Memory-Efficient 4-bit Preconditioned Stochastic Optimization

Supplementary Material

A. Practical 32-bit Shampoo

In this section, we provide the practical 32-bit Shampoo introduced in Sec. 3.1 and summarize it in Algorithm 2.

Algorithm 2 Practical 32-bit Shampoo

Input: initial weight $W_0 \in \mathbb{R}^{m\times n}$, initial preconditioning matrices $L_0 = \epsilon I_m$, $R_0 = \epsilon I_n$, $\hat{L}_0 = I_m$, $\hat{R}_0 = I_n$. Total update steps T, interval of updating preconditioners T_1 and T_2, momentum parameter $\beta \in (0,1)$. First-order optimizer $\mathcal{F}$ with initial optimizer state s_0.

Output: final weight W_T.

```
1:  for $k = 1, 2, \dots, T$ do
2:      Compute gradient $G_k = \nabla \mathcal{L}_k(W_k)$
3:      if $k \% T_1 \equiv 0$ then
4:          $\bar{L}_k = \beta L_{k-1} + (1-\beta) G_k G_k^T$
5:          $\bar{R}_k = \beta R_{k-1} + (1-\beta) G_k^T G_k$
6:      else
7:          $L_k = L_{k-1},\ R_k = R_{k-1}$
8:      end if
9:      if $k \% T_2 \equiv 0$ then
10:         Compute maximum singular values $\lambda_{\max}^L$ and $\lambda_{\max}^R$ of $L_k$ and $R_k$ by power iteration
11:         Compute $\hat{L}_k = (L_k + \lambda_{\max}^L \epsilon I_m)^{-1/4}$ and $\hat{R}_k = (R_k + \lambda_{\max}^R \epsilon I_n)^{-1/4}$ by Schur-Newton iteration
12:     else
13:         $\hat{L}_k = \hat{L}_{k-1}; \quad \hat{R}_k = \hat{R}_{k-1}$
14:     end if
15:     $\hat{G}_k = \hat{L}_k G_k \hat{R}_k; \quad \tilde{G}_k = (\|G_k\|_F / \|\hat{G}_k\|_F) \cdot \hat{G}_k$
16:     $W_k, s_k = \mathcal{F}(W_{k-1}, s_{k-1}, \tilde{G}_k)$
17: end for
```

B. Proofs in Theoretical Analysis

We vectorize the update scheme as follows. Starting with the matrix form:

$$W_{k+1} = W_k - \eta_k \mathcal{D}(\hat{L}_k) G_k \mathcal{D}(\hat{R}_k),$$

and applying vectorization, we get:

$$\mathrm{Vec}(W_{k+1}) = \mathrm{Vec}(W_k) - \eta_k \left(\mathcal{D}(\hat{R}_k) \otimes \mathcal{D}(\hat{L}_k)\right) \mathrm{Vec}(G_k).$$

Let $x_k := \mathrm{Vec}(W_k)$, $g_k := \mathrm{Vec}(G_k)$, and $H_k := \mathcal{D}(\hat{R}_k) \otimes \mathcal{D}(\hat{L}_k)$. we obtain the vectorized update scheme:

$$x_{k+1} = x_k - \eta_k H_k g_k, \tag{17}$$

where $\{H_k\}$ is a sequence of positive definite matrices.

Proposition B.1. *For a b-bit quantization and any vector $x \in \mathbb{R}^d$, the following bound holds:*

$$\|\mathcal{D}(\mathcal{Q}(x)) - x\|_\infty \le \frac{\|x\|_\infty}{2^b}.$$

Proof. Consider any real number $a \in [-1,1]$. In a b-bit quantization system, the interval between two consecutive representable values is given by $\Delta = \frac{2}{2^b} = \frac{1}{2^{b-1}}$. Thus, the quantization error satisfies $|\mathcal{Q}(a) - a| \le \frac{\Delta}{2} = \frac{1}{2^b}$.

For any vector $x \in \mathbb{R}^d$, we apply the definitions of the operators $\mathcal{Q}$ and $\mathcal{D}$ as follows:

$$\begin{aligned}
&\|\mathcal{D}(\mathcal{Q}(x)) - x\|_\infty \\
=&\left\| \|x\|_\infty \mathcal{Q}\left(\frac{x}{\|x\|_\infty}\right) - \|x\|_\infty \frac{x}{\|x\|_\infty} \right\|_\infty \\
=&\|x\|_\infty \left\| \mathcal{Q}\left(\frac{x}{\|x\|_\infty}\right) - \frac{x}{\|x\|_\infty} \right\|_\infty \\
\le&\|x\|_\infty \cdot \frac{1}{2^b}.
\end{aligned}$$

This completes the proof. □

Proposition B.2. *For the 4-bit Shampoo in Algorithm 1, let $M_k := (\mathcal{D}(\bar{C}_k^L)\mathcal{D}(\bar{C}_k^L)^T + \lambda_{\max}^L \epsilon I_m)^{-1/4}$, if $\|M_k\|_{\text{off,max}} \le C_B$, then its preconditioners hold that*

$$\mathcal{D}(\hat{L}_k) \preceq M_k + C_B n_k 2^{-b} I,$$

where $\|\cdot\|_{\text{off,max}}$ is the maximal absolute value of all off-diagonal entries and n_k is the number of rows in W_k. Furthermore, if for every row index i it holds that $|[M_k]_{ii}| > \left(1 + \frac{2}{2^b - 1}\right)\sum_{j\neq i} |[M_k]_{ij}|$, then $\mathcal{D}(\hat{L}_k) \succ 0$.

Proof. Unroll the update in Step 4, we have

$$\begin{aligned}
&L_k \\
=&\beta L_{k-1} + (1-\beta) G_k G_k^T \\
=&\beta(\beta L_{k-2} + (1-\beta) G_{k-1} G_{k-1}^T) + (1-\beta) G_k G_k^T \\
=&\beta^2 L_{k-2} + (1-\beta)(G_k G_k^T + \beta G_{k-1} G_{k-1}^T) \\
&\cdots \\
=&\beta^k L_0 + (1-\beta) \sum_{i=0}^{k-1} \beta^i G_{k-i} G_{k-i}^T \\
=&\beta^k L_0 + (1-\beta) \sum_{i=0}^{k-1} \beta^i G_{k-i} G_{k-i}^T \\
\succeq&0.
\end{aligned}$$

Thus Step 11 is well-defined. Since only off-diagonal part is quantized, by Step 6, we have

$$\begin{aligned}\mathcal{D}(\hat{L}_k) =&\mathcal{D}(\mathcal{Q}(M_k))\\ =&\mathcal{D}(\mathcal{Q}(S_k - \mathrm{Diag}(M_k))) + \mathrm{Diag}(M_k)\\ =&M_k - \mathrm{Diag}(M_k) + \mathrm{Diag}(M_k) + E_k\\ =&M_k + E_k,\end{aligned}\tag{18}$$

where $E_k = (M_k - \mathrm{Diag}(M_k)) - \mathcal{D}(\mathcal{Q}(M_k - \mathrm{Diag}(M_k)))$. By Proposition B.1, we have

$$\begin{aligned}&\|E_k\|_{\max}\\ \leq& \|M_k - \mathrm{Diag}(M_k)\|_{\max} 2^{-b}\\ \leq& \left\|(\mathcal{D}(\bar{C}_k^L)\mathcal{D}(\bar{C}_k^L)^T + \lambda_{\max}^L \epsilon I_m)^{-1/4}\right\|_{\mathrm{off},\max} 2^{-b}\\ \leq& C_B 2^{-b},\end{aligned}$$

where $\|\cdot\|_{\max}$ is the largest entry in magnitude of a matrix. Note that for any $x \in \mathbb{R}^d$,

$$|x^T E_k x| \leq C_B 2^{-b} (e^T|x|)^2 \leq C_B n_k 2^{-b} \|x\|^2,$$

where e is the vector with all elements being 1 and $|\cdot|$ is the operator of taking element-wise absolute value. Therefore, we have

$$\begin{aligned}&\mathcal{D}(\hat{L}_k)\\ =&(\mathcal{D}(\bar{C}_k^L)\mathcal{D}(\bar{C}_k^L)^T + \lambda_{\max}^L \epsilon I_m)^{-1/4} + E_k,\\ \preceq&(\mathcal{D}(\bar{C}_k^L)\mathcal{D}(\bar{C}_k^L)^T + \lambda_{\max}^L \epsilon I_m)^{-1/4} + C_B n_k 2^{-b} I.\end{aligned}$$

Moreover, if $|[M_k]_{ii}| > \left(1 + \frac{2}{2^b - 1}\right)\sum_{j\neq i} |[M_k]_{ij}|$ for any row index i, then by Eq. (18), we have

$$\begin{aligned}&\left|\left[\mathcal{D}(\hat{L}_k)\right]_{ii}\right| - \sum_{j\neq i}\left|\left[\mathcal{D}(\hat{L}_k)\right]_{ij}\right|\\ \geq&(|[M_k]_{ii}| - |[E_k]_{ii}|) - \left(\sum_{j\neq i}|[M_k]_{ij}| + \sum_{j\neq i}|[E_k]_{ij}|\right)\\ \geq&(1 - 2^{-b})|[M_k]_{ii}| + (1 + 2^{-b})\sum_{j\neq i}|[M_k]_{ij}|\\ >&0,\end{aligned}$$

where the second inequality follows from Proposition B.1 and the last inequality follows from the strongly diagonal dominance. By Gershgorin Circle Theorem, we have $\mathcal{D}(\hat{L}_k) \succ 0$. This completes the proof. □

Given a matrix S, the proof of Proposition B.2 shows that if we quantize only the off-diagonal entries of S while keeping the diagonal entries, the quantization error E satisfies $\|E\|_\infty \leq 2^{-b}\|S\|_{\mathrm{off},\infty}$. However, if the entire S is quantized, the error becomes $2^{-b}\|S\|_\infty$. When the diagonal entries of S dominate each row, this selective quantization method can significantly reduce the quantization error.

B.1. Smooth Nonconvex Training Loss

Theorem B.1. *Suppose Assumption 5.1 holds. Let $\eta_k = \frac{c}{\sqrt{T+1}}$ with $c \in \left(0, \frac{\lambda_{H,\min}}{2L(1+\sigma^2)\lambda_{H,\max}^2}\right)$, then we have*

$$\mathbb{E}\left[\|\nabla f(\bar{x}_T)\|_2^2\right] \leq \frac{2(f(x_0) - \bar{f} + c^2 L\sigma^2\lambda_{H,\max}^2)}{c\lambda_{H,\min}\sqrt{T+1}},$$

where $\bar{x}_T$ is randomly selected from $\{x_0, x_1, ..., x_T\}$, and $\bar{f} = \min_{x\in\mathbb{R}^d} f(x)$.

Proof. Without any ambiguity, $\|\cdot\|$ denotes the L_2 norm of a vector or the spectral norm of a matrix. By Lipschitz smoothness, we have

$$\begin{aligned}&f(x_{k+1})\\ \leq& f(x_k) + \langle\nabla f(x_k), x_{k+1} - x_k\rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2\\ =& f(x_k) - \eta_k\langle\nabla f(x_k), H_k g_k\rangle + \frac{L\eta_k^2}{2}\|H_k g_k\|^2\\ \leq& f(x_k) - \eta_k\langle\nabla f(x_k), H_k g_k\rangle + L\eta_k^2\|H_k\nabla f(x_k)\|^2\\ &+ L\eta_k^2\|H_k(\nabla f(x_k) - g_k)\|^2.\end{aligned}$$

Rearranging the terms and taking expectations, we get

$$\begin{aligned}&\eta_k\mathbb{E}\left[\|\nabla f(x_k)\|_{H_k}^2\right]\\ \leq&\mathbb{E}[f(x_k)] - \mathbb{E}[f(x_{k+1})] + L\eta_k^2\mathbb{E}\left[\|H_k\nabla f(x_k)\|^2\right]\\ &+ L\sigma^2\eta_k^2\|H_k\|^2(1 + \|\nabla f(x_k)\|^2).\end{aligned}$$

By the choice of c, we have

$$\begin{aligned}&\frac{1}{2}\eta_k\|\nabla f(x_k)\|_{H_k}^2\\ \geq& L\eta_k^2\left(\|H_k\nabla f(x_k)\|^2 + \sigma^2\|H_k\|^2\|\nabla f(x_k)\|^2\right),\end{aligned}$$

we have

$$\begin{aligned}&\frac{\sum_{k=0}^T \eta_k\mathbb{E}\left[\|\nabla f(x_k)\|_{H_k}^2\right]}{2\sum_{k=0}^T\eta_k}\\ \leq&\frac{f(x_0) - \bar{f} + L\sigma^2\lambda_{H,\max}^2\sum_{k=0}^T\eta_k^2}{\sum_{k=0}^T\eta_k}.\end{aligned}$$

In particular, when $\eta_k = \frac{c}{\sqrt{T+1}}$, we have

$$\mathbb{E}\left[\|\nabla f(\bar{x}_k)\|^2\right] \leq \frac{2(f(x_0) - \bar{f} + c^2 L\sigma^2\lambda_{H,\max}^2)}{c\lambda_{H,\min}\sqrt{T+1}}.$$

□

B.2. Nonsmooth Nonconvex Training Loss

Conventional techniques in stochastic optimization for nonsmooth nonconvex scenarios typically rely on the time-homogeneity of the associated dynamical system, as shown in [4, 13]. Given a locally Lipschitz function f, by Rademacher's theorem, f is differentiable almost everywhere. Thus, we have the following definition of subdifferential for a locally Lipschitz function.

Definition B.1. *The Clark subdifferential or subgradient [9] is defined as*

$$\partial f(x) := \left\{ \begin{matrix} y:\ x_k \to x,\, \nabla f(x_k) \to y, \\ \text{where } f \text{ is differentiable at } x_k \end{matrix} \right\}.$$

The class of locally Lipschitz functions is too broad for a meaningful convergence analysis: there even exist highly pathological examples whose subgradient trajectories fail to converge to any critical point [12]. In contrast, neural-network losses exhibit a much richer, but still well-structured nonsmooth geometry: they are in fact "piecewise-smooth" because their non-differentiabilities arise only from simple components (e.g. ReLU activations). A convenient formalism for capturing this structure is that of Whitney stratifiable functions, which we adopt throughout.

Definition B.2. *A locally Lipschitz function is C^p-Whitney stratifiable [13], if the graph of f:* $\mathrm{graph}(f) := \{(x,t) : f(x) = t\}$ *can be decomposed into finite C^p manifolds, called strata, satisfying*

1. *For any two strata $\mathcal{M}_1$ and $\mathcal{M}_2$, the following implication holds:*

$$\mathcal{M}_1 \cap \overline{\mathcal{M}_2} \neq \emptyset \implies \mathcal{M}_1 \subset \overline{\mathcal{M}_2}$$

2. *For any sequence of points z_k in a stratum $\mathcal{M}_1$ converging to a point $\bar{z}$ in a stratum $\mathcal{M}_2$, if the corresponding normal vectors $v_k \in N_{\mathcal{M}_1}(z_k)$ converge to a vector v, then the inclusion $v \in N_{\mathcal{M}_2}(\bar{z})$ holds. Here $N_{\mathcal{M}_i}$ is the normal space of $\mathcal{M}_i$.*

For instance, consider the function $x \mapsto -|x|$, which is C^∞-Whitney stratifiable: its graph decomposes into the smooth submanifolds

$$\{(0,0)\}, \quad \{(t,-t)\colon t>0\}, \quad \text{and} \quad \{(t,t)\colon t<0\}.$$

Moreover, it has been shown that the loss functions of virtually all modern neural networks admit a Whitney stratification [5]. Consequently, we restrict our convergence analysis to Whitney stratifiable functions. The key step is to prove that the continuous-time limit of the piecewise-linear interpolation

$$x(t) = x_k \ + \ \frac{t-t_{k-1}}{\eta_k}\,(x_{k+1}-x_k), \quad t \in [t_{k-1}, t_k),$$

is a solution to the subgradient differential inclusion, where $t_k = \sum_{i=1}^k \eta_i$, $t_0 = 0$. The Whitney stratification then allows us to transfer the nonsmooth analysis onto each smooth stratum by exploiting their topological and geometric regularity (Definition B.2). Such stratification-based methods have been widely employed to establish convergence guarantees for contemporary deep-learning algorithms in nonsmooth settings [5, 13, 55].

Theorem B.2. *Suppose Assumption 5.2 holds, and assume the sequence $\{x_k\}$ remains within a compact set. If the learning rate satisfies $\sum_{k=1}^\infty \eta_k = \infty$ and $\sum_{k=1}^\infty \eta_k^2 < \infty$, then*

$$\lim_{k\to\infty} \mathrm{dist}(x_k, \Omega) = 0,$$

where $\Omega := \{x : 0 \in \partial f(x)\}$ is the set of stationary points.

Proof. Define the interpolated process $x(t)$ for $\{x_k\}$ as follows:

$$x(t) := x_k + \frac{t-t_{k-1}}{\eta_k}(x_{k+1}-x_k), \quad \text{for } t \in [t_{k-1}, t_k),$$

where $t_k := \eta_1 + \cdots + \eta_k$, $t_0 = 0$. Define $y(t) := H_k d_k$ for $t \in [t_{k-1}, t_k)$, where $d_k \in \partial f(x_k)$. Thus, both $x(t)$ and $y(t)$ are piecewise linear functions. We also define time-shifted versions $y^t(\cdot) := y(t+\cdot)$.

Let $x_t(\cdot)$ denote the solution to the following ODE:

$$\dot{x}_t(\tau) = -y(\tau), \quad x_t(t) = x(t), \quad \text{for any } \tau \geq t.$$

By Assumption 5.2, $\sup_k \|d_k\| \leq \ell$, so $\sup_{t\geq 0} \|y(t)\| \leq M\ell$. Therefore, the class of functions $\{x_t(\cdot) : t \geq 0\}$ is uniformly equicontinuous. Using the assumptions on $\{\xi_k\}$, the learning rate $\{\eta_k\}$, and the boundedness of H_k, it follows from [18, Lemma A.1] that for any $T > 0$,

$$\lim_{t\to\infty} \sup_{\tau\in[t,t+T]} \|x(\tau) - x_t(\tau)\| = 0.$$

Since $x(\cdot)$ is pointwise bounded, $x_t(\cdot)$ is also pointwise bounded. By the Arzelà-Ascoli theorem, the equicontinuity of $\{x_t(\cdot) : t \geq 0\}$ implies that it is relatively compact in the space of continuous functions, under the topology of uniform convergence over any compact set. The relative compactness of $\{y^t(\cdot)\}$ can be similarly verified; see [4, 6] for further details on related functional analysis concepts.

For any fixed $T > 0$, by the definition of $x_t(\cdot)$, we have

$$x_t(t+T) = x_t(t) - \int_0^T y^t(s)\,ds.$$

Now, select a subsequence $\{t_{k_j}\}$ such that the sequences $\{x_t(\cdot)\}$ and $\{y^t(\cdot)\}$ converge to $\bar{x}(\cdot)$ and $\bar{y}(\cdot)$, respectively, as $j \to \infty$. Letting $j \to \infty$, we obtain

$$\bar{x}(T) = \bar{x}(0) - \int_0^T \bar{y}(s)\,ds.$$

Next, we show that $\bar{y}(s) \in \bar{H}\partial f(\bar{x}(s))$. Note that

$$\begin{aligned}
&\operatorname{dist}\left(\bar{y}(s), \bar{H}\partial f(\bar{x}(s))\right) \\
\leq &\left\| \frac{1}{N}\sum_{j=1}^{N} y^{t_{k_j}}(s) - \bar{y}(s) \right\| \\
&+ \operatorname{dist}\left(\frac{1}{N}\sum_{j=1}^{N} y^{t_{k_j}}(s), \bar{H}\partial f(\bar{x}(s)) \right) \\
\leq &\operatorname{dist}\left(\frac{1}{N}\sum_{j=1}^{N} H_{\lambda(t_{k_j}+s)} d_{\lambda(t_{k_j}+s)}, \bar{H}\partial f(\bar{x}(s)) \right) + o(1),
\end{aligned}$$

where $\lambda(t) = k$ such that $t_k < t \leq t_{k+1}$. Since $d_{\lambda(t_{k_j}+s)} \in \partial f(x_{\lambda(t_{k_j}+s)})$, by the outer-semicontinuity of ∂f, we have $\operatorname{dist}\left(d_{\lambda(t_{k_j}+s)}, \partial f(\bar{x}(s))\right) \to 0$. Using Assumption 5.2c), we have

$$\begin{aligned}
&\operatorname{dist}\left(\bar{y}(s), \bar{H}\partial f(\bar{x}(s))\right) \\
\leq &\operatorname{dist}\left(\frac{1}{N}\sum_{j=1}^{N} H_{\lambda(t_{k_j}+s)} d_{\lambda(t_{k_j}+s)}, \bar{H}\partial f(\bar{x}(s)) \right) + o(1) \\
&\to 0.
\end{aligned}$$

Thus, we conclude the following:

$$\bar{x}(T) = \bar{x}(0) - \int_0^T \bar{y}(s)\,ds, \quad \text{and } \bar{y}(s) \in \bar{H}\partial f(\bar{x}(s)). \tag{19}$$

Since f is stratifiable, by [13, Theorem 3.2], any limit point of $\{x_k\}$ converges to the stable set of (19), namely, $\{x : 0 \in \bar{H}\partial f(x)\} = \{x : 0 \in \partial f(x)\} = \Omega$. This completes the proof. □

C. Experimental Details

C.1. Toy Example

Here we compare Cholesky quantization (CQ) and vanilla quantization (VQ) on a toy 2×2 matrix using 4-bit linear-2 quantization as introduced in Sec. 3.2. The original matrix, with eigenvalues $(10.908, 0.092)$, has a high condition number. VQ perturbs matrix elements, distorting the spectrum and producing a negative eigenvalue -0.164, breaking PD. In contrast, CQ quantizes the Cholesky factor, preserving structure and yielding eigenvalues $(11.310, 0.109)$, closer to the original. This shows CQ is *more robust for ill-conditioned matrices*, mitigating instability and preserving spectral properties better than VQ.

C.2. Matrix Distance

For the Frobenius norm relative error (NRE) and angle error (AE) in Tab. 1, we report the cumulative errors over all preconditioners. For synthetic matrices, we randomly generate 100 instances of A via spectral decomposition to assess quantization robustness. Specifically, we construct A as:

$$A = U\Lambda U^{\top},$$

where U is a randomly sampled orthogonal matrix obtained via QR decomposition of a Gaussian random matrix, and Λ is a diagonal matrix with eigenvalues geometrically spaced from 10^{-3} to 10^{3}. This setup ensures a high dynamic range, making small values more susceptible to quantization errors, which are further amplified during inverse 1/4-th root computations.

Table 10. Comparison of VQ versus CQ on a toy 2×2 matrix L.

Method	Original	VQ	CQ
L	$\begin{bmatrix} 10.00 & 3.00 \\ 3.00 & 1.00 \end{bmatrix}$	$\begin{bmatrix} 10.00 & 3.60 \\ 3.60 & 1.11 \end{bmatrix}$	$\begin{bmatrix} 10.00 & 3.60 \\ 3.60 & 1.42 \end{bmatrix}$
Eigenvalues	$(10.908, 0.092)$	$(11.275, -0.164)$	$(11.310, 0.109)$

Additionally, we evaluate NRE and AE on preconditioners from 32-bit Shampoo training of Swin-Tiny on CIFAR-100. The results, summarized in Tab. 11, show that Cholesky quantization consistently reduces both NRE and AE compared to vanilla quantization, demonstrating its effectiveness in preserving spectral properties.

Table 11. NRE and AE on preconditioners of Swin-Tiny for vanilla quantization (VQ) and Cholesky quantization (CQ).

Preconditioner	VQ		CQ	
	NRE	AE	NRE	AE
Epoch 25	36.669	29.669	9.381	9.344
Epoch 50	36.853	29.269	8.803	8.775
Epoch 75	39.494	30.686	8.814	8.804
Epoch 100	41.068	30.848	8.943	8.918

C.3. Training Hyperparameters

For the first-order base optimizers SGDM and AdamW used in Shampoo, we maintain their optimizer states at the same precision as the model parameters, which is float32 for image classification and bfloat16 for LLM pre-training.

For SGDM, we set the initial learning rate to 0.1, the momentum parameter to 0.9, and the weight decay coefficient to 5×10^{-4} for training CNNs on CIFAR-100 and Tiny-ImageNet, and 1×10^{-4} for training ResNet-50 on ImageNet. For AdamW, we set the initial learning rate to 1×10^{-3}, the momentum parameters to $\beta_1 = 0.9$ and $\beta_2 = 0.999$, the small positive constant for the denominator to 1×10^{-8}, and the weight decay to 5×10^{-2} for image classification and 0 for LLM pre-training.

For quantization settings, we employ block-wise linear-2 quantization as introduced in Sec. 3.2, with a block size of $B \times B = 64 \times 64$. For tensors with fewer than 4096 elements, quantization is not applied.

For both 32-bit and 4-bit Shampoo, we set the small positive constant $\epsilon = 1 \times 10^{-6}$ and the preconditioner mo-

Table 12. Hyperparameters of LLaMA models for evaluation. Data amount are specified in tokens.

Params	Hidden	Intermediate	Heads	Layers
130M	768	2048	12	12
350M	1024	2736	16	24
1 B	2048	5461	24	32

mentum parameter $\beta = 0.95$. The error state momentum parameter is set to $\beta_e = 0.95$ to align with the preconditioner update. For update intervals, we use $T_1 = 100$ and $T_2 = 500$ for experiments on CIFAR-100 and Tiny-ImageNet, $T_1 = 200$ and $T_2 = 1000$ for training ResNet-50 on ImageNet, and $T_1 = T_2 = 200$ for LLM pre-training. Additionally, Shampoo applies layer-wise preconditioning to blocks derived from large matrices, with the maximum order of the preconditioner set to 1200.

For image classification tasks, we use the traditional cross-entropy loss as the training loss. For the learning rate schedule, we employ cosine annealing with 5 epochs of linear warmup across all experiments. For data augmentation, we apply horizontal flip, random crop, and color jitter for VGG and ResNet [23, 28], and Mixup [62], CutMix [61], RandomErasing [65], and RandAugment/AutoAugment [10, 11] for Swin and ViT [31, 34].

The batch size is set to 128 for experiments on CIFAR-100 and Tiny-ImageNet, 256 for training ResNet-50 on ImageNet, and 512 for training ViT-Base on ImageNet. For the total training epochs, we follow [23, 58] and train Shampoo with SGDM as the base optimizer for 200 epochs when training CNNs on CIFAR-100, while SGDM itself is trained for 300 epochs on CIFAR-100. For training CNNs on Tiny-ImageNet and ViTs on CIFAR-100 and Tiny-ImageNet, we follow [31, 34] and train Shampoo with the base optimizer for 100 epochs, and the base optimizer itself for 150 epochs. For training ResNet-50 on ImageNet, we train Shampoo with SGDM as the base optimizer for 100 epochs and SGDM for 120 epochs. For training ViT-Base on ImageNet, we train Shampoo with AdamW as the base optimizer for 120 epochs and AdamW for 150 epochs.

For LLM pre-training, we follow the model settings of [33, 64], with details provided in Tab. 12. All experiments use bfloat16 to reduce memory usage. Due to limited computational resources, we shorten training and run 10K steps for LLaMA-130M and LLaMA-350M, and 2K steps for LLaMA-1B. The total effective batch size per training step is 512 with gradient accumulation. The per-iteration batch size is set to 256 for LLaMA-130M, 128 for and LLaMA-350M, and 64 for LLaMA-1B.

C.4. Memory Efficiency

In our experiments, we report the peak GPU memory usage instead of the memory used solely by the optimizers, as the peak GPU memory usage is the primary constraint when training large-scale models in practice and is therefore our main concern. Furthermore, from the total peak GPU memory usage, we can deduce the additional memory cost introduced by the preconditioners of Shampoo on top of the base optimizers.

For instance, when training ResNet-34 on CIFAR-100, the base optimizer SGDM incurs a peak memory usage of 1254.7 MB. The additional peak GPU memory usage caused by storing the 32-bit preconditioners of Shampoo $(L_k, R_k, L_k^{-1/4}, R_k^{-1/4})$ is calculated as the peak memory usage of 32-bit Shampoo minus 1254.7 MB, which equals 627.9 MB. With vanilla 4-bit quantization for the preconditioners, this additional memory usage drops to 86.3 MB, which is less than $1/7$ of the additional memory required by 32-bit Shampoo. Furthermore, when using 4-bit Shampoo with Cholesky quantization, the additional peak memory usage decreases further to 64.8 MB.

We now provide a brief analysis of why the increased peak memory usage of 4-bit Shampoo with Cholesky quantization (e.g., 64.8 MB) is approximately 75% of that of vanilla 4-bit Shampoo (e.g., 86.3 MB). Vanilla 4-bit Shampoo stores the 4-bit preconditioners $(L_k, R_k, L_k^{-1/4}, R_k^{-1/4})$, as introduced in Sec. 4.1, which consist of four full matrices of the same shape in 4-bit precision. In contrast, 4-bit Shampoo with Cholesky quantization stores $(C_k^L, C_k^R, L_k^{-1/4}, R_k^{-1/4})$ as described in Sec. 4.2, where C_k^L and C_k^R are the lower triangular Cholesky factors of L_k and R_k, respectively. The storage of C_k^L and C_k^R requires only half the space of L_k and R_k, leading to the total storage cost of the preconditioners for 4-bit Shampoo with Cholesky quantization being approximately 75% of that of vanilla 4-bit Shampoo.

For $L_k^{-1/4}$ and $R_k^{-1/4}$, Cholesky quantization is not applied, as they are used to precondition stochastic gradients at each iteration, as described in Algorithm 2 and Algorithm 1. Restoring them from their Cholesky factors at each iteration would be computationally expensive.

Moreover, to analyze accuracy-memory trade-off, we experiment with varying preconditioner precisions on CIFAR-100. Results below show that 4-bit achieves a good balance, maintaining accuracy close to higher precision with significantly lower memory usage.

Table 13. Test accuracy (%) and peak GPU memory (MB) on CIFAR-100 with different preconditioner precisions.

Model	Base		4-bit		8-bit		32-bit	
Metric	Acc.	Mem.	Acc.	Mem.	Acc.	Mem.	Acc.	Mem.
VGG	74.43	597.3	75.21	662.2	75.63	727.1	75.02	1065.2
ResNet	79.12	1254.7	80.52	1341.0	80.67	1435.4	80.69	1882.6