The 40th Annual AAAI Conference on Artificial Intelligence

Paper ID: 26363

# LatentLLM: Activation-Aware Transform to Multi-Head Latent Attention

Toshiaki Koike-Akino, Xiangyu Chen, Jing Liu, Ye Wang,
Pu (Perry) Wang, Matthew Brand

January 2026

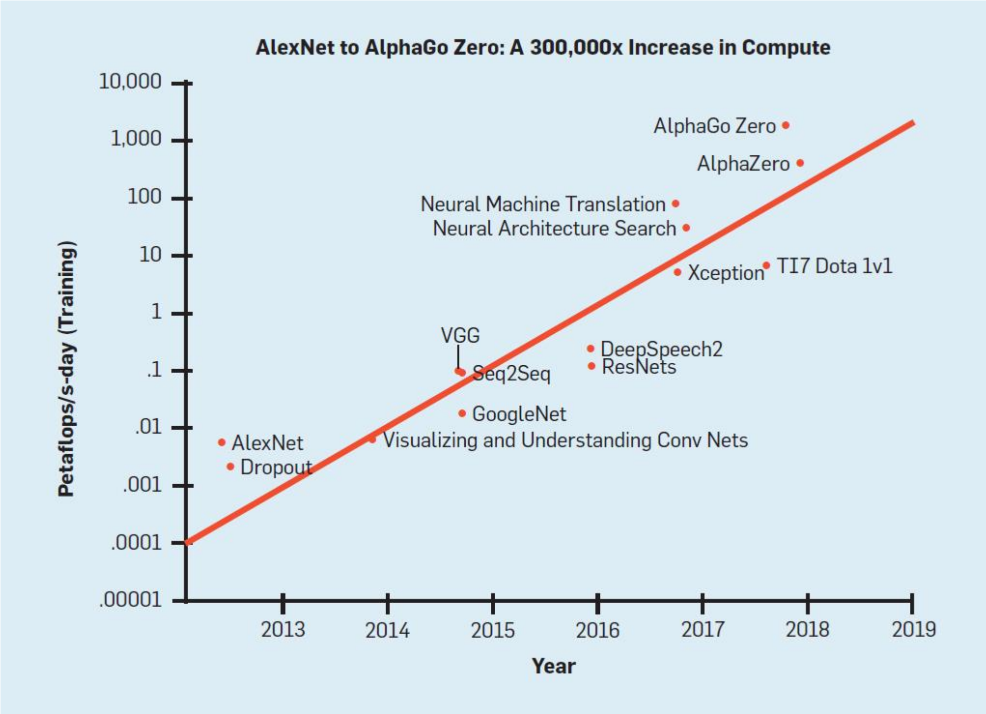
MITSUBISHI ELECTRIC RESEARCH LABORATORIES (MERL)
Cambridge, Massachusetts, USA
http://www.merl.com

# Agenda

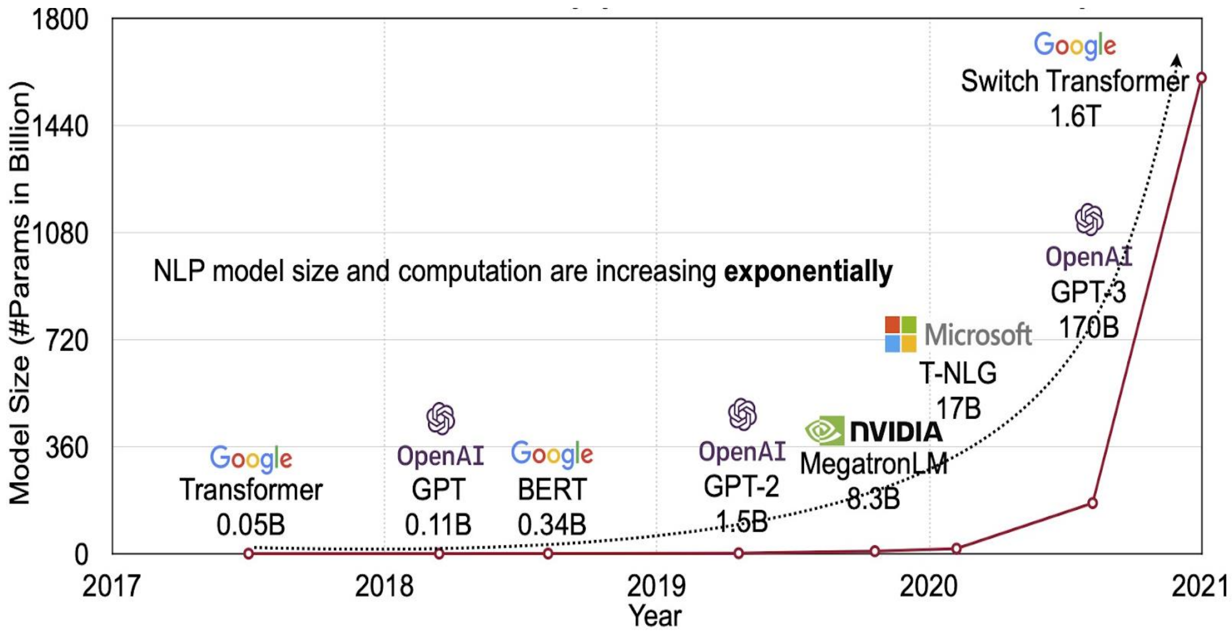
- Green AI overview
- Model compression: **LatentLLM**
  - Preconditioning matrix
  - Junction matrix
  - Attention-aware joint tensor decomposition
- Benchmark experiments
- Summary

# Social Challenge: Red AI


- R. Schwartz et al., “Green AI” 2020.
  - Training compute has increased exponentially: **10-fold annually**.
- Strubell et al. “Energy and policy considerations for deep learning in NLP” 2019.
  - Training a single NLP model requires **5-fold higher** carbon emission of *single car lifetime*.




The computation used to train deep learning models has increased 300,000x in six years: nearly 10x annually




Language model increases exponentially over years

# Power Demand by AI Explosion

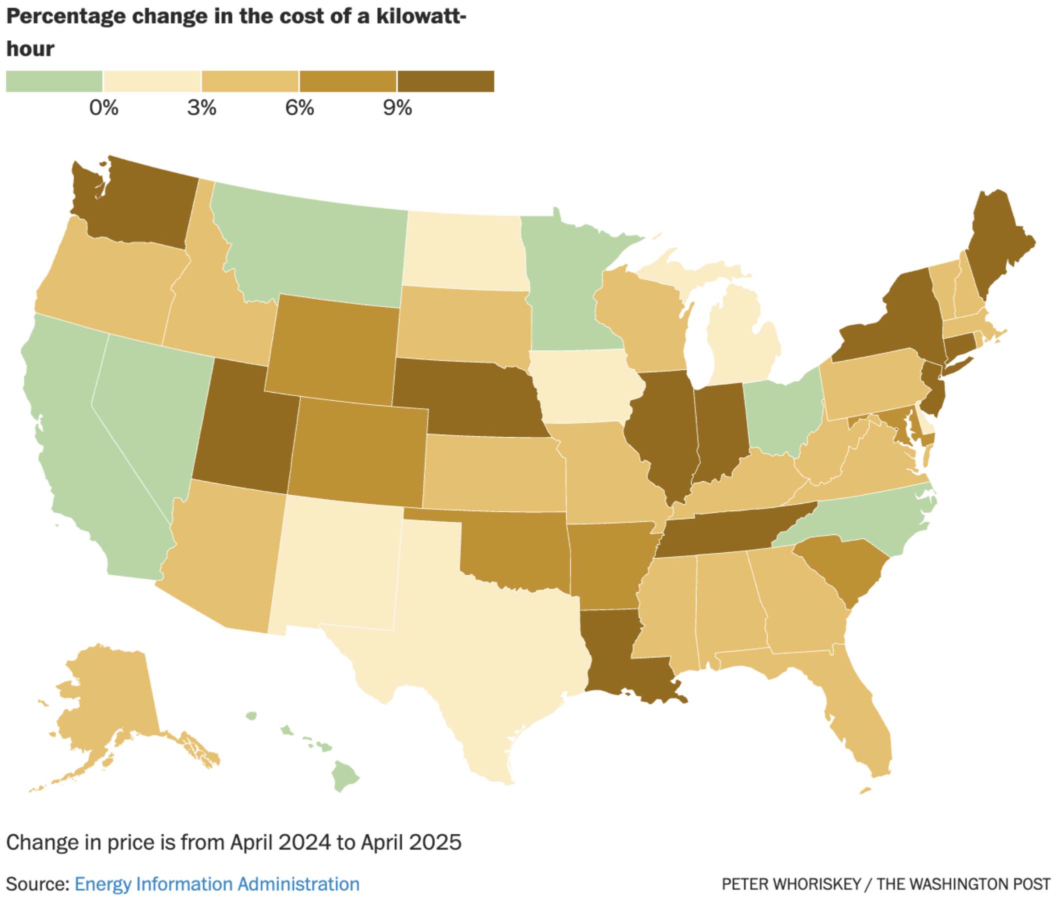
- High demand in electric power at data center driven by AI explosion, increasing bill
  - Electricity capacity prices jumped 833% in one regional auction
  - Some households saw $27/month increases
  - All traced directly to data center electricity demand

The Washington Post
*Democracy Dies in Darkness*

# The AI explosion means millions are paying more for electricity

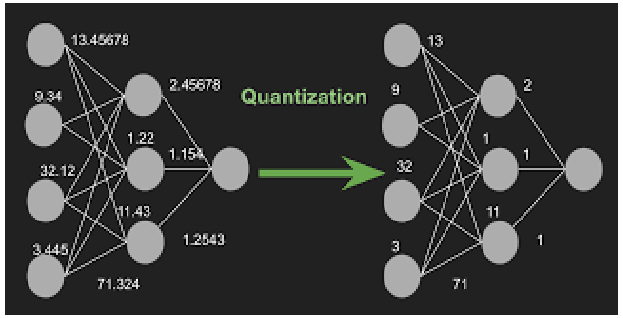
The data centers required for Big Tech are driving up electricity demand — and prices.

July 27, 2025

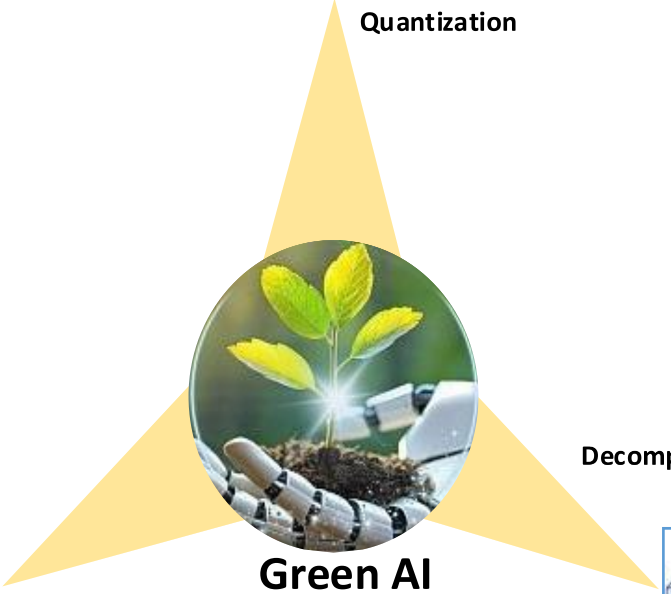


# Green AI Technologies

- ***Data*** efficiency:
  - Data distillation
  - Dimension reduction
  - Curriculum learning
- ***Training*** efficiency:
  - Few-shot learning
  - Parameter-efficient fine-tuning (PEFT)
- ***Model*** efficiency:
  - Quantization
  - Pruning
  - Decomposition

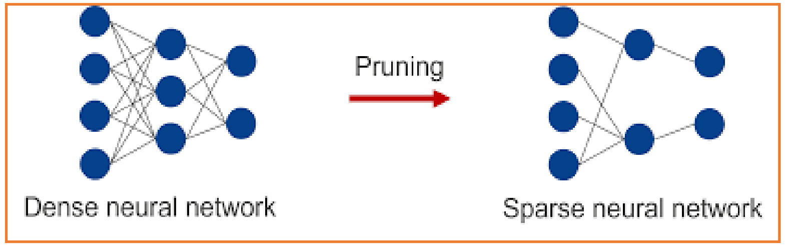



Quantization

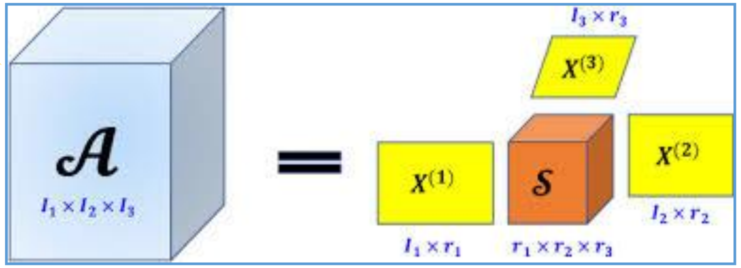


Green AI
Efficient Model

Pruning



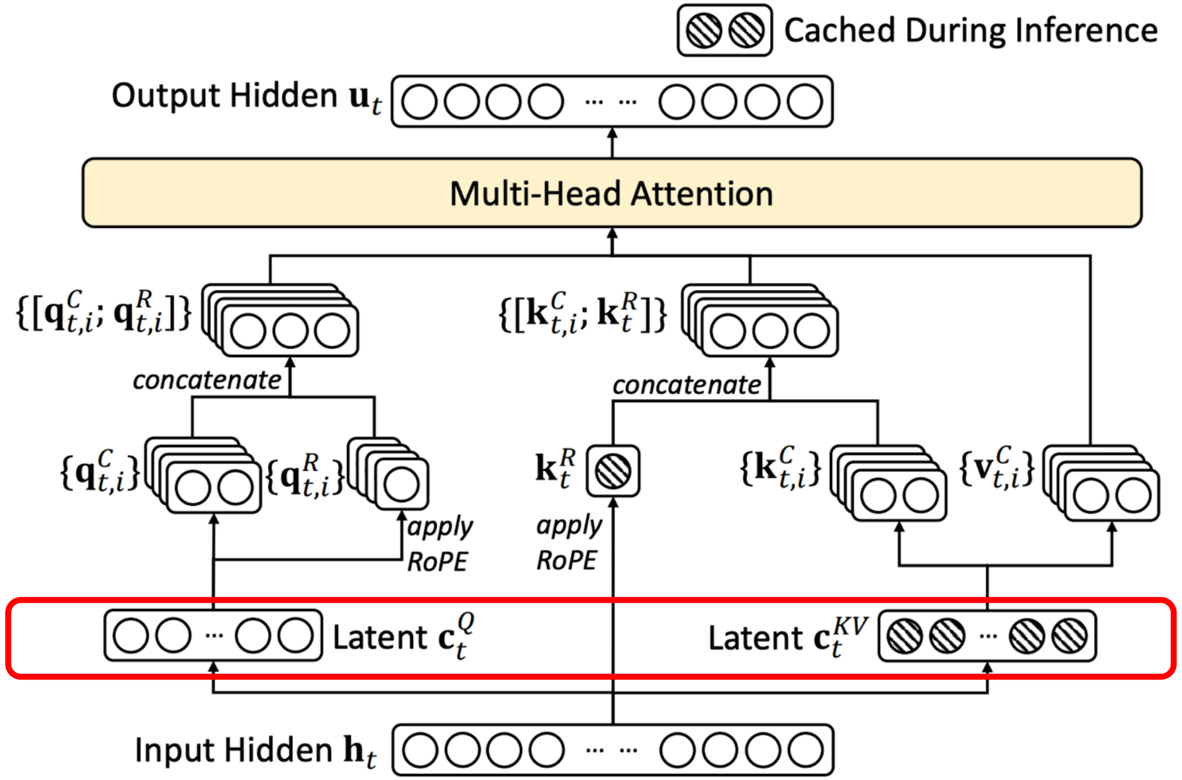

Decomposition



# LatentLLM

- We developed a solution to globally compress multi-head attention (MHA) into multi-head latent attention (MLA)
- MLA was introduced in **DeepSeek-V3** to improve efficiency: KV cache
- How to convert any LLMs into DeepSeek-like LLMs efficiently?
  - 1) Preconditioner; 2) Junction; 3) Joint tensor decomposition

**MLA in DeepSeek-V3**




QKV Low-Dimensional Projection

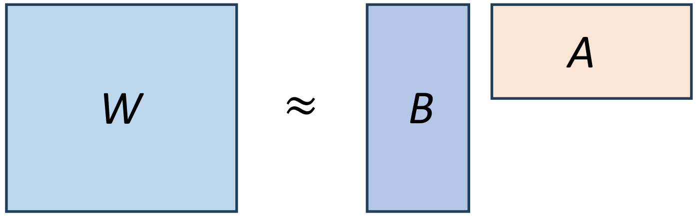
Q: 7168 → 1536
KV: 7168 → 512

# 1. Activation-Aware Rank Reduction

- How to compress pre-trained LLMs without fine-tuning?
- Adaptive SVD (ASVD) [Yuan24] compresses weights in a locally optimal manner
  - Activation statistics is compensated by preconditioning matrix
  - Calibration tokens are used to compute activation statistics (similar to AWQ/GPTQ)

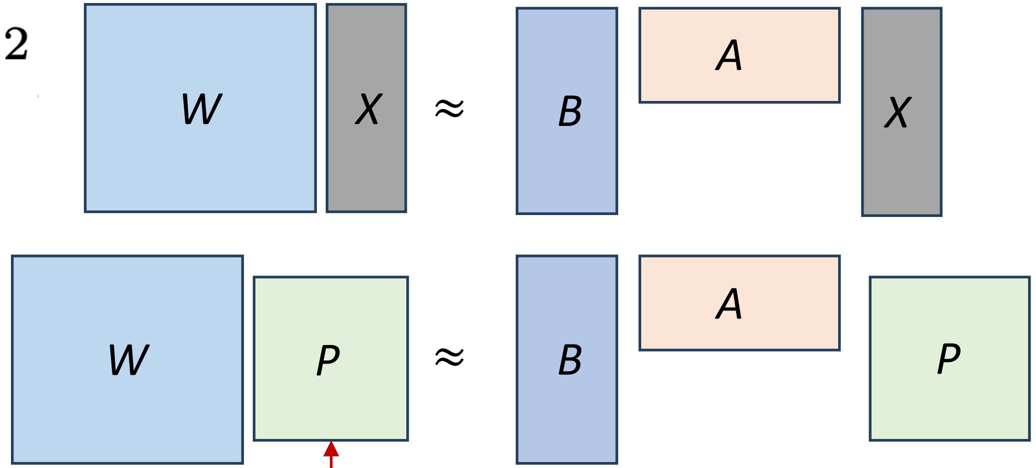
**Plain SVD:**

$$\mathcal{L}_0 = \left\| W - \hat{W} \right\|^2 = \left\| W - BA \right\|^2$$




**ASVD:**

$$\begin{aligned}\mathcal{L}_1 &= \mathbb{E}_X \left\| WX - \hat{W}X \right\|^2 = \mathbb{E}_X \left\| WX - BAX \right\|^2 \\ &= \mathrm{tr}\left[ (W - BA)\mathbb{E}_X[XX^\top](W - BA)^\top \right] \\ &= \left\| (W - BA)C^{\frac{1}{2}} \right\|^2 = \left\| WC^{\frac{1}{2}} - BAC^{\frac{1}{2}} \right\|^2\end{aligned}$$




$$\boxed{BAP = \mathsf{svd}_r[WP]}$$

Preconditioner

# Preconditioner

- Various preconditioners are introduced for model pruning, quantization, etc.
  - Original ASVD used diagonal L1-norm
  - We theoretically derived that root-covariance is optimal

| Preconditioner $P$ | Expression | Reference |
|---|---|---|
| Identity | $I$ | Plain SVD |
| Diagonal Hessian | $\mathrm{diag}[(XX^\top + \lambda I)^{-1}]^{\frac{-1}{2}}$ | OBS; GPTQ; SparseGPT |
| Diagonal $\ell_1$-norm | $\mathrm{diag}\left[\sum_j \lvert X_{1,j}\rvert, \ldots, \sum_j \lvert X_{d,j}\rvert\right]^\alpha / n^\alpha$ | ASVD; AWQ |
| Diagonal $\ell_2$-norm | $\mathrm{diag}[XX^\top]^{\frac{1}{2}}$ | Wanda |
| Covariance | $XX^\top + \lambda I$ | CorDA |
| Root-Covariance | $(XX^\top + \lambda I)^{\frac{1}{2}}$ | LatentLLM (Ours) |

Table: Variants of pre-conditioning matrices $P$ for activation-aware distillation.

# 2. Junction Matrix

- There is no unique decomposition to minimize the error
  - Any arbitrary full-rank junction matrix has no impact in performance
  - There are infinite choices to map SVD towards B and A

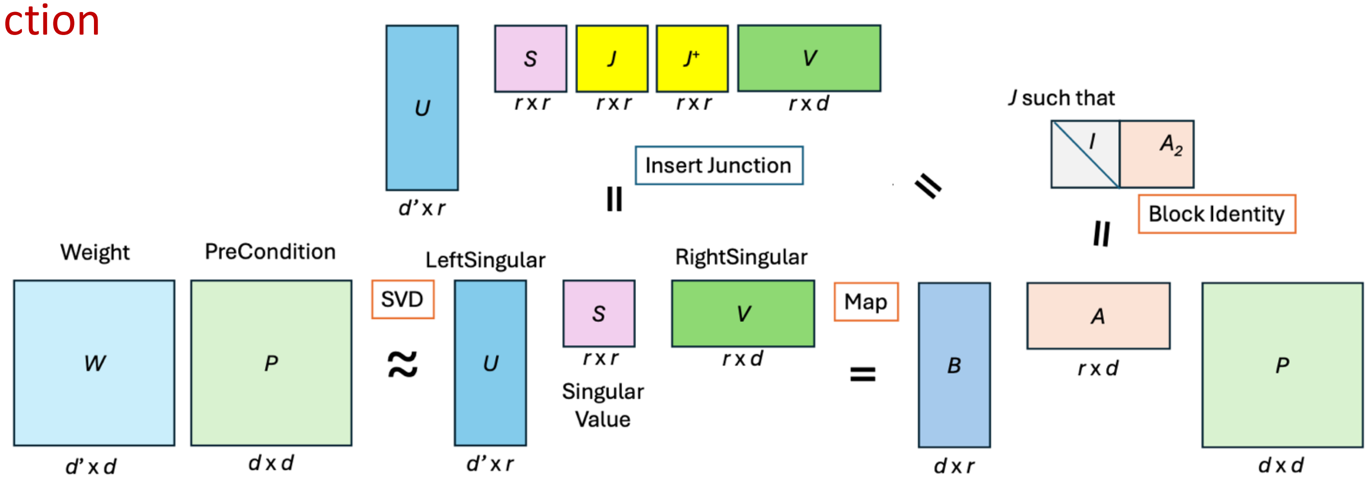
$$\boxed{USV = \mathsf{svd}_r[WP]} \quad \boxed{BAP = \mathsf{svd}_r[WP]}$$

General solution:

$$B = USJ,$$
$$A = J^{+}VP^{+},$$

Junction

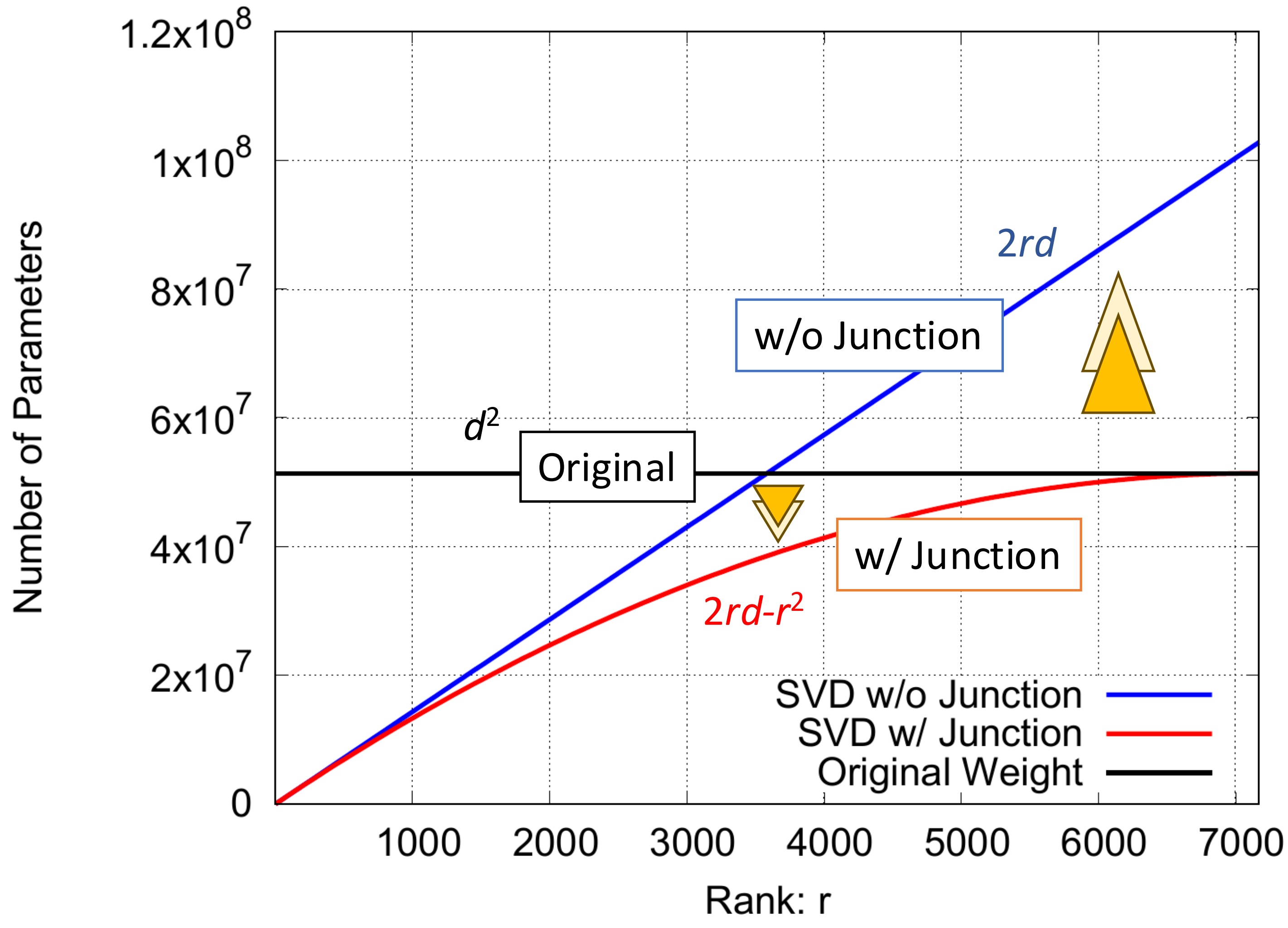
- Left singular: $J = I$;
- Right singular: $J = S^{+}$;
- Symmetry singular: $J = [S^{\frac{1}{2}}]^{+}$.
- Left block identity: $J = [US]^{+}_{:r}$
- Right block identity: $J = [V]_{:r}$

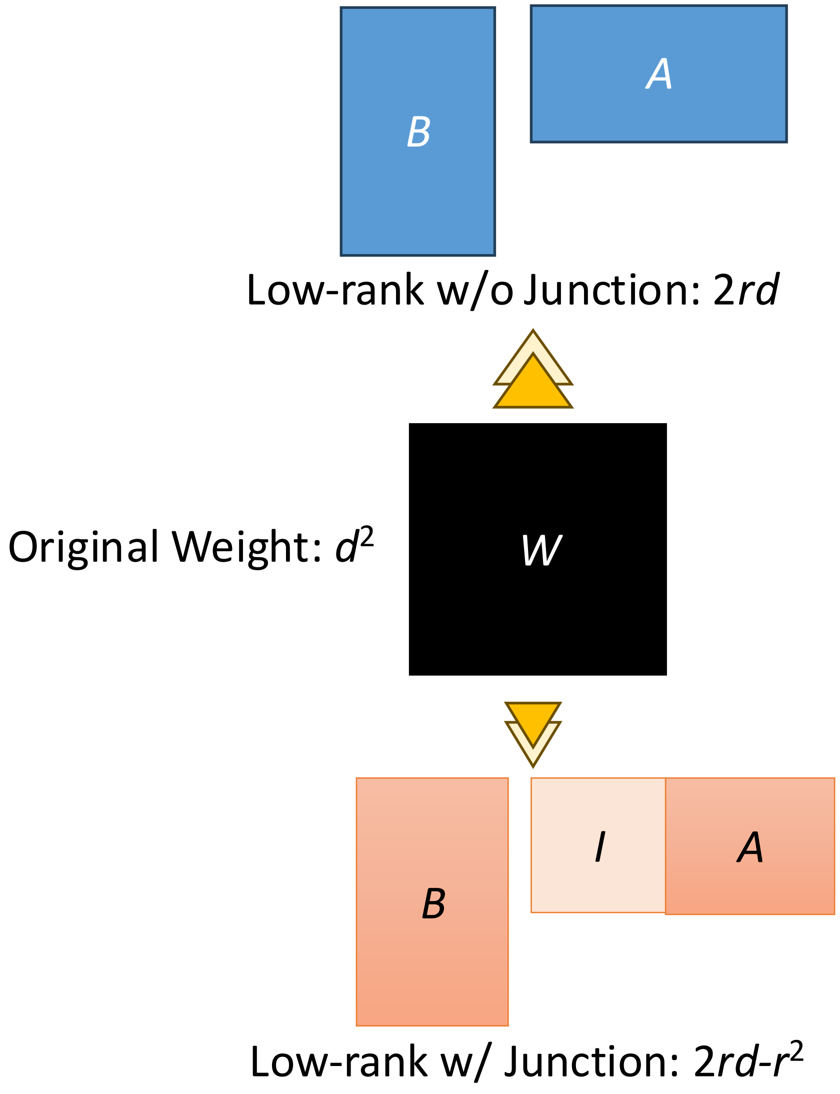



Total FLOPs/memory can be reduced

# Junction Matrix Impact

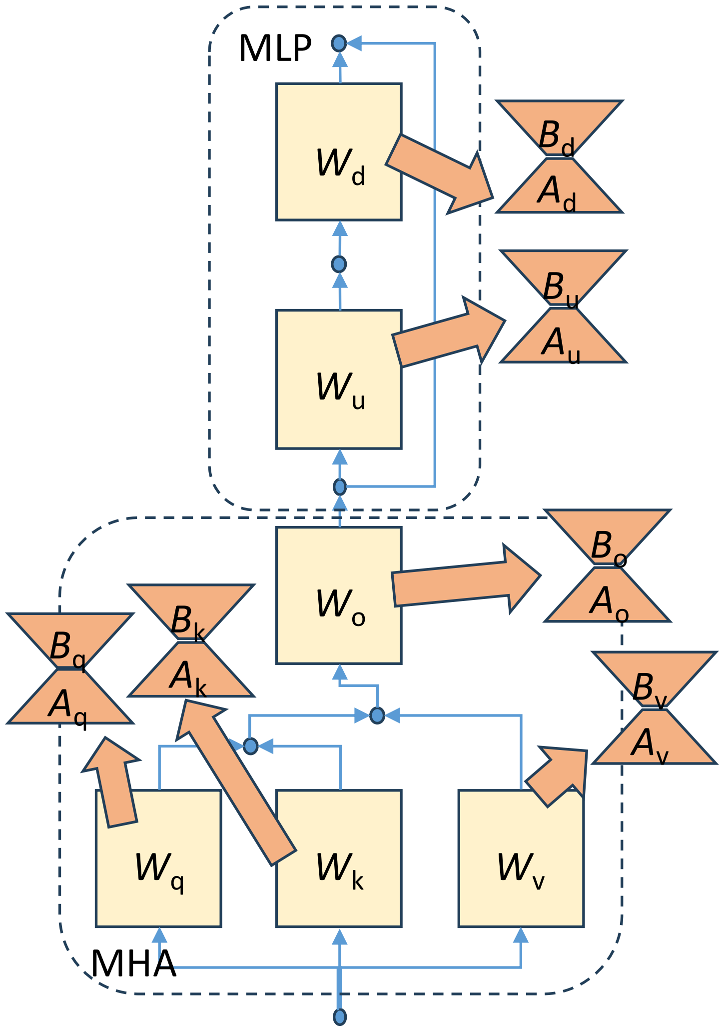
- SVD with block-identity junction can reduce the total number of parameters
  - SVD without junction can exceed the original parameter counts



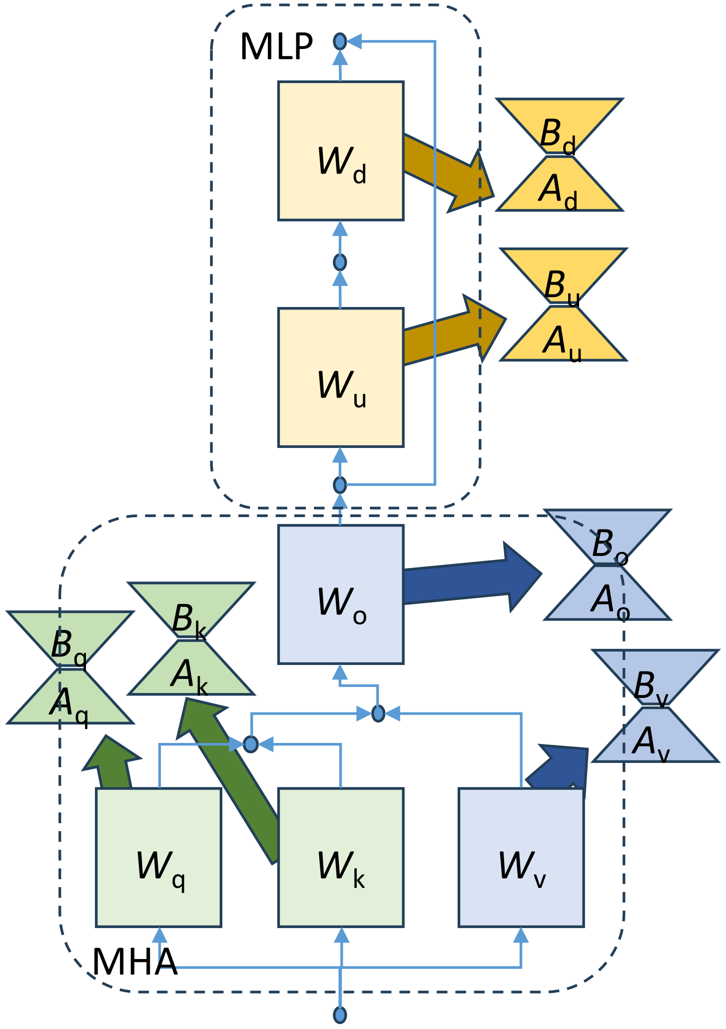



# 3. Attention-Aware Joint Rank Reduction

- We propose to compress multiple weights jointly
  - Joint QK compression
  - Joint VO compression
  - Joint UD compression




(a) LLM Local Tensor Compression




(b) LLM Joint Tensor Compression

# Joint QK Compression: High-Order Tensor Decomposition

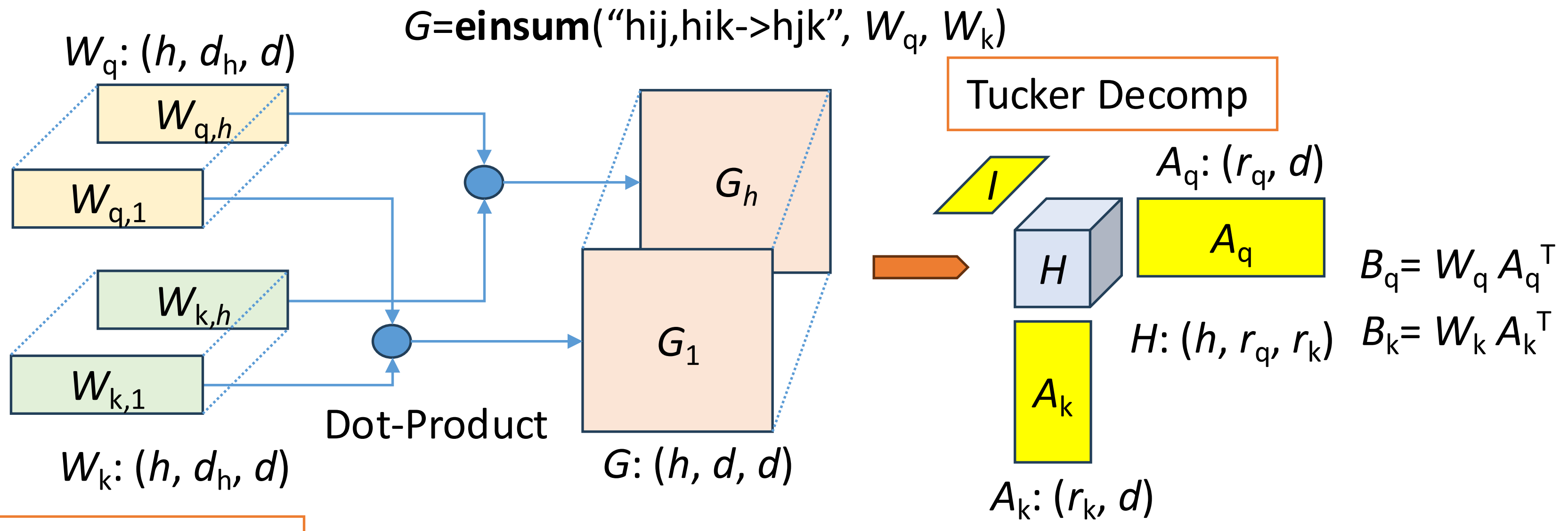
- Attention map error minimization:

$$M_i = X^\top W_{\mathrm{q},i}^\top W_{\mathrm{k},i} X,$$

$$\hat{M}_i = X^\top A_\mathrm{q}^\top B_{\mathrm{q},i}^\top B_{\mathrm{k},i} A_\mathrm{k} X,$$

$$\begin{aligned}\mathcal{L}_2 &= \sum_{i=1}^{h} \big\| M_i - \hat{M}_i \big\|^2 \\ &= \sum_{i=1}^{h} \Big\| \underbrace{C^{\frac{1}{2}} W_{\mathrm{q},i}^\top W_{\mathrm{k},} C^{\frac{1}{2}}}_{G_i \in \mathbb{R}^{d\times d}} - \underbrace{C^{\frac{1}{2}} A_\mathrm{q}^\top}_{A_\mathrm{q}'^\top} \underbrace{B_{\mathrm{q},i}^\top B_{\mathrm{k},i}}_{H_i \in \mathbb{R}^{r_\mathrm{q}\times r_\mathrm{k}}} \underbrace{A_\mathrm{k} C^{\frac{1}{2}}}_{A_\mathrm{k}'} \Big\|^2 \\ &= \sum_{i=1}^{h} \big\| G_i - A_\mathrm{q}'^\top H_i A_\mathrm{k}' \big\|^2. \qquad (14)\end{aligned}$$

Solution:
HO-SVD
(Tucker Decomposition)




Joint VO compression has similar solution

# Joint UD Compression: Decoupled Loss Minimization

- We use decoupled loss minimization trick, similar to SparseLLM

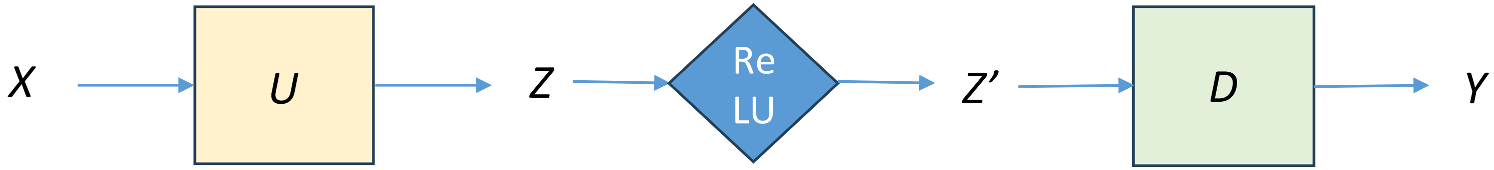
$$\mathcal{L} = \|W_\mathrm{d} Z' - \hat{W}_\mathrm{d}\sigma(\hat{W}_\mathrm{u} X)\|^2$$

$$\mathcal{L}_4 = \alpha\|W_\mathrm{u}X - Z\|^2 + \beta\|Z' - \sigma(Z)\|^2 + \gamma\|W_\mathrm{d}Z' - Y\|^2,$$

$$Z = W_\mathrm{u}X,$$
$$Z' = \sigma(Z),$$
$$Y = W_\mathrm{d}Z',$$

$$\boxed{\begin{aligned} Z_- &= W_\mathrm{u}X, \\ Z_+ &= \frac{1}{\alpha+\beta}(\alpha Z_- + \beta Z'), \end{aligned}}$$

$$\boxed{Z' = (\gamma W_\mathrm{d}^\top W_\mathrm{d} + \beta I)^+(\beta\sigma(Z) + \gamma W_\mathrm{d}^\top Y)}$$




Alternating optimization of auxiliary values Z/Z' and weight rank reduction

# Experiments: Complexity/Memory/Throughput Analysis

- FLOPs/MACs can decrease almost linearly
- Throughput improves almost quadratically
- KV cache reduces significantly

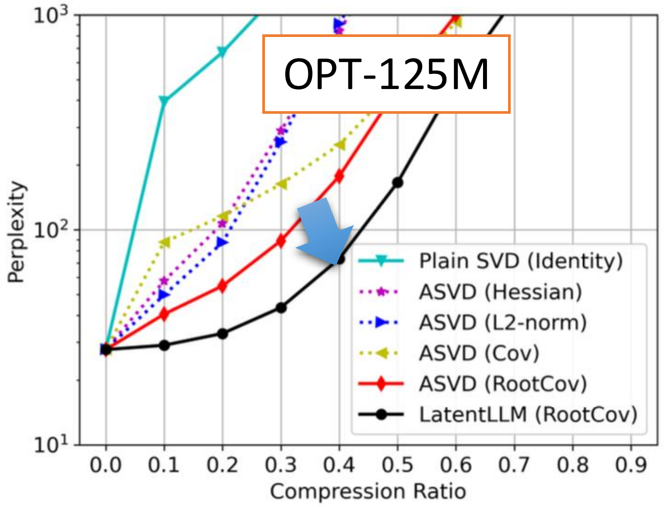
facebook
/opt-6.7b

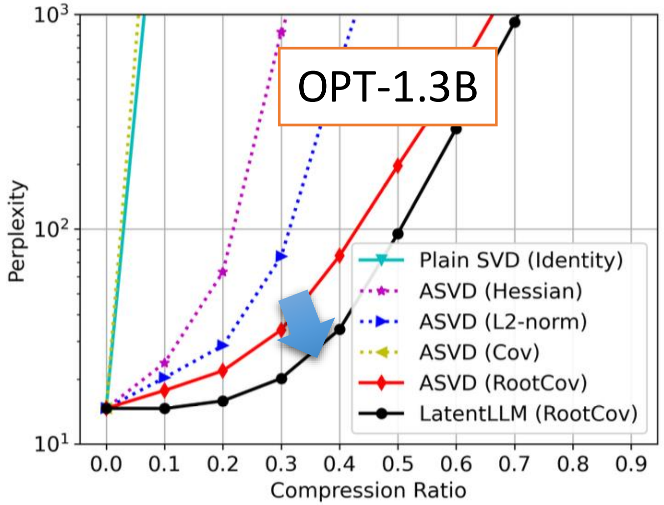
| Compression | FLOPs | MACs | Parameters (byte) | Speed (token/sec) | KV Cache (byte) |
|---|---|---|---|---|---|
| 0% | 109.0T | 54.5T | 13.32G | 6.64k | 5.37G |
| 10% | 98.1T | 49.1T | 12.40G | 6.64k | 3.67G |
| 20% | 87.2T | 43.6T | 11.06G | 7.14k | 2.97G |
| 30% | 76.3T | 38.2T | 9.74G | 7.34k | 2.43G |
| 40% | 65.4T | 32.7T | 8.40G | 8.92k | 1.98G |
| 50% | 54.5T | 27.3T | 7.08G | 9.54k | 1.57G |
| 60% | 43.6T | 21.8T | 5.74G | 11.55k | 1.21G |
| 70% | 32.7T | 16.4T | 4.42G | 13.14k | 0.88G |
| 80% | 21.8T | 10.9T | 3.08G | 16.20k | 0.57G |
| 90% | 10.9T | 5.4T | 1.76G | 19.82k | 0.28G |

LLM model: OPT-6.7B. 4-batch, 1024 tokens, torch.compile("max-autotune") on NVIDIA A40 GPU

# Experiments: LLM Benchmark

- Wikitext-2 perplexity over LLM model sizes and variants




facebook /opt-6.7b

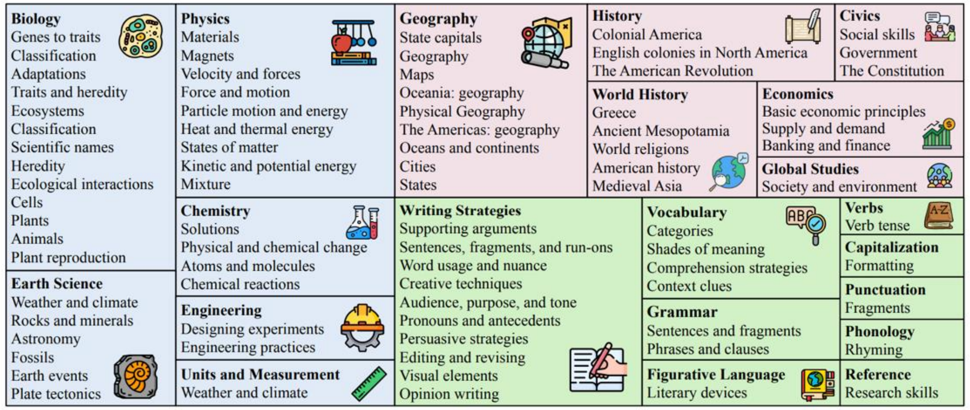
Qwen3

| | | | | | |
|---|---|---|---|---|---|
| OPT-6.7B (Perplexity: 10.9) | | | | | |
| Compression | 10% | 20% | 30% | 40% | 50% |
| Plain SVD (Identity) | 14839.0 | 67517.7 | 123286.4 | 27304.0 | 12780.0 |
| ASVD (Hessian) | 14.3 | 17.3 | 26.0 | 73.3 | 940.1 |
| ASVD ($\ell_2$-norm) | 12.6 | 14.6 | 18.7 | 30.6 | 146.4 |
| ASVD (Cov) | 9111.6 | 9842.6 | 11848.0 | 8514.7 | 8926.9 |
| ASVD (RootCov) | 11.8 | 13.5 | 17.0 | 27.2 | 56.71 |
| **LatentLLM (RootCov)** | ***10.7** | **11.5** | **13.5** | **18.0** | **33.3** |
| Qwen3-8B (Perplexity: 9.2) | | | | | |
| Plain SVD (Identity) | 2.4e5 | 9.0e6 | 2.8e7 | 5.3e7 | 4.3e8 |
| ASVD (Hessian) | 33.6 | 90.8 | 1250.8 | 5324.6 | 15933.7 |
| ASVD ($\ell_2$-norm) | 18.8 | 26.0 | 40.6 | 98.6 | 382.0 |
| ASVD (Cov) | 1.3e5 | 1.2e5 | 8.3e4 | 6.1e4 | 39455.9 |
| ASVD (RootCov) | 16.7 | 26.0 | 49.3 | 119.2 | 303.3 |
| **LatentLLM (RootCov)** | **11.8** | **14.2** | **22.4** | **53.9** | **166.3** |

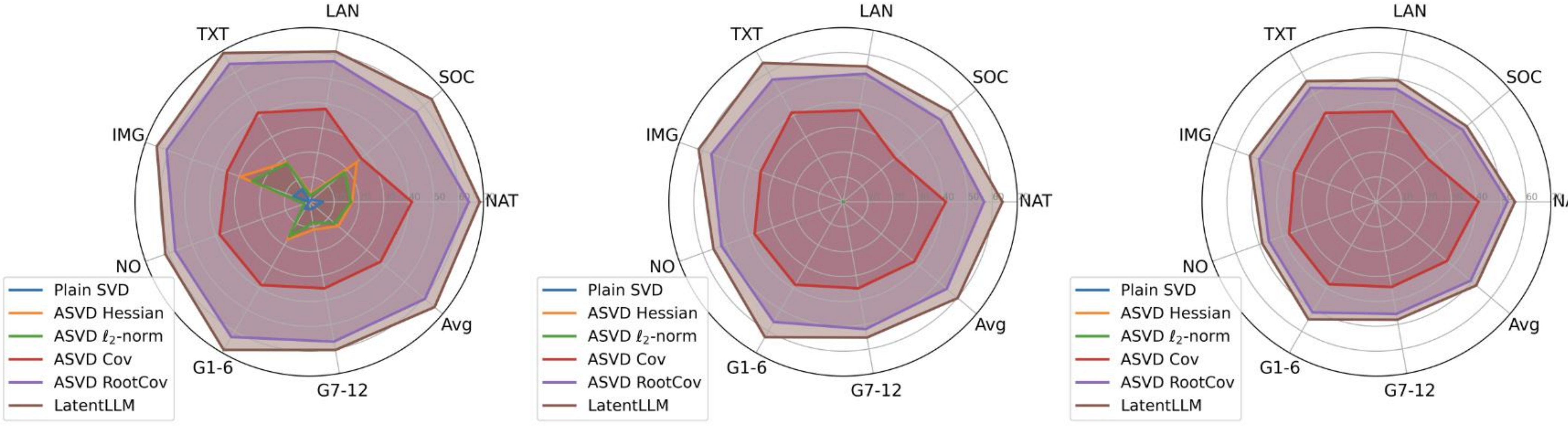
# Experiments: VLM Benchmark (ScienceQA)

- **LLaVA-7B** model for visual reasoning benchmark: **ScienceQA**

| Compression | 10% | 20% | 30% | 40% | 50% |
|---|---|---|---|---|---|
| Plain SVD (Identity) | 3.18 | 0.09 | 0.07 | 0.00 | 0.00 |
| ASVD (Hessian) | 15.21 | 2.62 | 0.00 | 0.17 | 0.00 |
| ASVD ($\ell_2$-norm) | 13.37 | 0.40 | 0.05 | 0.07 | 0.00 |
| ASVD (Cov) | 37.42 | 37.42 | 37.33 | 37.02 | 36.95 |
| ASVD (RootCov) | 60.67 | 57.53 | 54.37 | 52.23 | 49.30 |
| LatentLLM (RootCov) | **65.76** | **63.85** | **60.13** | **54.59** | **52.25** |




QA for Natural, Social, & Language Science




(a) 10% Compression (b) 30% Compression (c) 50% Compression

# Experiments: VLM Benchmark (TextVQA)

- **LLaVA-7B/Qwen2.5-VL-7B/3B** models for visual reasoning benchmark: **TextVQA**

| Compression | 10% | 20% | 30% | 40% | 50% |
|---|---|---|---|---|---|
| LLaVA-7B: Uncompressed Acc 61.32 | | | | | |
| Plain SVD (identity) | 2.36 | 0.48 | 0.35 | 0.34 | 0.36 |
| ASVD (Hessian) | 23.88 | 9.60 | 1.24 | 0.21 | 0.31 |
| ASVD ($\ell_2$-norm) | 24.41 | 9.53 | 2.77 | 0.82 | 0.75 |
| ASVD (Cov) | 0.38 | 0.36 | 0.40 | 0.33 | 0.35 |
| ASVD (RootCov) | 52.51 | 49.91 | 45.53 | 38.47 | 27.36 |
| **LatentLLM** (RootCov) | **60.06** | **57.65** | **52.63** | **46.90** | **35.94** |
| Qwen2.5-VL-7B: Uncompressed Acc 82.11 | | | | | |
| Plain SVD (identity) | 0.02 | 0.47 | 0.32 | 0.05 | 0.11 |
| ASVD (Hessian) | 58.76 | 7.03 | 0.23 | 0.45 | 0.41 |
| ASVD ($\ell_2$-norm) | 77.84 | 73.92 | 57.13 | 18.79 | 0.41 |
| ASVD (Cov) | 0.41 | 0.41 | 0.41 | 0.41 | 0.41 |
| ASVD (RootCov) | 79.46 | 74.76 | 66.31 | 51.80 | 34.91 |
| **LatentLLM** (RootCov) | **80.85** | **79.30** | **73.90** | **62.11** | **42.53** |
| Qwen2.5-VL-3B: Uncompressed Acc 78.17 | | | | | |
| Plain SVD (identity) | 0.01 | 0.08 | 0.09 | 0.09 | 0.01 |
| ASVD (Hessian) | 0.14 | 0.31 | 0.31 | 0.31 | 0.34 |
| ASVD ($\ell_2$-norm) | 44.23 | 0.14 | 0.00 | 0.41 | 0.37 |
| ASVD (Cov) | 0.41 | 0.41 | 0.41 | 0.41 | 0.41 |
| ASVD (RootCov) | 73.78 | 67.30 | 54.20 | 33.93 | 13.99 |
| **LatentLLM** (RootCov) | **76.44** | **74.29** | **64.28** | **45.80** | **19.67** |

Table: Accuracy in percent (↑) on TextVQA dataset for compressed LLaVA-7B and Qwen2.5-VL-7B/3B.


ASVD without RootCov:
Nearly 0% Acc

LatentLLM:
Best performance consistently

# Summary

- We introduced a new compression method **LatentLLM** for green AI
  - We discussed various preconditioning matrices, validating the optimality of root-covariance
  - We proposed to use junction matrix, improving the efficiency with block identity form
  - We derived a mathematically optimal joint tensor decomposition method, minimizing attention loss
  - LatentLLM can convert MHA to MLA like DeepSeek, without the need of re-training
  - We showed significant KV cache reduction and throughput improvement
  - We validated the superiority of LatentLLM over state-of-the-art rank reduction methods for various LLM/VLM models and benchmarks
- We plan:
  - to integrate pruning and quantization
  - to incorporate with fine-tuning
  - to apply to edge AI platforms
- Please contact us (koike@merl.com) for more discussions

MITSUBISHI ELECTRIC
Changes for the Better

MITSUBISHI ELECTRIC
Changes for the Better