Методы редукции дисперсии, не предполагающие вычисление полного градиента: повышение эффективности за счёт техники случайного перемешивания батчей

Алексей Витальевич Ребриков
Научный руководитель: к.ф.-м.н. А. Н. Безносиков

Кафедра интеллектуальных систем ФПМИ МФТИ
Специализация: Интеллектуальный анализ данных
Направление: 03.03.01 Прикладные математика и физика

2025

Редукция дисперсии и полный градиент

Проблема: Ставится задача оптимизации конечной суммы функций.

Цель: Предложить модификацию известного алгоритма редукции дисперсии, исключив необходимость подсчета полного градиента.

Решение: Предлагается модификацию алгоритма `SARAH` с использованием техники случайного перемешивания батчей.

Постановка задачи

Рассматривается задача минимизации конечной суммы:

$$\min_{x \in \mathbb{R}^d} f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x),$$

где $f_i : \mathbb{R}^d \to \mathbb{R}$, $n \gg 1$. Методы снижения дисперсии (VR) типа SARAH требуют вычисления полного градиента $\nabla f(x)$. Это дорого при большом n.

Предположения

Рассматриваются следующие условия на функции f_i:

- **L-гладкость:** $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|$ для любых $x, y \in \mathbb{R}^d$.
- **μ-сильная выпуклость:** $f_i(y) \geq f_i(x) + \langle \nabla f_i(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2$.
- **Невыпуклость функции f:** $f^* := \inf_{x \in \mathbb{R}^d} f(x) > -\infty$.

Алгоритм: No Full Grad SARAH

Обновление градиента:

$$v_s^t = \frac{1}{n}(\nabla f_{\pi_s^t}(x_s^t) - \nabla f_{\pi_s^t}(x_s^{t-1})) + v_s^{t-1}$$

Аппроксимация полного градиента:

$$\widetilde{v}_s^{t+1} = \frac{t-1}{t}\widetilde{v}_s^t + \frac{1}{t}\nabla f_{\pi_s^t}(x_s^t), \quad v_{s+1} = \widetilde{v}_s^{n+1}$$

Эвристика: при каждой эпохе осуществляется случайная перестановка индексов (random reshuffling, RR), что улучшает сходимость.

Алгоритм: No Full Grad SARAH (псевдокод)

Вход: $x_0^0 \in \mathbb{R}^d$, $\widetilde{v}_0^0 = 0^d$, $v_0 = 0^d$
Параметр: шаг $\gamma > 0$
for эпохи $s = 0, 1, 2, \ldots$ **do**
    случайная перестановка $\pi_s^1, \ldots, \pi_s^n$
    $v_s^0 = v_s$
    $x_s^1 = x_s^0 - \gamma v_s^0$
    for $t = 1, \ldots, n$ **do**
        $\widetilde{v}_s^{t+1} = \frac{t-1}{t}\widetilde{v}_s^t + \frac{1}{t}\nabla f_{\pi_s^t}(x_s^t)$
        $v_s^t = \frac{1}{n}(\nabla f_{\pi_s^t}(x_s^t) - \nabla f_{\pi_s^t}(x_s^{t-1})) + v_s^{t-1}$
        $x_s^{t+1} = x_s^t - \gamma v_s^t$
    end for
    $x_{s+1}^0 = x_s^{n+1}$, $\widetilde{v}_{s+1}^1 = 0^d$, $v_{s+1} = \widetilde{v}_s^{n+1}$
end for

Теоретические результаты

Все f_i — L-гладкие, n — размер выборки, γ — шаг метода.

Невыпуклый случай:

$$\gamma \leq \frac{1}{20L(n+1)} \quad \varepsilon^2 = \frac{1}{S}\sum_{s=1}^{S} \|\nabla f(x_s^0)\|^2 \quad \Longrightarrow \quad \mathcal{O}\left(\frac{nL}{\varepsilon^2}\right)$$

Сильно выпуклый случай:

$$\gamma \leq \frac{1}{20L(n+1)} \quad \varepsilon = f(x_{S+1}^0) - f(x^*) \quad \Longrightarrow \quad \mathcal{O}\left(\frac{nL}{\mu}\log\frac{1}{\varepsilon}\right)$$

Сравнение методов

Алгоритм	Нет полного градиента?	Память	Невыпуклый случай	Сильно выпуклый случай
SAGA	✓	$\mathcal{O}(nd)$	—	$\mathcal{O}\left(n\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$
IAG	✓	$\mathcal{O}(nd)$	—	$\mathcal{O}\left(n^2\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$
PIAG	✓	$\mathcal{O}(nd)$	—	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
DIAG	✓	$\mathcal{O}(nd)$	—	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
Prox-DFinito	✓	$\mathcal{O}(nd)$	—	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
AVRG	✓	$\mathcal{O}(d)$	—	$\mathcal{O}\left(n\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$
SVRG	✗	$\mathcal{O}(d)$	—	$\mathcal{O}\left(n^3\frac{L^2}{\mu^2}\log\frac{1}{\varepsilon}\right)$
SVRG	✗	$\mathcal{O}(d)$	$\mathcal{O}\left(\frac{nL}{\varepsilon^2}\right)$	$\mathcal{O}\left(n\frac{L^{3/2}}{\mu^{3/2}}\log\frac{1}{\varepsilon}\right)$
SARAH	✓	$\mathcal{O}(d)$	—	$\mathcal{O}\left(n^2\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$
NFG SARAH	✓	$\mathcal{O}(d)$	$\mathcal{O}\left(\frac{nL}{\varepsilon^2}\right)$	$\mathcal{O}\left(n\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$

Эксперимент: CIFAR-10 + ResNet18

Рассматривается задача многоклассовой классификации на датасете CIFAR-10,

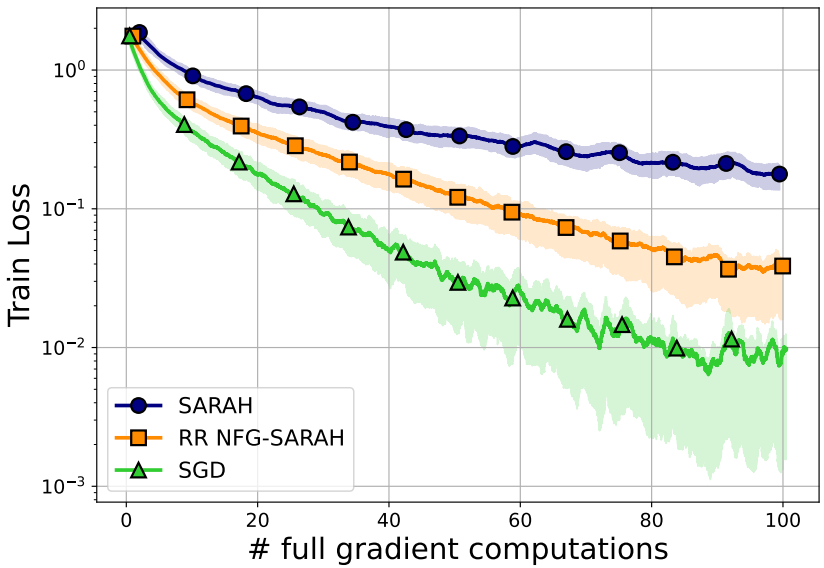
- 60 000 изображений размером 32×32
- 10 классов (по 6 000 изображений на класс)

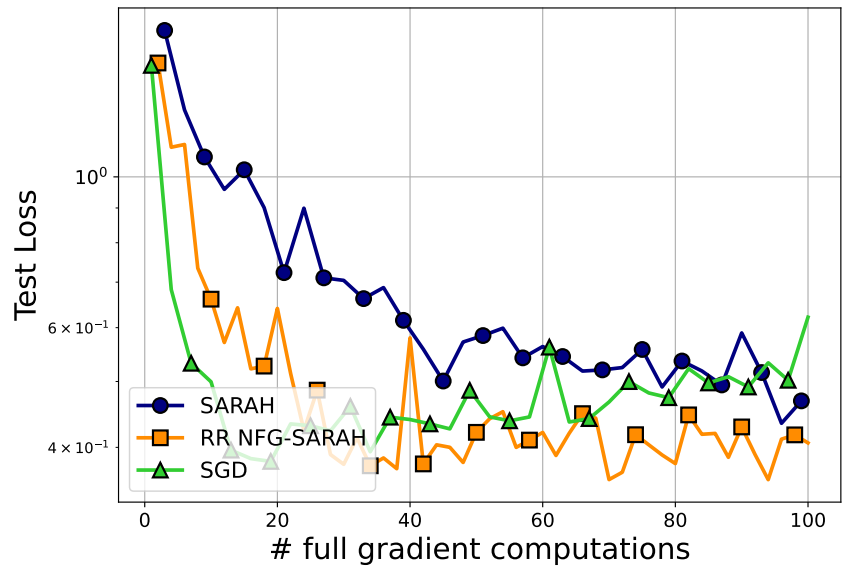
Используется классическая архитектура модели ResNet-18
Оптимизируется стандартная функция потерь — кросс-энтропия:

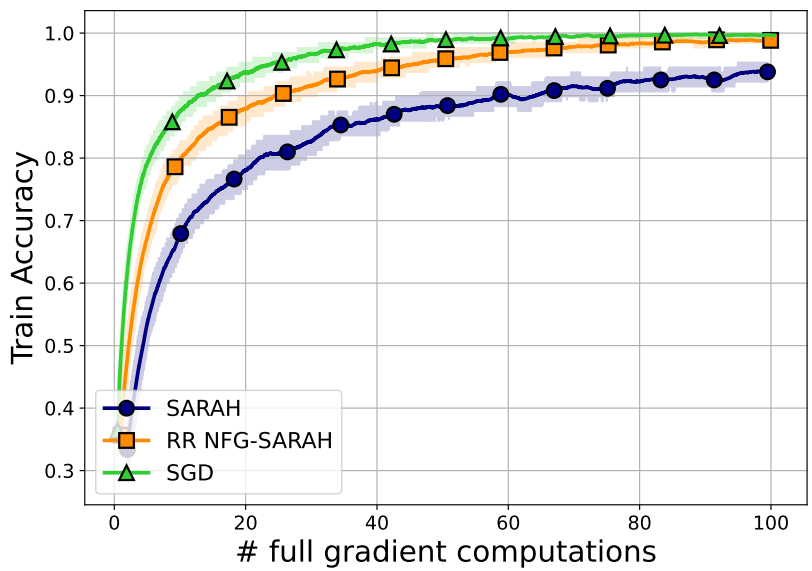
$$\min_w \frac{1}{M} \sum_{i=1}^{M} \ell(f_w(x_i), y_i),$$

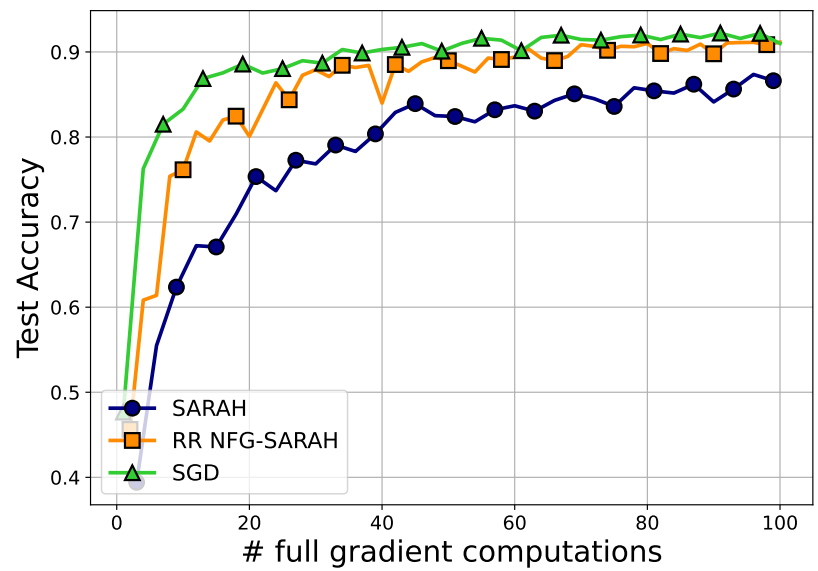
где w — параметры модели, $f_w(x_i)$ — предсказание модели на входе x_i, y_i — истинная метка, ℓ — кросс-энтропия.

Графики


Train Loss
full gradient computations
SARAH
RR NFG-SARAH
SGD


Test Loss
full gradient computations
SARAH
RR NFG-SARAH
SGD


Train Accuracy
full gradient computations
SARAH
RR NFG-SARAH
SGD


Test Accuracy
full gradient computations
SARAH
RR NFG-SARAH
SGD

Эксперимент: CIFAR-100 + ResNet18

Задача многоклассовой классификации на датасете CIFAR-100:

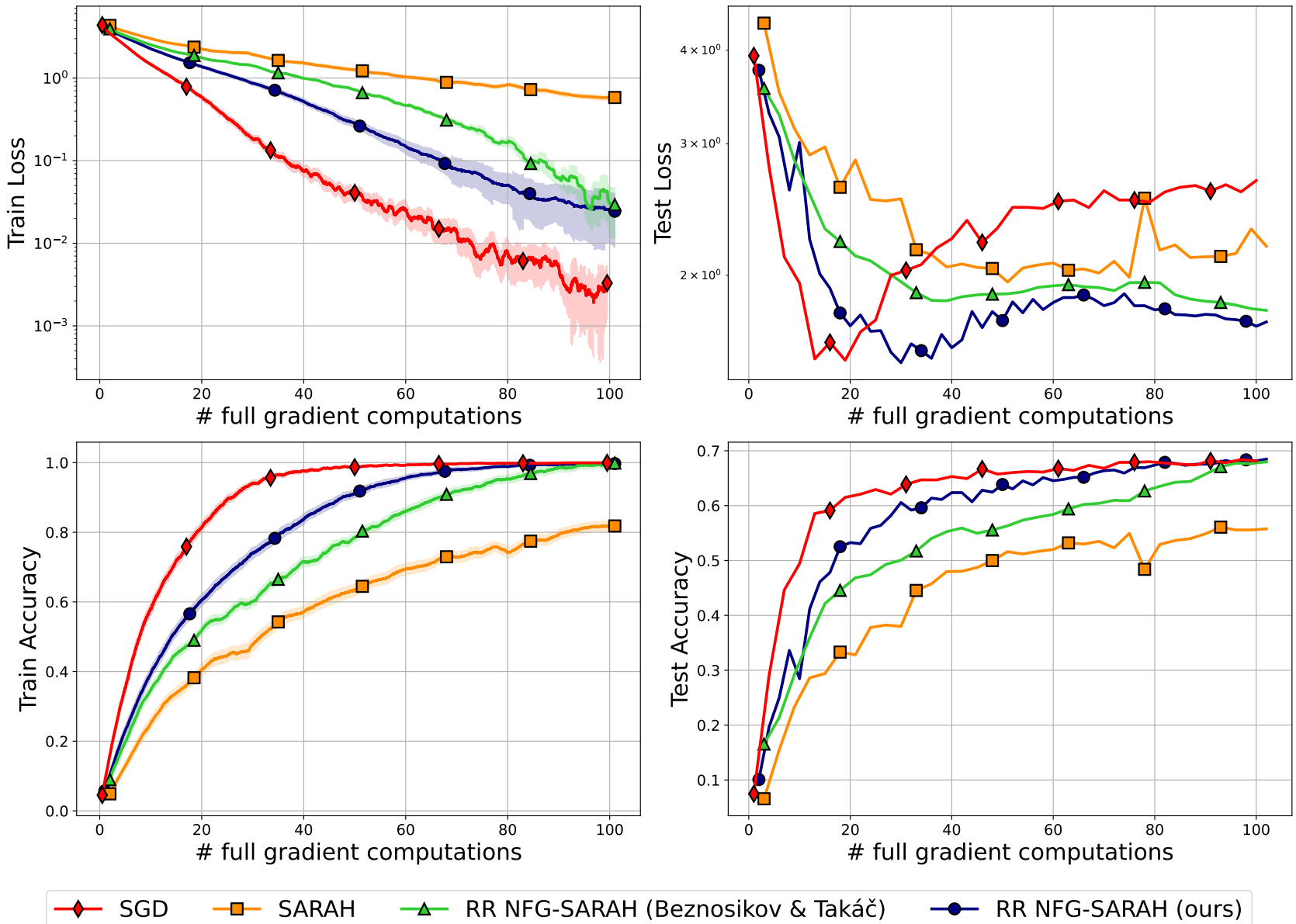
- 60 000 изображений 32×32
- 100 классов (по 600 изображений на класс)

Используется архитектура ResNet-18.
Функция потерь — кросс-энтропия:

$$\min_w \frac{1}{M} \sum_{i=1}^{M} \ell(f_w(x_i), y_i),$$

где w — параметры модели, $f_w(x_i)$ — выход модели, y_i — метка, ℓ — кросс-энтропия.
Метод No Full Grad SARAH сравнивается с классическим SARAH.

Графики: CIFAR-100



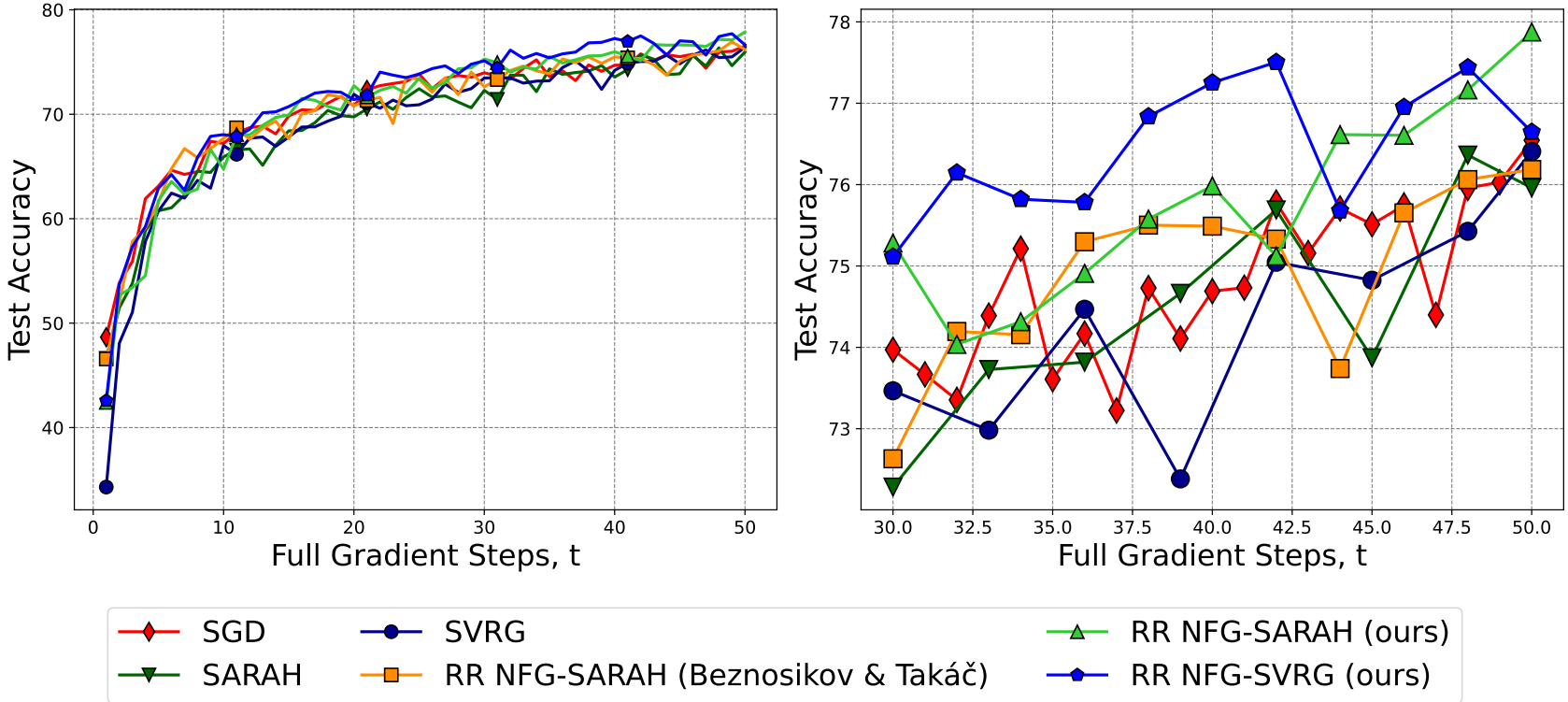
Эксперимент: Tiny ImageNet + Swin Transformer

Задача классификации изображений на Tiny ImageNet:

- 200 классов, изображения 64×64
- масштабирование до 224×224 для Swin

Используется модель Tiny Swin Transformer (`swin_T_patch4_window7_224`), инициализированная предобученными весами с ImageNet-1K.
Размер батча: 256, градиентный клиппинг: 1.0. Метрики: точность и кросс-энтропия. Сравниваются методы: SGD, SARAH, RR NFG-SARAH, предложенный No Full Grad SARAH.

Графики: Tiny ImageNet



Выносится на защиту

- Предложен новый вариант метода **SARAH**, не использующий вычисление полного градиента.
- Использование перемешивания и скользящего среднего позволило аппроксимировать полный градиент без дополнительной памяти.
- Методы обладают улучшенными
  - **Затратами памяти:** требуется $\mathcal{O}(d)$ вместо $\mathcal{O}(nd)$
  - **Сходимостью:** лучшие оценки по числу итераций
- Проведены эксперименты (CIFAR-10/CIFAR-100 + ResNet18 и Tiny ImageNet + Swin Transformer), подтверждающие теоретические преимущества.