APPENDIX: MITIGATING ERROR PROPAGATION IN LOW-RANK APPROXIMATION OF LARGE MODELS VIA DISTRIBUTION-AWARE WHITENING

Anonymous authors
Paper under double-blind review

A LIMITATIONS

This work focuses on leveraging input distribution statistics to improve the approximation fidelity of low-rank methods. The proposed framework can serve as a drop-in replacement for conventional SVD used in model compression, as well as in SVD-based initialization strategies such as PiSSA for LoRA. While our method shows improvements across diverse settings, we acknowledge that, in line with the *No Free Lunch Theorem*, no single approach can universally dominate across all tasks and deployment scenarios. We outline several limitations of our method: 1) Compared to standard SVD, our method introduces additional computational overhead due to the per-layer whitening operations; 2) The performance may be influenced by the choice and distribution of the calibration data. In our experiments, we adopt the 256-sample subset of WikiText-2 as suggested by ASVD. However, identifying optimal or adaptive calibration strategies remains a non-trivial and open research direction; 3) Although we evaluate our method across eight diverse models with varying sizes and architectures, we are currently unable to conduct experiments on very large-scale models (e.g., 65B or 70B parameters) due to computational constraints. Nonetheless, our method is architecture-agnostic and theoretically scalable, and we will release our code to support further evaluation and broader adoption by the community.

B THE USE OF LARGE LANGEUAGE MODELS

In this work, a large language model was employed solely for language refinement, aiming to produce more fluent expressions and to avoid grammatical issues. The model was not used for content generation. The rapid development of large models provides non-native English speakers with a valuable tool to better present their work. We hope that our research can make a modest contribution along the fast-evolving path of large model development.

C PROOF OF WHITENED COMPRESSION ERROR EQUIVALENCE

By definition, the whitening matrix

$$S_\ell = \left(X_{\ell-1}X_{\ell-1}^\top + \epsilon I\right)^{-\frac{1}{2}} \tag{1}$$

transforms the original inputs $X_{\ell-1} \in \mathbb{R}^{d_{\ell-1}\times n}$ into whitened inputs

$$\tilde{X}_{\ell-1} = S_\ell X_{\ell-1}, \tag{2}$$

which satisfy (approximately)

$$\tilde{X}_{\ell-1}\tilde{X}_{\ell-1}^\top \approx I, \tag{3}$$

where n is the number of input samples and $I \in \mathbb{R}^{d_{\ell-1}\times d_{\ell-1}}$ is the identity matrix.

Correspondingly, the transformed weight matrix is defined as

$$\tilde{W}_\ell = W_\ell S_\ell^{-1}. \tag{4}$$

We perform low-rank compression on $\tilde{W}_\ell$ via truncated singular value decomposition (SVD), obtaining the best rank-k approximation

$$\tilde{W}_\ell^{(k)} = \arg\min_{\mathrm{rank}(\cdot)\le k} \|\tilde{W}_\ell - \cdot\|_F. \tag{5}$$

The compressed weight in the original space is then recovered as

$$\hat{W}_\ell = \tilde{W}_\ell^{(k)} S_\ell. \tag{6}$$

To analyze the output reconstruction error, consider

$$\|W_\ell X_{\ell-1} - \hat{W}_\ell X_{\ell-1}\|_F = \|W_\ell X_{\ell-1} - \tilde{W}_\ell^{(k)} S_\ell X_{\ell-1}\|_F. \tag{7}$$

Substituting $\tilde{X}_{\ell-1} = S_\ell X_{\ell-1}$ and $W_\ell = \tilde{W}_\ell S_\ell$, we rewrite this as

$$\|\tilde{W}_\ell \tilde{X}_{\ell-1} - \tilde{W}_\ell^{(k)} \tilde{X}_{\ell-1}\|_F = \|(\tilde{W}_\ell - \tilde{W}_\ell^{(k)})\tilde{X}_{\ell-1}\|_F. \tag{8}$$

Using the Frobenius norm property and the whitened input covariance, we have

$$\|(\tilde{W}_\ell - \tilde{W}_\ell^{(k)})\tilde{X}_{\ell-1}\|_F^2 = \operatorname{trace}\left(\tilde{X}_{\ell-1}^\top (\tilde{W}_\ell - \tilde{W}_\ell^{(k)})^\top (\tilde{W}_\ell - \tilde{W}_\ell^{(k)})\tilde{X}_{\ell-1}\right). \tag{9}$$

Since $\tilde{X}_{\ell-1}\tilde{X}_{\ell-1}^\top = I$, the above reduces to

$$\operatorname{trace}\left((\tilde{W}_\ell - \tilde{W}_\ell^{(k)})(\tilde{W}_\ell - \tilde{W}_\ell^{(k)})^\top\right) = \|\tilde{W}_\ell - \tilde{W}_\ell^{(k)}\|_F^2. \tag{10}$$

By the Eckart-Young theorem, the truncated $\tilde{W}_\ell^{(k)}$ yields the optimal rank-k approximation, and the approximation error in Frobenius norm is given by the square root of the sum of the squares of the discarded singular values:

$$\|\tilde{W}_\ell - \tilde{W}_\ell^{(k)}\|_F = \left(\sum_{i=k+1}^{r} \sigma_i^2\right)^{1/2}, \tag{11}$$

where σ_i are the singular values of $\tilde{W}_\ell$ sorted in descending order, and $r = \min(d_\ell, d_{\ell-1})$.

Therefore, performing low-rank approximation in the whitened feature space guarantees that the output reconstruction error directly corresponds to the truncated SVD error of the whitened weight matrix, which explicitly takes the input covariance structure into account.

D Error Propagation under Layerwise Compression

This section formally derives the total reconstruction error of a deep model under low-rank layerwise approximation.

D.1 Notation

Consider a model with L layers. At each layer ℓ, the pre- and post-activation outputs can be fomulated as:

$$Z_\ell = W_\ell X_{\ell-1}, \qquad X_\ell = \phi_\ell(Z_\ell), \tag{12}$$

where ϕ_ℓ is ρ_ℓ-Lipschitz:

$$\|\phi_\ell(a) - \phi_\ell(b)\|_F \le \rho_\ell \|a - b\|_F, \quad \forall a, b. \tag{13}$$

Assume each weight matrix W_ℓ is compressed to a rank-k_ℓ approximation $\hat{W}_\ell$, producing compressed outputs $\hat{X}_\ell$ through a forward pass.

Define the final-layer reconstruction error:

$$E^{\text{total}} := \|X_L - \hat{X}_L\|_F. \tag{14}$$

D.2 Recursive Error Propagation

We analyze layerwise errors recursively. Using the Lipschitz property of ϕ_ℓ and the triangle inequality in Eq. (13):

$$\|X_\ell - \hat{X}_\ell\|_F = \|\phi_\ell(W_\ell X_{\ell-1}) - \phi_\ell(\hat{W}_\ell \hat{X}_{\ell-1})\|_F \leq \rho_\ell \|W_\ell X_{\ell-1} - \hat{W}_\ell \hat{X}_{\ell-1}\|_F. \tag{15}$$

Decompose the error into:

$$\|W_\ell X_{\ell-1} - \hat{W}_\ell \hat{X}_{\ell-1}\|_F \leq \underbrace{\|W_\ell X_{\ell-1} - \hat{W}_\ell X_{\ell-1}\|_F}_{E_\ell\text{:compression error}} + \underbrace{\|\hat{W}_\ell (X_{\ell-1} - \hat{X}_{\ell-1})\|_F}_{\text{propagated error}}. \tag{16}$$

Let $B_\ell := \|\hat{W}_\ell\|_2$. Then:

$$\|X_\ell - \hat{X}_\ell\|_F \leq \rho_\ell \left(E_\ell + B_\ell \|X_{\ell-1} - \hat{X}_{\ell-1}\|_F \right). \tag{17}$$

Unfolding this recursion yields a bound on the total output error:

$$E^{\text{total}} \leq \sum_{\ell=1}^{L} \left(\prod_{j=\ell+1}^{L} \rho_j B_j \right) \rho_\ell E_\ell. \tag{18}$$

In the case of spectral normalization or bounded operator norm ($B_j \leq 1$), this simplifies to:

$$E^{\text{total}} \leq \sum_{\ell=1}^{L} \left(\prod_{j=\ell+1}^{L} \rho_j \right) \rho_\ell E_\ell. \tag{19}$$

D.3 Influence of Different Input Whitening Strategies

This subsection analyzes the influence of different input whitening strategies on the per-layer error terms E_ℓ:

(a) Raw compression (no whitening):

$$E_\ell^{\text{raw}} = \|W_\ell X_{\ell-1} - \hat{W}_\ell^{\text{raw}} X_{\ell-1}\|_F.$$

(b) Static whitening: A fixed whitening matrix S_ℓ^{stat} is computed from uncompressed features. Define the whitened input:

$$\tilde{X}_{\ell-1} = S_\ell^{\text{stat}} \hat{X}_{\ell-1}, \quad \tilde{X}_{\ell-1} \tilde{X}_{\ell-1}^\top = I + \Delta_\ell, \quad \delta_\ell := \|\Delta_\ell\|_2.$$

Then, using sub-multiplicativity of norms, we have:

$$E_\ell^{\text{stat}} = \|(\tilde{W}_\ell - \tilde{W}_\ell^{(k)}) \tilde{X}_{\ell-1}\|_F \leq \|\tilde{W}_\ell - \tilde{W}_\ell^{(k)}\|_F \cdot \|\tilde{X}_{\ell-1}\|_2.$$

By definition of the whitened input, $\tilde{X}_{\ell-1} \tilde{X}_{\ell-1}^\top = I + \Delta_\ell$, and hence

$$\|\tilde{X}_{\ell-1}\|_2^2 = \|\tilde{X}_{\ell-1} \tilde{X}_{\ell-1}^\top\|_2 = \|I + \Delta_\ell\|_2 = 1 + \delta_\ell,$$

which gives

$$\|\tilde{X}_{\ell-1}\|_2 \leq \sqrt{1 + \delta_\ell}.$$

Substituting into the earlier bound, we obtain:

$$E_\ell^{\text{stat}} \leq \|\tilde{W}_\ell - \tilde{W}_\ell^{(k)}\|_F \cdot \sqrt{1 + \delta_\ell}.$$

(c) Distribution-aware whitening (proposed): Whitening is recomputed on the compressed input:

$$S_\ell = (\hat{X}_{\ell-1}\hat{X}_{\ell-1}^\top + \epsilon I)^{-1/2}, \quad \text{so that } \tilde{X}_{\ell-1}\tilde{X}_{\ell-1}^\top \approx I.$$

This yields a tight bound:

$$E_\ell^{\text{dist}} \approx \|\tilde{W}_\ell - \tilde{W}_\ell^{(k)}\|_F.$$

Plugging the respective per-layer errors into the total error bound:

$$E_{\text{raw}}^{\text{total}} \leq \sum_{\ell=1}^{L} \left(\prod_{j=\ell+1}^{L} \rho_j B_j \right) \rho_\ell E_\ell^{\text{raw}},$$

$$E_{\text{stat}}^{\text{total}} \leq \sum_{\ell=1}^{L} \left(\prod_{j=\ell+1}^{L} \rho_j B_j \right) \rho_\ell \|\tilde{W}_\ell - \tilde{W}_\ell^{(k)}\|_F \cdot \sqrt{1+\delta_\ell},$$

$$E_{\text{dist}}^{\text{total}} \leq \sum_{\ell=1}^{L} \left(\prod_{j=\ell+1}^{L} \rho_j B_j \right) \rho_\ell \|\tilde{W}_\ell - \tilde{W}_\ell^{(k)}\|_F.$$

Distribution-aware whitening aligns the compression with the actual post-compression distribution, ensuring $\tilde{X}_{\ell-1}\tilde{X}_{\ell-1}^\top \approx I$ and effectively minimizing the inflation term $\sqrt{1+\delta_\ell}$. This leads to more robust error propagation across layers.

E Layer-wise Lipschitz Analysis

In this section, we separately measure the Lipschitz constants for the query, key, value, and output projections in the attention sublayers, as well as for the two linear layers in the feed-forward sublayers. The measured constants across layers are visualized in Figure 1. Empirically, we observe the following patterns:

1) In attention sublayers, the query and key projections (Q/K) generally have higher Lipschitz constants than the value projections (V), reflecting their greater sensitivity to input perturbations.

2) In feed-forward sublayers, the first linear layer (W1) tends to have larger Lipschitz constants than the second (W2).

3) In deeper layers, particularly near the output, Lipschitz constants tend to increase slightly, implying that errors can be progressively amplified as they propagate toward the output.

These Lipschitz constants indicate that small errors can be amplified through the network, potentially harming model fidelity. Our distribution-aware whitening mechanism helps suppress such error propagation during low-rank compression.

F Performance Comparison under a Mild Compression Ratio

Due to page limitations, the main text only reports the performance of different compression methods at compression ratios of 30%, 40%, and 50%. Here, we provide a comparison at a more moderate compression ratio of 20%. As shown in Figure 2, on the WikiText-2 task, our proposed method outperforms SVD-LLM across all models, although the margin is relatively small. By comparison, on PTB and C4, our method exhibits much clearer advantages. For example, when compressing the Vicuna-7B model by 20%, our method achieves a PPL of 61.51 on PTB, which is 9.68 points better than SVD-LLM.

G Analysis of Inference Overhead

The computational complexity of a model primarily depends on its architecture, the number of parameters, input sequence length, and hardware implementation. In this section, we use the Vicuna-7B model as an example to analyze the computational savings introduced by our method, with experiments conducted in FP32 precision.

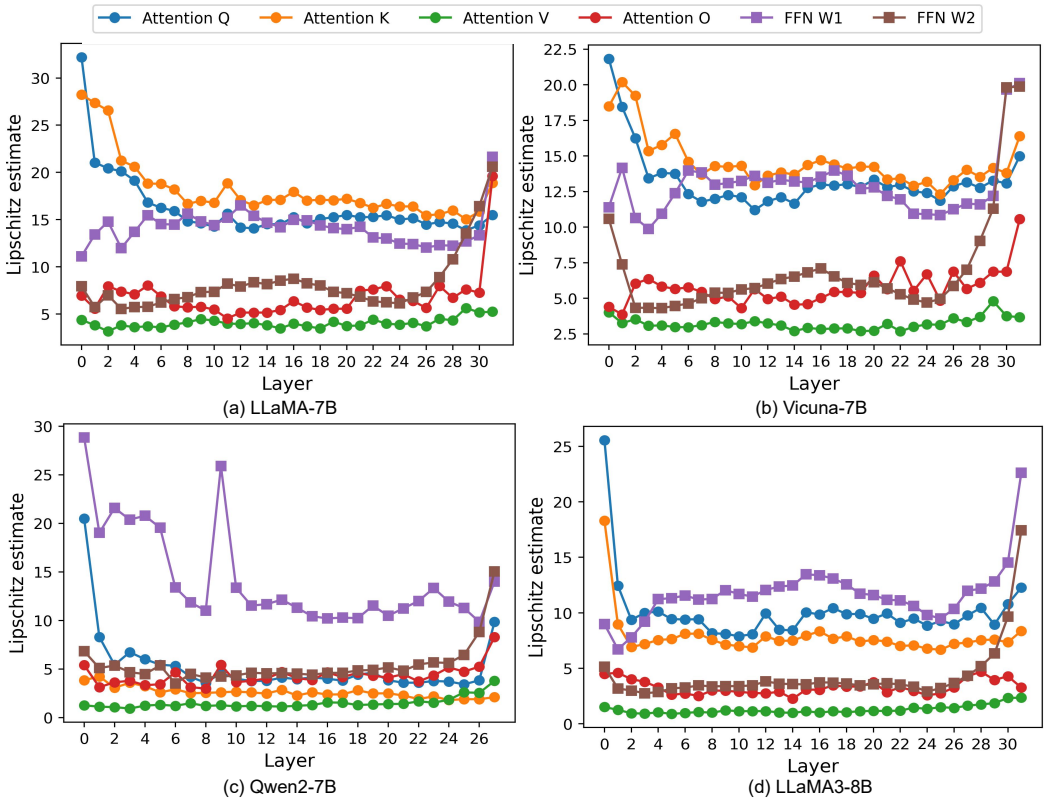


Figure 1: Layer-wise Lipschitz constants for different modules (attention and feed-forward) across various LLMs.

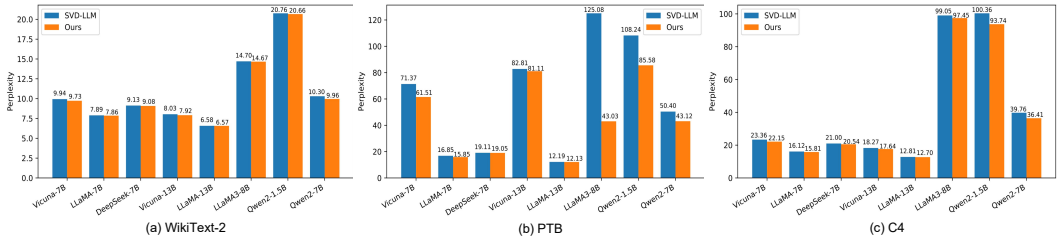


Figure 2: Comparison between our method and SVD-LLM on different models when compressing 20% of parameters.

Our method achieves significant computational reduction by applying structured low-rank approximations to the weight matrices. For a given weight matrix $W \in \mathbb{R}^{m \times n}$, we perform a truncated SVD decomposition: $W \approx U_k \Sigma_k V_k^T$, where $k < \min(m, n)$ is the target rank determined by the compression ratio. This decomposition transforms the original matrix multiplication WX (with $X \in \mathbb{R}^{n \times p}$ and p denoting the number of input vectors) into two sequential operations:

1) $\hat{X} = \Sigma_k V_k^T X$ with approximately knp FLOPs, and

2) $U_k \hat{X}$ with approximately mkp FLOPs,

resulting in a total computational cost of $C_{UV} \approx k \cdot p \cdot (m + n)$, where we simplify by ignoring the factor of 2 from multiply–add operations and other small contributions. The original multiplication has cost $C_{WX} \approx mnp$ under the same simplification.

To select k based on a desired parameter retention ratio α (e.g., $\alpha = 0.6$ for 60% of original parameters), we set: $k = \left\lfloor \frac{\alpha mn}{m+n} \right\rfloor$, ensuring $k < \min(m, n)$. The theoretical FLOPs reduction ratio is then $r = 1 - \frac{C_{UV}}{C_{WX}}$, ignoring additional computations from residual connections, LayerNorm, biases, and activation functions, which typically contribute a small fraction of total FLOPs.

Experiments in FP32 precision across different batch sizes (32, 64, 128), sequence lengths (32, 64, 128), and compression ratios (0.0–0.5) validate these savings (Fig. 3). As the compression ratio increase, memory usage drops (e.g., from 27.56GB to 15.49GB for sequence length 32, batch size 32, when increase ratio from 0.0 to 0.5), and throughput improves (from 475.19 to 598.59 tokens/sec). Larger batch sizes and longer sequences increase the absolute computational savings and throughput, as more input vectors benefit from the reduced computation.

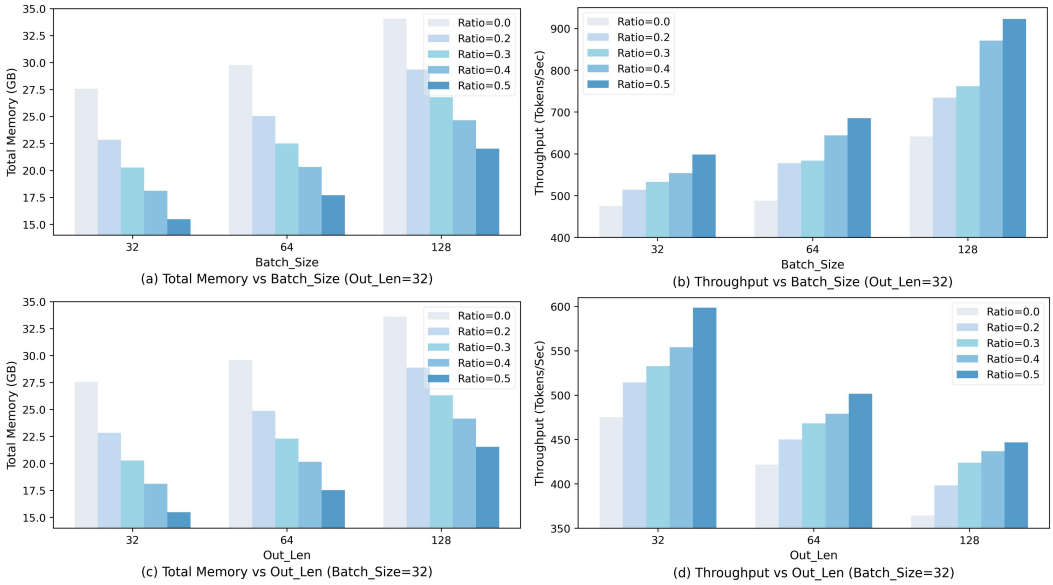


Figure 3: Memory usage (GB) and throughput (tokens/sec) of Vicuna-7B in FP32 precision under different compression ratios.

H Extension to Large Vision-Language Models

A natural question arises: since the proposed method is effective for various large language models, can this compression approach be extended to multimodal scenarios, such as large vision-language models (VLMs)? To investigate this, we conducted experiments on three VLMs: LLaVA-Next 7B, LLaVA-Next 13B, and LLaVA-V1.5 13B, using the ScienceQA task to evaluate the performance of the compressed models.

As shown in Table 1, compared to the standard SVD method, our approach introduces a whitening operation that effectively preserves important information during compression. This results in significantly smaller performance degradation. Moreover, for larger models such as LLaVA-Next 13B and LLaVA-V1.5 13B, the performance loss after compression is even smaller.

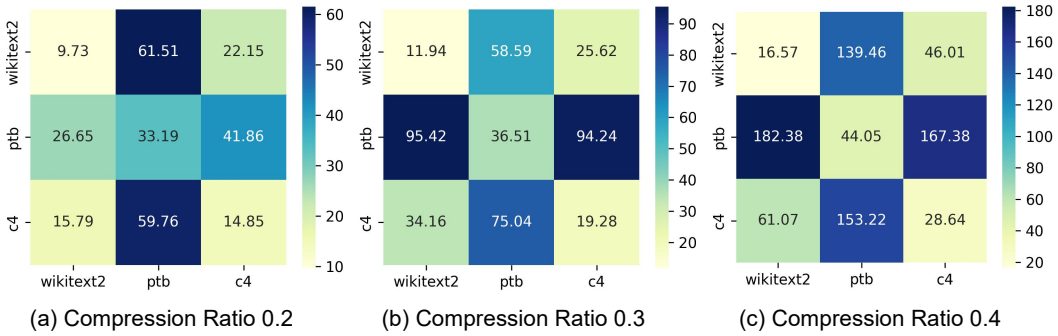
I Impact of Calibration Set Selection

This subsection examines the impact of calibration data used for the our method. Figure 4 illustrates the performance of compressed Vicuna-7B with different calibration datasets. The horizontal axis indicates the test set, while the vertical axis corresponds to the calibration set used during compression. The results show a clear trend: performance is best when the calibration set aligns with the test distribution. This pattern holds consistently across different compression ratios and supports the

Table 1: Performance (accuracy %) of compressed large vision-language models on the ScienceQA task under different compression ratios.

Ratio	LLaVA-Next 7B		LLaVA-Next 13B		LLaVA-V1.5 13B	
	Ours	Standard SVD	Ours	Standard SVD	Ours	Standard SVD
0.0	67.13	67.13	73.72	73.72	71.98	71.98
0.1	65.29	35.05	73.36	57.36	71.14	56.37
0.2	61.42	16.70	72.93	53.54	70.74	47.99
0.3	60.13	11.11	72.58	44.07	70.15	45.71

intuitive expectation that matching distributions reduce activation shift and improve whitening effectiveness. Conversely, domain mismatch between calibration and evaluation data leads to noticeable performance drops, especially at higher compression rates. These findings highlight the importance of choosing representative calibration data for distribution-aware methods. While we follow ASVD (Yuan et al., 2023) and adopt a 256-sample subset from WikiText-2 for most of our main experiments to ensure fair comparison, exploring more principled or adaptive calibration strategies remains a valuable direction for future work.



(a) Compression Ratio 0.2 (b) Compression Ratio 0.3 (c) Compression Ratio 0.4

Figure 4: Perplexity with different calibration datasets.

J Comparison of Compressed Models in Zero-Shot Settings

We conduct experiments on 16 representative subtasks from the **Massive Multitask Language Understanding (MMLU)** benchmark (Hendrycks et al., 2021), which covers a diverse set of knowledge domains including humanities, social sciences, STEM, and other disciplines. To make the results table concise and readable, we use the following abbreviations for subtask names:

- **HS_US_Hist**: High School US History
- **HS_World_Hist**: High School World History
- **Intl_Law**: International Law
- **Log_Fallacies**: Logical Fallacies
- **Bus_Ethics**: Business Ethics
- **Global_Facts**: Global Facts
- **Marketing**: Marketing
- **Virology**: Virology
- **HS_Gov_Pol**: High School Government and Politics
- **Human_Sexuality**: Human Sexuality
- **Prof_Psych**: Professional Psychology
- **US_Foreign_Pol**: US Foreign Policy
- **College_Bio**: College Biology

- **College_Phys**: College Physics
- **Comp_Sec**: Computer Security
- **Elec_Eng**: Electrical Engineering

Table 2 compares the zero-shot performance at a compression ratio of 20% between our proposed method and SVD-LLM, which represents the current state-of-the-art in SVD-based model compression. Accuracy is reported across three model backbones: Vicuna-V1.5-7B, Qwen2-7B, and LLaMA3-8B, with the performance gain of our method shown alongside. Overall, our method achieves better performance than the SVD-LLM in the majority of evaluated subtasks. For example, on the *High School Government and Politics* subtask under the Qwen2-7B model, our method improves accuracy from 24.35% to 41.97%, achieving a gain of +17.62%. Similarly, in the *Human Sexuality* subtask, accuracy increases from 24.43% to 39.69%, resulting in a gain of +15.26%. These substantial improvements across different domains and models indicate that our method generalizes well and provides a more robust alternative to traditional SVD-based compression strategies.

Table 2: Comparison of accuracy between SVD-LLM and our method across diverse subtasks in the Massive Multitask Language Understanding Benchmark.

Category	**Subtask**	**Vicuna-V1.5-7B**			**Qwen2-7B**			**LLaMA3-8B**		
		SVD-LLM	Ours	Gain	SVD-LLM	Ours	Gain	SVD-LLM	Ours	Gain
Humanities	HS_US_Hist	0.3529	0.4167	0.0638	0.3039	0.3873	0.0834	0.2843	0.2892	0.0049
	HS_World_Hist	0.3882	0.4473	0.0591	0.3418	0.4641	0.1223	0.2405	0.2785	0.0380
	Intl_Law	0.3884	0.4711	0.0827	0.2810	0.3140	0.0330	0.3802	0.4876	0.1074
	Log_Fallacies	0.2822	0.3620	0.0798	0.2699	0.3558	0.0859	0.2699	0.3129	0.0430
Other	Bus_Ethics	0.3200	0.3600	0.0400	0.2800	0.4000	0.1200	0.2700	0.3100	0.0400
	Global_Facts	0.2400	0.3500	0.1100	0.1900	0.2200	0.0300	0.3500	0.3400	-0.0100
	Marketing	0.3376	0.4530	0.1154	0.3547	0.4530	0.0983	0.3077	0.2991	-0.0086
	Virology	0.3072	0.3614	0.0542	0.2651	0.2952	0.0301	0.2229	0.2349	0.0120
Social science	HS_Gov_Pol	0.3420	0.3834	0.0414	0.2435	0.4197	0.1762	0.2280	0.3057	0.0777
	Human_Sexuality	0.3588	0.4046	0.0458	0.2443	0.3969	0.1526	0.2214	0.2366	0.0152
	Prof_Psych	0.2827	0.3072	0.0245	0.2843	0.3268	0.0425	0.2810	0.2810	0.0000
	US_Foreign_Pol	0.4200	0.5200	0.1000	0.3000	0.4600	0.1600	0.2700	0.3500	0.0800
STEM	College_Bio	0.2708	0.3056	0.0348	0.2639	0.3472	0.0833	0.2917	0.2431	-0.0486
	College_Phys	0.2941	0.3137	0.0196	0.2353	0.3039	0.0686	0.2255	0.2549	0.0294
	Comp_Sec	0.3200	0.3500	0.0300	0.2600	0.3200	0.0600	0.2900	0.3800	0.0900
	Elec_Eng	0.2897	0.3241	0.0344	0.2621	0.2759	0.0138	0.2897	0.3034	0.0137
Average		0.3247	0.3831	0.0585	0.2737	0.3587	0.0850	0.2764	0.3067	0.0303

K Additional Results for Lightweight Post-Compression Fine-Tuning

Lightweight Post-Compression Fine-Tuning is a widely used approach (Ma et al., 2023; Wang et al., 2025) for recovering performance after model compression. Due to page limitations, only a subset of the experimental results is presented in the main text; here we provide additional data in Table 3. In most cases, our method outperforms the baseline SVD-LLM across different language models after lightweight post-compression fine-tuning. For example, at a compression ratio of 0.3, with a full update on the LLaMA3-8B model, the perplexity of our method on the PTB dataset is 27.5872, which is 6.5991 points lower than that of SVD-LLM (34.1863). As the compression ratio increases, this performance improvement becomes even more pronounced.

L Additional Convergence Analysis of LoRA Initialization Strategies

Figure. 5 presents training convergence curves of the three LoRA initialization methods under various rank settings on mathematical (MetaMathQA) and programming (Code Feedback) tasks. The results further confirm the advantage of our proposed initialization in accelerating convergence and improving training stability.

Table 3: Perplexity of our method across three language models under different compression ratios. After lightweight post-compression fine-tuning, our proposed method outperforms the baseline method SVD-LLM in most cases.

Ratio	Method	Vicuna-7B			Qwen2-7B			LLaMA3-8B		
		WikiText-2	PTB	C4	WikiText-2	PTB	C4	WikiText-2	PTB	C4
0.3	SVD-LLM Partial Update	9.5785	54.2648	**16.0825**	11.3648	33.1960	26.6838	**14.5375**	42.4973	38.2274
	Ours Partial Update	**9.5781**	**53.7169**	16.2617	**11.2563**	**32.7206**	**26.5142**	15.8428	**31.0731**	**33.1449**
	SVD-LLM Full Update	9.1292	52.7985	**14.6065**	10.2916	25.4728	**20.7969**	**13.1556**	34.1863	32.8736
	Ours Full Update	**9.0960**	**52.1565**	14.7191	**10.1894**	**25.2227**	21.3162	14.8881	**27.5872**	**31.0844**
0.4	SVD-LLM Partial Update	11.3849	77.1259	19.8853	16.1168	50.0841	37.0297	**20.7856**	74.2456	64.5032
	Ours Partial Update	**11.3136**	**71.2715**	**19.8202**	**15.2767**	**44.3175**	**34.2671**	22.2111	**37.9897**	**49.5095**
	SVD-LLM Full Update	10.5504	79.9615	17.3544	12.2817	34.9910	26.9585	**17.7929**	51.4241	55.1955
	Ours Full Update	**10.5386**	**65.2732**	**17.3092**	**12.2373**	**32.5209**	**26.6492**	20.4144	**34.3254**	**43.0600**
0.5	SVD-LLM Partial Update	14.7036	130.6588	26.8014	22.4540	74.3111	54.7868	51.7419	193.0543	124.1097
	Ours Partial Update	**14.3285**	**108.6498**	**25.6585**	**20.2631**	**58.0048**	**46.4589**	**36.6911**	**75.2853**	**73.5259**
	SVD-LLM Full Update	12.7389	180.4358	21.5852	16.3075	46.8237	39.3111	49.4664	128.2454	85.4848
	Ours Full Update	**12.6762**	**124.4990**	**21.5079**	**16.0868**	**40.7479**	**35.7692**	**32.1073**	**64.5531**	**64.6075**

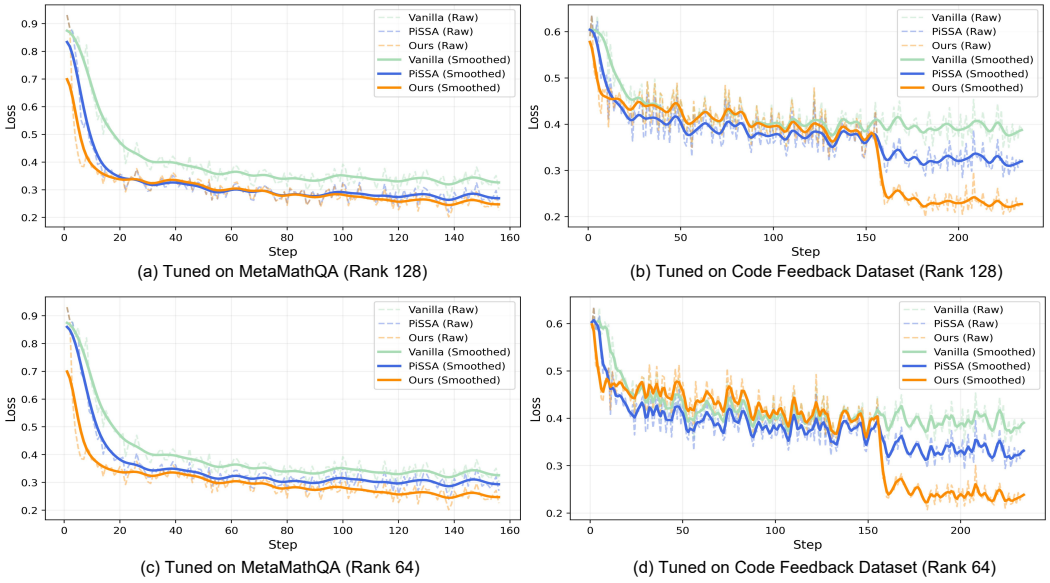


Figure 5: Training loss for fine-tuning Vicuna-V1.5-7B with LoRA (rank 128 & 64) under three different initialization schemes: Vanilla Kaiming Initialization, PiSSA, and our proposed method. (a) Results on the MetaMathQA dataset; (b) results on the Code Feedback dataset. Dashed lines represent raw training loss values, while solid lines denote Gaussian-smoothed curves for improved visual clarity.

M SUITABILITY OF SVD-LLM FOR PEFT INITIALIZATION

This section clarifies that distribution-aware low-rank approximations serve as a stronger initialization for LoRA-style PEFT than existing SVD-based methods.

Why distribution-aware initialization helps: Recent work such as PiSSA has highlighted that an initialization which better preserves the model's task-relevant subspace provides a superior starting point for PEFT. Concretely, distribution-aware whitening reduces the cumulative, layer-wise distortion introduced by low-rank approximation, so the removed components have smaller impact on model outputs. As a result, when used as an initialization for LoRA-style training, such approximations: 1)Start closer to a good local optimum. 2)Typically converge faster and achieve a better final solution.

Why SVD-LLM may be insufficient: Methods like SVD-LLM overlook approximation-induced distribution shifts across layers. When the retained rank is small, approximation errors accumulate, and the preserved subspace may fail to capture crucial predictive directions. Therefore, while SVD-LLM can improve over vanilla LoRA in some settings, it does not consistently match the benefits offered by distribution-aware initialization.

Empirical evidence: We evaluated Vicuna-7B across a set of math and coding tasks. Figure 6 reports results (higher is better) for different initializations and rank settings, averaged across 3 random seeds. These results demonstrate that: SVD-LLM can outperform vanilla LoRA. However, SVD-LLM is not consistently better than PiSSA. Our distribution-aware initialization yields the most consistent improvements across tasks and rank settings.

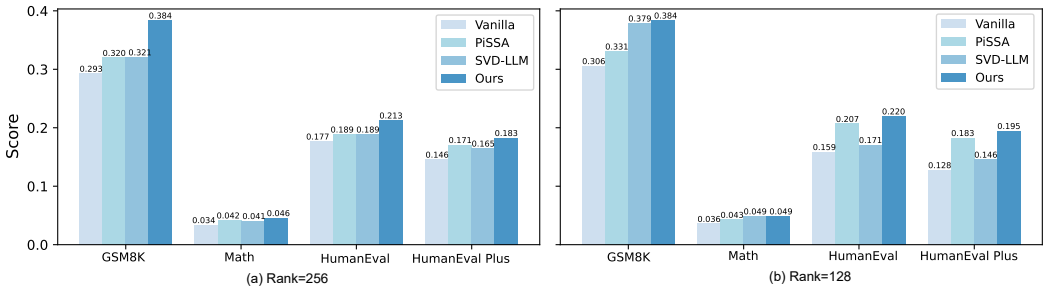


Figure 6: Performance comparison of four PEFT initialization methods on Vicuna-7B across four tasks for both rank 256 and 128.

N IMPACT OF PEFT ON OTHER TASKS AFTER FINE-TUNING ON SPECIFIC TASKS

Both our method and PiSSA initialize the LoRA modules using the important components of the model's weight matrices. A natural question arises: does this initialization lead to severe forgetting of previously learned knowledge? Table 4 compares the performance of the Qwen2-7B model fine-tuned on mathematical and coding tasks with a rank of 256, evaluated on the MMLU Benchmark. Interestingly, models fine-tuned using different initialization strategies show almost no degradation in performance compared to the original model. In some cases, slight improvements are even observed. These results demonstrate that initializing LoRA with the most important components of the weight matrices does not cause severe forgetting, indicating good usability and robustness of our approach.

Table 4: Performance of models fine-tuned on different tasks using various LoRA initialization methods, evaluated on the MMLU Benchmark. The results show that our method does not cause severe forgetting.

Category	**Subtask**	**Original**	**Peft on MetaMathQA**			**Peft on Code Feedback**		
			Vanilla	**PiSSA**	**Ours**	**Vanilla**	**PiSSA**	**Ours**
Humanities	HS_US_Hist	0.8431	0.8529	0.8284	0.8529	0.8431	0.8480	0.8431
	HS_World_Hist	0.8312	0.8354	0.8397	0.8312	0.8354	0.8439	0.8397
	Intl_Law	0.8182	0.8512	0.8347	0.8512	0.8512	0.8347	0.8512
	Log_Fallacies	0.8037	0.8037	0.7362	0.8221	0.7791	0.7853	0.7853
Other	Bus_Ethics	0.7800	0.7600	0.7700	0.7800	0.7900	0.7800	0.7700
	Global_Facts	0.4900	0.4900	0.2800	0.4900	0.4900	0.4800	0.4900
	Marketing	0.9274	0.9231	0.8846	0.9188	0.9274	0.9274	0.9188
	Virology	0.5301	0.5422	0.5482	0.5422	0.5422	0.5422	0.5482
Social science	HS_Gov_Pol	0.9016	0.9067	0.9016	0.9067	0.8964	0.9067	0.9067
	Human_Sexuality	0.8321	0.8473	0.8244	0.8397	0.8397	0.8321	0.8321
	Prof_Psych	0.7484	0.7500	0.7533	0.7614	0.7533	0.7533	0.7565
	US_Foreign_Pol	0.8800	0.9000	0.8900	0.9000	0.8900	0.8900	0.8900
Stem	College_Bio	0.8056	0.7917	0.7847	0.7917	0.7917	0.7986	0.7847
	College_Phys	0.4412	0.4216	0.3725	0.4020	0.4118	0.3922	0.4510
	Comp_Sec	0.7700	0.7800	0.7800	0.7800	0.7800	0.7800	0.7800
	Elec_Eng	0.7241	0.7448	0.6828	0.7310	0.7379	0.7241	0.7310
Average		0.7579	0.7625	0.7319	0.7626	0.7600	0.7574	0.7611

REFERENCES

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LNYIUouhdt.

Zhihang Yuan, Yuzhang Shang, Yue Song, Qiang Wu, Yan Yan, and Guangyu Sun. Asvd: Activation-aware singular value decomposition for compressing large language models. *arXiv preprint arXiv:2312.05821*, 2023.