
Supplemental Material: Compress Large Language Models via Collaboration Between Learning and Matrix Approximation

This appendix can be divided into 4 parts. To be precise,

1. Section A provides the experimental configurations.
2. Section B provides more detailed experimental results.
3. Section C describes the computational procedures of the individual operators.
4. Section D provides the proofs required for the theorems.

A Experimental Configurations.

A.1 Platform and Hyperparameters

Platform. All experiments were conducted using PyTorch on a single 80GB NVIDIA A100 GPU.

Experimental Hyperparameters. The training hyperparameters for Phi-3-mini pruning are specified in Table 6, and Table 7 details the configuration parameters of the truncated Gaussian distribution.

Table 6: Experimental hyperparameters of Phi-3-mini. Other models configurations are the same.

Train Set	Batch Size	Sequence Length	Optimizer	Learning Rate	inner_iters	Test Sequence Length
C4	32	256	AdamW	2e-3	80(20)	2048

Table 7: Hyperparameters of truncated Gaussian distribution.

Init. μ_κ	σ^2	γ_0	γ_{target}	$\xi_{warm\ up}$	ξ_{anneal}
0.25	5e-2	5e-2	5e-3	0.1	0.8

A.2 Other Details

Initialization. For μ_s , we employ uniform initialization ($\mu_s = \rho$) at low sparsity levels ($\leq 50\%$), as this already achieves satisfactory performance without significant degradation. At higher sparsity levels ($\geq 60\%$), we adopt OWL initialization for improved results, however, uniform initialization remains competitive after bilevel optimization. For μ_κ , a fixed uniform initialization (e.g., 0.25 in our experiments) is applied across all sparsity level. Hyperparameters’ setting are shown in Table 7. Following prior works [8, 31, 44], we use C4 [27] as the training dataset.

Final Return. Although the sampling has already converged with a sufficiently small γ , we return the final parameters $\mu = (\mu_s, \mu_\kappa)$ as deterministic results to for reproducibility.

γ decay. Following prior studies on hyperparameter annealing [50], we apply a cubic annealing schedule to gradually reduce γ from γ_0 to γ_{target} . Annealing process is defined as:

$$\gamma_t = \begin{cases} \gamma_0, & \text{if } t \leq T_{warm\ up} \\ \gamma_{target} + (\gamma_0 - \gamma_{target}) \cdot \left(1 - \frac{t - T_{warm\ up}}{T_{anneal} - T_{warm\ up}}\right)^3, & \text{if } T_{warm\ up} < t \leq T_{anneal} \\ \gamma_{target}, & \text{if } t > T_{anneal} \end{cases}$$

where t is the current iteration and $T_{warm\ up} = \xi_{warm\ up} \times T_{outer}$ is the number of warm up stage and $T_{anneal} = \xi_{anneal} \times T_{outer}$ is the total number of annealing steps. This schedule ensures a smooth and stable transition, preventing abrupt changes that may destabilize training.

B More Experimental Results.

B.1 Observation from Phi-3 Mini

Figure 5 displays the singular value distributions for layers 0 and 10 of the Phi-3 Mini. Right: Ratio of singular values needed to capture 90% energy for each matrix, which is defined as: $k_{90\%}/d$, with $k_{90\%} = \min\{k \mid \sum_{i=1}^k \sigma_i \geq 0.9 \times \sum_{j=1}^d \sigma_j\}$ (σ_i : descending-ordered singular values; d : full rank)

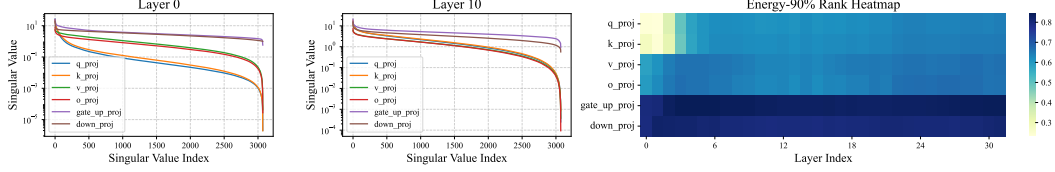


Figure 5: Left & Middle: Singular value spectra of Layer 0 and Layer 10 (log-scale y-axis). Right: The proportion of singular values required to capture 90% of the total energy

B.2 Detailed Results of zero-shot tasks

Table 8 presents the detailed zero-shot performance of various models across different sparsity levels. We incorporate selected results from the OATS benchmark study [44] for comparative analysis.

Table 8: Specific zero-shot Results.

Model	Compression	Method	PIQA	HellaSwag	WinoGrande	OpenBookQA	RTE	BoolQ	ARC-e	ARC-c
Phi-3 Mini	0%	Dense	81.23	77.50	73.56	46.80	75.81	85.32	78.45	57.25
		SparseGPT	78.94	76.94	69.85	49.60	73.29	84.13	76.39	55.89
	30%	Wanda	79.65	76.27	71.59	48.00	73.65	83.70	77.23	55.20
		OATS	80.03	77.07	72.61	47.60	74.37	84.92	77.44	57.76
		QR	80.52	76.52	72.93	48.0	76.17	85.11	77.86	57.35
		Ours	80.69	77.18	73.07	49.80	76.28	85.25	78.06	57.95
	40%	SparseGPT	78.35	75.07	68.59	47.00	72.20	83.67	75.29	53.24
		Wanda	78.35	73.87	69.30	45.40	71.84	83.18	76.52	51.96
		OATS	79.38	75.86	70.01	46.60	72.56	83.98	76.85	55.12
		QR	79.60	75.34	70.48	47.80	72.56	83.70	79.67	56.66
		Ours	79.77	75.97	70.62	48.00	72.67	84.12	79.87	56.85
	50%	SparseGPT	77.20	70.63	66.46	45.20	70.76	83.06	70.58	47.01
		Wanda	76.33	67.70	66.38	41.80	66.43	81.83	72.43	47.35
		OATS	77.26	71.64	69.53	44.80	73.65	81.28	77.10	52.05
		QR	76.28	70.77	69.54	45.60	74.09	82.51	76.46	51.29
		Ours	77.78	71.87	69.91	46.0	74.66	83.57	77.74	52.39
Llama-2 13B	0%	Dense	81.66	82.83	75.85	50.00	77.62	88.17	78.41	59.64
		SparseGPT	80.09	78.97	72.45	46.80	62.45	80.83	76.05	49.15
	30%	wanda	80.03	79.20	72.14	46.20	62.09	80.31	76.14	49.74
		OATS	79.98	79.31	72.06	46.20	64.98	80.95	76.56	50.09
		QR	79.98	78.81	71.90	47.00	61.37	80.86	76.47	50.51
		Ours	80.28	79.44	72.67	47.20	65.19	81.13	76.79	50.75
	40%	SparseGPT	79.71	77.65	71.19	45.40	67.15	80.64	73.74	47.61
		wanda	79.71	78.42	71.98	47.20	64.26	80.98	74.03	47.87
		OATS	79.71	78.29	72.85	45.60	63.18	80.09	76.05	49.49
		QR	79.87	78.53	71.11	46.20	65.34	80.89	76.94	49.91
		Ours	80.04	78.64	72.99	47.40	67.26	81.12	77.14	50.10
	50%	SparseGPT	79.00	75.58	72.45	44.60	63.90	81.53	71.38	44.97
		wanda	79.05	76.25	71.03	44.60	61.73	80.73	71.34	45.48
		OATS	78.78	76.23	72.69	44.00	68.95	80.98	73.19	47.35
		QR	79.00	76.92	72.85	43.80	67.15	80.61	72.73	48.12
		Ours	79.22	77.03	72.99	44.80	69.06	81.67	73.39	48.31
Llama-3 8B	0%	Dense	80.74	79.16	73.40	45.00	67.87	80.98	77.69	53.50
		SparseGPT	80.36	78.58	73.24	44.40	66.79	81.38	76.81	51.11
	30%	Wanda	79.98	78.00	73.64	44.40	64.26	81.62	76.18	50.94
		OATS	80.03	78.75	73.64	45.20	66.06	81.13	76.94	52.99
		QR	79.98	78.95	72.69	44.80	68.95	82.57	78.11	53.92
		Ours	80.53	79.06	73.78	45.40	69.06	82.71	78.31	54.11
	40%	SparseGPT	79.16	76.74	73.32	41.80	64.26	81.31	74.71	49.32
		Wanda	78.73	75.90	72.22	44.40	63.18	80.46	72.31	49.15
		OATS	79.71	77.18	74.19	43.80	67.51	82.39	74.92	49.74
		QR	78.78	77.01	73.56	42.20	66.43	80.40	76.64	51.28
		Ours	79.88	77.29	74.33	44.60	67.62	82.53	76.84	51.47
	50%	SparseGPT	77.58	73.12	72.85	40.80	59.21	79.30	69.28	45.14
		Wanda	77.53	69.34	70.24	40.00	61.73	76.57	66.96	43.77
		OATS	77.75	73.17	71.74	41.00	64.98	79.66	72.35	45.05
		QR	77.58	72.39	69.06	41.60	55.96	75.93	70.37	41.81
		Ours	78.12	73.26	73.07	41.90	65.47	80.04	72.88	45.42

B.3 More Experiments with OWL Allocation

We evaluate the generalization ability of different sparsity allocation strategies across multiple models. Specifically, we apply the sparsity configurations produced by our method and by OWL to compress models—Phi-3 mini, Llama2-7B, and Llama3-8B. We report the perplexity on the WikiText2 dataset as the evaluation metric, and present the results in the Figure 6.

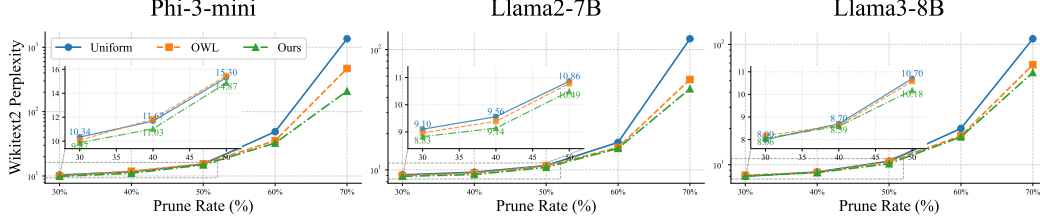


Figure 6: Comparison with OWL Allocation.

C Computation of Operators.

C.1 Projection

Following [48], we denote the feasible region of μ_s in problem (4.1) as \mathcal{C} , that is $\mathcal{C} = \{\mu_s : \|\mu_s\|_1 \geq \rho \times L \text{ and } \mu_s \in [0, 1]^L\}$. The theorem below shows that the projection of a vector onto \mathcal{C} can be calculated efficiently, which makes the sparsity constraint would always be satisfied during our training process.

Theorem 1. For each vector z , its projection s in the set \mathcal{C} can be calculated as follows:

$$\mu = \max(0, \min(1, v_2^* \mathbf{1} + \mathbf{1} - z)), \quad (14)$$

where $v_2^* = \max(0, v_1^*)$ with v_1^* being the solution of the following equation

$$\mathbf{1}^\top [\min(1, \max(0, z - v_1^* \mathbf{1}))] - (1 - \rho)K = 0. \quad (15)$$

The Eqn.(15) can be solved by bisection method efficiently. Now we can apply policy gradient descent (PGD) to solve problem 4.1 directly on probability space with explicit sparsity constraint. Theorem 1 is a special case of the problem proposed in [36]. A detailed proof is not the main focus of this paper and can be found in [48].

C.2 Inverse Transform Sampling for Truncated Gaussian Distribution

In this section, we describe the inverse transform sampling process for obtaining samples from a truncated Gaussian distribution. Given the truncated Gaussian probability density function (PDF):

$$p(x; \mu, \sigma, \gamma) = \begin{cases} \frac{1}{\sigma} \cdot \frac{\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{\gamma}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right)}, & \text{for } x \in [\mu - \gamma, \mu + \gamma], \\ 0, & \text{otherwise,} \end{cases}$$

we aim to generate samples from this distribution using the inverse transform sampling technique.

The CDF of this distribution is given by:

$$F(x) = \int_{-\infty}^x p(t; \mu, \sigma, \gamma) dt = \frac{\Phi\left(\frac{x-\mu}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right)}{\Phi\left(\frac{\gamma}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right)} \quad \text{for } x \in [\mu - \gamma, \mu + \gamma].$$

Given a uniform random variable $u \in [0, 1]$, we want to find x such that $F(x) = u$.

From the CDF expression, we can write:

$$\frac{\Phi\left(\frac{x-\mu}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right)}{\Phi\left(\frac{\gamma}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right)} = u.$$

Solving for $\Phi\left(\frac{x-\mu}{\sigma}\right)$, we get:

$$\Phi\left(\frac{x-\mu}{\sigma}\right) = u \cdot \left[\Phi\left(\frac{\gamma}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right) \right] + \Phi\left(\frac{-\gamma}{\sigma}\right).$$

Now, applying the inverse of the standard normal CDF, Φ^{-1} , we obtain:

$$\frac{x-\mu}{\sigma} = \Phi^{-1} \left(u \cdot \left[\Phi\left(\frac{\gamma}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right) \right] + \Phi\left(\frac{-\gamma}{\sigma}\right) \right).$$

Thus, we can solve for x as:

$$x = \mu + \sigma \cdot \Phi^{-1} \left(u \cdot \left[\Phi\left(\frac{\gamma}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right) \right] + \Phi\left(\frac{-\gamma}{\sigma}\right) \right).$$

To generate samples from the truncated Gaussian distribution, we can now follow these steps:

1. Sample a uniform random variable $u \in [0, 1]$.
2. Compute the value of x using the inverse CDF derived above.

This procedure ensures that the generated samples follow the exact truncated Gaussian distribution.

The inverse transform sampling method guarantees accurate sampling from a truncated Gaussian distribution by applying the inverse of the CDF. This method effectively maps uniformly distributed random numbers to the desired truncated Gaussian distribution, ensuring that the samples are drawn according to the correct statistical properties.

In practice, we can implement this sampling process efficiently by using standard functions for the normal CDF and its inverse.

C.3 QR-based RPCA

The pseudocode of our efficient RPCA algorithm based on QR decomposition is provided below.

Algorithm 2 Efficient QR-based RPCA

Require: Input matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$, target rank r , sparsity budget K , max iteration T

- 1: Initialize sparse component $\mathbf{S}_0 = \mathbf{0}$ and \mathbf{V}_0 with i.i.d. entries: $[\mathbf{V}_0]_{ij} \sim \mathcal{N}(0, \sigma^2)$
 - 2: **for** $t = 0$ to $T_{inner} - 1$ **do**
 - 3: Compute $\mathbf{A}_t = (\mathbf{W} - \mathbf{S}_t)\mathbf{V}_t^\top$
 - 4: QR decomposition: $\mathbf{A}_t = \mathbf{Q}_t\mathbf{R}_t$
 - 5: $\mathbf{U}_{t+1} \leftarrow \mathbf{Q}_t$
 - 6: $\mathbf{V}_{t+1} \leftarrow \mathbf{Q}_t^\top(\mathbf{W} - \mathbf{S}_t)$
 - 7: $\mathbf{L}_{t+1} \leftarrow \mathbf{U}_{t+1}\mathbf{V}_{t+1}$
 - 8: $\mathbf{S}_{t+1} \leftarrow \mathcal{P}_\omega(\mathbf{W} - \mathbf{L}_{t+1})$
 - 9: **end for**
 - 10: **return** Low-rank component $\mathbf{L}_T = \mathbf{U}_T\mathbf{V}_T$, sparse component \mathbf{S}_T
-

Alternatively, we can initialize using the previous results \mathbf{V}_T and \mathbf{S}_T . That is, we initialize $\mathbf{S}_0 = \mathbf{S}_T$. And for \mathbf{V}_T , if target rank r is greater than the former rank r' ($\Delta r = r - r'$), we initialize $\mathbf{V}_0 = [\mathbf{V}_T, \Delta\mathbf{V}]$ where $\Delta\mathbf{V} \in \mathbb{R}^{\Delta r \times n}$ with i.i.d. entries: $[\Delta\mathbf{V}]_{ij} \sim \mathcal{N}(0, \sigma^2)$. If $r < r'$ ($\Delta r = r' - r$), then we truncate the last Δr rows of \mathbf{V}_T , resulting in $\mathbf{V}_0 = \mathbf{V}_T[r, :] \in \mathbb{R}^{r \times n}$.

D Proofs

D.1 Policy Gradient

To simplify the derivation, we denote $\mathbf{x} = (\mathbf{s}, \boldsymbol{\kappa})$. The detailed derivation of the policy gradient estimation is as follows:

$$\begin{aligned}
& \nabla_{\boldsymbol{\mu}} \mathbb{E}_{p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)} [\mathcal{L}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa}))] = \nabla_{\boldsymbol{\mu}} \mathbb{E}_{p(\mathbf{x} | \boldsymbol{\mu}, \sigma^2, \gamma)} \mathcal{L}(\mathcal{W}(\mathbf{x})) \\
&= \nabla_{\boldsymbol{\mu}} \int p(\mathbf{x} | \boldsymbol{\mu}, \sigma^2, \gamma) \mathcal{L}(\mathcal{W}(\mathbf{x})) d\mathbf{x} = \int \mathcal{L}(\mathcal{W}(\mathbf{x})) \nabla_{\boldsymbol{\mu}} p(\mathbf{x} | \boldsymbol{\mu}, \sigma^2, \gamma) + \underbrace{p(\mathbf{x} | \boldsymbol{\mu}, \sigma^2, \gamma) \nabla_{\boldsymbol{\mu}} \mathcal{L}(\mathcal{W}(\mathbf{x}))}_{=0} d\mathbf{x} \\
&= \int \mathcal{L}(\mathbf{x}) p(\mathbf{x} | \boldsymbol{\mu}, \sigma^2, \gamma) \nabla_{\boldsymbol{\mu}} \log(p(\mathbf{x} | \boldsymbol{\mu}, \sigma^2, \gamma)) d\mathbf{x} = \mathbb{E}_{p(\mathbf{x} | \boldsymbol{\mu}, \sigma^2, \gamma)} [\mathcal{L}(\mathbf{x}) \nabla_{\boldsymbol{\mu}} \log(p(\mathbf{x} | \boldsymbol{\mu}, \sigma^2, \gamma))] \\
&= \mathbb{E}_{p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)} [\mathcal{L}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa})) \cdot \nabla_{\boldsymbol{\mu}} \log p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)].
\end{aligned}$$

D.2 Unbiasedness

Lemma 2. Suppose $(\mathbf{s}', \boldsymbol{\kappa}')$ are sampled independently from the truncated Gaussian distribution $p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)$, then $\mathbf{g}_{\boldsymbol{\mu}}^{vr}$ is an unbiased stochastic gradient of $\mathbb{E}_{p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)} [\mathcal{L}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa}))]$.

Proof. of Lemma 2

We aim to show that the stochastic gradient

$$\mathbf{g}_{\boldsymbol{\mu}}^{vr} = [\mathcal{L}_{\mathcal{B}}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa})) - \mathcal{L}_{\mathcal{B}}(\mathcal{W}(\mathbf{s}', \boldsymbol{\kappa}'))] \nabla_{\boldsymbol{\mu}, \sigma^2, \gamma} \log p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma),$$

is an unbiased estimator of $\nabla_{\boldsymbol{\mu}, \sigma^2, \gamma} \mathbb{E}_{p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu})} [\mathcal{L}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa}))]$, where \mathcal{B} is uniformly sampled from the training dataset, and $(\mathbf{s}', \boldsymbol{\kappa}') \sim p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu})$ independently.

The expectation is

$$\mathbb{E}[\mathbf{g}_{\boldsymbol{\mu}}^{vr}] = \mathbb{E}_{\mathcal{B}, \mathbf{s}, \boldsymbol{\kappa}, \mathbf{s}', \boldsymbol{\kappa}'} [(\mathcal{L}_{\mathcal{B}}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa})) - \mathcal{L}_{\mathcal{B}}(\mathcal{W}(\mathbf{s}', \boldsymbol{\kappa}')))] \nabla_{\boldsymbol{\mu}} \log p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma).$$

By linearity, this is

$$\mathbb{E}_{\mathcal{B}, \mathbf{s}, \boldsymbol{\kappa}} [\mathcal{L}_{\mathcal{B}}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa})) \nabla_{\boldsymbol{\mu}} \log p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)] - \mathbb{E}_{\mathcal{B}, \mathbf{s}', \boldsymbol{\kappa}'} [\mathcal{L}_{\mathcal{B}}(\mathcal{W}(\mathbf{s}', \boldsymbol{\kappa}'))] \mathbb{E}_{\mathbf{s}, \boldsymbol{\kappa}} [\nabla_{\boldsymbol{\mu}} \log p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)].$$

For the first term, let $f(\mathbf{s}, \boldsymbol{\kappa}) = \mathcal{L}_{\mathcal{B}}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa}))$. Due to $\nabla_{\boldsymbol{\mu}} \log p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma) = \frac{\nabla_{\boldsymbol{\mu}} p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)}{p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)}$, we have:

$$\mathbb{E}_{\mathbf{s}, \boldsymbol{\kappa}} [f(\mathbf{s}, \boldsymbol{\kappa}) \nabla_{\boldsymbol{\mu}} \log p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)] = \int f(\mathbf{s}, \boldsymbol{\kappa}) \nabla_{\boldsymbol{\mu}} p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma) d\mathbf{s} d\boldsymbol{\kappa}.$$

This equals

$$\nabla_{\boldsymbol{\mu}} \int f(\mathbf{s}, \boldsymbol{\kappa}) p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma) d\mathbf{s} d\boldsymbol{\kappa} = \nabla_{\boldsymbol{\mu}} \mathbb{E}_{\mathbf{s}, \boldsymbol{\kappa}} [f(\mathbf{s}, \boldsymbol{\kappa})].$$

Thus, the first term is

$$\mathbb{E}_{\mathcal{B}} [\nabla_{\boldsymbol{\mu}} \mathbb{E}_{\mathbf{s}, \boldsymbol{\kappa}} [\mathcal{L}_{\mathcal{B}}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa}))]].$$

For the second term, compute:

$$\mathbb{E}_{\mathbf{s}, \boldsymbol{\kappa}} [\nabla_{\boldsymbol{\mu}} \log p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)] = \int \nabla_{\boldsymbol{\mu}} p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma) d\mathbf{s} d\boldsymbol{\kappa} = \nabla_{\boldsymbol{\mu}} \int p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma) d\mathbf{s} d\boldsymbol{\kappa} = 0.$$

Thus, the second term is zero.

Combining, we get:

$$\mathbb{E}[\mathbf{g}_{\boldsymbol{\mu}}^{vr}] = \mathbb{E}_{\mathcal{B}} [\nabla_{\boldsymbol{\mu}} \mathbb{E}_{\mathbf{s}, \boldsymbol{\kappa}} [\mathcal{L}_{\mathcal{B}}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa}))]].$$

Since \mathcal{B} is uniformly sampled, $\mathbb{E}_{\mathcal{B}} [\mathcal{L}_{\mathcal{B}}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa}))] = \mathbb{E}_{\mathbf{s}, \boldsymbol{\kappa}} [\mathcal{L}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa}))]$. Hence:

$$\mathbb{E}[\mathbf{g}_{\boldsymbol{\mu}}^{vr}] = \nabla_{\boldsymbol{\mu}} \mathbb{E}_{p(\mathbf{s}, \boldsymbol{\kappa} | \boldsymbol{\mu}, \sigma^2, \gamma)} [\mathcal{L}(\mathcal{W}(\mathbf{s}, \boldsymbol{\kappa}))].$$

Thus, $\mathbf{g}_{\boldsymbol{\mu}}^{vr}$ is unbiased. \square

D.3 Bounded Variance

Theorem 2. Suppose (s', κ') are sampled independently from the truncated Gaussian distribution $p(s, \kappa | \mu, \sigma^2, \gamma)$ and $|\mathcal{L}_B(\mathcal{W}(s, \kappa))| \leq L$, then the variance of \mathbf{g}_μ^{vr} is bounded.

Proof. of Theorem 2

We need to prove that the variance of the stochastic gradient

$$\mathbf{g}_\mu^{vr} = [\mathcal{L}_B(\mathcal{W}(s, \kappa)) - \mathcal{L}_B(\mathcal{W}(s', \kappa'))] \nabla_\mu \log p(s, \kappa | \mu, \sigma^2, \gamma),$$

is bounded, i.e., $\text{Var}(\mathbf{g}_\mu^{vr}) = \mathbb{E}[\|\mathbf{g}_\mu^{vr} - \mathbb{E}[\mathbf{g}_\mu^{vr}]\|^2] < \infty$, where \mathcal{B} is sampled from the training dataset, and $(s, \kappa), (s', \kappa') \sim p(s, \kappa | \mu)$ independently from a truncated Gaussian distribution.

The variance of $\mathbf{g}_{s, \kappa}^{vr}$ equals to

$$\text{Var}(\mathbf{g}_\mu^{vr}) = \mathbb{E}[\|\mathbf{g}_\mu^{vr}\|^2] - \|\mathbb{E}[\mathbf{g}_\mu^{vr}]\|^2.$$

Since $\|\mathbb{E}[\mathbf{g}_\mu^{vr}]\|^2$ is finite, it suffices to show that $\mathbb{E}[\|\mathbf{g}_\mu^{vr}\|^2] < \infty$.

We first define the loss difference as

$$\Delta \mathcal{L} = \mathcal{L}_B(\mathcal{W}(s, \kappa)) - \mathcal{L}_B(\mathcal{W}(s', \kappa')).$$

Then,

$$\mathbf{g}_\mu^{vr} = \Delta \mathcal{L} \cdot \nabla_\mu \log p(s, \kappa | \mu, \sigma^2, \gamma),$$

and

$$\|\mathbf{g}_\mu^{vr}\|^2 = \Delta \mathcal{L}^2 \cdot \|\nabla_\mu \log p(s, \kappa | \mu, \sigma^2, \gamma)\|^2.$$

Thus, we need to bound

$$\mathbb{E}[\|\mathbf{g}_\mu^{vr}\|^2] = \mathbb{E}_{\mathcal{B}, s, \kappa, s', \kappa'} [\Delta \mathcal{L}^2 \cdot \|\nabla_\mu \log p(s, \kappa | \mu, \sigma^2, \gamma)\|^2].$$

Then, we have

$$|\Delta \mathcal{L}| \leq |\mathcal{L}_B(\mathcal{W}(s, \kappa))| + |\mathcal{L}_B(\mathcal{W}(s', \kappa'))| \leq 2L,$$

so

$$\Delta \mathcal{L}^2 \leq 4L^2.$$

Thus,

$$\mathbb{E}[\|\mathbf{g}_\mu^{vr}\|^2] \leq 4L^2 \cdot \mathbb{E}_{s, \kappa} [\|\nabla_\mu \log p(s, \kappa | \mu, \sigma^2, \gamma)\|^2].$$

Let $\mathbf{x} = (s, \kappa) \in \mathbb{R}^d$ have components $x_i \sim \mathcal{N}_\gamma(\mu_i, \sigma^2)$ independently, with PDF

$$p(x_i; \mu_i, \sigma, \gamma) = \frac{1}{\sigma} \cdot \frac{\phi\left(\frac{x_i - \mu_i}{\sigma}\right)}{\Phi\left(\frac{\gamma}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right)}, \quad x_i \in [\mu_i - \gamma, \mu_i + \gamma].$$

So $p(\mathbf{x} | \mu, \sigma^2, \gamma) = \prod_{i=1}^d p(x_i; \mu_i, \sigma, \gamma)$. By Lemma 1, $\nabla_{\mu_i} \log p(x_i; \mu_i, \sigma, \gamma) = \frac{x_i - \mu_i}{\sigma^2}$, we have

$$\nabla_\mu \log p(\mathbf{x} | \mu, \sigma^2, \gamma) = \frac{1}{\sigma^2} (\mathbf{x} - \mu),$$

and

$$\|\nabla_\mu \log p(\mathbf{x} | \mu, \sigma^2, \gamma)\|^2 = \frac{1}{\sigma^4} \sum_{i=1}^d (x_i - \mu_i)^2.$$

Thus,

$$\mathbb{E}_{\mathbf{x}} [\|\nabla_\mu \log p(\mathbf{x} | \mu, \sigma^2, \gamma)\|^2] = \frac{1}{\sigma^4} \sum_{i=1}^d \mathbb{E} [(x_i - \mu_i)^2].$$

For $x_i \sim \mathcal{N}_\gamma(\mu_i, \sigma^2)$, since truncation is symmetric, $\mathbb{E}[x_i] = \mu_i$, and

$$\mathbb{E} [(x_i - \mu_i)^2] = \text{Var}(x_i).$$

The variance of x_i is

$$\text{Var}(x_i) = \sigma^2 \left[1 - \frac{2\phi\left(\frac{\gamma}{\sigma}\right) \cdot \frac{\gamma}{\sigma}}{\Phi\left(\frac{\gamma}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right)} \right].$$

Let $a = \frac{\gamma}{\sigma}$, so $\Phi\left(\frac{\gamma}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right) = 2\Phi(a) - 1$. Define

$$\beta(a) = 1 - \frac{\phi(a)a}{\Phi(a) - 0.5},$$

where $\phi(a) = \frac{1}{\sqrt{2\pi}}e^{-a^2/2}$. Thus,

$$\text{Var}(x_i) = \sigma^2 \beta(a), \quad a = \frac{\gamma}{\sigma}.$$

Then

$$\sum_{i=1}^d \mathbb{E}[(x_i - \mu_i)^2] = d\sigma^2 \beta\left(\frac{\gamma}{\sigma}\right).$$

Therefore,

$$\mathbb{E}_{\mathbf{x}}[\|\nabla_{\boldsymbol{\mu}} \log p(\mathbf{x}|\boldsymbol{\mu}, \sigma^2, \gamma)\|^2] = \frac{d\sigma^2 \beta\left(\frac{\gamma}{\sigma}\right)}{\sigma^4} = \frac{d\beta\left(\frac{\gamma}{\sigma}\right)}{\sigma^2}.$$

So,

$$\mathbb{E}[\|\mathbf{g}_{\boldsymbol{\mu}}^{vr}\|^2] \leq 4L^2 \cdot \frac{d\beta\left(\frac{\gamma}{\sigma}\right)}{\sigma^2} = \frac{4L^2 d\beta\left(\frac{\gamma}{\sigma}\right)}{\sigma^2},$$

which is bounded. \square

D.4 Convergence Analysis

For completeness, we restate the convergence result of the bilevel algorithm as presented in [47]. The theorem is included here to ensure the paper remains self-contained.

Theorem 3. *For bilevel problem with the form as:*

$$\min_{\theta \in \mathcal{C}} \Phi(\theta) = \mathbb{E}_{(s, \kappa) \sim p(\cdot|\theta)} \mathcal{L}(\mathcal{W}(s, \kappa)) \quad \text{with} \quad \mathcal{W}(s, \kappa) = \text{RPCA}(\mathcal{W}, s, \kappa).$$

We assume $\Phi(\theta)$ is L -smooth, and the policy gradient variance satisfies:

$$\mathbb{E} \|\mathcal{L}(\mathcal{W}(s, \kappa)) \nabla_{\theta} \log p(s, \kappa | \theta) - \nabla_{\theta} \Phi(\theta)\|^2 \leq \tilde{\sigma}^2.$$

Let $\eta < 1/L$ and define the gradient mapping \mathcal{G}^t at iteration t as:

$$\mathcal{G}^t = \frac{1}{\eta} \left(\theta^t - \mathcal{P}_{\mathcal{C}} \left(\theta^t - \eta \nabla_{\theta} \Phi(\theta^t) \right) \right),$$

then when $T \rightarrow \infty$, we have:

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\mathcal{G}^t\|^2 \leq \frac{8 - 2L\eta}{2 - L\eta} \tilde{\sigma}^2.$$

We need the following lemmas about the properties of the projection operator, which can be found in [11].

Lemma 3. (Firmly Nonexpansive Operators) *Let $\mathcal{C} \subset \mathbb{R}^d$ be compact and convex. For any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$:*

$$\|\mathcal{P}_{\mathcal{C}}(\mathbf{u}) - \mathcal{P}_{\mathcal{C}}(\mathbf{v})\|^2 \leq (\mathbf{u} - \mathbf{v})^\top (\mathcal{P}_{\mathcal{C}}(\mathbf{u}) - \mathcal{P}_{\mathcal{C}}(\mathbf{v})).$$

Lemma 4. *Let $\mathcal{C} \subset \mathbb{R}^d$ be compact and convex. Then for any $\mathbf{c} \in \mathcal{C}$ and $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$:*

$$\|\mathcal{P}_{\mathcal{C}}(\mathbf{c} + \mathbf{u}) - \mathcal{P}_{\mathcal{C}}(\mathbf{c} + \mathbf{v})\| \leq \|\mathbf{u} - \mathbf{v}\|.$$

Proof. of Theorem 3

We define:

$$\mathbf{g}^t = \mathcal{L}_{\mathcal{B}}(\mathcal{W}(s, \kappa)) \nabla_{\theta} \log p(s, \kappa | \theta^t).$$

Our method update θ as:

$$\theta^{t+1} = \mathcal{P}_{\mathcal{C}} (\theta^t - \eta \mathbf{g}^t).$$

Let the stochastic and deterministic gradient mappings be

$$\begin{aligned} \hat{\mathcal{G}}^t &= \frac{1}{\eta} (\theta^t - \mathcal{P}_{\mathcal{C}} (\theta^t - \eta \mathbf{g}^t)) = \frac{1}{\eta} (\theta^t - \theta^{t+1}), \\ \mathcal{G}^t &= \frac{1}{\eta} (\theta^t - \mathcal{P}_{\mathcal{C}} (\theta^t - \eta \nabla \Phi(\theta^t))), \end{aligned}$$

we can have

$$\begin{aligned} \Phi(\theta^{t+1}) &\leq \Phi(\theta^t) + \langle \nabla \Phi(\theta^t), \theta^{t+1} - \theta^t \rangle + \frac{L}{2} \|\theta^{t+1} - \theta^t\|^2 \\ &= \Phi(\theta^t) - \eta \langle \nabla \Phi(\theta^t), \hat{\mathcal{G}}^t \rangle + \frac{L\eta^2}{2} \|\hat{\mathcal{G}}^t\|^2 \\ &= \Phi(\theta^t) - \eta \langle \nabla \Phi(\theta^t) - \mathbf{g}^t + \mathbf{g}^t, \hat{\mathcal{G}}^t \rangle + \frac{L\eta^2}{2} \|\hat{\mathcal{G}}^t\|^2 \\ &= \Phi(\theta^t) - \eta \langle \mathbf{g}^t, \hat{\mathcal{G}}^t \rangle + \frac{L\eta^2}{2} \|\hat{\mathcal{G}}^t\|^2 + \eta \langle \delta^t, \hat{\mathcal{G}}^t \rangle \quad (\text{here } \delta^t = \mathbf{g}^t - \nabla \Phi(\theta^t)) \\ &\leq \Phi(\theta^t) - \eta \|\hat{\mathcal{G}}^t\|^2 + \frac{L\eta^2}{2} \|\hat{\mathcal{G}}^t\|^2 + \eta \langle \delta^t, \hat{\mathcal{G}}^t \rangle \quad (\text{Lemma 3}) \\ &\leq \Phi(\theta^t) - (\eta - \frac{L\eta^2}{2}) \|\hat{\mathcal{G}}^t\|^2 + \eta \langle \delta^t, \hat{\mathcal{G}}^t \rangle \\ &= \Phi(\theta^t) - (\eta - \frac{L\eta^2}{2}) \|\hat{\mathcal{G}}^t\|^2 + \eta \langle \delta^t, \mathcal{G}^t \rangle + \eta \langle \delta^t, \hat{\mathcal{G}}^t - \mathcal{G}^t \rangle \\ &\leq \Phi(\theta^t) - (\eta - \frac{L\eta^2}{2}) \|\hat{\mathcal{G}}^t\|^2 + \eta \langle \delta^t, \mathcal{G}^t \rangle + \eta \|\delta^t\| \|\hat{\mathcal{G}}^t - \mathcal{G}^t\| \\ &\leq \Phi(\theta^t) - (\eta - \frac{L\eta^2}{2}) \|\hat{\mathcal{G}}^t\|^2 + \eta \langle \delta^t, \mathcal{G}^t \rangle + \eta \|\delta^t\|^2 \quad (\text{Lemma 4}) \end{aligned}$$

Therefore, we can get

$$(\eta - \frac{L\eta^2}{2}) \|\hat{\mathcal{G}}^t\|^2 \leq \Phi(\theta^t) - \Phi(\theta^{t+1}) + \eta \langle \delta^t, \mathcal{G}^t \rangle + \eta \|\delta^t\|^2.$$

Thus, we obtain

$$\sum_{t=1}^T (\eta - \frac{L\eta^2}{2}) \|\hat{\mathcal{G}}^t\|^2 \leq \Phi(\theta^1) - \Phi(\theta^{T+1}) + \eta \sum_{t=1}^T (\langle \delta^t, \mathcal{G}^t \rangle + \|\delta^t\|^2). \quad (16)$$

Now, we turn to analyze the term $\langle \delta^t, \mathcal{G}^t \rangle$ as follows:

$$\mathbb{E} \langle \delta^t, \mathcal{G}^t \rangle = \mathbb{E}_{\theta^t} \mathbb{E}_{\cdot | \theta^t} (\langle \mathbf{g}^t - \nabla \Phi(\theta^t), \mathcal{G}^t \rangle | \theta^t) = 0. \quad (17)$$

For $\|\delta^t\|^2$, we have

$$\mathbb{E} \|\delta^t\|^2 = \mathbb{E} \|\mathbf{g}^t - \nabla \Phi(\theta^t)\|^2 \leq \tilde{\sigma}^2. \quad (18)$$

Combining inequalities (16), (17), and (18), we have

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\hat{\mathcal{G}}^t\|^2 \leq \frac{\Phi(\theta^1) - \Phi^*}{(1 - L\eta/2)T} + \frac{\tilde{\sigma}^2}{1 - L\eta/2}. \quad (19)$$

Finally, we bound $\mathbb{E} \|\mathcal{G}^t\|^2$ as follows:

$$\begin{aligned} \mathbb{E} \|\mathcal{G}^t\|^2 &\leq 2\mathbb{E} \|\hat{\mathcal{G}}^t\|^2 + 2\mathbb{E} \|\mathbf{g}^t - \nabla \Phi(\theta^t)\|^2 \\ &\leq 2\mathbb{E} \|\hat{\mathcal{G}}^t\|^2 + 2\tilde{\sigma}^2. \end{aligned} \quad (20)$$

Combining inequalities (19) and (20), as $T \rightarrow \infty$, we obtain

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\mathcal{G}^t\|^2 \leq \frac{2}{1 - L\eta/2} \left(\frac{\Phi(\theta^1) - \Phi^*}{T} + (2 - L\eta/2)\tilde{\sigma}^2 \right) \rightarrow \frac{8 - 2L\eta}{2 - L\eta} \tilde{\sigma}^2.$$

□

E Limitation

Similar to existing methods, our approach does not employ industrial-grade code optimization during inference evaluation. This may result in differences between the actual acceleration achieved in real-world industrial applications and our reported results. In particular, current mainstream GPU libraries lack fully optimized inference kernels that can exploit such hybrid sparsity structures. This limitation prevents us from fully realizing the potential speedup in practice. We expect that with future support from the community in developing tailored GPU operators for hybrid low-rank–sparse matrices, the practical acceleration of our approach could be further improved.

F Broader Impacts

Our work aims to reduce the computational cost of LLMs by combining learning-based optimization with matrix approximation techniques. By improving efficiency without significantly compromising performance, our method can make language models more accessible in resource-constrained settings, such as mobile devices or low-power servers. This may help democratize access to AI technologies. However, we also recognize that lowering the barrier to deployment could increase the risk of misuse, such as the spread of misinformation or harmful content. We encourage careful evaluation of compressed models in terms of fairness, safety, and robustness before real-world applications.