

DELTA: Decoupling Latent Heterogeneity in Asymmetric Low-Rank Compression

Jialin Zhan* Fangxin Liu*,† Junjie Wang Ning Yang Li Jiang† Haibing Guan

School of Computer Science, Shanghai Jiao Tong University

{jialin.jane, liufangxin, wjj_conti, yn937391832, ljiang_cs, hbguan}@sjtu.edu.cn

*Equal Contribution

†Corresponding Authors

Abstract

Singular Value Decomposition (SVD) is a promising post-training compression technique for large language models (LLMs), as it is training-free, hardware-friendly, and preserves regular dense computation. However, existing SVD-based methods suffer from severe accuracy degradation under aggressive compression. This is mainly because hard rank truncation discards useful singular directions, while the two resulting SVD factors are usually compressed symmetrically despite their different roles, shapes, and quantization sensitivities. In this paper, we propose DELTA, an asymmetric low-rank compression method that decouples the latent heterogeneity between the two SVD factors. Under a fixed memory budget, the proposed method jointly determines the retained rank, factor-wise quantization bit-widths, and quantization directions, allowing more effective singular directions to be preserved with low-bit representations. To efficiently select compression configurations, we use an activation-aware reconstruction error as a layer-wise proxy objective. Experiments on multiple LLMs and datasets show that DELTA consistently outperforms existing post-training SVD-based compression methods, especially under high compression ratios. Compared with the strongest SVD-based baseline at the same compression ratio, DELTA reduces perplexity on WikiText-2 and C4 by up to 68.2% and 53.2%, respectively, and improves average zero-shot accuracy by up to 11 percentage points. These results demonstrate that asymmetric factor-aware compression can effectively mitigate the accuracy degradation of SVD-based LLM compression at high compression ratios. Code is available at <https://github.com/BlacksheepJane/DELTA>.

1 Introduction

The rapid growth of large language models (LLMs) has substantially increased deployment and inference costs (Zhao et al., 2026; Touvron et al.,

2023a,b). Due to massive parameter sizes and expensive matrix operations, deploying LLMs on resource-constrained or latency-sensitive platforms remains challenging (Liu et al., 2021). As a result, hardware-friendly post-training compression has become increasingly important.

Among existing compression techniques (Cheng et al., 2018; Choudhary et al., 2020; Zhu et al., 2024; Liu et al., 2025), quantization (Frantar et al., 2022; Xiao et al., 2023; Lin et al., 2024; Dettmers et al., 2023; Müller et al., 2025; Liu et al., 2024), pruning (Frantar and Alistarh, 2023; Ma et al., 2023; Ashkboos et al., 2024), and knowledge distillation (Gu et al., 2024; Ko et al., 2024) have been widely studied, but they often rely on specialized hardware support or additional training. In contrast, low-rank decomposition preserves dense computation while reducing storage requirements and computational costs. Singular Value Decomposition (SVD), as a representative low-rank method, is training-free, hardware-friendly, and flexible (Golub et al., 1987). However, directly applying SVD to LLMs usually causes significant accuracy degradation under aggressive compression.

Recent SVD-based LLM compression methods, including FWSVD (Hsu et al., 2022), ASVD (Yuan et al., 2023), SVD-LLM (Wang et al., 2025c), Dobi-SVD (Wang et al., 2025a), and SVD-LLM V2 (Wang et al., 2025b), mainly focus on improving rank truncation criteria or allocating compression ratios across layers. Although these methods reduce SVD-induced errors to some extent, they still suffer from severe performance loss at high compression ratios. We argue that this limitation stems from two overlooked issues. First, aggressive truncation weakens the low-rank assumption of SVD by discarding important singular directions. Second, the two SVD factors exhibit latent heterogeneity in functional roles, matrix structures, and directional properties, yet existing methods usually

process them in a symmetric manner.

In this paper, we propose DELTA, an SVD-based post-training compression method for LLMs. DELTA integrates low-rank decomposition with low-bit quantization, allowing more effective singular directions to be retained under the same memory budget. It further applies asymmetric mixed-precision quantization to the two SVD factors to account for differences in their quantization sensitivities, and assigns factor-specific quantization directions to capture their directional heterogeneity. By explicitly decoupling the latent heterogeneity between the two SVD factors, DELTA enables asymmetric low-rank compression and significantly mitigates the accuracy degradation of SVD-based methods under high compression ratios.

We evaluate DELTA on five LLMs, two in-domain language modeling datasets, and six zero-shot task datasets. The experimental results show that:

- DELTA consistently outperforms state-of-the-art SVD-based compression methods, with larger gains under higher compression ratios.
- DELTA surpasses structured pruning and standalone quantization baselines, showing the advantage of asymmetric low-rank compression under aggressive compression.
- DELTA is hardware-friendly: it requires no post-training, preserves regular dense computation, and achieves efficient compression without specialized hardware support.

2 Background

2.1 SVD-based Low-Rank Decomposition

Singular Value Decomposition (SVD) is a classical low-rank decomposition technique (Golub et al., 1987). Given a weight matrix $W \in \mathbb{R}^{m \times n}$, SVD decomposes it as

$$W = U\Sigma V^T, \quad (1)$$

where U and V denote the left and right singular vector matrices, respectively, and Σ is a diagonal matrix containing the singular values.

When applying SVD to compress linear layers in LLMs, the singular value matrix Σ is usually distributed to both sides of the decomposition for efficient forward inference. In this way, the original

three-matrix factorization can be rewritten as the product of two low-rank matrices:

$$\begin{aligned} W &\approx U_r \Sigma_r V_r^T \\ &= \left(U_r \Sigma_r^{1/2} \right) \left(\Sigma_r^{1/2} V_r^T \right) \\ &= W_u W_v. \end{aligned} \quad (2)$$

Here, W_v projects the input representation into a low-dimensional latent space, while W_u reconstructs the low-dimensional representation back to the output space. Some existing SVD-based methods further modify W_u or W_v using activation information or error compensation, but their basic computational structure and functional roles remain unchanged.

This formulation preserves dense matrix multiplication and usually requires no additional training, making SVD a promising hardware-friendly post-training compression method for LLMs.

2.2 SVD-based LLM Compression Methods

Although SVD is hardware-friendly and training-free, directly applying conventional SVD to LLM weight matrices often causes severe degradation because layer outputs depend on both the singular spectrum and input activations. Existing SVD-based methods therefore aim to reduce output reconstruction error after low-rank approximation.

Prior methods mostly improve SVD from two perspectives. The first line focuses on better truncation criteria: FWSVD (Hsu et al., 2022) uses Fisher information to estimate weight importance, ASVD (Yuan et al., 2023) rescales weights with activation statistics, SVD-LLM (Wang et al., 2025c) introduces activation whitening, and SVD-LLM V2 and Dobi-SVD further refine the truncation objective using more stable or activation-centric formulations. Despite different derivations, these methods essentially search for better rank truncation under a given compression constraint. The second line studies adaptive rank or compression-ratio allocation, as exemplified by SVD-LLM V2 and Dobi-SVD, but such layer-wise allocation mainly provides local refinement and remains insufficient under high compression ratios.

Some methods further introduce post-processing strategies, such as LoRA-based recovery in SVD-LLM (Hu et al., 2022) or quantization-aware remapping in Dobi-SVD. However, these techniques are usually applied after SVD truncation, rather than being deeply integrated with the struc-

ture of the decomposed factors. In particular, existing methods largely overlook the latent heterogeneity between the two SVD factors, which differ in functional role, matrix shape, numerical distribution, and quantization sensitivity.

3 Motivation

Observation 1: The Low-Rank Assumption Breaks Down under High Compression Ratios

SVD compression relies on the low-rank assumption that most information in a weight matrix is concentrated in a few dominant singular directions. While this assumption is reasonable under mild compression, high compression ratios require truncating many more directions, some of which may still be important to model outputs, leading to severe degradation.

From a storage perspective, conventional SVD truncation can be viewed as an extreme “0-bit + 16-bit” strategy: retained directions are stored in high precision, while discarded directions receive 0 bits. Under aggressive compression, this hard truncation removes useful components and weakens the low-rank approximation, resulting in severe model degradation.

Therefore, simply increasing the truncation rate is suboptimal. DELTA introduces quantization bit-width as an additional compression degree of freedom, providing low-bit representations between 0 and 16 bits. By combining moderate truncation with low-bit representation, DELTA preserves more singular directions under the same memory budget and achieves a better trade-off between compression and information preservation.

Observation 2: The Two Low-Rank Factors Exhibit Functional and Shape Heterogeneity.

In the SVD-based low-rank representation, W_u and W_v are not fully symmetric factors. During the forward pass of a compressed linear layer, the input representation is first projected into a low-dimensional latent space by W_v , and then mapped back to the output space by W_u . Therefore, the two factors play different roles and are located at different positions in the computation chain, which may cause the two factors to affect the final output error differently.

Meanwhile, W_u and W_v can also differ in matrix shape. When the input and output dimensions are not equal, the two factors naturally have different parameter scales and storage costs. In general, the average dimensionality of W_u in LLMs is larger

than that of W_v , which means that applying the same rank truncation to both factors can result in a larger truncation loss on W_u .

Existing methods usually apply symmetric compression strategies to the two SVD factors, implicitly ignoring their functional and shape heterogeneity. To address this issue, we introduce dynamic bit-width allocation for W_u and W_v , enabling asymmetric mixed-precision quantization. Under the same overall compression ratio, this design allows the compression loss to be allocated more appropriately between the two factors, thereby improving the performance of the compressed model.

Observation 3: The Two Low-Rank Factors Exhibit Directional Heterogeneity.

Beyond bit-width allocation, quantization direction also affects the accuracy of SVD-compressed models. From the SVD perspective, the columns of W_u and the rows of W_v correspond to left and right singular directions, respectively, suggesting that W_u and W_v may prefer different quantization directions. However, in LLM linear layers, W_u and W_v also inherit input-output channel semantics: the rows of W_u correspond to output channels, while the columns of W_v correspond to input channels. This channel perspective may imply a preference for a quantization direction different from that suggested by the singular-vector perspective.

Moreover, activation-calibrated SVD methods often modify W_u and W_v using activation statistics, making their numerical scales, outlier patterns, and directional distributions layer-dependent. Therefore, the optimal quantization direction cannot be reliably determined by a fixed rule. Applying the same direction to both factors ignores their directional heterogeneity and is likely suboptimal. We thus search over the candidate quantization directions for each factor independently, allowing the selected direction to adapt to its numerical and structural characteristics.

4 Method

4.1 Overview

As illustrated in Figure 1, the overall workflow of DELTA is as follows. First, we apply activation-calibrated SVD to the original weight matrix W and obtain the full-rank left and right factors W_u and W_v . Then, instead of compressing the two factors with a uniform rank or a uniform bit-width, DELTA searches for an asymmetric compression

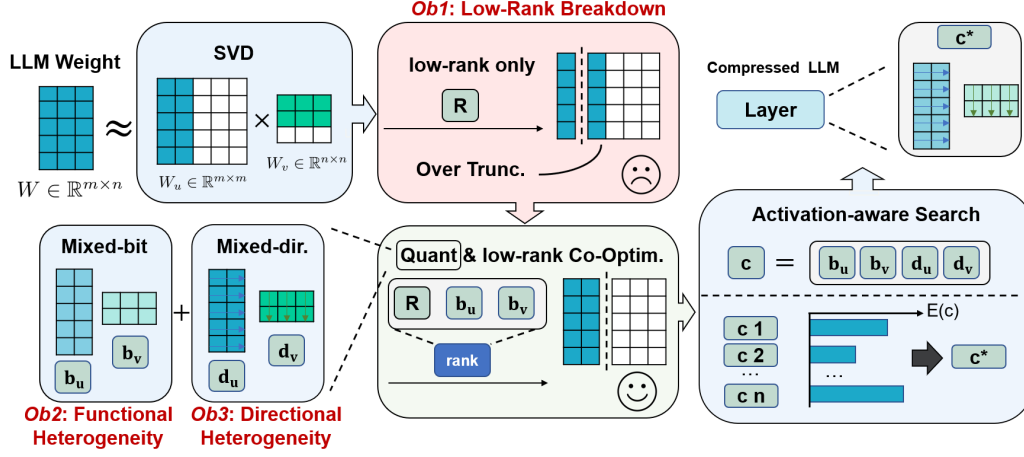


Figure 1: Overview of the DELTA workflow.

configuration for each weight matrix:

$$c = (b_u, b_v, d_u, d_v), \quad (3)$$

where b_u and b_v denote the quantization bit-widths of W_u and W_v , respectively, and d_u and d_v denote their quantization directions.

For each candidate configuration, DELTA computes the corresponding retained rank according to the target memory ratio, and then applies direction-aware quantization to the truncated W_u and W_v . Finally, DELTA selects the optimal configuration using the layer-wise output reconstruction error, and obtains the final compressed low-rank factors and weight approximation:

$$\hat{W}_c = \hat{W}_u \hat{W}_v. \quad (4)$$

In the following, we describe the two major compression strategies of DELTA.

4.2 Compression Strategy

The compression strategy of DELTA consists of two parts: asymmetric mixed-precision low-rank quantization and direction-aware quantization.

4.2.1 Asymmetric Mixed-Precision Low-Rank Quantization

For a weight matrix $W \in \mathbb{R}^{m \times n}$, let b_0 denote the original storage bit-width and ρ denote the target memory ratio. If the two SVD factors W_u and W_v are quantized using the same bit-width b , and the average per-element storage overhead introduced by auxiliary quantization parameters is denoted as α , then the compressed storage cost can be approximated as

$$S_{\text{uni}}(b, r) = r(m+n)(b+\alpha). \quad (5)$$

To satisfy the target compression constraint, we have

$$S_{\text{uni}}(b, r) \leq \rho b_0 m n. \quad (6)$$

Therefore, the retained rank corresponding to this bit-width is

$$r_{\text{uni}}(b) = \left\lfloor \frac{\rho b_0 m n}{(m+n)(b+\alpha)} \right\rfloor. \quad (7)$$

This formula shows that, under the same target memory ratio, reducing the quantization bit-width can increase the retained rank, thereby preserving more singular directions.

DELTA applies asymmetric mixed-precision quantization to the two factors. Under the same overall compression ratio, this design provides a more flexible trade-off among rank preservation, quantization error, and storage cost.

Specifically, when the quantization bit-widths of W_u and W_v are b_u and b_v , respectively, and the corresponding per-element auxiliary storage overheads are α_u and α_v , the retained rank can be written as

$$r(c) = \left\lfloor \frac{\rho b_0 m n}{m(b_u + \alpha_u) + n(b_v + \alpha_v)} \right\rfloor. \quad (8)$$

4.2.2 Direction-Aware Quantization

Quantization directions differ in how scales and zero points are shared, leading to different sensitivities to outliers and channel-wise scale variations. DELTA therefore assigns a quantization direction to each factor independently.

For a low-rank factor A , we define direction-aware quantization as

$$\hat{A} = Q_{b,d}(A), \quad (9)$$

where b denotes the quantization bit-width and d denotes the quantization direction.

In this paper, we consider three candidate quantization directions:

$$d \in \mathcal{D} = \{\text{row}, \text{column}, \text{dual}\}. \quad (10)$$

For row-wise quantization, elements within each row or row-wise group share the same scale and zero point. For column-wise quantization, elements in the same column or column-wise group share the same quantization parameters. Dual-direction quantization jointly considers both row-wise and column-wise quantization.

Accordingly, DELTA searches the quantization directions of W_u and W_v independently:

$$d_u, d_v \in \mathcal{D}. \quad (11)$$

In this way, DELTA allows the two low-rank factors to use different quantization directions, so that the quantization process can better adapt to their numerical distributions and directional characteristics.

4.3 Activation-aware Configuration Search

DELTA searches both the quantization bit-widths and quantization directions for the two SVD factors. For each weight matrix, a candidate compression configuration is defined as

$$c = (b_u, b_v, d_u, d_v), \quad (12)$$

where $b_u, b_v \in \mathcal{B}$ denote the bit-widths of W_u and W_v , and $d_u, d_v \in \mathcal{D}$ denote their quantization directions. The candidate space for a single matrix is

$$|\mathcal{C}| = |\mathcal{B}|^2 |\mathcal{D}|^2. \quad (13)$$

Since the search space for a full-model joint search grows exponentially with the number of compressed matrices, DELTA performs a layer-wise local search using activation-aware output reconstruction error.

For a linear layer with weight matrix W and calibration input activations X , a candidate configuration c produces a compressed weight

$$\hat{W}_c = \hat{W}_u \hat{W}_v. \quad (14)$$

DELTA evaluates the candidate by measuring the output discrepancy between the original and compressed linear layers:

$$\mathcal{E}(c) = \left\| XW^T - X\hat{W}_c^T \right\|_F^2. \quad (15)$$

Compared with weight reconstruction error, this objective better reflects the actual forward error induced by compression.

Directly computing this error for every candidate can be expensive because the calibration activation X contains many tokens. To reduce the evaluation cost, we compress the activation statistics using Cholesky decomposition. Let

$$\Delta W_c = W - \hat{W}_c, \quad H = X^T X. \quad (16)$$

We compute the Cholesky factorization

$$H = LL^T. \quad (17)$$

Then the output reconstruction error can be rewritten as

$$\begin{aligned} \mathcal{E}(c) &= \|X\Delta W_c^T\|_F^2 \\ &= \text{Tr}(\Delta W_c X^T X \Delta W_c^T) \\ &= \text{Tr}(\Delta W_c L L^T \Delta W_c^T) \\ &= \|\Delta W_c L\|_F^2. \end{aligned} \quad (18)$$

Thus, DELTA only needs the Cholesky factor L to efficiently evaluate different candidate configurations:

$$\mathcal{E}(c) = \|(W - \hat{W}_c)L\|_F^2. \quad (19)$$

Finally, for each weight matrix, DELTA selects the configuration with the minimum activation-aware reconstruction error:

$$c^* = \arg \min_{c \in \mathcal{C}} \mathcal{E}(c). \quad (20)$$

The selected configuration determines the retained rank, U/V bit-widths, and quantization directions, which are then used to generate the final compressed low-rank factors.

5 Experiments

We evaluate DELTA along five dimensions: compression performance, cross-model generalization, comparison with other compression paradigms, deployment efficiency, and ablation analysis.

5.1 Experimental Setup

Baselines. We mainly compare DELTA with existing SVD-based LLM compression methods, including ASVD, SVD-LLM, and Dobi-SVD.

Models and Datasets. To demonstrate the generality of our method, we evaluate DELTA on five models with different backbones and scales, including LLaMA-7B, LLaMA-13B, LLaMA-30B,

Ratio	Method	PPL ($\downarrow$)		Accuracy ($\uparrow$)						Avg. ($\uparrow$)	Drop ($\downarrow$)
		Wiki2	C4	Openb.	ARC-e	WinoG.	HellaS.	PIQA	MathQA		
0.0	Original	5.68	7.34	0.34	0.75	0.70	0.57	0.79	0.27	0.57	0.0%
0.2	ASVD	11.14	15.93	0.29	0.53	0.64	0.41	0.68	0.17	0.45	20.5%
	SVD-LLM $\dagger$	7.73	12.23	0.33	0.67	0.69	0.55	0.79	0.26	0.55	3.8%
	Dobi-SVD	6.08	7.83	0.32	0.72	0.71	0.54	0.77	0.27	0.55	2.6%
	DELTA	5.69 ($\downarrow 6.4\%$)	7.36 ($\downarrow 6.0\%$)	0.34	0.75	0.69	0.56	0.78	0.27	0.56 (+0.01)	0.8% ($\downarrow 1.6\%$)
0.4	ASVD	1407	1109	0.08	0.11	0.09	0.08	0.13	0.08	0.10	83.3%
	SVD-LLM $\dagger$	9.27	15.63	0.29	0.59	0.68	0.52	0.69	0.20	0.50	13.2%
	Dobi-SVD	8.12	12.63	0.32	0.72	0.64	0.45	0.72	0.25	0.52	8.7%
	DELTA	5.82 ($\downarrow 28.3\%$)	7.55 ($\downarrow 40.2\%$)	0.33	0.74	0.70	0.56	0.78	0.26	0.56 (+0.04)	0.5% ($\downarrow 8.2\%$)
0.6	ASVD	57057	43036	0.05	0.04	0.06	0.09	0.08	0.05	0.06	89.2%
	SVD-LLM $\dagger$	15.00	26.26	0.18	0.42	0.44	0.31	0.35	0.12	0.30	46.8%
	Dobi-SVD	9.95	17.94	0.27	0.58	0.58	0.38	0.65	0.23	0.44	21.3%
	DELTA	6.14 ($\downarrow 38.3\%$)	8.40 ($\downarrow 53.2\%$)	0.30	0.73	0.68	0.54	0.78	0.25	0.54 (+0.10)	2.6% ($\downarrow 18.7\%$)
0.8	ASVD	80425	67927	0.04	0.03	0.03	0.02	0.01	0.01	0.03	94.7%
	SVD-LLM $\dagger$	31.79	43.71	0.11	0.23	0.21	0.14	0.17	0.08	0.16	71.9%
	Dobi-SVD	93.74	484.79	0.14	0.28	0.50	0.27	0.53	0.20	0.32	43.8%
	DELTA	10.10 ($\downarrow 68.2\%$)	26.94 ($\downarrow 38.4\%$)	0.23	0.56	0.60	0.36	0.65	0.22	0.43 (+0.11)	20.1% ($\downarrow 23.7\%$)

Table 1: Performance comparison of compressed LLaMA-7B models on language modeling and zero-shot downstream tasks. The best result in each column is shown in bold. Ratio denotes the parameter reduction ratio, and Drop denotes the relative decrease in average accuracy compared with the original model. Most baseline results are taken from the SVD-LLM and Dobi-SVD papers, whereas the Dobi-SVD result at a compression ratio of 0.8 is obtained using its official implementation. $\dagger$ indicates the use of LoRA fine-tuning.

LLaMA-2-7B (Touvron et al., 2023a,b), and OPT-6.7B (Zhang et al., 2022). We use eight datasets in total, including two language modeling datasets, WikiText-2 (Merity et al., 2017) and C4 (Raffel et al., 2020), and six classification datasets: OpenBookQA (Mihaylov et al., 2018), WinoGrande (Sakaguchi et al., 2020), HellaSwag (Zellers et al., 2019), ARC-e (Clark et al., 2018), PIQA (Bisk et al., 2020), and MathQA (Amini et al., 2019).

All artifacts are publicly available and used only for research evaluation under their respective licenses and terms.

Implementation Details. We adopt the activation calibration method proposed by SVD-LLM to correct W_v , so that the low-rank decomposition can better adapt to the input activation distribution. To show that the advantage of DELTA does not rely on sophisticated quantization algorithms, we use the basic RTN method for row-wise and column-wise quantization. For dual-direction quantization, we adopt the iterative algorithm proposed in SINQ (Müller et al., 2025). For a fair comparison, we follow the experimental setup of ASVD and randomly select 256 samples from WikiText-2 as the calibration data. All experiments are conducted on NVIDIA H20 GPUs.

Reported baseline results are normalized according to their relative changes from the corresponding original models to reduce discrepancies across evaluation settings.

5.2 Main Result

We first evaluate DELTA on LLaMA-7B under compression ratios ranging from 20% to 80%, and compare it with existing SVD-based LLM compression methods. As shown in Table 1, DELTA consistently outperforms current state-of-the-art SVD-based methods across all evaluations, demonstrating its effectiveness under high compression ratios.

Under a relatively low compression ratio, DELTA achieves performance comparable to that of the original model. At a 20% compression ratio, DELTA increases perplexity by less than 1% on both WikiText-2 and C4, whereas the previous best SVD-based method incurs an increase of about 7%. Notably, the degradation of the previous best method at a 20% compression ratio is comparable to that of DELTA at a 60% compression ratio, indicating that DELTA maintains strong model performance at substantially higher compression levels.

As the compression ratio further increases, the advantage of DELTA becomes more pronounced. When the compression ratio reaches 40%, ASVD suffers from severe performance collapse. At an 80% compression ratio, both SVD-LLM and Dobi-SVD exhibit significant performance degradation, while DELTA still retains useful model capabilities. These results suggest that DELTA can effectively preserve both language modeling ability and zero-shot task performance even under aggressive compression.

Ratio	Method	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	Avg. ($\uparrow$)	Drop ($\downarrow$)
0.0	Original	0.78	0.57	0.69	0.76	0.43	0.65	0.0%
0.4	LLM-Pruner	0.70	0.41	0.53	0.53	0.27	0.48	26.2%
	SliceGPT	0.65	0.57	0.60	0.43	0.32	0.51	21.5%
	Bonsai	0.72	0.45	0.58	0.59	0.30	0.53	18.5%
	Wanda-sp	0.70	0.42	0.53	0.57	0.29	0.50	23.1%
	DELTA	0.77	0.56	0.67	0.75	0.43	0.63	2.1%
0.5	LLM-Pruner	0.67	0.35	0.52	0.48	0.22	0.45	30.8%
	SliceGPT	0.58	0.46	0.55	0.37	0.28	0.45	30.8%
	Bonsai	0.66	0.40	0.54	0.49	0.26	0.47	27.7%
	Wanda-sp	0.63	0.32	0.53	0.43	0.20	0.42	35.4%
0.6	DELTA	0.76	0.55	0.66	0.72	0.39	0.61	5.2%

Table 2: Comparison with structured pruning methods on LLaMA-2-7B. Accuracy is reported on five zero-shot downstream tasks. The best result in each column is shown in bold. Results for the pruning methods are taken from their original papers.

Group	Method	U/V Bits	Direction	PPL $\downarrow$	Δ vs. Orig. $\downarrow$	CE Gap Recovered $\uparrow$
Baseline	Original	FP16	—	5.68	0.00	100.0%
	Activation-calibrated SVD	FP16	Row	13.73	+8.05	—
	+ INT8 Quantization	INT8	Row	6.26	+0.58	—
	+ INT4 Quantization	INT4	Row	6.16	+0.48	0.0%
DELTA Ablation	INT4 + Direction Search	INT4	Search	6.01	+0.33	30.4%
	Mixed Bit-width + Row Quant.	Mixed	Row	5.89	+0.21	55.2%
	DELTA	Mixed	Search	5.82	+0.14	70.0%

Table 3: Ablation study on LLaMA-7B under the target memory ratio of 0.6. Δ vs. Orig. denotes the PPL increase over the original FP16 model. CE Gap Recovered measures the fraction of the INT4-to-FP16 cross-entropy gap eliminated by each configuration: $(\log \text{PPL}_{\text{INT4}} - \log \text{PPL}_M) / (\log \text{PPL}_{\text{INT4}} - \log \text{PPL}_{\text{FP16}})$.

5.3 Generalization Across Models

To evaluate the generalizability of DELTA across model architectures and scales, we conduct experiments on LLaMA-13B, LLaMA-30B, LLaMA-2-7B, and OPT-6.7B. As shown in Table 4, experiments on LLaMA-13B and LLaMA-30B examine scalability within the LLaMA family, whereas those on LLaMA-2-7B and OPT-6.7B assess the applicability of DELTA across model generations and architectures.

We compare DELTA at target memory ratios of 80% and 60% against existing SVD-based methods. DELTA consistently achieves PPL values close to those of the original models and outperforms the existing SVD-based baselines. For example, under the 80% target memory ratio, DELTA achieves PPL values of 5.09 on LLaMA-13B and 10.98 on OPT-6.7B. These results demonstrate that DELTA generalizes well across model scales and architectures.

5.4 Robustness to Calibration Sampling

Since DELTA uses calibration activations for configuration search, we further evaluate its sensitivity to calibration sampling. We repeat the LLaMA-7B experiments using five independent random seeds (1–5), while keeping the calibration size fixed at 256 samples. At compression ratios of 0.6 and 0.8, DELTA achieves WikiText-2 PPLs of 6.178 ± 0.031 and 10.070 ± 0.028 , respectively. The low variability indicates that DELTA is robust to the choice of calibration samples, even under aggressive compression.

5.5 Comparison with Pruning and Quantization Methods

In addition to SVD-based methods, we compare DELTA with structured pruning (Ma et al., 2023; Ashkboos et al., 2024; Kolawole et al., 2024; Sun et al., 2024) and weight-only quantization. As shown in Table 2, DELTA achieves an average accuracy of 0.63 at a compression ratio of 0.4, with only a 2.1% drop from the original model, substantially outperforming all structured pruning

Method	LLaMA-13B	LLaMA-30B	LLaMA-2-7B	OPT-6.7B
Original	5.09	4.10	5.47	10.87
ASVD	6.74	—	10.10	82.00
SVD-LLM	6.61	5.14	7.73	16.04
SVD-LLM V2	5.46	—	—	13.46
DELTA (0.2)	5.09	4.10	5.48	10.98
DELTA (0.4)	5.12	4.14	6.15	11.85

Table 4: Perplexity comparison across different model families and scales. Baseline methods are evaluated at a compression ratio of 0.2, while DELTA is evaluated at 0.2 and 0.4. The best performance is marked in bold. Most baseline results are reported in the SVD-LLM V2 paper; unavailable results are indicated by dashes.

Method	LLaMA-7B		LLaMA-13B	
	PPL ↓	Size (GB)	PPL ↓	Size (GB)
Original	5.68	12.5	5.09	24.2
RTN-4bit	6.29	3.6	5.53	6.7
DELTA(0.7)	6.84	4.1	5.46	7.6
RTN-3bit	25.54	3.0	11.40	5.4
DELTA(0.8)	10.10	2.9	6.70	5.4

Table 5: Comparison with RTN quantization on WikiText-2. PPL and checkpoint size are reported. The RTN results are reported in the GPTQ paper.

baselines evaluated at the same ratio. Even at the more aggressive compression ratio of 0.6, DELTA still outperforms all pruning baselines evaluated at compression ratios of 0.4 or 0.5.

We further compare DELTA with RTN quantization (Frantar et al., 2022) under comparable checkpoint sizes. As shown in Table 5, RTN-4bit is competitive under a moderate budget, but PPL 3-bit RTN has a PPL of 25.54 on LLaMA-7B, while DELTA achieves 10.10 PPL with a slightly smaller checkpoint. This suggests that DELTA better preserves effective information under aggressive compression.

5.6 Compression and Inference Efficiency

We further evaluate the practical efficiency of DELTA in terms of compression time and inference throughput.

First, we compare the compression time of ASVD, SVD-LLM, Dobi-SVD, and DELTA when compressing LLaMA-7B at an 80% compression ratio on a single NVIDIA H20 GPU. As shown in Figure 2, DELTA reduces the compression cost by an order of magnitude. While existing SVD-based

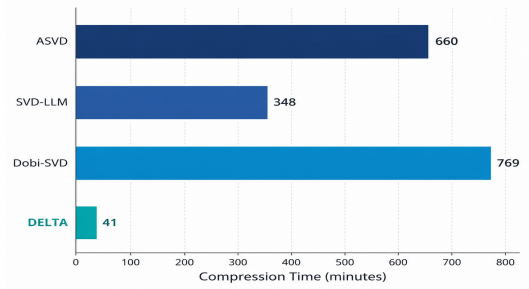


Figure 2: Compression time comparison on LLaMA-7B.

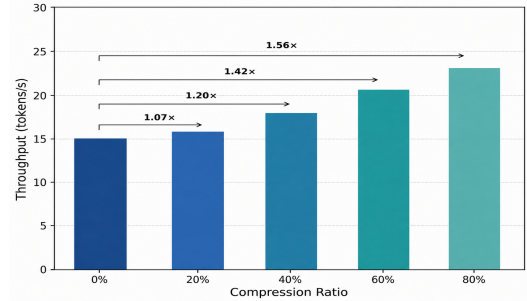


Figure 3: Generation throughput of DELTA-compressed LLaMA-7B under different compression ratios.

methods usually require from several to dozens of GPU-hours, DELTA completes the full compression pipeline within tens of minutes. This efficiency comes from its training-free design: DELTA only performs a local configuration search for each weight matrix and uses layer-wise output reconstruction error as a proxy objective, avoiding repeated end-to-end perplexity evaluations or additional post-training.

Second, we evaluate the generation throughput of DELTA-compressed LLaMA-7B under compression ratios from 0% to 80%. As shown in Figure 3, all models are evaluated on a single NVIDIA H20

GPU with a batch size of 4, a prefill length of 1,024, and a decoding length of 256. Following the evaluation protocol of Dobi-SVD, the quantized low-rank factors are dequantized before throughput measurement, and inference is performed using standard floating-point low-rank matrix multiplications. This setting isolates the computational benefit provided by the low-rank structure and avoids confounding effects from different quantization kernel implementations. DELTA consistently improves throughput as the compression ratio increases, achieving a $1.56\times$ speedup at the 80% compression ratio.

5.7 Ablation Study

To better understand the sources of improvement in DELTA, we conduct an ablation study on LLaMA-7B under the target memory ratio of 0.6, as shown in Table 3.

The results reveal two complementary sources of improvement. First, introducing low-bit representations substantially alleviates the information loss caused by aggressive rank truncation. Activation-calibrated SVD alone increases the PPL from 5.68 to 13.73, whereas applying uniform INT8 and INT4 quantization reduces the PPL to 6.26 and 6.16, respectively. This confirms that quantization enables more singular directions to be retained under the same memory budget and therefore establishes a strong quantized low-rank baseline.

Second, once this strong uniform INT4 baseline is established, explicitly modeling the heterogeneity between the two SVD factors further closes most of the remaining performance gap between the compressed model and the original FP16 model. Direction search reduces the PPL from 6.16 to 6.01, recovering 30.4% of the remaining cross-entropy gap. Mixed bit-width allocation with fixed row-wise quantization further reduces the PPL to 5.89, recovering 55.2% of the gap. Combining both designs, the full DELTA achieves the best PPL of 5.82 and recovers 70.0% of the remaining INT4-to-FP16 cross-entropy gap, leaving only a 0.14 PPL difference from the original model. These results show that factor-wise bit-width allocation and direction search provide complementary benefits beyond uniform low-bit SVD compression.

Proxy Objective. We further evaluate the effectiveness of the activation-aware reconstruction error used for configuration selection. Although the local reconstruction objective cannot theoretically

Proxy Objective	Ratio 0.6	Ratio 0.8
Weight Frobenius Error	6.47	21.42
Activation-aware Error	6.14	10.12

Table 6: WikiText-2 PPL of LLaMA-7B when using different objectives for configuration selection.

guarantee that its ranking of candidate configurations matches the ranking induced by model-level perplexity, minimizing output perturbations on representative calibration inputs is aligned with the goal of preserving downstream model behavior.

To empirically verify this design choice, we replace the activation-aware objective with the plain weight-space Frobenius reconstruction error while keeping the candidate space, compression budget, and all other settings unchanged. As shown in Table 6, at a compression ratio of 0.6, the Frobenius objective yields a PPL of 6.47, whereas the activation-aware objective yields 6.14. At the more aggressive compression ratio of 0.8, the gap becomes substantially larger: the Frobenius baseline deteriorates to a PPL of 21.42, whereas the activation-aware objective maintains a PPL of 10.10. These results demonstrate that incorporating the actual activation distribution provides a substantially more reliable criterion for selecting configurations that preserve end-to-end model performance, especially under aggressive compression.

6 Conclusion

Experiments show that DELTA significantly outperforms existing post-training SVD-based LLM compression methods in perplexity and downstream task performance. Additional experiments on LLaMA-30B and across multiple calibration seeds further demonstrate its scalability and robustness, while proxy-objective ablations validate the importance of activation-aware configuration selection.

Limitations

Although DELTA achieves strong performance at high compression ratios, it still has several limitations. First, DELTA uses Cholesky decomposition to compress the second-order statistics of calibration activations during the search process. This design significantly reduces the cost of error evaluation, but may be affected by numerical stability issues in practice, especially when the activation covariance matrix is poorly conditioned. Future work may explore more stable matrix decomposi-

tion methods or damping strategies to improve the robustness of the search process. Second, DELTA has not yet been fully integrated with more advanced quantization techniques. To highlight the effectiveness of the asymmetric low-rank compression design itself, this paper only adopts the basic RTN method for row-wise and column-wise quantization, and uses the iterative algorithm in SING to implement dual-direction quantization. In the future, DELTA can be further combined with high-performance quantization methods such as GPTQ and AWQ to achieve better performance under lower storage budgets.

Acknowledgments

This work was partially supported by the National Key Research and Development Program of China (2024YFE0204300), National Natural Science Foundation of China (Grant No. 62402311), Natural Science Foundation of Shanghai (Grant No. 24ZR1433700), Key Research and Development Program of Shanghai (25LN3201200), and Alibaba Innovative Research Program. Fangxin Liu and Li Jiang are corresponding authors.

References

- Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. [MathQA: Towards interpretable math word problem solving with operation-based formalisms](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.
- Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennaro Nascimento, Torsten Hoefer, and James Hensman. 2024. [SliceGPT: Compress large language models by deleting rows and columns](#). In *International Conference on Learning Representations*.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. [PIQA: Reasoning about physical commonsense in natural language](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439.
- Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. 2018. [Model compression and acceleration for deep neural networks: The principles, progress, and challenges](#). *IEEE Signal Processing Magazine*, 35(1):126–136.
- Tejalal Choudhary, Vipul Mishra, Anurag Goswami, and Jagannathan Sarangapani. 2020. [A comprehensive survey on model compression and acceleration](#). *Artificial Intelligence Review*, 53(7):5113–5155.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. [Think you have solved question answering? try ARC, the AI2 reasoning challenge](#). Preprint, arXiv:1803.05457.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. [QLoRA: Efficient fine-tuning of quantized LLMs](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115.
- Elias Frantar and Dan Alistarh. 2023. [SparseGPT: Massive language models can be accurately pruned in one-shot](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.
- Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. 2022. [GPTQ: Accurate post-training quantization for generative pre-trained transformers](#). Preprint, arXiv:2210.17323.
- Gene H. Golub, Alan Hoffman, and Gilbert W. Stewart. 1987. [A generalization of the Eckart-Young-Mirsky matrix approximation theorem](#). *Linear Algebra and its Applications*, 88–89:317–327.
- Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. [MiniLLM: Knowledge distillation of large language models](#). In *International Conference on Learning Representations*.
- Yen-Chang Hsu, Ting Hua, Sung-En Chang, Qian Lou, Yilin Shen, and Hongxia Jin. 2022. [Language model compression with weighted low-rank factorization](#). In *International Conference on Learning Representations*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. [DistiLLM: Towards streamlined distillation for large language models](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 24872–24895. PMLR.
- Steven Kolawole, Lucio Dery, Jean-François Kagy, Virginia Smith, Graham Neubig, and Ameet Talwalkar. 2024. [Everybody prune now: Structured pruning of LLMs with only forward passes](#). Preprint, arXiv:2402.05406.
- Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. [AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration](#). In *Proceedings of Machine Learning and Systems*, volume 6, pages 87–100.

- Fangxin Liu, Junjie Wang, Ning Yang, Zongwu Wang, Junping Zhao, Li Jiang, and Haibing Guan. 2025. Aster: Adaptive dynamic layer-skipping for efficient transformer inference via markov decision process. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 11853–11861.
- Fangxin Liu, Ning Yang, Haomin Li, Zongwu Wang, Zhuoran Song, Songwen Pei, and Li Jiang. 2024. Spark: Scalable and precision-aware acceleration of neural networks via efficient encoding. In *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 1029–1042. IEEE.
- Fangxin Liu, Wenbo Zhao, Zhezhi He, Yanzhi Wang, Zongwu Wang, Changzhi Dai, Xiaoyao Liang, and Li Jiang. 2021. Improving neural network efficiency via post-training quantization with adaptive floating-point. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5261–5270. IEEE.
- Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. LLM-Pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 21702–21720.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.
- Lorenz K. Müller, Philippe Bich, Jiawei Zhuang, Ahmet Çelik, Luca Benfenati, and Lukas Cavigelli. 2025. SINO: Sinkhorn-normalized quantization for calibration-free low-precision LLM weights. Preprint, arXiv:2509.22944.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.
- Mingjie Sun, Zhuang Liu, Anna Bair, and Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. Preprint, arXiv:2302.13971.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. Preprint, arXiv:2307.09288.
- Qinsi Wang, Jinghan Ke, Masayoshi Tomizuka, Kurt Keutzer, and Chenfeng Xu. 2025a. Dobi-SVD: Differentiable SVD for LLM compression and some new perspectives. In *International Conference on Learning Representations*.
- Xin Wang, Samiul Alam, Zhongwei Wan, Hui Shen, and Mi Zhang. 2025b. SVD-LLM v2: Optimizing singular value truncation for large language model compression. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4287–4296, Albuquerque, New Mexico. Association for Computational Linguistics.
- Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. 2025c. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *International Conference on Learning Representations*.
- Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 38087–38099. PMLR.
- Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. 2023. ASVD: Activation-aware singular value decomposition for compressing large language models. Preprint, arXiv:2312.05821.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu

Wang, and Luke Zettlemoyer. 2022. [OPT: Open pre-trained transformer language models](#). *Preprint*, arXiv:2205.01068.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, Xiaolei Wang, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, and 4 others. 2026. [A survey of large language models](#). *Frontiers of Computer Science*, 20:2012627.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. [A survey on model compression for large language models](#). *Transactions of the Association for Computational Linguistics*, 12:1556–1577.