DIRECTION-AWARE LOW-RANK WEIGHT FACTORIZATION FOR POST-TRAINING COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-rank factorization is a widely used post-training compression technique that typically minimizes weight reconstruction error. However, preserving reconstruction quality alone does not necessarily preserve model performance: it describes how much the weights are changed, but not whether the induced perturbation moves the model toward lower-loss or higher-loss regions. In neural networks, this distinction is crucial. Under a local first-order view, the loss variation is determined by the interaction between the loss gradient and the factorization perturbation, indicating that perturbation direction is a key factor overlooked by reconstruction-only methods. Motivated by this observation, we propose a direction-aware low-rank factorization framework that encourages low-rank perturbations to follow descent-aligned directions while maintaining compression constraints. We further develop two practical variants that emphasize either performance preservation or stronger compression. As a post-training factorization method, our approach does not require multi-epoch post-factorization finetuning. Experiments on vision models, NLP models, and large language models show that our method consistently achieves a better performance-compression trade-off than existing low-rank baselines.

1 INTRODUCTION

Deep neural networks have achieved remarkable success across vision and language tasks, but their growing model size makes efficient deployment increasingly difficult. Among post-training compression techniques, low-rank factorization remains attractive because it reduces parameters and computation through structured matrix decomposition while preserving the original architecture.

Reconstruction remains a central criterion in post-training low-rank factorization, even as recent methods incorporate task- or data-dependent information. Given a pretrained weight matrix W, they seek a low-rank approximation $\widehat{W}$ that minimizes a reconstruction error such as $\|W - \widehat{W}\|_F$ Kalman (1996); Wei (2021); Hsu et al. (2022); Yuan et al. (2025); Zhang et al. (2025). Even when additional information such as activations or curvature is introduced, the optimization target remains largely centered on recovering the original weights as faithfully as possible. This design implicitly assumes that smaller weight error will translate into smaller performance degradation.

However, for neural networks, this assumption is often insufficient. What matters after factorization is not only *how much* the weights move, but also *which direction* the induced perturbation moves the model on the loss landscape. A perturbation with a small norm can still harm performance if it pushes the model toward a loss-increasing region; conversely, a perturbation of similar magnitude can be much less harmful if it is aligned with a descent-favorable direction.

Figure 1a(left) provides the intuition behind this gap. Standard reconstruction-based factorization constrains the length of the perturbation by minimizing weight error, but it does not explicitly control whether the perturbation moves the model uphill or downhill with respect to task loss. As a result, low-rank approximation may produce a compressed model that remains close to the original weights in Euclidean distance, yet is functionally worse because the perturbation points in an unfavorable direction. Figure 1b(right) shows that this issue is not merely conceptual. It reports the results of SVD-based low-rank approximations at different ranks on representative vision models. Since SVD is the canonical solution for minimizing reconstruction error at a fixed rank, it serves as a direct probe of what reconstruction-only factorization actually optimizes. The figure reveals a clear mismatch: lower

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

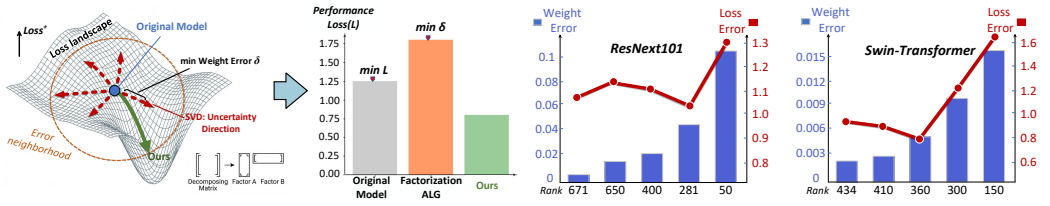


Figure 1: Motivation of direction-aware post-training factorization. (a) Reconstruction error controls perturbation magnitude, but task loss also depends on perturbation direction; (b) SVD-based rank sweeps show that lower weight error does not necessarily yield lower loss.

reconstruction error does not necessarily yield lower task loss. For example, in Swin-Transformer, the SVD solution at rank 360 has larger weight error than the one at rank 410, yet achieves lower loss. This observation suggests that reconstruction magnitude alone is an incomplete criterion for post-training factorization, and that the missing factor is the direction of the factorization perturbation with respect to the loss landscape.

To make this missing factor explicit, we view factorization as introducing a perturbation $\delta = \widehat{W} - W$ to a pretrained model and analyze its local effect on the loss. Under a first-order approximation, the loss variation is determined by the interaction between the loss gradient and the perturbation. This immediately explains why reconstruction-only factorization may fail: preserving performance is not only about keeping $\|\delta\|$ small, but also about ensuring that δ is oriented in a loss-favorable direction.

Motivated by this view, we propose a direction-aware low-rank factorization framework for post-training compression. Instead of treating the factorization error as purely destructive noise, our method uses gradient information to steer low-rank perturbations toward descent-aligned directions under compression and local-neighborhood constraints. In this way, factorization is guided not only by parameter approximation quality, but also by its predicted impact on downstream loss.

We instantiate this idea in two practical modes. The first is a **loss-preserving** mode, which prioritizes performance preservation at a target compression level. The second is a **compact** mode, which prioritizes stronger compression by seeking the smallest feasible rank that still yields favorable loss behavior. Both modes are built on the same principle: low-rank perturbations should be judged not only by how small they are, but also by whether their direction is compatible with loss preservation.

Our contributions are summarized as follows:

1) We identify the signed direction of the complete factorization residual as an explicit and independently measurable factor in post-training low-rank compression, complementing reconstruction quality, task-aware decomposition, and rank allocation.

2) We propose a factorization framework that rectifies low-rank perturbations according to their first-order effect on task loss, supporting both performance-first and compression-first modes.

3) We demonstrate consistent gains across vision, NLP, and large language models. Experiments show that our method achieves a better performance–compression trade-off than reconstruction-only low-rank baselines, without requiring multi-epoch post-factorization finetuning.

2 Related Work

An intuitive way to distinguish prior low-rank methods from ours is to separate approximation quality from loss effect. In post-training low-rank compression, a pretrained weight matrix is replaced by a low-rank surrogate $W \approx \widehat{W} = LR^\top$, and the classical objective is

$$\min_{L,R} \|W - \widehat{W}\|_F^2. \tag{1}$$

Most existing methods Kim et al. (2015); Denton et al. (2014) improve low-rank compression by making the surrogate $\widehat{W}$ a better approximation of W under Eq. (1). Our work is motivated by a different observation: even when $\widehat{W}$ is a better approximation in reconstruction, the induced

perturbation $\delta = \widehat{W} - W$ can still be harmful if it pushes the model toward higher task loss. Under a local first-order view, $\Delta\mathcal{L} \approx \langle \nabla_W \mathcal{L}, \delta \rangle$, so the key issue is not only how small the perturbation is, but also whether its direction is favorable to the task objective.

Classical low-rank factorization. Classical low-rank compression treats factorization primarily as a reconstruction problem, with truncated SVD serving as the canonical fixed-rank solution Dax et al. (2014); Zhang et al. (2023); Wei (2021). This view is simple and broadly applicable, but reconstruction error only measures the magnitude of the induced perturbation. A smaller $\|\widehat{W} - W\|_F$ does not necessarily imply a smaller change in task loss, because perturbations of similar magnitude can have different effects depending on their directions in the loss landscape. Our work starts from this distinction between perturbation magnitude and its loss-relevant direction.

Task-sensitive post-training factorization. Recent low-rank methods go beyond plain SVD by incorporating task- or data-dependent information into factorization. FWSVD Hsu et al. (2022) uses Fisher-weighted reconstruction, DRONE Chen et al. (2021) exploits data-aware low-dimensional representations, ASVD Yuan et al. (2025) uses activation statistics and layer sensitivity, and SVD-LLM Wang et al. (2025) introduces truncation-aware whitening and layer-wise updates. BALF González-Martínez (2025) further considers budget-aware rank allocation.

Overall, these task-sensitive methods can implicitly favor more loss-preserving perturbations through reconstruction weighting, sensitivity estimation, truncation, or rank allocation. This implicitly reflects the importance of perturbation direction, but the directional effect itself is not explicitly modeled. Our work makes this factor explicit by characterizing the signed interaction between the factorization residual and the task-loss gradient.

3 Direction-Aware Weight Factorization

The central idea of our method is that post-training factorization should control not only the magnitude of the induced perturbation, but also its direction relative to the loss gradient.

3.1 From Reconstruction Error to Loss Direction

For a pretrained weight matrix W, low-rank factorization replaces it with a surrogate $\widehat{W} = LR^\top$, which induces a perturbation $\delta = \widehat{W} - W$. Existing post-training low-rank methods mainly assess this perturbation through reconstruction quality, such as $\|W - \widehat{W}\|_F$. However, reconstruction quality alone does not reveal whether the induced perturbation is favorable or harmful to the task objective.

To expose this missing factor, we analyze the loss locally around the pretrained model. Let $G = \nabla_W \mathcal{L}(W)$ denote the gradient of the task loss with respect to W. Under a first-order approximation, the loss variation caused by the perturbation can be written as

$$\mathcal{L}(W + \delta) - \mathcal{L}(W) \approx \langle G, \delta \rangle. \tag{2}$$

This expression clarifies the desired direction of compression noise. If $\langle G, \delta \rangle > 0$, then the perturbation has a positive component along the gradient and tends to move the model toward higher loss. If $\langle G, \delta \rangle < 0$, then the perturbation points against the gradient and is locally descent-favorable. Therefore, for post-training low-rank factorization, the key issue is not only how small the perturbation is, but also whether it suppresses its component along $+G$ and moves instead toward $-G$.

First-order signal under local control. The local effect of a factorization perturbation can be decomposed into a signed first-order term and a remainder controlled by the perturbation scale. Assume that, with other layers fixed, $\mathcal{L}$ has β_l-Lipschitz gradient with respect to W_l inside the layer-adaptive neighborhood. For any feasible perturbation δ_l with $\|\delta_l\|_F \le \epsilon_l$, Taylor theorem gives

$$\mathcal{L}(W_{-l}, W_l + \delta_l) - \mathcal{L}(W_{-l}, W_l) = \langle G_l, \delta_l \rangle + R_l(\delta_l), \qquad |R_l(\delta_l)| \le \frac{\beta_l}{2} \|\delta_l\|_F^2. \tag{3}$$

This decomposition motivates our design: the layer-adaptive neighborhood limits the perturbation scale, while rectification reduces the signed component along the local loss-increasing direction. We do not estimate the curvature-dependent remainder; instead, we use the first-order term as a lightweight local criterion under the controlled neighborhood constraint.

For the factorized weight $\widehat{W} = LR^\top$, we use $a(L, R) = \langle G, LR^\top - W \rangle$ to measure the loss-directional component of the induced perturbation. A smaller value of $a(L, R)$ means that the induced perturbation has a weaker component along the loss-increasing direction.

3.2 Feasible Region: Compression and Layer-Adaptive Locality

The gradient-alignment criterion is useful only when the factorized model remains in a local region around the pretrained solution. In practice, we enforce this through a layer-adaptive neighborhood:

$$\|LR^\top - W_l\|_F \le \epsilon_l, \qquad \epsilon_l = \gamma \|W_l\|_F, \tag{4}$$

where l indexes the layer and $\gamma \in (0, 1)$ is a global ratio. This avoids using a single absolute threshold for all layers and instead scales the admissible perturbation range according to each layer's own magnitude. At the same time, the factorized form must provide actual compression. Therefore, we only consider compression-effective ranks, i.e., ranks for which the low-rank parameterization uses fewer parameters than the original dense layer.

Combining these two requirements, our design principle becomes the following: among low-rank factorizations that satisfy the compression constraint and remain inside the layer-adaptive local region, we prefer those whose perturbations are more strongly opposed to the gradient. This leads to Eq. 5:

$$\min_{L,R} \langle G_l, LR^\top - W_l \rangle \quad \text{s.t.} \quad \|LR^\top - W_l\|_F \le \epsilon_l,\ \mathrm{rank}(LR^\top) \le k. \tag{5}$$

Eq. (5) is used as a local rectification criterion: starting from an existing low-rank factorization, it indicates how the factors should be corrected to reduce the loss-increasing component while remaining compression-feasible and locally valid. Thus, our method should be interpreted as a cheap first-order correction to low-rank factorization.

3.3 Initialize-and-Rectify

Directly solving Eq. (5) is unnecessary for post-training compression. Instead, we adopt an initialize-and-rectify strategy.

Initialization. We begin from a reconstruction-strong low-rank approximation at the chosen rank. For SVD-based initialization, given $W_k = U_k \Sigma_k V_k^\top$, we use the balanced factor form $L = U_k \Sigma_k^{1/2}$ and $R = V_k \Sigma_k^{1/2}$. This fixed scaling removes ambiguity from equivalent factorizations such as $(cL)(R/c)^\top$ and makes rectification reproducible for a given initializer. Our method can still be applied on top of different low-rank initializers; once an initializer is chosen, we use its balanced factor form for rectification.

Directional rectification. Starting from the initialized factors, we compute the perturbation and evaluate its gradient alignment. If the alignment is already non-positive, the perturbation is first-order descent-aligned, meaning that its signed linearized loss signal is non-increasing, and we stop under the first-order criterion. Otherwise, we update the factors so that the perturbation becomes less aligned with $+G$ and more aligned with $-G$.

Concretely, we perform a gradient step on the gradient-alignment quantity in factor space:

$$L' = L - \eta\, GR, \qquad R' = R - \eta\, G^\top L, \tag{6}$$

where $\eta > 0$ is the step size. This update has a direct interpretation: it reduces the inner product between the perturbation and the gradient, thereby suppressing the component of the perturbation along the loss-increasing direction. The update is justified by the following proposition.

Proposition 3.1 (Local descent of gradient alignment)**.** *Let $a(L, R) = \langle G, LR^\top - W \rangle$. In any bounded factor region visited by rectification, there exists a step-size bound $\eta_0 > 0$ such that the update in Eq.* (6) *satisfies $a(L', R') \le a(L, R)$ for any $0 < \eta \le \eta_0$. If $(GR, G^\top L) \ne 0$, the decrease is strict to first order. Moreover, the rank constraint is preserved since $L'(R')^\top$ has rank at most k.*

Proof sketch. The gradients of $a(L, R)$ are $\nabla_L a = GR$ and $\nabla_R a = G^\top L$, so Eq. (6) is exactly a gradient descent step on $a(L, R)$ in factor space. By the standard descent lemma, a sufficiently small step size decreases $a(L, R)$. The update changes only the values of the two factors, not their dimensions, so the reconstructed matrix remains rank at most k. □

Proposition 3.1 shows that rectification decreases the signed first-order component in this local decomposition, while the neighborhood constraint limits the residual scale.

Neighborhood-preserving correction. Because rectification is intended to operate inside the layer-adaptive local region, every tentative update must also satisfy Eq. (4). In practice, if a proposed update leaves the admissible region, we shrink the step size and retry. This backtracking-style mechanism keeps rectification local while still allowing the perturbation direction to improve.

3.4 Loss-Preserving Mode

As shown in Algorithm 1, our first operating mode prioritizes performance preservation. In this setting, the target rank is specified in advance by the compression budget. We first compute a low-rank factorization at that rank, and then apply only a small number of rectification steps when the perturbation has a positive component along the gradient.

It stays close to the reconstruction-oriented starting point and uses descent-aligned only to eliminate perturbation directions that would otherwise increase loss. As a result, this mode is suitable when the primary goal is to preserve performance under a prescribed compression level.

Algorithm 1 Loss-Preserving Mode

Input: Weight W, gradient G, rank k, radius ϵ_l, step η, tolerance τ, max iters $T_{\max}$
Output: Low-rank factors L, R

Compute rank-k SVD and initialize balanced factors $L = U_k\Sigma_k^{1/2}$, $R = V_k\Sigma_k^{1/2}$
for $t = 1, \ldots, T_{\max}$ **do**
  $\delta \leftarrow LR^\top - W$
  $a \leftarrow \langle G, \delta \rangle$
  if $a \le \tau$ **then break**
  Compute tentative update L', R' using Eq. (6)
  $\delta' \leftarrow L'(R')^\top - W$
  while $\|\delta'\|_F > \epsilon_l$ **do**
    $\eta \leftarrow \eta/2$
    Recompute L', R' and δ'
  end while
  $a' \leftarrow \langle G, \delta' \rangle$, $\Delta a \leftarrow a - a'$
  if $\Delta a \le \tau$ **then break**
  $L \leftarrow L'$, $R \leftarrow R'$
end for
return L, R

Algorithm 2 Compact Mode

Input: Model M, gradients $\{G_l\}$, ratio γ, step η, tolerance τ, max iters $T_{\max}$
Output: Compressed model $\widehat{M}$

for each compressible layer $W_l \in M$ **do**
  $\epsilon_l \leftarrow \gamma\|W_l\|_F$
  Find the smallest admissible rank k and initialize balanced factors $L = U_k\Sigma_k^{1/2}$, $R = V_k\Sigma_k^{1/2}$
  for $t = 1, \ldots, T_{\max}$ **do**
    $\delta \leftarrow LR^\top - W_l$, $a \leftarrow \langle G_l, \delta \rangle$
    if $a \le \tau$ **then break**
    Compute tentative update L', R' using Eq. (6)
    while $\|L'(R')^\top - W_l\|_F > \epsilon_l$ **do**
      $\eta \leftarrow \eta/2$ and recompute L', R'
    end while
    if the alignment improvement is below τ **then break**
    $L \leftarrow L'$, $R \leftarrow R'$
  end for
  Replace W_l with $LR^\top$
end for
return $\widehat{M}$

3.5 Compact Mode

As shown in Algorithm 2, our second operating mode prioritizes stronger compression. Instead of fixing the rank in advance, we first identify the smallest admissible rank whose initial factorization still satisfies the layer-adaptive neighborhood constraint. This gives the most aggressive compression level that remains locally valid under our framework.

Starting from this minimum admissible rank, we iteratively rectify the factors until the perturbation no longer has a positive component along the gradient, improvement becomes negligible, or the iteration budget is reached. Compared with loss-preserving mode, compact mode starts from a more aggressive compression point, so rectification plays a larger role. Its purpose is to turn the most compact admissible factorization into one whose perturbation opposes the gradient.

When a target parameter or compute budget is required, we use γ as a global budget-control variable. For each candidate γ, all layers determine their admissible ranks independently, and the resulting total parameters are computed. We then perform an outer binary search over γ and choose the smallest value that satisfies the target budget, which gives feasible solution under the requested compression.

3.6 Algorithm Complexity and Cost

Our method is an offline local correction on top of standard post-training factorization rather than a solver for the full non-convex compression problem. Starting from a reconstruction-faithful low-rank initializer, rectification only adjusts the factors to suppress the first-order loss-increasing component while preserving the rank and local neighborhood. For $W \in \mathbb{R}^{N\times M}$ with rank k, each rectification step costs $O(NMk)$, and only a small number of steps is typically required. The remaining overhead

is one calibration backward pass and the initial decomposition, both incurred offline. Compared with second-order alternatives, our method avoids explicit Fisher/Hessian estimation and multi-epoch post-factorization finetuning. Detailed higher-order analysis is provided in Appendix.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate our method on three representative architecture families: decoder-only large language models Touvron et al. (2023a;b); Grattafiori et al. (2024); Yang et al. (2025), vision backbones Vaswani et al. (2017); He et al. (2016); Dosovitskiy et al. (2020); Todi et al. (2023); Liu et al. (2021); Tu et al. (2022); Huang et al. (2017), and encoder-based NLP models. For LLMs, we evaluate LLaMA and Qwen-family models and report perplexity on WikiText-2/PTB/C4 Merity et al. (2016); Marcus et al. (1993); Raffel et al. (2023), together with zero-shot downstream Mihaylov et al. (2018); Sakaguchi et al. (2021); Bisk et al. (2020); Clark et al. (2018); Zellers et al. (2019); Amini et al. (2019) accuracy on standard reasoning and commonsense benchmarks. For vision models, we evaluate both Transformer-based and convolutional backbones on ImageNet-1K Deng et al. (2009). For encoder-based NLP models, we evaluate BERT on MNLI Williams et al. (2018). We also use SWAG Zellers et al. (2018) to measure the practical cost of constructing the compressed model.

Protocol. Our method is a post-training factorization procedure and does not require multi-epoch post-factorization finetuning. For each model, we compute the task gradient once on a calibration set and then perform low-rank factorization with directional rectification. Unless otherwise stated, we use the same optimization settings across experiments and do not perform task-specific tuning. All experiments are implemented in PyTorch and run on an NVIDIA A800 GPU.

We compute one-shot calibration gradients using the standard task loss for each modality. ImageNet models use supervised cross-entropy with ground-truth labels and randomly sampled training images. Encoder-based NLP models use the corresponding supervised classification loss for MNLI/SWAG. LLMs use next-token cross-entropy on WikiText calibration text with shifted-token labels. Unless otherwise stated, we use 512 calibration images for vision models, 256 calibration examples for NLP models, and 256 WikiText sequences for LLMs. Unless otherwise stated, we use $\gamma = 0.05$, $\eta = 10^{-4}$, $\tau = 0$, and $T_{\max} = 50$ for all experiments. Calibration data are used only for one-shot gradient computation and are excluded from final evaluation.

Table 1: LLM results under post-training low-rank compression at two parameter ratios. We report perplexity and zero-shot downstream accuracy.

Ratio	Method	Wiki2 ↓	PTB ↓	OpenB.	Arc-e	Arc-c	WinoG.	PIQA	MathQA	HellaS.	Avg. ↑
1.0	LLaMA-7B	5.68	8.34	0.34	0.75	0.42	0.69	0.79	0.27	0.57	0.55
0.8	ASVD	11.14	16.55	0.25	0.53	0.27	0.64	0.68	0.24	0.41	0.43
	SVD-LLM	7.94	16.22	0.22	0.58	0.29	0.63	0.69	0.24	0.43	0.44
	Dobi-SVD	8.54	14.83	0.26	0.59	0.31	**0.66**	0.70	0.23	0.44	0.46
	Dip-SVD	7.95	15.60	0.27	0.63	0.33	0.64	0.71	0.24	0.45	0.47
	Ours	**7.06**	**14.69**	**0.29**	**0.67**	**0.35**	0.65	**0.72**	**0.25**	**0.47**	**0.49**
0.6	ASVD	1407	3292	0.13	0.28	0.22	0.48	0.55	0.19	0.26	0.30
	SVD-LLM	13.11	63.75	0.19	0.42	0.25	0.58	0.60	0.21	0.33	0.37
	Dobi-SVD	13.54	46.38	0.22	0.41	0.27	0.58	0.61	0.23	0.34	0.38
	Ours	**10.40**	**45.11**	**0.23**	**0.51**	**0.30**	**0.62**	**0.65**	**0.23**	**0.37**	**0.42**

4.2 MAIN RESULTS ACROSS ARCHITECTURES

We first verify whether direction-aware factorization works across three representative architectures: decoder-only LLMs, vision models, and encoder-based NLP models.

Decoder-only LLMs. Table 1 compares our method with ASVD Yuan et al. (2025), SVD-LLM Wang et al. (2025), and recent SVD-based LLM compression baselines Qinsi et al. (2025); Ding et al. (2025) on LLaMA-7B. At parameter ratio 0.8, our method achieves the lowest Wiki2 and PTB perplexity, together with the best average zero-shot accuracy. At the more aggressive ratio 0.6, all methods degrade more substantially, but our method remains the most stable, again obtaining the lowest perplexity and the highest average downstream accuracy. These results show that directional

rectification is particularly useful in LLM compression, where reconstruction-only perturbations can easily cause large degradation.

Vision models. Table 2 compares our compact mode with existing post-training baselines Hsu et al. (2022); Guo et al. (2024); Yuan et al. (2025) on ConvNeXt-Base and ResNet-50 Chen et al. (2023); Narshana et al. (2023); González-Martínez (2025). Under similar computation or parameter budgets, our method achieves the best accuracy-efficiency trade-off on both models. This indicates that the benefit of direction-aware factorization is not limited to autoregressive LLMs, but also applies to convolutional and modern vision architectures.

Table 2: Comparison with existing post-training baselines on ConvNeXt-Base and ResNet-50.

ConvNeXt-Base	Top-1 ↑	GFLOPs	Params (M)	ResNet-50	Top-1 ↑	GFLOPs	Params (M)
Original	85.8	30.8	88.6	Original	77.39	4.1	25.6
PELA	19.6	15.9	46.6	IFM	75.93	–	24.1
FW-SVD	52.4	15.6	47.4	DFPC	71.61	–	22.8
ASVD	54.1	15.6	46.9	BALF	76.27	3.3	20.48
Ours	**55.8**	15.6	**46.6**	Ours	**76.62**	**3.1**	**18.69**

Encoder-based NLP models. Table 3 middle evaluates compressed BERT on MNLI. Ours preserves the original BERT performance and outperforms existing methods Wei (2021); Zhang et al. (2023); Chen et al. (2021) on both validation and test accuracy. Together with the LLM and vision results, this establishes that direction-aware factorization works across decoder-only LLMs, vision models, and encoder-based NLP models.

Table 3: Comparison with structured pruning, BERT compression, and extreme low-rank compression. Left: LLaMA-7B WikiText2 perplexity under different memory budgets against structured pruning methods. Middle: BERT results on MNLI. Right: perplexity under extreme compression at ratio 0.4.

LLaMA-7B on WikiText2 PPL ↓					BERT on MNLI			Extreme Compression on WikiText2 ↓			
Method	10GB	9GB	8GB	7GB	Method	Val Acc. ↑	Test Acc. ↑	Model	Method	PPL	Ratio
LLM-Pruner	9.88	12.21	18.94	21.68	Original	82.77	83.91	LLaMA2-7B	ERC-SVD	60.08	0.4
SliceGPT	8.78	12.73	16.39	27.41	SVD	81.69	82.65		ZS-SVD	45.52	0.4
BlockPruner	9.40	12.76	19.78	43.05	CANDECOMP	81.46	81.54		Ours	**45.02**	**0.4**
NSVD	7.92	–	–	13.74	DRONE	82.15	83.07	LLaMA3-8B	OBD-LLM	167.3	0.4
Ours	**7.06**	**8.24**	**8.95**	**10.39**	Ours	**82.78**	**83.92**		Ours	**144.31**	**0.4**

4.3 Generalization and Broader Comparisons

We further examine whether the gains generalize across model families, whether the two operating modes capture different performance–compression trade-offs, and whether the method remains competitive compared to other types of compression methods.

Extreme compression. We evaluate stability under an aggressive compression ratio of 0.4, where low-rank perturbations become much larger and reconstruction-only methods are more likely to collapse. As shown in the right block of Table 3, our method achieves the lowest perplexity on both LLaMA2-7B and LLaMA3-8B. On LLaMA2-7B, Ours slightly improves over ZS-SVD Abbasi et al. (2026) and clearly outperforms ERC-SVD Bai et al. (2026). On the more challenging LLaMA3-8B, OBD-LLM Li et al. (2026) uses curvature information to assess compression sensitivity, whereas our method directly rectifies the signed direction of the induced low-rank residual. Our method achieves lower PPL than OBD-LLM, reducing it from 167.3 to 144.31 without high-order estimation. This distinction shows that our goal is not to estimate where the model is sensitive, but to correct where the compression perturbation moves. The results suggest that direction-aware residual rectification remains effective under extreme compression. More details on high-order terms are in **Appendix**.

Diverse LLMs. Table 4 extends the evaluation to LLaMA2-7B/13B, LLaMA3-8B, and Qwen3-8B. Across all model families, our method improves perplexity over SVD-LLM on both Wiki2 and C4. This suggests that the direction-aware correction is not tied to a specific LLM architecture.

Vision backbones and Algorithm modes. Table 5 reports both CE and accuracy on ImageNet-1K. Because our method optimizes a local loss-directional surrogate, CE provides a direct task-loss proxy, while Top-1/Top-5 accuracy verifies actual performance. Ours(L) consistently reduces CE relative

Table 4: Perplexity comparison across LLM families under post-training low-rank compression.

Method	LLaMA2-7B		LLaMA2-13B		LLaMA3-8B		Qwen3-8B	
	Wiki2	C4	Wiki2	C4	Wiki2	C4	Wiki2	C4
Dense	5.47	7.28	4.88	6.73	6.24	9.57	9.71	15.52
SVD-LLM	8.41	20.34	6.65	14.99	14.16	79.14	37.52	47.25
Ours	**7.02**	**17.84**	**6.11**	**13.56**	**12.43**	**67.39**	**35.10**	**44.34**

to SVD at comparable compression levels and preserves accuracy, indicating that the surrogate improvement translates to real task behavior. Ours(C) instead selects smaller feasible ranks, achieving stronger parameter reduction with an expected accuracy trade-off. These results show that the two modes target distinct points on the performance–compression trade-off.

Table 5: ImageNet-1K results on ViT and CNN backbones. Ours(L): Loss-Preserving Mode. Ours(C): Compact Mode. Param denotes parameter reduction. CE denotes cross-entropy(%).

ViT-based Models	CE ↓	Top-1 ↑	Top-5 ↑	Param ↓	Conv. Models	CE ↓	Top-1 ↑	Top-5 ↑	Param ↓
ViT_B	83.84	81.070	95.310	–	ResNeXt101	92.66	79.312	94.526	–
ViT_B-SVD	83.96	80.761	95.190	-3.0%	ResNeXt101-SVD	87.01	78.154	94.012	-75.0%
ViT_B-Ours(L)	**83.80**	**81.007**	**95.311**	-3.0%	ResNeXt101-Ours(L)	**86.91**	**78.160**	**94.018**	-75.0%
ViT_B-Ours(C)	83.85	81.038	95.290	**-41.0%**	ResNeXt101-Ours(C)	92.45	76.308	91.482	**-81.0%**
MaxViT	73.93	83.390	96.590	–	DenseNet169	99.78	75.600	92.806	–
MaxViT-SVD	74.03	83.327	96.585	-30.0%	DenseNet169-SVD	100.14	75.426	**92.802**	-2.0%
MaxViT-Ours(L)	**73.60**	**83.341**	**96.593**	-30.0%	DenseNet169-Ours(L)	**97.18**	**75.446**	92.790	-2.0%
MaxViT-Ours(C)	73.91	82.233	94.490	**-67.0%**	DenseNet169-Ours(C)	99.48	74.790	92.572	**-52.0%**
Swin_T	73.97	82.788	95.610	–	ConvNeXt_L	77.09	84.120	96.870	–
Swin_T-SVD	74.11	82.771	95.589	-2.0%	ConvNeXt_L-SVD	88.09	84.103	96.720	-2.0%
Swin_T-Ours(L)	**73.66**	82.790	**95.590**	-2.0%	ConvNeXt_L-Ours(L)	**77.08**	**84.115**	**96.876**	-2.0%
Swin_T-Ours(C)	74.02	**82.800**	94.844	**-28.0%**	ConvNeXt_L-Ours(C)	77.09	82.344	96.866	**-33.0%**

Comparison with structured pruning compression. Table 3 left compares our method with existing structured pruning methods Ma et al. (2023); Ashkboos et al. (2024); Zhong et al. (2025); Lu et al. (2025) on LLaMA-7B across memory budgets. Our method achieves the lowest WikiText2 PPL at every budget. The advantage grows under stricter memory constraints, where several pruning baselines degrade sharply. This indicates that direction-aware factorization is not only stronger than low-rank baselines, but also competitive with pruning for memory-constrained LLM compression.

4.4 Ablation and Robustness

We now examine whether the gains come from the proposed direction-aware design and whether the required one-shot calibration is stable here, and provide additional details in the **Appendix**.

Directional rectification. The left block of Table 6 compares four variants: no direction control, random direction, direct factor finetuning, and our directional rectification. Random direction severely degrades accuracy, showing that arbitrary perturbation steering is harmful. Direct factor finetuning improves over w/o Dir., but remains below our method on both ViT_B and ResNeXt101. This indicates that the gain comes not merely from adding an update step, but from explicitly correcting the low-rank perturbation toward the loss-favorable direction.

Rank–loss behavior. The left and middle panels of Figure 2 compare the loss–rank curves of SVD and our method. At the same rank, our method reaches lower loss, showing that directional rectification changes the functional effect of the low-rank perturbation. The curves also illustrate the two modes: one selects a loss-preserving rank, while the other searches for a compact feasible rank.

Adaptive local neighborhood. The right panel of Figure 2 shows that different layers exhibit substantially different perturbation tolerance. Some layers collapse when the noise scale reaches 10^{-2}, while others remain stable until much larger perturbations. This indicates that a single global perturbation bound cannot provide a reliable local region for all layers. By setting $\epsilon_l = \gamma \|W_l\|_F$, our method assigns each layer a scale-aware neighborhood, allowing directional rectification to suppress gradient-aligned perturbations without leaving the layer's valid local range.

Sensitivity to γ. The right block of Table 6 shows how the neighborhood ratio γ controls the compression–stability trade-off. Larger γ allows stronger parameter reduction, but also increases

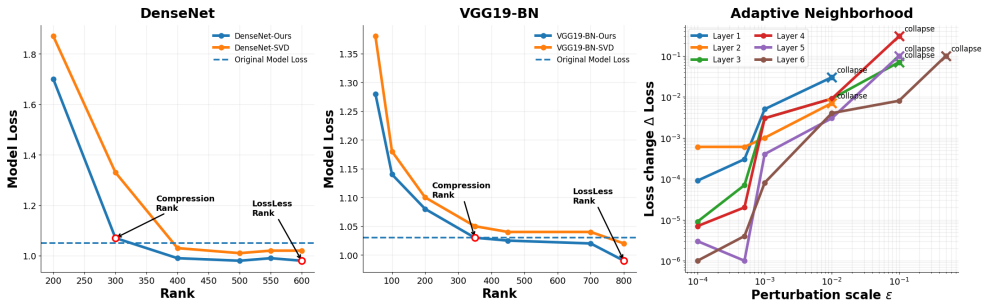


Figure 2: Left and middle: compared with SVD, our method reaches lower loss at the same rank and identifies both a compression-oriented rank and a loss-preserving rank. Right: different layers exhibit markedly different collapse thresholds, supporting the use of a layer-adaptive neighborhood.

rectification iterations and eventually hurts accuracy. A moderate value, such as $\gamma = 0.05$, provides a better balance between compression strength and performance preservation.

Calibration robustness. The middle block of Table 6 evaluates sensitivity to calibration size and data corruption. Performance improves rapidly from 150 to 500 calibration samples and then saturates around 500–1000 samples. Under corrupted calibration data, the accuracy decreases smoothly rather than collapsing, even with input or label noise. These results suggest that the one-shot gradient estimate is practically robust and does not require unusually large or perfectly clean calibration data.

Table 6: Ablation and robustness analysis. We evaluate directional rectification and direct factor finetuning, calibration robustness, and sensitivity to the neighborhood ratio γ.

Direction	Top-1 Acc. ↑	
	ViT_B	ResNeXt101
w/o Dir.	78.37±0.02	75.66±0.04
Random Dir.	71.92±0.05	68.00 ±0.07
Factor-FT	79.05±0.03	76.74 ±0.06
w/ Dir.	**81.01**±0.03	**78.16**±0.05

Calib. Size		Data Quality	
#	Acc.	Noise	Acc.
150	69.11±0.09	Input 10%	74.88±0.01
500	76.60±0.03	Input 20%	74.01±0.03
1000	**76.61**±0.04	Label 10%	73.75±0.07
5000	76.60±0.03	Label 20%	72.30±0.06

γ	Param.	Top-1	Iters
0.02	16.5%	77.03±0.03	3.0
0.05	30.0%	**76.62**±0.02	5.5
0.10	39.4%	74.91±0.06	9.0
0.15	48.8%	70.94±0.07	15.5

4.5 Cost and Efficiency

We evaluate whether the direction-aware rectification introduces practical overhead. Table 7 reports both construction cost and deployment efficiency. On SWAG, our method requires much less time and data than post-factorization baselines while achieving lower CE. On LLaMA-7B, the compressed model reduces parameters and MACs, leading to lower latency and higher throughput. These results show the proposed method is efficient both when constructing the compressed model and deployment.

Table 7: Cost and efficiency analysis. Left: post-training factorization cost on SWAG. Right: inference efficiency on LLaMA-7B. Results with ± are averaged over 5 runs; tok/s denotes tokens per second.

Method	CE ↓	Acc. ↑	Time (min) ↓	Data (M)
SVD_ft	59.10 ± 0.21	78.44 ± 0.05	134.6	27
CANDECOMP	64.03 ± 0.17	78.56 ± 0.02	151.0	27
Ours(L)	**56.64** ± 0.18	**78.58** ± 0.03	11.4	**7.6**
Ours(C)	57.65 ± 0.20	78.31 ± 0.05	**10.3**	**7.6**

Method	Params	MACs	Latency ↓	Tok/s ↑
Original	6.74B	6.61G	13.2089 ± 1.2209	1338.37
Ours	4.85B	4.90G	**10.2659** ± 1.2463	**1541.88**
Gain	**-28.0%**	**-25.9%**	**-22.3%**	**+15.2%**

5 Conclusion and Limitations

We propose a direction-aware framework for post-training low-rank factorization. Instead of relying only on reconstruction error, our method rectifies the induced low-rank perturbation according to its first-order effect on task loss, with variants for performance preservation and compact compression. Experiments across vision, NLP, and LLM benchmarks show improved performance–compression trade-offs without multi-epoch post-factorization finetuning. Our current evaluation mainly focuses on unimodal models; extending the framework to multimodal architectures is an important direction.

REFERENCES

Ali Abbasi, Chayne Thrash, Haoran Qin, Shansita Sharma, Sepehr Seifi, and Soheil Kolouri. Zero sum svd: Balancing loss sensitivity for low rank llm compression, 2026. URL https://arxiv.org/abs/2602.02848.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 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)*, pp. 2357–2367, 2019.

Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. Slicegpt: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*, 2024.

Haolei Bai, Siyong Jian, Tuo Liang, Yu Yin, and Huan Wang. Erc-svd: Error-controlled svd for large language model compression, 2026. URL https://arxiv.org/abs/2505.20112.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Patrick Chen, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. Drone: Data-aware low-rank compression for large nlp models. *Advances in neural information processing systems*, 34:29321–29334, 2021.

Yiting Chen, Zhanpeng Zhou, and Junchi Yan. Going beyond neural network feature similarity: The network feature complexity and its interpretation using category theory, 2023. URL https://arxiv.org/abs/2310.06756.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Achiya Dax et al. Low-rank positive approximants of symmetric matrices. *Advances in Linear Algebra & Matrix Theory*, 4(3):172–185, 2014.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Emily Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *CoRR*, abs/1404.0736, 2014. URL http://arxiv.org/abs/1404.0736.

Xuan Ding, Rui Sun, Yunjian Zhang, Xiu Yan, Yueqi Zhou, Kaihao Huang, Suzhong Fu, Chuanlong Xie, and Yao Zhu. Dipsvd: Dual-importance protected svd for efficient llm compression, 2025. URL https://arxiv.org/abs/2506.20353.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

David González-Martínez. Balf: Budgeted activation-aware low-rank factorization for fine-tuning-free model compression, 2025. URL https://arxiv.org/abs/2509.25136.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian et.al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Yangyang Guo, Guangzhi Wang, and Mohan Kankanhalli. Pela: Learning parameter-efficient models with low-rank approximation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15699–15709, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. Language model compression with weighted low-rank factorization. *arXiv preprint arXiv:2207.00112*, 2022.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Dan Kalman. A singularly valuable decomposition: the svd of a matrix. *The college mathematics journal*, 27(1):2–23, 1996.

Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. *CoRR*, abs/1511.06530, 2015. URL https://api.semanticscholar.org/CorpusID:15602035.

Yuhang Li, Donghyun Lee, Ruokai Yin, and Priyadarshini Panda. Optimal brain decomposition for accurate llm low-rank approximation, 2026. URL https://arxiv.org/abs/2604.00821.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Jun Lu, Tianyi Xu, Bill Ding, David Li, and Yu Kang. Large language model compression via the nested activation-aware decomposition, 2025. URL https://arxiv.org/abs/2503.17101.

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.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. URL https://arxiv.org/abs/1609.07843.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 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*, pp. 2381–2391, 2018.

Tanay Narshana, Chaitanya Murti, and Chiranjib Bhattacharyya. DFPC: Data flow driven pruning of coupled channels without data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=mhnHqRqcjYU.

Wang Qinsi, Jinghan Ke, Masayoshi Tomizuka, Kurt Keutzer, and Chenfeng Xu. Dobi-SVD: Differentiable SVD for LLM compression and some new perspectives. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kws76i5XB8.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

Agastya Todi, Navya Narula, Moolch Sharma, and Umesh Gupta. Convnext: A contemporary architecture for convolutional neural networks for image classification. In *2023 3rd International Conference on Innovative Sustainable Computational Technologies (CISCT)*, pp. 1–6, 2023. doi: 10.1109/CISCT57197.2023.10351320.

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. Llama: Open and efficient foundation language models, 2023a. URL https://arxiv.org/abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, and Amjad Almahairi et.al. Llama 2: Open foundation and fine-tuned chat models, 2023b. URL https://arxiv.org/abs/2307.09288.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *European conference on computer vision*, pp. 459–479. Springer, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression, 2025. URL https://arxiv.org/abs/2403.07378.

SONG Bing-Bing; ZHANG Hao; WU Zi-Feng; LIU Jun-Hui; LIANG Yu; ZHOU Wei. Automated tensor decomposition to accelerate convolutional neural networks. *Journal of Software*, 32(11): 3468–3481, 2021.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2018. URL https://arxiv.org/abs/1704.05426.

An Yang, Anfeng Li, Baosong Yang, and Beichen Zhang et.al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. Asvd: Activation-aware singular value decomposition for compressing large language models, 2025. URL https://arxiv.org/abs/2312.05821.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*, 2018.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Boyang Zhang, Suping Wu, Leyang Yang, Bin Wang, and Wenlong Lu. A lightweight grouped low-rank tensor approximation network for 3d mesh reconstruction from videos. In *2023 IEEE International Conference on Multimedia and Expo*, pp. 930–935. IEEE, 2023.

Boyang Zhang, Daning Cheng, and Yunquan Zhang. Rethinking parameter sharing as graph coloring for structured compression, 2025. URL https://arxiv.org/abs/2511.06786.

Longguang Zhong, Fanqi Wan, Ruijun Chen, Xiaojun Quan, and Liangzhi Li. Blockpruner: Fine-grained pruning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5065–5080, 2025.

LLM USAGE

Large language models were used solely for grammar checking and language polishing. All research ideas, methods, experiments, analyses, and conclusions were developed and verified by the authors.

A Additional Statistics

This appendix reports additional statistics for our method only. Unless otherwise stated, all results are computed over $n = 5$ independent calibration seeds. The pretrained checkpoint, compression budget, and evaluation set are fixed; only the calibration subset used for the one-shot gradient computation is resampled. Calibration data are used only to estimate the local gradient signal and are excluded from final evaluation.

For each metric x, we report the sample mean

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i,$$

the sample standard deviation

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2},$$

and the two-sided 95% confidence interval

$$\bar{x} \pm t_{0.975,n-1} \frac{s}{\sqrt{n}}.$$

For $n = 5$, we use $t_{0.975,4} = 2.776$.

A.1 LLM Results with Calibration-Seed Error Bars

Table 8 reports calibration-seed variation for our method on LLaMA-7B under the two compression ratios used in the main paper. The small standard deviations show that the one-shot calibration gradient leads to stable compression results across different calibration subsets.

Table 8: LLaMA-7B results for our method with calibration-seed error bars. Perplexity and average zero-shot accuracy are reported as mean $\pm$ standard deviation over $n = 5$ calibration seeds.

Ratio	Wiki2 PPL ↓	95% CI	PTB PPL ↓	95% CI	Avg. Acc. ↑	95% CI
0.8	7.06 ± 0.03	$[7.02, 7.10]$	14.69 ± 0.12	$[14.54, 14.84]$	0.490 ± 0.001	$[0.489, 0.491]$
0.6	10.40 ± 0.14	$[10.23, 10.57]$	45.11 ± 0.81	$[44.10, 46.12]$	0.420 ± 0.002	$[0.418, 0.422]$

Table 9 reports calibration-seed variation for our method across different LLM families. All results are shown at the same parameter ratio used in the corresponding main-paper family comparison.

Table 9: Cross-family LLM results for our method with calibration-seed error bars. Perplexity is reported as mean $\pm$ standard deviation over $n = 5$ calibration seeds.

Model	Ratio	Wiki2 PPL ↓	95% CI	C4 PPL ↓	95% CI
LLaMA2-7B	0.8	7.02 ± 0.04	$[6.97, 7.07]$	17.84 ± 0.35	$[17.41, 18.27]$
LLaMA2-13B	0.8	6.11 ± 0.02	$[6.09, 6.13]$	13.56 ± 0.12	$[13.41, 13.71]$
LLaMA3-8B	0.8	12.43 ± 0.09	$[12.32, 12.54]$	67.39 ± 0.32	$[66.99, 67.79]$
Qwen3-8B	0.8	35.10 ± 0.10	$[34.98, 35.22]$	44.34 ± 0.10	$[44.22, 44.46]$

A.2 ImageNet Results with Error Bars

Table 10 reports calibration-seed variation for our two operating modes. CE denotes cross-entropy multiplied by 100 for readability, not a percentage. Ours(L) denotes the loss-preserving mode, while Ours(C) denotes the compact mode.

Table 10: ImageNet-1K results for our method with calibration-seed error bars. All values are mean $\pm$ standard deviation over $n = 5$ calibration seeds.

Model	Method	Param. Reduction	CE ↓	Top-1 ↑	Top-5 ↑
ViT-B	Ours(L)	3.0%	83.80 ± 0.05	81.007 ± 0.037	95.311 ± 0.017
ViT-B	Ours(C)	41.0%	83.85 ± 0.06	81.038 ± 0.038	95.290 ± 0.018
MaxViT	Ours(L)	30.0%	73.60 ± 0.01	83.341 ± 0.021	96.593 ± 0.012
MaxViT	Ours(C)	67.0%	73.91 ± 0.03	82.233 ± 0.052	94.490 ± 0.024
Swin-T	Ours(L)	2.0%	73.66 ± 0.03	82.790 ± 0.024	95.590 ± 0.013
Swin-T	Ours(C)	28.0%	74.02 ± 0.02	82.800 ± 0.033	94.844 ± 0.021
ResNeXt101	Ours(L)	75.0%	86.91 ± 0.01	78.160 ± 0.055	94.018 ± 0.033
ResNeXt101	Ours(C)	81.0%	92.45 ± 0.05	76.308 ± 0.036	91.482 ± 0.041
DenseNet169	Ours(L)	2.0%	97.18 ± 0.03	75.446 ± 0.041	92.790 ± 0.019
DenseNet169	Ours(C)	52.0%	99.48 ± 0.02	74.790 ± 0.023	92.572 ± 0.022
ConvNeXt-L	Ours(L)	2.0%	77.08 ± 0.07	84.115 ± 0.025	96.876 ± 0.010
ConvNeXt-L	Ours(C)	33.0%	77.09 ± 0.04	82.344 ± 0.051	96.866 ± 0.017

B High-Order Analysis for Our Method

The proposed rectification uses the signed first-order alignment

$$a(L, R) = \langle G, LR^\top - W \rangle$$

as a lightweight local criterion. This section clarifies why the first-order term is sufficient for our post-training rectification setting, both mathematically and empirically. The key observation is that the second-order residual is much smaller than the signed first-order term inside the layer-adaptive neighborhood. Moreover, explicitly adding a second-order correction gives negligible or negative downstream gains while substantially increasing construction cost.

B.1 Layer-Wise First- and Second-Order Decomposition

For a compressed layer W_l, let

$$\delta_l = \widehat{W}_l - W_l, \qquad \widehat{W}_l = L_l R_l^\top.$$

Assume that the loss has a β_l-Lipschitz gradient with respect to W_l inside the local neighborhood

$$\|\delta_l\|_F \le \epsilon_l.$$

Taylor's theorem gives

$$\mathcal{L}(W_{-l}, W_l + \delta_l) - \mathcal{L}(W_{-l}, W_l) = \underbrace{\langle G_l, \delta_l \rangle}_{A_l} + \underbrace{R_l(\delta_l)}_{B_l},$$

where

$$|B_l| = |R_l(\delta_l)| \le \frac{\beta_l}{2} \|\delta_l\|_F^2.$$

Here A_l is the signed first-order term and B_l is the higher-order residual. Since our method uses

$$\epsilon_l = \gamma \|W_l\|_F,$$

we have

$$|B_l| \le \frac{\beta_l}{2} \gamma^2 \|W_l\|_F^2.$$

In contrast, the first-order term scales linearly with the perturbation:

$$|A_l| = |\langle G_l, \delta_l \rangle| = \|G_l\|_F \|\delta_l\|_F |\cos \theta_l|,$$

where θ_l is the angle between G_l and δ_l. Therefore,

$$\frac{|B_l|}{|A_l|} \le \frac{\beta_l \|\delta_l\|_F}{2\|G_l\|_F |\cos \theta_l|} \le \frac{\beta_l \gamma \|W_l\|_F}{2\|G_l\|_F |\cos \theta_l|}.$$

Table 11: Empirical non-zero first-order signal. The column "Non-zero seeds" reports how often the measured calibration gradient norm is strictly non-zero across $n = 5$ calibration seeds. The first-order term is normalized to 1.0 to report the relative size of the measured second-order residual.

Model	Non-zero seeds	Normalized $\lvert A\rvert$	Relative $\lvert B\rvert/\lvert A\rvert$	Interpretation
ResNet-50	$5/5$	1.00	< 0.09	first-order dominates
BERT-base	$5/5$	1.00	< 0.15	first-order dominates

This ratio is small when the calibration gradient is non-zero, the rectified perturbation has a non-negligible signed alignment with the gradient, and the neighborhood radius $\gamma\|W_l\|_F$ is small. In this regime, the first-order term dominates the local loss variation.

More formally, if

$$\rho_l = \frac{|B_l|}{|A_l| + \varepsilon} < 1,$$

then

$$|A_l + B_l - A_l| = |B_l| \leq \rho_l |A_l|,$$

up to the numerical stabilizer ε. Thus, the second-order term can only perturb the first-order prediction by a relative factor of ρ_l. When $\rho_l \ll 1$, minimizing the signed first-order term is already sufficient to determine the local direction of the loss change, and explicitly optimizing a second-order correction is unlikely to change the compression decision substantially.

B.2 Multi-Layer Expansion

Let $\Delta = \{\delta_l\}_{l \in \mathcal{C}}$ denote the collection of perturbations over compressed layers. If the full calibration loss has a β-Lipschitz gradient over the concatenated compressed parameters, then

$$\mathcal{L}(W + \Delta) - \mathcal{L}(W) = \underbrace{\sum_{l \in \mathcal{C}} \langle G_l, \delta_l \rangle}_{A} + \underbrace{R(\Delta)}_{B},$$

where

$$|B| \leq \frac{\beta}{2} \sum_{l \in \mathcal{C}} \|\delta_l\|_F^2.$$

The global residual B includes both layer-wise curvature and cross-layer interaction terms. Define

$$\rho = \frac{|B|}{|A| + \varepsilon}.$$

When $\rho \ll 1$, the first-order term A dominates the local loss variation. In this case, a second-order objective can only make a small correction to the already-dominant first-order signal, while requiring additional curvature estimation or Hessian-vector computation.

This is the regime observed in our experiments: the empirical calibration gradients are non-zero across calibration seeds, and the measured second-order residual is much smaller than the first-order signed term.

B.3 Empirical Non-Zero First-Order Signal

A common concern is that a pretrained model may be close to a stationary point, in which case the full-batch training gradient would be small. Our method does not rely on the original training-set gradient. Instead, it uses the empirical calibration gradient under the target compression/evaluation distribution. Table 11 shows that the empirical first-order signal is non-zero for all calibration seeds used in our evaluation.

The empirical result supports the use of the signed first-order term. The calibration gradient is not zero in practice, and the measured higher-order residual remains substantially smaller than the first-order magnitude on the tested models. Therefore, the first-order approximation provides a sufficiently accurate and substantially cheaper criterion for post-training rectification.

B.4 SECOND-ORDER AUGMENTED VARIANT

For completeness, we also evaluate a second-order augmented variant. This variant uses the objective

$$a_2(L, R) = \langle G, \delta \rangle + \frac{\lambda}{2} \delta^\top \widehat{H} \delta, \qquad \delta = LR^\top - W,$$

where $\widehat{H}$ denotes an empirical curvature estimate computed from the calibration set, and λ controls the strength of the second-order correction. This variant is more expensive because it requires additional curvature estimation and second-order matrix-vector operations. It is included only as a diagnostic to test whether explicit second-order information improves the final compressed model.

Table 12 shows that adding the second-order term does not improve practical task performance. On ResNet-50, the second-order augmented variant is 0.7% lower than the first-order method while requiring an additional 3 hours of construction time. On BERT-base, it is 1.4% lower while requiring an additional 5 hours. These results indicate that the second-order term is not only unnecessary under our local neighborhood, but also unfavorable in the post-training setting due to its computation cost.

Table 12: Comparison between our first-order rectification and a second-order augmented variant. "Perf. change" is measured relative to our first-order method; negative values indicate worse performance after adding the second-order term.

Model	Main metric	First-order Ours	Second-order variant	Extra time
ResNet-50	ImageNet Top-1	76.62	75.92 (-0.70)	$+3$ h
BERT-base	MNLI Test Acc.	83.92	82.52 (-1.40)	$+5$ h

C IMPLEMENTATION DETAILS FOR STABLE RECTIFICATION

We add several implementation safeguards to ensure that the rectification procedure is well-defined. For each compressible layer, we first compute the rank-k SVD initialization and check whether the initial perturbation

$$\delta = LR^\top - W$$

satisfies the local constraint

$$\|\delta\|_F \le \epsilon_l.$$

If the initialization violates this constraint, the selected rank is not feasible under the prescribed local neighborhood. In this case, we increase the rank until the SVD initialization becomes feasible. If no compression-effective rank satisfies the constraint, the layer is left uncompressed. This avoids the case where backtracking would be applied inside an empty feasible region.

During rectification, a tentative update is accepted only when it satisfies both the neighborhood constraint and the first-order descent condition. Specifically, after computing tentative factors L' and R', we set

$$\delta' = L'(R')^\top - W, \qquad a' = \langle G, \delta' \rangle.$$

The update is accepted only if

$$\|\delta'\|_F \le \epsilon_l \quad \text{and} \quad a' \le a,$$

where $a = \langle G, \delta \rangle$ is the current alignment value. If either condition is violated, we halve η and recompute the tentative update. In the next outer iteration, η is restored to its prescribed value rather than being permanently decayed.

The same rule is used in compact mode. The only difference is that compact mode first selects the smallest rank whose SVD initialization satisfies the layer-adaptive constraint. Therefore, both modes perform rectification only from feasible low-rank initializations and only accept updates that reduce the signed first-order alignment while staying inside the local neighborhood.

D EXPERIMENTAL SETTINGS

We follow the general post-training evaluation protocol used in SVD-LLM for LLM compression. All LLM experiments are conducted with HuggingFace checkpoints and implemented in PyTorch. Following the SVD-LLM setting, the calibration data are used only for collecting local statistics and computing the one-shot compression signal, and are not used for final evaluation. For each model, we sample a fixed number of calibration sequences from WikiText and keep the sampling seed fixed unless we explicitly evaluate calibration-seed variation. The calibration objective is the standard next-token cross-entropy with shifted-token labels. We use the same model sequence length for calibration and compression statistics as in the corresponding LLM evaluation setting, and we keep the tokenizer, preprocessing pipeline, and maximum sequence length fixed across all compared compression ratios.

For LLaMA-family and Qwen-family models, we compress the major linear projections in the attention and MLP blocks whenever the resulting low-rank factorization is parameter-efficient. Specifically, a matrix $W \in \mathbb{R}^{m\times n}$ is considered compressible only when the selected rank k satisfies

$$k(m+n) < mn.$$

Layers for which no compression-effective rank satisfies the layer-adaptive neighborhood constraint are left uncompressed. This rule follows the same practical principle as SVD-based LLM compression: the compressed model should reduce parameters at the module level rather than merely replacing a dense matrix by a larger two-factor representation.

For LLM perplexity evaluation, we report results on WikiText-2, PTB, and C4 following the standard language-modeling protocol used by SVD-style LLM compression methods. The compressed model is evaluated directly after post-training factorization and rectification, without multi-epoch finetuning. For zero-shot evaluation, we use the same compressed checkpoint and report downstream accuracy on the reasoning and commonsense benchmarks used in the main paper. The evaluation data are disjoint from the calibration data.

For vision models, calibration images are randomly sampled from the ImageNet-1K training set and used only to compute the supervised cross-entropy gradient. The final Top-1 and Top-5 accuracy are evaluated on the ImageNet-1K validation set. For BERT experiments, calibration examples are sampled from the training split of the corresponding task, and final validation and test accuracy are reported on MNLI. In all modalities, the calibration set is used only once for post-training compression; no multi-epoch post-factorization finetuning is performed.

For calibration-seed experiments, we repeat the compression procedure over five independently sampled calibration subsets while keeping the pretrained checkpoint, compression ratio, and evaluation set fixed. The reported standard deviations therefore measure the stability of the one-shot calibration procedure rather than variation from model retraining.

For efficiency evaluation, we follow the SVD-LLM-style deployment protocol and report both model construction cost and inference efficiency. The low-rank factors are deployed as two consecutive linear operations rather than being reconstructed into dense matrices. Model-loading time is excluded from latency measurement. We run several warm-up iterations before timing and report the average latency and throughput under the same batch size, precision, and sequence-length configuration for the dense and compressed models.

Table 13: Sensitivity to the rectification step size η on ResNet-50. Top-1 and Top-5 accuracy are reported as mean $\pm$ standard deviation over $n = 5$ calibration seeds. The default setting $\eta = 10^{-4}$ is used in the main experiments.

η	Top-1 Acc. ↑	Top-5 Acc. ↑	Avg. Iters
1×10^{-5}	76.32 ± 0.05	93.90 ± 0.03	47.8
5×10^{-5}	76.54 ± 0.04	93.99 ± 0.03	22.6
1×10^{-4}	76.61 ± 0.02	94.01 ± 0.03	5.5
5×10^{-4}	76.47 ± 0.06	93.97 ± 0.04	4.9
1×10^{-3}	76.19 ± 0.08	93.83 ± 0.06	4.2

D.1 Definition of Ablation Variants

We clarify the ablation variants used in the directional rectification study. All variants are evaluated under the same model, compression budget, selected ranks, calibration data, local neighborhood constraint, and rectification iteration budget as our method. The only difference is how the post-SVD factorized weights are updated after initialization. The variant *w/o Dir.* uses the same low-rank initialization and selected ranks but does not use the signed first-order direction for rectification; it therefore measures the performance of the reconstruction-oriented factorization under the same ablation setting. The variant *Random Dir.* keeps the same update format and neighborhood constraint, but replaces the task gradient direction with a random direction of matching shape, testing whether arbitrary perturbation steering can explain the gain. The variant *Factor-FT* starts from the same low-rank factors and directly optimizes the calibration loss with respect to the factors under the same post-training budget; it is included to test whether the improvement comes merely from updating the factors, rather than from the proposed signed alignment criterion. Finally, *w/ Dir.* is the proposed method, which rectifies the factors by minimizing the signed first-order alignment $\langle G, LR^\top - W\rangle$ while preserving the rank and the layer-adaptive neighborhood. As reported in the main ablation table, *w/ Dir.* achieves the best accuracy on both ViT-B and ResNeXt101, outperforming direct factor fine-tuning and substantially improving over both no-direction and random-direction variants. This indicates that the gain is not simply due to extra factor updates, but to using the direction in a controlled local rectification step.

Table 14: Sensitivity to the maximum number of rectification iterations $T_{\max}$ on ResNet-50. Top-1 and Top-5 accuracy are reported as mean $\pm$ standard deviation over $n = 5$ calibration seeds. The default setting $T_{\max} = 50$ is used in the main experiments.

$T_{\max}$	Top-1 Acc. ↑	Top-5 Acc. ↑	Avg. Used Iters	Construction Time
10	75.38 ± 0.07	92.63 ± 0.07	3.8	$0.7\times$
25	76.29 ± 0.04	93.59 ± 0.03	5.1	$0.9\times$
50	76.62 ± 0.02	94.01 ± 0.03	5.5	$1.0\times$
100	76.63 ± 0.01	94.02 ± 0.01	5.6	$1.1\times$
200	76.60 ± 0.02	94.00 ± 0.02	5.6	$1.2\times$

E Sensitivity to Step Size and Rectification Iterations

We further evaluate the sensitivity of our method to the rectification step size η and the maximum number of rectification iterations $T_{\max}$. The experiments are conducted on ResNet-50 under the same compression setting as the main ImageNet experiment. We fix the neighborhood ratio to $\gamma = 0.05$ and vary one hyperparameter at a time.

As shown in Table 13, the method is stable around the default step size $\eta = 10^{-4}$. A very small step size leads to slower rectification and lower accuracy, since the alignment term is not sufficiently reduced within the iteration budget. In contrast, overly large step sizes reduce the number of accepted iterations but slightly degrade accuracy, indicating less stable local rectification. The default value $\eta = 10^{-4}$ achieves the best Top-1 accuracy among the tested settings while requiring only a small number of rectification iterations.

Table 14 shows that increasing $T_{\max}$ improves performance up to $T_{\max} = 50$, after which the gains become negligible. Although $T_{\max} = 100$ gives a marginally higher Top-1 accuracy, the improvement over $T_{\max} = 50$ is only 0.01%, while the construction time increases. Further increasing the budget to $T_{\max} = 200$ does not improve performance. This is because most layers terminate early once the alignment becomes non-positive or the improvement falls below the stopping tolerance. These results support the use of $T_{\max} = 50$ as a practical default and confirm that our method does not rely on long post-factorization finetuning.

F Algorithmic Cost Analysis

We provide an additional cost analysis of the proposed rectification procedure on two representative models, ResNet-50 and BERT-base. For a compressible weight matrix $W_l \in \mathbb{R}^{m_l \times n_l}$ with selected rank k_l, the offline construction cost consists of three parts: one calibration backward pass, low-rank initialization, and directional rectification. The calibration backward pass is performed once to obtain the empirical gradient G_l. The initialization cost is the standard truncated-SVD cost used by SVD-based post-training compression. The extra cost introduced by our method comes only from the factor-space rectification steps.

For each rectification step, the update

$$L_l' = L_l - \eta G_l R_l, \qquad R_l' = R_l - \eta G_l^\top L_l$$

is dominated by the two matrix products $G_l R_l$ and $G_l^\top L_l$. Therefore, the leading-order cost of one rectification step is

$$O(m_l n_l k_l).$$

If T_l rectification steps are used for layer l, the additional layer-wise rectification cost is

$$O(T_l m_l n_l k_l).$$

Across all compressed layers $\mathcal{C}$, the total offline construction cost is

$$\sum_{l \in \mathcal{C}} \mathrm{Cost}_{\mathrm{SVD}}(W_l) + O\left(\sum_{l \in \mathcal{C}} T_l m_l n_l k_l\right)$$

plus one calibration backward pass. This cost is incurred only once when constructing the compressed model and does not introduce any additional training or inference-time gradient computation.

The alignment value can be computed without permanently materializing a dense reconstructed weight:

$$\langle G_l, L_l R_l^\top - W_l \rangle = \mathrm{tr}(L_l^\top G_l R_l) - \langle G_l, W_l \rangle,$$

where $\langle G_l, W_l \rangle$ is precomputed once per layer. The neighborhood norm can also be checked using

$$\|L_l R_l^\top - W_l\|_F^2 = \|W_l\|_F^2 + \|L_l R_l^\top\|_F^2 - 2\,\mathrm{tr}(L_l^\top W_l R_l).$$

Thus, when $k_l \ll \min(m_l, n_l)$, the leading-order per-step rectification cost remains $O(m_l n_l k_l)$.

Table 15: Construction-time breakdown of our method on ResNet-50 and BERT-base. "Calib." denotes the one-shot gradient computation, and "Rect." denotes directional rectification. All costs are offline post-training costs.

Model	Setting	Calib.	SVD Init.	Rect.	Total Time
ResNet-50	Ours(C)	2.1 min	1.4 min	0.9 min	4.4 min
BERT-base	Ours	3.3 min	3.1 min	2.5 min	8.9 min

Table 15 shows that the directional rectification overhead is modest relative to the total construction cost. On ResNet-50, rectification takes 0.9 minutes out of a total 4.4 minutes. On BERT-base, rectification takes 2.5 minutes out of a total 8.9 minutes. These costs are one-time offline construction costs and are substantially different from multi-epoch post-factorization finetuning.

Table 16: Rectification behavior under the default setting $\gamma = 0.05$, $\eta = 10^{-4}$, and $T_{\max} = 50$. The average number of used iterations is much smaller than $T_{\max}$, indicating that most layers stop early.

Model	Setting	Avg. Used Iters	Max Iters	Early-stop Rate
ResNet-50	Ours(C)	5.5	50	93.3%
BERT-base	Ours	6.2	50	95.1%

As shown in Table 16, rectification usually terminates far before reaching the maximum iteration budget. Although $T_{\max} = 50$ is allowed, the average number of used iterations is only 5.5 on ResNet-50 and 6.2 on BERT-base. This early stopping behavior explains why the practical rectification overhead remains small. It also confirms that the proposed method relies on a short post-training correction rather than long post-factorization optimization.