\boldsymbol{\theta}_t)^T\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}; \mathcal{B}) + \frac{1}{2}\eta^2\ell\left\|\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_t; \mathcal{B})\right\|^2.$$

因此

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] &\leq \mathcal{L}(\boldsymbol{\theta}_{t+1}) - \eta \left\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\right\|^2 + \frac{1}{2}\eta^2 \left\langle \mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T \cdot \mathbf{1}(\mathcal{A})\right], \boldsymbol{H}(\boldsymbol{\theta}_t)\right\rangle \\ &\quad + \frac{1}{2}\eta^2 \ell \mathbb{E}\left[\left\|\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B})\right\|^2 \cdot \mathbf{1}(\neg\mathcal{A})\right] \\ &= \mathcal{L}(\boldsymbol{\theta}_{t+1}) - \eta \left\|\nabla \mathcal{L}(\boldsymbol{\theta}_t)\right\|^2 + \frac{1}{2}\eta^2 \left\langle \mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T\right], \boldsymbol{H}(\boldsymbol{\theta}_t)\right\rangle \\ &\quad \frac{1}{2}\eta^2 \left\langle \mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T \cdot \mathbf{1}(\neg\mathcal{A})\right], \ell I - \boldsymbol{H}(\boldsymbol{\theta}_t)\right\rangle.\end{aligned}$$

后一项可以有界如下

$$\begin{aligned}\frac{1}{2}\eta^2 \left\langle \mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T \cdot \mathbf{1}(\neg\mathcal{A})\right], \ell I - \boldsymbol{H}(\boldsymbol{\theta}_t)\right\rangle &\leq \eta^2 \ell \mathbb{E}\left[\left\|\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\right\|^2 \cdot \mathbf{1}(\neg\mathcal{A})\right] \\ &\leq \eta^2 \ell \mathbb{E}\left[\left\|\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\right\|^4\right]^{\frac{1}{2}} \Pr[\neg\mathcal{A}]^{1/2}.\end{aligned}$$

梯度估计 $\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})$ 满足

$$\left\|\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\right\| \leq \frac{1}{n}\sum_{i\in[n]} \left|\boldsymbol{z}_i^T \nabla \mathcal{L}(\boldsymbol{\theta};\mathcal{B})\right| \cdot \|\boldsymbol{z}_i\|$$

期望项的上限为

$$\begin{aligned}\mathbb{E}\left[\left\|\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\right\|^4\right] &\leq \frac{1}{n}\sum_{i\in[n]} \mathbb{E}\left[\left|\boldsymbol{z}^T \nabla \mathcal{L}(\boldsymbol{\theta};\mathcal{B})\right|^4 \cdot \|\boldsymbol{z}\|^4\right] \\ &\leq \mathbb{E}\left[\left|\boldsymbol{z}^T \nabla \mathcal{L}(\boldsymbol{\theta};\mathcal{B})\right|^8\right]^{1/2} \mathbb{E}\left[\|\boldsymbol{z}\|^8\right]^{1/2} \\ &\leq \sqrt{105}(d+6)^2 G(\boldsymbol{\theta}_t)^4,\end{aligned}$$

，我们在其中插入了高斯矩和 χ^2 随机变量的显式公式。接下来，请注意在事件 $\neg\mathcal{A}$ 中，我们有

$$2\eta d G(\boldsymbol{\theta}_t) \leq \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\| = \eta \left\|\hat{\nabla}\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B})\right\| \leq \eta \cdot \frac{1}{n}\sum_{i\in[n]} \|\boldsymbol{z}_i\|^2 G(\boldsymbol{\theta}_t).$$

因此

$$\Pr[\neg\mathcal{A}] \leq \Pr\left[\sum_{i\in[n]} \|\boldsymbol{z}_i\|^2 \geq 2nd\right]$$

Lemma 6 (Standard χ^2 -tail bound)**.** *设 Z 为具有 k 自由度的 χ^2 随机变量。然后*

$$\Pr[Z \geq k+u] \leq \exp\left(-\min\left(\frac{u^2}{16k}, \frac{u}{16}\right)\right)$$

由于 $\sum_{i\in[n]} \|\boldsymbol{z}_i\|^2$ 是具有 nd 自由度的 χ^2 随机变量，因此我们有

$$\Pr[\neg\mathcal{A}] \leq \exp\left(-\frac{nd}{16}\right).$$

总而言之，

$$\begin{aligned}\frac{1}{2}\eta^2 \left\langle \mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T \cdot \mathbf{1}(\neg\mathcal{A})\right], \ell I - \boldsymbol{H}(\boldsymbol{\theta}_t)\right\rangle &\leq \eta^2 \ell 105^{1/4}(d+6) G(\boldsymbol{\theta}_t)^2 \exp(-\frac{nd}{32}) \\ &= \eta^2 \ell G(\boldsymbol{\theta}_t)^2 \exp(-\Omega(nd)).\end{aligned}$$

最后，插入 (8) 以及 $\|\boldsymbol{H}(\boldsymbol{\theta}_t)\|_{op} \leq \ell$ 和 $\operatorname{tr}(\boldsymbol{H}(\boldsymbol{\theta}_t)) \leq \ell r$ ，

$$\begin{aligned}\left\langle \mathbb{E}\left[\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})\hat{\nabla}\mathcal{L}(\boldsymbol{\theta};\mathcal{B})^T\right], \boldsymbol{H}(\boldsymbol{\theta}_t)\right\rangle &= \frac{r+n+1}{n}\cdot\ell\left(\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{1}{B}\operatorname{tr}(\boldsymbol{\Sigma}(\boldsymbol{\theta}_t))\right) \\ &= \frac{r+n+1}{n}\cdot\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B})\|^2\right]\end{aligned}$$

因此让 $\gamma = \frac{r+n+1}{n}$ 产生

$$\begin{aligned}&\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_{t+1}) \mid \boldsymbol{\theta}_t] - \mathcal{L}(\boldsymbol{\theta}_t) \\ &\quad\leq -\eta\|\nabla\mathcal{L}(\boldsymbol{\theta}_t)\|^2 + \frac{1}{2}\eta^2\ell\cdot\gamma\cdot\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{\theta}_t;\mathcal{B})\|^2] + \eta^2\ell G(\boldsymbol{\theta}_t)^2\exp(-\Omega(nd)),\end{aligned}$$

根据需要。 □