TO PREDICT OR NOT TO PREDICT: BUDGET-AWARE KV-CACHE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Key–value (KV) caching avoids repeated computation during autoregressive decoding, but its memory cost grows with context length. Same-layer cross-head prediction can improve the low-rank compressibility of KV representations, but retaining anchor heads consumes cache capacity. Under a fixed KV-cache budget, prediction must therefore improve compressibility enough to justify storing the anchors. We propose TPNP (To Predict or Not to Predict), a KV-cache compression method that jointly determines whether to use cross-head prediction and how much of the prediction residual, termed the innovation, to retain. TPNP considers the native representation, prediction-assisted innovation coding, and prediction-free joint low-rank coding in a unified search space. When prediction is selected, TPNP retains same-layer anchors and jointly represents the target innovations in a low-rank subspace. TPNP ranks innovation basis directions using Fisher–Rao sensitivity and jointly selects anchor–target partitions and retained innovation ranks for each layer-wise K/V component to minimize the estimated Fisher–Rao compression cost under a global KV-cache budget. Our analyses reveal a key–value asymmetry: prediction generally improves compressibility in the value partitions selected by TPNP for cross-head prediction, while TPNP predominantly selects prediction-free joint low-rank coding for keys. We evaluate TPNP across four backbones and multiple calibrations. Across four KV saving ratios from 10% to 70%, comparisons with six methods show that TPNP ranks first or second in 14 of 16 backbone–ratio combinations for average perplexity and in 14 of 16 for average accuracy. On Qwen2.5-7B and Yi-1.5-6B, it also achieves the highest average LongBench score in six of eight combinations, ranking second in the remaining two.

1 INTRODUCTION

Autoregressive inference in large language models (LLMs) commonly relies on key–value (KV) caching to avoid repeated computation during decoding. However, the KV cache grows linearly with context length, increasing both memory usage and memory traffic (Zhang et al., 2023; Liu et al., 2024b). Existing KV-cache compression methods address this cost through quantization, token eviction, or low-rank representations (Zhang et al., 2023; Liu et al., 2024b; Chang et al., 2025). Prediction-based compression provides a complementary approach by exploiting dependencies among cached representations (Shutova et al., 2025; Ji et al., 2026; Shaikh et al., 2026). We consider same-layer cross-head prediction as an intermediate step before low-rank compression, retaining selected KV heads as *anchors* to predict the remaining *target heads* and compressing only their *innovations*, the residuals after prediction. This creates a budget-dependent trade-off: prediction can improve the low-rank compressibility of the target innovations, but the anchors themselves consume cache capacity. Under a fixed KV-cache budget, the relevant question is therefore not simply whether KV heads are predictable, but whether prediction improves compressibility enough to justify storing the anchors.

As described in Section 5, our analyses on Qwen2.5-7B and Yi-1.5-6B show that the answer varies substantially across layer-wise K/V components and KV-cache budgets. For the value partitions selected for cross-head prediction, prediction-assisted innovation coding generally yields lower reconstruction error than storage-matched direct SVD, with the largest median NMSE reductions at a 70% KV saving ratio. Keys exhibit a different pattern. At a 10% KV saving ratio, prediction-assisted innovation coding does not outperform direct SVD in reconstruction error for any selected

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

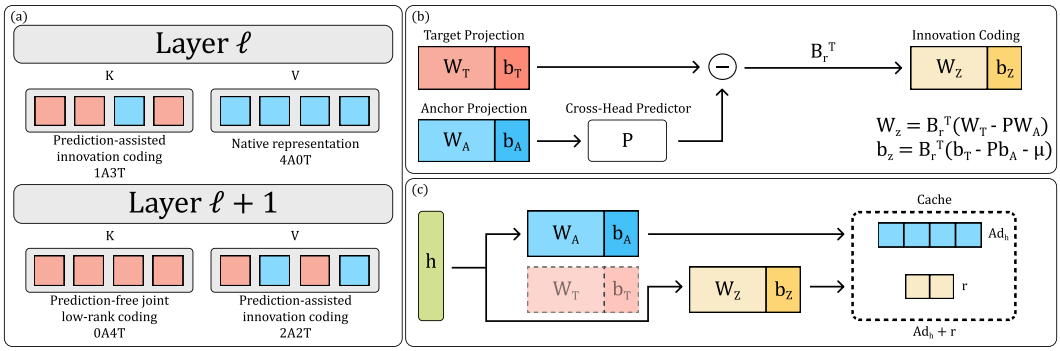


Figure 1: **Colors:** anchors, targets, K/V projection input, innovation coding. **(a)** Example layer-wise K/V configurations selected by TPNP from a unified search space comprising the native representation, prediction-assisted innovation coding, and prediction-free joint low-rank coding; e.g., 1A3T denotes one anchor and three target heads. **(b)** In prediction-assisted innovation coding, anchor heads are retained to predict the target heads, whose innovations are jointly represented in a low-rank subspace. **(c)** The retained innovation coordinates are produced directly from the K/V projection input through a single linear projection and cached together with the anchors, requiring $Ad_h + r$ cached scalars per token. RoPE handling for keys is omitted for clarity.

key partition using cross-head prediction. At KV saving ratios of 30%, 50%, and 70%, the global KV-cache budget allocator selects prediction-free joint low-rank coding for every key component on both backbones.

Motivated by these observations, we propose TPNP (To Predict or Not to Predict), a KV-cache compression method that jointly determines whether to use cross-head prediction and how much innovation to retain. As shown in Figure 1, TPNP considers the *native representation*, *prediction-assisted innovation coding*, and *prediction-free joint low-rank coding* within a unified search space for each layer-wise K/V component. TPNP further ranks the innovation basis directions using Fisher–Rao sensitivity rather than explained variance alone. Given a KV saving ratio, the global KV-cache budget allocator minimizes the estimated Fisher–Rao compression cost by jointly selecting the *anchor–target partition* and retained *innovation rank* for each layer-wise K/V component.

Our contributions are summarized as follows:

(1) We analyze when same-layer cross-head prediction improves KV-cache compressibility under a fixed KV-cache budget. Our analyses reveal distinct patterns across keys and values, while comparisons with prediction-free joint low-rank coding show that the benefit of prediction depends on both the backbone and the KV-cache budget.

(2) We propose TPNP, a KV-cache compression method that considers prediction-assisted innovation coding and prediction-free joint low-rank coding within a unified search space. TPNP ranks innovation basis directions using Fisher–Rao sensitivity and jointly selects anchor–target partitions and retained innovation ranks under a global KV-cache budget.

(3) We evaluate TPNP across four backbones and multiple calibrations. Across four KV saving ratios from 10% to 70%, comparisons with six methods show that TPNP ranks first or second in 14 of 16 backbone–ratio combinations for average perplexity and in 14 of 16 for average accuracy. On the two main backbones, it also achieves the highest average LongBench score in six of eight combinations, ranking second in the remaining two.

2 Related Work

Low-rank representations provide a direct approach to reducing KV-cache memory by exploiting redundancy in the hidden dimension of K/V representations. ASVD (Yuan et al., 2025) incorporates activation statistics into singular value decomposition and extends the resulting factorization to reduce the channel dimension of cached K/V representations, while CARE (Zhou et al., 2026) uses covariance-aware factorization and non-uniform rank allocation to convert pretrained MHA/GQA

modules into multi-head latent attention. Eigen Attention (Saxena et al., 2024) constructs low-rank subspaces from calibration representations and performs attention directly in the compressed space. Palu (Chang et al., 2025) applies grouped low-rank decomposition with Fisher-based rank allocation, whereas ReCalKV (Yan et al., 2025) reorders and groups similar key heads before SVD and uses offline calibration and matrix fusion for value compression. SVD-LLM (Wang et al., 2025), more generally, improves SVD-based model compression through truncation-aware whitening and post-truncation updates. Related low-rank approaches exploit additional structure: MatryoshkaKV (Lin et al., 2025) learns orthogonal projections and adaptively allocates compression rates across layers and heads, STAR-KV (Bhatnagar et al., 2026) learns soft thresholds for adaptive rank control, and xKV (Chang et al., 2026) shares low-rank structure across layers through aligned dominant singular directions.

A complementary line of work focuses on predictability among KV representations. AQUA-KV (Shutova et al., 2025) uses compact predictors to capture dependencies among KV representations and quantizes the residual information that cannot be predicted. EchoKV (Ji et al., 2026) retains partial KV information and reconstructs the remaining cache with lightweight networks that exploit intra- and inter-layer similarities. Linear Predictability (Shaikh et al., 2026) reconstructs K/V activations from a small number of peer heads by retaining reference heads and recovering target heads on demand. TPNP instead treats whether to use same-layer cross-head prediction itself as a compression decision under the global KV-cache budget. When prediction is selected, TPNP uses prediction-assisted innovation coding to jointly represent the remaining target-head innovations in a low-rank subspace, while also considering prediction-free joint low-rank coding within the same search space.

3 METHODOLOGY

3.1 CROSS-HEAD PREDICTION AND INNOVATION CODING

Preliminaries are provided in Appendix A. For layer ℓ and component $c \in \{\mathrm{K}, \mathrm{V}\}$, we partition the KV heads into an anchor set $\mathcal{A}_{\ell,c}$ and a target set $\mathcal{T}_{\ell,c}$:

$$\mathcal{A}_{\ell,c} \dot{\cup} \mathcal{T}_{\ell,c} = \{1, \ldots, H_{\mathrm{kv}}\}, \tag{1}$$

with

$$A = |\mathcal{A}_{\ell,c}|, \qquad T = |\mathcal{T}_{\ell,c}| = H_{\mathrm{kv}} - A. \tag{2}$$

Let $\mathbf{x}_{\ell,c,h} \in \mathbb{R}^{d_h}$ denote the pre-RoPE representation of head h, where $\mathbf{x}_{\ell,\mathrm{K},h} = \mathbf{k}_{\ell,h}$ and $\mathbf{x}_{\ell,\mathrm{V},h} = \mathbf{v}_{\ell,h}$. For each target head $t \in \mathcal{T}_{\ell,c}$, we estimate its predictable component from the anchors as

$$\widehat{\mathbf{x}}_{\ell,c,t} = \sum_{a \in \mathcal{A}_{\ell,c}} \mathbf{P}_{\ell,c,t,a} \mathbf{x}_{\ell,c,a}, \tag{3}$$

where $\mathbf{P}_{\ell,c,t,a} \in \mathbb{R}^{d_h \times d_h}$. When $\mathcal{A}_{\ell,c} = \varnothing$, we define $\widehat{\mathbf{x}}_{\ell,c,t} = \mathbf{0}$. For each target head, the mappings are fitted jointly over all anchors from calibration representations using ridge-regularized least squares (Hoerl & Kennard, 1970). The predictor has no separate intercept; any systematic residual offset is captured by the innovation mean introduced below. For keys, each mapping is constrained to commute with RoPE:

$$\mathcal{R}_p \left(\mathbf{P}_{\ell,\mathrm{K},t,a} \mathbf{x} \right) = \mathbf{P}_{\ell,\mathrm{K},t,a} \mathcal{R}_p(\mathbf{x}). \tag{4}$$

Accordingly, we restrict each key mapping to be block diagonal over RoPE pairs, with each two-dimensional block taking the form

$$\mathbf{C}_m = \begin{bmatrix} a_m & -b_m \\ b_m & a_m \end{bmatrix}, \tag{5}$$

which commutes with the corresponding two-dimensional rotation. Values are not transformed by RoPE, so their mappings are unrestricted dense linear transformations.

We define the innovation of target head t as

$$\mathbf{e}_{\ell,c,t} = \mathbf{x}_{\ell,c,t} - \widehat{\mathbf{x}}_{\ell,c,t}, \tag{6}$$

and jointly represent the innovations of all target heads:

$$\mathbf{e}_{\ell,c,\mathcal{T}} = \mathrm{Concat}\left(\{\mathbf{e}_{\ell,c,t}\}_{t \in \mathcal{T}_{\ell,c}} \right) \in \mathbb{R}^{T d_h}. \tag{7}$$

We similarly use $\mathbf{x}_{\ell,c,\mathcal{T}}$ and $\widehat{\mathbf{x}}_{\ell,c,\mathcal{T}}$ for the concatenated targets and their predictions. Omitting subscripts when clear, let $\boldsymbol{\mu}$ denote the mean innovation over the calibration hidden-state samples and let $\mathbf{E}$ stack the centered innovation samples row-wise. We compute its singular value decomposition (SVD) (Golub & Reinsch, 1970; Eckart & Young, 1936):

$$\mathbf{E} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{B}^\top, \tag{8}$$

where $\mathbf{B} \in \mathbb{R}^{Td_h \times Td_h}$ is an orthonormal innovation basis. As described in Section 3.2, we reorder its directions by Fisher–Rao sensitivity. For an integer rank $0 \le r < Td_h$, let $\mathbf{B}_r \in \mathbb{R}^{Td_h \times r}$ contain the first r directions after this reordering. The innovation coordinates and reconstructed target representation are

$$\mathbf{z} = \mathbf{B}_r^\top (\mathbf{e}_\mathcal{T} - \boldsymbol{\mu}), \qquad \widetilde{\mathbf{x}}_\mathcal{T} = \widehat{\mathbf{x}}_\mathcal{T} + \boldsymbol{\mu} + \mathbf{B}_r\mathbf{z}. \tag{9}$$

We denote an anchor–target configuration with A anchors and T targets by $AA{T}\mathrm{T}$. For each layer-wise K/V component, we consider configurations from $H_{\mathrm{kv}}\mathrm{A0T}$ to $0\mathrm{A}H_{\mathrm{kv}}\mathrm{T}$. For a configuration with $T > 0$ targets, each of the $\binom{H_{\mathrm{kv}}}{T}$ choices of target set, together with the remaining heads as anchors, defines an anchor–target partition. For each partition, we consider every integer innovation rank $r \in \{0, \dots, Td_h - 1\}$, and each partition–rank pair defines a candidate; the full rank $r = Td_h$ yields no cache saving and is therefore dominated by the native option.

The $H_{\mathrm{kv}}\mathrm{A0T}$ configuration corresponds to the native representation. Configurations with $A > 0$ and $T > 0$ use prediction-assisted innovation coding: the anchors are stored directly, while the target innovations are jointly represented in a low-rank subspace. The $0\mathrm{A}H_{\mathrm{kv}}\mathrm{T}$ configuration corresponds to prediction-free joint low-rank coding, which jointly represents all heads of the current K/V component in a low-rank subspace without anchors. In this case, $\widehat{\mathbf{x}}_\mathcal{T} = \mathbf{0}$, so the innovations equal the target representations themselves. We therefore use "innovation basis," "innovation coordinates," and "innovation rank" for both prediction-assisted innovation coding and prediction-free joint low-rank coding.

Let $\mathbf{P}_{\ell,c}$ denote the block linear map induced by Eq. (3), and let $\mathbf{W}_\mathcal{A}, \mathbf{W}_\mathcal{T}$ and $\mathbf{b}_\mathcal{A}, \mathbf{b}_\mathcal{T}$ denote the projection parameters concatenated in the corresponding anchor and target head order:

$$\mathbf{x}_\mathcal{A} = \mathbf{W}_\mathcal{A}\mathbf{h} + \mathbf{b}_\mathcal{A}, \qquad \mathbf{x}_\mathcal{T} = \mathbf{W}_\mathcal{T}\mathbf{h} + \mathbf{b}_\mathcal{T}, \tag{10}$$

where $\mathbf{h}$ is the input to the K/V projections at layer ℓ. Substituting these projections into Eq. (9) yields

$$\mathbf{z} = \mathbf{W}_z\mathbf{h} + \mathbf{b}_z, \tag{11}$$

with

$$\mathbf{W}_z = \mathbf{B}_r^\top (\mathbf{W}_\mathcal{T} - \mathbf{P}_{\ell,c}\mathbf{W}_\mathcal{A}), \qquad \mathbf{b}_z = \mathbf{B}_r^\top (\mathbf{b}_\mathcal{T} - \mathbf{P}_{\ell,c}\mathbf{b}_\mathcal{A} - \boldsymbol{\mu}). \tag{12}$$

Hence, the retained innovation coordinates are produced directly from the K/V projection input through a single linear projection, without materializing the full target representations. At inference time, a compressed component stores per token only the selected anchors and innovation coordinates. Key anchors are cached after RoPE, whereas innovation coordinates remain in the pre-RoPE latent space. Because the key mappings commute with RoPE, target keys can be predicted directly from the cached post-RoPE anchors; the innovation term is decoded and rotated when reconstructing the target keys before attention. Values require no positional rotation and are reconstructed directly from their cached anchors and innovation coordinates.

3.2 Fisher–Rao Basis Ranking

The SVD basis in Eq. (8) is initially ordered by explained variance, which need not reflect the importance of individual basis directions to the model objective. We therefore reorder the basis directions using Fisher–Rao sensitivity (Fisher, 1922). For an anchor–target partition $(\ell, c, \mathcal{A}, \mathcal{T})$, let $D = |\mathcal{T}|d_h$. For calibration sequence n, let $\mathcal{L}_n$ denote the original model's causal language-modeling loss summed over the sequence. At token position p, we define

$$\mathbf{z}^{(n,p)} = \mathbf{B}^\top \left(\mathbf{e}^{(n,p)} - \boldsymbol{\mu}\right), \qquad \mathbf{g}^{(n,p)} = \frac{\partial \mathcal{L}_n}{\partial \mathbf{x}_\mathcal{T}^{(n,p)}}, \qquad \mathbf{u}^{(n,p)} = \mathbf{B}^\top \mathbf{g}^{(n,p)}. \tag{13}$$

For keys, $\mathbf{x}_\mathcal{T}^{(n,p)}$ denotes the pre-RoPE target representation. Interpreting α_j as a multiplicative gate on basis coordinate j, shared across token positions, we aggregate its gradient contribution over a

sequence of length S_{seq} as

$$s_{n,j} = \frac{1}{\sqrt{S_{\text{seq}}}} \sum_{p=1}^{S_{\text{seq}}} z_j^{(n,p)} u_j^{(n,p)}, \qquad F_j = \frac{1}{N} \sum_{n=1}^{N} s_{n,j}^2, \tag{14}$$

where N is the number of calibration sequences. Under a diagonal empirical-Fisher approximation, the cost of removing a coordinate is estimated as one half of its sensitivity. We sort coordinates by decreasing Fisher–Rao sensitivity and define the estimated cost of truncating to rank r as

$$F_{\pi(1)} \geq F_{\pi(2)} \geq \cdots \geq F_{\pi(D)}, \qquad C(r) = \frac{1}{2} \sum_{j=r+1}^{D} F_{\pi(j)}. \tag{15}$$

The same permutation π is applied to the columns of $\mathbf{B}$. This procedure is applied independently to every anchor–target partition of each layer-wise K/V component.

3.3 Global KV-Cache Budget Allocation

A candidate consists of an anchor–target partition with A anchors and T targets together with an innovation rank r. It stores A full anchor heads and r innovation coordinates. Its per-token storage and saving are

$$M_{\ell,c}(A, r) = Ad_h + r, \qquad S_{\ell,c}(A, r) = Td_h - r. \tag{16}$$

The native configuration has zero saving and zero compression cost. For a target KV saving ratio ρ, the native per-token cache size and required saving are

$$M_{\text{native}} = \sum_{\ell=1}^{L} \sum_{c \in \{\text{K,V}\}} H_{\text{kv}}^{(\ell)} d_h^{(\ell)}, \qquad S_{\text{req}} = M_{\text{native}} - \lfloor (1-\rho) M_{\text{native}} \rfloor. \tag{17}$$

Let $\mathcal{O}_{\ell,c}$ contain the native option together with all candidates formed by pairing each anchor–target partition with an admissible innovation rank defined in Section 3.1. Each option $o \in \mathcal{O}_{\ell,c}$ has saving $S(o)$ from Eq. (16) and Fisher–Rao cost $C(o)$ from Eq. (15). Under the diagonal empirical-Fisher approximation, we treat the estimated compression costs of different layer-wise K/V components as additive. We select one option per layer-wise K/V component by solving

$$\min_{\{o_{\ell,c}\}} \sum_{\ell=1}^{L} \sum_{c \in \{\text{K,V}\}} C(o_{\ell,c}) \quad \text{s.t.} \quad \sum_{\ell=1}^{L} \sum_{c \in \{\text{K,V}\}} S(o_{\ell,c}) \geq S_{\text{req}}, \quad o_{\ell,c} \in \mathcal{O}_{\ell,c}. \tag{18}$$

Eq. (18) is a multiple-choice knapsack problem over layer-wise K/V components. We solve this discrete optimization exactly by dynamic programming over cache-saving units, retaining the minimum-cost state for each accumulated saving and backtracking from the minimum-cost state satisfying S_{req}. Because cache storage is discrete, the resulting allocation is guaranteed to achieve at least the target KV saving ratio ρ and may slightly exceed it.

4 Experiments

4.1 Experimental Setup

In the main experiments, we evaluate TPNP on two backbones, Qwen2.5-7B (Yang et al., 2025) and Yi-1.5-6B (01.AI et al., 2025). Following prior work (Ashkboos et al., 2024; Xiang & Zhang, 2025), we use 128 calibration sequences of 2,048 tokens sampled from the WikiText-2 training split (Merity et al., 2017). Calibration robustness evaluation is provided in Appendix E.

Using the LM Evaluation Harness (Gao et al., 2024), we evaluate on WikiText-2 (Merity et al., 2017), Penn Treebank (Marcus et al., 1993), ARC-Easy (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), OpenBookQA (Mihaylov et al., 2018), RACE (Lai et al., 2017), and WinoGrande (Sakaguchi et al., 2020). Using LongBench (Bai et al., 2024), we evaluate on QMSum (Zhong et al., 2021), MultiNews (Fabbri et al., 2019), TriviaQA (Joshi et al., 2017), SAMSum (Gliwa et al., 2019), LCC (Guo et al., 2023), and RepoBench-P (Liu et al., 2024a).

Method	ρ	Perplexity			Accuracy						
		WikiText-2	Penn Treebank	Average ↓	ARC	OBQA	HellaSwag	PIQA	RACE	WinoGrande	Average ↑
Qwen2.5-7B											
Original	0.0	6.80	11.31	9.06	77.44	34.00	78.87	78.67	42.11	72.61	63.95
ASVD	0.1	6.82	11.36	9.09	**78.28**	33.00	78.61	79.00	41.53	**73.32**	63.96
CARE	0.1	7.07	11.72	9.40	77.10	34.40	77.86	78.02	40.67	72.69	63.46
Eigen Attention	0.1	6.92	11.56	9.24	76.77	31.80	**79.01**	**79.27**	40.29	71.03	63.03
Palu	0.1	6.86	11.39	9.13	78.16	34.00	78.69	79.05	41.91	71.82	63.94
ReCalKV	0.1	6.86	11.39	9.13	78.11	34.40	78.74	78.84	41.72	72.38	64.03
SVD-LLM	0.1	7.01	11.62	9.32	72.94	**35.00**	78.53	78.89	39.90	72.61	62.98
TPNP	0.1	**6.81**	**11.33**	**9.07**	78.20	33.40	78.83	78.84	**42.11**	72.93	**64.05**
ASVD	0.3	7.04	11.72	9.38	78.20	**34.00**	77.82	**78.84**	40.67	71.35	63.48
CARE	0.3	8.16	15.22	11.69	60.77	27.40	71.07	76.17	38.76	69.69	57.31
Eigen Attention	0.3	7.52	13.57	10.55	76.39	30.40	73.18	77.48	38.18	66.14	60.30
Palu	0.3	7.18	12.03	9.61	80.18	**34.00**	77.83	78.29	40.67	71.43	63.73
ReCalKV	0.3	7.18	12.04	9.61	**80.47**	33.60	77.96	78.07	40.48	71.59	63.70
SVD-LLM	0.3	7.41	12.49	9.95	73.82	31.00	76.04	77.80	39.81	71.43	61.65
TPNP	0.3	**6.86**	**11.42**	**9.14**	78.83	33.60	**78.94**	78.62	**40.77**	**72.30**	**63.84**
ASVD	0.5	8.03	13.53	10.78	72.81	**32.80**	75.24	78.35	38.37	68.67	61.04
CARE	0.5	10.24	22.01	16.13	46.84	22.80	55.56	68.82	34.93	62.98	48.66
Eigen Attention	0.5	9.29	22.16	15.73	59.34	26.00	62.80	72.80	34.64	61.09	52.78
Palu	0.5	7.86	13.46	10.66	76.05	31.60	74.90	76.55	40.00	70.17	61.55
ReCalKV	0.5	7.83	13.40	10.62	76.81	32.00	75.33	76.93	**40.10**	**70.32**	61.92
SVD-LLM	0.5	8.21	14.97	11.59	63.34	27.80	68.61	73.88	38.18	69.30	56.85
TPNP	0.5	**7.14**	**12.27**	**9.71**	**80.85**	32.00	**77.38**	**79.05**	40.00	69.93	**63.20**
ASVD	0.7	17.19	32.35	24.77	**68.90**	**31.20**	63.36	**75.52**	33.49	58.41	55.15
CARE	0.7	15.27	42.42	28.85	38.38	15.00	39.67	62.24	30.62	55.49	40.23
Eigen Attention	0.7	–	–	–	–	–	–	–	–	–	–
Palu	0.7	10.16	22.84	16.50	59.30	25.80	63.56	71.38	36.27	62.19	53.08
ReCalKV	0.7	9.85	22.21	16.03	59.85	24.60	64.38	71.55	**36.46**	**64.80**	53.61
SVD-LLM	0.7	15.28	45.54	30.41	38.30	16.80	37.82	58.49	32.44	55.49	39.89
TPNP	0.7	**9.45**	**19.89**	**14.67**	66.92	29.20	**66.77**	74.86	35.31	60.54	**55.60**
Yi-1.5-6B											
Original	0.0	5.78	17.71	11.75	73.78	31.40	75.47	78.51	38.47	72.61	61.71
ASVD	0.1	5.79	17.85	11.82	**75.97**	32.40	75.06	**79.00**	37.80	71.51	**61.96**
CARE	0.1	5.98	18.36	12.17	72.10	31.40	75.68	78.40	38.09	72.38	61.34
Eigen Attention	0.1	5.95	18.74	12.35	73.36	32.00	74.16	78.29	38.37	70.72	61.15
Palu	0.1	5.93	17.98	11.96	72.31	32.60	75.66	78.62	38.18	72.45	61.64
ReCalKV	0.1	5.94	17.98	11.96	72.47	**32.80**	75.78	78.40	**38.76**	**72.69**	61.82
SVD-LLM	0.1	6.32	19.10	12.71	67.42	31.20	**76.12**	78.13	**38.76**	71.43	60.51
TPNP	0.1	**5.78**	**17.65**	**11.72**	75.13	31.60	75.28	78.29	38.56	72.61	61.91
ASVD	0.3	6.08	19.41	12.75	71.63	**32.20**	73.25	78.18	**39.23**	68.35	60.47
CARE	0.3	7.22	25.87	16.55	57.24	26.00	69.60	72.63	35.31	67.40	54.70
Eigen Attention	0.3	7.01	29.36	18.19	62.12	29.80	54.53	74.65	34.26	62.43	52.97
Palu	0.3	6.35	19.84	13.10	67.89	32.00	76.22	78.02	37.99	71.59	60.62
ReCalKV	0.3	6.37	19.96	13.17	67.72	32.00	**76.27**	77.97	38.66	**71.90**	60.75
SVD-LLM	0.3	6.57	20.79	13.68	66.33	31.20	74.82	76.71	38.66	68.67	59.40
TPNP	0.3	**5.84**	**18.32**	**12.08**	**75.59**	31.80	75.12	**78.29**	38.66	71.59	**61.84**
ASVD	0.5	7.43	36.99	22.21	**66.16**	27.20	69.06	74.81	36.17	65.43	56.47
CARE	0.5	9.12	47.98	28.55	49.79	27.40	55.93	69.86	34.83	59.91	49.79
Eigen Attention	0.5	9.25	58.16	33.71	44.78	22.20	48.83	68.88	30.62	58.96	45.71
Palu	0.5	6.94	23.31	15.13	57.79	27.60	72.35	76.12	36.75	68.59	56.53
ReCalKV	0.5	6.97	23.64	15.31	56.57	25.40	72.25	75.46	35.69	68.67	55.67
SVD-LLM	0.5	7.08	25.03	16.06	57.49	27.00	68.42	73.72	**38.37**	65.59	55.10
TPNP	0.5	**6.11**	**22.38**	**14.25**	64.94	**30.80**	**73.80**	**77.04**	**38.37**	**69.14**	**59.02**
ASVD	0.7	20.94	158.19	89.57	47.35	17.00	48.23	66.70	29.67	51.70	43.44
CARE	0.7	12.77	107.84	60.31	38.85	16.20	43.33	62.19	29.19	54.14	40.65
Eigen Attention	0.7	–	–	–	–	–	–	–	–	–	–
Palu	0.7	9.15	**43.19**	**26.17**	42.17	22.00	60.31	69.04	34.35	59.27	47.86
ReCalKV	0.7	9.42	43.40	26.41	42.09	19.20	59.86	68.34	33.49	59.98	47.16
SVD-LLM	0.7	9.63	55.06	32.35	40.61	17.80	46.67	62.79	33.11	56.67	42.94
TPNP	0.7	**7.82**	45.42	26.62	**52.15**	**25.40**	**65.74**	**72.96**	**35.89**	**62.67**	**52.47**

Table 1: Perplexity and accuracy comparison on Qwen2.5 and Yi-1.5. Bold and underlined values indicate the best and second-best results among compression methods for each backbone and ρ. TPNP rows are shaded in gray for visual clarity.

We compare TPNP against six representative low-rank methods for KV-cache compression: ASVD (Yuan et al., 2025), CARE (Zhou et al., 2026), Eigen Attention (Saxena et al., 2024), Palu (Chang et al., 2025), ReCalKV (Yan et al., 2025), and SVD-LLM (Wang et al., 2025). Experimental details are provided in Appendix B. Efficiency analysis is provided in Appendix G.

4.2 Results

Table 1 presents the results on perplexity and accuracy. Compared with the six methods, TPNP achieves the lowest average perplexity in seven of the eight backbone–ratio combinations and the highest average accuracy in seven of the eight. On Qwen2.5-7B, TPNP consistently achieves the lowest average perplexity and highest average accuracy across all KV saving ratios. On Yi-1.5-6B, TPNP obtains the lowest average perplexity at KV saving ratios up to 50% and the highest average accuracy at 30%, 50%, and 70%.

At a 50% KV saving ratio, TPNP achieves an average perplexity of 9.71 and an average accuracy of 63.20 on Qwen2.5-7B, compared with 10.62 and 61.92, respectively, for ReCalKV; on Yi-1.5-6B, TPNP achieves 14.25 and 59.02, compared with 15.13 and 56.53, respectively, for Palu. Additional backbone evaluation is provided in Appendix F.

Method	ρ	QMSum	MultiNews	TriviaQA	SAMSum	LCC	RepoBench-P	Average ↑
				Qwen2.5-7B				
Original	0.0	24.54	27.46	89.54	46.06	68.56	58.00	52.36
ASVD	0.1	23.58	26.69	**90.34**	**46.54**	**69.39**	**58.55**	**52.52**
CARE		**24.40**	27.12	89.64	45.18	60.75	52.63	49.95
Eigen Attention		22.56	26.26	87.84	44.00	62.67	53.14	49.41
Palu		24.09	26.66	89.73	46.07	68.33	57.03	51.99
ReCalKV		23.83	26.73	89.63	45.88	68.35	55.96	51.73
SVD-LLM		21.83	**27.20**	89.98	46.23	57.64	50.65	48.92
TPNP		24.06	27.00	90.04	45.81	68.84	56.79	52.09
ASVD	0.3	22.36	26.61	89.89	44.19	64.80	55.00	50.48
CARE		**24.43**	24.57	84.10	42.73	32.12	40.82	41.46
Eigen Attention		20.70	23.74	71.45	40.26	23.76	31.57	35.25
Palu		23.96	**26.91**	**89.97**	**45.97**	62.64	55.52	50.83
ReCalKV		24.35	26.88	89.68	45.47	62.93	55.15	50.74
SVD-LLM		23.97	26.05	88.03	45.08	21.97	34.62	39.95
TPNP		23.23	26.64	89.81	45.31	**67.45**	**56.50**	**51.49**
ASVD	0.5	21.24	24.83	78.84	41.37	53.46	44.16	44.02
CARE		19.76	17.35	52.04	35.44	10.55	28.91	28.84
Eigen Attention		19.97	15.23	28.60	27.25	25.00	27.81	23.98
Palu		**23.43**	25.90	87.86	43.37	51.68	45.64	46.31
ReCalKV		22.73	**26.08**	88.48	44.14	55.66	47.74	47.47
SVD-LLM		22.08	23.97	71.20	39.58	15.98	24.09	32.82
TPNP		21.86	26.02	**90.74**	**44.47**	**62.60**	**58.67**	**50.73**
ASVD	0.7	10.47	12.48	15.45	4.76	**35.00**	**35.69**	18.98
CARE		14.15	4.82	32.05	22.47	14.44	17.39	17.55
Eigen Attention		–	–	–	–	–	–	–
Palu		19.84	20.46	64.91	39.42	23.82	27.23	32.61
ReCalKV		**20.14**	21.59	**73.14**	40.02	22.14	24.62	**33.61**
SVD-LLM		12.67	8.64	34.33	24.00	11.76	12.91	17.39
TPNP		17.66	**21.97**	71.14	**40.06**	21.83	25.61	33.05
				Yi-1.5-6B				
Original	0.0	19.64	23.62	86.16	16.83	69.54	62.12	46.32
ASVD	0.1	18.97	**26.38**	86.46	13.02	68.91	61.09	45.81
CARE		19.43	22.03	83.94	15.92	67.78	58.83	44.66
Eigen Attention		18.57	17.71	**87.14**	18.50	62.36	55.63	43.32
Palu		18.49	21.99	83.91	**20.29**	**70.23**	58.67	45.60
ReCalKV		18.39	21.50	83.81	19.77	69.96	58.72	45.36
SVD-LLM		18.35	20.13	83.05	18.26	65.96	56.98	43.79
TPNP		**19.81**	22.80	85.81	17.02	69.63	**62.51**	**46.26**
ASVD	0.3	17.06	**24.35**	**86.46**	7.84	**68.16**	58.33	43.70
CARE		17.09	10.98	80.26	10.25	18.93	24.46	27.00
Eigen Attention		10.34	8.77	46.83	3.12	29.04	24.54	20.44
Palu		19.32	23.66	84.69	**22.42**	59.15	51.12	43.39
ReCalKV		19.20	23.21	84.71	22.15	59.32	48.97	42.93
SVD-LLM		17.46	15.63	85.47	17.05	62.10	49.54	41.21
TPNP		**19.48**	21.23	84.22	10.98	66.75	**60.94**	**43.93**
ASVD	0.5	16.44	16.24	78.54	4.06	**48.94**	43.27	34.58
CARE		12.88	3.30	41.28	3.54	3.00	2.70	11.12
Eigen Attention		9.19	5.64	8.70	8.87	9.55	11.67	8.94
Palu		17.33	**22.58**	83.07	12.94	18.68	35.68	31.71
ReCalKV		15.04	21.17	83.59	12.01	15.12	33.36	30.05
SVD-LLM		13.40	2.18	75.95	8.34	46.07	44.28	31.70
TPNP		**19.00**	14.52	**88.18**	**13.44**	45.27	**56.40**	**39.47**
ASVD	0.7	7.29	7.39	2.29	1.28	25.01	25.91	11.53
CARE		**9.39**	0.71	1.53	2.49	1.75	0.94	2.80
Eigen Attention		–	–	–	–	–	–	–
Palu		7.83	0.36	34.20	10.83	31.49	29.55	19.04
ReCalKV		7.58	0.10	**38.85**	6.31	21.53	21.58	15.99
SVD-LLM		6.49	0.09	16.47	7.79	18.60	9.43	9.81
TPNP		9.37	**9.97**	34.13	**17.56**	**35.86**	**32.06**	**23.16**

Table 2: LongBench comparison on Qwen2.5 and Yi-1.5.

On LongBench, TPNP achieves the highest average score in six of the eight backbone–ratio combinations and ranks second in the remaining two. At a 50% KV saving ratio, it achieves average scores of 50.73 and 39.47 on Qwen2.5-7B and Yi-1.5-6B, compared with the next-best results of 47.47 and 34.58. At a 70% KV saving ratio, TPNP ranks second on Qwen2.5-7B and first on Yi-1.5-6B. RULER evaluation is provided in Appendix D.

4.3 Ablation Study

Variant	Anchor–Target Partition Selection	Cross-Head Prediction	Fisher–Rao Basis Ranking	Global KV-Cache Budget Allocation	WikiText-2	Penn Treebank	Average ↓
TPNP_1	×	✓	✓	✓	84.65 / 10.75	464.32 / 89.85	274.49 / 50.30
TPNP_2	✓	×	✓	✓	11.81 / 35.43	40.54 / 259.24	26.18 / 147.34
TPNP_3	✓	✓	×	✓	9.45 / 8.07	21.59 / 45.41	15.52 / 26.74
TPNP_4	✓	✓	✓	×	22.73 / 10.84	75.99 / 87.95	49.36 / 49.40
TPNP	✓	✓	✓	✓	9.45 / 7.82	19.89 / 45.42	14.67 / 26.62

Table 3: TPNP ablation at a 70% KV saving ratio. Perplexities are reported as Qwen2.5 / Yi-1.5.

We conduct an ablation study on Qwen2.5-7B and Yi-1.5-6B at a fixed 70% KV saving ratio. Table 3 presents four variants, each removing one component of TPNP. TPNP_1 removes adaptive anchor–target partition selection by fixing every layer-wise K/V component to the $1\text{A}3\text{T}$ configuration with the first KV head assigned as the anchor. TPNP_2 disables cross-head prediction while keeping the candidates selected by TPNP fixed, and recomputes the innovation bases and Fisher–Rao basis ranking accordingly. TPNP_3 removes Fisher–Rao basis ranking and retains the original SVD basis order. It

reruns the global KV-cache budget allocation to select one option per layer-wise K/V component, using Fisher–Rao compression costs computed from the basis directions discarded under this order. TPNP_4 replaces global KV-cache budget allocation with uniform allocation of the required per-token cache saving across layer-wise K/V components, with integer rounding so that local saving amounts differ by at most one cached scalar per token. Each component independently selects the candidate with the lowest estimated Fisher–Rao compression cost among those that exactly satisfy its assigned saving.

TPNP achieves the lowest average perplexity among the variants on both Qwen2.5-7B and Yi-1.5-6B, with 14.67 and 26.62, respectively. The largest degradation among these ablations differs across backbones: fixing the anchor–target partition selection causes the largest degradation on Qwen2.5-7B, increasing the average perplexity to 274.49, whereas disabling cross-head prediction with TPNP-selected candidates held fixed is most detrimental on Yi-1.5-6B, increasing it to 147.34. Global KV-cache budget allocation is important on both backbones, as uniform allocation raises the average perplexity to 49.36 and 49.40, respectively. Fisher–Rao basis ranking yields a smaller improvement in average perplexity, from 15.52 to 14.67 on Qwen2.5-7B and from 26.74 to 26.62 on Yi-1.5-6B. Comparison with prediction-free joint low-rank coding is provided in Appendix C.

5 Analysis

5.1 When Does Cross-Head Prediction Improve Compressibility?

We examine whether prediction-assisted innovation coding yields lower reconstruction error than direct SVD coding of target representations under the same storage budget. For each TPNP-selected key or value partition that uses cross-head prediction, we keep its anchor and target sets, fitted predictor, calibration innovation mean, and fitted SVD bases fixed. We evaluate NMSE reduction and Wins at the innovation rank selected by TPNP and vary the retained rank for the AUC comparison. To isolate the effect of prediction from Fisher–Rao basis ranking, we compare two storage-matched variants: direct SVD coding of the centered target representations and prediction-assisted innovation coding of the centered innovations in Eq. (7). Both variants retain basis directions in descending order of singular value, without Fisher–Rao reordering. Both retain the same anchors and use the same rank r, while the direct SVD method reconstructs the targets without using the anchors. Thus, the two variants have identical per-token KV-cache storage. The bases are fitted using representations from the WikiText-2 training split, and the two codings are evaluated without refitting on 4,096 held-out hidden-state samples per layer from the test split.

Let $\mathbf{X}_{\mathcal{T}}$ and $\widehat{\mathbf{X}}_{\mathcal{T}}$ denote the held-out target representations and their cross-head predictions, respectively, and let $\boldsymbol{\mu}_{\mathcal{T}}^{\text{te}}$ denote the test-split target mean. We measure cross-head predictability as

$$R^2 = 1 - \frac{\left\|\mathbf{X}_{\mathcal{T}} - \widehat{\mathbf{X}}_{\mathcal{T}}\right\|_F^2}{\left\|\mathbf{X}_{\mathcal{T}} - \boldsymbol{\mu}_{\mathcal{T}}^{\text{te}}\right\|_F^2}. \tag{19}$$

Here, $\widehat{\mathbf{X}}_{\mathcal{T}}$ contains only the anchor-conditioned prediction in Eq. (3); the innovation mean is used only for reconstruction.

Both variants center their representations using the corresponding training-split means. Let $\widetilde{\mathbf{X}}_{\mathcal{T}}$ denote the corresponding reconstructed target representations. We measure normalized mean squared error (NMSE) as

$$\text{NMSE} = \frac{\left\|\mathbf{X}_{\mathcal{T}} - \widetilde{\mathbf{X}}_{\mathcal{T}}\right\|_F^2}{\left\|\mathbf{X}_{\mathcal{T}} - \boldsymbol{\mu}_{\mathcal{T}}^{\text{te}}\right\|_F^2}. \tag{20}$$

At the selected innovation rank, we report NMSE reduction (%) as $100\,(1 - \text{NMSE}_{\text{pred}} \,/\, \text{NMSE}_{\text{direct}})$. We also report the AUC ratio, defined as the area under the NMSE–rank curve for prediction-assisted innovation coding divided by that under the direct SVD curve, with the rank axis normalized to $[0, 1]$; a ratio below 1 favors prediction-assisted innovation coding. For each KV saving ratio, R^2, NMSE reduction, and AUC ratio are reported as medians across the selected partitions, while Wins counts the partitions for which prediction-assisted innovation coding achieves lower NMSE at the selected innovation rank.

ρ	R^2	NMSE Reduction	Wins	AUC Ratio
		Qwen2.5-7B		
0.10	0.56	7.71	5/5	0.42
0.30	0.16	2.10	7/9	0.96
0.50	0.22	6.81	18/18	0.90
0.70	0.24	10.93	22/22	0.87
All	0.22	7.64	52/54	0.89
		Yi-1.5-6B		
0.10	0.13	3.48	4/4	0.94
0.30	0.31	3.74	17/22	0.90
0.50	0.20	4.63	28/29	0.92
0.70	0.21	7.05	27/27	0.90
All	0.22	5.34	76/82	0.91

Table 4: Analysis of TPNP-selected value partitions using cross-head prediction.

Table 4 shows that the selected value partitions exhibit exploitable cross-head predictability. Across all selected value partitions, the median R^2 is 0.22 on both backbones. At the selected innovation rank, prediction-assisted innovation coding outperforms direct SVD in 52 of 54 partitions on Qwen2.5-7B and 76 of 82 on Yi-1.5-6B, with median NMSE reductions of 7.64% and 5.34%, respectively.

At the selected innovation rank, the advantage is largest at a 70% KV saving ratio. Prediction-assisted innovation coding outperforms direct SVD in every selected value partition on both backbones, with median NMSE reductions of 10.93% and 7.05%.

Keys exhibit a different pattern. At a 10% KV saving ratio, the selected anchor–target partitions with cross-head prediction have negative median R^2 values of -0.59 and -0.64 on Qwen2.5-7B and Yi-1.5-6B, respectively, and prediction-assisted innovation coding does not outperform direct SVD in any of the 7 Qwen2.5-7B or 3 Yi-1.5-6B partitions. At KV saving ratios of 30%, 50%, and 70%, the global KV-cache budget allocator instead selects prediction-free joint low-rank coding, corresponding to the $0\mathrm{A}H_{\mathrm{kv}}\mathrm{T}$ configuration, for every key component on both backbones. Thus, cross-head prediction is useful for the selected value partitions but should be applied selectively rather than uniformly across keys and values.

5.2 What Does the Global KV-Cache Budget Allocator Select?

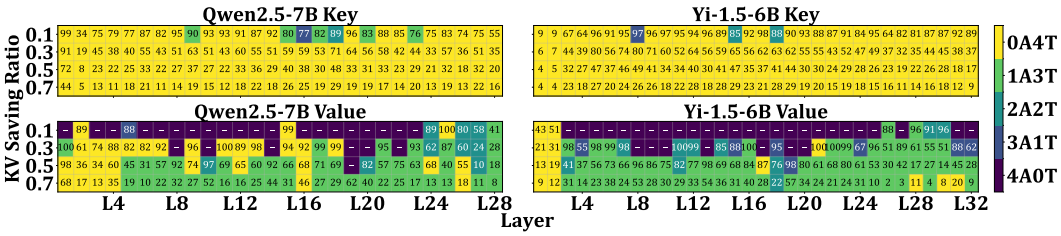


Figure 2: Layer-wise configurations and retained innovation ranks selected by TPNP. Colors indicate anchor–target configurations; cell values show retained ranks as percentages of the corresponding target dimensions. Dashes denote native configurations.

Figure 2 shows a consistent key–value asymmetry that mirrors the reconstruction analysis above. For keys, the allocator predominantly favors prediction-free joint low-rank coding. Values remain substantially more heterogeneous across layers, with the allocator selecting different anchor–target configurations under the global KV-cache budget.

The retained innovation ranks, expressed relative to their corresponding target dimensions, also vary with the global KV-cache budget and generally decrease as the KV saving ratio increases.

6 Conclusion

We presented TPNP, a KV-cache compression method that selects anchor–target partitions and retained innovation ranks under a global KV-cache budget. TPNP unifies prediction-assisted innovation coding and prediction-free joint low-rank coding, using Fisher–Rao sensitivity to guide basis ranking and budget allocation. Our analyses show that prediction generally improves compressibility in value partitions selected for cross-head prediction, while the allocator predominantly favors prediction-free joint low-rank coding for keys. Discussion is provided in Appendix H.

AI USE STATEMENT

In this work, we used generative AI tools to help formulate and refine mathematical claims in the methodology, provide key elements for proving mathematical claims, write mathematical proofs, provide feedback on experimental design, implement the proposed method and associated experimental code, and translate text. We did not use generative AI tools to originate the core research idea, develop theoretical models or conceptual frameworks, propose or refine research hypotheses, or interpret experimental results. We did not use generative AI tools for any other tasks for which disclosure is required in this work. Additionally, we used generative AI tools to assist with editing the manuscript for readability, summarizing and identifying relevant literature, and searching for relevant information. We reviewed all AI-assisted work. AI-assisted code was tested for correctness, mathematical formulations and proofs were verified by the authors, and AI-assisted text was manually reviewed. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

01.AI, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yanpeng Li, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. Yi: Open foundation models by 01.ai, 2025. URL https://arxiv.org/abs/2403.04652.

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/.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=dfqsW38v1X.

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.

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. doi: 10.1609/aaai.v34i05.6239. URL https://doi.org/10.1609/aaai.v34i05.6239.

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.

Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Hung-Yueh Chiang, Yash Akhauri, Xilai Dai, Huiqiang Jiang, Yucheng Li, Luis Ceze, Kai-Chiang Wu, and Mohamed S. Abdelfattah. xKV:

Cross-layer KV-cache compression via aligned singular vector extraction. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=XDb69lykA8.

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.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. doi: 10.1007/BF02288367.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1074–1084, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1102. URL https://aclanthology.org/P19-1102/.

R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London, Series A: Containing Papers of a Mathematical or Physical Character*, 222(594-604):309–368, 1922. doi: 10.1098/rsta.1922.0009.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In Lu Wang, Jackie Chi Kit Cheung, Giuseppe Carenini, and Fei Liu (eds.), *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5409. URL https://aclanthology.org/D19-5409/.

G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numer. Math.*, 14(5):403–420, April 1970. ISSN 0029-599X. doi: 10.1007/BF02163027. URL https://doi.org/10.1007/BF02163027.

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.

Daya Guo, Canwen Xu, Nan Duan, Jian Yin, and Julian Mcauley. LongCoder: A long-range pre-trained language model for code completion. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12098–12107. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/guo23j.html.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Shiyu Ji, Yixuan Wang, Yijun Liu, Qingfu Zhu, and Wanxiang Che. Echokv: Efficient kv cache compression via similarity-based reconstruction, 2026. URL https://arxiv.org/abs/2603.22910.

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.

Donghyeon Joo, Helya Hosseini, Ramyad Hadidi, and Bahar Asgari. MUSTAFAR: Promoting unstructured sparsity for KV cache pruning in LLM inference. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=C69741fMFX.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082/.

Bokai Lin, Zihao Zeng, Zipeng Xiao, Siqi Kou, TianQi Hou, Xiaofeng Gao, Hao Zhang, and Zhijie Deng. MatryoshkaKV: Adaptive KV compression via trainable orthogonal projection. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=BQwsRy1h3U.

Tianyang Liu, Canwen Xu, and Julian McAuley. Repobench: Benchmarking repository-level code auto-completion systems. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=pPjZIOuQuF.

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*, 2024b. URL https://openreview.net/forum?id=L057s2Rq8O.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. URL https://aclanthology.org/J93-2004/.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

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/.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849. Curran Associates, Inc., 2024. doi: 10.52202/079017-0970. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/370df50ccfdf8bde18f8f9c2d9151bda-Paper-Datasets_and_Benchmarks_Track.pdf.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399. URL https://doi.org/10.1609/aaai.v34i05.6399.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. Eigen attention: Attention in low-rank space for KV cache compression. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15332–15344, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.899. URL https://aclanthology.org/2024.findings-emnlp.899/.

Khalid Shaikh, Asmit Kumar Singh, Rebecca Dsouza, and Shikhar Shiromani. Linear predictability of attention heads in large language models. In *The 1st International Workshop on Sustainable and Efficient Language, Vision, and Action Models (SELVA) at ACL 2026*, 2026. URL https://openreview.net/forum?id=rsQm4VxNKV.

Alina Shutova, Vladimir Malinovskii, Vage Egiazarian, Denis Kuznedelev, Denis Mazur, Surkov Nikita, Ivan Ermakov, and Dan Alistarh. Cache me if you must: Adaptive key-value quantization for large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=COowwJOAZi.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C), February 2024. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

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, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LNYIUouhdt.

Jingyang Xiang and Sai Qian Zhang. DFRot: Achieving outlier-free and massive activation-free for rotated LLMs with refined rotation. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=WzGypILLDb.

Yuhui Xu, Zhanming Jie, Hanze Dong, Lei Wang, Xudong Lu, Aojun Zhou, Amrita Saha, Caiming Xiong, and Doyen Sahoo. Think: Thinner key cache by query-driven pruning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=n0OtGl6VGb.

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, 2025. URL https://arxiv.org/abs/2505.24357.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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

Rowan Zellers, 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/.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Re, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=RkRrPp7GKO.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A new benchmark for query-based multi-domain meeting summarization. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5905–5921, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.472. URL https://aclanthology.org/2021.naacl-main.472/.

Zhongzhu Zhou, Fengxiang Bie, Ziyan Chen, Zhenyu Zhang, Yibo Yang, Junxiong Wang, Ben Athiwaratkun, Xiaoxia Wu, and Shuaiwen Leon Song. CARE: Covariance-aware and rank-enhanced decomposition for enabling multi-head latent attention. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DVurf4kGag.

A PRELIMINARIES

We consider a decoder-only Transformer (Vaswani et al., 2017) with L layers. At layer ℓ, let H_{kv} denote the number of key–value (KV) heads and let d_h denote the head dimension; we omit their layer dependence when no ambiguity arises. In grouped-query attention (GQA) (Ainslie et al., 2023), multiple query heads share each KV head. Rotary position embedding (RoPE) (Su et al., 2024) applies a position-dependent rotation $\mathcal{R}_p$ to queries and keys before attention. During autoregressive decoding, the native cache stores $H_{\mathrm{kv}} d_h$ scalars per token for each K/V component. We count only token-dependent cached representations and exclude token-independent parameters of the compression transform.

B EXPERIMENTAL DETAILS

For TPNP, we fit the cross-head mappings and innovation bases using 4,096 hidden-state samples per layer and compute Fisher–Rao sensitivity using all 128 calibration sequences. For the cross-head mappings, we set the ridge penalty to 10^{-5} times the mean diagonal of the corresponding Gram matrix. For perplexity, we use a context length of 2,048 for both Qwen2.5-7B and Yi-1.5-6B. For accuracy, we cap the context length at 2,048 for both backbones. For LongBench, we cap the context length at 8,192 for Qwen2.5-7B and 4,096 for Yi-1.5-6B. We report perplexity for WikiText-2 and Penn Treebank, accuracy for ARC-Easy, HellaSwag, PIQA, OpenBookQA, RACE, and WinoGrande, F1 for TriviaQA, ROUGE-L for QMSum, MultiNews, and SAMSum, and edit similarity for LCC and RepoBench-P. All averages are computed from unrounded metric values and rounded to two decimal places only for reporting, and may therefore occasionally differ by 0.01 from averages recomputed from the displayed values.

For CARE, we use the one-shot CARE-E without post-SVD healing. For SVD-LLM, we apply its truncation-aware whitening/SVD to the K/V projections and omit its sequential low-rank parameter update to maintain a training-free comparison. Eigen Attention is not evaluated at a 70% KV saving ratio because its original implementation restricts the minimum retained rank, making evaluation at this ratio infeasible. All experiments are conducted on a single NVIDIA A100 GPU.

C COMPARISON WITH PREDICTION-FREE JOINT LOW-RANK CODING

Method	ρ	Perplexity			Accuracy						
		WikiText-2	Penn Treebank	Average ↓	ARC	OBQA	HellaSwag	PIQA	RACE	WinoGrande	Average ↑
Qwen2.5-7B											
Prediction-Free Joint Low-Rank Coding	0.1	6.81	11.32	9.07	78.49	34.40	78.75	78.62	42.11	72.53	64.15
TPNP		6.81	11.33	9.07	78.20	33.40	78.83	78.84	42.11	72.93	64.05
Prediction-Free Joint Low-Rank Coding	0.3	6.86	11.43	9.14	79.25	33.80	79.04	78.45	40.57	71.82	63.82
TPNP		6.86	11.42	9.14	78.83	33.60	78.94	78.62	40.77	72.30	63.84
Prediction-Free Joint Low-Rank Coding	0.5	7.17	12.30	9.74	81.73	32.80	77.50	79.22	38.66	69.46	63.23
TPNP		7.14	12.27	9.71	80.85	32.00	77.38	79.05	40.00	69.93	63.20
Prediction-Free Joint Low-Rank Coding	0.7	9.63	19.72	14.68	55.72	26.40	65.97	72.20	35.50	62.98	53.13
TPNP		9.45	19.89	14.67	66.92	29.20	66.77	74.86	35.31	60.54	55.60
Yi-1.5-6B											
Prediction-Free Joint Low-Rank Coding	0.1	5.78	17.66	11.72	75.25	31.80	75.20	78.40	38.56	71.98	61.87
TPNP		5.78	17.65	11.72	75.13	31.60	75.28	78.29	38.56	72.61	61.91
Prediction-Free Joint Low-Rank Coding	0.3	5.84	18.14	11.99	74.87	32.20	75.21	78.51	38.85	71.43	61.85
TPNP		5.84	18.32	12.08	75.59	31.80	75.12	78.29	38.66	71.59	61.84
Prediction-Free Joint Low-Rank Coding	0.5	6.23	22.77	14.50	71.34	31.60	72.65	77.31	37.70	68.59	59.87
TPNP		6.11	22.38	14.25	64.94	30.80	73.80	77.04	38.37	69.14	59.02
Prediction-Free Joint Low-Rank Coding	0.7	9.62	55.21	32.41	45.24	21.00	58.13	69.97	31.58	61.33	47.88
TPNP		7.82	45.42	26.62	52.15	25.40	65.74	72.96	35.89	62.67	52.47

Table 5: Comparison of TPNP with prediction-free joint low-rank coding on Qwen2.5 and Yi-1.5.

For comparison, we construct a prediction-free variant that restricts each layer-wise K/V component to the native representation or prediction-free joint low-rank coding, while retaining Fisher–Rao basis ranking and global KV-cache budget allocation. As shown in Table 5, TPNP yields its clearest gains over this prediction-free variant at a 70% KV saving ratio. On Qwen2.5-7B, TPNP improves average accuracy from 53.13 to 55.60 while maintaining nearly identical average perplexity (14.68 vs. 14.67). On Yi-1.5-6B, TPNP reduces average perplexity from 32.41 to 26.62 and improves average accuracy from 47.88 to 52.47, with accuracy gains on all six tasks.

At lower KV saving ratios, the differences are smaller and more mixed. On Qwen2.5-7B, TPNP and the prediction-free variant achieve similar average perplexity and average accuracy at saving ratios up to 50%. On Yi-1.5-6B, they have the same rounded average perplexity at 10%, while the prediction-free variant achieves slightly lower average perplexity at 30% (11.99 vs. 12.08); average accuracy remains similar at both ratios. At 50%, TPNP reduces average perplexity from 14.50 to 14.25 but yields lower average accuracy (59.02 vs. 59.87). Thus, the benefit of cross-head prediction depends on the backbone, the KV-cache budget, and the evaluation metric, with the clearest gains at a 70% KV saving ratio. Together with the analysis in Section 5.1, these results support retaining both prediction-assisted innovation coding and prediction-free joint low-rank coding in the unified search space, with the global KV-cache budget allocator selecting one option per layer-wise K/V component.

D RULER Evaluation

Method	4K	8K	16K	32K	Average ↑
Original	93.50	89.30	87.52	84.65	88.74
ASVD	13.68	11.61	9.37	5.15	9.95
CARE	32.00	27.38	17.42	5.95	20.69
Eigen Attention	–	–	–	–	–
Palu	64.32	55.10	49.05	37.47	51.48
ReCalKV	64.94	57.33	50.57	41.32	53.54
SVD-LLM	46.42	39.42	28.80	17.05	32.92
TPNP	55.74	47.32	41.28	32.77	44.28

Table 6: RULER comparison on Qwen2.5 at a 70% KV saving ratio.

We evaluate TPNP on RULER (Hsieh et al., 2024) using Qwen2.5-7B at a fixed 70% KV saving ratio across context lengths of 4K, 8K, 16K, and 32K. As shown in Table 6, TPNP ranks third at every context length behind ReCalKV and Palu. It achieves an average score of 44.28, compared with 53.54 for ReCalKV and 51.48 for Palu, while consistently outperforming SVD-LLM, CARE, and ASVD across all context lengths.

At the same 70% KV saving ratio on Qwen2.5-7B, ASVD achieves an average accuracy close to that of TPNP in the accuracy evaluation (55.15 vs. 55.60), but a substantially lower average RULER score (9.95 vs. 44.28). At 32K, TPNP scores 32.77, whereas ASVD scores 5.15. These results show that relative performance can differ between the accuracy evaluation and RULER, although all evaluated compression methods substantially underperform the original model on RULER.

E Calibration Robustness Evaluation

To assess robustness to calibration, we replace the WikiText-2 calibration used in the main experiments with two alternatives, C4 (Raffel et al., 2020) and FineWeb-Edu (Penedo et al., 2024), and evaluate all methods at a fixed 70% KV saving ratio. As shown in Table 7, TPNP achieves the lowest average perplexity and highest average accuracy on Qwen2.5-7B under both C4 and FineWeb-Edu. On Yi-1.5-6B, TPNP achieves the highest average accuracy under both calibrations and nearly matches the lowest average perplexity under FineWeb-Edu.

F Additional Backbone Evaluation

To assess generalization beyond the main experiments, we report results on Llama-3.1-8B (Grattafiori et al., 2024) and Mistral-7B-v0.3 (Jiang et al., 2023). As shown in Table 8, on Llama-3.1-8B, TPNP achieves the lowest average perplexity among compression methods at KV saving ratios of 10%, 30%, and 50%, and the second-lowest at 70%, where it also achieves the highest average accuracy. On Mistral-7B-v0.3, TPNP achieves the lowest average perplexity at KV saving ratios of 50% and 70% and the second-highest average accuracy among compression methods at all four ratios, with a tie at 30%.

Method	Calibration	Perplexity			Accuracy						
		WikiText-2	Penn Treebank	Average ↓	ARC	OBQA	HellaSwag	PIQA	RACE	WinoGrande	Average ↑
Qwen2.5-7B											
ASVD		19.47	38.94	29.20	69.53	**28.20**	62.50	75.24	33.01	58.88	54.56
CARE		37.48	53.32	45.40	38.64	13.60	43.69	64.47	32.92	58.72	42.01
Eigen Attention		–	–	–	–	–	–	–	–	–	–
Palu	C4	12.86	22.53	17.70	65.70	25.20	64.66	73.23	37.03	64.64	55.08
ReCalKV		12.43	21.88	17.15	66.12	24.00	65.54	74.05	37.70	**65.90**	55.55
SVD-LLM		39.47	59.46	49.46	41.96	16.80	47.83	64.42	36.84	58.01	44.31
TPNP		**11.39**	**18.96**	**15.17**	**72.18**	27.00	**68.95**	**76.66**	**38.28**	63.85	**57.82**
ASVD		17.72	35.59	26.66	65.70	**29.60**	63.26	75.63	31.96	59.59	54.29
CARE		39.13	58.65	48.89	43.86	17.20	44.11	64.58	33.21	56.91	43.31
Eigen Attention		–	–	–	–	–	–	–	–	–	–
Palu	FineWeb-Edu	14.35	26.56	20.46	69.02	27.40	64.52	73.07	37.42	63.69	55.85
ReCalKV		13.81	25.54	19.68	69.53	27.80	64.89	72.96	37.70	**65.35**	56.37
SVD-LLM		42.52	82.96	62.74	50.63	20.60	46.84	62.40	36.08	58.72	45.88
TPNP		**12.35**	**25.29**	**18.82**	**75.46**	29.20	**68.44**	**77.15**	**37.80**	62.83	**58.48**
Yi-1.5-6B											
ASVD		36.42	226.24	131.33	44.65	17.20	44.94	66.27	28.90	51.30	42.21
CARE		26.73	139.23	82.98	43.27	18.80	48.76	67.63	31.48	56.99	44.49
Eigen Attention		–	–	–	–	–	–	–	–	–	–
Palu	C4	12.81	50.58	31.70	46.51	20.00	60.94	73.29	34.64	61.48	49.48
ReCalKV		12.89	**48.77**	**30.83**	47.18	19.60	60.93	72.20	34.55	63.22	49.61
SVD-LLM		15.98	66.08	41.03	50.55	20.00	54.24	70.40	34.64	59.59	48.24
TPNP		**11.85**	55.58	33.71	**57.83**	**25.00**	**64.24**	**74.16**	**35.12**	**63.38**	**53.29**
ASVD		26.20	212.68	119.44	45.12	16.20	47.02	66.49	30.33	52.72	42.98
CARE		25.70	146.20	85.95	46.30	19.80	49.67	67.63	30.62	56.83	45.14
Eigen Attention		–	–	–	–	–	–	–	–	–	–
Palu	FineWeb-Edu	**12.05**	54.17	33.11	47.43	24.60	61.37	71.33	33.97	60.62	49.89
ReCalKV		12.22	**53.94**	**33.08**	47.10	23.40	60.94	71.27	34.26	60.54	49.59
SVD-LLM		16.05	82.15	49.10	**56.02**	22.80	54.99	70.08	33.78	57.93	49.27
TPNP		12.21	53.97	33.09	53.62	**25.60**	**62.83**	**73.61**	**35.12**	**61.80**	**52.10**

Table 7: Calibration robustness on Qwen2.5 and Yi-1.5 at a 70% KV saving ratio.

G Efficiency Analysis

G.1 Theoretical KV-Cache Footprint

Following prior work (Zhou et al., 2026), we evaluate memory efficiency in terms of the theoretical token-dependent KV-cache footprint. Since KV-cache memory scales linearly with context length, we report the footprint at a context length of 32K. For Qwen2.5-7B with batch size 1 and 16-bit cached representations, the original model stores 28,672 cached scalars per token across all layers, corresponding to 1,792.00 MiB at 32K. We count only token-dependent cached representations and exclude token-independent parameters of the compression transform. As shown in Table 9, the realized KV-cache reductions closely match the target KV saving ratio ρ across methods, with a maximum deviation of 0.31 percentage points. The remaining differences arise from the discrete ranks or compression settings supported by each method. Thus, the Qwen2.5-7B performance comparisons are conducted at comparable token-dependent KV-cache footprints rather than materially different cache capacities.

G.2 Offline Preprocessing Cost

Offline preprocessing time and peak GPU memory can depend on implementation details and the hardware and software environment, and therefore should not be interpreted as implementation-independent measures of computational complexity. To provide a practical reference, we measure both quantities for all methods on Qwen2.5-7B under the same single-GPU environment. We measure offline preprocessing time as synchronized wall-clock time from method-specific preprocessing through artifact serialization, excluding model, tokenizer, and calibration-data loading. Peak GPU memory is measured after resetting the CUDA peak-memory statistics immediately before each method.

As shown in Table 10, TPNP requires 126.07–132.24 seconds for offline preprocessing across the four KV saving ratios, showing little variation with the target ratio. It consistently requires less preprocessing time than ASVD, ReCalKV, and Eigen Attention at the three ratios where Eigen Attention is evaluated, but more than CARE, Palu, and SVD-LLM. TPNP's peak GPU memory

Method	ρ	Perplexity			Accuracy						
		WikiText-2	Penn Treebank	Average ↓	ARC	OBQA	HellaSwag	PIQA	RACE	WinoGrande	Average ↑
Llama-3.1-8B											
Original	0.0	6.19	9.98	8.09	82.70	34.00	79.26	79.22	39.52	74.19	64.82
ASVD	0.1	6.23	10.04	8.14	81.65	34.60	79.27	**79.65**	**40.29**	**74.11**	**64.93**
CARE	0.1	6.41	10.66	8.54	81.86	**35.40**	**80.16**	79.05	38.76	73.80	64.84
Eigen Attention	0.1	6.41	10.20	8.31	79.59	34.00	78.88	78.07	39.43	72.93	63.82
Palu	0.1	6.23	10.04	8.14	81.78	34.80	79.14	79.38	39.71	73.56	64.73
ReCalKV	0.1	6.23	10.04	8.14	81.48	34.60	79.26	78.94	39.62	73.88	64.63
SVD-LLM	0.1	6.44	10.35	8.40	79.63	33.80	79.04	78.67	38.85	72.85	63.81
TPNP	0.1	**6.20**	**10.00**	**8.10**	**82.74**	34.00	79.44	78.78	39.14	73.95	64.68
ASVD	0.3	6.47	10.38	8.43	**81.65**	34.00	79.10	79.27	39.62	74.11	64.63
CARE	0.3	6.92	12.46	9.69	77.15	32.40	77.64	76.93	37.89	73.32	62.56
Eigen Attention	0.3	7.49	12.65	10.07	75.21	31.60	73.68	76.50	39.90	66.69	60.60
Palu	0.3	6.57	10.75	8.66	80.35	34.20	**79.34**	79.43	**40.77**	73.95	64.67
ReCalKV	0.3	6.58	10.74	8.66	80.68	**34.60**	79.13	**79.65**	**40.77**	**74.59**	**64.90**
SVD-LLM	0.3	6.79	10.97	8.88	77.95	31.00	75.89	78.18	35.50	72.85	61.90
TPNP	0.3	**6.25**	**10.15**	**8.20**	80.60	32.80	79.04	79.38	39.81	73.24	64.15
ASVD	0.5	7.46	11.84	9.65	**79.92**	**34.60**	77.64	78.45	**39.14**	**74.19**	**63.99**
CARE	0.5	9.10	18.58	13.84	59.76	25.60	67.23	72.14	36.94	69.38	55.18
Eigen Attention	0.5	12.35	32.09	22.22	53.20	21.60	44.19	65.61	29.28	56.99	45.15
Palu	0.5	7.98	14.44	11.21	70.29	33.20	71.14	76.88	38.18	70.72	60.07
ReCalKV	0.5	7.94	14.29	11.12	72.05	33.40	72.02	77.09	38.56	71.35	60.75
SVD-LLM	0.5	7.84	13.66	10.75	68.10	27.80	61.68	73.50	32.63	69.14	55.48
TPNP	0.5	**6.53**	**11.49**	**9.01**	78.45	31.40	**77.97**	**80.14**	39.04	72.93	63.32
ASVD	0.7	12.18	**19.41**	**15.80**	63.93	26.80	67.41	74.48	33.40	68.03	55.68
CARE	0.7	12.21	31.27	21.74	47.43	19.60	53.22	66.54	27.46	65.82	46.68
Eigen Attention	0.7	–	–	–	–	–	–	–	–	–	–
Palu	0.7	15.34	43.32	29.33	44.57	21.60	52.26	68.12	29.47	61.09	46.19
ReCalKV	0.7	13.78	34.26	24.02	51.94	23.80	56.03	70.62	29.57	63.30	49.21
SVD-LLM	0.7	13.69	38.03	25.86	42.13	17.40	37.98	60.88	29.19	58.09	40.95
TPNP	0.7	**8.04**	23.85	15.95	**71.55**	**31.60**	**72.35**	**77.53**	**35.79**	**69.22**	**59.67**
Mistral-7B-v0.3											
Original	0.0	5.37	28.29	16.83	80.01	35.20	80.73	79.98	40.10	74.59	65.10
ASVD	0.1	**5.37**	28.47	16.92	**80.01**	35.60	80.69	80.03	39.62	74.11	65.01
CARE	0.1	5.44	28.72	17.08	78.96	**37.20**	**81.15**	79.87	40.48	**74.51**	**65.36**
Eigen Attention	0.1	5.53	28.38	16.96	77.99	34.60	80.31	79.71	**42.20**	73.88	64.78
Palu	0.1	5.39	28.53	16.96	79.50	35.60	80.76	79.92	39.90	74.27	64.99
ReCalKV	0.1	5.39	28.54	16.97	79.55	35.40	80.78	79.82	40.48	74.11	65.02
SVD-LLM	0.1	5.43	**27.56**	**16.50**	78.28	36.40	80.65	79.98	40.10	74.03	64.91
TPNP	0.1	5.38	28.32	16.85	79.59	36.40	80.64	**80.41**	39.81	74.19	65.17
ASVD	0.3	5.41	30.75	18.08	**80.13**	35.60	80.73	80.09	40.96	74.51	**65.34**
CARE	0.3	5.76	31.63	18.70	74.87	32.60	78.02	78.13	39.81	73.95	62.90
Eigen Attention	0.3	6.10	45.99	26.05	73.74	30.00	70.67	75.79	40.38	66.30	59.48
Palu	0.3	5.50	28.58	17.04	78.58	**36.40**	**81.00**	79.98	39.81	74.35	65.02
ReCalKV	0.3	5.47	28.72	17.10	79.34	35.60	80.98	79.92	40.57	74.11	65.09
SVD-LLM	0.3	5.52	**27.64**	**16.58**	77.57	34.40	79.09	79.43	**41.15**	73.95	64.27
TPNP	0.3	**5.39**	28.80	17.10	79.21	35.20	80.49	**80.41**	40.00	**75.22**	65.09
ASVD	0.5	5.64	32.06	18.85	**79.00**	**35.20**	**79.99**	**80.58**	**41.53**	**74.19**	**65.08**
CARE	0.5	6.71	40.73	23.72	62.63	28.00	68.20	74.16	37.99	70.88	56.98
Eigen Attention	0.5	8.21	87.72	47.97	53.54	21.60	48.31	67.74	32.15	58.88	47.04
Palu	0.5	5.94	31.47	18.71	74.33	34.80	77.88	78.67	40.10	73.40	63.20
ReCalKV	0.5	5.82	31.47	18.65	75.25	34.20	78.06	78.62	40.00	73.88	63.34
SVD-LLM	0.5	5.79	30.47	18.13	73.06	32.60	74.76	78.67	41.05	73.32	62.24
TPNP	0.5	**5.50**	**29.81**	**17.66**	78.70	35.00	79.82	79.92	40.19	74.11	64.62
ASVD	0.7	7.01	43.59	25.30	**73.57**	**32.00**	72.98	**78.73**	**39.90**	**71.11**	**61.38**
CARE	0.7	8.03	51.58	29.81	52.27	21.00	59.04	70.02	35.60	65.75	50.61
Eigen Attention	0.7	–	–	–	–	–	–	–	–	–	–
Palu	0.7	7.67	47.15	27.41	54.50	22.40	62.48	72.91	37.13	62.35	51.96
ReCalKV	0.7	7.17	42.26	24.72	63.30	24.80	65.17	74.97	37.22	66.22	55.28
SVD-LLM	0.7	6.93	45.09	26.01	55.26	21.60	55.56	69.91	37.99	62.90	50.54
TPNP	0.7	**6.11**	**38.09**	**22.10**	70.29	29.20	**73.77**	78.02	39.71	70.72	60.29

Table 8: Comparison of KV-cache compression methods on Llama-3.1 and Mistral-v0.3.

remains at 28.971 GiB across all four ratios, exceeding the 16.421–24.371 GiB range observed for the other methods.

H Discussion

H.1 Compatibility

Following prior work (Lin et al., 2025; Xu et al., 2025; Joo et al., 2025), we evaluate the compatibility of TPNP with complementary KV-cache compression methods by combining it separately with

Method	ρ	Cached Scalars / Token	KV-Cache Memory @ 32K (MiB)	Reduction vs. Original (%)
Original	0.0	28672	1792.00	–
ASVD	0.1	25820	1613.75	9.95
CARE		25804	1612.75	10.00
Eigen Attention		25816	1613.50	9.96
Palu		25824	1614.00	9.93
ReCalKV		25824	1614.00	9.93
SVD-LLM		25816	1613.50	9.96
TPNP		25804	1612.75	10.00
ASVD	0.3	20079	1254.94	29.97
CARE		20070	1254.38	30.00
Eigen Attention		20060	1253.75	30.04
Palu		20160	1260.00	29.69
ReCalKV		20160	1260.00	29.69
SVD-LLM		20048	1253.00	30.08
TPNP		20070	1254.38	30.00
ASVD	0.5	14338	896.12	49.99
CARE		14336	896.00	50.00
Eigen Attention		14312	894.50	50.08
Palu		14304	894.00	50.11
ReCalKV		14304	894.00	50.11
SVD-LLM		14336	896.00	50.00
TPNP		14336	896.00	50.00
ASVD	0.7	8604	537.75	69.99
CARE		8602	537.62	70.00
Eigen Attention		–	–	–
Palu		8672	542.00	69.75
ReCalKV		8672	542.00	69.75
SVD-LLM		8624	539.00	69.92
TPNP		8601	537.56	70.00

Table 9: Theoretical token-dependent KV-cache footprint on Qwen2.5.

Method	ρ	Offline Preprocessing Time (s) ↓	Peak GPU Memory (GiB) ↓
ASVD	0.1	22893.772	16.526
CARE		45.778	17.780
Eigen Attention		608.445	16.421
Palu		85.385	24.371
ReCalKV		255.174	24.371
SVD-LLM		40.456	17.788
TPNP		128.373	28.971
ASVD	0.3	22720.255	16.526
CARE		45.337	17.780
Eigen Attention		577.816	16.421
Palu		83.533	24.371
ReCalKV		259.173	24.371
SVD-LLM		40.861	17.788
TPNP		130.025	28.971
ASVD	0.5	23244.767	16.526
CARE		49.534	17.780
Eigen Attention		586.865	16.421
Palu		87.026	24.371
ReCalKV		264.659	24.371
SVD-LLM		40.904	17.788
TPNP		132.241	28.971
ASVD	0.7	22709.791	16.526
CARE		43.521	17.780
Eigen Attention		–	–
Palu		82.306	24.371
ReCalKV		248.352	24.371
SVD-LLM		38.952	17.788
TPNP		126.067	28.971

Table 10: Offline preprocessing cost on Qwen2.5.

H2O for token eviction and KIVI for quantization. For H2O, we use a 20% heavy-plus-recent token budget, with 10% allocated to each component. For KIVI, we apply 2-bit quantization to the KV representations stored by TPNP before reconstruction. Here, ρ denotes the KV saving ratio of TPNP before applying H2O or KIVI.

As shown in Table 11, both combinations increase average perplexity and reduce average accuracy relative to TPNP alone. Nevertheless, KIVI further reduces the nominal packed KV-cache footprint by approximately 86.5–86.8% relative to TPNP alone, with average accuracy drops of 1.21–3.96 percentage points on Qwen2.5-7B and 0.12–0.53 percentage points on Yi-1.5-6B. With its 20% token

Method	ρ	Perplexity			Accuracy						
		WikiText-2	Penn Treebank	Average ↓	ARC	OBQA	HellaSwag	PIQA	RACE	WinoGrande	Average ↑
Qwen2.5-7B											
TPNP + H2O	0.1	7.76	12.59	10.17	59.68	19.60	75.73	72.80	41.63	52.09	53.59
TPNP + KIVI	0.1	10.24	15.38	12.81	75.72	31.80	76.25	78.13	40.86	70.64	62.23
TPNP	0.1	6.81	11.33	9.07	78.20	33.40	78.83	78.84	42.11	72.93	64.05
TPNP + H2O	0.3	7.81	12.79	10.30	56.78	18.60	75.92	72.69	40.10	52.33	52.73
TPNP + KIVI	0.3	10.01	14.99	12.50	73.15	31.60	76.33	77.20	41.91	70.56	61.79
TPNP	0.3	6.86	11.42	9.14	78.83	33.60	78.94	78.62	40.77	72.30	63.84
TPNP + H2O	0.5	8.17	13.91	11.04	59.93	19.40	74.18	72.25	38.66	52.72	52.86
TPNP + KIVI	0.5	10.70	16.32	13.51	79.71	30.00	74.50	77.75	40.19	69.77	61.99
TPNP	0.5	7.14	12.27	9.71	80.85	32.00	77.38	79.05	40.00	69.93	63.20
TPNP + H2O	0.7	12.75	29.70	21.23	41.71	17.60	60.08	67.30	32.44	51.62	45.12
TPNP + KIVI	0.7	18.08	39.26	28.67	59.76	26.80	57.47	71.44	33.68	60.69	51.64
TPNP	0.7	9.45	19.89	14.67	66.92	29.20	66.77	74.86	35.31	60.54	55.60
Yi-1.5-6B											
TPNP + H2O	0.1	5.86	18.71	12.29	45.08	19.00	73.23	68.50	38.47	53.67	49.66
TPNP + KIVI	0.1	6.05	18.70	12.38	74.83	31.80	74.28	78.29	39.33	71.82	61.73
TPNP	0.1	5.78	17.65	11.72	75.13	31.60	75.28	78.29	38.56	72.61	61.91
TPNP + H2O	0.3	5.92	18.95	12.43	46.42	19.20	72.54	69.75	38.47	55.72	50.35
TPNP + KIVI	0.3	6.11	19.27	12.69	74.79	31.60	74.43	78.07	39.52	71.90	61.72
TPNP	0.3	5.84	18.32	12.08	75.59	31.80	75.12	78.29	38.66	71.59	61.84
TPNP + H2O	0.5	6.23	23.66	14.95	41.75	19.60	70.86	68.93	36.36	56.20	48.95
TPNP + KIVI	0.5	6.38	23.16	14.77	64.48	30.40	73.22	77.20	37.70	68.98	58.66
TPNP	0.5	6.11	22.38	14.25	64.94	30.80	73.80	77.04	38.37	69.14	59.02
TPNP + H2O	0.7	8.03	49.15	28.59	37.84	16.80	61.93	64.53	34.07	53.59	44.79
TPNP + KIVI	0.7	8.29	48.19	28.24	51.81	25.60	63.84	72.42	35.79	62.19	51.94
TPNP	0.7	7.82	45.42	26.62	52.15	25.40	65.74	72.96	35.89	62.67	52.47

Table 11: Compatibility with H2O and KIVI on Qwen2.5 and Yi-1.5.

budget, H2O incurs a larger average accuracy drop than KIVI. These results demonstrate that TPNP can be combined with both quantization and token eviction as complementary KV-cache compression mechanisms.

H.2 Complexity

We analyze the offline preprocessing and inference costs of TPNP. Assume the same H_{kv} and d_h across layers, and let d_{model} denote the input hidden dimension. Let $m \geq H_{\mathrm{kv}} d_h$ be the number of fitting samples, as in our experiments, and let N and S_{seq} denote the number and length of calibration sequences. For a partition with A anchors and T targets, let $D = Td_h$. Each layer-wise K/V component has $J = 2^{H_{\mathrm{kv}}} - 1$ non-native anchor–target partitions, whose fitted mappings and bases are reused across innovation ranks.

Computing the fitting representations costs $O(md_{\mathrm{model}} H_{\mathrm{kv}} d_h)$ per component. For $A > 0$, separate ridge solves for the target heads give fitting cost bounds of

$$O\big(T[m(Ad_h)^2 + (Ad_h)^3]\big) \quad \text{and} \quad O\big(Td_h[mA^2 + A^3]\big)$$

for value and RoPE-constrained key predictors, respectively. Computing the D-dimensional innovation basis from $\mathbf{E} \in \mathbb{R}^{m \times D}$ using an economy-size SVD costs $O(mD^2)$. Beyond the shared calibration forward and backward passes, Fisher–Rao evaluation costs $O\big(NS_{\mathrm{seq}}[D^2 + ATd_h^2]\big)$ per partition using the dense predictor bound, followed by $O(D \log D)$ sorting and $O(D)$ suffix accumulation to obtain costs for all ranks. Exhaustive candidate preprocessing therefore scales exponentially with H_{kv} through J. Storing all fitted mappings and bases requires $O\big(LJ(H_{\mathrm{kv}} d_h)^2\big)$ scalars in the worst case.

Retaining only the minimum-cost option for each distinct saving reduces the $1 + H_{\mathrm{kv}} d_h\, 2^{H_{\mathrm{kv}}-1}$ options to at most $H_{\mathrm{kv}} d_h + 1$ per component without changing the optimum. With $M_{\mathrm{native}} = 2LH_{\mathrm{kv}} d_h$, the dynamic program for Eq. (18) requires $O(LM_{\mathrm{native}} H_{\mathrm{kv}} d_h)$ time, $O(M_{\mathrm{native}})$ state storage, and $O(LM_{\mathrm{native}})$ backtracking storage. The partition-specific fitted mappings and bases, together with the candidate Fisher–Rao costs, can be reused across target KV saving ratios.

For a selected compressed component, constructing the folded projection in Eq. (12) costs $O\big(d_{\mathrm{model}}[ATd_h^2 + Dr]\big)$ using dense predictor products. Encoding each new token costs $O(d_{\mathrm{model}}[Ad_h + r])$ and stores $Ad_h + r$ persistent cached scalars, as in Eq. (16). Reconstructing a cached context of length n_{ctx} before attention costs

$$O\big(n_{\mathrm{ctx}}[ATd_h^2 + Dr + H_{\mathrm{kv}} d_h]\big)$$

per compressed component under the same dense-product bound. This excludes the subsequent attention computation. Prediction-free joint low-rank coding follows by setting $A = 0$ and $D = H_{\mathrm{kv}} d_h$.

H.3 LIMITATIONS

(i) Partition enumeration: TPNP exhaustively considers anchor–target partitions for each layer-wise K/V component, so the offline candidate preprocessing cost scales exponentially with the number of KV heads; more scalable partition enumeration may be needed for architectures with larger numbers of KV heads.

(ii) Kernel support: our current implementation does not include dedicated fused kernels for the heterogeneous anchor–target partitions and innovation ranks selected by TPNP. Reconstruction adds computation before attention, but its wall-clock impact depends on kernel implementation. We report the theoretical token-dependent KV-cache footprint and offline preprocessing cost, while end-to-end latency and throughput with optimized kernels remain to be evaluated.

(iii) Sensitivity approximation: TPNP estimates coordinate-removal costs using a diagonal empirical-Fisher approximation and assumes additive compression costs across layer-wise K/V components. These approximations do not capture interactions among basis directions or across layer-wise K/V components; richer sensitivity models remain to be explored.