# ORG-KV: FINDING OPTIMAL RANK AND GROUPING FOR LOW RANK KV-CACHE COMPRESSION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Low-rank KV-cache compression reduces LLM memory usage by storing compact key and value representations, but introduces a fundamental trade-off between accuracy and efficiency: Higher ranks better preserve accuracy but increase both memory and decoding costs and larger decomposition groups improve accuracy at the cost of decoding computation. Existing approaches typically fix the decomposition granularity in advance and subsequently allocate or search for ranks, potentially excluding more favorable granularity and rank configurations. We propose ORG-KV (Optimal Rank and Group KV), a low-rank KV-cache compression framework that jointly finds optimal rank for each key-value projection and decomposition granularity, under explicit KV-cache memory and decoding computation resource constraints. Specifically, we formulate finding joint rank allocation and grouping granularity as a constrained integer optimization that minimizes estimated compression error under KV-cache memory and decoding FLOPs constraints. Experiments across multiple LLM models demonstrate that ORG-KV consistently outperforms existing training-free low-rank KV-cache compression methods under the same compression budgets.

## 1 INTRODUCTION

Large language models (LLMs) have demonstrated strong performance across language understanding and generation tasks through advances in model scale and context length Team et al. (2024); Grattafiori et al. (2024). To support efficient autoregressive decoding over long contexts, the KV cache stores and reuses the key and value states of previous tokens. However, its memory footprint grows linearly with context length Kwon et al. (2023); Pope et al. (2023). Repeated access to the KV cache at each decoding step also makes memory capacity and bandwidth major bottlenecks in inference Dao et al. (2022). These limitations have motivated extensive research on KV cache compression Liu et al. (2024); Li et al. (2024); Chang et al. (2025).

Low-rank KV cache compression factorizes key and value projections into low-rank matrices and caches their low-dimensional representations, reducing cache size and decoding cost at the expense of approximation error Chang et al. (2025); Yan et al. (2026); Bhatnagar et al. (2026). The first key design factor is the **rank**: higher ranks better preserve accuracy but increase both cache memory and computation. PaLU Chang et al. (2025) further introduces group-wise decomposition, which jointly decomposes subsets of attention heads to exploit inter-head correlations while increasing decoding cost. This introduces a second design factor, **decomposition granularity**, defined by the set of heads decomposed together. Larger decomposition groups can better preserve accuracy but incur higher decoding costs, vice versa. Therefore, rank and decomposition granularity together govern the trade-off among model accuracy, KV cache size, and decoding computation.

Optimal rank and decomposition granularity vary across KV projections, making their joint allocation crucial for preserving accuracy under memory and decoding compute constraints. Existing methods instead determine them sequentially: they first fix the decomposition granularity through empirical comparisons Chang et al. (2025); Bhatnagar et al. (2026), and then allocate ranks using importance metrics such as Fisher information Chang et al. (2025); Yan et al. (2026) or training Bhatnagar et al. (2026). However, different granularity-rank combinations can yield different compression errors and decoding costs under the same cache budget. Moreover, the computational costs of key and value depend differently on decomposition granularity due to positional encoding



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.

and reconstruction matrix absorption Chang et al. (2025); Yan et al. (2026); Bhatnagar et al. (2026). Fixing granularity in advance can therefore exclude configurations that better preserve accuracy within the same resource budgets. These observations motivate jointly optimizing granularity and rank across KV projections while accounting for both cache memory and decoding computation.

ORG-KV proposes **dual-constrained granularity and rank allocation**, which jointly determines decomposition granularity and rank across all key and value projections. For each key and value projection, we construct a candidate space spanning different granularity–rank combinations and evaluate each candidate's compression error and resource costs. Compression error is measured by the change in the output distribution relative to the original model, while resource costs account for KV cache size and decoding FLOPs. We then define an allocation problem that selects one candidate per projection to minimize the sum of estimated compression errors while satisfying both resource constraints. We formulate this as a multiple-choice multidimensional knapsack problem (MMKP) Moser et al. (1997), enabling joint selection of granularity–rank combinations under global resource budgets without predetermining the decomposition granularity in advance.

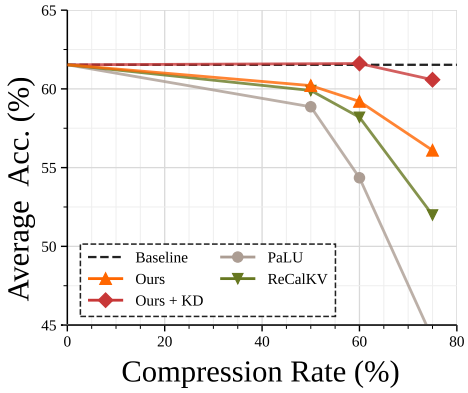


Figure 1: **Zero-shot accuracy of low-rank KV cache compression methods across compression rates.** We report average accuracy on LongChat-7B-v1.5 Li et al. (2023) over OpenBookQA Mihaylov et al. (2018), PIQA Bisk et al. (2020), ARC-Easy and ARC-Challenge Clark et al. (2018), HellaSwag Zellers et al. (2019), and WinoGrande Sakaguchi et al. (2019). ORG-KV achieves higher accuracy than PaLU Chang et al. (2025) and ReCalKV Yan et al. (2026) at the same compression rate, with larger accuracy gaps at higher compression rates.

Our main contributions are as follows. First, we propose ORG-KV, which jointly optimizes decomposition granularity and rank for each KV projection under global KV cache size and decoding FLOPs constraints. To the best of our knowledge, ORG-KV is the first KV cache compression method to jointly optimize these two factors under both memory and computation resources. Second, ORG-KV consistently outperforms existing training-free methods Chang et al. (2025); Yan et al. (2026) at different compression rate (See Figure 1) and, with efficient recovery training, achieves accuracy comparable to training-based state-of-the-art methods Bhatnagar et al. (2026). Combined with quantization, ORG-KV reduces KV cache storage by up to $\mathbf{20}\times$ 3 while maintaining competitive accuracy. Third, ORG-KV presents practical inference efficiency, achieving up to $\mathbf{2.33}\times$ attention speedup and $\mathbf{8.80}\times$ 5 relative to PyTorch's FP16 scaled dot-product attention (SDPA), which automatically selects FlashAttention-2 Dao (2024) when applicable.

## 2 Related Works

### 2.1 Low-Rank Decomposition of KV Projections

Storing keys and values in LLM as low-dimensional representations rather than at their original dimensions reduces KV cache memory usage. Low-rank decomposition methods achieve this by factorizing key and value projections into an encoder that produces a low-dimensional representation and a decoder that reconstructs it Saxena et al. (2024); Chang et al. (2025). These representations are cached and either reconstructed or used directly in attention computations during decoding Chang et al. (2025); Yan et al. (2026). The rank determines the dimensionality of the cached representations and thus the cache size. Reducing the rank saves memory but can increase information loss when approximating the original projections, potentially degrading model performance. To mitigate this trade-off, recent methods allocate different ranks across projections according to their importance. PaLU Chang et al. (2025); Yan et al. (2026) allocates ranks using a Fisher-information-based heuristic without training, whereas StarKV Bhatnagar et al. (2026) learns the allocation through training with knowledge distillation (KD). Beyond KV cache compression, LEMS Thoma et al.

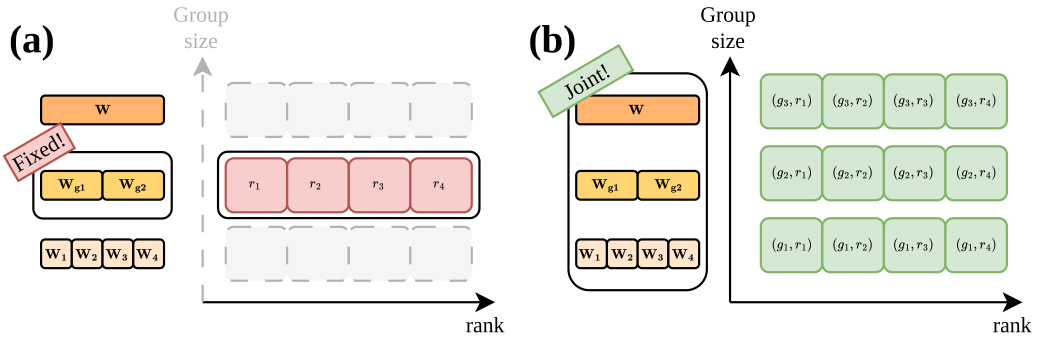


Figure 2: **Proposed ORG-KV overview compared with prior works.** (a) Prior methods predetermine the decomposition granularity and optimize ranks within the selected structure. (b) ORG-KV jointly optimizes group size and rank under KV cache compression ratio and decoding FLOPs budgets.

(2026) formulates global rank allocation for LLM weight compression as an integer linear program, minimizing a calibrated error surrogate under a parameter budget. Our method jointly selects decomposition granularity and rank through constrained optimization under KV cache and decoding FLOPs budgets, without training.

### 2.2 Decomposition Granularity and Decoding Cost

Decomposition granularity refers to the set of heads jointly decomposed to share a low-dimensional representation: head-wise decomposition treats each head independently, joint decomposition treats all heads together, and group-wise decomposition treats the heads within each group together Chang et al. (2025). Decomposing multiple heads together can exploit redundant information to reduce approximation error, but the decoding costs differ between key and value. With RoPE, past keys must be reconstructed to apply positional transformations, whereas value allows head-specific attention-weighted sums to be computed in the low-dimensional space, requiring reconstruction only of the results Chang et al. (2025); Yan et al. (2026). Thus, even at the same cache size, decoding costs vary with decomposition granularity, and the appropriate granularity may differ between key and value. Existing methods do not explicitly optimize decomposition granularity. PaLU Chang et al. (2025) adopts group-wise decomposition with a fixed number of groups and optimizes only the rank allocation, while ReCalKV Yan et al. (2026) adjusts only the head grouping under a fixed group size, without optimizing the group size itself. In contrast, we jointly optimize decomposition granularity and rank allocation without training.

## 3 ORG-KV Framework

ORG-KV proceeds through candidate-wise compression error estimation (Section 3.2), joint allocation of group sizes and ranks under resource budgets (Section 3.3), and low-rank decomposition and recovery based on the selected configuration (Section 3.4).

### 3.1 Problem Formulation

Our goal is to jointly select the decomposition granularity and rank for each key and value projection to minimize compression error under KV cache and decoding FLOPs budgets. We formulate this as a multiple-choice multidimensional knapsack problem (MMKP), selecting one candidate for each key and value projection subject to the two resource constraints Moser et al. (1997). Let $N$ denote the total number of projections. For projection $n$, we define the group size $g^{[n]}$ as the number of heads jointly decomposed within each group, and $r^{[n]}$ as the total rank allocated to that projection. The group sizes and ranks across all projections are collected into $\mathbf{g} = (g^{[1]}, \ldots, g^{[N]})$ and $\mathbf{r} = (r^{[1]}, \ldots, r^{[N]})$, respectively. Let $\mathcal{G}^{[n]}$ denote the set of admissible group sizes for projection $n$, and $\mathcal{R}^{[n]}(g^{[n]})$ the set of admissible total ranks for group size $g^{[n]}$. The candidate set for this projection

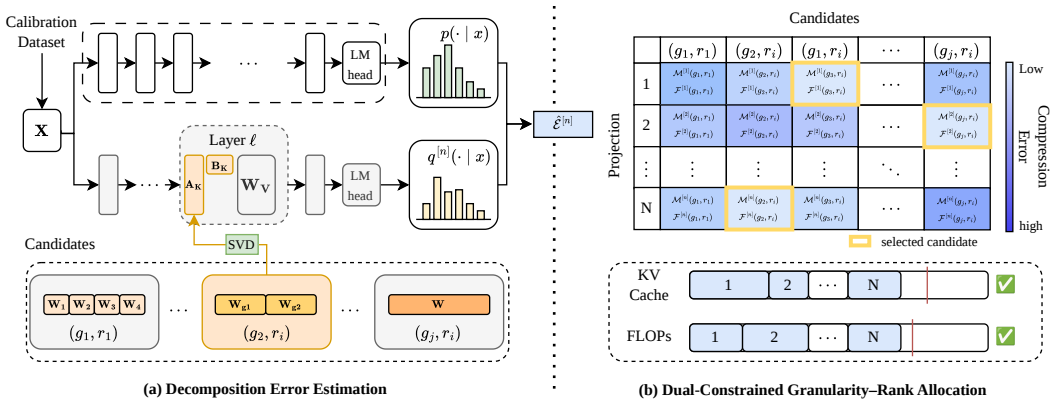


Figure 3: **Compression error estimation and granularity–rank allocation.** (a) Candidate errors are estimated from output distribution changes when compressing one projection at a time. (b) Candidates are jointly selected to minimize estimated error under KV cache and decoding FLOPs budgets. Colors indicate errors; yellow borders mark selected candidates.

is then $\mathcal{S}^{[n]} = \{(g^{[n]}, r^{[n]}) \mid g^{[n]} \in \mathcal{G}^{[n]}, r^{[n]} \in \mathcal{R}^{[n]}(g^{[n]})\}$. We define the full search space as $\mathcal{S} = \{(\mathbf{g}, \mathbf{r}) \mid (g^{[n]}, r^{[n]}) \in \mathcal{S}^{[n]}, \forall n = 1, \ldots, N\}$ (See Figure 2) and formulate the allocation problem as

$$\min_{(\mathbf{g},\mathbf{r})\in\mathcal{S}} \mathcal{E}(\mathbf{g}, \mathbf{r}) \quad \text{s.t.} \quad \sum_{n=1}^{N} \mathcal{M}^{[n]}(g^{[n]}, r^{[n]}) \le B_M, \quad \sum_{n=1}^{N} \mathcal{F}^{[n]}(g^{[n]}, r^{[n]}) \le B_F. \tag{1}$$

Here, $\mathcal{E}$ denotes the compression error, while $\mathcal{M}^{[n]}$ and $\mathcal{F}^{[n]}$ quantify the KV cache size and decoding FLOPs of projection $n$, respectively. The corresponding total budgets are denoted by $B_M$ and $B_F$.

### 3.2 Decomposition Error Estimation

Jointly allocating group sizes and ranks requires estimating the compression error of different $(g^{[n]}, r^{[n]})$ candidates for each projection. The most direct approach is to apply every candidate combination to the model and measure the resulting deviation from the original model's outputs. However, the number of combinations grows exponentially with the number of projections, making exhaustive evaluation impractical Yao et al. (2021). Moreover, the large vocabulary of LLMs makes storing and comparing full output distributions costly for each evaluation Zaidi et al. (2025). We address these challenges by (i) evaluating compression errors for each projection separately and using them to approximate the error of a complete allocation, and (ii) aggregating output probabilities to reduce the cost of each error evaluation.

**Projection-wise error approximation.** Given the candidate set $\mathcal{S}^{[n]}$ for each projection $n$, exhaustive evaluation would require evaluating $\prod_{n=1}^{N} |\mathcal{S}^{[n]}|$ compressed models. To make the allocation problem tractable, we adopt an additive approximation of the compression error. Specifically, we measure the error induced by each candidate while compressing only the corresponding projection and use it as a proxy for that projection's contribution to the overall error.

Let $\mathcal{D}$ be a calibration dataset of input contexts $x$. We denote the original model's next-token distribution by $p(\cdot \mid x)$ and the distribution obtained by applying candidate $(g, r) \in \mathcal{S}^{[n]}$ only to projection $n$, while leaving all other projections unchanged, by $q^{[n]}(\cdot \mid x; g, r)$. As illustrated in Figure 3(a), we decompose projection $n$ using the specified group size $g$ and rank $r$, then compare the original and compressed models' output distributions on the same calibration inputs. We define the compression error of candidate $(g, r)$ for projection $n$ as follows.

$$\mathcal{E}^{[n]}(g, r) = \frac{1}{|\mathcal{D}|} \sum_{x\in\mathcal{D}} D_{\mathrm{KL}}\big(p(\cdot \mid x) \| q^{[n]}(\cdot \mid x; g, r)\big). \tag{2}$$

Under the independence assumption, we approximate the total compression error of a candidate combination $(\mathbf{g}, \mathbf{r})$ as the sum of the errors associated with the selected candidates:

$$\mathcal{E}(\mathbf{g}, \mathbf{r}) \approx \sum_{n=1}^{N} \mathcal{E}^{[n]}(g^{[n]}, r^{[n]}). \tag{3}$$

This approximation may not fully capture interactions between projections when they are compressed simultaneously. By precomputing candidate errors separately, however, we reduce the number of model evaluations to $\sum_{n=1}^{N} |\mathcal{S}^{[n]}|$ while retaining the full search space $\mathcal{S}$. We discuss the validity of this independence assumption and the accuracy of the approximation under simultaneous compression in Section 5.

**Output distribution aggregation.** To efficiently evaluate candidate errors for each projection, we follow the probability aggregation approach of Zhou et al. Zhou et al. (2021): we retain the individual probabilities of the original model's top-$k$ tokens and aggregate the probability mass of all remaining tokens into a single term.

Let $p_j(x)$ and $q_j^{[n]}(x; g, r)$ denote the probabilities assigned by the original and compressed models, respectively, to the $j$-th token among the original model's top-$k$ tokens. We approximate the candidate error in equation 2 using the KL divergence between the aggregated output distributions, yielding the estimator $\widehat{\mathcal{E}}^{[n]}(g, r)$:

$$\widehat{\mathcal{E}}^{[n]}(g, r) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \left[ \sum_{j=1}^{k} p_j(x) \log \frac{p_j(x)}{q_j^{[n]}(x; g, r)} + \left(1 - \sum_{j=1}^{k} p_j(x)\right) \log \frac{1 - \sum_{j=1}^{k} p_j(x)}{1 - \sum_{j=1}^{k} q_j^{[n]}(x; g, r)} \right]. \tag{4}$$

This reduces the KL divergence calculation from the full vocabulary to $k + 1$ terms. In practice, we use $\widehat{\mathcal{E}}^{[n]}$ in place of $\mathcal{E}^{[n]}$ in equation 3 for allocation. Since candidate errors are computed independently of the resource budgets, the same error table can be reused for allocation under different KV cache and decoding FLOPs budgets.

### 3.3 Dual-Constrained Granularity-Rank Allocation

**KV cache cost.** For a batch size of 1, the KV cache cost of candidate $(g, r)$ is $\mathcal{M}^{[n]}(g, r) = sTr$, where $T$ is the cache length and $s$ is the storage size per value in bytes.

**Decoding FLOPs cost.** Let $C_{\text{in}}$ and $C_{\text{out}}$ denote the input and output dimensions of a projection, respectively, and $H_{\text{dim}}$ the head dimension. For multi-head attention, the decoding FLOPs of candidate $(g, r)$ are

$$\mathcal{F}^{[n]}(g, r) = 2C_{\text{in}}r + \begin{cases} 2TgH_{\text{dim}}r + 3TC_{\text{out}}, & \text{key}, \\ 2Tgr + 2gH_{\text{dim}}r, & \text{value}. \end{cases} \tag{5}$$

For grouped-query attention, the attention-sum and recovery costs in the value path are scaled by the number of query heads sharing each KV head. Details are provided in Appendix A.3.

**Global allocation.** We combine the estimated compression errors with resource costs to construct a candidate table for each projection, as shown in Figure 3(b). Let $N$ denote the number of projections, $C$ the number of candidates per projection, and $F$ the cost of evaluating one model configuration on the calibration data. Exhaustively evaluating all candidate combinations requires $O(C^N F)$ computation, whereas our additive approximation reduces this evaluation cost to $O(NCF)$ by evaluating candidates independently for each projection. We then approximate the objective in equation 1 by $\sum_{n=1}^{N} \widehat{\mathcal{E}}^{[n]}(g^{[n]}, r^{[n]})$ and solve the resulting MMKP using the branch-and-cut MIP solver in HiGHS to minimize the total estimated compression error while satisfying both resource budgets Huangfu & Hall (2018). Since the candidate tables are computed independently of the resource budgets, they can be reused across budgets without re-evaluating candidate errors. The complexity of solving the MMKP is discussed in Appendix A.2.

| Training -Free | Model/Method | Comp (%) | Perplexity ↓ | | | Zero-Shot Accuracy (%) ↑ | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Wiki2 | C4 | lm-avg. | OBQA | PIQA | ARC-e | ARC-c | Hella | Wino | Avg. |
| | **LongChat-7B-v1.5** | 0 | 6.86 | 9.93 | 8.40 | 41.00 | 76.28 | 71.84 | 41.38 | 71.20 | 67.48 | 61.53 |
| ✓ | PaLU | 60 | 8.31 | 13.68 | 10.99 | 37.60 | 69.42 | 60.02 | 34.90 | 62.39 | 61.80 | 54.35 |
| ✓ | ReCalKV | 60 | 7.36 | 11.43 | 9.39 | 40.40 | 72.91 | 68.18 | 37.29 | 65.93 | **64.48** | 58.20 |
| ✓ | Ours | 60 | **7.22** | **11.15** | **9.18** | **41.60** | **74.76** | **68.48** | **38.99** | **66.78** | 64.33 | **59.15** |
| | STAR-KV | 60 | 6.83 | 9.98 | 8.41 | **41.80** | 75.90 | **72.69** | **41.47** | 70.49 | 66.85 | 61.53 |
| | Ours + KD | 60 | **6.28** | **9.29** | **7.78** | 41.40 | **76.55** | 72.18 | 41.04 | **71.56** | **66.93** | **61.61** |
| | STAR-KV | 75 | 7.34 | 10.44 | 8.89 | **42.60** | 74.86 | 71.63 | **41.55** | 68.52 | 63.77 | 60.49 |
| | Ours + KD | 75 | **6.51** | **9.91** | **8.21** | 40.40 | **75.52** | **71.68** | 40.36 | **69.79** | **65.75** | **60.58** |
| | **LLaMA-2-7B** | 0 | 5.12 | 7.04 | 6.08 | 44.20 | 78.07 | 76.30 | 46.42 | 76.00 | 69.30 | 65.05 |
| ✓ | PaLU | 60 | 6.82 | 10.73 | 8.77 | 39.40 | 73.99 | 68.69 | 38.31 | 68.49 | 63.46 | 58.72 |
| ✓ | ReCalKV | 60 | 5.82 | 8.87 | 7.35 | 40.80 | 75.79 | 72.98 | 41.30 | 72.08 | 67.32 | 61.71 |
| ✓ | Ours | 60 | **5.43** | **8.02** | **6.73** | **43.40** | **77.20** | **73.15** | **42.66** | **73.49** | **68.98** | **63.15** |
| | STAR-KV | 60 | 5.49 | **7.45** | **6.47** | 43.40 | **79.22** | 75.55 | **44.71** | 74.63 | 68.51 | **64.34** |
| | Ours + KD | 60 | **5.39** | 7.79 | 6.59 | **44.60** | 77.53 | **75.72** | 44.28 | **74.90** | **68.75** | 64.29 |
| | STAR-KV | 75 | 5.86 | **8.02** | **6.94** | 43.00 | **78.18** | 73.78 | 41.64 | 73.14 | 66.61 | 62.73 |
| | Ours + KD | 75 | **5.70** | 8.45 | 7.07 | **43.80** | 76.50 | **73.82** | **42.41** | **73.53** | **67.40** | **62.91** |
| | **LLaMA-3-8B-Instruct** | 0 | 7.74 | 12.61 | 10.18 | 43.00 | 78.51 | 81.61 | 56.57 | 75.95 | 71.59 | 67.87 |
| ✓ | PaLU | 60 | 15.09 | 33.15 | 24.12 | 38.20 | 70.84 | 55.85 | 36.35 | 62.10 | 65.19 | 54.75 |
| ✓ | ReCalKV | 60 | 16.16 | 29.27 | 22.71 | 38.80 | 70.29 | 52.90 | 35.67 | 63.55 | 64.56 | 54.30 |
| ✓ | Ours | 60 | **8.55** | **16.92** | **12.74** | **41.20** | **76.17** | **76.30** | **48.12** | **69.01** | **68.90** | **63.29** |
| | STAR-KV | 60 | 8.01 | 10.81 | 9.41 | **44.00** | **80.52** | 79.80 | 52.13 | 78.79 | 71.11 | 67.73 |
| | Ours + KD | 60 | **5.73** | **9.60** | **7.67** | **44.00** | 80.25 | **81.31** | **54.86** | **80.09** | **71.35** | **68.65** |

Table 1: **KV cache compression under training-free and training-based settings.** We report perplexity on WikiText-2 and C4 and zero-shot accuracy on six commonsense reasoning benchmarks. lm-avg. and Avg. denote the mean perplexity and mean accuracy, respectively. Bold indicates the best result among compressed methods for each model.

### 3.4 Decomposition and Recovery

**Group-wise low-rank decomposition.** For each candidate $(g, r) \in \mathcal{S}^{[n]}$ of projection $n$, we partition the heads into groups of size $g$ and apply K-FAC-based weighted SVD to each group's weight matrix Li et al. (2026). The low-rank factors incorporate the covariances of input activations and output gradients collected from the calibration data. The total projection rank $r$ is allocated across groups based on their singular values Wang et al. (2025). Specifically, we maximize the sum of squared retained singular values subject to the group ranks summing to $r$, allowing groups of the same size to receive different ranks. This decomposition procedure is also used to construct the candidate error tables.

**Efficient recovery.** To recover performance after compression, we perform KD using the original model as the teacher Kangaslahti et al. (2026). We train only the low-rank encoder and decoder factors of the key and value projections, keeping all other model parameters frozen. The training objective is $\mathcal{L}_{\text{recovery}} = \mathcal{L}_{\text{KL}} + \lambda \mathcal{L}_{\text{CE}}$, where $\mathcal{L}_{\text{KL}}$ denotes $D_{\text{KL}}(p_{\text{teacher}} \| p_{\text{student}})$ averaged over the training data. Here, $p_{\text{teacher}}$ and $p_{\text{student}}$ are the output distributions of the original and compressed models, respectively. $\mathcal{L}_{\text{CE}}$ is the cross-entropy on the ground-truth tokens, and $\lambda$ controls its weight.

## 4 Experiments

### 4.1 Experimental Setup

**Models and baselines.** To evaluate effectiveness across MHA and GQA architectures, we use LongChat-7B-v1.5 Li et al. (2023), LLaMA-2-7B Touvron et al. (2023), LLaMA-3-8B-Instruct Grattafiori et al. (2024), LLaMA-3.1-8B-Instruct Grattafiori et al. (2024), and Mistral-7B-Instruct-v0.2 Jiang et al. (2023). We compare against PaLU Chang et al. (2025) and ReCalKV Yan et al. (2026) without recovery training, and STAR-KV Bhatnagar et al. (2026) with recovery training. We define the compression rate as the percentage reduction in KV cache storage relative to the original model and evaluate primarily at compression rates of 60% and 75%.

**Evaluation.** We measure perplexity on WikiText-2 Biderman et al. (2026) and C4 Raffel et al. (2020) using Hugging Face Wolf et al. (2020), with an evaluation sequence length of 4K. We eval-

uate zero-shot and long-context performance using LM Evaluation Harness Biderman et al. (2026). Zero-shot evaluation covers OpenBookQA Mihaylov et al. (2018), PIQA Bisk et al. (2020), ARC-Easy Clark et al. (2018), ARC-Challenge Clark et al. (2018), HellaSwag Zellers et al. (2019), and WinoGrande Sakaguchi et al. (2019). We report individual task scores and their average across the six tasks. For long-context evaluation, we assess LongChat-7B-v1.5 Li et al. (2023) and LLaMA-3.1-8B-Instruct Grattafiori et al. (2024) on seven tasks from LongBench Bai et al. (2024) and nine tasks from RULER Hsieh et al. (2024), with the RULER input length set to 4K.

**Implementation details.** We use WikiText-2 to collect statistics for low-rank decomposition and to estimate candidate compression errors. The decomposition and allocation settings are detailed in Appendices A.7 and A.6, respectively. For recovery training, we use FinePDFs-Edu Kydlíček et al. (2025); the training settings are given in Appendix A.5. We apply mixed-precision quantization to the recovered models as described in Appendix A.1. For the quantization experiments, average zero-shot accuracy is computed over five tasks, excluding WinoGrande Sakaguchi et al. (2019).

### 4.2 Main Results

**Training-free compression.** Table 1 presents results at 60% KV cache compression without recovery training. ORG-KV achieves lower average perplexity and higher average zero-shot accuracy than PaLU and ReCalKV across all three models. On LLaMA-3-8B-Instruct in particular, it achieves an average accuracy of 63.29%, exceeding PaLU and ReCalKV by 8.54 and 8.99 percentage points, respectively. Its average perplexity is 12.74, compared with 24.12 and 22.71 for the two methods.

**Compression with recovery.** With performance recovery with KD, ORG-KV achieves average zero-shot accuracy comparable to or higher than STAR-KV (Table 1). On LongChat, it achieves accuracies of 61.61% and 60.58% at compression rates of 60% and 75%, respectively, with lower perplexity than STAR-KV in both settings. On LLaMA-2, accuracy is comparable, with differences within 0.2 percentage points, while STAR-KV achieves lower C4 perplexity.

| Method | Comp. | LongBench | RULER |
|---|---|---|---|
| LongChat-7B-v1.5 | 0% | 40.81 | 88.80 |
| PaLU | 50% | 36.82 | 84.06 |
| STAR-KV | 60% | **39.09** | 88.26 |
| Ours + KD | 60% | 38.73 | **89.93** |
| LLaMA-3.1-8B-Instruct | 0% | 42.23 | 96.94 |
| PaLU | 50% | 35.25 | 87.39 |
| STAR-KV | 60% | **39.58** | 89.03 |
| Ours + KD | 60% | 39.07 | **94.47** |

Table 2: **Average results on LongBench Bai et al. (2024) and RULER Hsieh et al. (2024),** PaLU uses 50% KV cache compression, while STAR-KV and ours use 60%. Bold indicates the best result among compressed methods for each model.

**Long-context performance.** Tables 2 present results on LongBench and RULER. On LongBench, ORG-KV achieves average scores of 38.73 and 39.07 on LongChat and LLaMA-3.1, respectively, trailing STAR-KV by 0.36 and 0.51 points. On RULER, however, ORG-KV achieves average scores of 89.93 and 94.47, surpassing STAR-KV by 1.67 and 5.44 points, respectively. Across both benchmarks and models, ORG-KV at 60% compression also achieves higher average scores than PaLU at 50% compression. Detailed experimental results are provided in Appendix A.8.

### 4.3 Combination with quantization

Our allocation framework can be extended to mixed-precision quantization of latent representations for further KV cache compression. By constructing candidates with different channel-band split ratios and bit widths for each projection, we can formulate the allocation as an MCKP that minimizes the sum of estimated quantization errors under a global average-bit budget Kellerer et al. (2004). Details are provided in Appendix A.1.

Table 3 reports results on LongChat-7B-v1.5 with average 3.2-bit quantization. At 60% and 75% low-rank compression, average zero-shot accuracies are 60.33% and 59.57%, respectively, exceeding STAR-KV by 0.21%p and 0.39%p at the same compression factors. Combining 75% low-rank compression with quantization reduces KV cache storage to $20\times$ of its original size, excluding quantization metadata. Average accuracy decreases by 0.77%p compared with the uncompressed model.

### 4.4 Ablation Study

Table 4 presents ablation results on LongChat-7B-v1.5 across different KV cache compression rates, using the mean zero-shot accuracy over six commonsense reasoning tasks as the evaluation metric.

**Effect of Optimization-based Rank Allocation.** Unlike PaLU Chang et al. (2025), which allocates ranks using a Fisher-information-based heuristic with a fixed group size, our rank-only variant formulates rank allocation as an optimization problem under the same fixed-granularity setting. As shown in the second row of Table 4, our optimization-based rank allocation consistently achieves higher accuracy than PaLU across all compression rates.

**Effect of joint granularity–rank allocation.** The second and third rows of Table 4 compare rank-only allocation with a fixed group size of 4 against joint optimization of granularity and rank. Joint optimization consistently improves accuracy across all compression rates, with the gain becoming more pronounced at higher compression. At 75% compression, it improves the mean zero-shot accuracy from 47.88% to 56.10%, a gain of 8.22%p.

**Effect of the Recovery Objective.** The third and fourth rows of Table 4 evaluate the effect of recovery training after joint granularity–rank optimization. Using FinePDFs-Edu for recovery training consistently improves accuracy at higher compression rates. At 75% compression, the mean zero-shot accuracy increases from 56.10% to 60.58%, demonstrating that the recovery objective effectively mitigates the performance loss caused by aggressive KV cache compression. Detailed training settings are provided in Appendix A.5.

| Method | Bit | Comp. | Avg. (%) |
|---|---|---|---|
| LongChat-7B-v1.5 | 16 | 1× | 60.34 |
| KIVI-2-gs32-r32 | 3.16 | 5.1× | 60.30 |
| KVQuant-2bit-1% | 2.33 | 6.9× | 58.00 |
| PaLU (50%) | 3 | 10.4× | 56.42 |
| PaLU (50%) | 4 | 8.0× | 56.71 |
| STAR-KV (60%) | 3.2 | 12.5× | 60.12 |
| Ours + KD (60%) | 3.2 | 12.5× | **60.33** |
| STAR-KV (75%) | 3.2 | 20.0× | 59.18 |
| Ours + KD (75%) | 3.2 | 20.0× | **59.57** |

Table 3: **Quantization results on LongChat-7B-v1.5 Li et al. (2023).** Accuracy is averaged over OpenBookQA Mihaylov et al. (2018), PIQA Bisk et al. (2020), ARC-Easy, ARC-Challenge Clark et al. (2018), and HellaSwag Zellers et al. (2019). Ours uses average 3.2-bit mixed-precision quantization after recovery (Appendix A.1). Our compression factors exclude quantization metadata. Bold marks the higher accuracy between STAR-KV and ours at the same compression factor.

| Method | | Compression (%) | | |
|---|---|---|---|---|
| Joint Opt. | KD | 50 | 60 | 75 |
| PaLU | | 58.86 | 54.35 | 44.15 |
| × | × | 58.92 | 58.01 | 47.88 |
| ✓ | × | 60.21 | 59.20 | 56.10 |
| ✓ | ✓ | – | 61.61 | 60.58 |

Table 4: **Effect of joint optimization and KD.** We report mean zero-shot accuracy (%) over six tasks on LongChat-7B-v1.5.

## 5 Discussion

**Post-hoc analysis of the additive approximation.** We evaluated the selected allocation and 200 randomly generated candidate allocations under the same resource constraints at 60% compression on LongChat-7B-v1.5. Although additive KL underestimates the actual KL, as shown in Figure 4(a), it largely preserves the relative ordering of candidate allocations. Figure 4(b) shows a high Spearman rank correlation of 0.954 between additive KL and actual KL. These results support using additive KL as a proxy for comparing allocations within the evaluated set.

**Post-hoc analysis of the top-$k$ approximation.** Figure 4(c) compares full-vocabulary KL with top-64+tail KL across individual projection compression candidates on LongChat-7B-v1.5. The points lie close to the identity line, and the two measures have a Pearson correlation of 0.9995. Thus, top-64+tail KL closely tracks full-vocabulary KL for the evaluated candidates.

**Decoding throughput.** Table 5 reports decoding throughput on a single RTX 4090. We use PyTorch's SDPA with an uncompressed FP16 KV cache as an optimized baseline. SDPA automatically selects an available backend, including FlashAttention-2 Dao (2024), which accelerates attention

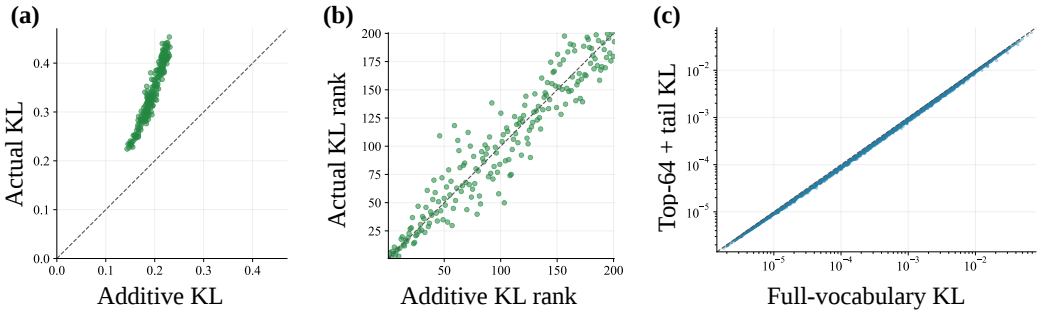


Figure 4: **Post-hoc analysis of KL approximations on LongChat-7B-v1.5.** (a) Additive KL versus actual KL. (b) Allocation ranks according to additive KL and actual KL. Panels (a) and (b) show 200 candidate allocations satisfying the same resource constraints at 60% compression. (c) Full-vocabulary KL versus top-64+tail KL for individual projection candidates.

through reduced memory traffic and improved GPU parallelism, when applicable. Each method generates 4K tokens per sequence at its maximum feasible batch size.

With an 8K input, ORG-KV with mixed-bit latent KV caches achieves $8.80\times$ the decoding throughput of SDPA. Its maximum feasible batch size is 22, compared with 1 for SDPA; the throughput gain therefore includes the benefit of increased batch capacity. At 16K, ORG-KV achieves $8.06\times$ the throughput of SDPA under the separate FP32 decoding MLP and residual setting used for both methods. Detailed measurement settings are provided in Appendix A.4.

| | Throughput (tokens/s) ↑ | | Speedup ↑ |
|---|---|---|---|
| **Tokens** | **SDPA** | **Ours (Mixed-bit)** | |
| 1K | 164.21 (4) | **795.14** (52) | **4.84×** |
| 2K | 118.81 (3) | **623.69** (43) | **5.25×** |
| 4K | 74.94 (2) | **440.01** (33) | **5.87×** |
| 8K | 31.22 (1) | **274.73** (22) | **8.80×** |
| 16K† | 12.51 (1) | **100.90** (12) | **8.06×** |

Table 5: **Decoding throughput of LLaMA-2-7B.** We compare PyTorch SDPA with an uncompressed FP16 KV cache against our method with a mixed-bit latent KV cache on a single RTX 4090. Each method generates 4K tokens per sequence at its maximum feasible batch size. Tokens denotes the input context length; parentheses indicate batch sizes, and speedups are relative to SDPA. †: At 16K, both methods use FP32 for decoding MLP and residual operations to prevent FP16 overflow.

## 6 Conclusion

We present ORG-KV, a low-rank KV cache compression framework that jointly allocates decomposition granularity and rank under KV cache memory and decoding FLOPs constraints. Without fixing granularity in advance, ORG-KV selects a configuration for each key and value projection through integer optimization based on candidate-wise output errors (Section 3.3). Recovery training updates only the low-rank factors to restore performance (Section 3.4), and the allocation framework extends to mixed-precision quantization of latent representations for further memory reduction (Section 4.3). Post-hoc analysis shows that additive KL underestimates actual KL, but their allocation rankings have a Spearman correlation of 0.954. This supports using additive KL as a proxy for comparing allocations within the evaluated set (Section 5). Across multiple LLMs, ORG-KV achieves lower average perplexity and higher average zero-shot accuracy than existing training-free methods at the same compression rates. With recovery training, it achieves average accuracy comparable to or higher than STAR-KV (Table 1). On LongChat, combining low-rank compression with quantization reduces KV cache storage to $1/20$ of its original size, excluding quantization metadata, with an average accuracy drop of 0.77%p relative to the uncompressed model (Table 3). Inference experiments show up to $8.80\times$ the decoding throughput of SDPA, including the benefit of processing larger batches (Table 5).

## REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.298. URL https://aclanthology.org/2023.emnlp-main.298/.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Priyansh Bhatnagar, Ashkan Moradifirouzabadi, Se-Hyun Yang, SeungJae Lee, Jungwook Choi, and Mingu Kang. STAR-KV: Low-rank KV cache compression via soft thresholding for adaptive rank control. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lJjH1q6RwY.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models, 2026. URL https://arxiv.org/abs/2405.14782.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/6239.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S. Abdelfattah, and Kai-Chiang Wu. Palu: KV-cache compression with low-rank projection. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LWMS4pk2vK.

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, 2018. URL https://arxiv.org/abs/1803.05457.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mZn2Xyh9Ec.

Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Re. Flashattention: Fast and memory-efficient exact attention with IO-awareness. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=H4DqfPSibmx.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin,

Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias

Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Coleman Richard Charles Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W. Mahoney, Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=0LXotew9Du.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

QHJAJ Huangfu and JA Julian Hall. Parallelizing the dual revised simplex method. *Mathematical programming computation*, 10(1):119–142, 2018.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Sara Kangaslahti, Nihal V. Nayak, Jonathan Geuter, Marco Fumero, Francesco Locatello, and David Alvarez-Melis. Boomerang distillation enables zero-shot model size interpolation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4ZU8v4s3IR.

Hans Kellerer, Ulrich Pferschy, and David Pisinger. *The Multiple-Choice Knapsack Problem*, pp. 317–347. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-24777-7. doi: 10.1007/978-3-540-24777-7_11. URL https://doi.org/10.1007/978-3-540-24777-7_11.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Hynek Kydlíček, Guilherme Penedo, and Leandro von Werra. Finepdfs. https://huggingface.co/datasets/HuggingFaceFW/finepdfs_edu, 2025.

Dacheng Li, Rulin Shao, Anze Xie, Ying Sheng, Lianmin Zheng, Joseph Gonzalez, Ion Stoica, Xuezhe Ma, and Hao Zhang. How long can context length of open-source LLMs truly promise? In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023. URL https://openreview.net/forum?id=LywifFNXV5.

Yuhang Li, Donghyun Lee, Ruokai Yin, and Priyadarshini Panda. Optimal brain decomposition for accurate llm low-rank approximation. *arXiv preprint arXiv:2604.00821*, 2026.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=L057s2Rq8O.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. Fineweb-edu: the finest collection of educational content, 2024. URL https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu.

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 Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Martin Moser, Dusan P Jokanovic, and Norio Shiratori. An algorithm for the multidimensional multiple-choice knapsack problem. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 80(3):582–589, 1997.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

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. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019. URL https://arxiv.org/abs/1907.10641.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. Eigen attention: Attention in low-rank space for kv cache compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15332–15344, 2024.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdieh, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry

Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snaider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurumurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayana Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tsihlas, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy

Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rrustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiujia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khuslen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xiang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirnschall, Weiyi Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C. Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhuyun Xiao, Rupert Kemp, Huaixiu Steven Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Wojciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Ko-

mal Jalan, Kalesha Bullard, Achintya Singhal, Thang Luong, Boyu Wang, Sujeevan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Kopparapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturel, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Villela, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Tsendsuren Munkhdalai, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Ruddock, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnapalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Bartek Perz, Wooyeol Kim, Nandita Dukkipati, Anthony Baryshnikov, Christos Kaplanis, XiangHai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Kiam Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Cho-Jui Hsieh, Irene Cai, Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadsy, Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kepa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohman, Junwen Bai,

Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeff Dean, and Oriol Vinyals. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

Moritz Thoma, Maximilian Groezinger, Maximilian Forstenhäusler, Emad Aghajanzadeh, Manoj Rohit Vemparala, Christos Anagnostopoulos, Pierpaolo Mori, Nael Fasfous, Alexander Frickenstein, Daniel Mueller-Gritschneder, and Ulf Schlichtmann. Advancing SVD-based LLM compression via layer-wise error model search. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=IjIgNPFuCt.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Xin Wang, Samiul Alam, Zhongwei Wan, Hui Shen, and Mi Zhang. SVD-LLM v2: Optimizing singular value truncation for large language model compression. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4287–4296, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.217. URL https://aclanthology.org/2025.naacl-long.217/.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Xianglong Yan, Zhiteng Li, Tianao Zhang, Haotong Qin, Linghe Kong, Yulun Zhang, and Xiaokang Yang. RecalKV: Low-rank KV cache compression via head reordering and offline calibration, 2026. URL https://openreview.net/forum?id=LQ1qYDOk6o.

Zhewei Yao, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael Mahoney, and Kurt Keutzer. Hawq-v3: Dyadic neural network quantization. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11875–11886. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/yao21a.html.

Mohd Abbas Zaidi, Akhil Kedia, Jinwoo Ahn, Taehwak Kwon, Kangwook Lee, Haejun Lee, Joohyung Lee, et al. Sparse logit sampling: Accelerating knowledge distillation in llms. In *Pro-*

*ceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18085–18108, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Xuan Zhou, Xiao Zhang, Chenyang Tao, Junya Chen, Bing Xu, Wei Wang, and Jing Xiao. Multi-grained knowledge distillation for named entity recognition. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational linguistics: human language technologies*, pp. 5704–5716, 2021.

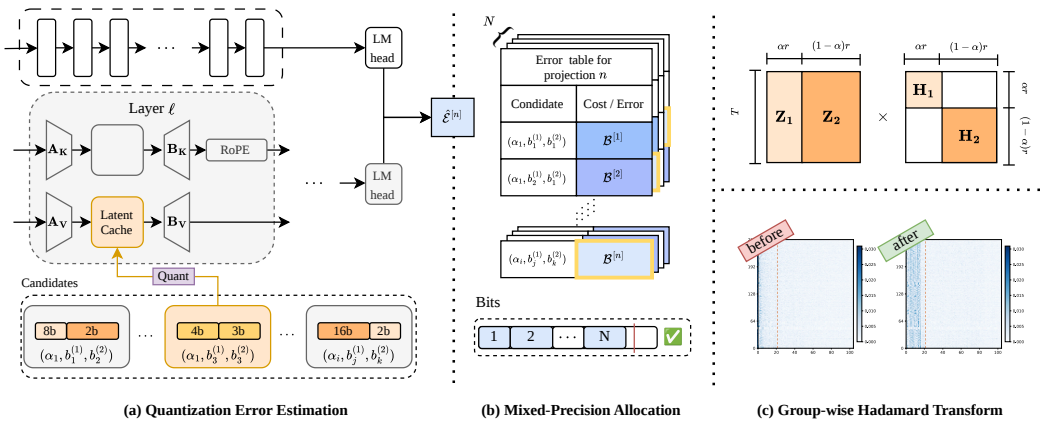


Figure 5: Projection-wise quantization error estimation and allocation. Each candidate specifies a band split and bit widths for the latent KV cache. The resulting error table is used to select candidates under an average-bit budget. Hadamard rotation is applied within each band before quantization.

# A Appendix

## A.1 Quantization

**Quantization error estimation.** Within each decomposition group of projection $n$, we split the latent channels into two channel bands in spectral order and define a quantization candidate as $c = (\alpha, b^{(1)}, b^{(2)}) \in \mathcal{S}_{\mathrm{q}}^{[n]}$. Here, $\alpha$ denotes the fraction of channels in the first band, and $b^{(1)}$ and $b^{(2)}$ specify the precision assigned to each band. Using the compressed model before quantization as the reference, we apply candidate $c$ to the latent cache of one projection at a time and measure the additional output error. As illustrated in Figure 5(a), we compare the output distributions before and after quantization on the same calibration inputs and estimate the KL divergence for each candidate using the top-$k$ probability aggregation in equation 4. We denote the estimated error for candidate $c$ of projection $n$ by $\widehat{\mathcal{E}}_{\mathrm{q}}^{[n]}(c)$. For a joint candidate configuration $\mathbf{c} = (c^{[1]}, \ldots, c^{[N]})$, we approximate the total quantization error by the following additive estimate:

$$\widehat{\mathcal{E}}_{\mathrm{q}}(\mathbf{c}) = \sum_{n=1}^{N} \widehat{\mathcal{E}}_{\mathrm{q}}^{[n]}(c^{[n]}). \tag{6}$$

**Mixed-precision allocation.** Let $r^{[n]}$ denote the total number of latent channels in projection $n$, and $m^{[n]}(\alpha)$ the number of channels assigned to the first band according to the split ratio $\alpha$. We define the per-token storage cost of candidate $c = (\alpha, b^{(1)}, b^{(2)})$ as $\mathcal{B}^{[n]}(c) = b^{(1)} m^{[n]}(\alpha) + b^{(2)}(r^{[n]} - m^{[n]}(\alpha))$, measured in bits. Given a target average precision $\bar{b}$, the mixed-precision allocation in Figure 5(b) is formulated as follows:

$$\min_{\mathbf{c} \in \mathcal{S}_{\mathrm{q}}} \widehat{\mathcal{E}}_{\mathrm{q}}(\mathbf{c}) \quad \text{s.t.} \quad \sum_{n=1}^{N} \mathcal{B}^{[n]}(c^{[n]}) \le \bar{b} \sum_{n=1}^{N} r^{[n]}. \tag{7}$$

This is a multiple-choice knapsack problem (MCKP) Kellerer et al. (2004) that selects one quantization candidate per projection while constraining the average precision across all latent channels. We solve this problem to select candidates that minimize the total estimated quantization error under the given precision budget Huangfu & Hall (2018). Before final quantization, we apply a Hadamard rotation within each channel band Chang et al. (2025); Bhatnagar et al. (2026). We implement Triton kernels for 2-, 3-, 4-, and 8-bit latent KV cache quantization, drawing on the packed-cache implementations of KIVI Liu et al. (2024) and KVQuant Hooper et al. (2024).

**Effect of Quantization on Latency.** Table 6 evaluates the effect of mixed-bit latent KV cache quantization on attention latency. Compared with our low-rank method alone, adding quantization consistently reduces latency across all context lengths.

| Tokens | SDPA | Ours | Ours + Quant. |
|---|---|---|---|
| 4K | 42.44 | 23.72 (1.79×) | **21.58** (1.97×) |
| 8K | 78.46 | 42.20 (1.86×) | **36.26** (2.16×) |
| 16K | 150.59 | 78.67 (1.91×) | **65.88** (2.29×) |
| 32K | 295.22 | 153.11 (1.93×) | **126.36** (2.34×) |
| 64K | 582.50 | 309.39 (1.88×) | **249.91** (2.33×) |

Table 6: **Attention latency and speedup.** We report attention latency (ms) of LLaMA-2-7B on a single RTX 4090 with batch size 16. Parentheses indicate speedups over SDPA. Ours uses 75% low-rank compression; Ours + Quant. additionally applies mixed-bit latent KV cache quantization. Latency includes QKV and output projections and is summed across 32 individually measured layers. Bold indicates the lowest latency at each context length.

### A.2 MMKP Complexity

The $O(NCF)$ term accounts for evaluating candidate errors. Solving the MMKP incurs an additional cost.

**Dynamic programming.** Let $B'_M$ and $B'_F$ denote the memory and FLOPs budgets expressed in integer units, respectively. Dynamic programming requires $O(NCB'_M B'_F)$ time and $O(B'_M B'_F)$ space.

**Branch-and-cut.** With $NC$ binary variables and an upper bound of $T_{\text{node}}$ on the work per node, the worst-case search cost is $O(2^{NC} T_{\text{node}})$. In practice, the cost depends on the number of nodes explored.

### A.3 Decoding Cost

For candidate $(g, r)$, the candidate-dependent FLOPs for decoding one token at batch size 1 are given in equation 5. We count each multiplication and addition as one FLOP. Here, $g$ is the decomposition group size, and $r$ is the sum of the ranks across all groups. We exclude costs that are independent of the candidate, such as the Q and O projections and the QK operation.

**Latent projection.** Both the key and value are projected into a rank-$r$ latent representation for the new token. Each projection costs $2C_{\text{in}}r$ FLOPs.

**Key path.** Reconstructing the $T$ cached latent keys costs $2TgH_{\text{dim}}r$ FLOPs. Applying RoPE to the reconstructed keys costs $3TC_{\text{out}}$ FLOPs.

**Value path.** Computing the weighted sum of the cached latent values costs $2Tgr$ FLOPs. Reconstructing the resulting vector in the output dimension costs $2gH_{\text{dim}}r$ FLOPs.

**Multi-head attention.** Adding the latent projection costs to the key and value path costs yields the MHA Vaswani et al. (2017) decoding cost in equation 5.

**Grouped-query attention.** In GQA Ainslie et al. (2023), multiple query heads share a KV head, but each query head has distinct attention weights. We therefore multiply the latent value aggregation and output reconstruction terms by the ratio of query heads to KV heads. Each key is reconstructed only once per KV head, so this ratio does not apply to the key path.

### A.4 Inference Setup

**Attention latency.** For Table 6, we fix the batch size at 16 and prepare KV caches with context lengths from 4K to 64K tokens. The timed operation includes QKV projections, RoPE, the current token's KV-cache write, attention, and the output projection. We measure each attention layer separately on the GPU and sum the latencies across all 32 layers. Thus, the 64K result does not establish that the full model can run at that context length with all layers resident on the GPU.

**Full-model decoding throughput.** For Table 5, the full model and the KV caches of all layers reside on the GPU simultaneously. We find the maximum feasible batch size for each method and input length, then generate 4,096 tokens autoregressively per sequence. Throughput is the total number of tokens generated across the batch divided by decoding time. The timed operation includes attention, MLPs, the final output layer, argmax, and KV-cache updates. The reported throughput gains therefore include the benefit of larger batches enabled by KV-cache compression.

**Precision at 16K.** At a 16K input length 5, the compressed model produced a non-finite output in the FP16 decoding MLP. We therefore remeasured the 16K throughput of all methods with FP32 MLP and residual computations during decoding. This change does not apply to the attention latency measurements.

### A.5 Training settings

**Data and objective.** For each model and compression rate, we perform recovery training on 10M tokens from the English (eng_Latn) subset of FinePDFs-Edu Kydlíček et al. (2025), using sequences of 2048 tokens. The original model serves as the teacher. We train only the low-rank encoder and decoder factors of the compressed key and value projections. The objective is $\mathcal{L}_{\mathrm{KL}} + \lambda\mathcal{L}_{\mathrm{CE}}$ (Section 3.4), with temperature 1. We set $\lambda = 0.6$ for LLaMA-2 and $\lambda = 0.5$ for the other models.

**Optimization.** We use AdamW with a constant learning rate of $2 \times 10^{-7}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-4}$, and weight decay 0.01. The microbatch size is 1, with four gradient accumulation steps. We clip encoder and decoder gradient norms separately to 1.0. Forward computation uses FP16, while the trainable factors and optimizer states are maintained in FP32. We enable gradient checkpointing.

**Checkpoint selection.** We evaluate the recovery loss approximately every 250K training tokens on a fixed 20K-token validation stream drawn from the same dataset with a different shuffle seed. The training and validation seeds are 42 and 43, respectively. We use the factors with the lowest validation loss for final evaluation.

### A.6 Allocation Setup

**Decomposition candidates.** We use the same group-size candidates for key and value projections: $\{1, 2, 4, 8, 16, 32\}$ for LongChat-7B-v1.5 and LLaMA-2-7B, and $\{1, 2, 4, 8\}$ for LLaMA-3.1-8B-Instruct. Total-rank candidates are generated from 10% to 100% of the original KV output dimension in increments of 10%, then rounded and limited by group capacity.

**Candidate error estimation.** We estimate each decomposition candidate's compression error by compressing one key or value projection while leaving all other projections unchanged. We compare the compressed model's output distribution with that of the original model on four 256-token WikiText-2 samples. For KL computation, we retain the original model's top 64 token probabilities individually and aggregate the remaining probability mass into one tail term.

**Resource budgets.** At each compression rate, we set both the KV cache and decoding FLOPs budgets based on PaLU Chang et al. (2025). We calculate decoding FLOPs for one generated token with a batch size of 1 and a cache length of 4096 tokens. At 60% and 75% compression, the FLOPs ceilings relative to the uncompressed model are $16.5\times$ and $10\times$ for LongChat, and $15\times$ and $10\times$ for LLaMA-2, respectively. For LLaMA-3.1-8B-Instruct at 60% compression, the ceiling is $5\times$.

**MMKP solver.** We use the branch-and-cut MIP solver in HiGHS Huangfu & Hall (2018) to select one candidate for each key and value projection. The solver minimizes the sum of candidate KL estimates subject to the KV cache target and decoding FLOPs ceiling. Before solving, we remove candidates dominated in both estimated KL and FLOPs by another candidate with the same cache cost. We set the relative MIP gap to 0.03 and the time limit to 3600 seconds.

**Quantization candidates.** For the recovered LongChat models, we split the latent channels of each decomposition group into two bands in spectral order. A quantization candidate specifies the

first band's fraction $\alpha \in \{0, 0.1, \ldots, 1\}$ and a bit width for each band, chosen from $\{2, 3, 4, 8, 16\}$. These candidates are allocated separately from the decomposition candidates, under a target average precision of 3.2 bits (Appendix A.1).

### A.7 Decomposition settings

**Calibration statistics.** We collect the input-activation and output-gradient statistics required for K-FAC-based weighted SVD from 64 WikiText-2 samples of 2048 tokens each. We set the K-FAC regularization coefficient to $10^{-4}$.

**Low-rank factor construction.** For each group size and total rank, we allocate rank across groups according to the singular values from the weighted SVD. We maximize the sum of squared retained singular values and construct the low-rank encoder and decoder factors for each key or value projection using the resulting group ranks.

### A.8 Detailed Long-context performance

Tables 7 and 8 provide detailed task-level results on LongBench and RULER, respectively. We evaluate ORG-KV against PaLU and STAR-KV on LongChat-7B-v1.5 and LLaMA-3.1-8B-Instruct. These results provide detailed comparisons across individual tasks and compression methods.

| Model / Method | Comp. (%) | Qasper | QMSum | TriviaQA | MultiQA | TREC | MultiNews | VCSum | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| **LongChat-7B-v1.5** | 0 | 27.70 | 22.70 | 82.30 | 43.00 | 68.50 | 26.10 | 15.40 | 40.81 |
| PaLU | 50 | 20.29 | 22.06 | 75.65 | **41.17** | 63.00 | 23.31 | 12.24 | 36.82 |
| STAR-KV | 60 | 22.58 | 22.40 | 81.30 | 41.04 | **65.00** | 25.70 | **15.60** | **39.09** |
| Ours + KD | 60 | **23.80** | **22.85** | **82.40** | 40.57 | 62.00 | **26.04** | 13.45 | 38.73 |
| **LLaMA-3.1-8B-Instruct** | 0 | 25.19 | 23.20 | 92.00 | 39.90 | 72.50 | 26.90 | 15.91 | 42.23 |
| PaLU | 50 | 19.83 | **23.28** | 86.11 | 30.17 | 64.00 | 21.26 | 2.06 | 35.25 |
| STAR-KV | 60 | **23.20** | 22.40 | 89.06 | **36.34** | 67.00 | 25.89 | 13.20 | **39.58** |
| Ours + KD | 60 | 12.35 | 22.76 | **89.09** | 33.06 | **75.00** | **26.99** | **14.21** | 39.07 |

Table 7: LongBench results with PaLU at 50% and STAR-KV and our method at 60% KV cache compression. Bold indicates the best result among the compressed methods for each model.

| Model / Method | Comp. (%) | MK1 | MK2 | MQ | MV | S1 | S2 | S3 | FWE | SQ | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **LongChat-7B-v1.5** | 0 | 99.80 | 99.60 | 98.40 | 96.55 | 100.00 | 100.00 | 100.00 | 48.20 | 56.72 | 88.80 |
| PaLU | 50 | 99.60 | 98.20 | 64.75 | 96.05 | **100.00** | **100.00** | 98.00 | 45.80 | 54.18 | 84.06 |
| STAR-KV | 60 | 99.40 | 99.60 | **98.50** | **96.55** | **100.00** | **100.00** | **99.80** | 46.00 | **54.45** | 88.26 |
| Ours + KD | 60 | **99.80** | **100.00** | 96.60 | 93.85 | **100.00** | **100.00** | **99.80** | **69.67** | 49.68 | **89.93** |
| **LLaMA-3.1-8B-Instruct** | 0 | 100.00 | 99.80 | 99.90 | 98.95 | 100.00 | 100.00 | 99.60 | 96.07 | 78.12 | 96.94 |
| PaLU | 50 | 93.80 | 98.40 | 89.00 | 86.60 | 99.20 | 94.20 | 80.60 | 88.87 | 55.88 | 87.39 |
| STAR-KV | 60 | 98.40 | 86.00 | 85.10 | 84.00 | **100.00** | **100.00** | 92.00 | 87.73 | 68.05 | 89.03 |
| Ours + KD | 60 | **99.80** | **99.80** | **96.45** | **90.60** | **100.00** | **100.00** | **99.80** | **94.53** | **69.23** | **94.47** |

Table 8: RULER results at a sequence length of 4K, with PaLU at 50% and STAR-KV and our method at 60% KV cache compression. Bold indicates the best result among the compressed methods for each model.

### A.9 Effect of the recovery objective and dataset

We compare recovery objectives and training datasets on LongChat-7B-v1.5 at 60% compression, using 10M tokens per setting. (Table 9) On FineWeb-Edu, adding CE to KL reduces average perplexity from 8.52 to 7.73 and increases average zero-shot accuracy from 60.76% to 61.38%. Keeping KL+CE and switching to FinePDFs-Edu increases average accuracy to 61.61%, while average perplexity increases to 7.78.

| Dataset | Loss | **lm-avg.** ↓ | Acc. (%) ↑ |
|---|---|---|---|
| FineWeb-Edu | KL | 8.52 | 60.76 |
| FineWeb-Edu | KL+CE | **7.73** | 61.38 |
| FinePDFs-Edu | KL+CE | 7.78 | **61.61** |

Table 9: **Effect of the recovery objective and training data.** Following Section 3.4, we train only the low-rank factors of the key and value projections using the original model as the teacher. We compare KL-only and KL+CE objectives on FineWeb-Edu Lozhkov et al. (2024) and KL+CE on FinePDFs-Edu Kydlíček et al. (2025), using LongChat-7B-v1.5 Li et al. (2023) at 60% compression with 10M training tokens per setting. lm-avg. and Acc. denote mean perplexity on WikiText-2 Biderman et al. (2026) and C4 Raffel et al. (2020) and mean accuracy over six zero-shot tasks, respectively.

## AI Assistance Disclosure

We used generative AI tools to identify relevant literature, brainstorm research ideas, and improve the clarity and readability of the manuscript. These tools also assisted with translation and provided feedback on research methodology. The authors reviewed all AI-assisted content and take full responsibility for the final text, claims, and artifacts presented in this work.