arXiv:2301.03573v2 [cs.LG] 5 Dec 2023

平衡即本质：通过自适应梯度校正加速稀疏训练

Bowen Lei[1*]、Dongkuan Xu[2], Ruqi Zhang[3], ShurenHe[1], Bani Mallick[11]德克萨斯农工大学， [2]北卡罗来纳州立大学， [3]普渡大学*bowenlei@stat.tamu.edu

尽管深度神经网络性能出色，但它们需要大量内存和计算资源，因此难以应用于资源受限场景。稀疏训练是降低这些成本最常用的技术之一；然而，稀疏性约束增加了优化难度，导致训练时间延长且不稳定。在本研究中，我们旨在克服这一问题，实现时空协同效率。为加速并稳定稀疏训练的收敛，我们分析了梯度变化，并提出一种自适应梯度校正方法。具体而言，我们估算当前梯度与先前梯度之间的相关性，并据此平衡这两个梯度，以获得校正后的梯度。我们的方法可用于标准和对抗设置下最常用的稀疏训练流程。从理论上讲，我们证明了该方法能够加快稀疏训练的收敛速率。在多个数据集、模型架构和稀疏度上的大量实验表明，在训练轮数相同的情况下，我们的方法准确率最高可比领先的稀疏训练方法提升**5.0%**；在达到相同准确率时，所需训练轮数最多可减少**52.1%**。我们的代码见：https://github.com/StevenBoys/AGENT。

1. 引言

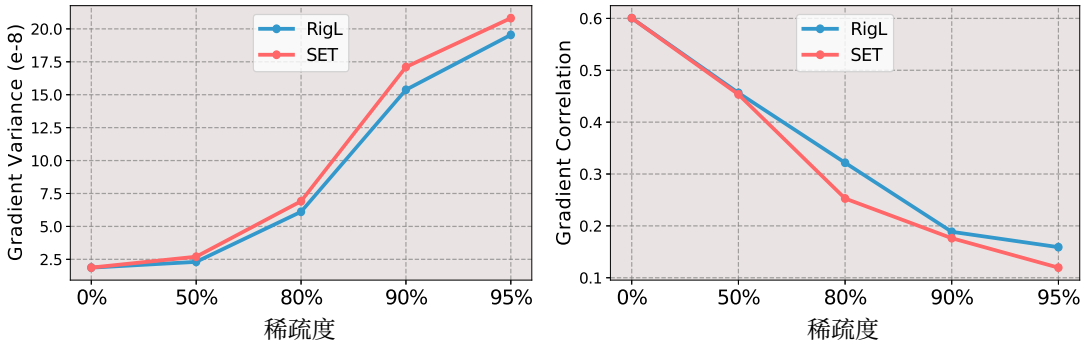
稀疏训练 [1–3] 是提升深度神经网络（DNN）的空间效率（例如内存存储）的最流行方法类别之一，并且正受到越来越多的关注，尤其是在资源受限情形[2–4]下。在稀疏训练过程中，会移除一定比例的连接以节省内存 [2, 4]。稀疏模式描述了保留或移除连接的位置，并会被迭代更新 [2, 5–7]。其目标是找到一种资源高效的稀疏神经网络（即移除部分连接），其性能可与原始稠密模型（即保留所有连接）相当，甚至更高 [8–10]。

然而，稀疏训练可能会给训练过程带来一些副作用，尤其是在高稀疏度情况下（例如，99%的权重为零）。首先，稀疏度会增加随机梯度的方差，导致模型沿次优方向移动，从而减缓收敛 [11, 12]。如图1（a）所示，我们通过实验证实，梯度方差会随着稀疏度的增加而增大（更多细节见第C.1节）。其次，它可能导致训练不稳定性（即测试准确率相对于迭代次数呈现噪声轨迹） [13, 14], ，这需要额外时间来弥补准确率下降，进而导致收敛变慢 [15]。此外，为了将稀疏训练应用于广泛的现实场景，还需要考虑模型的鲁棒性，因为这些场景通常会面临数据集偏移带来的挑战 [7, 11, 16, 17]。为了解决这些问题，我们提出以下问题：

问题 ***1***。 如何同时提升稀疏训练的收敛速度和训练稳定性？

以往的梯度校正方法 [18–20]用于加速并稳定稠密训练，但我们发现它们在稀疏训练中会失效。这些方法通常假设当前梯度与先前梯度高度相关，因此会加入大量先前梯度来校正梯度 [19, 21, 22]。然而，这一假设在稀疏训练中并不成立。图 1（b）显示

第一届简约与学习会议（CPAL 2024）。



(a) 梯度方差　　(b) 梯度相关性

图 1：RigL 和 SET 在不同稀疏度下（包括 0%（稠密）、50%、80%、90% 和 95%）得到的模型的梯度方差（a）和梯度相关性（b）。梯度方差随稀疏度增加而增大，梯度相关性则随稀疏度增加而降低。与稠密模型相比，稀疏模型的梯度方差更大，梯度相关性更低。

不同稀疏度下的梯度相关性，这表明梯度相关性会随着稀疏度增加而降低（更多细节见 C.1 节），从而打破当前梯度与先前梯度之间的平衡。因此，我们提出根据先前梯度与当前梯度之间的相关性，自适应地调整二者的权重，以加入适量的先前梯度。

问题 ***2.*** 如何设计一种加速且稳定的稀疏训练方法，使其在存在数据集偏移的现实场景中依然有效?

此外，稀疏训练在现实应用中的研究仍不充分。以往方法采用对抗训练来提升模型鲁棒性并应对数据偏移的挑战，但这通常会在梯度估计的方差之外引入额外偏差 [23], , 增加梯度校正的难度（详情见第 4.2 节）。因此，为了更准确地近似全梯度，尤其是在对抗设置下，我们设计了一种缩放策略来控制两个梯度的权重，决定将多少先前的梯度信息添加到当前梯度中，从而有助于实现平衡并进一步加速收敛。

在本文中，我们提出了一种**自适应梯度校正**（AGENT）方法，以加速并稳定标准和对抗设置下的稀疏训练。从理论上讲，我们证明了该方法能够提高稀疏训练的收敛速率。在实验方面，我们在多个基准数据集、模型架构和稀疏度上进行了大量实验。在标准和对抗设置下，与领先的稀疏训练方法相比，在训练轮数相同的情况下，我们的方法最高可将准确率提升**5.0%**；若要达到相同性能，则可将训练轮数减少**52.1%**。与以往主要关注结构化稀疏模式的稀疏训练加速研究不同，我们的方法兼容非结构化和结构化稀疏训练流程 [24, 25]。

2. 相关工作

稀疏训练： 近年来，人们对稀疏深度神经网络的关注日益增加。目标是在满足约束的同时，利用稀疏权重实现可比的性能。由此涌现出多种稀疏训练方法，它们在训练过程中保持权重稀疏。研究者提出了各种剪枝和生长准则，例如权重/梯度幅值、随机选择和权重符号 [1–7, 26–33]。然而，上述研究着眼于提升性能，却忽视了稀疏训练的副作用。稀疏度不仅会增大梯度方差，从而延缓收敛 [11, 12], , 还会导致训练不稳定性 [14]。如何兼顾空间效率与时间效率仍是一项挑战。此外，稀疏训练还可能加剧模型对对抗样本的脆弱性，而这正是深度神经网络的弱点之一 [7]。当模型遇到经过蓄意操纵的数据时，其性能可能迅速下降，进而引发日益严峻的安全问题 [34, 35]。本文聚焦于稀疏训练。一般而言，我们的方法可应用于任何基于随机梯度下降的稀疏训练流程。

加速训练： 近年来，针对深度神经网络如何提升时间效率开展了许多研究 [36, 37], , 其中一个热门方向是获得更准确的梯度估计来更新

模型 [20], , 例如采用方差缩减。在随机梯度下降中，人们使用较小的数据批次来逼近全梯度。批次估计量通常是无偏的，但方差可能很大，从而误导模型，因此催生了关于方差缩减的研究 [18–20, 38–43]。虽然对抗训练会使估计量产生偏差 [23], , 但在进行梯度校正时，我们仍需面对偏差-方差权衡。一种普遍思路是利用噪声较小的旧梯度来平衡梯度噪声 [19, 44, 45]。其他一些基于动量的方法也采用了类似策略，利用旧信息 [46, 47]。然而，上述工作都只考虑了非稀疏情形下的加速。

稀疏训练中的加速更具挑战性，先前关于这一问题的研究主要聚焦于结构化稀疏训练 [24, 25, 48]。首先，稀疏训练会导致更大的方差 [11]。此外，梯度校正方法所依赖的一些关键假设在稀疏约束下并不成立。在非稀疏情形下，通常假设旧梯度与新梯度高度相关，因此我们可以从旧梯度中获取大量信息 [19, 21, 22]。然而，稀疏性往往会降低相关性，而这些无关信息可能有害，使以往的方法不再适用于稀疏训练，并要求在新旧梯度之间进行更精细的权衡。此外，结构化稀疏模式不够灵活，可能导致模型准确率下降。相比之下，我们的方法从优化角度加速稀疏训练，并兼容非结构化和结构化稀疏训练流程。

3. 预备知识：随机方差缩减梯度

随机方差缩减梯度（SVRG） [21, 39, 49] 是一种广泛使用的梯度校正方法，旨在获得更准确的梯度估计，许多研究都采用了该方法 [18, 19, 50]。具体而言，每个训练轮次结束后，我们会评估当时基于 $\boldsymbol{\theta}$ 的全梯度 $\widetilde{\boldsymbol{g}}$，并将其存储起来以供后续使用~。在下一个训练轮次中，通过式（1）使用存储的旧梯度更新$\mathbf{B}_t$上的批次梯度估计。

$$\hat{\boldsymbol{g}}(\boldsymbol{\theta}_t) = \frac{1}{n} \sum_{i \in \mathbf{B}_t} \left(\boldsymbol{g}_i(\boldsymbol{\theta}_t) - \boldsymbol{g}_i(\widetilde{\boldsymbol{\theta}}) \right) + \widetilde{\boldsymbol{g}} \tag{1}$$

其中， $\boldsymbol{g}_i(\boldsymbol{\theta}_t) = \nabla l(\mathbf{x}_i|\boldsymbol{\theta}_t)$， $l(\boldsymbol{\theta}_t) = (\sum_{i=1}^{N} l(\mathbf{x}_i|\boldsymbol{\theta}_t))/N$ 是损失函数， $\widetilde{\boldsymbol{g}} = \nabla l(\widetilde{\boldsymbol{\theta}})$， $\boldsymbol{\theta}_t$ 是当前参数， n 是每个小批量数据中的样本数， N 是样本总数。SVRG在非稀疏情形下成功加速了许多训练任务，但与许多其他梯度校正方法类似，它在稀疏训练中效果不佳。

4. 方法

我们提出了一种**自**适应**梯**度校**正**方法（AGENT），并将其整合到近期的稀疏训练流程中，以实现加速并提高训练稳定性。为实现这一目标，AGENT 会过滤掉相关性较低的信息，并从旧梯度中获取经过良好控制且随时间变化的知识量。我们的方法克服了 SVRG 等先前加速方法的局限性 [21, 49, 51], , 成功加速并稳定了稀疏训练。AGENT 方法概述于算法1，并将在以下章节中详细阐述。

4.1. 对旧梯度的自适应控制

在 AGENT 中，我们设计了一种自适应方式，将旧梯度加到新梯度中，以过滤相关性较低的信息，并平衡新旧梯度。具体而言，我们按照公式（2）为旧梯度添加自适应权重 $c_t \in [0, 1]$，其中分别用 $\boldsymbol{g}_{\text{new}} = \frac{1}{n} \sum_{i \in \mathbf{B}_t} \boldsymbol{g}_i(\boldsymbol{\theta}_t)$ 和$\boldsymbol{g}_{\text{old}} = \frac{1}{n} \sum_{i \in \mathbf{B}_t} \boldsymbol{g}_i(\widetilde{\boldsymbol{\theta}})$ 表示随机子集 $\mathbf{B}_t$ 上当前参数 $\boldsymbol{\theta}_t$ 和先前参数 $\widetilde{\boldsymbol{\theta}}$ 对应的梯度¯。当新旧梯度高度相关时，我们需要较大的 c，以从旧梯度中获取更多有用信息。反之，当相关性较低时，我们需要较小的 c，以免无关信息干扰新梯度。

$$\hat{\boldsymbol{g}}(\boldsymbol{\theta}_t) = \frac{1}{n} \sum_{i \in \mathbf{B}_t} \left(\boldsymbol{g}_i(\boldsymbol{\theta}_t) - c_t \cdot \boldsymbol{g}_i(\widetilde{\boldsymbol{\theta}}) \right) + c_t \cdot \widetilde{\boldsymbol{g}} = \boldsymbol{g}_{\text{new}} - c_t \cdot \boldsymbol{g}_{\text{old}} + c_t \cdot \widetilde{\boldsymbol{g}}. \tag{2}$$

合适的 c_t 应能有效降低 $\hat{g}(\theta_t)$ 的方差。我们在公式（3）中对 $\hat{g}(\theta_t)$ 的方差进行分解（这里对符号作了一些宽泛的使用），其中更新后的梯度的方差是关于 c_t的二次函数。为简单起见，考虑 $\hat{g}(\theta_t)$ 为标量的情况，最优的 c_t^* 将具有公式（3）所示的形式。可以看出，当新梯度与旧梯度的相关性不高时， c_t^* 不会接近 1。由于在稀疏训练中， g_{new} 与 g_{old} 之间的低相关性更为常见，因此直接将先前方法中的 $c_t = 1$ 设为固定值并不合适，我们需要估计自适应权重c_t^*。为支持这一观点，我们在附录 B.6 中进行了讨论和实证分析，说明随着稀疏度增加，梯度变化加快，导致 g_{new} 与 g_{old}之间的相关性降低。

算法1 自适应梯度校正

输入： $\widetilde{\theta} = \theta_0$，轮次长度 m，步长 η_t，$c_0 = 0$，缩放参数 γ，平滑因子 α
对于 $t = 0$ **到** $T-1$，**执行**
  如果 t 模 $m = 0$，**则**
    $\widetilde{\theta} = \theta_t$
    $\widetilde{g} = (\sum_{i=1}^{N} \nabla l(\mathbf{x}_i|\theta))/N$
    如果 $t > 0$，**那么**
      计算 $\hat{c}_t^*$ 通过公式（4）
      $c_t = (1-\alpha)c_{t-1} + \alpha\,\hat{c}_t^*$
    结束如果
  else
    $c_t = c_{t-1}$
  结束如果
  采样一个大小为 n的小批量数据 $\mathbf{B}_t$
  $\theta_{t+1} = \theta_t - \eta_t \cdot \left(\frac{1}{n} \sum_{i \in \mathbf{B}_t} \left(g_i(\theta_t) - \gamma c_t \cdot g_i(\widetilde{\theta}) \right) + \gamma c_t \cdot \widetilde{g} \right)$
结束循环

$$\mathrm{Var}(\hat{g}(\theta_t)) = \mathrm{Var}(g_{\text{new}}) + c_t^2 \cdot \mathrm{Var}(g_{\text{old}}) - 2c_t \cdot \mathrm{Cov}(g_{\text{new}}, g_{\text{old}}), \quad c_t^* = \frac{\mathrm{Cov}(g_{\text{new}}, g_{\text{old}})}{\mathrm{Var}(g_{\text{old}})}. \tag{3}$$

我们发现，计算精确的 c_t^* 并不切实际，因此提出一种近似算法，以在新梯度和旧梯度之间取得平衡。计算精确的 c_t^*面临两项挑战。一方面，为了接近精确值，我们需要计算每批数据上的梯度，这在每次迭代中都代价过高。另一方面，梯度通常具有很高的维度，而且不同梯度对应的精确最优 c_t^* 也会有所不同。因此，受 Deng 等人 [52], 的启发，我们设计了一种近似算法，充分利用损失信息，同时仅少量增加计算开销。具体而言，我们根据损失变化估计 c_t^*，如公式（4）所示，并在每个训练轮次开始前使用动量自适应地更新 $\widehat{c}_t^*$。损失是一个标量，因此可以估计所有当前和先前梯度的共享相关性。此外，直观而言，损失与梯度相关，损失之间的相关性可以让我们深入了解梯度之间的相关性。

$$\widehat{c}_t^* = \frac{\mathrm{Cov}(l(\mathbf{B}|\theta_t), l(\mathbf{B}|\widetilde{\theta}))}{\mathrm{Var}(l(\mathbf{B}|\widetilde{\theta}))}, \tag{4}$$

其中$\mathbf{B}$ 表示用于估计梯度的样本子集。

4.2. 额外缩放参数的重要性

为确保在稀疏训练和对抗训练中成功实现加速，我们进一步提出一种缩放策略，将估计的 c_t^* 乘以一个较小的缩放参数 γ。使用缩放参数有两大主要优点。首先，缩放参数 γ 可以减小对抗训练 [23] 中梯度估计的偏差。在标准训练中，批次梯度估计量是全梯度的无偏估计量。然而，在对抗训练中，我们会对样本$\mathbf{B}_t$ 构成的迷你批次 进行扰动，得到$\mathbf{B}_{t\circ}$。旧梯度 g_{old} 是在批次数据$\mathbf{B}_t$上计算的，但存储的旧梯度 $\widetilde{g}$ 来自包含$\mathbf{B}_t$的原始数据，这使得 $\mathbb{E}[g_{\text{old}} - \widetilde{g}]$ 不等于零。因此，如公式（5）所示，全梯度的校正估计量将不再是无偏的。它可能具有较小的方差，但偏差较大，从而导致性能不佳。因此，我们提出一个介于0和1之间的缩放参数 γ，将偏差从 $c_t(g_{\text{old}} - \widetilde{g})$减小到 $\gamma c_t(g_{\text{old}} - \widetilde{g})$。

$$\mathbb{E}[\hat{g}(\theta_t)] = \mathbb{E}[g_{\text{new}} - c_t(g_{\text{old}} - \widetilde{g})] \neq \mathbb{E}[g_{\text{old}}] = \frac{1}{N}\sum_{i=1}^{N} g_i(\theta_t). \tag{5}$$

其次，缩放参数 γ 保证即使面对 c_t^* 的最坏情况估计，仍能降低方差，从而加快训练。其核心思想如图2所示，其中

x 轴和 y 轴分别对应权重 c_t 和梯度方差。蓝色曲线是一个二次函数，表示 c_t 与方差之间的关系。假设真正的最优值为 c^*，而我们对其进行近似。在最坏情况下，这个近似可能差到 $\hat{c}_1$，导致方差甚至高于 a_3（随机梯度下降中的方差），并减慢训练速度。接着，如果我们将 $\hat{c}_1$ 替换为 $\gamma\hat{c}_1$，就能降低方差并加快训练。

4.3. 与其他优化器的联系

基于动量的方法： 我们的 AGENT 采用了与基于动量的方法 [53, 54]，相似的思路，即利用旧梯度来改进当前批次梯度。然而，基于动量的方法没有考虑稀疏训练和对抗训练的特点，例如当前梯度与之前梯度之间相关性降低，以及梯度估计器可能存在偏差；同时，它也无法在新旧信息之间实现自适应平衡。当相关性较低时，它仍可能纳入过多旧信息，从而增大梯度方差或偏差。

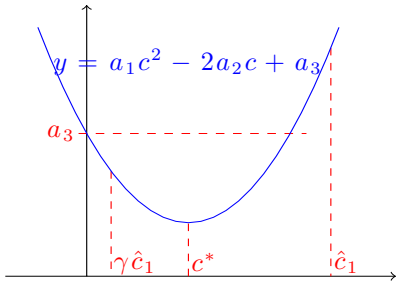


图2：展示缩放参数$\gamma = 0.1$ 如何在面对 c_t^*的最坏情况估计时确保加速。蓝色曲线是一个二次函数，表示 c_t与方差之间的关系。 c^* 是最优值。$\hat{c}_1$ 是一个较差的估计，会使方差大于 a_3 （随机梯度下降中的方差）。 $\gamma\hat{c}_1$ 可以降低方差。

自适应梯度方法： 我们的 AGENT 可以视为一种新型的自适应梯度方法，它会自适应地调整用于更新参数的梯度信息量，例如 Adam [55]。然而，先前的方法并非为稀疏训练而设计。尽管它们采用了自适应梯度，但其自适应机制有所不同，且没有考虑相关性的降低。

相反，我们的 **AGENT** 专为稀疏训练的特性量身打造：它近似相关性，并为旧梯度添加自适应权重，从而在新旧梯度之间取得平衡（更多比较见附录 B.9、B.10）。

5. 理论论证

从理论上讲，我们对 AGENT 进行了收敛性分析，并将其与 SVRG [56]进行了比较。我们用 $l(.)$ 表示损失函数，用 $\boldsymbol{g}$ 表示梯度。我们的证明基于假设1-2，详细推导见附录 A。

假设1。 （L-光滑）：可微损失函数 $l: \mathbb{R}^n \to \mathbb{R}$ 是 L-光滑的，即对于所有 $\boldsymbol{x}$、$\boldsymbol{y} \in \mathbb{R}^n$，损失 l 满足 $||\nabla l(\boldsymbol{x}) - \nabla l(\boldsymbol{y})|| \leq L||\boldsymbol{x} - \boldsymbol{y}||$。
假设 2. （σ-有界）：损失函数 l 具有 σ-有界梯度，即 $||\nabla l_i(\boldsymbol{x})|| \leq \sigma$ 对所有$i \in [N]$ 和 $\boldsymbol{x} \in \mathbb{R}^n$均成立。

Give在假设 1-2 下，我们遵循上述分析框架并建立定理1。
定理1. 在假设 *1-2* 下，适当选择步长 η_t 和 c_t后，使用 *AGENT* 训练 T个训练轮次后的梯度 $\mathbb{E}[||g(\boldsymbol{\theta_\pi})||^2]$可界定为：

$$\mathbb{E}[||g(\boldsymbol{\theta_\pi})||^2] \leq \frac{(l(\boldsymbol{\theta}_0) - l(\boldsymbol{\theta}_*))LN^\alpha}{Tn\nu} + \frac{2\kappa\mu^2\sigma^2}{N^\alpha m\nu}$$

其中，$\boldsymbol{\theta}_\pi$从 $\{\{\boldsymbol{\theta}_t^s\}_{t=0}^{m-1}\}_{s=0}^{T-1}$中均匀采样，$N$表示数据大小，$n$表示小批量大小，$m$表示轮次长度，$\boldsymbol{\theta}_0$为初始点，$\boldsymbol{\theta}_*$为最优解，$\nu$、$\mu$、$\kappa$、$\alpha > 0$为依赖于 η_t 和 c_t、N 及 n的常数。

关于定理1，我们作出以下说明，以阐明我们的 AGENT 所实现的加速：**注释1.** （更快的梯度变化速度）稀疏训练与稠密训练的一个重要区别在于梯度变化速度，这体现在假设*1*（L-光滑）中。通常，稀疏训练中的 L 会大于稠密训练中的 L。

注释2.（第一项分析）在定理*1*中，我们的 *AGENT* 界限中的第一项衡量的是最优参数偏差所带来的误差；当训练轮数 T 趋于无穷时，该项会趋于零。然而，在实际的稀疏训练应用中，T 是有限的，并且由于稀疏训练中的 L 增大，该项会扩大，这表明稀疏约束下的优化更具挑战性。

注释3.（第二项分析）在定理*1*中，第二项衡量了优化过程中含噪梯度和有限数据所带来的误差。在我们的深度神经网络训练中，由于 σ^2 相对较小，而 N 通常较大，因此假设 T 有限时，与第一项相比，第二项可以忽略不计或小得多。

根据上述分析，我们可以比较 AGENT 和 SVRG 的界，并发现，在稀疏训练的情况下，适当选择 c_t，借助校正良好的梯度，可以使 AGENT 的界比 SVRG 的界更紧。

注释4.（与 *SVRG* 的比较）在假设 *1-2* 下，使用 *SVRG* 训练 T 轮后，梯度 $\mathbb{E}[||g(\boldsymbol{\theta_\pi})||^2]$ 的上界为 [*56*]:

$$\mathbb{E}[||g(\boldsymbol{\theta_\pi})||^2] \leq \frac{(l(\boldsymbol{\theta}_0) - l(\boldsymbol{\theta}_*))LN^\alpha}{Tn\nu^*}.$$

该上界的形式与定理*1*中的第一项相似。由于定理*1*中的第二项可以忽略不计，我们只需比较第一项。适当选择 c_t 后，$\hat{}\boldsymbol{g}(\boldsymbol{\theta}_t)$ 的方差会降低，从而使 *AGENT* 的 ν 小于 *SVRG* 的 ν^*（详见附录 *A* 注 *6*）。因此，与 *SVRG* 相比，*AGENT* 的第一项更小，这表明 *AGENT* 的界比 *SVRG* 的界更紧。

6. 实验

我们将 AGENT 应用于四种近期的稀疏训练流程，即 SET [1], RigL [2], BSR-Net [7] 和 ITOP [6]（说明见附录 D）。数据集、模型架构以及其他训练和评估设置的详细信息如下所示。

数据集与模型架构：数据集方面，我们使用 CIFAR-10、CIFAR-100 [57],SVHN [58]，和 ImageNet-2012 [59]。模型架构方面，我们使用 VGG-16 [60], ResNet-18、ResNet-50 [61],和 Wide-ResNet-28-4 [62]。

训练设置：对于稀疏训练，我们选择两种稀疏度（90% 和 99%）。对于 BSR-Net，我们同时考虑标准和对抗训练设置。在 SET、RigL 和 ITOP 中，我们专注于标准训练。在标准训练中，我们仅使用原始数据，而不使用扰动样本。对于对抗部分，我们使用两种常见目标（对抗训练和 TRADES）[63, 64]，并遵循 BSR-Net [7]中的评估设置。

表 1：基于 BSR-Net 的模型的测试准确率（%）。稀疏 VGG-16 在标准设置下训练。在相同训练轮数下，我们的方法的准确率通常高于 BSR-Net。

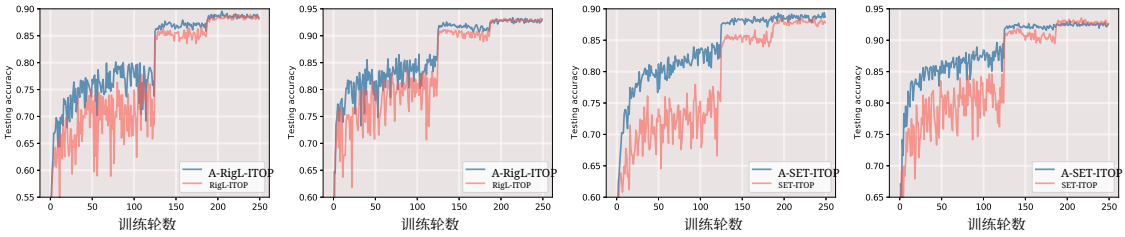
	训练轮次	90% 稀疏度		99% 稀疏度	
		BSR-Net	Ours	BSR-Net	Ours
AT	20	55.0 (1.59)	**63.6 (1.31)**	49.8 (1.46)	**56.4 (1.39)**
	40	62.2 (1.88)	**64.9（0.81）**	54.1（1.72）	**57.7（0.39）**
	70	73.1（0.39）	**75.1（0.27）**	64.7（0.30）	**66.0（0.23）**
	90	73.2（0.29）	**74.1（0.25）**	63.7（0.25）	**65.8（0.24）**
	140	76.7（0.27）	**77.4 (0.26)**	68.4 (0.20)	**69.8 (0.14)**
	200	76.6 (0.25)	**78.1 (0.24)**	69.0 (0.15)	**70.7 (0.06)**
TRADES	20	62.0 (0.82)	**65.0 (0.61)**	55.7 (0.76)	**57.6 (0.45)**
	40	65.4 (0.97)	**66.0 (0.34)**	**60.6 (0.69)**	58.4 (0.34)
	70	73.4 (0.52)	**73.5 (0.33)**	66.3 (0.35)	**67.3（0.30）**
	90	73.0（0.36）	**73.6（0.28）**	66.2（0.33）	**67.5（0.24）**
	140	76.4（0.25）	**76.8 (0.25)**	**70.0 (0.29)**	69.9 (0.21)
	200	75.6 (0.23)	**77.0 (0.24)**	70.8 (0.19)	**70.9 (0.25)**
Standard	20	70.4 (2.50)	**81.8 (0.62)**	60.6 (1.26)	**69.8 (1.45)**
	40	77.6 (1.39)	**82.4 (0.47)**	62.6 (2.47)	**73.7 (0.36)**
	70	86.8 (0.78)	**89.7 (0.38)**	79.7 (0.72)	**83.7 (0.24)**
	90	87.6 (0.63)	**89.3 (0.22)**	80.5 (0.55)	**83.9 (0.42)**
	140	91.7 (0.44)	**92.5 (0.06)**	85.7 (0.42)	**86.9 (0.07)**
	200	91.8 (0.23)	**92.6 (0.12)**	85.8 (0.12)	**87.1 (0.25)**

6.1. 收敛速度与稳定性比较

我们通过两个标准比较收敛速度，包括 (a) 在遍历数据（训练轮次）次数相同的情况下的**测试准确率**，以及 (b) 达到相同测试准确率所需的**遍历数据（训练轮次）次数**；后者是比较优化器速度的常用指标 [18, 22, 46, 49]。

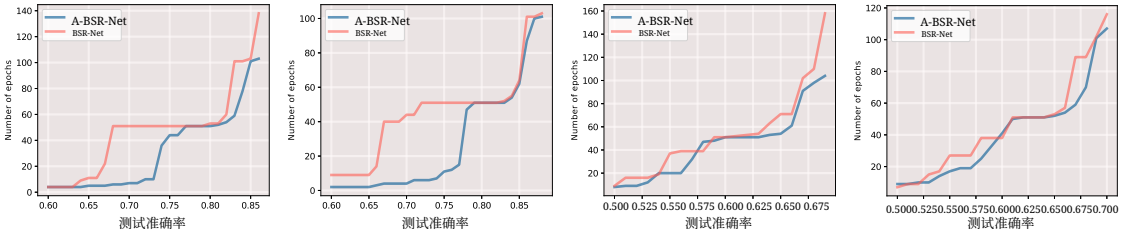
表2：采用 VGG-16 的基于 BSR-Net 的模型（BSR）在对抗样本上的测试准确率（%）。在相同训练轮数下，我们的方法在几乎所有情况下都比 BSR-Net 准确率更高。

训练轮次		90% 稀疏		99% 稀疏			90% 稀疏		99% 稀疏	
		BSR	Ours	BSR	Ours		BSR	Ours	BSR	Ours
70	AT	37.8	**45.2**	34.9	**39.4**	TRADES	34.8	**45.4**	33.5	**39.0**
90	AT	33.6	**44.8**	35.8	**39.8**	TRADES	36.8	**44.8**	31.7	**39.1**
140	AT	**46.5**	43.8	40.8	**41.2**	TRADES	45.1	**46.3**	38.2	**41.5**
200	AT	43.3	**44.6**	**42.2**	42.0	TRADES	**47.2**	46.2	39.3	**41.2**



(a) VGG-C(RigL) (b) ResNet-34(RigL) (c) VGG-C(SET) (d) ResNet-34(SET)

图3：CIFAR-10 上稀疏度为 99% 时基于 ITOP 的模型的测试准确率。A-RigL-ITOP 和 A-SET-ITOP（蓝色曲线）的收敛速度快于 RigL-ITOP 和 SET-ITOP（粉色曲线）。



(a) VGG-16，标准 (b) WRN-28-4，标准 (c) VGG-16，对抗训练 (d) WRN-28-4，对抗训练

图4：在 99% 稀疏度下达到目标准确率所需的训练轮数。与 BSR-Net（粉色曲线）相比，我们的 A-BSR-Net（蓝色曲线）达到目标准确率所需的时间更短。

遍历数据（训练轮次）次数相同时的测试准确率：对于基于 BSR-Net 的结果，表 1–2 列出了稀疏 VGG-16 在 CIFAR-10 干净样本和对抗样本上的准确率，其中较高的准确率以粗体标出。在标准设置中，我们仅展示干净准确率。在几乎所有训练轮次和设置下，我们的方法都能保持更高的干净准确率和鲁棒准确率，证明了我们的方法成功实现了加速。尤其是在训练时间有限的情况下（例如 20 个训练轮次），我们的 A-BSR-Net 通常表现出显著提升，干净准确率最高可达**11.4%**，表明前期搜索时间大幅缩短。此外，考虑 6 种时间预算下的平均准确率提升，我们的方法在准确率上最多比 BSR-Net 高**5.0%**。

对于基于 ITOP 的结果，如图3所示，蓝色曲线（A-RigL-ITOP 和 A-SET-ITOP）始终高于粉色曲线（RigL-ITOP 和 SET-ITOP），表明使用我们的 AGENT 能够实现更快的训练。此外，我们可以看到，粉色曲线出现了严重的上下波动，尤其是在训练的早期阶段。相比之下，蓝色曲线在所有设置下都更加稳定，这表明 AGENT 能够有效提升稀疏训练的稳定性。

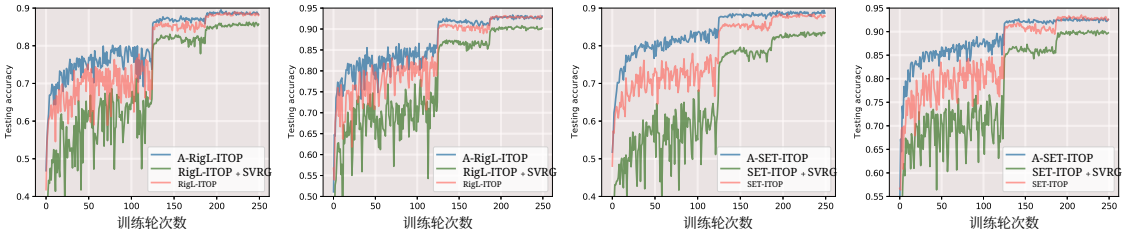
达到相同测试准确率所需的遍历数据次数（训练轮次）：图4展示了达到特定准确率所需的训练轮数。蓝色曲线（A-BSR-Net）始终低于粉色曲线（BSR-Net）；平均而言，我们的方法最多可将训练轮数减少 **52.1%**，表明采用我们提出的 A-BSR-Net 能够实现更快的训练。

6.2. 最终准确率比较

此外，我们还比较了充分训练后的最终准确率。对于表3中基于 ITOP 的结果，我们在 ImageNet-12 上使用 ResNet-50，将 A-RigL-ITOP 与 RigL-ITOP 进行比较；我们的方法始终保持最终准确率。对于表 4 中基于 BSR-Net 的结果，我们将我们的 A-BSR-Net

表3：基于 ITOP 的 ResNet-50 在 ImageNet-2012 上的最终准确率（%）。我们的方法保持了准确率。

稀疏度	RigL-ITOP	Ours
80%	75.5 (0.10)	**75.6**(0.12)
90%	73.6 (0.12)	**73.4**(0.11)



(a) VGG-C (b) ResNet-34 (c) VGG-C (d) ResNet-34

图5：基于 ITOP 的模型（99% 稀疏度，CIFAR-10）的测试准确率。与随机梯度下降（粉色曲线）相比，SVRG（绿色曲线）会减慢训练速度，而我们的方法（蓝色曲线）则会加速训练。在 SVHN 上使用 VGG-16 和 WRN-28-4 的 BSR-Net 模型中，我们的方法通常表现最佳。这表明，我们的 AGENT 能够加速稀疏训练，同时保持甚至提高准确率。

6.3. 与其他梯度校正方法的比较

我们还将 AGENT 与 SVRG [50]，（一种在非稀疏情形下广受欢迎的梯度校正方法）进行了比较。所展示的基于 ITOP 的结果是在 CIFAR-10 上使用稀疏率为 99% 的 VGG-C 和 ResNet-34 得出的。图 -

表4：采用对抗目标（TRADES）的 BSR-Net 模型在 SVHN 上、稀疏度为 90% 和 99% 时的最终准确率（%）。我们的 AGENT 能够保持甚至提高准确率。

	BSR-Net（90%）	我们的方法（90%）	BSR-Net（99%）	我们的方法（99%）
VGG-16	89.4 (0.29)	**94.4 (0.25)**	86.4 (0.25)	**90.9 (0.26)**
WRN-28-4	92.8 (0.24)	**95.5 (0.23)**	89.5 (0.22)	**92.2 (0.19)**

图5 (a)-(b) 展示了 A-RigL-ITOP（蓝色）、RigL-ITOP（粉色）和 RigL-ITOP+SVRG（绿色）的测试准确率。我们可以看到，RigL-ITOP+SVRG 的绿色曲线通常低于 A-RigL-ITOP 和 RigL-ITOP 的另外两条曲线，这表明 SVRG 会减缓模型收敛。至于我们提出的 A-RigL-ITOP 的蓝色曲线，它始终高于 RigL-ITOP 的粉色曲线，而且比 RigL-ITOP+SVRG 的绿色曲线更平滑，这表明其成功实现了加速和稳定。图5 (c)-(d) 所示的基于 SET-ITOP 的结果也呈现出类似的规律。绿色曲线（SET-ITOP+SVRG）通常低于蓝色曲线（A-SET-ITOP）和粉色曲线（SET-ITOP）。这表明 SVRG 不适用于稀疏训练，而我们提出的 AGENT 克服了其局限性，从而实现了稀疏训练的加速与稳定。

6.4. 与其他梯度校正方法的结合

除了与 SVRG 配合使用外，我们的 AGENT 还可以与其他梯度校正方法结合，以实现稀疏训练加速，例如基于动量的方差缩减方法（MVR） [46]。

表5：MVR 与 AGENT+MVR 的测试准确率对比。AGENT 加速了稀疏训练中的 MVR。

	第20位	第40位	第70位	第90位	第140位	第200位
MVR	62.6	66.8	69.8	71.2	73.5	74.4
AGENT+MVR	**71.6**	**75.7**	**77.9**	**79.1**	**82.3**	**82.3**

我们分别使用 MVR 和 MVR+AGENT，在基于 99% SET-ITOP 的稀疏 VGG-C 上训练 CIFAR-10。如表5所示，在不同训练轮次下，MVR+AGENT 通常比 MVR 获得更高的测试准确率，这表明了 AGENT 的加速效果及其通用性。

6.5. 消融研究

我们通过逐一移除方法 AGENT 的各个组成部分并比较结果，来展示每个组成部分的重要性。具体而言，我们考察旧梯度的时变权重 c_t 和缩放参数 γ的作用。“Fixed c_t”表示在训练过程中固定权重 $c_t = 0.1$，“No γ”表示直接使用公式（4）中的 $\hat{c}_t^*$ 和动量方案，且不添加缩放参数 γ。

表 6 展示了在不同训练周期预算下，使用 VGG-16 在 CIFAR-10 上进行标准训练和对抗训练（AT）时，稀疏度为 90% 和 99% 的模型在干净样本和对抗样本上的准确率。在对抗训练（AT 和 TRADES）中，我们可以看到，“No γ”学习效果不佳，结果最差。我们的方法在几乎所有情况下都优于“Fix c_t”和“No γ”，尤其是在高稀疏度任务中（即 99% 稀疏度）。对于标准训练，“No γ”能够学习到一些信息，但仍然

表 6：消融研究：在稀疏 VGG-16 上，与固定 c 和无 γ 的测试准确率（%）比较。结果以干净/鲁棒准确率（%）表示。在训练轮数相同的情况下，几乎所有情况下我们的方法都比固定 c 和无 γ 具有更高的准确率。

		90% 稀疏度			99% 稀疏度		
		固定 c_t	No γ	OURS	固定 c_t	No γ	OURS
AT	第 20	54.1/36.2	28.6/20.1	**63.6**/**37.3**	10.0/10.0	10.0/10.0	**56.4**/**31.4**
	第40位	58.9/37.1	20.4/13.0	**64.9**/**37.9**	10.0/10.0	10.0/10.0	**57.7**/**34.5**
	第70位	66.8/41.6	19.9/14.7	**75.1**/**45.2**	10.0/10.0	10.0/10.0	**66.0**/**39.4**
	第90	67.7/43.3	21.8/15.6	**74.1**/**44.8**	10.0/10.0	10.0/10.0	**65.8**/**39.8**
	第140位	71.4/43.4	20.0/12.1	**77.4**/**43.8**	10.0/10.0	10.0/10.0	**69.8**/**41.2**
	第200	71.7/43.0	20.5/9.5	**78.1**/**44.6**	10.0/10.0	10.0/10.0	**70.7**/**42.0**
STANDARD	第 20 个	80.9/0.0	70.6/0.0	**81.8**/o.o	**73.7**/o.o	51.8/0.0	69.8/0.0
	第40位	**83.3**/o.o	68.0/0.0	82.4/0.0	**74.9**/o.o	55.2/0.0	73.7/0.0
	第70位	**90.2**/o.o	77.3/0.0	89.7/0.0	**84.1**/o.o	65.9/0.0	83.7/0.0
	第90	**89.8**/o.o	77.8/0.0	89.3/0.0	80.5/0.0	67.8/0.0	**83.9**/o.o
	第140位	92.4/0.0	80.7/0.0	**92.5**/o.o	**87.2**/o.o	71.9/0.0	86.9/0.0
	第 200 个	92.1/0.0	78.6/0.0	**92.6**/o.o	86.4/0.0	70.0/0.0	**87.1**/o.o

不如另外两种方法。对于“Fix c_t”，其收敛速度与我们的方法相近，但我们的方法往往能取得更好的最终得分。因此，c_t 的自适应更新以及缩放参数 γ 的乘法运算对于加速都很重要。

6.6. 缩放参数设置

缩放参数 γ 用于避免因近似 c_t^*时产生的误差而引入较大方差，以及对抗训练带来的偏差。γ 的选择很重要，可视为一个超参数调优过程。我们的结果基于 $\gamma = 0.1$。我们检查了从 0 到 1 的不同取值，发现通常最好不要将 γ 设为接近 1 或 0。如果将 γ 设为接近 1，我们就无法完全避免方差增大，从而导致性能下降，这与表 6 中的“No γ”类似。如果 γ 设置得太小，例如 0.01，旧梯度的权重就会太小，旧梯度对模型更新的影响也有限，导致随机梯度下降速度变慢且训练不稳定（更多讨论见附录 B.3）。

7. 讨论与结论

我们从优化角度提出了一种用于稀疏训练的自适应梯度校正（AGENT）方法，以提高时间效率并减少训练不稳定性。该方法可融入任何基于随机梯度下降的稀疏训练流程，并适用于标准和对抗场景。为精细控制当前梯度与先前梯度之间的平衡，我们利用损失信息分析梯度变化，并为旧梯度添加自适应权重。此外，我们设计了一个缩放参数，以减少对抗样本引入的梯度估计器偏差，并改善自适应权重估计的最坏情况。理论上，我们证明 AGENT 能够加快稀疏训练的收敛速率。在多个数据集、模型架构和稀疏度上的实验结果表明，就准确率而言，我们的方法优于最先进的稀疏训练方法，最高提升**5.0%**；在达到相同准确率的情况下，训练轮数最多可减少**52.1%** 。

有多种方法可以降低我们的 AGENT 中的浮点运算次数。与 SVRG 类似，我们的 AGENT 会因每次迭代中额外执行前向传播和反向传播来计算旧梯度，而增加训练所需的浮点运算次数。为减少浮点运算次数，第一种方法是使用稀疏梯度 [51]，这能有效降低稀疏训练中的反向传播开销，也很容易应用于我们的方法。第二种方法是采用并行计算 Allen-Zhu 和 Hazan [49]。由于针对旧模型参数的额外前向传播和反向传播可以完全并行化，因此我们可以将其视为把小批量大小加倍。第三，我们可以借鉴 SAGA [40] 的思路，存储每个单个样本对应的梯度。这样便不需要额外的前向传播和反向传播步骤，从而节省计算开销。不过，这种方法需要额外的内存来存储梯度。

参考文献

[1] Decebal Constantin Mocanu、Elena Mocanu、Peter Stone、Phuong H Nguyen、Madeleine Gibescu 和 Antonio Liotta。《受网络科学启发的自适应稀疏连接人工神经网络的可扩展训练》。《自然·通讯》，9(1):1–12，2018年。[2] Utku Evci、Trevor Gale、Jacob Menick、Pablo Samuel Castro 和 Erich Elsen。《操纵彩票：让所有中奖彩票都成为赢家》。载于国际机器学习会议，第 2943–2952 页。PMLR，2020年。[3] Chuang Liu、Xueqi Ma、Yinbing Zhan、Liang Ding、Dapeng Tao、Bo Du、Wenbin Hu 和 Danilo Mandic。《面向图神经网络稀疏训练的全面图渐进式剪枝》。*arXiv* 预印本 *arXiv:2207.08629*，2022年。[4] Guillaume Bellec、David Kappel、Wolfgang Maass 和 Robert Legenstein。《深度重连：训练极稀疏深度网络》。国际学习表征会议（*ICLR*），2018年。[5] Tim Dettmers 和 Luke Zettlemoyer。《从零开始构建稀疏网络：加快训练而不损失性能》。*arXiv* 预印本 *arXiv:1907.04840*，2019年。[6] Shiwei Liu、Lu Yin、Decebal Constantin Mocanu 和 Mykola Pechenizkiy。《我们真的需要稠密过参数化吗？稀疏训练中的及时过参数化》。载于国际机器学习会议，第 6989–7000 页。PMLR，2021年。[7] Ozan Özdenizci 和 Robert Legenstein。《通过贝叶斯连接采样训练具有对抗鲁棒性的稀疏网络》。载于国际机器学习会议，第 8314–8324 页。PMLR，2021年。[8] Rong Yu 和 Peichun Li。《迈向移动边缘计算中资源高效的联邦学习》。*IEEENetwork*，35(1):148–155，2021年。[9] Johanna Rock、Wolfgang Roth、Mate Toth、Paul Meissner 和 Franz Pernkopf。《用于汽车雷达干扰抑制的资源高效深度神经网络》。*IEEE* 信号处理选题期刊，15(4):927–940，2021年。[10] Clayton Frederick Souza Leite 和 Yu Xiao。《面向资源高效深度活动识别的最优传感器通道选择》。载于第 *20* 届国际传感器网络信息处理会议论文集（与 *2021* 年 *CPS-IoT* 周同期举办），第 371–383 页，2021年。[11] Torsten Hoefler、Dan Alistarh、Tal Ben-Nun、Nikoli Dryden 和 Alexandra Peste。《深度学习中的稀疏度：神经网络中用于高效推理和训练的剪枝与增长》。*arXiv* 预印本 *arXiv:2102.00554*，2021年。[12] Laura Graesser、Utku Evci、Erich Elsen 和 Pablo Samuel Castro。《深度强化学习中的稀疏训练现状》。载于国际机器学习会议，第 7766–7792 页。PMLR，2022年。[13] Vikash Sehwag、Shiqi Wang、Prateek Mittal 和 Suman Jana。《HYDRA：剪枝具有对抗鲁棒性的神经网络》。神经信息处理系统进展，33:19655–19666，2020年。[14] Brian Bartoldson、Ari Morcos、Adrian Barbu 和 Gordon Erlebacher。《神经网络剪枝中的泛化—稳定性权衡》。神经信息处理系统进展，33:20852–20864，2020年。[15] Xia Xiao、Zigeng Wang 和 Sanguthevar Rajasekaran。《Autoprune：通过正则化辅助参数实现自动网络剪枝》。神经信息处理系统进展，32，2019年。

[16] Shaokai Ye、Kaidi Xu、Sijia Liu、Hao Cheng、Jan-Henrik Lambrechts、Huan Zhang、Aojun Zhou、Kaisheng Ma、Yanzhi Wang 和 Xue Lin。对抗鲁棒性与模型压缩，或两者兼得? 发表于*IEEE/CVF*国际计算机视觉会议论文集，第 111–120 页，2019年。[17] Souvik Kundu、Mahdi Nazemi、Peter A Beerel 和 Massoud Pedram。DNR：通过深度神经网络的动态网络重连实现可调的鲁棒剪枝框架。发表于第*26*届亚太设计自动化会议论文集，第 344–350 页，2021年。[18] Difan Zou、Pan Xu 和 Quanquan Gu。子采样随机方差缩减梯度朗之万动力学。发表于人工智能不确定性国际会议，2018年。[19] Changyou Chen、Wenlin Wang、Yizhe Zhang、Qinliang Su 和 Lawrence Carin。一类实用方差缩减随机梯度 MCMC 方法的收敛性分析。《中国科学：信息科学》，62(1):1–13，2019年。[20] Eduard Gorbunov、Filip Hanzely 和 Peter Richtárik。随机梯度下降的统一理论：方差缩减、采样、量化与坐标下降。发表于人工智能与统计国际会议，第 680–690 页。PMLR，2020年。[21] Kumar Avinava Dubey、Sashank J Reddi、Sinead A Williamson、Barnabas Poczos、Alexander J Smola 和 Eric P Xing。随机梯度朗之万动力学中的方差缩减。发表于神经信息处理系统进展，29:1154–1162，2016年。[22] Niladri Chatterji、Nicolas Flammarion、Yian Ma、Peter Bartlett 和 Michael Jordan。论随机梯度蒙特卡洛方差缩减理论。发表于国际机器学习会议，第 764–773 页。PMLR，2018年。[23] Yan Li、Ethan Fang、Huan Xu 和 Tuo Zhao。可分数据上基于梯度下降的对抗训练的隐式偏置。发表于国际学习表征会议，2020年。[24] Itay Hubara、Brian Chmiel、Moshe Island、Ron Banner、Joseph Naor 和 Daniel Soudry。加速稀疏神经网络训练：一种可证明且高效的 n:m 可转置掩码查找方法。发表于神经信息处理系统进展，34，2021年。[25] Beidi Chen、Tri Dao、Kaizhao Liang、Jiaming Yang、Zhao Song、Atri Rudra 和 Christopher Re。像素化蝴蝶：用于神经网络模型的简单高效稀疏训练。*arXiv* 预印本 *arXiv:2112.00029*，2021年。[26] Jonathan Frankle 和 Michael Carbin。彩票假设：寻找稀疏可训练神经网络。国际学习表征会议（*ICLR*），2019年。[27] Hesham Mostafa 和 Xin Wang。通过动态稀疏重参数化实现深度卷积神经网络的参数高效训练。发表于国际机器学习会议，第 4646–4655 页。PMLR，2019年。[28] Siddhant Jayakumar、Razvan Pascanu、Jack Rae、Simon Osindero 和 Erich Elsen。Top-kast：Top-k 始终稀疏训练。发表于神经信息处理系统进展，33:20744–20754，2020年。[29] Xiao Zhou、Weizhong Zhang、Hang Xu 和 Tong Zhang。具有全局稀疏性约束的神经网络有效稀疏化。发表于*IEEE/CVF*计算机视觉与模式识别会议论文集，第 3599–3608 页，2021年。[30] Jonathan Schwarz、Siddhant Jayakumar、Razvan Pascanu、Peter E Latham 和 Yee Teh。幂传播：一种诱导稀疏性的权重重参数化方法。发表于神经信息处理系统进展，34:28889–28903，2021年。

[31] Shaoyi Huang、Bowen Lei、Dongkuan Xu、Hongwu Peng、Yue Sun、Mimi Xie 和 Caiwen Ding。通过平衡探索-利用权衡实现动态稀疏训练。*arXiv* 预印本 *arXiv:2211.16667*，2022年。[32] Bowen Lei、Ruqi Zhang、Dongkuan Xu 和 Bani Mallick。校准被操纵的彩票：让所有票据都可靠。*arXiv* 预印本 *arXiv:2302.09369*，2023。[33] Shaoyi Huang、Haowen Fang、Kaleel Mahmood、Bowen Lei、Nuo Xu、Bin Lei、Yue Sun、Dongkuan Xu、Wujie Wen 和 Caiwen Ding。受神经发生动态启发的脉冲神经网络训练加速。*arXiv* 预印本 *arXiv:2304.12214*，2023。[34] Adnan Siraj Rakin、Zhezhi He、Li Yang、Yanzhi Wang、Liqiang Wang 和 Deliang Fan。鲁棒稀疏正则化：同时优化神经网络的鲁棒性与紧凑性。*arXiv* 预印本 *arXiv:1905.13074*，2019年。[35] Naveed Akhtar 和 Ajmal Mian。计算机视觉中深度学习面临的对抗攻击威胁：综述。*IEEE Access*，6:14410–14430，2018年。[36] Hongwu Peng、Shaoyi Huang、Tong Geng、Ang Li、Weiwen Jiang、Hang Liu、Shusen Wang 和 Caiwen Ding。使用列平衡块剪枝在 FPGA 上加速基于 Transformer 的深度学习模型。发表于*2021*年第*22*届国际电子设计质量研讨会（*ISQED*），第 142–148 页。IEEE，2021年。[37] Panjie Qi、Edwin Hsing-Mean Sha、Qingfeng Zhuge、Hongwu Peng、Shaoyi Huang、Zhenglun Kong、Yuhong Song 和 Bingbing Li。通过硬件设计与模型压缩协同优化实现 Transformer 加速的框架。发表于*2021 IEEE/ACM*国际计算机辅助设计会议（*ICCAD*），第 1–9 页。IEEE，2021年。[38] Nicolas Roux、Mark Schmidt 和 Francis Bach。一种针对有限训练集、具有指数收敛速率的随机梯度法。发表于神经信息处理系统进展，25，2012年。[39] Rie Johnson 和 Tong Zhang。使用预测方差缩减加速随机梯度下降。发表于神经信息处理系统进展，26:315–323，2013年。[40] Aaron Defazio、Francis Bach 和 Simon Lacoste-Julien。SAGA：一种支持非强凸复合目标的快速增量梯度法。发表于神经信息处理系统进展，27，2014年。[41] Lin Xiao 和 Tong Zhang。一种具有渐进方差缩减的近端随机梯度法。*SIAM*优化杂志，24(4):2057–2075，2014年。[42] Fanhua Shang、Kaiwen Zhou、Hongying Liu、James Cheng、Ivor W Tsang、Lijun Zhang、Dacheng Tao 和 Licheng Jiao。VR-SGD：一种用于机器学习的简单随机方差缩减方法。*IEEE*知识与数据工程汇刊，32(1):188–202，2018年。[43] Robert M Gower、Mark Schmidt、Francis Bach 和 Peter Richtárik。机器学习的方差缩减方法。*IEEE*会刊，108(11):1968–1983，2020年。[44] Lam M Nguyen、Jie Liu、Katya Scheinberg 和 Martin Takáč。Sarah：一种使用随机递归梯度解决机器学习问题的新方法。发表于国际机器学习会议，第 2613–2621 页。PMLR，2017年。[45] Cong Fang、Chris Junchi Li、Zhouchen Lin 和 Tong Zhang。Spider：通过随机路径积分微分估计器实现近乎最优的非凸优化。发表于神经信息处理系统进展，31，2018年。[46] Ashok Cutkosky 和 Francesco Orabona。非凸随机梯度下降中的基于动量的方差缩减。发表于神经信息处理系统进展，32，2019年。

[47] El Mahdi Chayti 和 Sai Praneeth Karimireddy。利用辅助信息进行优化。*arXiv*预印本 *arXiv:2206.00395*，2022年。[48] Xiao Zhou、Weizhong Zhang、Zonghao Chen、Shizhe Diao 和 Tong Zhang。通过前向与反向传播稀疏化实现高效神经网络训练。神经信息处理系统进展，34：15216–15229，2021年。[49] Zeyuan Allen-Zhu 和 Elad Hazan。通过方差缩减加速非凸优化。载于国际机器学习会议，第699–707页。PMLR，2016年。[50] Jack Baker、Paul Fearnhead、Emily B Fox 和 Christopher Nemeth。随机梯度马尔可夫链蒙特卡罗的控制变量。统计与计算，29(3)：599–615，2019年。[51] Melih Elibol、Lihua Lei 和 Michael I Jordan。利用稀疏梯度进行方差缩减。*arXiv*预印本 *arXiv:2001.09623*，2020年。[52] Wei Deng、Qi Feng、Georgios Karagiannis、Guang Lin 和 Faming Liang。通过方差缩减加速副本交换随机梯度马尔可夫链蒙特卡罗的收敛。*arXiv*预印本 *arXiv:2010.01084*，2020年。[53] Ning Qian。论梯度下降学习算法中的动量项。神经网络，12(1)：145–151，1999年。[54] Sebastian Ruder。梯度下降优化算法综述。*arXiv*预印本 *arXiv:1609.04747*，2016年。[55] Diederik P Kingma 和 Jimmy Ba。Adam：一种随机优化方法。*arXiv*预印本 *arXiv:1412.6980*，2014年。[56] Sashank J Reddi、Ahmed Hefny、Suvrit Sra、Barnabás Póczos 和 Alex Smola。非凸优化的随机方差缩减。载于国际机器学习会议，第314–323页。PMLR，2016年。[57] Alex Krizhevsky、Geoffrey Hinton 等。利用微型图像学习多层特征。多伦多大学技术报告，2009年。[58] Yuval Netzer、Tao Wang、Adam Coates、Alessandro Bissacco、Bo Wu 和 Andrew Y Ng。利用无监督特征学习识别自然图像中的数字。2011年。[59] Olga Russakovsky、Jia Deng、Hao Su、Jonathan Krause、Sanjeev Satheesh、Sean Ma、Zhiheng Huang、Andrej Karpathy、Aditya Khosla、Michael Bernstein 等。ImageNet大规模视觉识别挑战。国际计算机视觉杂志，115(3)：211–252，2015年。[60] Karen Simonyan 和 Andrew Zisserman。用于大规模图像识别的超深卷积网络。国际学习表征会议（*ICLR*），2015年。[61] Kaiming He、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun。用于图像识别的深度残差学习。载于*IEEE*计算机视觉与模式识别会议会刊，第770–778页，2016年。[62] Sergey Zagoruyko 和 Nikos Komodakis。宽残差网络。英国机器视觉会议，2016年。[63] Aleksander Madry、Aleksandar Makelov、Ludwig Schmidt、Dimitris Tsipras 和 Adrian Vladu。迈向能够抵御对抗攻击的深度学习模型。载于国际学习表征会议（*ICLR*），2018年。[64] Hongyang Zhang、Yaodong Yu、Jiantao Jiao、Eric Xing、Laurent El Ghaoui 和 Michael Jordan。鲁棒性与准确率之间有理论依据的权衡。载于国际机器学习会议，第7472–7482页。PMLR，2019年。

[65] Utku Evci、Yani Ioannou、Cem Keskin 和 Yann Dauphin。稀疏神经网络中的梯度流及彩票票据为何胜出。载于*AAAI*人工智能会议会刊，第36卷，第6577–6586页，2022年。[66] Varun Sundar 和 Rajat Vadiraj Dwaraknath。 [可复现性报告] 操纵彩票：让所有票据都成为赢家。*arXiv*预印本 *arXiv:2103.15767*，2021年。[67] Ilya Loshchilov 和 Frank Hutter。解耦权重衰减正则化。载于国际学习表征会议（*ICLR*），2019年。[68] Kaiming He、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun。深入研究整流器：在ImageNet分类中超越人类水平的性能。载于*IEEE/CVF*国际计算机视觉会议会刊，第1026–1034页，2015年。[69] Mahdi Nikdan、Tommaso Pegolotti、Eugenia Iofinova、Eldar Kurtic 和 Dan Alistarh。Sparse-prop：用于边缘端神经网络加速训练的高效稀疏反向传播。载于国际机器学习会议，第26215–26227页。PMLR，2023。[70] Chao Fang、Wei Sun、Aojun Zhou 和 Zhongfeng Wang。通过算法、架构与数据流协同设计实现高效的 N:M 稀疏 DNN 训练。*IEEE*集成电路与系统计算机辅助设计汇刊，2023。[71] Lei Zhang、Jie Zhang、Bowen Lei、Subhabrata Mukherjee、Xiang Pan、Bo Zhao、Caiwen Ding、Yao Li 和 Dongkuan Xu。通过模型增强加速数据集蒸馏。载于*IEEE/CVF*计算机视觉与模式识别会议会刊，第11950–11959页，2023。[72] Dongyao Zhu、Bowen Lei、Jie Zhang、Yanbo Fang、Yiqun Xie、Ruqi Zhang 和 Dongkuan Xu。重新思考数据蒸馏：切勿忽视校准。载于*IEEE/CVF*国际计算机视觉会议会刊，第4935–4945页，2023。[73] Cheng Wang、Jiacheng Sun、Zhenhua Dong、Ruixuan Li 和 Rui Zhang。用于点击率预测中类别数据蒸馏的梯度匹配。载于第*17*届*ACM*推荐系统会议会刊，第161–170页，2023。[74] Jiangyuan Li、Thanh Nguyen、Chinmay Hegde 和 Ka Wai Wong。隐式稀疏正则化：深度与提前停止的影响。神经信息处理系统进展，34：28298–28309，2021年。[75] Hung-Hsu Chou、Johannes Maly 和 Holger Rauhut。多即是少：通过过参数化诱导稀疏性。《信息与推断：*IMA*期刊》，12(3)：iaad012，2023。[76] Jiangyuan Li、Thanh V Nguyen、Chinmay Hegde 和 Raymond KW Wong。组稀疏性的隐式正则化。*arXiv*预印本 *arXiv:2301.12540*，2023。[77] Zhiyao Li、Jiaxiang Li、Taijie Chen、Dimin Niu、Hongzhong Zheng、Yuan Xie 和 Mingyu Gao。Spada：利用自适应数据流加速稀疏矩阵乘法。载于第*28*届*ACM*编程语言与操作系统架构支持国际会议会刊

p_{eratin} *g Sy*系统，第*2*卷， pages 747–761, 2023。

A. 附录：收敛速率理论证明

本节将详细证明我们 AGENT 方法的收敛速率。首先，我们给出一些假设，并据此提出若干有用的引理。然后，我们将基于这些引理确立我们 AGENT 方法的收敛速率。

A.1. 算法重构

我们将自适应梯度校正（AGENT）重构为更便于数学分析的版本，如算法 2 所示。

算法 2 自适应梯度校正 输入：初始化 $\boldsymbol{\theta}_0^0$和 $c_{-1}=0$，设置训练轮次数量 S、轮次长度 m、步长 η_t、缩放参数γ和平滑因子 α对于 $s=0$到 $S-1$执行~$\boldsymbol{\theta}=\boldsymbol{\theta}_0^s\widetilde{g}=(\sum_{i=1}^{N}\nabla l(\mathrm{x}_i;\ \widetilde{\boldsymbol{\theta}}))/N$通过公式（4）计算 $\widehat{c}_s^*\,\widetilde{c}_s=(1-\alpha)\widetilde{c}_{s-1}+\alpha\widehat{c}_{s\sim}^*c_s=\gamma c_s$对于 $t=0$到 $m-1$执行 从 B_t 中抽取一个大小为 n的小批量数据 $\boldsymbol{\theta}_{t+1}^s=\boldsymbol{\theta}_t^s-\eta_t\left(\frac{1}{n}\sum_{i\in\mathrm{B}_t}\left(g_i(\boldsymbol{\theta}_t^s)\ -c_s\cdot g_i(\widetilde{\boldsymbol{\theta}})\right)+c_s\cdot\widetilde{g}\right)$结束循环$\boldsymbol{\theta}_0^{s+1}=\boldsymbol{\theta}_m^s$结束循环 输出：从 $\{\{\boldsymbol{\theta}_t^s\}_{t=0}^{m-1}\}_{s=0}^{S-1}$ 中均匀随机选取的迭代点 $\boldsymbol{\theta}_\pi$

–

A.2. 假设

L-光滑：若可微函数 l： $\mathbb{R}^n\to\mathbb{R}$ 对于所有 $\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^n$ 均满足$||\nabla l(\boldsymbol{x})\ -\nabla l(\boldsymbol{y})||\le L||\boldsymbol{x}-\boldsymbol{y}||$，则称其为 L-光滑函数。一个等价定义是，对于所有 $\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^n$:

$$-\frac{L}{2}||\boldsymbol{x}-\boldsymbol{y}||^2\le l(x)-l(y)-\langle\nabla l(\boldsymbol{x}),\boldsymbol{x}-\boldsymbol{y}\rangle\le\frac{L}{2}||\boldsymbol{x}-\boldsymbol{y}||^2$$

σ-有界：若函数 l 的梯度有界，且上界为 σ，即对于所有 $i\in[N]$ 和 $\boldsymbol{x}\in\mathbb{R}^n$，均有 $||\nabla l_i(\boldsymbol{x})||\le\sigma$

A.3. 分析框架

在上述假设下，我们可以开始分析 AGENT 在**算法 2**中的收敛速率。为了更清晰地介绍收敛性分析，我们为证明提供一个简要的分析框架。

- 首先，我们需要证明，在适当选择 c_s的情况下，我们的梯度估计器的方差小于小批量随机方差缩减梯度法的方差。由于在标准训练中，AGENT 和小批量随机方差缩减梯度法的梯度估计器均为无偏估计量，因此我们只需证明我们的界 $E[||\mathbf{u}_t||^2]$小于小批量随机方差缩减梯度法的界。（见引理 1）

- 基于上述事实，接下来我们将应用李雅普诺夫函数证明 AGENT 在任意一个训练轮次中的收敛速率。（见引理 3）
- 然后，我们将之前的结果推广到整个训练轮次（从第 0 轮到第 S轮），并推导出**算法 2**输出的 θ_π 的收敛速率。（见引理 4）

• 最后，我们将 AGENT 的收敛速率与小批量随机方差缩减梯度法进行比较。根据稀疏学习的实际情况设置引理 4 中的参数后，我们得到了一个比小批量随机方差缩减梯度法更严格的界。

A.4. 引理

我们首先将步长记为 $\eta_t = N \cdot h_{t\circ}$ 由于我们主要关注单个训练轮次，因此省略上标 s，并将 $\mathbf{u}_t = \frac{1}{n}\sum_{i\in\mathbf{B}_t}\left(g_i(\boldsymbol{\theta}_t) - c\cdot g_i(\widetilde{\boldsymbol{\theta}})\right) + c\cdot\widetilde{g}$ 记为算法中的梯度估计器，将 $\boldsymbol{\tau}_t = \frac{1}{n}\sum_{i\in\mathbf{B}_t}\left(g_i(\boldsymbol{\theta}_t) - c\cdot g_i(\widetilde{\boldsymbol{\theta}})\right)$ 记为.....，那么 **算法 2**中的更新过程可替换为 $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \cdot \mathbf{u}_t$

A.4.1. 引理 1

对于上述定义的 $\mathbf{u}_t$，若函数 l 是 L-光滑、 λ -强凸且具有 σ-有界梯度的函数，则有以下结果:

$$\mathbb{E}\left[||\mathbf{u}_t||^2\right] \leq 2\mathbb{E}\left[||g(\boldsymbol{\theta}_t)||^2\right] + \frac{4c^2L^2}{n}\mathbb{E}\left[||\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}||^2\right] + \frac{4(1-c)^2}{n}\sigma^2 \tag{6}$$

证明:

$$\begin{aligned}
\mathbb{E}\left[||\mathbf{u}_t||^2\right] &= \mathbb{E}\left[||\boldsymbol{\tau}_t + c\cdot\widetilde{g}||^2\right] = \mathbb{E}\left[||\boldsymbol{\tau}_t + c\cdot\widetilde{g} - g(\boldsymbol{\theta}_t) + g(\boldsymbol{\theta}_t)||^2\right] \\
&\leq 2\mathbb{E}\left[||g(\boldsymbol{\theta}_t)||^2\right] + 2\mathbb{E}\left[||\boldsymbol{\tau}_t - \mathbb{E}(\boldsymbol{\tau}_t)||^2\right] \leq 2\mathbb{E}\left[||g(\boldsymbol{\theta}_t)||^2\right] + \frac{2}{n}\mathbb{E}\left[\boldsymbol{\tau}_t^2\right] \\
&= 2\mathbb{E}\left[||g(\boldsymbol{\theta})||^2\right] + \frac{2}{n}\mathbb{E}\left[||c(g_i(\boldsymbol{\theta}_t) - g_i(\tilde{\boldsymbol{\theta}})) + (1-c)g_i(\boldsymbol{\theta}_t)||^2\right] \\
&\leq 2\mathbb{E}\left[||g(\boldsymbol{\theta})||^2\right] + \frac{4}{n}\mathbb{E}\left[||c(g_i(\boldsymbol{\theta}_t) - g_i(\tilde{\boldsymbol{\theta}}))||^2\right] + \frac{4(1-c)^2}{n}\mathbb{E}\left[||g_i(\boldsymbol{\theta}_t)||^2\right] \\
&\leq 2\mathbb{E}\left[||g(\boldsymbol{\theta})||^2\right] + \frac{4c^2L^2}{n}\mathbb{E}\left[||\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}||^2\right] + \frac{4(1-c)^2}{n}\sigma^2
\end{aligned}$$

第一个和第三个不等式是因为 $||a+b||^2 \leq 2||a||^2 + 2||b||^2$，第二个不等式根据 $\mathbb{E}\left[||\boldsymbol{\tau} - E\left[\boldsymbol{\tau}\right]||^2\right] \leq \mathbb{E}\left[||\boldsymbol{\tau}||^2\right]$ 得出，最后一个不等式则根据函数 l_i 的 L-光滑性和 σ-有界性得出。

注 5. 与小批量随机方差缩减梯度法的梯度估计器相比，当 L 较大、 σ 相对较小且 c 选择适当时，$\mathbb{E}[||\boldsymbol{u}_t||^2]$ 的界更小。

A.4.2. 引理 2

$$\mathbb{E}\left[l(\boldsymbol{\theta}_{t+1})\right] \leq \mathbb{E}\left[l(\boldsymbol{\theta}_t) + \eta_t||g(\boldsymbol{\theta}_t)||^2 + \frac{L\eta^2}{2}||\mathbf{u}_t||^2\right] \tag{7}$$

证明:

根据函数 l 的 L-光滑性，我们有

$$\mathbb{E}\left[l(\boldsymbol{\theta}_{t+1})\right] \leq \mathbb{E}\left[l(\boldsymbol{\theta}_t) + \langle g(\boldsymbol{\theta}_t), \boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\rangle + \frac{L}{2}||\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t||^2\right]$$

根据算法 2 的更新过程和无偏性，右侧还可以进一步由以下上界限定

$$\mathbb{E}\left[l(\boldsymbol{\theta}_t) + \eta_t||g(\boldsymbol{\theta}_t)||^2 + \frac{L\eta_t^2}{2}||\mathbf{u}_t||^2\right]$$

A.4.3. 引理 3

对于 b_t， b_{t+1}， $\zeta_t > 0$ 、 b_t 和 b_{t+1} 具有以下关系

$$b_t = b_{t+1}(1 + \eta_t\zeta_t + \frac{4c^2\eta_t^2L^2}{n}) + 2\frac{c^2\eta_t^2L^3}{n}$$

并定义

$$\Phi_t \coloneqq \eta_t - \frac{b_{t+1}\eta_t}{\zeta_t} - \eta_t^2L - 2b_{t+1}\eta_t^2$$

$$\Psi_t \coloneqq \mathbb{E}\left[l(\boldsymbol{\theta}_t) + b_t||\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}||^2\right] \tag{8}$$

可以选取 η_t、 ζ_t 和 b_{t+1}，使得 $\Phi_t > 0$。那么，算法**1**中的 x_t 满足以下界：

$$\mathbb{E}[||g(\boldsymbol{\theta}_t)||^2] \leq \frac{\Psi_t - \Psi_{t+1} + \frac{2(L\eta_t^2+2b_{t+1}\eta_t^2)(1-c)^2}{n}\sigma^2}{\Phi_t}$$

证明：
我们应用李雅普诺夫函数

$$\Psi_t = \mathbb{E}\left[l(\boldsymbol{\theta}_t) + b_t||\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}||^2\right]$$

˜接下来，我们需要对 $||\boldsymbol{\theta}_t - \boldsymbol{\theta}||$ 进行界定

$$\begin{aligned}
\mathbb{E}\left[||\boldsymbol{\theta}_{t+1} - \tilde{\boldsymbol{\theta}}||^2\right] &= \mathbb{E}\left[||\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t + \boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}||^2\right] \\
&= \mathbb{E}\left[||\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t||^2 + ||\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}||^2 + 2\langle\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t, \boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}\rangle\right] \\
&= \mathbb{E}\left[\boldsymbol{\eta_t^2}||\mathbf{u}_t||^2 + ||\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}||^2\right] - 2\eta_t\mathbb{E}\left[\langle g(\boldsymbol{\theta}_t), \boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}\rangle\right] \\
&\leq \mathbb{E}[\eta_t^2||\mathbf{u}_t^{s+1}||^2 + ||\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}||^2] + 2\eta_t\mathbb{E}\left[\frac{1}{2\zeta_t}||g(\boldsymbol{\theta}_t)|| + \frac{\zeta_t}{2}||\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}||^2\right]
\end{aligned} \tag{9}$$

第三个等号源于更新的无偏性，最后一个不等式则由柯西-施瓦茨不等式和杨氏不等式得出。将公式（6）、公式（7）和公式（9）代入公式（8），可得以下界：

$$\begin{aligned}
\Psi_{t+1} &\leq \mathbb{E}\left[l(\boldsymbol{\theta}_t)\right] + \left(b_{t+1}(1 + \eta_t\zeta_t + \frac{4c^2\eta_t^2L^2}{n}) + \frac{2c^2\eta_t^2L^3}{n}\right)\mathbb{E}[||\boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}||^2] \\
&\quad - (\eta_t - \frac{b_{t+1}\eta_t}{\zeta_t} - L\eta_t^2 - 2b_{t+1}\eta_t^2)\mathbb{E}\left[||g(\boldsymbol{\theta}_t)||^2\right] + 4(\frac{L\eta_t^2}{2} + b_{t+1}\eta_t^2)\frac{(1-c)^2}{n}\sigma^2 \\
&= \Psi_t - (\eta_t - \frac{b_{t+1}\eta_t}{\zeta_t} - L\eta_t^2 - 2b_{t+1}\eta_t^2)\mathbb{E}\left[||g(\boldsymbol{\theta}_t)||^2\right] + 4(\frac{L\eta_t^2}{2} + b_{t+1}\eta_t^2)\frac{(1-c)^2}{n}\sigma^2
\end{aligned}$$

A.4.4. 引理 4

现在我们考虑训练轮次的影响，并用 s 表示轮次编号。令 $b_m^s = 0, \eta_t^s = \eta$,$\zeta_t^s = \zeta$和 $b_t^s = b_{t+1}^s($ $1+\eta\zeta+\frac{4c_s^2\eta L^2}{n})+2\frac{c_s^2\eta^2L^2}{n}$, $\Phi_t^s = \eta - \frac{b_{t+1}^s\eta}{\zeta_t} - \eta^2L - 2b_{t+1}^s\eta^2$定义 $\phi \coloneqq \min_{t,s}\Phi_t^s$。由此可得：

$$\mathbb{E}[||g(\boldsymbol{\theta_\pi})||^2] \leq \frac{l(\theta_0) - l(\theta_*)}{T\phi} + \sum_{s=0}^{S-1}\sum_{t=0}^{m-1} \frac{2(L + 2b_{t+1}^s)(1-c_s)^2\eta^2\sigma^2}{Tn\phi}$$

证明：

U在 $\eta_t^s = \eta$ 的条件下，我们对**引理 3**应用裂项求和，随后可得：

$$\sum_{t=1}^{m-1} \mathbb{E}[||g(\boldsymbol{\theta}_t^s)||^2] \leq \frac{\Psi_0^s - \Psi_m^s}{\phi} + \sum_{t=0}^{m-1} \frac{2(L + 2b_{t+1}^s)(1-c_s)^2\eta^2\sigma^2}{n\phi}$$

根据前面的定义，我们知道 $\Psi_0^s = l(\tilde{\boldsymbol{\theta}}^s)$、 $\Psi_m^s = l(\tilde{\boldsymbol{\theta}}^{s+1})$，代入前面的等式可得：

$$\sum_{t=1}^{m-1} \mathbb{E}[||g(\boldsymbol{\theta}_t^s)||^2] \leq \frac{l(\tilde{\boldsymbol{\theta}}^s) - l(\tilde{\boldsymbol{\theta}}^{s+1})}{\phi} + \sum_{t=0}^{m-1} \frac{2(L + 2b_{t+1}^s)(1-c_s)^2\eta^2\sigma^2}{n\phi}$$

对所有训练轮次求和，并利用 $\tilde{\boldsymbol{\theta}}^0 = \theta_0$、 $l(\tilde{\boldsymbol{\theta}}^S) \leq l(\boldsymbol{\theta}_*)$，即可得到：

$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=1}^{m-1} \mathbb{E}[||g(\boldsymbol{\theta}_t^s)||^2] \leq \frac{l(\boldsymbol{\theta}_0) - l(\boldsymbol{\theta}_*)}{\phi} + \sum_{s=0}^{S-1}\sum_{t=0}^{m-1} \frac{2(L + 2b_{t+1}^s)(1-c_s)^2\eta^2\sigma^2}{Tn\phi} \tag{10}$$

A.5. 定理

A.5.1. 定理1

定义 $\xi_s = \sum_{t=0}^{m-1}(L + 2b_{t+1}^s)$ 和 $\xi := \min_s \xi_s$。令 $\eta = \frac{\mu n}{LN^\alpha}(0 < \mu < 1)\, and\, (0 < \alpha \leq 1)$, $\zeta = \frac{L}{N^{\alpha/2}}$ and $m = \frac{N^2}{\mu n}$。则存在常数 $\nu, \mu, \alpha, \kappa > 0$，使得 $\phi \geq \frac{n\nu}{LN^\alpha}$ 和 $\xi \leq \kappa L$。然后， $\mathbb{E}[||g(\boldsymbol{\theta_\pi})||^2]$ 可进一步界定为：

$$\mathbb{E}[||g(\boldsymbol{\theta_\pi})||^2] \leq \frac{(l(\boldsymbol{\theta}_0) - l(\boldsymbol{\theta}_*))LN^\alpha}{Tn\nu} + \frac{2\kappa\mu^2\sigma^2}{N^\alpha\nu m}$$

证明：

对关系 $b_t^s = b_{t+1}^s(1 + \eta_t\zeta_t + \frac{4c_s^2\eta_t^2L^2}{n}) + 2\frac{c_s^2\eta_t^2L^3}{n}$ 应用等比数列的求和公式，可得 $b_t^s = \frac{2c_s^2\eta^2L^3}{n}\frac{(1+\omega_s)^{m-t}-1}{\omega_s}$其中：

$$\omega_s = \eta\zeta + \frac{4c_s^2\eta^2L}{n} = \frac{\mu n}{N^{\frac{3\alpha}{2}}} + \frac{4c_s^2\mu^2 n}{N^{2\alpha}} \leq \frac{(4c_s^2+1)\mu n}{N^{\frac{3\alpha}{2}}}$$

该界成立，是因为 $\mu \leq 1$ 且 $N \geq 1$，因此 $\frac{4c_s^2\mu^2 n}{N^{2\alpha}} = \frac{4c_s^2\mu n}{N^{\frac{3\alpha}{2}}} \times \frac{\mu}{N^{\frac{\alpha}{2}}} \leq \frac{4c_s^2\mu n}{N^{\frac{3\alpha}{2}}}$。利用该界，我们得到：

$$
\begin{aligned}
b_0^s &= \frac{2\eta^2 c_s^2 L^3}{n}\frac{(1+\omega_s)^m-1}{\omega_s} = \frac{2\mu^2 n c_s^2 L}{N^{2\alpha}}\frac{(1+\omega_s)^m-1}{\omega_s} \\
&\le \frac{2\mu n c_s^2 L((1+\omega_s)^m-1)}{N^{\frac{\alpha}{2}}(4c_s+1)} \le \frac{2\mu n c_s^2 L((1+\frac{(4c_s^2+1)\mu n}{N^{\frac{3\alpha}{2}}})^{\frac{N^{\frac{3\alpha}{2}}}{\mu n}}-1)}{N^{\frac{\alpha}{2}}(4c_s^2+1)} \\
&\le \frac{2\mu n c_s^2 L(e^{\frac{1}{4c_s^2+1}}-1)}{N^{\frac{\alpha}{2}}(4c_s^2+1)}
\end{aligned}
$$

最后一个不等式成立，是因为当 $x>0$ 时，$(1+\frac{1}{x})^x$ 是关于 x 的单调递增函数。因此，在第三个不等式中有 $(1+\frac{(4c_s^2+1)\mu n}{N^{\frac{3\alpha}{2}}})^{\frac{N^{\frac{3\alpha}{2}}}{\mu n}} \le e^{\frac{1}{4c_s^2+1}}$。并且我们可以得到 ϕ 的下界

$$
\phi = \min_{t,s}\Phi_t^s \ge \min_s(\eta - \frac{b_0^s\eta}{\zeta} - \eta^2 L - 2b_0^s\eta^2) \ge \frac{n\nu}{LN^\alpha}
$$

第一个不等式成立，是因为 b_s^t 是 t的单调递减函数。同时，第二个不等式成立，是因为存在一致常数 ν，使得 $\nu \ge \mu(1-\frac{b_0^s\eta}{\zeta} - L\eta - b_0^s\eta)$。

注 6. 在实践中，$b_0^s \approx 0$，因为 γ和 c_s 都小于 *0.1*，从而得到 $\mu(1-\frac{b_0^s}{\zeta} - L\eta - b_0^s\eta) \approx \mu(1-L\eta)$，且该值通常远大于小批量随机方差缩减梯度法上界中的 ν^*。

我们需要求出 ξ的上界

$$
\begin{aligned}
\xi_s &= \sum_{t=0}^{m-1}(L+2b_{t+1}^s) = mL + 2\sum_{t=0}^{m-1} b_{t+1}^s \\
&= mL + 2\sum_{t=0}^{m-1}\frac{2c_s^2\eta^2L^3}{n}\frac{(1+\omega_s)^{m-t}-1}{\omega_s} \\
&= mL + \frac{2c_s^2\eta^2L^3}{n\omega_s}[\frac{(1+\omega_s)^{m+1}-(1+\omega_s)}{\omega_s} - m] \\
&\le mL + \frac{2c_s^2\eta^2L^3}{n}[\frac{1+\omega_s}{\omega_s^2}(e^{\frac{1}{4c_s^2+1}}-1) - m] \\
&\le mL + \frac{2c_s^2LN^\alpha}{n}(1+\frac{\mu n}{N^{3\alpha/2}})(e^{\frac{1}{4c_s^2+1}}-1) - \frac{2c_s^2\mu^2 nmL}{N^{2\alpha}} \\
&= L[(1-\frac{2c_s^2\mu^2 nL}{N^{2\alpha}})m + \frac{2c_s^2N^\alpha}{n}(1+\frac{\mu n}{N^{3\alpha/2}})(e^{\frac{1}{4c_s^2+1}}-1)]
\end{aligned}
$$

第一个不等式成立的原因已在前文解释；第二个不等式成立，是因为当 $x>0$、$\omega_s = \frac{\mu n}{N^{\frac{3\alpha}{2}}} + \frac{4c_s^2\mu^2 n}{N^{2\alpha}} \le \frac{\mu n}{N^{\frac{3\alpha}{2}}}$和$\eta = \frac{\mu n}{LN^\alpha}$时，$\frac{1+x}{x^2}$ 是 x 的单调递减函数。因此，$\xi_s \le \max_s \xi_s \le \kappa L$，其中 $\kappa \ge \max_s((1-\frac{2c_s^2\mu^2 nL}{N^{2\alpha}})m + \frac{2c_s^2N^\alpha}{n}(1+\frac{\mu n}{N^{3\alpha/2}})(e^{\frac{1}{4c_s^2+1}}-1))$。当 $c_s \approx 0$时，$(1-\frac{2c_s^2\mu^2 nL}{N^{2\alpha}})m + \frac{2c_s^2N^\alpha}{n}(1+\frac{\mu n}{N^{3\alpha/2}})(e^{\frac{1}{4c_s^2+1}}-1) \approx m$。

现在我们得到了 ϕ 的下界和 ξ的上界，将它们代入方程 (10)，可得:

$$\begin{aligned}\mathbb{E}[||g(\boldsymbol{\theta_\pi})||^2] &\leq \frac{l(\boldsymbol{\theta}_0)-l(\boldsymbol{\theta}_*)}{\phi}+\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\frac{2(L+2b_{t+1}^s)(1-c_s)^2\eta^2\sigma^2}{Tn\phi}\\ &\leq \frac{(l(\boldsymbol{\theta}_0)-l(\boldsymbol{\theta}_*))LN^\alpha}{Tn\nu}+\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\frac{2(L+2b_{t+1}^s)\eta^2\sigma^2}{Tn\phi}\\ &\leq \frac{(l(\boldsymbol{\theta}_0)-l(\boldsymbol{\theta}_*))LN^\alpha}{Tn\nu}+\sum_{s=0}^{S-1}(\frac{2\eta^2\sigma^2}{Tn\phi})\sum_{t=0}^{m-1}(L+2b_{t+1}^s)\\ &\leq \frac{(l(\boldsymbol{\theta}_0)-l(\boldsymbol{\theta}_*))LN^\alpha}{Tn\nu}+\frac{2\kappa\mu^2\sigma^2}{N^\alpha\nu m}\end{aligned}$$

备注 7。 在上述理论分析中，我们将 c 视为每个训练轮次中的常数，这仍与我们的实际算法一致，原因如下。

(i) 在我们的算法1中， $\widehat{c}_t^*$ 实际上在每个训练轮次内都是一个固定常数，不同训练轮次的取值可能不同。由于在每次迭代中计算精确的 $\widehat{c}_t^*$ 代价过高，我们会在每个训练轮次开始时计算它，并在接下来的训练轮次中将其用作近似值。

(ii) 关于我们的证明，我们首先展示任意一个训练周期的收敛速率。在这一步中，将 c视为常数与我们的实际算法一致。

(iii) 随后，当我们将单个训练轮次的结果推广到整个训练过程时，会为每个训练轮次中不同的 c 建立一个上界。因此，即使 c 在不同训练轮次之间有所变化，该界仍然适用，从而使我们的理论分析与实际算法保持一致。

A.6. 稀疏训练的实际案例分析

A.6.1. CIFAR-10/100数据集

在实验中，我们在 CIFAR-10 和 CIFAR-100 数据集上应用 SVRG 和 AGENT，采用$\eta = 0.1$、$\gamma = 0.1$、批次大小为 $m = 128$，训练样本总数为 50000。在此参数设置下，**定理1**和**注释4** 中的 ν 和 ν^*分别约为 0.1 和 0.06。 $\frac{2\kappa\mu^2\sigma^2}{N^\alpha\nu m}$ 约为 10^{-5}，这一数值可以忽略不计。因此，我们知道在这种情况下，AGENT 的上界应比 SVRG 更紧，这与 $Figure$ 6中展示的实验结果相符。

A.6.2. SVHN数据集

同时，在 SVHN 数据集上，我们使用以下参数训练模型： $\eta = 0.1$、 $\gamma = 0.1$、批次大小$m = 573$和样本大小 $N = 73257$。 ν、 ν^* 分别等于 0.4 和 0.06， $\frac{2\kappa\mu^2\sigma^2}{N^\alpha\nu m}$ 约为10^{-4}。尽管**定理1**中的第二项更大，但由于此处的 ν 远大于 ν^*，导致**定理1**中的第一项远小于**注释4**中的对应项。因此，与 SVRG 相比，我们仍能得到更严格的界，这也与 $Figure$ 9 中呈现的结果一致。

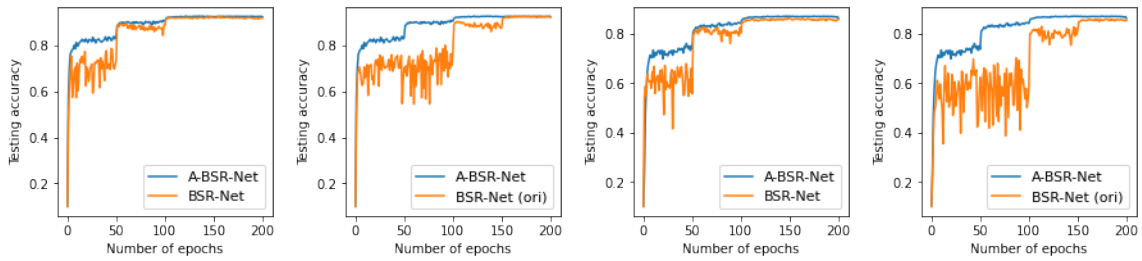
B. 其他实验结果

我们总结了基于 BSR-Net 的 [7], 、基于 RigL 的 [2]，和基于 ITOP 的 [6] 模型的其他实验结果。

B.1. 不同训练轮次下的准确率比较

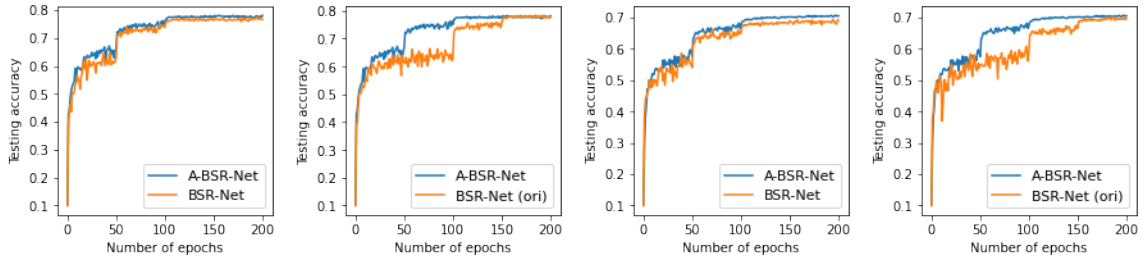
与正文保持一致，我们比较给定训练轮数下的准确率，以同时比较收敛速度和训练稳定性。本节首先展示基于 BSR-Net 的结果。由于我们的方法收敛更快，且无需较长的预热期，

因此，衰减调度器的分界点设为第 50 和第 100 个训练轮次。在正文中，为了进行准确比较，我们也对 BSR-Net 使用这一调度策略。在附录中，我们给出了使用其原始调度策略的结果。BSR-Net 和 BSR-Net (ori) 分别表示使用我们的学习率调度策略和原始调度策略训练得到的结果 [7], 。如图 6、7、8、9、10、11、12 所示，蓝色曲线（A-BSR-Net）始终高于黄色曲线，且比黄色曲线（BSR-Net 和 BSR-Net (ori)）平滑得多，这表明使用我们提出的 A-BSR-Net 时，训练速度更快且更加稳定。



(a) 90% 稀疏度　(b) 90% 稀疏度　(c) 99% 稀疏度　(d) 99% 稀疏度

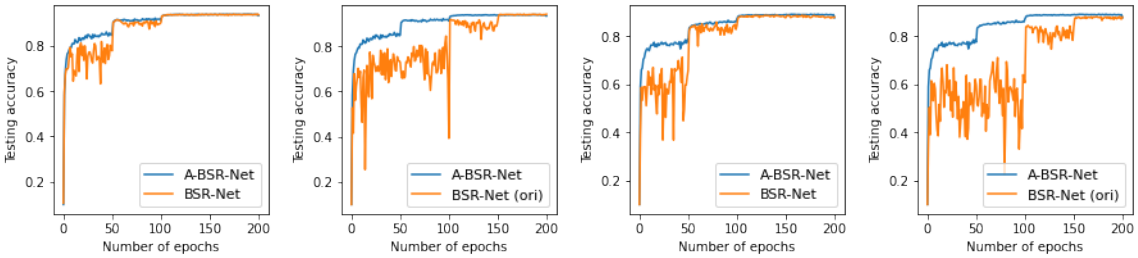
图 6：与 BSR-Net [7] 的比较（给出不同训练轮数下的准确率）。我们评估了使用 VGG-16 在 CIFAR-10 上通过自然训练学习得到的稀疏网络（99% 或 90% 稀疏度）。



(a) 90% 稀疏度　(b) 90% 稀疏度　(c) 99% 稀疏度　(d) 99% 稀疏度

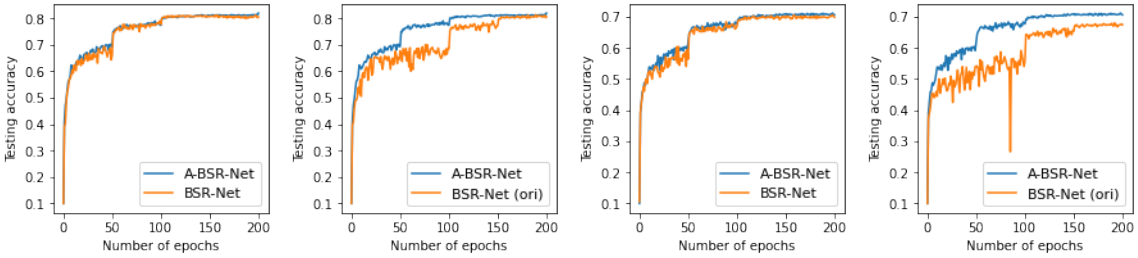
图 7：使用 BSR-Net [7] 进行比较（给定训练轮数时的准确率）。我们在 CIFAR-10 上使用 VGG-16，对通过对抗训练（目标：对抗训练）学习得到的稀疏网络（99% 或 90%）进行了评估。

我们还展示了基于 ITOP 的 ImageNet-2012 结果。如图 13 所示，红色和蓝色曲线分别表示在 80% 和 90% 稀疏度下，AGENT + RigL-ITOP 和 RigL-ITOP 在 ResNet-50 上的结果。对于 80% 稀疏度，红色曲线高于蓝色曲线，表明我们的 AGENT 具有加速效果，尤其是在训练早期阶段。对于 90% 稀疏度，可以看到红色曲线比蓝色曲线更稳定，这表明我们的 AGENT 在大型数据集上具有稳定效果。如果在这种情况下使用 SVRG，我们不仅无法实现稳定训练，还会降低训练速度。相比之下，我们的 AGENT 可以克服 SVRG 的局限性。



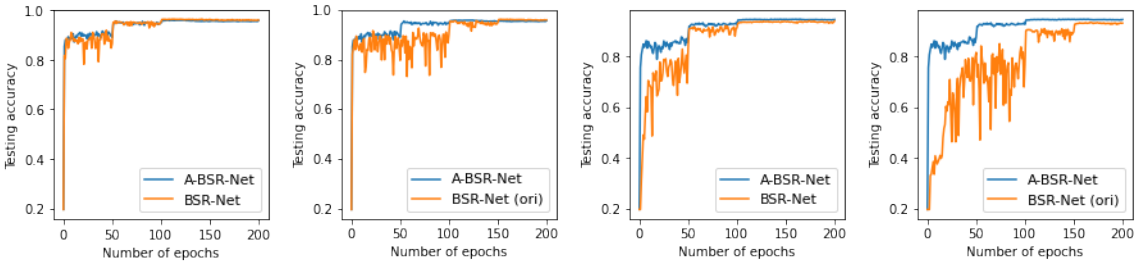
(a) 90% 稀疏度 (b) 90% 稀疏度 (c) 99% 稀疏度 (d) 99% 稀疏度

图8：与 BSR-Net [7] 的比较（给出不同训练轮数下的准确率）。我们评估了使用 Wide-ResNet-28-4 在 CIFAR-10 上通过自然训练学习得到的稀疏网络（99% 或 90%）。



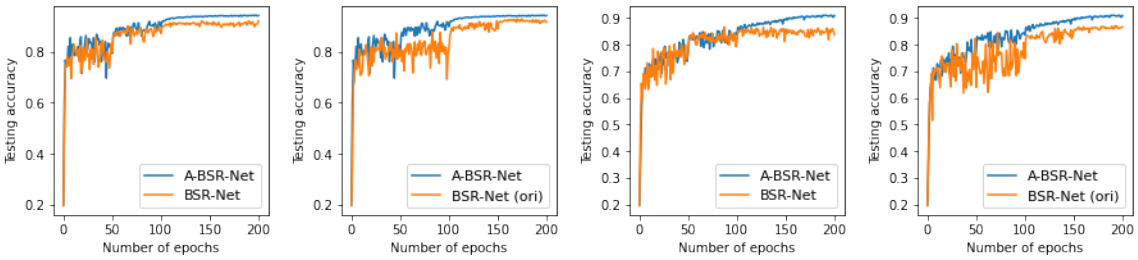
(a) 90% 稀疏度 (b) 90% 稀疏度 (c) 99% 稀疏度 (d) 99% 稀疏度

图9：与 BSR-Net [7] 的比较（给定训练轮数时的准确率）。我们在 CIFAR-10 上使用 Wide-ResNet-28-4 对通过对抗训练（目标：对抗训练）学习得到的稀疏网络（99% 或 90%）进行评估。



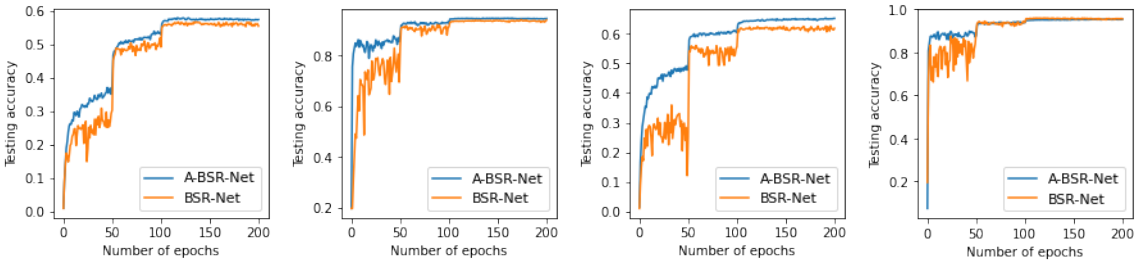
(a) 90% 稀疏度 (b) 90% 稀疏度 (c) 99% 稀疏度 (d) 99% 稀疏度

图 10：与 BSR-Net [7] 的比较（给定训练轮数时的准确率）。我们评估了使用 VGG-16 在 SVHN 上通过自然训练学习得到的稀疏网络（99% 或 90%）。



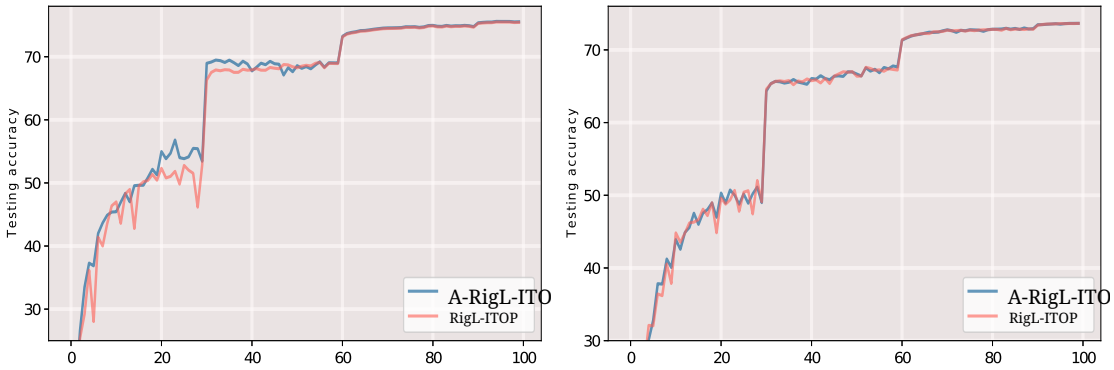
(a) 90% 稀疏度 (b) 90% 稀疏度 (c) 99% 稀疏度 (d) 99% 稀疏度

图 11：与 BSR-Net [7] 的比较（给定训练轮数时的准确率）。我们使用 VGG-16，在 SVHN 上评估通过对抗训练（目标：TRADES）学习得到的稀疏网络（99% 或 90%）。



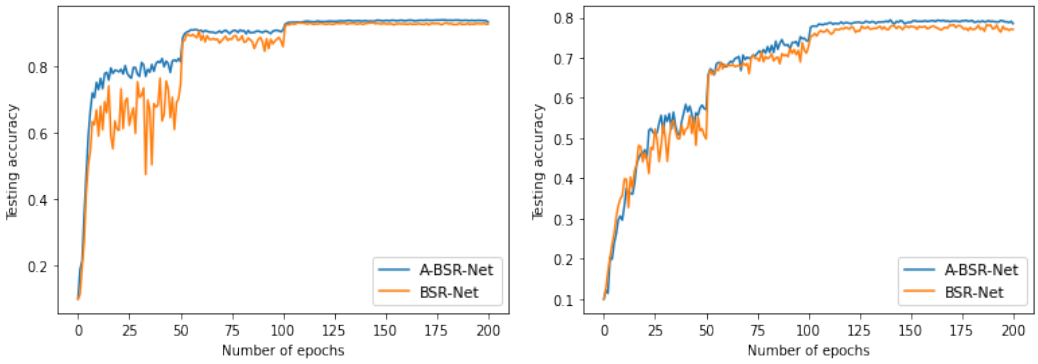
(a) CIFAR-100，VGG-16　(b) SVHN，VGG-16　(c) CIFAR-100，WRN-28-4　(d) SVHN，WRN-28-4

图 12：基于 BSR-Net 的模型 [7] 的训练曲线（给定训练轮数时的准确率）。在标准设置下，在以下数据集和模型上学习稀疏度为 99% 的稀疏网络：(a) 使用 VGG-16 的 CIFAR-100；(b) 使用 VGG-16 的 SVHN；(c) 使用 WRN-28-4 的 CIFAR-100；(d) 使用 WRN-28-4 的 SVHN。



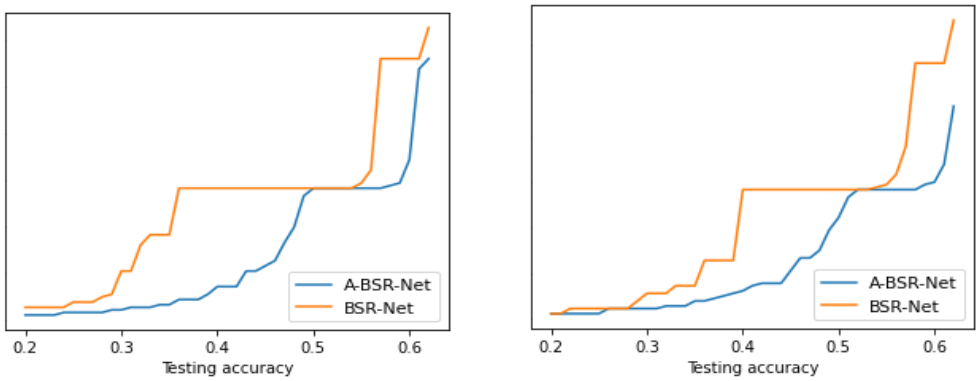
(a) 80% 稀疏度　(b) 90% 稀疏度

图13：ImageNet-2012 上稀疏度为 80% 和 90% 的基于 ITOP 的模型的测试准确率。A-RigL-ITOP（蓝色曲线）比 RigL-ITOP（粉色曲线）收敛得更快。



(a) 标准　(b) 对抗（对抗训练）

图14：基于 BSR-Net 的模型 [7] 的训练曲线（达到给定准确率所需的训练轮次）。在 CIFAR-10 上使用 VGG-16，以标准和对抗设置训练稠密网络。

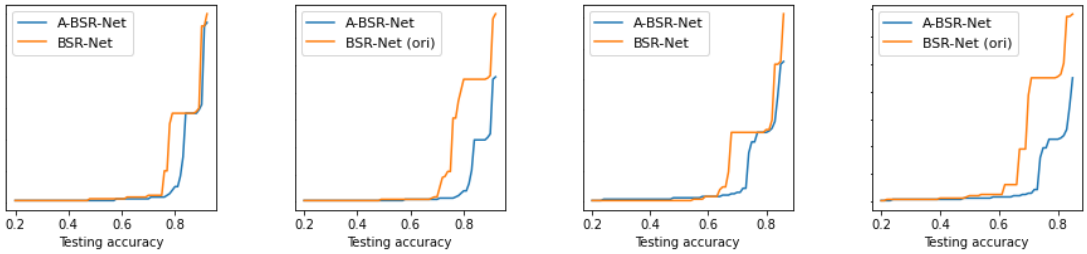


(a) Wide-ResNet-28-4 (b) ResNet-18

图15：比较（达到指定准确率所需的小时数）。我们评估了在 CIFAR-100 上使用自然训练学习的稀疏网络（99%），分别采用 (a) Wide-ResNet-28-4 和 (b) ResNet-18。
在图14中，我们还比较了无稀疏度时的收敛速度。我们展示了基于 BSR-Net 的结果，其中稠密网络使用 VGG-16 在 CIFAR-10 上通过对抗训练（AT）和标准训练学习得到。我们的方法 A-BSR-Net 的蓝色曲线趋于高于 BSR-Net 的黄色曲线，表明加速效果显著。这证明了我们的方法具有广泛的适用性。

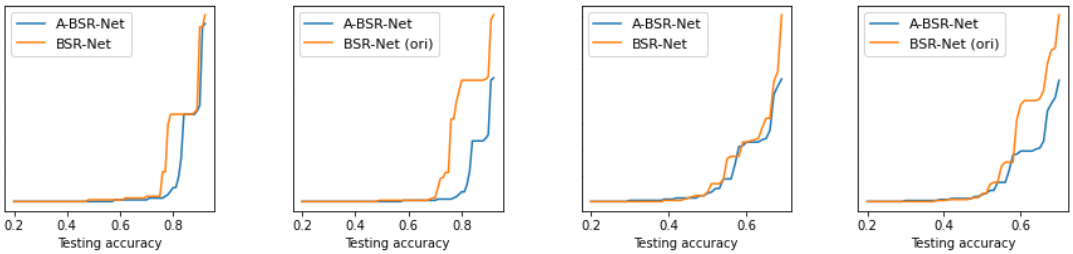
B.2. 训练轮数比较

我们还比较了基于 BSR-Net 的结果达到相同准确率所需的训练轮数。在图15、16、17、18、19、20、21中，蓝色曲线（A-BSR-Net）始终低于黄色曲线（BSR-Net 和 BSR-Net (ori)），表明 A-BSR-Net 收敛更快。



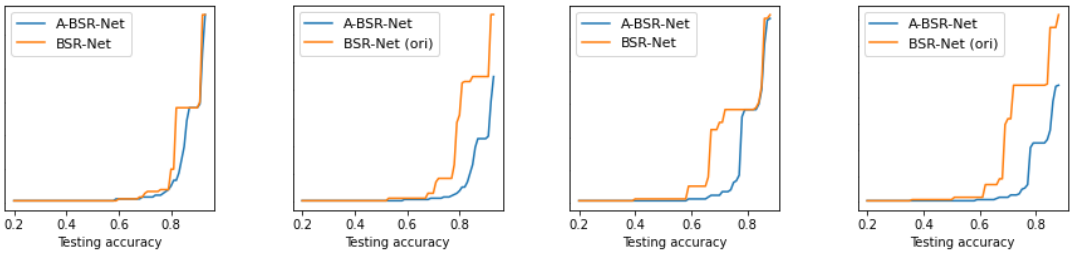
(a) 90% 稀疏度 (b) 90% 稀疏度 (c) 99% 稀疏度 (d) 99% 稀疏度

图16：对比（达到指定准确率所需的小时数）。我们评估了在 CIFAR-10 上使用 VGG-16 通过自然训练学习得到的稀疏网络（99% 或 90%）。



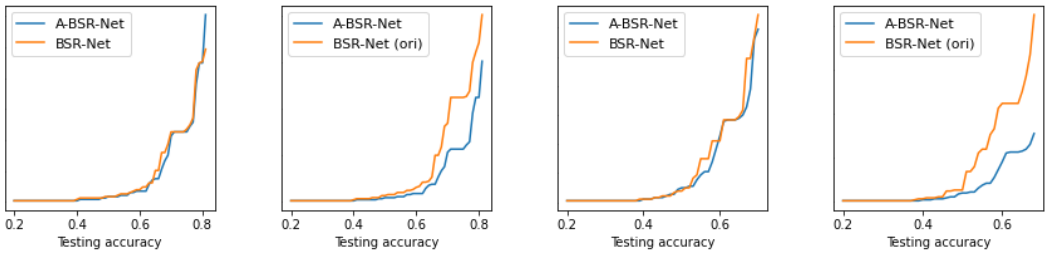
(a) 90% 稀疏度 (b) 90% 稀疏度 (c) 99% 稀疏度 (d) 99% 稀疏度

图17：比较（达到指定准确率所需的小时数）。我们使用 VGG-16，在 CIFAR-10 上评估通过对抗训练（目标：对抗训练）学习的稀疏网络（99% 或 90%）。



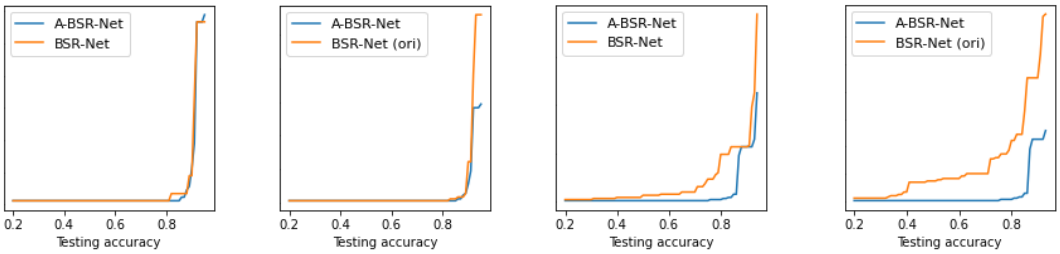
(a) 90% 稀疏度 (b) 90% 稀疏度 (c) 99% 稀疏度 (d) 99% 稀疏度

图18：比较（达到指定准确率所需的小时数）。我们评估了使用 Wide-ResNet-28-4 在 CIF AR-10 上通过自然训练学习得到的稀疏网络（99% 或 90%）。



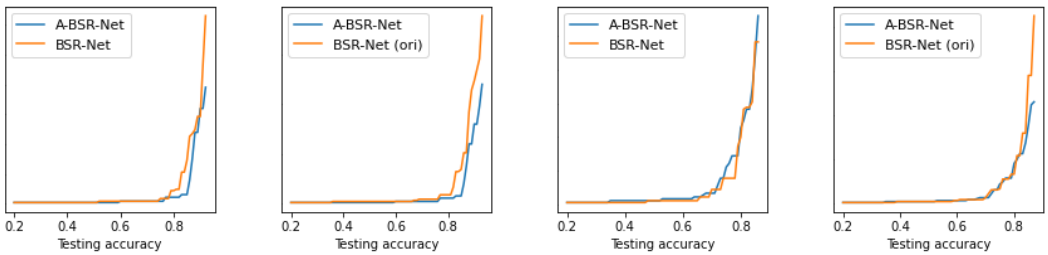
(a) 90% 稀疏度 (b) 90% 稀疏度 (c) 99% 稀疏度 (d) 99% 稀疏度

图19：比较（达到指定准确率所需的小时数）。我们使用 Wide-ResNet-28-4，在 CIFAR-10 上评估通过对抗训练（目标：对抗训练）学习得到的稀疏网络（99% 或 90% 稀疏度）。



(a) 90% 稀疏度 (b) 90% 稀疏度 (c) 99% 稀疏度 (d) 99% 稀疏度

图20：对比（达到指定准确率所需的小时数）。我们评估了在 SVHN 上使用 VGG-16 通过自然训练得到的稀疏网络（99% 或 90%）。



(a) 90% 稀疏度 (b) 90% 稀疏度 (c) 99% 稀疏度 （d）99% 稀疏度

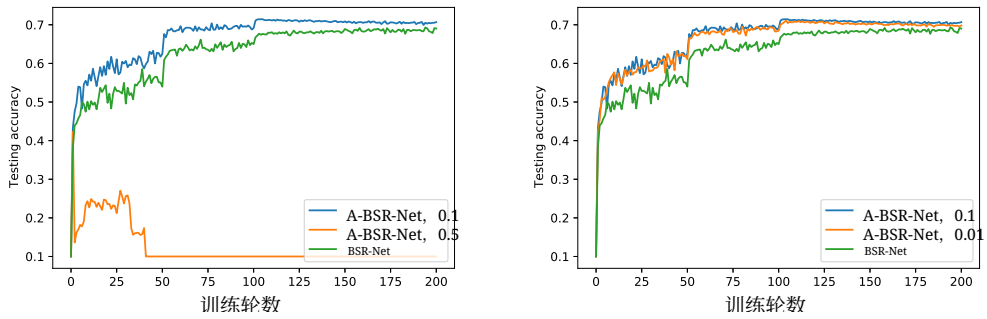
图21：比较（达到给定准确率所需的小时数）。我们使用 VGG-16，在 SVHN 上评估通过对抗训练（目标：TRADES）学习得到的稀疏网络（99% 或 90%）。

B.3. 缩放参数设置

缩放参数 γ 的选择对加速至关重要，可视为一个超参数调优过程。我们尝试了 γ 的不同取值，发现将其设为 $\gamma = 0.1$ 是有效加速训练的一个不错选择。若根据不同设置下的梯度相关性调整 γ 的取值，则有可能获得比报告结果更快的收敛速率。我们首先展示使用 VGG-16 在 CIFAR-10 上通过对抗训练（目标：AT）学习的稀疏网络（99%）的结果。稀疏训练方法为 BSR-Net。

$\gamma = 0.1$ 与 $\gamma = 0.5$：如图22 (a)所示，我们比较了 A-BSR-Net ($\gamma = 0.1$)、A-BSR-Net ($\gamma = 0.5$) 和 BSR-Net 的训练曲线（不同训练轮次下的测试准确率）。A-BSR-Net ($\gamma = 0.5$)的黄色曲线在训练约40轮后崩溃，表明模型发散。原因是，如果将 γ 设得接近1（例如0.5），就无法完全避免方差增大。方差增大会导致性能下降，这与论文第5.4节中的“Noγ”类似。

$\gamma = 0.1$ 与 $\gamma = 0.01$：如图22 (b) 所示，我们比较了 A-BSR-Net ($\gamma = 0.1$)、A-BSR-Net ($\gamma = 0.01$) 和 BSR-Net 的训练曲线（不同训练轮次下的测试准确率）。A-BSR-Net ($\gamma = 0.01$) 的黄色曲线低于 A-BSR-Net ($\gamma = 0.1$) 的蓝色曲线，表明其收敛速度较慢。原因在于，如果 γ 设置得较小（例如0.01），旧梯度的权重就会较小。因此，旧梯度对模型更新方向的影响有限，这往往会减慢收敛速度，有时还会导致训练不稳定性加剧。



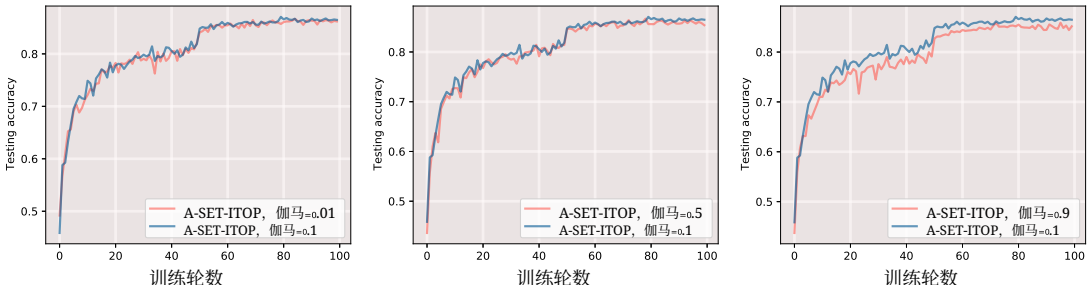
(a) 缩放参数 = 0.1 或 0.5　　(b) 缩放参数 = 0.1 或 0.01

图22：基于 BSR-Net 的模型 [7] 在不同缩放参数下的比较（给定训练轮数时的验证准确率）。我们使用 VGG-16，在 CIFAR-10 上通过对抗训练（目标：AT）训练稀疏网络（99%）。(a) 缩放参数 = 0.1 或 0.5；(b) 缩放参数 = 0.1 或 0.01。

我们还展示了基于使用 VGG-C 在 CIFAR-10 上通过标准训练学习得到的稀疏网络（99%）的结果。稀疏训练方法为 SET-ITOP。如图 23 所示，设置 $\gamma = 0.01$，0。5 的结果与 $\gamma = 0.1$ 相似，而设置 $\gamma = 0.9$ 的结果则不如 $\gamma = 0.1$。这可能是因为对于相对较低的梯度相关性而言，0.9 太大了。

B.4. 其他方差缩减方法比较

我们还补充了更多结果，对比 AGENT 与随机方差缩减梯度（SVRG） [18, 19, 50], ——一种在非稀疏情况下广泛使用的方差缩减方法，以展示现有方法的局限性。



(a) $\gamma = 0.1$ 与 $\gamma = 0.01$对比　(b) $\gamma = 0.1$ 与 $\gamma = 0.5$对比　(c) $\gamma = 0.1$ 与 $\gamma = 0.9$对比

图 23：不同缩放因子 γ 的对比（给定训练轮数时的验证准确率）。我们评估了在 CIFAR-10 上使用 VGG-C 通过标准训练学习得到的稀疏网络（99%），其中设置为 (a) $\gamma = 0.1$ 与 $\gamma = 0.01$对比，(b) $\gamma = 0.1$ 与 $\gamma = 0.5$对比，以及 (c) $\gamma = 0.1$ 与 $\gamma = 0.9$对比。

B.4.1. 基于 BSR-Net 的结果

所展示的结果基于使用 VGG-16 在 CIFAR-10 上通过对抗训练（目标：对抗训练）学习得到的稀疏网络（99%）。如图 24 所示，我们展示了 A-BSR-Net、BSR-Net 以及使用 SVRG 的 BSR-Net 的训练曲线（不同训练轮次下的测试准确率）。使用 SVRG 的 BSR-Net 的黄色曲线升至约 0.4，随后迅速下降至约 0.1 的较低值，表明模型发散。这说明 SVRG 不适用于稀疏训练。至于我们提出的 A-BSR-Net 的蓝色曲线，它始终高于 BSR-Net 的绿色曲线，表明实现了有效加速。

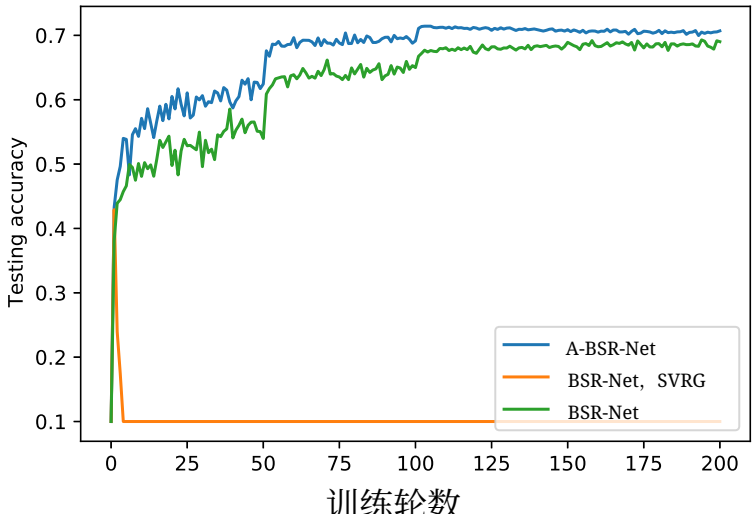


图 24：基于 BSR-Net 的模型 [7]中采用不同方差缩减方法的比较（给定训练轮数的测试准确率）。我们评估了在 CIFAR-10 上使用 VGG-16、通过对抗训练（目标：对抗训练）学习得到的稀疏网络（99%）。

B.4.2. 基于 RigL 的结果

所展示的结果基于使用 ResNet-50 在 CIFAR-100 上通过标准训练学习得到的稀疏网络（90%）。如图 25 所示，我们展示了 A-RigL、RigL 以及使用 SVRG 的 RigL 的训练曲线（不同训练轮次下的测试准确率）。使用 SVRG 的 RigL 的黄色曲线始终低于另外两条曲线，表明其模型收敛速度较慢。这说明

SVRG 不适用于稀疏训练。至于我们提出的 A-RigL 的蓝色曲线，它始终位于 RigL 的绿色曲线之上，表明加速是成功的。

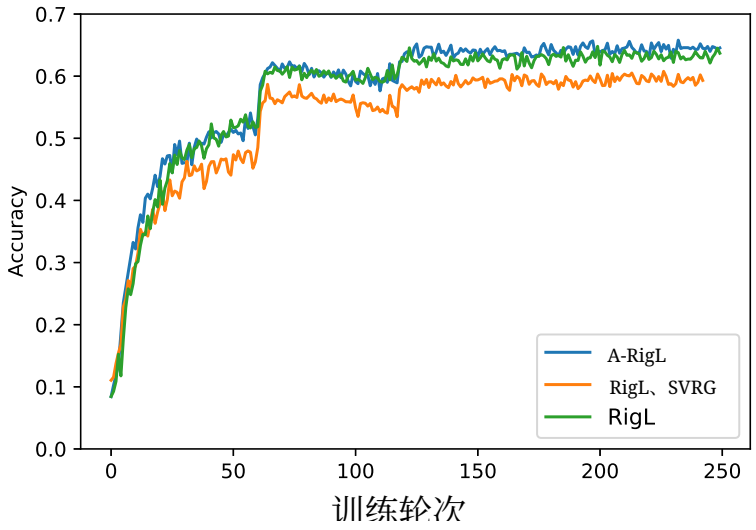


图 25：在基于 RigL 的模型中，采用不同方差缩减方法进行比较（给定训练轮数时的测试准确率）[2]。我们评估了使用 ResNet-50 在 CIFAR-100 上通过标准训练学习得到的稀疏网络（90%）。

B.5. 最终准确率比较

此外，我们还补充了更多结果，比较了在 CIFAR-10/100 上基于 RigL 的结果经过充分训练后的最终准确率，结果见表 7。在几乎所有情况下，我们的方法 A-RigL 往往表现最佳。这表明，我们的 AGENT 能够加速稀疏训练，同时保持甚至提高准确率。

表 7：基于 RigL 的模型在 0%（稠密）、90% 和 99% 稀疏度下的最终准确率（%）。AGENT +RigL（A-RigL）相比 RigL 保持了准确率，甚至有所提升。

		稠密	90%	99%
CIFAR-10	A-RigL	**95.2 (0.24)**	**95.0 (0.21)**	**93.1 (0.25)**
	RigL	95.0（0.26）	94.2（0.22）	92.5（0.33）
CIFAR-100	A-RigL	72.9 (0.19)	**72.1(0.20)**	**66.4 (0.14)**
	RigL	**73.1 (0.17)**	71.6(0.26)	66.0 (0.19)

我们还提供了更多基于 BSR-Net 的结果，用于比较最终准确率。除了正文中的 BSR-Net 和 A-BSR-Net 外，我们还在附录中加入了 HYDRA，它也是一种先进的稀疏对抗训练流程。我们使用 VGG-16 和 WideResNet-28-4（WRN-28-4）在 SVHN 上进行训练。BSR-Net 和 HYDRA 的最终结果取自[7]，并采用其原始学习率调度策略。如表8所示，令人欣喜的是，在使用干净测试样本时，我们的方法在所有情况下往往都表现最佳。在鲁棒性方面，我们的 A-BSR-Net 在大多数情况下优于 HYDRA，但与 BSR-Net 相比，性能有所下降。

表8：BSR-Net [7] 与 HYDRA [13]的比较。在 SVHN 上使用 VGG-16 和 WideResNet-28-4，通过鲁棒训练目标（TRADES）学习稀疏网络。完整训练（200 个训练轮次）后进行评估，结果以干净/鲁棒准确率（%）表示。鲁棒准确率通过 PGD^{50} 评估，重启次数为 10 $\epsilon = 8/255$。

		BSR-Net	HYDRA	Ours
90% 稀疏度	VGG-16	89.4/**53.7**	89.2/52.8	**94.4**/51.9
	WRN-28-4	92.8/**55.6**	94.4/43.9	**95.5**/46.2
99% 稀疏度	VGG-16	86.4/**48.7**	84.4/47.8	**90.9**/47.9
	WRN-28-4	89.5/**52.7**	88.9/39.1	**92.2**/51.1

B.6. 梯度变化速度与稀疏度

在稀疏训练中，权重发生微小变化时，梯度变化比稠密训练更快。这一现象可以表示为当前梯度与先前梯度之间的低相关性，从而使现有的方差缩减方法失效。

直观理解：考虑计算当前梯度和先前梯度时所对应的权重，在稀疏训练中，当当前梯度和先前梯度的掩码不同时，有三种情况需要讨论。第一，如果当前权重被剪枝，我们无需考虑它们的相关性，因为不需要使用对应的先前权重来更新当前权重。第二，如果当前权重未被剪枝，但先前权重被剪枝，那么先前权重为零，且两个权重之间的差异相对较大，导致相关性较低。第三，如果当前权重和先前权重均未被剪枝，被剪枝的权重仍可能发生显著变化，导致当前模型与先前模型之间出现较大差异。因此，这些权重的当前梯度与先前梯度之间的相关性相对较小。因此，在稀疏训练中直接设置 $c = 1$并不是好主意，甚至可能增加方差并减慢收敛速度。

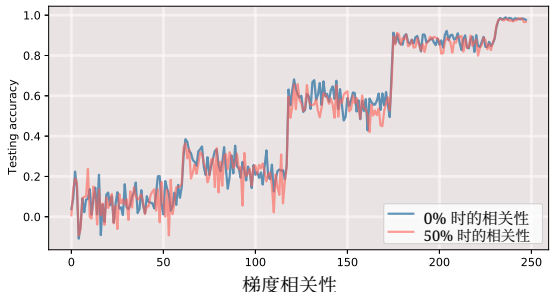
当前梯度与前一梯度的掩码相同时，相关性仍往往较弱。众所周知，$c_t^* = \mathrm{Cov}(g_{\text{new}}, g_{\text{old}})\ \mathrm{Var}(g_{\text{old}})$。即使 $\mathrm{Cov}(g_{\text{new}}, g_{\text{old}})$ 没有降低，稀疏训练中的方差 $\mathrm{Var}(g_{\text{old}})$ 也会增大，从而导致 c_t^* 降低。

除了上述分析之外，我们还进行了一些实验，以证明梯度变化会随着稀疏度的增加而加快。为衡量变化速率，我们的实验描述如下。

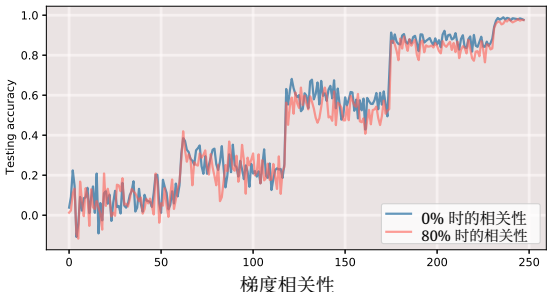
训练过程中的相关性：我们还分析了使用 RigL 在 CIFAR-100 上对稀疏 ResNet-50 进行标准训练时的梯度相关性。结果汇总于图 26，其中蓝色曲线表示稠密训练（0% 稀疏度）时的梯度相关性，粉色曲线表示稀疏训练（即 RigL）时的相关性。如图所示，稠密训练与稀疏训练之间的相关性较为接近。对于 80% 的稀疏度，与稠密训练相比，稀疏训练的相关性往往较低，尤其是在训练后期阶段。对于 90% 和 95% 的稀疏度，稀疏训练的相关性也低于稠密训练，且差异随稀疏度增加而扩大。

完全训练模型的相关性：我们从 ResNet-50 在 CIFAR-100 上使用 RigL 和 SET、稀疏度分别为 0%、50%、80%、90% 和 95% 时的完全训练检查点开始。我们计算并存储每个权重在所有训练数据上的梯度。然后，我们对所有权重添加高斯扰动（std = 0.015），并再次计算梯度。最后，我们计算新扰动权重的梯度与旧原始权重的梯度之间的相关性。

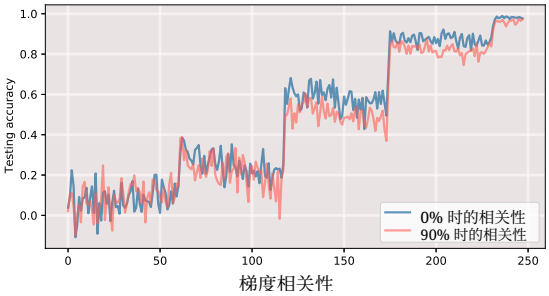
众所周知，旧权重与新权重之间总是存在差异。如果在权重中加入少量噪声后，梯度发生了很大变化，那么新旧梯度的相关性往往会较低。如果加入少量噪声后，梯度变化不大，那么



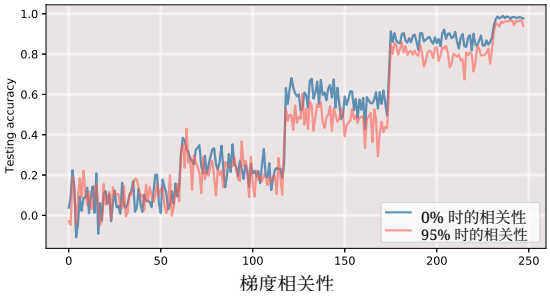
(a) 稀疏度=0% 与稀疏度=50%



(b) 稀疏度=0% 与稀疏度=80%



(c) 稀疏度=0% 与稀疏度=90%



(d) 稀疏度=0% 与稀疏度=95%

图26：稠密训练与 SET-ITOP 中的梯度相关性。我们使用 VGG-C，在 CIFAR-10 上通过标准训练学习得到的稀疏网络进行评估。我们比较了稠密训练（稀疏度=0%）与不同稀疏度下的稀疏训练之间的相关性：(a) 50%、(b) 80%、(c) 90% 和 (d) 95%。

旧梯度与新梯度的相关性就会较高。因此，我们在权重中加入高斯噪声，以模拟新旧梯度之间的差异。如表9所示，相关性随稀疏度的增加而降低，这表明稀疏训练中的相关性较弱，也支持了我们的论点。

表9：由 RigL 和 SET 在不同稀疏度（包括 0%、50%、80%、90%、95%、99%）下生成的 CIFAR-100 上 ResNet-50 的新扰动权重的梯度与旧原始权重之间的相关性。

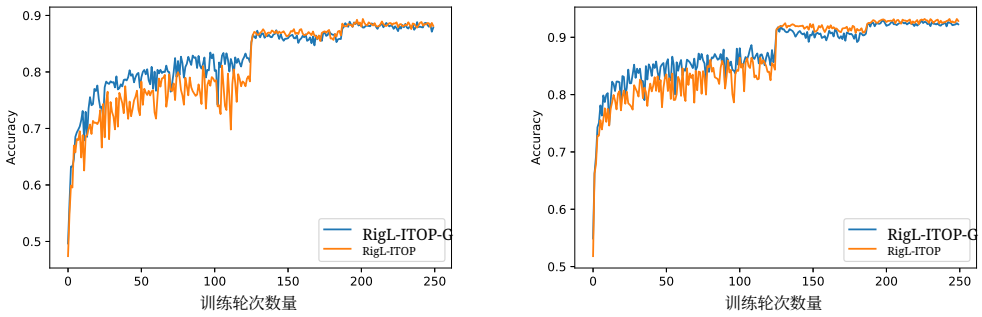
稀疏度	0%	50%	80%	90%	95%
ResNet-50，CIFAR-100（RigL）	0.6005	0.4564	0.3217	0.1886	0.1590
ResNet-50，CIFAR-100（SET）	0.6005	0.4535	0.2528	0.1763	0.1195

B.7. RigL 的变体

RigL 是最受欢迎的动态稀疏训练流程之一，它使用权重幅值进行剪枝，并使用梯度幅值进行增长。我们的方法利用旧的存储梯度（其噪声通常较小）自适应地更新新的批次梯度。因此，新批次梯度的方差得以降低，从而加快收敛。目前，我们仅在权重更新中使用方差校正后的梯度。一个自然的问题是，如果我们也在 RigL 中使用这种方差校正梯度进行权重增长，效果会如何。

我们在 CIFAR-10 上训练的基于 RigL 的模型中进行了一些实验。如图 27 所示，蓝色曲线（RigL-ITOP-G）和黄色曲线（RigL-ITOP）分别对应使用和不使用方差校正梯度时的权重增长。可以看到，在训练初期，蓝色曲线高于黄色曲线。但在第一次学习率衰减后，蓝色曲线趋于低于黄色曲线。这表明，使用方差校正梯度进行权重增长

在训练初期有助于模型更快地提高准确率。不过，这可能会导致训练后期的准确率略有下降。这可能是因为，随着模型接近最优点，梯度中的某些方差有助于模型更好地探索局部区域，并找到更好的掩码。



(a) VGG-C (b) ResNet-34

图 27：比较采用（RigL-ITOP-G）和未采用（RigL-ITOP）方差校正梯度 [6]的权重增长效果（给定训练轮数时的测试准确率）。我们评估使用标准训练在 CIFAR-10 上训练得到的稀疏网络（99%），分别采用 (a) VGG-C 和 (b) ResNet-34。

B.8. 与降低学习率的比较

为展示缩放参数 γ 的设计，我们将 AGENT 与“Reduce LR”进行比较：从 AGENT 中移除缩放参数 γ，并将学习率设为原来的 0.1 倍。如表 10 所示，降低学习率在训练早期可以达到相近的收敛速率。然而，这会减缓训练后期的进程，并导致次优最终准确率。原因在于，这种做法同时降低了信号和噪声，因此既未改善信噪比，也未加快稀疏训练。

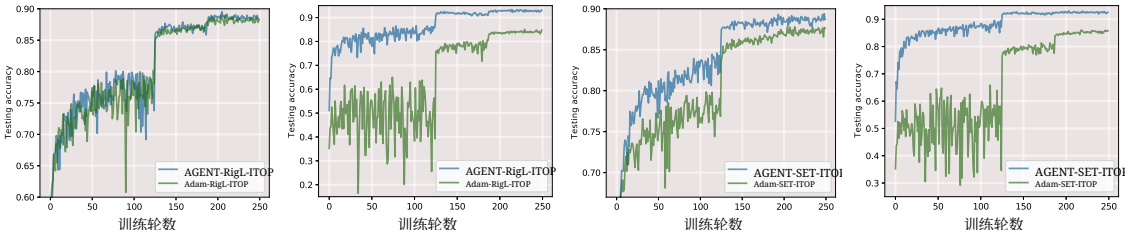
γ 的动机是避免因近似 c_t 时产生的误差而引入较大的方差，以及对抗训练导致的偏差。真实的相关性取决于许多因素，例如数据集、架构和稀疏度。在某些情况下，它可能大于或小于 10%。至于γ 的取值，它是一个超参数，我们可以针对不同的设置选择不同的值。在我们的情况下，为简单起见，我们对所有设置都选择 $\gamma = 0.1$，并发现其效果良好且能够加速收敛。如果根据不同设置对应的相关性调整 γ 的取值，则有可能获得更快的收敛速率。

表10：AGENT（我们的方法）和“Reduce LR”的基于 SET-ITOP 的模型的测试准确率（%）。稀疏 VGG-C 和 ResNet-34 在标准设置下进行学习。

训练轮次	20	80	130	180	240
Reduce LR（VGG-C，SET-ITOP）	**76.5**	81.3	84.6	85.5	85.5
AGENT（VGG-C，SET-ITOP）	76.1	**81.5**	**87.6**	**87.1**	**88.6**
Reduce LR（ResNet-34，SET-ITOP）	81.4	**85.9**	89.3	89.5	89.8
AGENT（ResNet-34，SET-ITOP）	**83.0**	85.6	**92.0**	**92.3**	**92.5**

B.9. 与基于动量的方法的比较

在某种程度上，我们的 AGENT 采用了与基于动量的方法相似的思路 [53,54]，，即利用旧梯度来改善当前批次梯度。基于动量的方法在稠密设置下效果良好。然而，由于稀疏性约束，该方法仍面临优化困难。这是因为它没有考虑稀疏训练和对抗训练的特点，例如当前梯度与



(a) VGG-C(RigL) (b) ResNet-34(RigL) (c) VGG-C(SET) (d) ResNet-34(SET)

图28：CIFAR-10 上稀疏度为 99% 时基于 ITOP 的模型的测试准确率。基于 AGENT 的训练（蓝色曲线）比基于 Adam 的训练（粉色曲线）收敛更快。

先前梯度之间的相关性降低，以及梯度估计器可能存在偏差；此外，它也无法在旧信息与新信息之间实现自适应平衡。当相关性较低时，基于动量的方法仍可能纳入过多旧信息，从而增大梯度方差或偏差。相比之下，我们的 AGENT 专为稀疏训练和对抗训练而设计，能够更精细地自适应控制从旧信息中汲取多少内容来帮助新信息。

例如，在我们的基线随机梯度下降（SGD）中，按照原始代码库的做法，我们也在优化器中加入了动量。然而，如图2中的粉色曲线所示，它仍存在训练不稳定性和收敛问题。原因在于，它们没有考虑稀疏训练和对抗训练的特点，也无法在旧信息和新信息之间实现自适应平衡。

我们的方法 AGENT 专为稀疏训练和对抗训练而设计，能够更精细地控制从旧信息中获取多少信息来助力新信息。在第 6.4 节中，我们通过消融研究展示这种细粒度自适应平衡的重要性。在“固定 c_t”中，我们设置 $c_t = 0.1$ 并测试没有自适应控制时的收敛速率。我们发现，在几乎所有情况下，自适应平衡（我们的方法）都优于“固定 c_t”，尤其是在对抗训练中。对于标准训练，“固定 c_t”的收敛速率与我们的方法相近，而我们的方法往往能取得更好的最终分数。

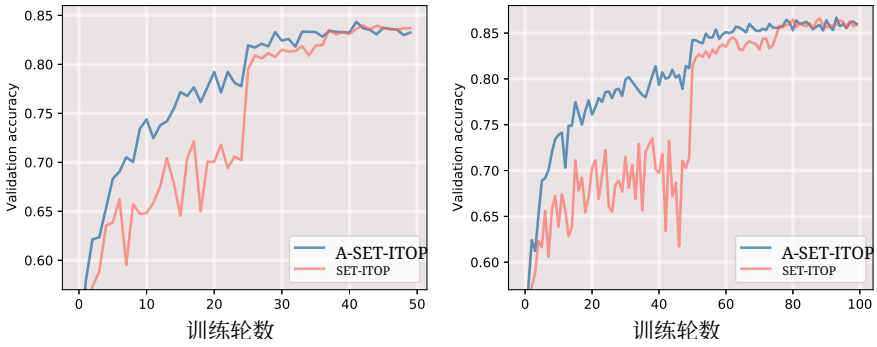
B.10. 与其他自适应梯度方法的比较

我们还将 AGENT 与其他自适应梯度方法进行了比较，并以 Adam [55]为例。如图28所示，AGENT-RigL-ITOP 和 AGENT-SET-ITOP（蓝色曲线）通常高于 Adam-RigL-ITOP 和 Adam-SET-ITOP（绿色曲线），这表明与 Adam 相比，我们的 AGENT 收敛更快。这证明了在稀疏训练中利用相关性来平衡新旧信息的重要性。

B.11. 不同的训练轮数

本节展示，我们的方法在不同训练预算（即训练轮数）下均可实现加速，而不是相较于基线方法仅在早期表现更佳的伪命题。为此，我们补充了不同总训练轮数下的实验，并相应调整学习率调度器，以确保模型收敛。

以 SET-ITOP 为例。在主论文中，我们遵循基线论文的设置：训练轮次为 250，学习率调度器采用阶梯式学习率，并在第 125 轮（即 0.5×250）和第 187 轮（即 0.75×250）进行衰减。为了减少训练轮次并确保收敛，我们将训练轮次设为 50 和 100，并分别将衰减点设为 $\{25, 37\}$ 和 $\{50, 75\}$。如图 29 所示，蓝色曲线（我们的 A-SET-ITOP）通常高于粉色曲线（SET-ITOP），这表明我们的 AGENT 实现了加速。

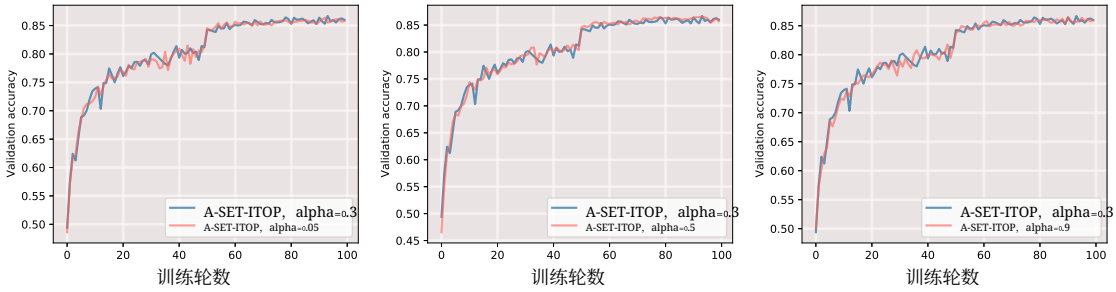


(a) 50 个训练轮次　　(b) 100 个训练轮次

图29：A-SET-ITOP 与 SET-ITOP 的比较（给定训练轮数时的验证准确率）。我们评估了在 CIFAR-10 上使用 VGG-C 通过标准训练学习得到的稀疏网络（99%），其中 (a) 训练 50 轮，(b) 训练 100 轮。

B.12. 平滑因子调优

对于平滑因子 α，我们遵循 Deng 等人 [52] 的默认值，将其设为 0.3。我们补充了一些实验，以测试 α的影响。我们进一步比较了不同平滑因子 α下的验证准确率。如图 30 所示，将 $\alpha = 0.05$，0.5，0.9 设为相应值时，结果与 $\alpha = 0.3$相似。因此，我们的方法对 α的选择并不敏感，可以采用默认值 0.3。

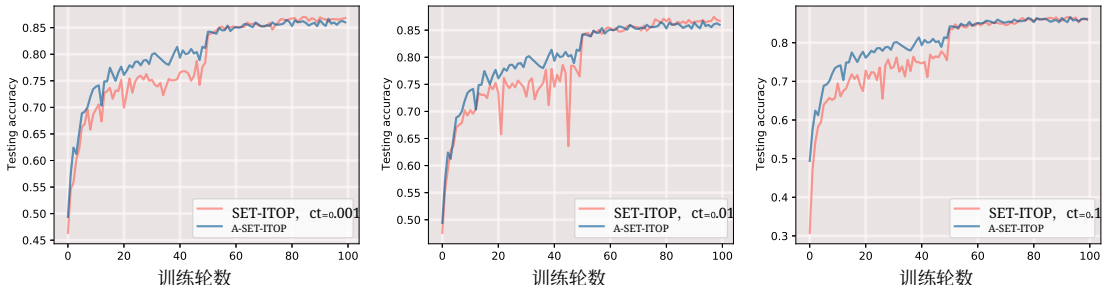


(a) $\alpha = 0.3$与 $\alpha = 0.05$对比　(b) $\alpha = 0.3$ 与 $\alpha = 0.5$对比　(c) $\alpha = 0.3$与 $\alpha = 0.9$对比

图 30：不同平滑因子 α 的比较（给出随训练轮数变化的验证准确率）。我们使用 VGG-C，在 CIFAR-10 上通过标准训练学习稀疏网络（99%），并评估了 (a) $\alpha = 0.3$ 与 $\alpha = 0.05$对比、(b) $\alpha = 0.3$ 与 $\alpha = 0.5$对比，以及 (c) $\alpha = 0.3$ 与$\alpha = 0.9$对比。

B.13. 固定 c_t 调优

我们加入了一个更贴近实际的基线“固定 c_t”，并进行了良好的超参数调优，以表明自适应重加权至关重要。“固定 c_t”指的是在训练期间固定权重 $c_t = 0.1$，这一点已在第 6.6 节的消融研究中提及。具体而言，我们进一步考察了“固定 c_t”中的不同c_t，并将其验证准确率与我们的 A-SET-ITOP 进行比较。如图31所示，当 c_t 固定为 0.001、0.001 和 0.1 时，粉色曲线（“固定 c_t”）在早期阶段低于蓝色曲线（A-SET-ITOP），这表明“固定 c_t”的早期收敛较慢。当 c_t固定为 0.5、0.8 和 1.0 时，粉色曲线整体（“固定 c_t”）均低于蓝色曲线（A-SET-ITOP），这意味着“固定 c_t”的收敛较慢。



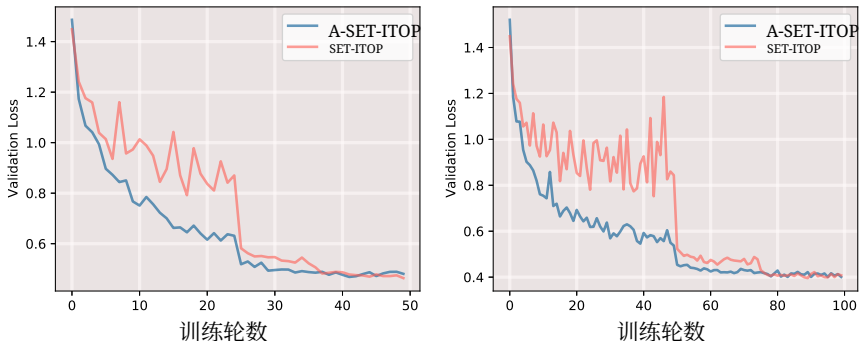
(a) $c_t = 0.001$ (b) $c_t = 0.01$ (c) $c_t = 0.1$

(d) $c_t = 0.5$ (e) $c_t = 0.8$ (f) $c_t = 1.0$

图31：在“固定 c_t”中，不同 c_t 的比较（给定训练轮数的验证准确率）。我们在 CIFAR-10 上使用 VGG-C，通过标准训练学习稀疏网络（99%），并将 A-SET-ITOP 与 (a) $c_t = 0.001$、(b) $c_t = 0.01$、(c) $c_t = 0.1$、(d) $c_t = 0.5$、(e) $c_t = 0.8$、(f) $c_t = 1.0$ 进行比较

B.14. 损失值比较

除了准确率之外，我们还比较了损失，以展示加速效果。如图32所示，我们的 A-SET-ITOP 对应的蓝色曲线通常低于 SET-ITOP 对应的粉色曲线，这表明加速效果显著。

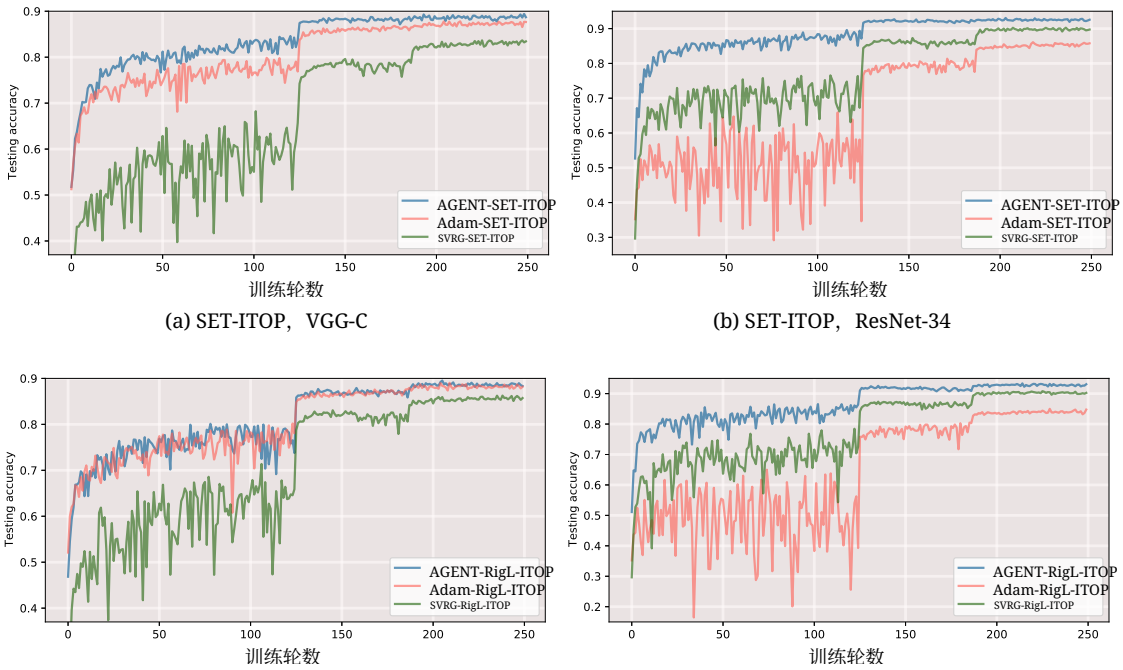


(a) 50 个训练轮次 (b) 100 个训练轮次

图32：A-SET-ITOP 与 SET-ITOP 的比较（给定训练轮数的验证损失）。我们在 CIFAR-10 上使用 VGG-C，通过标准训练学习稀疏网络（99%），分别在 (a) 50 个训练轮次和 (b) 100 个训练轮次下进行评估。

B.15. 更多基线比较

我们补充了更多将 ADAM 与 SVGR 进行对比的结果。如图3所示，蓝色曲线、粉色曲线和绿色曲线分别代表我们的 AGENT、Adam 和 SVRG。我们的 AGENT 的蓝色曲线通常高于粉色曲线和绿色曲线，表明与另外两种方法相比，使用我们的 AGENT 能够更快收敛。



(a) SET-ITOP，VGG-C

(b) SET-ITOP，ResNet-34

(c) RigL-ITOP，VGG-C

(d) RigL-ITOP，ResNet-34

图33：我们比较了 AGENT、Adam 和 SVRG。在 CIFAR-10 上评估通过标准训练得到的稀疏网络。(a) SET-ITOP，VGG-C；(b) SET-ITOP，ResNet-34；(c) RigL-ITOP，VGG-C；(d) RigL-ITOP，ResNet-34。

B.16. 更多训练时间比较

我们考察了我们的方法与基线方法的训练时间。对于基于 ITOP 的结果，我们的 A-SET-ITOP 与 SET-ITOP 之间的训练时间比为 5:3，而我们的 A-RigL-ITOP 与 RigL-ITOP 之间的训练时间比为 2:1。对于基于 BSR-Net 的结果，我们的 A-BSR-Net 与 BSR-Net 之间的训练时间比为 5:4。尽管目前我们的训练时间并不比基线方法更具优势，但可以通过以下方式轻松缩短训练时间。

- 我们可以在稀疏训练中使用稀疏梯度，从而有效降低稀疏训练中的反向传播开销，并且可以轻松应用于我们的方法 [51]。
- 我们可以使用并行计算。由于针对旧模型参数的额外前向传播和反向传播完全可以并行执行，因此可以将其视为将小批量大小加倍 [49]。
- 我们可以借鉴 SAGA 的思路，为每个样本存储梯度。这样就不需要额外的前向传播和反向传播步骤，从而节省实际运行时间 [40]。

B.17. 梯度范数比较

较大的梯度范数对稀疏训练很重要 [65]。我们进行了实验，发现 AGENT 能够提升梯度范数。具体而言，我们在 CIFAR-10 上训练了稀疏度为 99% 的 VGG-C 和 ResNet-34。我们比较了 AGENT + RigL-ITOP（A-RigL-ITOP）与 RigL-ITOP，以及 AGENT+ SET-ITOP（A-SET-ITOP）与 SET-ITOP。对于梯度范数，我们计算了每个训练阶段的平均梯度范数，即第 1 至第 50 训练轮次、第 51 至第 100 训练轮次、第 101 至第 150 训练轮次、第 151 至第 200 训练轮次，以及第 201 至第 250 训练轮次。结果汇总如下表。AGENT 能够略微提升梯度范数，而这对稀疏训练很重要。

表 11：AGENT（我们的方法）和稀疏训练基线方法中，基于 RigL-ITOP 和 SET-ITOP 的模型的梯度范数。在标准设置下学习 99% 稀疏的 VGG-C。

	第 1 至第 50	第 51 至第 100	第 101 至第 150	第 151 至第 200	第 201 至第 250
RigL-ITOP	**3.23**	2.40	2.19	3.04	3.47
A-RigL-ITOP	3.19	**2.43**	**2.29**	**3.16**	**3.54**
SET-ITOP	3.25	2.79	2.89	4.21	4.62
A-SET-ITOP	**3.27**	**2.85**	**2.96**	**4.22**	**4.88**

C. 关于实验设置的更多细节

C.1. 梯度方差与相关性计算

我们计算了 RigL [2]和 SET [1] 在不同稀疏度（包括 0%、50%、80%、90% 和 95%）下，ResNet-50 在 CIFAR-100 上的梯度方差和相关性。计算基于 Sundar 和 Dwaraknath [66]提供的检查点。

梯度方差：我们首先加载训练完成的 0%、50%、80%、90% 和 95% 稀疏模型检查点。然后，为了观察收敛最优点附近的梯度方差，我们对权重添加微小扰动，并计算梯度方差的均值。每个检查点进行三次重复实验。

梯度相关性：我们从稀疏度为 0%、50%、80%、90% 和 95% 的训练完成检查点开始。我们计算并存储所有训练数据上每个权重的梯度。然后，对所有权重添加高斯扰动，并再次计算梯度。最后，计算新扰动权重的梯度与原始权重的梯度之间的相关性。每个检查点进行三次重复实验。

C.2. 实现

在基于 BSR-Net 的结果中，按照 Özdenizci 和 Legenstein [7]，的选择，所有模型的梯度均通过带动量和解耦权重衰减的随机梯度下降计算 [67]。所有模型均训练 200 轮，批次大小为 128。

在基于 RigL 的结果中，我们遵循 Evci 等人 [2], Sundar 和 Dwaraknath [66] 的设置。我们对所有模型训练 250 个训练轮次，批次大小为 128，并使用带动量的随机梯度下降优化参数。

在基于 ITOP 的结果中，我们遵循 Liu 等人 [6] 的设置。对于 CIFAR-10 和 CIFAR-100，我们对所有模型训练 250 个训练轮次，批次大小为 128。对于 ImageNet-2012，我们对所有模型训练 100 个训练轮次，批次大小为 64。参数使用带动量的随机梯度下降进行优化。

C.3. 学习率

与流行的稀疏训练方法 [2, 6, 7], 一致，我们为学习率和权重衰减选择分段常数衰减调度器。在我们的 A-BSR-Net 中，我们将第 50 和第 100 个训练轮次作为学习率衰减调度器的分界点。这是因为我们的方法收敛速度更快，不需要较长的预热期。在稿件所示的评估中，我们还对 BSR-Net 使用该调度器，以进行更准确、公平的比较。

C.4. 初始化（基于 BSR-Net 的结果）

由 ent with [7],我们还选择 Kaiming 初始化来初始化网络权重 [6 8]

C.5. 基准数据集（基于 BSR-Net 的结果）

为了进行公平比较，我们选择与 [7] 相同的基准数据集。具体而言，我们在实验中使用 CIFAR-10、CIFAR-100 [57] 和 SVHN [58] 。CIFAR-10 和 CIFAR-100 数据集均包含 50, 000 张训练图像和 10, 000 张测试图像。SVHN 数据集包含 73, 257 个训练样本和 26, 032 个测试样本。

C.6. 数据增强

我们采用了 [7, 61] 中使用的一种常见数据增强方法。具体而言，我们随机将图像向左或向右平移，再裁剪回原始尺寸，并沿水平方向翻转。此外，所有像素值都归一化到 $[0, 1]$ 范围内。

D. 稀疏训练方法说明

D.1. 稀疏训练概述

随着深度神经网络（DNN）的成本不断增加，人们对模型效率和数据效率等效率问题的关注日益增加 [69–73]。提高模型效率的一种常见解决方案是采用稀疏度；研究发现，训练后的 DNN 参数中存在稀疏模式 [74–76]。稀疏训练是一种常用方法，可将稀疏性引入 DNN，从而提高 DNN 的资源效率。具体而言，要获得一个稀疏度为 90% 的 DNN，我们会随机初始化一个稀疏度为 90% 的 DNN。随后，在整个训练过程中保持权重稀疏，并每隔 m 次迭代剪枝和重新生长一定数量的权重。这样，我们便能节省训练内存，同时生成性能与稠密模型相当的稀疏模型。下面将介绍几种广泛使用的稀疏训练方法。

D.2. 稀疏训练方法：SET

SET [1] 是一种广泛使用的稀疏训练方法，通过检查权重的大小来剪枝并重新生长连接。

D.3. 稀疏训练方法：RigL

RigL [2] 是另一种流行的动态稀疏训练方法，它利用权重和梯度的大小来学习连接。

D.4. 稀疏训练方法：BSR-Net

贝叶斯稀疏鲁棒训练（BSR-Net） [7] 是一种贝叶斯稀疏鲁棒训练流程。基于贝叶斯后验采样原理，网络重连过程会依据对抗学习目标，同时学习稀疏连接结构以及鲁棒性与准确率权衡。具体而言，在掩码更新方面，该方法会剪除所有负权重，并随机生成新的权重。

D.5. 稀疏训练方法：ITOP

ITOP [6] 是另一种近期提出的动态稀疏训练流程，它通过充分且可靠的参数探索，实现及时过参数化并找到性能良好的稀疏模型。

E. 自适应梯度校正方法的局限性

E.1. 额外浮点运算次数

与 SVRG 类似，我们的 AGENT 会增加每次迭代的训练浮点运算次数，因为计算旧梯度需要额外进行前向传播和反向传播。
然而，实际的计算差异可能更小，而且基于图形处理器运行的 SVRG，其运行时间受到的影响不会那么大。例如，在对抗场景中，我们需要额外计算来生成对抗样本，这一过程耗时较长，并且在我们的 AVR 和随机梯度下降中，每次迭代只需执行一次。对于 BSR-Net，我们通过实验证实，我们的 AVR 与随机梯度下降每次迭代所需时间之比约为 1.2。

还有几种方法可以减少 SVRG 带来的额外计算。第一种方法是使用 M Elibol（2020年）提出的稀疏梯度 [51]。它可以有效降低 SVRG 的计算成本，并且很容易应用于我们的方法。第二种方法由 Allen-Zhu 和 Hazan（2016） [49]提出。计算旧模型参数上的批次梯度所产生的额外开销完全可以并行化。因此，我们可以将 SVRG 视为把小批量大小加倍。第三，我们可以采用 SAGA [40] 的思路，存储单个样本的梯度。这样，我们就不需要额外执行前向传播和反向传播步骤，从而节省计算量。但这种方法需要额外的内存来存储梯度。

在正文中，我们选择比较 ADSVRG 与随机梯度下降在遍历相同数据次数（训练轮次）时的收敛速度，这是比较基于 SVRG 的优化方法与随机梯度下降时广泛采用的标准 [18, 22, 46, 49]。这种比较方式可以展示该优化方法的加速效果，并为未来的工作提供启发。

E.2. 大型数据集或模型中的低效问题

当数据量较大时，我们的 AGENT 中额外的梯度计算可能会非常庞大。但我们可以随机抽取一小部分数据，并在这部分数据上计算 $\tilde{g}$，从而大幅节省额外计算量。具体而言，我们可以抽取一个规模大于批次数据、但小于完整数据的子数据集。在这部分小规模数据上计算 $\tilde{g}$时，与在批次数据上计算梯度相比，噪声会更少。因此，我们可以在 AGENT 中使用这部分小规模数据上的 $\tilde{g}$ 来校正梯度并加速稀疏训练。以 ImageNet-2012 为例，这是一个大型数据集。我们使用 128 的批次大小。在每个训练轮次中，我们在大小为 50,000 的子数据集上计算$\tilde{g}$，而不是在大小为 128 万的完整数据集上计算 $\tilde{g}$。

当模型参数数量较大时，存储和访问额外梯度所带来的内存负担可能相当显著。但我们可以通过稀疏矩阵 [77]大幅节省内存。在稀疏训练期间，我们只需更新活跃权重。因此，只需存储活跃权重对应的梯度。存储的梯度是稀疏的，可以存放在稀疏矩阵中以节省内存。

E.3. 缩放参数调优

在我们的自适应方差缩减方法（AVR）中，我们增加了一个需要调整的缩放参数 γ。我们发现，对于 BSR-Net、RigL 和 ITOP，设置为 $\gamma = 0.1$ 是一个不错的选择。不过，对于其他不同的稀疏训练流程，该参数可能有所不同。

E.4. 鲁棒准确率下降

对于基于 BSR-Net 的模型，其最终准确率结果在使用 AVR 后，鲁棒准确率略有下降。在稀疏训练和对抗训练中使用自适应方差缩减时，如何进一步提高鲁棒准确率仍是一个尚未解决的问题。