AdaSVD: Singular Value Decomposition with Adaptive Mechanisms for Large Multimodal Models

Zhiteng Li*, Mingyuan Xia*, Jingyuan Zhang*, Zheng Hui, Haotong Qin, Linghe Kong, Yulun Zhang, Xiaokang Yang

*Equal contribution | Shanghai Jiao Tong University · Baidu Inc. · ETH Zürich

CVPR 2026

LMM Deployment is Bottlenecked by Memory Cost

Large Multimodal Models (LMMs) such as LLaVA achieve impressive performance across vision–language tasks, yet their massive parameter counts — often exceeding 7 billion — impose prohibitive memory requirements that prevent deployment on resource–limited devices such as smartphones and IoT hardware.

WHY SVD COMPRESSION?

- Reduces model size and memory cost **without requiring custom hardware kernels**
- Is **orthogonal to quantization and pruning**, enabling combination for further gains
- Preserves the original weight structure, making it easy to integrate into existing pipelines

TARGET

Compress LMMs with SVD at high compression ratios **(40%—80%)** while preserving model utility across both language and vision tasks.

Two Core Challenges Limit Existing SVD Methods

Despite its appeal, existing SVD compression methods face two fundamental challenges that become especially severe at high compression ratios:

Challenge 1: Uncompensated Truncation Error

SVD truncation discards the smallest singular values, leaving a residual low-rank approximation error that is never corrected. This error accumulates across layers and degrades model performance significantly at compression ratios above 50%.

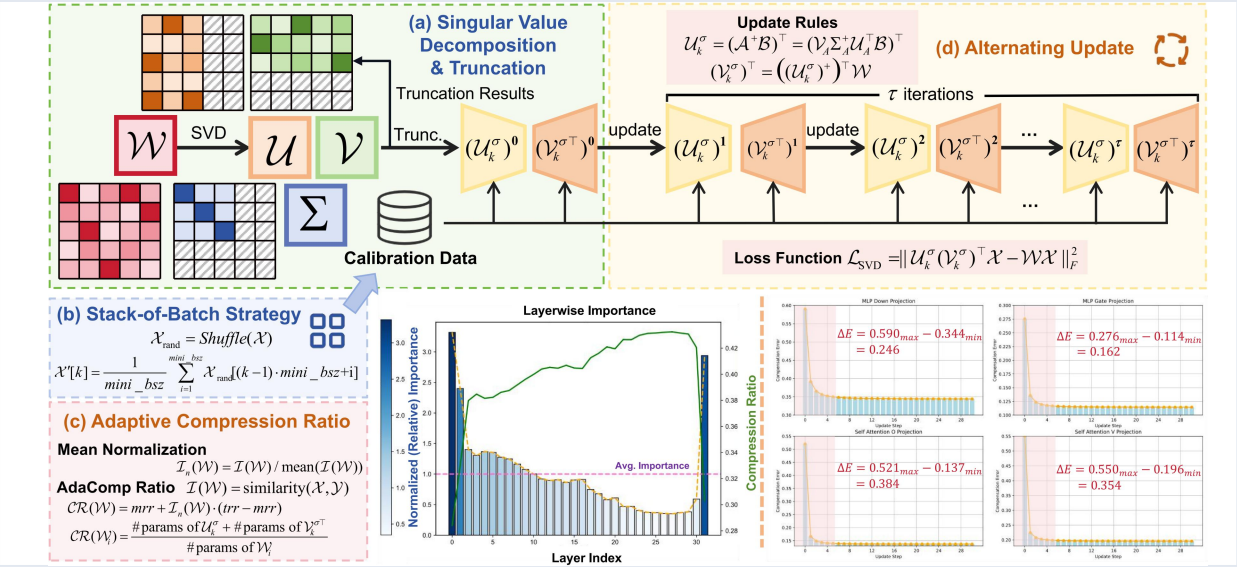
Challenge 2: Uniform Compression Ignores Layer Importance

Applying the same compression ratio to every transformer layer is suboptimal. Different layers contribute differently to the model's output — forcing uniform compression wastes capacity on less critical layers while over-compressing important ones.

INSIGHT

AdaSVD addresses both challenges with two adaptive mechanisms: **adaComp** (adaptive compensation) and **adaCR** (adaptive compression ratio).

AdaSVD Method Overview



(a) SVD & Truncation
Decompose weight W into U, Σ, V; truncate to rank k to obtain initial low–rank factors.

(b) Stack–of–Batch Strategy
Shuffle calibration data into mini–batches to improve gradient stability during compensation.

(c) adaCR — Adaptive Compression Ratio
Assign layer–specific ranks based on input–output cosine similarity importance scores.

(d) adaComp — Alternating Update
Iteratively refine U and V factors to minimize reconstruction error using calibration data.

adaComp: Iterative Alternating Update Eliminates Truncation Error

After SVD truncation produces initial factors $(U_k^{\sigma})^0$ and $(V_k^{\sigma T})^0$, **adaComp** iteratively refines them by minimizing the reconstruction loss:

Mathematical Formulation

LOSS FUNCTION:

$$L_{SVD} = \| U_k^{\sigma}(V_k^{\sigma})^T X - WX \|_F^2$$

ALTERNATING UPDATE RULES (CLOSED-FORM):

$$U_k^{\sigma} = (A^+B)^T = (V_A \Sigma_A^+ U_A^T B)^T$$

$$(V_k^{\sigma})^T = ((U_k^{\sigma})^+)^T W$$

Iterative Process

The process runs for τ **iterations**, alternately fixing one factor and solving for the other. This efficiently reduces the compensation error ΔE across all projection types (MLP Down/Gate, Self-Attention O/V), with observed error reductions of 0.162—0.384 in practice.

KEY INSIGHT

Even a single iteration (τ=1) already outperforms the previous state-of-the-art SVD-LLM baseline.

adaCR: Layer Importance Drives Adaptive Rank Allocation

Not all transformer layers are equally important. **adaCR** measures each layer's importance by the cosine similarity between its input X and output Y, then redistributes the compression budget accordingly.

ALGORITHM

1. Compute importance:

```
I(W) = similarity(X, Y)
```

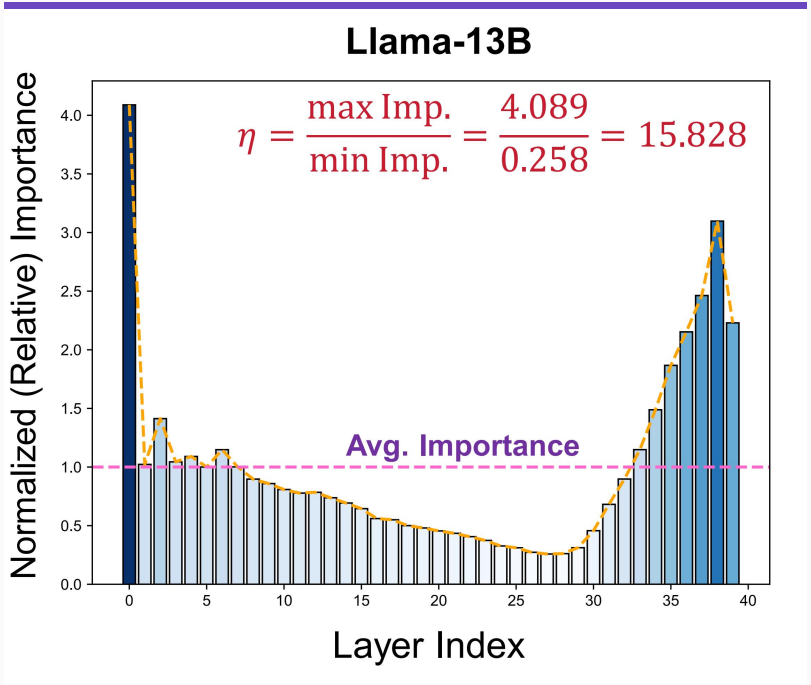
2. Mean–normalize:

```
Iₙ(W) = I(W) / mean(I(W))
```

3. Assign compression ratio:

```
CR(W) = mrr + Iₙ(W) × (trr − mrr)
```

Where **mrr** is the minimum retention ratio and **trr** is the target retention ratio. Layers with $I_n(W) > 1$ retain more ranks; layers below average are compressed more aggressively.



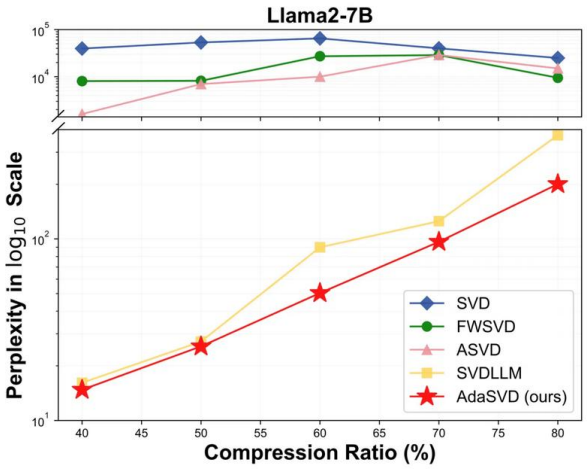
Bowl–shaped importance curve across 40 transformer layers

AdaSVD Consistently Outperforms All SVD Baselines

Zero-shot performance comparison on **LLaMA2-7B** across 40%, 50%, and 60% compression ratios, evaluated on language modeling and commonsense reasoning datasets:

RATIO	METHOD	WIKITEXT-2 ↓	PTB ↓	C4 ↓	AVERAGE ↑
40%	SVD-LLM	16.11	719.44	61.95	40.69
	AdaSVD	**14.76** (↓18%)	**304.62** (↓58%)	**56.98** (↓18%)	**42.63**
50%	SVD-LLM	27.19	1,772.91	129.66	37.83
	AdaSVD	**25.58** (↓16%)	**593.14** (↓67%)	**113.84** (↓12%)	**39.17**
60%	SVD-LLM	89.90	2,052.89	561.00	35.48
	AdaSVD	**50.33** (↓44%)	**1,216.95** (↓41%)	**239.18** (↓57%)	**36.87**

AdaSVD's advantage grows substantially at higher compression ratios, demonstrating its particular effectiveness in resource-constrained deployment scenarios.



Perplexity vs. Compression Ratio (LLaMA2-7B)

AdaSVD Generalizes Across Model Families and Scales

To validate generalizability, AdaSVD is evaluated on four LLM families at a **60% compression ratio** on WikiText-2 perplexity (lower is better):

METHOD	OPT-6.7B	LLAMA2-7B	MISTRAL-7B	VICUNA-7B
SVD	18,607	65,187	30,378	78,705
FWSVD	8,570	27,213	5,481	8,186
ASVD	10,326	10,004	22,706	20,241
SVD-LLM	92.10	89.90	72.17	64.06
AdaSVD	**86.64** (↓6%)	**50.33** (↓44%)	**67.22** (↓7%)	**56.97** (↓11%)

AdaSVD consistently achieves the lowest perplexity across all four model families, with especially pronounced improvements on **LLaMA2-7B** and **Vicuna-7B** where SVD-LLM still struggles.

Ablation: Both adaComp and adaCR Contribute Independently

Effectiveness of adaComp

Adding adaComp to the base SVD-LLM framework consistently reduces perplexity across all compression ratios. The performance gap widens at higher compression (**60%+**), confirming adaComp's critical role in high-compression scenarios.

Effectiveness of adaCR

Replacing uniform compression ratios with adaptive ones (adaCR) further reduces perplexity at all tested ratios. The two mechanisms are **complementary** — combining both achieves the best results.

Method	Tgt. CR	adaComp	WikiText2 ↓	PTB ↓	C4 ↓
SVD-LLM	40%	✗	16.11	719.44	61.95
AdaSVD	40%	✗	15.47	406.83	66.29
AdaSVD	40%	✓	14.76	304.62	56.98
SVD-LLM	50%	✗	27.19	1,772.91	129.66
AdaSVD	50%	✗	30.00	1101.15	166.02
AdaSVD	50%	✓	25.58	593.14	113.84
SVD-LLM	60%	✗	89.90	2,052.89	561.00
AdaSVD	60%	✗	78.82	6,929.39	339.31
AdaSVD	60%	✓	50.33	1,216.95	239.18

(a) Effectiveness of Adaptive Compensation

Method	Tgt. CR	CR	WikiText2 ↓	PTB ↓	C4 ↓
SVD-LLM	40%	Const	16.11	719.44	61.95
AdaSVD	40%	Const	15.38	617.11	60.43
AdaSVD	40%	Adapt	14.76	304.62	56.98
SVD-LLM	50%	Const	27.19	1,772.91	129.66
AdaSVD	50%	Const	27.33	1,177.53	126.85
AdaSVD	50%	Adapt	25.58	593.14	113.84
SVD-LLM	60%	Const	89.90	2,052.89	561.00
AdaSVD	60%	Const	69.46	2,670.20	336.90
AdaSVD	60%	Adapt	50.33	1,216.95	239.18

(b) Effectiveness of Adaptive Compression Ratio

Ablation studies on LLaMA2-7B

AdaSVD Preserves Visual Semantics in Multimodal Models

AdaSVD is applied to the **LLaVA-7B** visual language model, compressing only the language backbone (which dominates model size) at a **40% compression ratio**, evaluated on COCO image captioning:

- **SVD:** Produces hallucinated or incoherent captions (e.g., "a man standing on one side of the road..." with wrong object identification).
- **SVD-LLM:** Generates partially correct but often confused descriptions.
- **AdaSVD:** Produces accurate, semantically coherent captions that correctly identify objects, actions, and scene context.

This demonstrates that AdaSVD's adaptive mechanisms preserve not only language modeling capability but also the cross-modal alignment critical for vision-language tasks.

Figure 5: Image captioning comparison on COCO dataset

AdaSVD + GPTQ-INT4 Achieves Further Compression Gains

Since AdaSVD is orthogonal to quantization, it can be combined with **GPTQ-INT4** weight quantization for additional compression. Results on LLaMA2-7B (WikiText-2 perplexity):

RATIO	SVD-LLM	SVD-LLM + GPTQ	ADASVD	ADASVD + GPTQ
40%	16.11	33.56	**14.76**	**22.55**
50%	27.19	41.70	**25.58**	**37.34**
60%	89.90	119.46	**60.08**	**82.08**
70%	125.16	159.53	**107.90**	**118.75**
80%	372.48	420.25	**206.51**	**214.51**

AdaSVD + GPTQ-INT4 consistently outperforms SVD-LLM + GPTQ across all ratios, and at high compression (60%—80%), **AdaSVD alone even surpasses SVD-LLM + GPTQ**, demonstrating superior compression efficiency.

AdaSVD: Adaptive SVD Compression for Practical LMM Deployment

SUMMARY OF CONTRIBUTIONS

- **adaComp (Adaptive Compensation):** Iteratively refines SVD factors via alternating least-squares updates, efficiently eliminating truncation-induced reconstruction errors without additional training.
- **adaCR (Adaptive Compression Ratio):** Assigns layer-specific compression ratios based on input-output cosine similarity importance, ensuring critical layers retain sufficient capacity under any target compression budget.

IMPACT

AdaSVD consistently outperforms all prior SVD-based methods across multiple LLM families, compression ratios (40%—80%), and modalities (language + vision), with especially large gains at high compression ratios where resource-constrained deployment matters most.

FUTURE WORK

Extending AdaSVD to larger-scale models (13B, 70B), video LMMs, and structured sparsity combinations.

RESOURCES: Paper (arXiv) | Code (GitHub)