KVgrad: Query-Agnostic KV Cache Eviction via Gradient-based Global Importance Scoring

Jihwan Kwak [*1] **Sunghwan Joo** [*2] **Jung Yoon Hwang** [3] **Jaeseok Byun** [1] **Taesup Moon** [1 2 3 4]

Abstract

Transformer-based Large Language Models (LLMs) suffer from substantial memory overhead and latency bottlenecks due to the linear scaling of the Key-Value (KV) cache. Existing KV cache eviction methods typically rely on local heuristics, scoring cached entries by their immediate contribution to the attention output while ignoring the downstream propagation of information to the final representation. To that end, we propose **KVgrad**, which carries out the eviction by using a query-agnostic importance score that quantifies a cache entry's global impact on the final representation. Based on a simple analysis using Taylor approximation and chain-rule, KVgrad factorizes the impact of a cached value into a linear local component and a gradient-based downstream sensitivity term. Then, we apply a magnitude-aware scoring scheme to stabilize the importance score. Evaluations on RULER and LongBench demonstrate that KVgrad achieves an average maximum compression ratio of 6.42$\times$ with negligible performance loss ($< 2\%$), significantly outperforming strong local-only state-of-the-art baselines. Furthermore, we demonstrate that our signal is highly amenable to distillation; by training a lightweight MLP to predict KVgrad scores from local hidden-state features, we enable high-fidelity, on-the-fly compression at negligible computational cost.

1. Introduction

Autoregressive generation in large language models is fundamentally *memory-bandwidth bound* (Shazeer, 2019; Pope et al., 2023). During inference, the entire key-value (KV) cache must be fetched from GPU memory to compute attention for each new token (Shah et al., 2024). In long-context scenarios, such as agentic workflows or multimodal processing, KV cache transfers can account for over 93% of GPU kernel time (Jiang et al., 2025).

To alleviate this bottleneck, post-hoc KV cache eviction(Zhang et al., 2023; Li et al., 2024; Cai et al., 2024; Feng et al., 2024) has emerged as a high-utility strategy: it does not require retraining the base model and is compatible with other compression techniques like quantization (Kim et al., 2025; Ramachandran et al., 2025). The primary challenge lies in "eviction policy", *i.e.,* determining which KV pairs to retain based on an assigned importance score. Query-agnostic approaches (Kim et al., 2025; Jegou & Jeblick, 2026) are particularly desirable for deployment because their scores depend only on the context. This allows scores to be computed offline for multi-turn reuse or distilled into a lightweight surrogate for efficient online compression (Jegou & Jeblick, 2026). Currently, state-of-the-art query-agnostic methods like KVzip (Kim et al., 2025) derive these scores by measuring how essential a cache entry is for the model to reconstruct its own context.

However, existing eviction methods—whether query-aware or query-agnostic—typically evaluate the importance of cached key-values from a **local** view, considering only each entry's direct contribution to the attention output ($\mathbf{a}_p^\ell$) at its own layer (Figure 1 (a)). This perspective ignores the **downstream propagation** from that attention output through subsequent Feed-Forward Network (FFN) and higher layers before reaching the final hidden state ($\mathbf{x}_p^{L+1}$). Because a model's prediction ultimately depends on this **global** influence, local-only heuristics can be misleading.

To bridge this gap, we propose **KVgrad**, a query-agnostic importance score derived from a first-order Taylor expansion that quantifies the global impact of removing a cache entry. Using a per-probe chain-rule decomposition, we factorize this global influence into two distinct stages: (1) **Local component**: The direct, linear path from the value vector $\mathbf{v}_c^\ell$ to the attention output $\mathbf{a}_p^\ell$, and (2) **Downstream component**: The gradient sensitivity of the final hidden state with respect to that attention output.

To ensure robustness against the noise and directional insta-

[*]Equal contribution . Authors listed in alphabetical order. [1]Department of ECE [2]ASRI [3]IPAI [4]INMC/AIIS, Seoul National University, Seoul, Republic of Korea. Correspondence to: Taesup Moon <tsmoon@snu.ac.kr>.

ICML 2026 Workshop on Resource-Adaptive Foundation Model Inference, Seoul, South Korea.

bility inherent in downstream gradients, KVgrad employs a magnitude-aware scoring scheme and a probe-specific preservation strategy. Together, they result in a scoring function ($S^{\text{KVgrad}}_{\ell,c}$ in Eq. 8) that robustly captures the global influence from the cache entry to the final model output (Figure 1 (b)). Finally, we show that KVgrad strictly generalizes recent state-of-the-art methods like KVzip (Kim et al., 2025) and KVzip+ (Jegou & Jeblick, 2026), which focus solely on the local component while omitting the downstream component.

We extensively evaluate KVgrad on four LLMs using RULER (Hsieh et al., 2024) and LongBench (Bai et al., 2024), demonstrating that it significantly pushes the boundaries of KV cache compression. Specifically, KVgrad outperforms the state-of-the-art KVzip+ by up to 32.4% in near-lossless compression ($< 2\%$), achieving an average maximum compression of $6.42\times$. While this high-fidelity scoring requires an additional backward pass, we show that the cost can be fully amortized via a lightweight surrogate. KVzap (Jegou & Jeblick, 2026) demonstrated that query-agnostic importance scores—despite being defined over the full repeat-stage computation—can be accurately predicted by a small MLP from prefill-stage hidden states alone. Following their training recipe and simply replacing the supervision target with KVgrad scores, we obtain a KVgrad (MLP) surrogate that runs alongside prefill at negligible extra cost. We show that KVgrad (MLP) not only outperforms KVzip+ (MLP), but also surpasses the original KVzip+ scoring itself. This suggests that gradient-based importance provides a better signal for neural surrogates, enabling high-performance compression at negligible computational cost during inference (Figure 1 (c)).

2. Related Works

2.1. KV Cache Compression

KV cache compression spans several orthogonal axes: token eviction (Zhang et al., 2023; Li et al., 2024; Cai et al., 2024; Feng et al., 2024), low-rank projection (Chang et al., 2025), precision (Liu et al., 2024b; Hooper et al., 2024), and head-level reduction during pretraining (Ainslie et al., 2023; Liu et al., 2024a). These axes are largely composable — eviction can be applied on top of a quantized cache (Kim et al., 2025; Ramachandran et al., 2025), for instance — so progress along one axis does not preclude gains along another. We focus on post-hoc *token eviction*, which is complementary to precision and dimension reduction already in use.

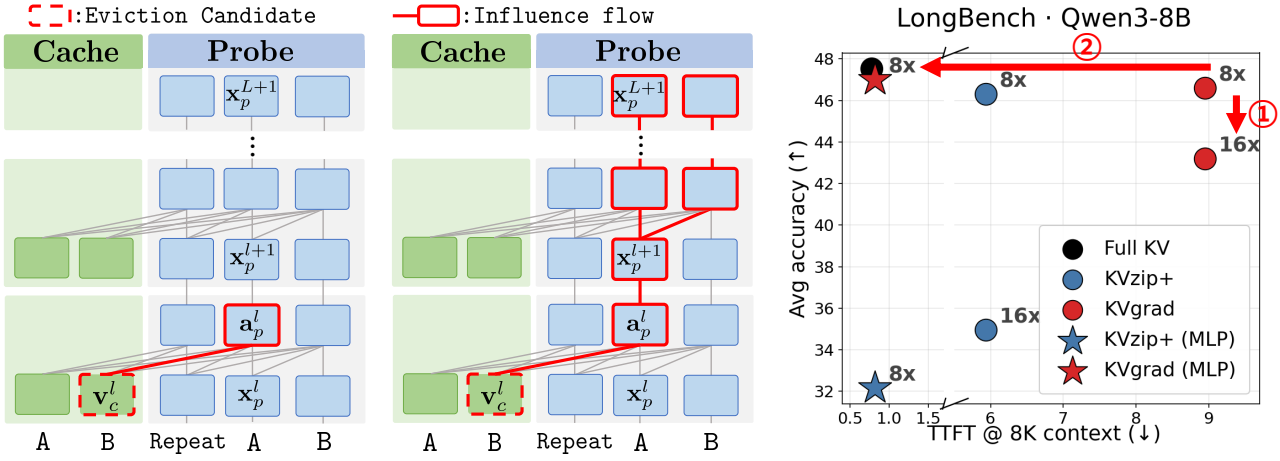
Attention-weight-based eviction Traditional methods score the importance of KV cache pairs by the attention weights they receive. Query-aware methods like H2O (Zhang et al., 2023) and SnapKV (Li et al., 2024) use cumulative or windowed attention to identify "Heavy Hitters." PyramidKV (Cai et al., 2024) further optimizes this by observing a "pyramidal" information funnel, allocating higher budgets to lower layers. However, these methods are primarily query-dependent; re-compressing the cache is necessary for every new query. In contrast, KVzip (Kim et al., 2025) introduces a query-agnostic score by measuring context reconstruction, enabling the same compressed cache to be reused across arbitrary future queries. Recent work (Jegou & Jeblick, 2026) trains a lightweight MLP to predict query-agnostic importance scores directly from internal hidden states. This reduces scoring overhead to a negligible level during generation, showcasing the practical efficiency of the query-agnostic setting for real-world deployment.

Beyond attention weights A line of work (Goel et al., 2025; Jegou & Jeblick, 2026; Gu et al., 2025; Feng et al., 2025) observes that attention weights are a coarse signal for token importance and instead incorporates value-vector information to better approximate the attention output. Specifically, KVzip+ (Jegou & Jeblick, 2026) scales attention weights by value-vector norms to capture relative importance at the layer level. However, these methods—whether query-agnostic or query-aware—remain limited to the local attention output, discarding the information from subsequent FFNs and downstream layers. KVgrad bridges this gap by utilizing a global gradient signal to capture the end-to-end sensitivity of each KV cache entry, accounting for the cumulative impact on the model's final hidden state.

2.2. Attribution Methods for KV Eviction

Attribution methods for quantifying the end-to-end contribution of individual inputs have been extensively developed within the Explainable AI (XAI) community. While perturbation-based attribution methods (Lundberg & Lee, 2017; Petsiuk et al., 2018; Wang et al., 2020) explicitly measure the influence of input eviction on the model response, they incur impractical computational demands. In contrast, back-propagation-based attribution methods (Simonyan et al., 2013; Shrikumar et al., 2016; Bach et al., 2015; Selvaraju et al., 2017) efficiently quantify feature importance in a single step by leveraging the back-propagation process. Such gradient-based methods can be unified under the *Taylor decomposition* (Bach et al., 2015; Montavon et al., 2017), which expresses a model's output as a sum of first-order contributions from individual inputs and motivates the gradient-input product as a local attribution signal.

In practice, however, the raw output gradients of deep neural networks are unstable, producing erratic attributions (Balduzzi et al., 2017; Smilkov et al., 2017; Sundararajan et al., 2017). To mitigate this issue, previous methods (Smilkov et al., 2017; Sundararajan et al., 2017) aggregate gradients across perturbed inputs or integration paths to obtain more stable attributions. However, these techniques require mul-



(a) **Prior work** *(b)* **KVgrad (Ours)** *(c)* **Avg acc vs. TTFT**

Figure 1. **Overview of KVgrad**: **(a) Prior work** uses a **local view**, scoring cache entry $\mathbf{v}_c^l$ solely by its immediate contribution to attention output $\mathbf{a}_p^l$. **(b) KVgrad** adopts a **global view**, quantifying the total influence flow from the entry to the final hidden states **(c) Accuracy vs. TTFT**: ① KVgrad maintains near-lossless accuracy ($< 2\%$) even at high compression ratios (up to $16\times$), remaining stable under regimes where KVzip+ experiences significant accuracy drops. ② The distilled KVgrad (MLP), matches the accuracy of the offline target with zero measurable overhead, whereas the KVzip+ (MLP) suffers a sharp performance drop.

tiple gradient evaluations, which incur an impractical computational cost. A complementary line of work focuses on *positive evidence* (Springenberg et al., 2014; Noh et al., 2015; Bach et al., 2015; Montavon et al., 2017) or *aggregation* (Selvaraju et al., 2017) to obtain more stable explanations, *e.g.*, Grad-CAM (Selvaraju et al., 2017) pools gradients across spatial dimensions to yield more robust saliency. These methods are designed for scalar-output classification, and their extension to KV eviction in long-context LLMs remains largely unexplored. Our proposed methods, *e.g.*, KVgrad, bridge the gap by adapting aggregated gradient-based attribution to yield a token-level eviction criterion that reflects the global influence of each KV cache entry in a single backward pass.

3. Proposed Methods

3.1. Preliminary: Query-agnostic KV eviction

Repeat-reconstruction probes Designing a KV cache eviction strategy requires a clear objective that specifies which information must be preserved. KVzip (Kim et al., 2025) formulates this objective as retaining the signals necessary to faithfully reconstruct the original context from the post-eviction cache. Specifically, we append a sequence of *probe tokens* consisting of a fixed instruction, such as `"Repeat the previous context:"`, followed by a copy of the original context. For example, in Figure 1, the original context tokens A and B are duplicated after the instruction; together they form probe tokens used to score the cache entries corresponding to A and B. In the remainder of the paper, we denote c as a cache token index and $p \in \mathcal{P}$ as a probe token index, where $c \leq p$ by definition.

Notation Consider a Transformer with L layers, in which each layer is indexed by $\ell \in \{1, \ldots, L\}$. For simplicity, we consider a single-head attention setting; the extension to multi-head or grouped-query attention is straightforward, as each head can be treated independently.

Let $\mathbf{x}_p^\ell \in \mathbb{R}^{1\times D}$ denote the hidden state entering layer ℓ at p, $\mathbf{v}_t^\ell \in \mathbb{R}^{1\times D}$ the value vector of token t, $\mathbf{W}_O^\ell \in \mathbb{R}^{D\times D}$ the output projection matrix, and $A_{p,t}^\ell \in [0,1]$ the attention weight from query p to key $t \leq p$, satisfying $\sum_{t=1}^{p} A_{p,t}^\ell = 1$. The attention output at position p in layer ℓ, $\mathbf{a}_p^\ell \in \mathbb{R}^{1\times D}$, is the weighted sum of value vectors projected by $\mathbf{W}_O^\ell$:

$$\mathbf{a}_p^\ell = \sum_{t\leq p} A_{p,t}^\ell \, \mathbf{v}_t^\ell \, \mathbf{W}_O^\ell. \tag{1}$$

The overall update at layer ℓ is then:

$$\mathbf{x}_p^{\ell+1} = \mathbf{x}_p^\ell + \mathbf{a}_p^\ell + \mathrm{FFN}^\ell\left(\mathbf{x}_p^\ell + \mathbf{a}_p^\ell\right), \tag{2}$$

where FFN^ℓ denotes the Feed-Forward Network (FFN) block at layer ℓ.

Importance scoring KVzip (Kim et al., 2025) scores the importance of each cached KV pair by the maximum attention weight it receives from *any* probe token:

$$S_{\ell,c}^{\mathrm{kvzip}} = \max_{p\in\mathcal{P}} A_{p,c}^\ell. \tag{3}$$

Subsequently, KVzip+ (Jegou & Jeblick, 2026) improves upon this by quantifying the contribution of each cached

KV pair to the *attention output* $\mathbf{a}_p^\ell$:

$$S_{\ell,c}^{\text{kvzip+}} = \max_{p\in\mathcal{P}} A_{p,c}^\ell \frac{\left\|\mathbf{v}_c^\ell \mathbf{W}_O^\ell\right\|_2}{\left\|\mathbf{x}_p^\ell\right\|_2}. \quad (4)$$

Here, $A_{p,c}^\ell \left\|\mathbf{v}_c^\ell \mathbf{W}_O^\ell\right\|_2$ reflects the contribution of the c-th KV pair to $\mathbf{a}_p^\ell$, while the factor $1/\left\|\mathbf{x}_p^\ell\right\|_2$ rescales this contribution relative to the magnitude of the incoming hidden state at position p.

Limitations of local scoring. The importance scores in Eqs. (3) and (4) both quantify how a cache entry locally affects the attention output $\mathbf{a}_p^\ell$ at its own layer. However, this local contribution does not necessarily reflect the impact on the final hidden states, since the signal propagates through the remaining $L-\ell$ blocks. Figure 1 illustrates this gap: prior scores terminate at layer ℓ (left), whereas a faithful importance measure must trace the *influence flow* from $\mathbf{v}_c^\ell$ to the final hidden state $\mathbf{x}_p^{L+1}$ (right).

3.2. KVgrad: A Gradient-Based Global Importance Score

A first-order Taylor approximation To address the above-mentioned limitation of local scoring and quantify the global influence, we introduce a scalar objective Φ that summarizes the final representation. Specifically, we define $\Phi = \sum_{p\in\mathcal{P}} \|\mathbf{x}_p^{L+1}\|_2^2$ to represent the cumulative magnitude of the final layer hidden states. Note that this objective depends on the entire set of cached KV pairs and current input representations. With a slight abuse of notation, we write Φ as a function of a single cached value, $\Phi(\mathbf{v}_c^\ell)$, by holding all other cached KV pairs and model parameters fixed. This allows us to approximate the impact of removing $\mathbf{v}_c^\ell$ via a first-order Taylor approximation around its current value:

$$\Phi(\mathbf{v}_c^\ell + \boldsymbol{\delta}) \approx \Phi(\mathbf{v}_c^\ell) + \boldsymbol{\delta}\cdot\nabla_{\mathbf{v}_c^\ell}\Phi. \quad (5)$$

Now, setting $\boldsymbol{\delta} = -\mathbf{v}_c^\ell$ (i.e., complete removal) gives the first-order importance score $\mathbf{v}_c^\ell \cdot \nabla_{\mathbf{v}_c^\ell}\Phi$ as a proxy for the change in Φ.

To disentangle the local influence within the attention layer from global propagation through the model, we apply the chain rule through the attention outputs $\{\mathbf{a}_p^\ell : p\in\mathcal{P}\}$ as follows:

$$\begin{aligned}\Phi(\mathbf{v}_c^\ell) - \Phi(\mathbf{0}) &\approx \mathbf{v}_c^\ell \cdot \nabla_{\mathbf{v}_c^\ell}\Phi \\ &= \mathbf{v}_c^\ell \cdot \Big(\sum_{p\in\mathcal{P}} \nabla_{\mathbf{v}_c^\ell}\mathbf{a}_p^\ell \cdot \nabla_{\mathbf{a}_p^\ell}\Phi\Big) \\ &= \sum_{p\in\mathcal{P}} \underbrace{A_{p,c}^\ell \left(\mathbf{v}_c^\ell \mathbf{W}_O^\ell\right)\cdot\left(\nabla_{\mathbf{a}_p^\ell}\Phi\right)}_{T_{p,c}^\ell}, \quad (6)\end{aligned}$$

in which Eq. (6) follows from the linear dependency of the attention output on the value vectors, *i.e.,* $\nabla_{\mathbf{v}_c^\ell}\mathbf{a}_p^\ell = A_{p,c}^\ell \mathbf{W}_O^\ell$. Note that the scalar attention weight $A_{p,c}^\ell$ and $\mathbf{v}_c^\ell$ have been commuted for notational clarity. In this decomposition, the *local component* $A_{p,c}^\ell \mathbf{v}_c^\ell \mathbf{W}_O^\ell$ captures the direct linear contribution of the value vector to the attention output at position p. In contrast, the *downstream component* $\nabla_{\mathbf{a}_p^\ell}\Phi$ accounts for how this influence propagates through subsequent non-linear layers to affect the final objective Φ.

From the first-order approximation to KVgrad score. While Eq. (6) provides a principled derivation, it is not directly suitable as a robust importance score. We identify two key limitations in the vanilla first-order approximation and propose modifications to address them.

(i) Magnitude-aware scoring. While the first-order approximation $T_{p,c}^\ell$ is exact for the linear local component $A_{p,c}^\ell \mathbf{v}_c^\ell \mathbf{W}_O^\ell$, its reliability diminishes when combined with the non-linear downstream term $\nabla_{\mathbf{a}_p^\ell}\Phi$. In particular, previous studies have highlighted that vanilla gradients of deep neural networks are often noisy (Smilkov et al., 2017; Balduzzi et al., 2017) and the gradient direction can easily be changed with small perturbations (Ghorbani et al., 2019), failing to provide a faithful measure of importance in deep networks. To mitigate such distortions, various approaches, such as the LRP z^+-rule (Bach et al., 2015) and Grad-CAM (Selvaraju et al., 2017), have attempted to focus on positive signals or aggregate gradient information to improve attribution stability.

To jointly compensate for the noise and directional uncertainty of gradients, we adopt a *magnitude-aware scoring* scheme based on the Cauchy-Schwarz upper bound of $T_{p,c}^\ell$:

$$|T_{p,c}^\ell| \le A_{p,c}^\ell \left\|\mathbf{v}_c^\ell \mathbf{W}_O^\ell\right\|_2 \left\|\nabla_{\mathbf{a}_p^\ell}\Phi\right\|_2, \quad (7)$$

in which such relaxation effectively discards the influence of unreliable directional information while preserving the magnitude, which complements the limitations of the first-order approximation.

(ii) Probe-specific preservation. The bound in Eq. (7) resolves directional cancellation *within* a single probe but does not address *cross-probe* dilution. In other words, summing over $\mathcal{P}$ treats all probes equally. Under this inductive bias, a KV entry that is critically important for a specific probe—yet irrelevant to others— may be erroneously discarded as its high local contribution is washed out by the majority of negligible signals.

To ensure the score retains these sparse but vital dependencies, we replace $\sum_{p\in\mathcal{P}}$ with $\max_{p\in\mathcal{P}}$ operator. Combining it with magnitude-aware scoring, we define the final **KVgrad** score:

$$S_{\ell,c}^{\text{KVgrad}} = \max_{p\in\mathcal{P}} A_{p,c}^\ell \left\|\mathbf{v}_c^\ell \mathbf{W}_O^\ell\right\|_2 \left\|\nabla_{\mathbf{a}_p^\ell}\Phi\right\|_2. \quad (8)$$

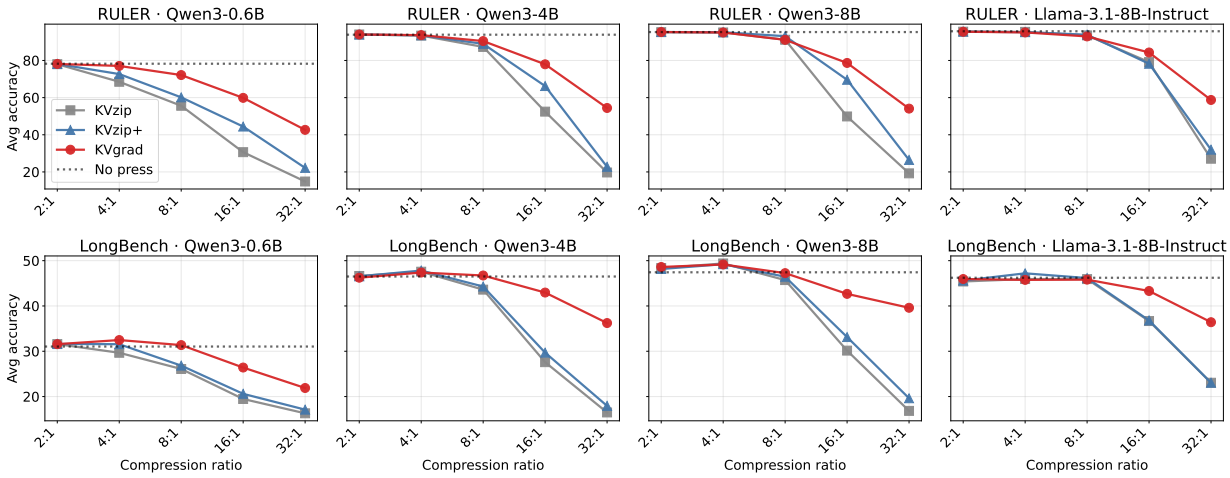


Figure 2. Average accuracy vs. compression ratio on diverse model scales and families

Our ablations in Sec. 4.4 empirically validate that the adopted magnitude-aware scoring and probe-specific preservation are crucial for maintaining performance in long-context retrieval tasks. Further justifications of the introduced MAS and PSP are in Appendix A.1.

Remarks 1 (Relation to prior work): While KVzip and KVzip+ incorporate $A^{\ell}_{p,c}$ and $\mathbf{v}^{\ell}_{c}\mathbf{W}^{\ell}_{O}$ as local contribution, respectively, both omit the third factor $\nabla_{\mathbf{a}^{\ell}_{p}}\Phi$. This omission motivates us to propose a more comprehensive metric that explicitly incorporates this downstream sensitivity.

4. Experimental Results

4.1. Setup

Models and datasets To assess query-agnostic compression performance across model scales and families, we evaluate KVgrad on three sizes of Qwen3 (Yang et al., 2025) (0.6B, 4B, 8B) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). Our evaluation is done on RULER (Hsieh et al., 2024), comprising 13 synthetic retrieval and tracing subtasks, and LongBench (Bai et al., 2024), consisting of 16 English and code tasks spanning QA and summarization. Detailed information regarding categories, dataset statistics, and evaluation metrics is provided in Appendix A.2.1.

Baselines and implementation We compare against the two strongest query-agnostic baselines: KVzip (Kim et al., 2025) and its refined variant KVzip+ (Jegou & Jeblick, 2026). Both have been shown to outperform a broad set of query-aware methods — including H2O (Zhang et al., 2023), SnapKV (Li et al., 2024), PyramidKV (Cai et al., 2024), and DuoAttention (Xiao et al., 2024) — in their respective papers; we therefore restrict our direct comparison to these two methods to maintain a focused analysis. Further details are provided in Appendix A.2.3.

Evaluation We follow the query-agnostic KV cache eviction and evaluation procedure of KVzip (Kim et al., 2025), which consists of four stages: 1) Prefill 2) Importance scoring 3) Compression 4) Generaiton. Please check details for each stage at A.2.3

Following (Kim et al., 2025), we adopt a chunking strategy for importance scoring: the context is divided into fixed-size chunks of 512 tokens and processed iteratively. The model is prompted to reconstruct the current segment based on a prefix of the preceding chunk. The exact prompt template for chunking strategy is provided in Appendix A.2.4.

Rather than allocating a fixed budget per layer or per head, we rank all KV entries jointly across layers and heads and evict the lowest-scoring ones until the target CR is met. Notably, we find this strategy effective in practice, which appears to stem from our output-gradient-based method capturing globally calibrated importance.

4.2. Main Results

Figure 2 presents the main results across four models (columns) and two datasets (rows). Each subplot shows the averaged accuracy over sub-tasks from RULER or LongBench under varying compression ratios ($2\times, 4\times, 8\times, 16\times, 32\times$). The results show that **KVgrad outperforms both strong baselines**, with the performance gap widening as the compression ratio increases.

4.3. Distilled Surrogate and Compression Cost

Overview of KVgrad (MLP) KVzap (Jegou & Jeblick, 2026) showed that the KVzip+ importance score $S^{\text{KVzip+}}_{\ell,c}$ can be predicted from the corresponding hidden state $\mathbf{x}^{\ell}_{c}$ using a small per-layer MLP, letting scoring run alongside prefill at negligible extra cost. This distillation is only fea-

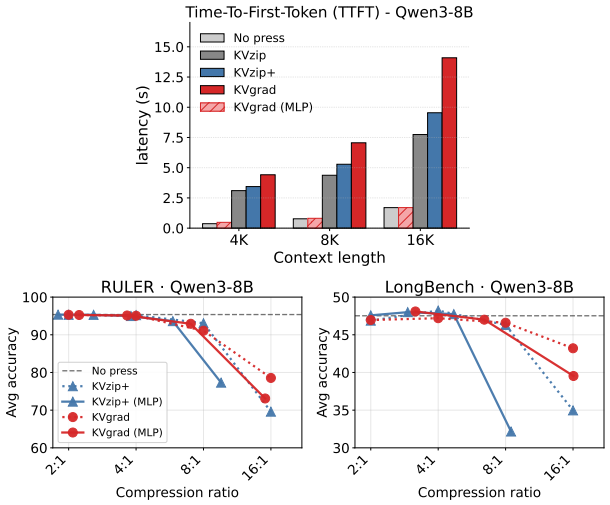


Figure 3. Qwen3-8B: (**Above**) TTFT across context lengths. Note that the TTFT of KVzip+ (MLP) and KVgrad (MLP) is identical, as they share the same MLP architecture. (**Below**) Accuracy vs. compression ratio on RULER and LongBench.

sible for query-agnostic scores: training the MLP requires score targets computed independently of any query, which query-aware methods cannot supply.

Following KVzap's training recipe verbatim and swapping only the supervision target, we train an MLP to predict the KVgrad score $S^{\text{KVgrad}}_{\ell,c}$, which we denote KVgrad (MLP); analogously, we re-train the same architecture to predict KVzip+'s score, denoted KVzip+ (MLP). Each per-layer surrogate is a two-layer MLP trained on precomputed (hidden state, score) pairs over a fixed set of contexts, making it lightweight to train. KVzap's implementation does not enforce powers-of-2 CRs exactly, so the CRs achieved by the MLP variants differ slightly from KVzip+ and KVgrad. Full settings and extended results are in Appendix A.3.

Benchmark Results Figure 3 (Left) reports Qwen3-8B accuracy on RULER and LongBench. We highlight two findings: (i) **KVgrad scores distill more cleanly than KVzip+ scores** Under an identical training pipeline, the gap from the full method to the MLP variant is consistently smaller for KVgrad than for KVzip+, suggesting gradient-based scores are easier to predict from hidden states. (ii) **KVgrad (MLP) approaches the offline KVzip+ at a fraction of the cost** It outperforms KVzip+ (MLP) at every compression ratio and nearly matches the much more expensive offline KVzip+ at moderate CRs, diverging only at the most aggressive ratios.

Compression cost We report Time-To-First-Token (TTFT) — the wall-clock time from receiving the prompt to emitting the first generated token — which covers the prefill and importance-scoring stages. Figure 3 (Right) shows TTFT on Qwen3-8B at context lengths of 4K, 8K, and 16K. Full KVgrad requires an additional backward pass for importance scoring; this overhead can be considered as a reasonable cost in offline settings, where it pays for the better performance. KVgrad (MLP) eliminates this overhead: scoring runs in a single forward pass alongside prefill, so TTFT matches the no-compression baseline at all measured context lengths.

Table 1. Score-variant ablation study.

Variant	PSP	MAS	8:1	16:1	32:1
KVgrad	✓	✓	90.44	**77.94**	**54.42**
w/o MAS	✓	✗	**91.05**	75.43	31.54
w/o PSP	✗	✓	82.63	63.82	29.63
w/o Both	✗	✗	88.48	61.85	23.92

Together, the two variants make KVgrad practical across deployment regimes: full KVgrad delivers the strongest offline compression, and KVgrad (MLP) brings competitive scoring quality to online inference at no measurable overhead.

4.4. Ablation Studies

We conduct ablation studies to evaluate the efficacy of Magnitude-Aware Scoring (MAS), Probe-Specific Preservation (PSP), and the impact of probe token chunk size. All experiments are performed using Qwen3-4B on RULER.

Magnitude-aware scoring and probe-specific preservation. Table 1 describes the individual contributions of MAS and PSP, which constitute the transition from the Taylor approximation form in Eq. (6) to our proposed KVgrad Eq. (8). Our results indicate that omitting the MAS or substituting the $\max$ operator with $\sum$ leads to a severe degradation in performance. Notably, this performance gap widens significantly as the compression ratio increases.

5. Conclusion

We introduce KVgrad, a query-agnostic KV cache eviction framework that evaluates the importance of cache entries through their global impact on a model's final representation. By factorizing importance into a direct local component and a downstream gradient sensitivity term, KVgrad captures the global sensitivity of the final model output to each cached entry across the entire network depth. Our evaluations on RULER and LongBench demonstrate that KVgrad achieves up to 6.42× compression with negligible performance loss, consistently outperforming existing query-agnostic baselines. Furthermore, we show that our gradient-based signal is highly amenable to distillation, enabling a lightweight MLP surrogate to provide on-the-fly compression with no measurable overhead.

References

Ainslie, J., Lee-Thorp, J., De Jong, M., Zemlyanskiy, Y., Lebrón, F., and Sanghai, S. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4895–4901, 2023.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Bai, Y., Lv, X., Zhang, J., Lyu, H., Tang, J., Huang, Z., Du, Z., Liu, X., Zeng, A., Hou, L., et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 2024.

Balduzzi, D., Frean, M., Leary, L., Lewis, J., Ma, K. W.-D., and McWilliams, B. The shattered gradients problem: If resnets are the answer, then what is the question? In *International conference on machine learning*, pp. 342–350. PMLR, 2017.

Basant, A., Khairnar, A., Paithankar, A., Khattar, A., Renduchintala, A., Malte, A., Bercovich, A., Hazare, A., Rico, A., Ficek, A., et al. Nvidia nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model. *arXiv preprint arXiv:2508.14444*, 2025.

Cai, Z., Zhang, Y., Gao, B., Liu, Y., Li, Y., Liu, T., Lu, K., Xiong, W., Dong, Y., Hu, J., et al. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

Chang, C.-C., Lin, W.-C., Lin, C.-Y., Chen, C.-Y., Hu, Y.-F., Wang, P.-S., Huang, N.-C., Ceze, L., Abdelfattah, M. S., and Wu, K.-C. Palu: Kv-cache compression with low-rank projection. In *The Thirteenth International Conference on Learning Representations*, 2025.

Devoto, A., Jeblick, M., and Jégou, S. Expected attention: Kv cache compression by estimating attention from future queries distribution. *arXiv preprint arXiv:2510.00636*, 2025. URL https://arxiv.org/abs/2510.00636.

Feng, Y., Lv, J., Cao, Y., Xie, X., and Zhou, S. K. Ada-kv: Optimizing kv cache eviction by adaptive budget allocation for efficient llm inference. *arXiv preprint arXiv:2407.11550*, 2024.

Feng, Y., Lv, J., Cao, Y., Xie, X., and Zhou, S. K. Identify critical kv cache in llm inference from an output perturbation perspective. *arXiv preprint arXiv:2502.03805*, 2025.

Ghorbani, A., Abid, A., and Zou, J. Interpretation of neural networks is fragile. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33013681. URL https://doi.org/10.1609/aaai.v33i01.33013681.

Goel, R., Park, J., Gagrani, M., Jones, D., Morse, M., Langston, H., Lee, M., and Lott, C. Caote: Kv cache selection for llms via attention output error-based token eviction. *arXiv preprint arXiv:2504.14051*, 2025.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gu, Y., Liang, X., Zhao, J., and Diao, E. Obcache: Optimal brain kv cache pruning for efficient long-context llm inference. *arXiv preprint arXiv:2510.07651*, 2025.

Hooper, C., Kim, S., Mohammadzadeh, H., Mahoney, M. W., Shao, Y. S., Keutzer, K., and Gholami, A. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303, 2024.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., Zhang, Y., and Ginsburg, B. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Jegou, S. and Jeblick, M. Kvzap: Fast, adaptive, and faithful kv cache pruning. *arXiv preprint arXiv:2601.07891*, 2026.

Jiang, B., Yang, T., Liu, Y., He, X., Di, S., and Jin, S. Packkv: Reducing kv cache memory footprint through llm-aware lossy compression. *arXiv preprint arXiv:2512.24449*, 2025.

Kim, J.-H., Kim, J., Kwon, S., Lee, J. W., Yun, S., and Song, H. O. Kvzip: Query-agnostic kv cache compression with context reconstruction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Łańcucki, A., Staniszewski, K., Nawrot, P., and Ponti, E. M. Inference-time hyper-scaling with kv cache compression. *arXiv preprint arXiv:2506.05345*, 2025.

Li, Y., Huang, Y., Yang, B., Venkitesh, B., Locatelli, A., Ye, H., Cai, T., Lewis, P., and Chen, D. Snapkv: Llm knows what you are looking for before generation. *Advances*

in Neural Information Processing Systems, 37:22947–22970, 2024.

Liu, A., Feng, B., Wang, B., Wang, B., Liu, B., Zhao, C., Dengr, C., Ruan, C., Dai, D., Guo, D., et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024a.

Liu, Z., Yuan, J., Jin, H., Zhong, S., Xu, Z., Braverman, V., Chen, B., and Hu, X. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*, 2024b.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K.-R. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017.

Noh, H., Hong, S., and Han, B. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pp. 1520–1528, 2015.

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

Pope, R., Douglas, S., Chowdhery, A., Devlin, J., Bradbury, J., Heek, J., Xiao, K., Agrawal, S., and Dean, J. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

Ramachandran, A., Neseem, M., Sakr, C., Venkatesan, R., Khailany, B., and Krishna, T. Thinkv: Thought-adaptive kv cache compression for efficient reasoning models. *arXiv preprint arXiv:2510.01290*, 2025.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Shah, J., Bikshandi, G., Zhang, Y., Thakkar, V., Ramani, P., and Dao, T. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, 37:68658–68685, 2024.

Shazeer, N. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., and Hu, X. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 24–25, 2020.

Xiao, G., Tang, J., Zuo, J., Guo, J., Yang, S., Tang, H., Fu, Y., and Han, S. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. *arXiv preprint arXiv:2410.10819*, 2024.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhang, Z., Sheng, Y., Zhou, T., Chen, T., Zheng, L., Cai, R., Song, Z., Tian, Y., Ré, C., Barrett, C., et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

A. Supplementary Materials

A.1. Justifications of proposed methods

Remarks 2 (Connection to Grad-CAM): Grad-CAM (Selvaraju et al., 2017) scores importance by multiplying a spatially aggregated gradient weight into its activation map. In our context, weighting $\mathbf{a}_p^\ell$ with its aggregated gradient and substituting with Eq. (1) yield $\mathbf{a}_p^\ell \left\| \nabla_{\mathbf{a}_p^\ell} \Phi \right\|_2 = \sum_{c \le p} A_{p,c}^\ell \, \mathbf{v}_c^\ell \, \mathbf{W}_O^\ell \left\| \nabla_{\mathbf{a}_p^\ell} \Phi \right\|_2$. The ℓ_2 norm of the summand corresponds to our KVgrad score $S_{\ell,c}^{\mathrm{KVgrad}}$ (before $\max$), effectively decomposing the Grad-CAM-style importance into fine-grained contributions of individual cache entries c.

Remarks 3 (Max operator): KVzip is designed to preserve entries critical to at least one output token, which corresponds to $\max_{p \in \mathcal{P}} \Phi_p$ in our formulation, where $\Phi_p := \left\| \mathbf{x}_p^{L+1} \right\|_2^2$ is the per-probe target. To avoid per-probe backward pass, KVgrad evaluates importance using the gradient collapsed at the intermediate activation $\mathbf{a}_p^\ell$ during a single pass on $\Phi = \sum_p \Phi_p$, providing a position-localized proxy for per-probe importance.

A.2. Detailed Experiment Settings

A.2.1. Benchmark specification

Table 2. Summary of benchmarks used for text and multimodal evaluation

Benchmark	Modality	Samples	Key Metrics	Context Range (Avg)
RULER (Hsieh et al., 2024)	Text	6,500	Recall, EM	4,096 tokens
LongBench (Bai et al., 2024)	Text	3,750	F1, ROUGE-L, EM	5k $\sim$ 15k tokens

† Mean vision tokens per instance under our 256-frame sampling configuration.

RULER Comprises 13 synthetic sub-tasks across four categories: Retrieval (Single/Multi-query/Multi-key Needle-in-a-Haystack), Multi-hop Tracing (variable tracking), Aggregation (unique Words, frequent words), and Question Answering (Single/Multi-hop Reasoning). Following previous work (Kim et al., 2025; Jegou & Jeblick, 2026) for establishing baseline long-context robustness, we utilize the 4096-token context length configuration for all tasks. Each task has 500 samples each, which in total has 6500 samples. Evaluation is based on Recall (for retrieval tasks) and Exact Match (for tracking and QA).

LongBench While the full LongBench suite includes bilingual tasks, we evaluate on the 16 English and code tasks following the selection in KVPress (Devoto et al., 2025). The subset comprises 3,750 samples (150$\sim$500 per task) spanning Single-doc QA, Multi-doc QA, Summarization, Few-shot Learning, Synthetic Tasks, and Code Completion. Each task has 150 $\sim$ 500 samples per task, which in total has 3,750 samples. Context lengths range from 1k to 18k words, with most tasks centered between 5k and 15k tokens. Evaluation metrics include F1-score (for QA), ROUGE-L (for summarization), and Exact Match or Edit Similarity (for code and synthetic tasks).

A.2.2. Hardware and Software Environment

We evaluate KVgrad across two primary compute environments. The larger models—including Qwen3-4B, Qwen3-8B, Llama-3.1-8B-Instruct—were evaluated on NVIDIA A100 GPU. The Qwen3-0.6B was evaluated on NVIDIA RTX A5000 GPU. Both environments were standardized on PyTorch 2.6.0 and CUDA 12.4 to ensure consistent performance and KV cache handling.

Table 3. Hardware and software specifications

Component	NVIDIA A100 (80GB)	NVIDIA RTX A5000 (24GB)
CPU	AMD EPYC 74F3	AMD Threadripper PRO 3975WX
System RAM	2.0 TiB	256 GiB
Operating System	Ubuntu 24.04 LTS	Ubuntu 22.04 LTS
PyTorch / CUDA	2.6.0 / 12.4	2.6.0 / 12.4

A.2.3. Implementation details

In this section, we provide additional technical details regarding the implementation of KVgrad and the configuration of our evaluation frameworks across text, image, and video modalities.

Detailed description of quary-agnostic KV cache eviction stages

- *Prefill*: constructing the KV cache from a shared context
- *Importance scoring*: computing the importance score $S_{\ell,c}^{\text{KVgrad}}$ for each KV entry without access to any query
- *Compression*: evicting KV entries with the lowest importance scores until reaching a target compression ratio (CR), defined as the ratio between the original and compressed cache sizes
- *Generation*: generating outputs using the compressed KV cache

Text Tasks For all text-centric evaluations (RULER and LongBench), we utilize KVpress (Devoto et al., 2025), an open-source library designed for the modular evaluation of KV-cache eviction policies. We implemented KVgrad as a new eviction policy within this framework, enabling a direct and fair comparison against the built-in implementations of KVzip (Kim et al., 2025) and KVzip+ (Jegou & Jeblick, 2026).

A.2.4. Chunking-base compression

As described in Section 4.1, our importance scoring utilizes a chunk-based reconstruction strategy. This approach ensures that the KV cache is scored in a query-agnostic manner by forcing the model to attend to and "reconstruct" the context. For the Qwen3 model family, we utilize the specific chat-template format shown below:

Prompt for chunk-based reconstruction (Qwen3 families)

```
{context}

Repeat the previous context exactly, starting with {previous chunk[-8:]} <|im_end|>
<|im_start|>assistant
<think>

</think>

{chunk}
```

A.3. Distillation Details for KVgrad (MLP)

This section provides training and inference details for the KVgrad (MLP) introduced in §4.3. In short, we adopt the KVzap training and evaluation protocol of Jegou & Jeblick (2026) verbatim[1], with the only change being the regression target: the data, features, architecture, optimizer, and hyperparameters are identical, and only the supervision signal is swapped from KVzip+ to KVgrad. The sole inference-time deviation is the threshold range, which is re-calibrated because KVzip+ and KVgrad produce scores on different log-scales. Results are reported for both the new KVgrad (MLP) and the recipe's original KVzip+ (MLP); because the two MLP variants differ only in supervision signal, the gap between them isolates the contribution of the regression target from that of the distillation recipe itself.

A.3.1. Method

Computing either score — $S_{\ell,c}^{\text{kvzip+}}$ in Eq. (4) or $S_{\ell,c}^{\text{kvgrad}}$ in Eq. (8) — requires multi-pass forward passes over the probe context (and, for KVgrad, an additional backward pass), which dominates time-to-first-token (Figure 3). Following Jegou & Jeblick (2026), we replace this multi-pass scorer with a per-layer model f_θ^ℓ that predicts the score from the cache-token

[1]Reference implementation: https://github.com/NVIDIA/kvpress/tree/main/kvzap.

hidden state in a single forward pass:

$$\hat{S}_{\ell,c} \;=\; f_\theta^\ell(\mathbf{x}_c^\ell), \qquad f_\theta^\ell : \mathbb{R}^{1\times D} \to \mathbb{R}. \tag{9}$$

At inference, $\hat{S}_{\ell,c}$ feeds the same DMS-style (Łańcucki et al., 2025) inference rule used by KVzap: cache entries with $\hat{S}_{\ell,c} < \tau$ are evicted independently per (layer, KV-head), and a sliding window of the most recent 128 tokens is excluded from eviction.

A.3.2. Implementation details

Training data We use the `Nemotron-Pretraining-Dataset-sample` (Basant et al., 2025), the same pretraining mixture as Jegou & Jeblick (2026), drawing from all nine sub-corpora (math, code, factual QA, etc.). Prompts whose tokenized length lies outside $[750, 1250]$ are filtered, leaving $\approx$2,367 prompts balanced as 500 train + 5 test per sub-corpus. Every prompt is wrapped with the model's chat template so captured hidden states match the evaluation distribution; we sub-sample 500 token positions per prompt with a fixed seed.

Training pairs $(\mathbf{x}_c^\ell,\, S_{\ell,c})$ For each prompt and each layer ℓ, the input feature is the hidden state $\mathbf{x}_c^\ell$ at the sampled cache position c, paired with the score $S_{\ell,c}$ at the same (ℓ, c). Both KVzip+ and KVgrad targets are computed under the repeat-prompt protocol in Section 4.1; KVgrad additionally requires one backward pass through the probe context to obtain the gradient term $\nabla_{\mathbf{a}_p^\ell}\Phi$ in Eq. (8). Targets are log-transformed (clamped for numerical stability) to compress their dynamic range.

Model and optimization For each layer ℓ, f_θ^ℓ is a two-layer MLP, $\mathbb{R}^{1\times D} \xrightarrow{\text{Linear}} \mathbb{R}^{1\times 512} \xrightarrow{\text{GELU}} \mathbb{R}^{1\times 512} \xrightarrow{\text{Linear}} \mathbb{R}$, producing the scalar score $\hat{S}_{\ell,c}$. For multi-head or grouped-query attention, we extend the final linear to match the number of KV heads, so that the MLP predicts all per-KV-head scores at layer ℓ jointly in a single forward pass. Following Jegou & Jeblick (2026), we optimize with AdamW (learning rate 5×10^{-3}, cosine annealing over 15 epochs, batch size 512, gradient-norm clipping at 1.0, MSE loss).

Evaluation We follow the main-paper text protocol (Section. 4.1): RULER 4096 (Hsieh et al., 2024) and LongBench v1 (Bai et al., 2024). Owing to compute constraints, this supplementary's comparison covers four scoring configurations — KVzip+, KVzip+ (MLP), KVgrad, and KVgrad (MLP) — alongside the no-compression baseline. Because threshold pruning yields a content-dependent compression ratio, we sweep six thresholds per (model, target) and linearly interpolate the achieved-ratio→accuracy curve to the standard ratios $\{2{:}1, 4{:}1, 8{:}1, 16{:}1\}$. Threshold ranges differ between targets because KVzip+ and KVgrad produce scores on different log-scales: $\tau \in [-8, -3]$ (KVzip+, 0.6B), $[-6, -2]$ (KVzip+, 4B/8B), $[7, 12]$ (KVgrad, 0.6B), and $[8, 14]$ (KVgrad, 4B/8B).

A.4. Limitations and Broader Impact

This work presents a method for improving the algorithmic efficiency of KVgrad. While our contributions are primarily foundational, they carry several societal implications and limitations.

A.4.1. Positive Societal Impacts

Environmental Sustainability By reducing the computational overhead required for inference, our approach contributes to a lower carbon footprint for AI systems. This is increasingly critical as the energy demands of large-scale model deployment continue to rise.

Democratization of AI Lowering the hardware requirements for high-performance models helps democratize AI research. By making these methods accessible to researchers and organizations with limited GPU resources, we foster a more inclusive research environment and reduce the "compute gap" between well-funded labs and the broader community.

A.4.2. Potential Negative Impacts and Limitations

Generalization and Reliability While our method demonstrates significant improvements on specific accuracy-oriented benchmarks, its evaluation is currently focused on a targeted subset of tasks. There is a risk that these efficiency gains may not inherently mitigate—and could potentially mask—underlying hallucinations in more open-ended or out-of-distribution

Table 4. RULER (fr=0.1) string-match mean across 11 attention-importance variants, on Qwen3-0.6B and Qwen3-4B. The $\sum$-family and max-family rows pair up by suffix formula. **Bold** = best per (model, cr) among the 10 ablation variants (GxI excluded). max-family $\|\psi\|_2\|\phi\|_2$ is the final method.

Model	Family	Variant	2:1	4:1	8:1	16:1	32:1
Qwen3-0.6B	*ref*	$\left\|\sum_m \langle \psi, \phi \rangle\right\|$	78.29	76.24	72.40	53.70	26.12
	Σ	$\sum_m \left\|\langle \psi, \phi \rangle\right\|$	77.79	76.75	**73.15**	54.51	24.37
		$\sum_m \left\|\langle \|\psi\|, \phi \rangle\right\|$	77.85	77.08	67.41	45.62	25.36
		$\sum_m \left\|\langle \psi, \|\phi\| \rangle\right\|$	77.92	76.96	70.54	50.44	29.96
		$\sum_m \langle \|\psi\|, \|\phi\| \rangle$	77.71	76.91	71.97	**57.35**	37.02
		$\sum_m \|\psi\|_2 \|\phi\|_2$	77.75	76.78	72.33	55.96	36.32
	max	$\max_m \left\|\langle \psi, \phi \rangle\right\|$	78.13	76.56	72.24	55.83	24.45
		$\max_m \left\|\langle \|\psi\|, \phi \rangle\right\|$	78.14	76.79	71.14	53.04	31.30
		$\max_m \left\|\langle \psi, \|\phi\| \rangle\right\|$	77.57	**77.20**	72.59	56.14	35.33
		$\max_m \langle \|\psi\|, \|\phi\| \rangle$	78.05	76.74	70.48	57.32	38.76
		$\max_m \|\psi\|_2 \|\phi\|_2$	**78.19**	76.60	69.66	57.21	**40.20**
Qwen3-4B	*ref*	$\left\|\sum_m \langle \psi, \phi \rangle\right\|$	94.57	93.91	91.60	70.18	33.15
	Σ	$\sum_m \left\|\langle \psi, \phi \rangle\right\|$	**94.52**	93.91	92.36	67.34	23.40
		$\sum_m \left\|\langle \|\psi\|, \phi \rangle\right\|$	94.45	93.87	85.81	58.93	26.63
		$\sum_m \left\|\langle \psi, \|\phi\| \rangle\right\|$	94.45	94.05	84.80	57.35	25.61
		$\sum_m \langle \|\psi\|, \|\phi\| \rangle$	94.28	94.08	84.70	67.09	31.30
		$\sum_m \|\psi\|_2 \|\phi\|_2$	94.45	94.13	84.23	67.23	29.77
	max	$\max_m \left\|\langle \psi, \phi \rangle\right\|$	**94.52**	93.94	91.71	76.40	32.22
		$\max_m \left\|\langle \|\psi\|, \phi \rangle\right\|$	94.47	94.06	**92.62**	76.67	39.56
		$\max_m \left\|\langle \psi, \|\phi\| \rangle\right\|$	94.50	94.26	91.49	76.28	40.70
		$\max_m \langle \|\psi\|, \|\phi\| \rangle$	94.19	94.26	91.88	**78.92**	55.02
		$\max_m \|\psi\|_2 \|\phi\|_2$	94.51	**94.40**	92.44	78.17	**55.03**

scenarios. Furthermore, as with any efficiency-focused optimization, there is a possibility of unintended performance trade-offs in edge cases not captured by our current evaluation suite. We encourage practitioners to perform comprehensive audits for truthfulness before deploying this method in safety-critical or user-facing applications.

A.4.3. Limitations and Future Work

Despite the novelty of the proposed Taylor-based compression algorithm and its strong empirical performance, several limitations remain to be addressed in future research.

Unified Importance Formulation Our current Taylor decomposition estimates the importance of values exclusively, whereas the eviction process is applied to both keys and values. While this formulation is currently supplemented by a heuristic that preserves a fixed-length attention sink, it lacks a unified framework that accounts for the mutual importance of the key-value pair. Developing a rigorous formulation that integrates both components remains a primary objective for future work.

Contextual Redundancy The Taylor-based approach focuses on individual token importance but does not explicitly account for information redundancy within the context. This can lead to suboptimal compression where important but redundant information is retained. Addressing this, particularly for video datasets containing temporally redundant frames, could significantly enhance the compression ratio.

Implementation Efficiency Our current implementation involves redundant computations of the attention matrix. Transitioning to a specialized kernel-level implementation would drastically improve throughput. Future iterations should explore deep-level optimizations, such as intervening in the backpropagation process and integration with high-performance inference frameworks like vLLM.

Evaluation Scope in Multimodality While we adhered to standard Large Language Model (LLM) benchmarks, our evaluation on Multimodal Large Language Models (MLLMs) remains preliminary. Although we have demonstrated the

Table 5. Chunk-size ablation of KVgrad.

Chunk size	2:1	4:1	8:1	16:1	32:1
256	**94.02**	**93.70**	**90.59**	77.53	52.79
512	93.97	93.64	90.44	77.94	54.42
1024	93.95	93.45	90.25	**78.33**	**55.15**
2048	93.99	93.38	90.33	77.77	54.34

Table 6. Aggregator ablation on the gradient ϕ, RULER (fr=0.1) string-match mean. The score is $\max_m \alpha_{m,c} \cdot \|\psi\|_2 \cdot |\mathrm{Aggr}(\phi_m)|$; the ψ-side aggregator is fixed at $\|\cdot\|_2$ and only $\mathrm{Aggr}(\phi)$ is varied. **Bold** = best per (model, cr).

Model	$\mathrm{Aggr}(\phi)$	2:1	4:1	8:1	16:1	32:1
Qwen3-0.6B	$\|\phi\|_2$ *(default)*	**78.19**	76.60	69.66	**57.21**	40.20
	$\|\phi\|_1$	78.05	**76.70**	**70.83**	57.18	38.93
	$\|\phi\|_\infty$	78.03	75.84	68.61	53.87	**40.42**
	$\overline{\phi}$ (mean, signed)	77.65	74.84	66.58	46.28	24.62
Qwen3-4B	$\|\phi\|_2$ *(default)*	94.51	94.40	**92.44**	78.17	55.03
	$\|\phi\|_1$	94.55	94.42	91.89	**78.36**	**56.17**
	$\|\phi\|_\infty$	**94.64**	**94.44**	91.75	69.68	32.59
	$\overline{\phi}$ (mean, signed)	94.16	94.04	91.93	69.52	35.40

feasibility of our method across different modalities, more extensive benchmarking on diverse image and video datasets, including the comparison with the SOTA baseline methods, is required to fully validate its generalizability.

Future works Several promising directions for future research emerge from this study. By integrating our method with KV-reuse settings, we can investigate hybrid strategies for KV compression, offloading, and re-entry. This includes handling KV pairs from independent sources that may contain causally independent compressed data. Furthermore, applying Explainable AI (XAI) techniques to compute end-to-end importance scores represents a compelling avenue for achieving more interpretable and efficient context management.

A.5. Additional Ablation Studies

A.5.1. Robust to chunk size

Table 5 summarizes the sensitivity analysis of KVgrad regarding chunk size. The results demonstrate that KVgrad maintains stable performance across a wide range of variations, demonstrating high robustness to this hyperparameter. Consequently, we utilized a consistent configuration without extensive tuning for our primary evaluations.

A.5.2. Ablation Study on Importance Scoring Design

Table 4 represents the ablation study on the importance scoring design. The result show that the Hadamard product and the l2 product with the max operation are the only combination that exhibits the best performance for both Qwen3-2B and 4B, respectively. This exhibits that the magnitude-aware scoring and probe-specific preservation proposed in the main paper are crucial for identifying the influence of KV cache entries.

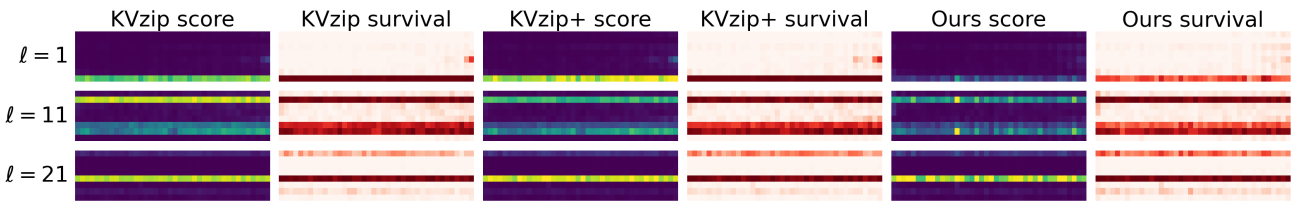


Figure 4. Importance score and survived KV entries of KVzip, KVzip+, and KVgrad

A.5.3. Ablation on Aggregation Function

Table 6 represented experimental results with varying aggregation methods. While both ℓ_1 and ℓ_2 exhibits strong performance, the ℓ_∞ and mean aggregation exhibits poor performance.

A.5.4. Score Visualization

Figure 4 show the importance score of KVzip, KVzip+, and KVgrad. While all three method show similar eviction pattern overall, KVgrad assigns lower importance to the first layer compared to the others. We attribute this difference to KVzip+, which appears to overestimate the importance of lower layers due to the application of $1/\|\mathbf{x}_p^\ell\|_2$.

A.6. Full Experimental Results

A.6.1. RULER RESULTS - QWEN3 0.6B

Table 7. Per-task RULER accuracy and NL-CR for Qwen3-0.6B.

		Accuracy (per CR)						NL-CR (per τ)	
Task	Press	No press	2:1	4:1	8:1	16:1	32:1	τ=2	τ=5
	KVzip		**0.5**	0.3	0.3	0.6	0.5	**32:1**	**32:1**
cwe	KVzip+	0.5	0.4	**0.4**	0.7	0.5	0.7	**32:1**	**32:1**
	KVgrad		0.4	0.3	**1.0**	**2.8**	**1.5**	**32:1**	**32:1**
	KVzip		76.7	**73.5**	61.2	38.9	22.7	1:1	**2:1**
fwe	KVzip+	79.5	**76.9**	72.9	**63.1**	**51.7**	35.6	1:1	**2:1**
	KVgrad		76.6	71.8	62.1	49.9	**36.1**	1:1	**2:1**
	KVzip		96.2	78.4	61.6	27.8	2.8	2:1	2:1
niah-mk1	KVzip+	97.8	**97.4**	82.8	68.0	45.2	5.8	2:1	2:1
	KVgrad		**97.4**	**96.0**	**87.8**	**69.0**	**43.6**	**4:1**	**4:1**
	KVzip		**92.6**	69.2	33.6	6.0	0.4	2:1	2:1
niah-mk2	KVzip+	91.6	92.4	90.2	43.8	7.4	0.8	**4:1**	**4:1**
	KVgrad		91.8	**92.8**	**85.4**	**58.0**	**6.4**	**4:1**	**4:1**
	KVzip		**91.2**	43.4	21.2	3.2	0.0	**2:1**	2:1
niah-mk3	KVzip+	90.0	90.6	62.6	19.8	4.4	0.0	**2:1**	2:1
	KVgrad		90.2	**87.8**	**69.6**	**26.4**	**1.2**	**2:1**	**4:1**
	KVzip		98.2	94.8	83.2	38.8	3.4	2:1	4:1
niah-mq	KVzip+	98.0	98.0	96.2	87.7	68.5	12.9	4:1	4:1
	KVgrad		**98.2**	**98.3**	**96.8**	**91.5**	**73.8**	**8:1**	**8:1**
	KVzip		88.5	74.5	57.7	25.2	2.6	1:1	2:1
niah-mv	KVzip+	90.7	89.2	78.6	60.9	37.0	8.8	**2:1**	2:1
	KVgrad		**90.1**	**87.4**	**81.6**	**74.0**	**60.0**	**2:1**	**4:1**
	KVzip		**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
niah-s1	KVzip+	100.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
	KVgrad		**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
	KVzip		**100.0**	99.2	99.0	25.4	5.6	**8:1**	8:1
niah-s2	KVzip+	100.0	**100.0**	99.8	**100.0**	82.4	17.0	**8:1**	8:1
	KVgrad		**100.0**	**100.0**	99.8	**96.2**	**62.0**	**8:1**	**16:1**
	KVzip		**99.4**	99.2	95.6	64.8	2.2	4:1	8:1
niah-s3	KVzip+	99.4	**99.4**	99.2	96.2	88.0	46.0	4:1	8:1
	KVgrad		**99.4**	**99.4**	**99.0**	**96.2**	**63.6**	**8:1**	**16:1**
	KVzip		58.4	54.6	38.0	20.2	7.8	2:1	2:1
qa1	KVzip+	58.2	**59.2**	56.2	44.2	24.2	8.0	2:1	**4:1**
	KVgrad		59.0	**57.2**	**48.0**	**27.2**	**14.6**	**4:1**	**4:1**
	KVzip		**33.0**	28.2	21.0	12.6	10.6	**2:1**	2:1
qa2	KVzip+	33.4	**33.0**	28.8	22.6	15.6	10.8	**2:1**	2:1
	KVgrad		**33.0**	**31.8**	**28.2**	**20.0**	**13.8**	**2:1**	**4:1**
	KVzip		79.2	75.4	50.6	34.7	34.6	2:1	4:1
vt	KVzip+	78.1	79.1	77.0	74.9	52.4	41.8	4:1	8:1
	KVgrad		**79.4**	**78.9**	**78.7**	**67.1**	**77.9**	**32:1**	**32:1**

A.6.2. RULER RESULTS - QWEN3 4B

Table 8. Per-task RULER accuracy and NL-CR for Qwen3-4B.

		Accuracy (per CR)						NL-CR (per τ)	
Task	Press	No press	2:1	4:1	8:1	16:1	32:1	τ=2	τ=5
	KVzip		94.2	93.7	77.0	34.3	0.7	**4:1**	**4:1**
cwe	KVzip+	93.9	94.3	**94.2**	84.8	61.6	2.2	**4:1**	**4:1**
	KVgrad		**94.6**	93.8	**87.8**	**77.5**	**52.9**	**4:1**	**4:1**
	KVzip		**87.8**	85.9	78.3	48.8	0.0	**4:1**	**4:1**
fwe	KVzip+	87.2	87.7	**86.0**	**79.5**	70.7	0.6	**4:1**	**4:1**
	KVgrad		87.6	85.1	78.7	**73.1**	**63.9**	2:1	**4:1**
	KVzip		**100.0**	99.6	91.8	60.6	1.8	4:1	4:1
niah-mk1	KVzip+	100.0	**100.0**	**100.0**	97.8	62.8	7.2	4:1	**8:1**
	KVgrad		**100.0**	**100.0**	**99.0**	**90.4**	**52.2**	**8:1**	**8:1**
	KVzip		**100.0**	**100.0**	98.0	76.8	9.8	**8:1**	**8:1**
niah-mk2	KVzip+	100.0	**100.0**	**100.0**	93.4	**89.8**	4.8	4:1	4:1
	KVgrad		**100.0**	**100.0**	**99.0**	78.6	**32.8**	**8:1**	**8:1**
	KVzip		**100.0**	**100.0**	97.8	20.0	0.0	4:1	**8:1**
niah-mk3	KVzip+	100.0	**100.0**	**100.0**	99.2	24.6	0.0	**8:1**	**8:1**
	KVgrad		**100.0**	**100.0**	**99.6**	**51.2**	**4.4**	**8:1**	**8:1**
	KVzip		**100.0**	97.0	93.1	46.4	0.2	2:1	4:1
niah-mq	KVzip+	100.0	**100.0**	97.6	93.4	62.4	9.7	2:1	4:1
	KVgrad		**100.0**	**99.0**	**96.7**	**81.8**	**46.1**	**4:1**	**8:1**
	KVzip		**100.0**	98.7	76.5	27.1	1.1	**4:1**	4:1
niah-mv	KVzip+	100.0	**100.0**	98.4	81.3	40.6	14.2	**4:1**	4:1
	KVgrad		**100.0**	**99.8**	**96.7**	**84.6**	**55.8**	**4:1**	**8:1**
	KVzip		**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
niah-s1	KVzip+	100.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
	KVgrad		**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
	KVzip		**100.0**	**100.0**	97.4	56.8	0.6	4:1	8:1
niah-s2	KVzip+	100.0	**100.0**	**100.0**	99.6	86.4	10.6	**8:1**	8:1
	KVgrad		**100.0**	**100.0**	**99.8**	**97.6**	**64.4**	**8:1**	**16:1**
	KVzip		**99.8**	**99.8**	98.0	32.6	0.2	8:1	8:1
niah-s3	KVzip+	99.8	**99.8**	**99.8**	97.8	75.4	0.0	4:1	8:1
	KVgrad		**99.8**	**99.8**	**99.8**	**99.6**	**79.4**	**16:1**	**16:1**
	KVzip		**80.2**	**81.6**	73.2	43.0	21.8	**4:1**	**4:1**
qa1	KVzip+	79.8	79.8	81.0	**74.0**	**44.8**	24.0	**4:1**	**4:1**
	KVgrad		**80.2**	80.0	62.6	35.6	**24.8**	**4:1**	**4:1**
	KVzip		**60.0**	57.8	53.6	35.6	20.2	2:1	**4:1**
qa2	KVzip+	59.6	59.4	59.2	55.4	41.8	22.4	**4:1**	**4:1**
	KVgrad		59.4	**59.8**	**56.2**	**43.2**	**31.4**	**4:1**	**4:1**
	KVzip		**100.0**	**100.0**	**100.0**	**100.0**	99.7	**32:1**	**32:1**
vt	KVzip+	100.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
	KVgrad		**100.0**	**100.0**	**100.0**	100.0	99.5	**32:1**	**32:1**

A.6.3. RULER RESULTS - QWEN3 8B

Table 9. Per-task RULER accuracy and NL-CR for Qwen3-8B.

		Accuracy (per CR)						NL-CR (per τ)	
Task	Press	No press	2:1	4:1	8:1	16:1	32:1	τ=2	τ=5
cwe	KVzip	98.9	99.1	**99.1**	94.3	43.9	0.6	4:1	**8:1**
	KVzip+		**99.2**	**99.1**	**97.4**	81.3	4.1	**8:1**	**8:1**
	KVgrad		99.1	98.5	96.0	**89.0**	**66.4**	4:1	**8:1**
fwe	KVzip	95.3	95.5	**95.8**	89.3	61.9	0.0	**4:1**	4:1
	KVzip+		95.5	95.7	**91.5**	**77.9**	2.1	**4:1**	**8:1**
	KVgrad		**95.6**	95.5	88.9	76.8	**56.2**	**4:1**	4:1
niah-mk1	KVzip	100.0	**100.0**	99.8	90.8	47.0	0.2	4:1	4:1
	KVzip+		**100.0**	**100.0**	97.4	47.4	21.4	4:1	**8:1**
	KVgrad		**100.0**	**100.0**	**98.8**	**89.6**	**60.8**	**8:1**	**8:1**
niah-mk2	KVzip	100.0	**100.0**	**100.0**	99.6	74.8	2.0	**8:1**	**8:1**
	KVzip+		**100.0**	**100.0**	**100.0**	**87.4**	1.2	**8:1**	**8:1**
	KVgrad		**100.0**	**100.0**	96.8	62.6	**24.4**	4:1	**8:1**
niah-mk3	KVzip	100.0	**100.0**	99.8	98.4	24.2	0.0	**8:1**	**8:1**
	KVzip+		**100.0**	99.8	**99.2**	**55.2**	0.0	**8:1**	**8:1**
	KVgrad		**100.0**	**100.0**	85.0	39.2	**2.2**	4:1	4:1
niah-mq	KVzip	99.9	**100.0**	100.0	99.2	41.2	0.1	**8:1**	**8:1**
	KVzip+		**100.0**	99.9	**99.7**	68.0	13.9	**8:1**	**8:1**
	KVgrad		99.8	**100.0**	98.8	**89.0**	**55.2**	**8:1**	**8:1**
niah-mv	KVzip	100.0	100.0	99.2	84.8	28.9	0.1	4:1	4:1
	KVzip+		**100.0**	99.7	89.1	50.4	22.1	4:1	4:1
	KVgrad		100.0	**99.8**	**98.9**	**93.4**	**69.8**	**8:1**	**8:1**
niah-s1	KVzip	100.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
	KVzip+		**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
	KVgrad		**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
niah-s2	KVzip	100.0	**100.0**	**100.0**	97.2	24.2	0.0	4:1	**8:1**
	KVzip+		**100.0**	**100.0**	**100.0**	75.0	22.2	**8:1**	**8:1**
	KVgrad		**100.0**	**100.0**	**100.0**	**94.6**	**54.4**	**8:1**	**8:1**
niah-s3	KVzip	100.0	**100.0**	**100.0**	98.6	26.8	0.0	8:1	8:1
	KVzip+		**100.0**	**100.0**	99.0	70.6	3.6	8:1	8:1
	KVgrad		**100.0**	**100.0**	**100.0**	**99.4**	**58.2**	**16:1**	**16:1**
qa1	KVzip	81.8	80.2	**81.4**	76.6	43.0	25.6	**4:1**	**4:1**
	KVzip+		80.8	81.2	**77.6**	**51.6**	**30.4**	**4:1**	**4:1**
	KVgrad		**81.0**	79.0	63.8	41.8	28.8	2:1	**4:1**
qa2	KVzip	62.8	63.4	61.0	56.2	32.4	22.6	2:1	**4:1**
	KVzip+		**63.6**	62.2	**59.0**	40.0	22.8	**4:1**	**4:1**
	KVgrad		**63.6**	**62.8**	57.0	**45.8**	**31.2**	**4:1**	**4:1**
vt	KVzip	100.0	**100.0**	**100.0**	**100.0**	100.0	98.8	**32:1**	**32:1**
	KVzip+		**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
	KVgrad		**100.0**	**100.0**	**100.0**	**100.0**	97.5	16:1	**32:1**

A.6.4. RULER RESULTS - LLAMA3.1 8B

Table 10. Per-task RULER accuracy and NL-CR for Llama-3.1-8B-Instruct.

		Accuracy (per CR)						**NL-CR (per τ)**	
Task	Press	No press	2:1	4:1	8:1	16:1	32:1	τ=2	τ=5
	KVzip		99.6	99.0	91.8	45.8	0.4	**4:1**	4:1
cwe	KVzip+	99.6	**99.6**	**99.2**	**95.6**	**50.4**	0.0	**4:1**	**8:1**
	KVgrad		99.5	97.3	83.7	46.8	**6.0**	2:1	4:1
	KVzip		94.3	94.5	**93.1**	82.8	1.6	**8:1**	**8:1**
fwe	KVzip+	94.8	**94.5**	**94.6**	92.9	83.5	25.8	4:1	**8:1**
	KVgrad		94.2	93.9	91.9	**88.7**	**50.6**	4:1	**8:1**
	KVzip		**100.0**	**100.0**	99.8	91.4	16.8	8:1	8:1
niah-mk1	KVzip+	99.8	**100.0**	**100.0**	99.8	93.4	24.6	8:1	8:1
	KVgrad		**100.0**	**100.0**	**100.0**	**98.4**	**78.6**	**16:1**	**16:1**
	KVzip		**100.0**	**100.0**	**99.8**	89.8	48.8	8:1	8:1
niah-mk2	KVzip+	100.0	**100.0**	99.8	99.2	95.4	18.2	8:1	**16:1**
	KVgrad		**100.0**	**100.0**	**99.8**	**98.0**	**63.4**	**16:1**	**16:1**
	KVzip		**99.8**	**100.0**	98.6	79.0	0.0	**8:1**	**8:1**
niah-mk3	KVzip+	99.8	**99.8**	**100.0**	98.2	61.6	0.0	**8:1**	**8:1**
	KVgrad		**99.8**	99.8	**99.8**	**80.6**	**7.4**	**8:1**	**8:1**
	KVzip		**100.0**	**100.0**	99.7	90.7	7.2	8:1	8:1
niah-mq	KVzip+	99.9	**100.0**	**100.0**	99.5	90.3	37.0	8:1	8:1
	KVgrad		100.0	**100.0**	**99.9**	**98.3**	**79.4**	**16:1**	**16:1**
	KVzip		**99.9**	**100.0**	97.8	61.9	7.8	4:1	8:1
niah-mv	KVzip+	99.9	**99.9**	99.8	97.8	67.5	20.9	4:1	8:1
	KVgrad		**99.9**	99.9	**99.6**	**95.1**	**70.2**	**8:1**	**16:1**
	KVzip		**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
niah-s1	KVzip+	100.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
	KVgrad		**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**32:1**	**32:1**
	KVzip		99.8	**100.0**	**100.0**	98.0	26.4	**16:1**	**16:1**
niah-s2	KVzip+	100.0	99.8	**100.0**	99.8	95.2	39.8	8:1	**16:1**
	KVgrad		**100.0**	**100.0**	99.8	**98.8**	**89.6**	**16:1**	**16:1**
	KVzip		**100.0**	**100.0**	**100.0**	93.2	1.6	8:1	8:1
niah-s3	KVzip+	100.0	**100.0**	**100.0**	**100.0**	88.4	10.0	8:1	8:1
	KVgrad		**100.0**	**100.0**	**100.0**	**98.8**	**65.8**	**16:1**	**16:1**
	KVzip		**87.0**	**85.0**	77.8	**53.6**	22.0	**2:1**	**4:1**
qa1	KVzip+	87.8	**87.0**	84.6	**80.6**	47.2	20.4	**2:1**	**4:1**
	KVgrad		86.6	84.4	76.4	48.0	**24.8**	**2:1**	**4:1**
	KVzip		**61.2**	**61.8**	57.2	43.8	25.8	**4:1**	**4:1**
qa2	KVzip+	62.8	**61.2**	60.0	56.8	**45.4**	23.4	1:1	**4:1**
	KVgrad		**61.2**	61.0	**57.4**	44.8	**30.8**	1:1	**4:1**
	KVzip		**99.9**	**99.9**	99.7	98.6	94.1	**16:1**	16:1
vt	KVzip+	99.9	**99.9**	**99.9**	**99.9**	98.6	95.6	**16:1**	**32:1**
	KVgrad		**99.9**	**99.9**	**99.9**	**99.6**	**97.0**	**16:1**	**32:1**

A.6.5. LONGBENCH RESULTS - QWEN3 0.6B

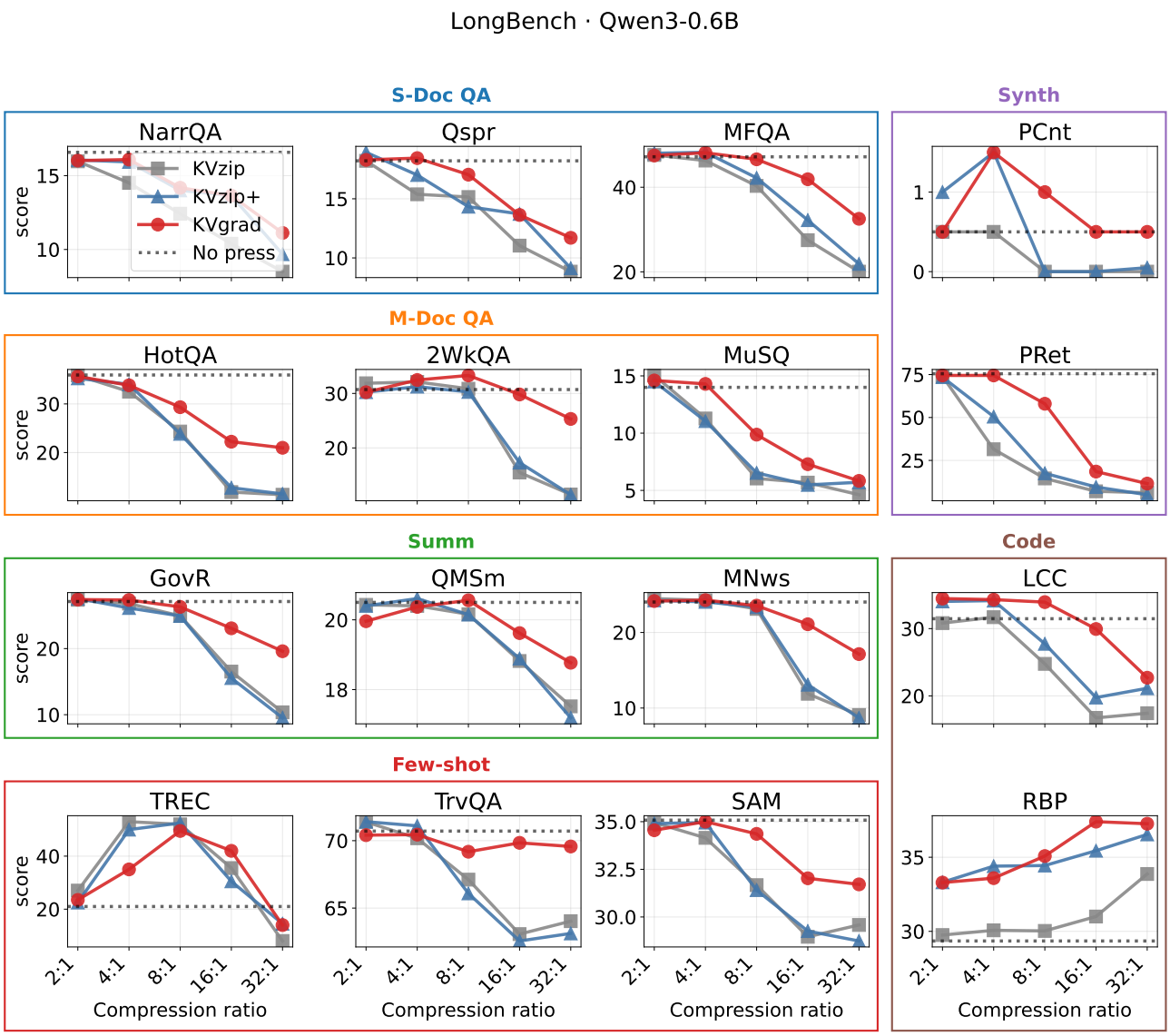


Figure 5. LongBench per-category curves for Qwen3-0.6B.

A.6.6. LONGBENCH RESULTS - QWEN3 4B

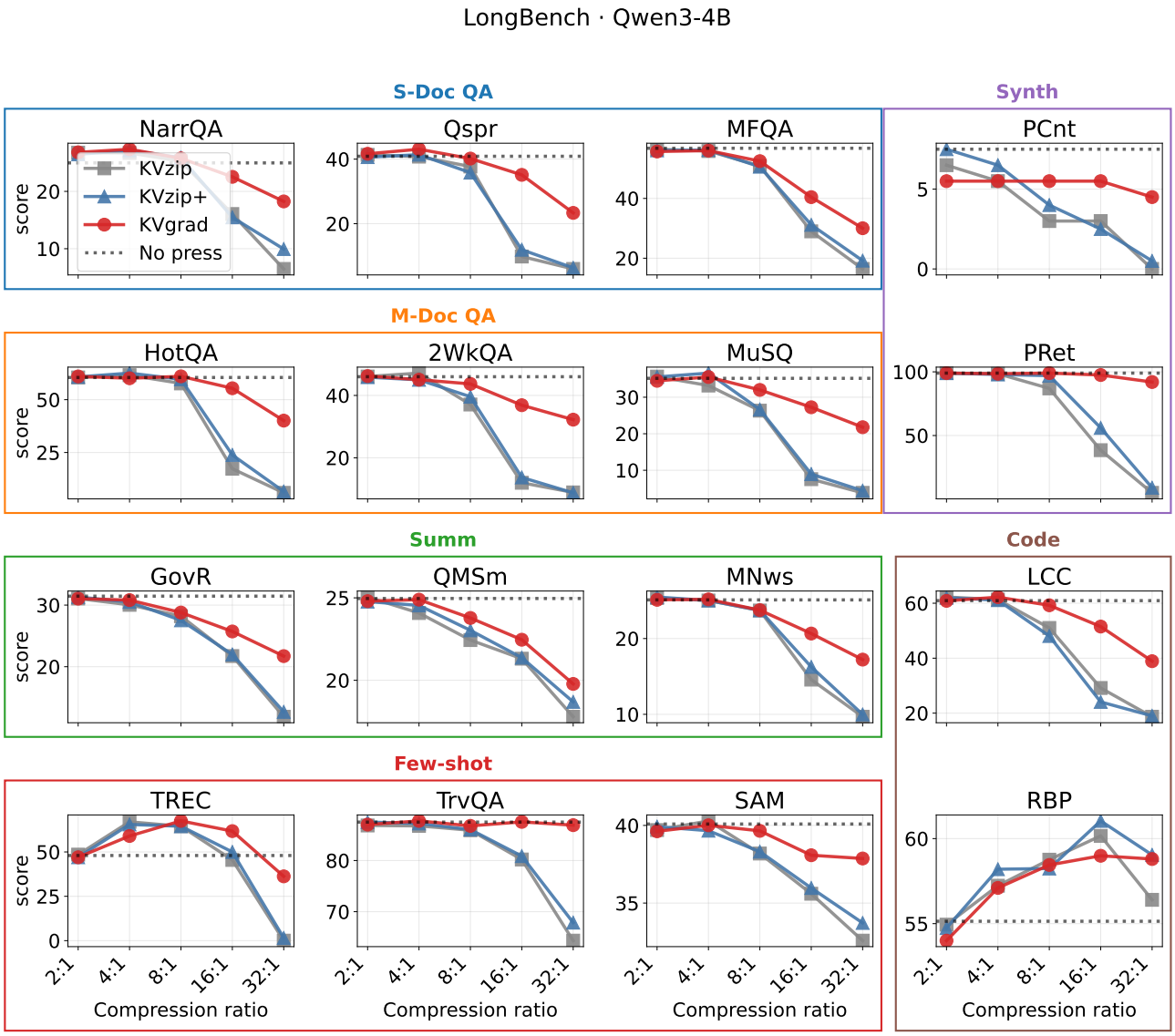


Figure 6. LongBench per-category curves for Qwen3-4B.

A.6.7. LONGBENCH RESULTS - QWEN3 8B

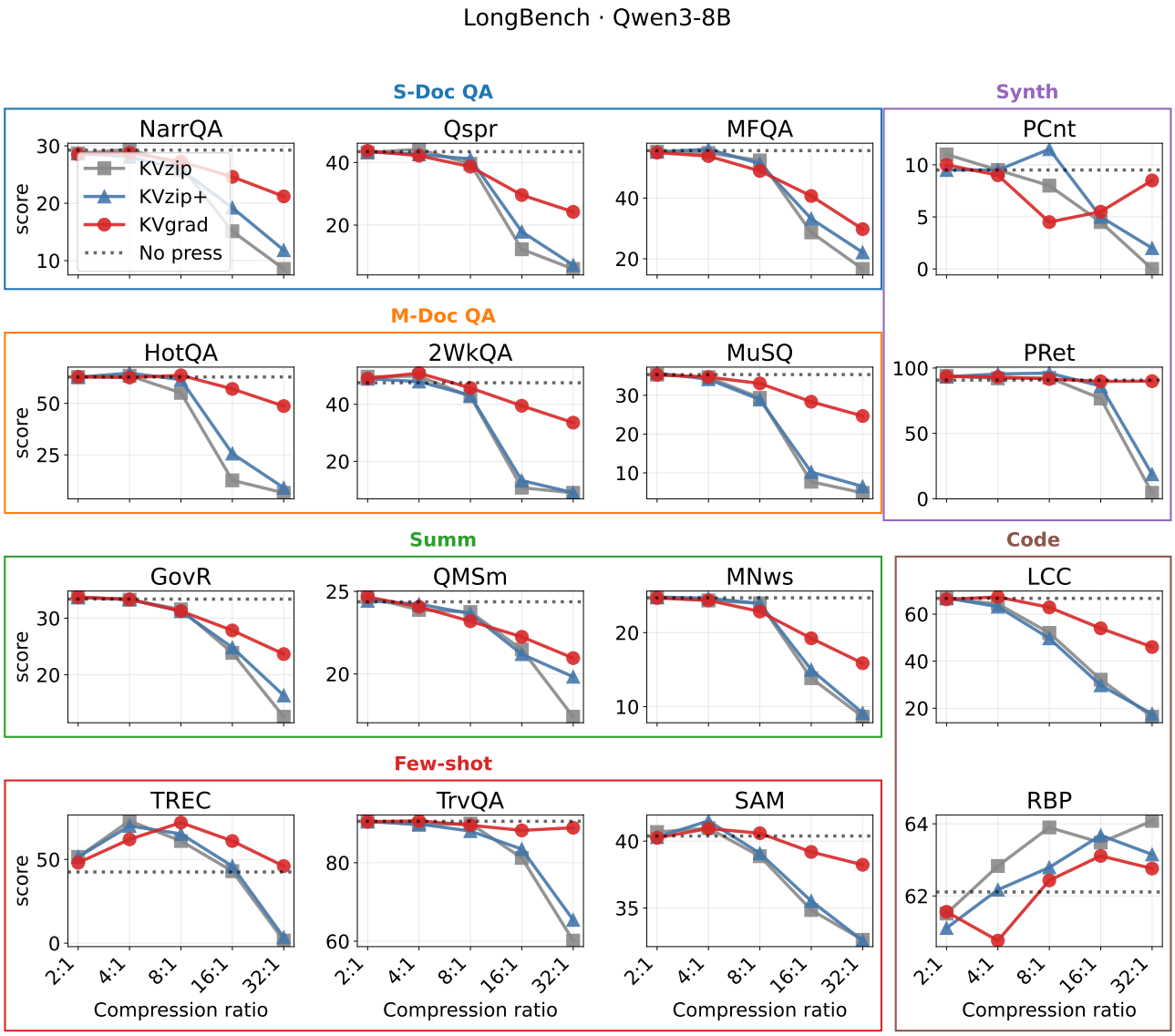


Figure 7. LongBench per-category curves for Qwen3-8B.

A.6.8. LONGBENCH RESULTS - LLAMA3.1 8B

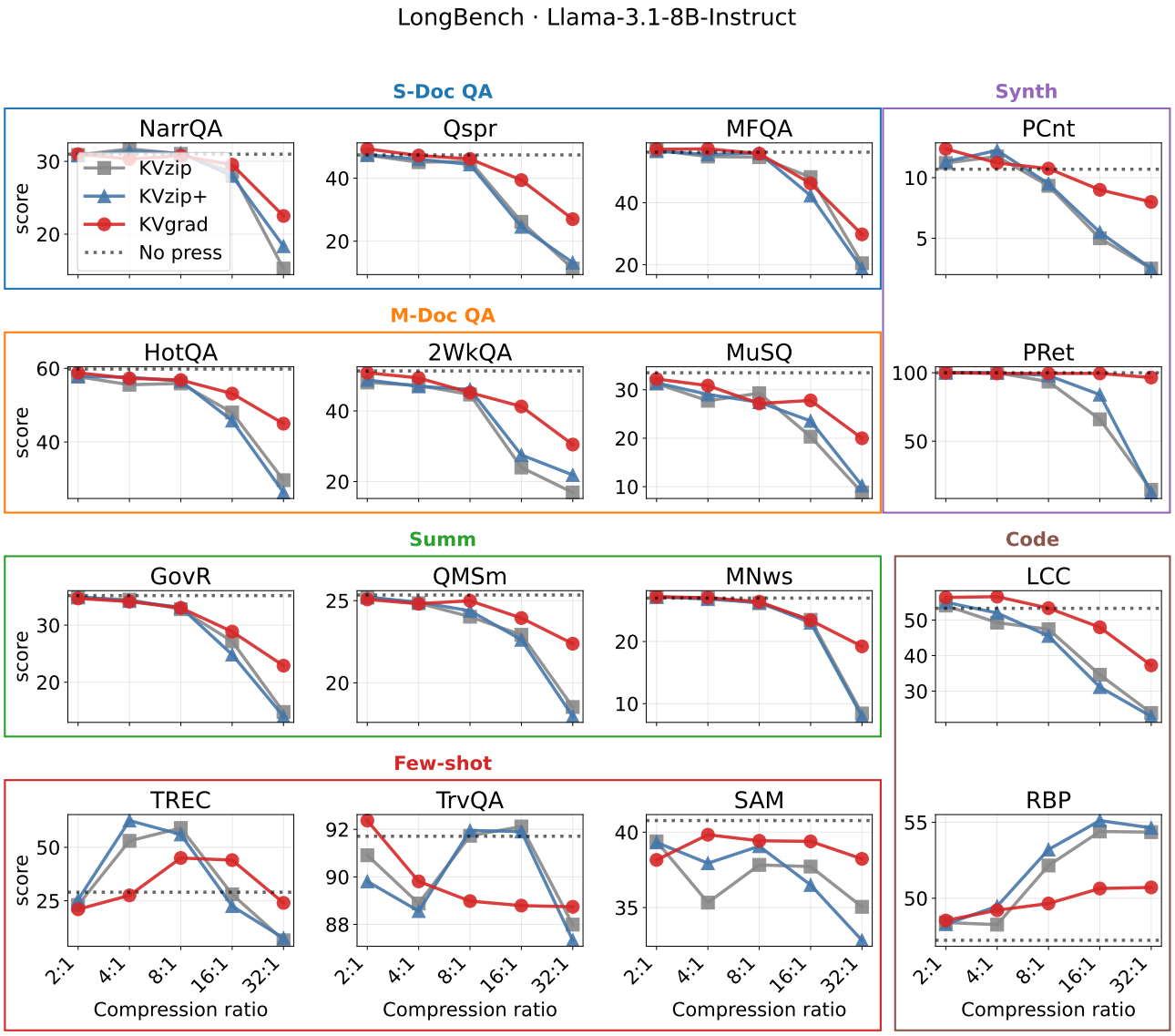


Figure 8. LongBench per-category curves for Llama-3.1-8B-Instruct.

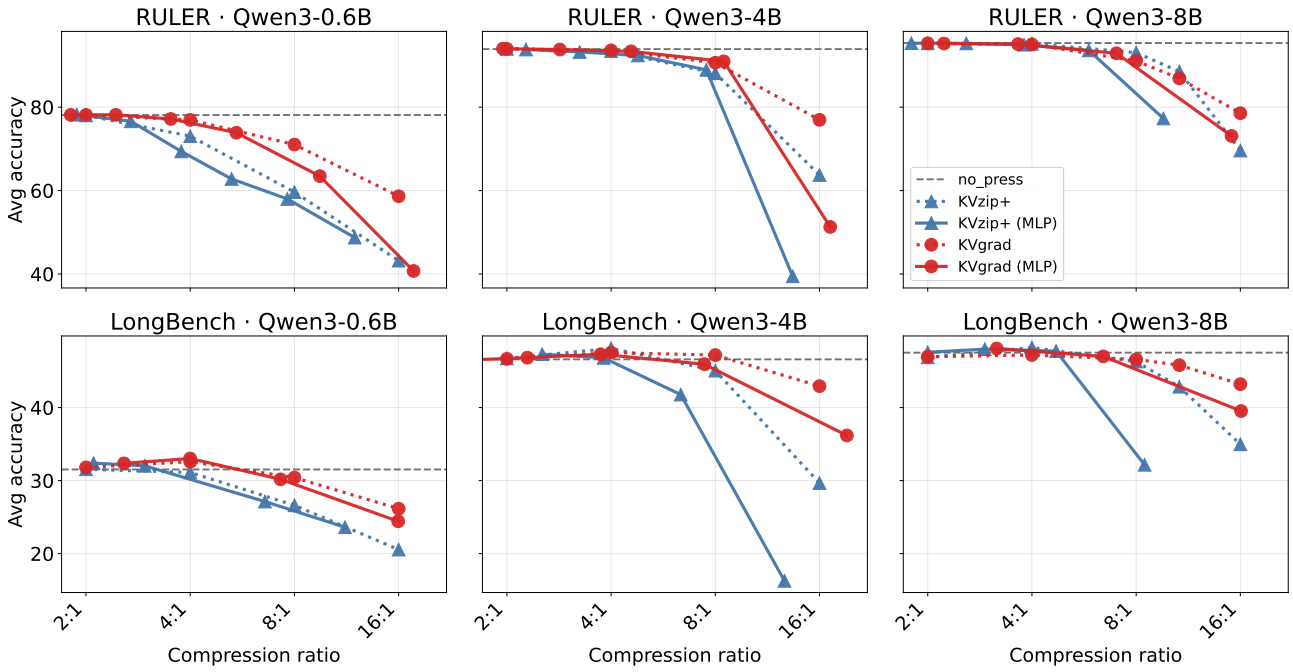


Figure 9. Average accuracy vs. Compression ratio on RULER and LongBench

Table 11. LongBench results and NL-CR for Qwen3-0.6B.

			Accuracy (per CR)						NL-CR (per τ)	
Cat	Task	Press	No press	2:1	4:1	8:1	16:1	32:1	τ=2	τ=5
S-Doc QA	NarrQA	KVzip	16.6	16.0	14.5	12.4	10.4	8.5	1:1	2:1
		KVzip+		**16.1**	15.9	14.0	**13.7**	9.7	1:1	**4:1**
		KVgrad		16.0	**16.1**	**14.2**	**13.7**	**11.1**	1:1	**4:1**
	Qspr	KVzip	18.2	18.2	15.4	15.2	11.0	8.8	2:1	2:1
		KVzip+		**18.9**	17.0	14.3	**13.7**	9.1	2:1	2:1
		KVgrad		18.3	**18.4**	**17.0**	13.6	**11.7**	**4:1**	**4:1**
	MFQA	KVzip	47.2	47.6	46.3	40.3	27.5	20.1	4:1	4:1
		KVzip+		**48.0**	**48.2**	42.2	32.2	21.9	4:1	4:1
		KVgrad		47.5	48.1	**46.6**	**41.9**	**32.5**	**8:1**	**8:1**
M-Doc QA	HotQA	KVzip	36.0	**35.9**	32.5	24.3	11.9	11.3	**2:1**	**2:1**
		KVzip+		35.3	**34.0**	23.9	12.8	11.5	**2:1**	**2:1**
		KVgrad		35.7	33.8	**29.3**	**22.2**	**21.0**	**2:1**	**2:1**
	2WkQA	KVzip	30.7	**31.9**	32.1	30.9	15.5	11.5	**8:1**	8:1
		KVzip+		30.2	31.2	30.3	17.3	11.4	**8:1**	8:1
		KVgrad		30.2	**32.5**	**33.3**	**29.8**	**25.3**	**8:1**	**16:1**
	MuSQ	KVzip	14.0	**15.0**	11.3	6.0	5.7	4.6	2:1	2:1
		KVzip+		14.5	11.1	6.5	5.5	5.7	2:1	2:1
		KVgrad		14.6	**14.3**	**9.9**	**7.3**	**5.8**	**4:1**	**4:1**
Summ	GovR	KVzip	27.1	27.4	26.8	24.9	16.5	10.4	**4:1**	4:1
		KVzip+		**27.6**	26.1	24.9	15.5	9.5	2:1	4:1
		KVgrad		27.4	**27.4**	**26.3**	**23.1**	**19.6**	**4:1**	**8:1**
	QMSm	KVzip	20.5	**20.4**	20.4	20.2	18.8	17.5	**8:1**	8:1
		KVzip+		20.4	**20.6**	20.1	18.9	17.2	**8:1**	8:1
		KVgrad		20.0	20.4	**20.6**	**19.6**	**18.8**	**8:1**	**16:1**
	MNws	KVzip	24.0	**24.5**	24.2	23.1	11.9	9.1	4:1	**8:1**
		KVzip+		24.2	24.1	23.4	13.1	8.7	4:1	**8:1**
		KVgrad		24.2	**24.3**	**23.6**	**21.1**	**17.1**	**8:1**	**8:1**
Few-shot	TREC	KVzip	21.0	**27.0**	**53.0**	52.0	35.5	8.0	**16:1**	**16:1**
		KVzip+		22.5	50.0	**52.5**	30.5	**14.5**	**16:1**	**16:1**
		KVgrad		23.5	35.0	49.5	**42.0**	14.0	**16:1**	**16:1**
	TrvQA	KVzip	70.7	71.4	70.2	67.1	63.1	64.0	4:1	4:1
		KVzip+		**71.4**	**71.1**	66.1	62.6	63.1	4:1	4:1
		KVgrad		70.4	70.5	**69.2**	**69.8**	**69.6**	**32:1**	**32:1**
	SAM	KVzip	35.1	**35.0**	34.1	31.7	29.0	29.6	2:1	4:1
		KVzip+		34.9	35.0	31.4	29.3	28.7	**4:1**	4:1
		KVgrad		34.5	**35.0**	**34.4**	**32.0**	**31.7**	**4:1**	**8:1**
Synth	PCnt	KVzip	0.5	0.5	0.5	0.0	0.0	0.0	4:1	4:1
		KVzip+		**1.0**	**1.5**	0.0	0.0	0.1	4:1	4:1
		KVgrad		0.5	**1.5**	**1.0**	**0.5**	**0.5**	**32:1**	**32:1**
	PRet	KVzip	75.5	74.0	31.5	14.5	7.0	6.2	2:1	2:1
		KVzip+		73.5	50.5	17.5	9.5	5.0	1:1	2:1
		KVgrad		**74.5**	**74.5**	**58.0**	**18.5**	**11.5**	**4:1**	**4:1**
Code	LCC	KVzip	31.5	30.8	31.7	24.8	16.7	17.4	4:1	4:1
		KVzip+		34.0	34.2	27.7	19.8	21.1	4:1	4:1
		KVgrad		**34.5**	**34.3**	**33.9**	**29.9**	**22.7**	**8:1**	**16:1**
	RBP	KVzip	29.4	29.8	30.1	30.0	31.0	33.9	**32:1**	**32:1**
		KVzip+		**33.3**	**34.4**	34.4	35.4	36.5	**32:1**	**32:1**
		KVgrad		33.3	33.6	**35.1**	**37.4**	**37.3**	**32:1**	**32:1**

Table 12. LongBench results and NL-CR for Qwen3-4B.

			Accuracy (per CR)						NL-CR (per τ)	
Cat	Task	Press	No press	2:1	4:1	8:1	16:1	32:1	τ=2	τ=5
S-Doc QA	NarrQA	KVzip	25.0	26.8	26.7	25.3	16.1	6.5	**8:1**	**8:1**
		KVzip+		26.5	27.2	**25.9**	15.5	9.9	**8:1**	**8:1**
		KVgrad		**26.8**	**27.4**	25.8	**22.5**	**18.2**	**8:1**	**8:1**
	Qspr	KVzip	40.9	41.3	40.8	37.7	9.8	6.0	4:1	4:1
		KVzip+		40.6	41.4	35.8	11.9	6.3	4:1	4:1
		KVgrad		**41.7**	**43.1**	**40.2**	**35.1**	**23.3**	**8:1**	**8:1**
	MFQA	KVzip	56.8	**56.5**	55.9	50.9	29.0	16.4	**4:1**	**4:1**
		KVzip+		56.1	**56.2**	50.6	31.2	19.2	**4:1**	**4:1**
		KVgrad		55.7	56.0	**52.5**	**40.4**	**30.0**	**4:1**	**4:1**
M-Doc QA	HotQA	KVzip	60.5	60.5	61.6	57.7	17.3	5.9	4:1	**8:1**
		KVzip+		60.8	**62.6**	59.6	23.8	6.5	**8:1**	**8:1**
		KVgrad		**61.0**	60.1	**61.0**	**55.4**	**40.1**	**8:1**	**8:1**
	2WkQA	KVzip	46.0	46.1	**47.2**	37.1	12.0	8.9	**4:1**	4:1
		KVzip+		45.9	45.0	39.6	13.7	8.8	2:1	4:1
		KVgrad		**46.2**	45.0	**43.7**	**36.9**	**32.2**	2:1	**8:1**
	MuSQ	KVzip	35.2	**35.6**	33.2	26.3	7.6	3.8	2:1	2:1
		KVzip+		35.5	**36.6**	26.6	8.9	4.4	**4:1**	**4:1**
		KVgrad		34.5	35.5	**32.0**	**27.2**	**21.8**	**4:1**	**4:1**
Summ	GovR	KVzip	31.4	31.1	30.0	28.2	21.7	11.9	**2:1**	**4:1**
		KVzip+		**31.3**	30.5	27.5	22.0	12.7	**2:1**	**4:1**
		KVgrad		31.1	**30.8**	**28.8**	**25.7**	**21.7**	**2:1**	**4:1**
	QMSm	KVzip	25.0	**25.1**	24.1	22.4	21.3	17.8	2:1	4:1
		KVzip+		24.8	24.5	23.0	21.4	18.7	**4:1**	4:1
		KVgrad		24.8	**24.9**	**23.8**	**22.5**	**19.8**	**4:1**	**8:1**
	MNws	KVzip	25.1	25.4	25.0	23.7	14.6	9.7	**4:1**	**4:1**
		KVzip+		**25.5**	25.0	23.7	16.3	10.0	**4:1**	**4:1**
		KVgrad		25.1	**25.2**	**23.8**	**20.7**	**17.2**	**4:1**	**4:1**
Few-shot	TREC	KVzip	48.0	**48.5**	**67.0**	64.5	45.5	0.0	8:1	8:1
		KVzip+		47.0	65.5	64.5	50.0	1.5	**16:1**	**16:1**
		KVgrad		47.0	59.0	**67.5**	**61.8**	**36.2**	**16:1**	**16:1**
	TrvQA	KVzip	87.5	86.8	86.8	85.9	80.2	64.3	8:1	8:1
		KVzip+		**87.5**	87.2	86.0	80.9	67.9	8:1	8:1
		KVgrad		87.0	**87.7**	**86.8**	**87.5**	**86.9**	**32:1**	**32:1**
	SAM	KVzip	40.1	39.7	**40.3**	38.2	35.6	32.6	4:1	**8:1**
		KVzip+		**39.9**	39.7	38.3	36.0	33.7	4:1	**8:1**
		KVgrad		39.6	40.0	**39.6**	**38.1**	**37.9**	**8:1**	**8:1**
Synth	PCnt	KVzip	7.5	6.5	5.5	3.0	3.0	0.0	1:1	1:1
		KVzip+		**7.5**	**6.5**	4.0	2.5	0.5	**2:1**	**2:1**
		KVgrad		5.5	5.5	**5.5**	**5.5**	**4.5**	1:1	1:1
	PRet	KVzip	99.0	**99.0**	**98.5**	87.0	38.5	5.0	4:1	4:1
		KVzip+		**99.0**	97.9	97.0	56.0	9.0	4:1	8:1
		KVgrad		**99.0**	**98.5**	**99.0**	**97.5**	**92.0**	**16:1**	**16:1**
Code	LCC	KVzip	60.9	**62.3**	61.3	51.0	29.2	18.7	**4:1**	4:1
		KVzip+		62.0	61.1	48.1	24.1	19.0	**4:1**	4:1
		KVgrad		60.9	**62.2**	**59.2**	**51.5**	**38.9**	**4:1**	**8:1**
	RBP	KVzip	55.1	**54.9**	57.2	**58.8**	60.2	56.4	**32:1**	**32:1**
		KVzip+		54.7	**58.2**	58.2	**61.0**	**59.0**	**32:1**	**32:1**
		KVgrad		54.0	57.1	58.5	59.0	58.8	**32:1**	**32:1**

Table 13. LongBench results and NL-CR for Qwen3-8B.

			Accuracy (per CR)						NL-CR (per τ)	
Cat	Task	Press	No press	2:1	4:1	8:1	16:1	32:1	τ=2	τ=5
S-Doc QA	NarrQA	KVzip	29.3	28.7	**29.4**	26.6	15.1	8.6	**4:1**	**4:1**
		KVzip+		**28.7**	28.1	25.9	19.2	11.8	2:1	**4:1**
		KVgrad		28.5	28.8	**27.3**	**24.6**	**21.2**	**4:1**	**4:1**
	Qspr	KVzip	43.5	43.1	**44.2**	39.6	12.3	6.0	**4:1**	**4:1**
		KVzip+		43.4	42.6	**41.1**	17.9	7.2	**4:1**	**4:1**
		KVgrad		**43.7**	42.2	38.7	**29.6**	**24.2**	2:1	**4:1**
	MFQA	KVzip	55.6	**55.3**	54.8	**52.4**	28.7	16.6	**4:1**	**4:1**
		KVzip+		55.2	**56.1**	51.3	33.2	22.1	**4:1**	**4:1**
		KVgrad		55.0	53.8	48.9	**40.7**	**29.8**	2:1	**4:1**
M-Doc QA	HotQA	KVzip	62.8	62.8	63.4	55.1	12.7	6.6	4:1	4:1
		KVzip+		62.7	**64.7**	61.5	25.6	9.0	4:1	**8:1**
		KVgrad		**62.8**	62.5	**63.6**	**57.0**	**48.7**	**8:1**	**8:1**
	2WkQA	KVzip	47.5	**49.7**	49.8	42.8	10.8	9.0	**4:1**	4:1
		KVzip+		48.9	48.0	43.1	13.3	9.0	**4:1**	4:1
		KVgrad		49.0	**50.9**	**45.7**	**39.5**	**33.5**	**4:1**	**8:1**
	MuSQ	KVzip	35.3	35.3	34.7	29.3	7.7	4.9	**4:1**	**4:1**
		KVzip+		**35.7**	34.1	28.9	10.2	6.5	2:1	**4:1**
		KVgrad		35.2	**34.7**	**33.0**	**28.3**	**24.7**	**4:1**	**4:1**
Summ	GovR	KVzip	33.4	**33.7**	33.2	**31.5**	23.9	12.6	**4:1**	**4:1**
		KVzip+		33.6	**33.3**	31.2	24.8	16.4	**4:1**	**4:1**
		KVgrad		33.7	33.3	31.2	**27.8**	**23.7**	**4:1**	**4:1**
	QMSm	KVzip	24.4	**24.7**	23.9	**23.8**	21.5	17.4	2:1	**8:1**
		KVzip+		24.4	**24.2**	23.6	21.2	19.8	**4:1**	**8:1**
		KVgrad		**24.7**	24.1	23.2	**22.2**	**21.0**	**4:1**	**8:1**
	MNws	KVzip	24.7	24.7	24.3	**24.0**	13.8	8.6	**4:1**	**8:1**
		KVzip+		**24.8**	**24.6**	23.9	15.0	9.2	**4:1**	**8:1**
		KVgrad		24.7	24.4	22.9	**19.2**	**15.9**	**4:1**	4:1
Few-shot	TREC	KVzip	42.5	**51.5**	**73.0**	61.0	43.0	1.5	16:1	16:1
		KVzip+		**51.5**	70.0	65.2	46.0	3.5	16:1	16:1
		KVgrad		48.0	62.0	**72.0**	**61.0**	**46.0**	**32:1**	**32:1**
	TrvQA	KVzip	90.6	**90.7**	89.6	**90.0**	81.2	60.1	8:1	8:1
		KVzip+		90.5	89.9	88.1	83.5	65.4	4:1	8:1
		KVgrad		90.5	**90.7**	89.6	**88.2**	**88.9**	**32:1**	**32:1**
	SAM	KVzip	40.4	**40.7**	41.0	38.9	34.9	32.6	4:1	8:1
		KVzip+		40.3	**41.5**	39.1	35.5	32.6	4:1	8:1
		KVgrad		40.2	40.9	**40.6**	**39.2**	**38.2**	**8:1**	**16:1**
Synth	PCnt	KVzip	9.5	**11.0**	**9.5**	8.0	4.5	0.0	4:1	4:1
		KVzip+		9.5	**9.5**	**11.5**	5.0	2.0	**8:1**	**8:1**
		KVgrad		10.0	9.0	4.5	**5.5**	**8.5**	2:1	2:1
	PRet	KVzip	90.8	**93.9**	92.1	92.6	76.7	4.5	8:1	8:1
		KVzip+		93.3	**95.5**	**96.1**	86.2	18.7	8:1	16:1
		KVgrad		93.6	93.1	91.6	**89.8**	**90.0**	**32:1**	**32:1**
Code	LCC	KVzip	66.7	66.6	64.3	52.0	32.1	16.4	2:1	**4:1**
		KVzip+		**66.8**	63.2	49.7	29.8	17.6	2:1	2:1
		KVgrad		66.3	**67.3**	**62.9**	**54.0**	**46.0**	**4:1**	**4:1**
	RBP	KVzip	62.1	61.5	**62.8**	**63.9**	63.5	**64.1**	**32:1**	**32:1**
		KVzip+		61.1	62.2	62.8	**63.7**	63.1	**32:1**	**32:1**
		KVgrad		**61.6**	60.8	62.4	63.1	62.8	**32:1**	**32:1**

Table 14. LongBench results and NL-CR for Llama-3.1-8B-Instruct.

			Accuracy (per CR)						NL-CR (per τ)	
Cat	Task	Press	No press	2:1	4:1	8:1	16:1	32:1	τ=2	τ=5
S-Doc QA	NarrQA	KVzip	31.0	30.9	**31.7**	31.0	28.6	15.3	**8:1**	8:1
		KVzip+		30.8	31.4	**31.1**	28.0	18.4	**8:1**	8:1
		KVgrad		**31.0**	30.3	30.8	**29.6**	**22.5**	**8:1**	**16:1**
	Qspr	KVzip	47.4	47.3	45.1	45.3	26.1	11.3	2:1	**8:1**
		KVzip+		47.4	46.0	44.4	24.6	13.2	2:1	4:1
		KVgrad		**49.3**	**47.2**	**46.1**	**39.4**	**27.0**	**4:1**	**8:1**
	MFQA	KVzip	56.3	57.0	54.9	54.7	**48.3**	20.5	2:1	**8:1**
		KVzip+		56.6	55.6	**56.0**	42.3	18.8	**8:1**	**8:1**
		KVgrad		**57.2**	**57.4**	55.9	46.2	**29.8**	**8:1**	**8:1**
M-Doc QA	HotQA	KVzip	59.8	57.7	55.5	55.9	48.0	29.6	1:1	2:1
		KVzip+		57.8	**57.6**	56.5	45.8	26.3	1:1	4:1
		KVgrad		**58.8**	57.3	**56.8**	**53.1**	**45.0**	**2:1**	**8:1**
	2WkQA	KVzip	51.3	48.1	47.2	44.6	24.0	16.9	1:1	1:1
		KVzip+		48.8	47.0	**46.1**	27.5	21.9	1:1	2:1
		KVgrad		**50.8**	**49.4**	45.2	**41.3**	**30.5**	**2:1**	**4:1**
	MuSQ	KVzip	33.5	31.3	27.7	**29.3**	20.3	8.8	1:1	1:1
		KVzip+		31.4	29.0	27.4	23.6	10.2	1:1	1:1
		KVgrad		**32.2**	**30.9**	27.2	**27.8**	**20.0**	1:1	**2:1**
Summ	GovR	KVzip	35.2	34.8	**34.5**	32.8	27.2	14.8	**2:1**	**4:1**
		KVzip+		**35.0**	34.2	**33.1**	24.8	14.0	**2:1**	**4:1**
		KVgrad		34.7	34.1	33.0	**28.9**	**22.9**	**2:1**	**4:1**
	QMSm	KVzip	25.3	**25.2**	24.8	24.0	22.9	18.5	4:1	4:1
		KVzip+		25.2	**24.9**	24.4	22.6	18.0	4:1	**8:1**
		KVgrad		25.1	24.8	**25.0**	**23.9**	**22.4**	**8:1**	**8:1**
	MNws	KVzip	27.0	27.1	27.0	26.2	**23.5**	8.4	**4:1**	**8:1**
		KVzip+		**27.2**	26.8	26.3	23.0	8.0	**4:1**	**8:1**
		KVgrad		27.2	**27.1**	**26.4**	23.4	**19.2**	**4:1**	**8:1**
Few-shot	TREC	KVzip	29.0	23.0	53.0	**59.0**	28.0	6.5	8:1	**16:1**
		KVzip+		**25.5**	**62.5**	56.0	22.5	7.5	8:1	8:1
		KVgrad		21.0	27.5	45.0	**44.0**	**24.0**	**16:1**	**16:1**
	TrvQA	KVzip	91.7	90.9	88.9	91.7	**92.1**	88.0	**16:1**	**32:1**
		KVzip+		89.8	88.6	**92.0**	91.9	87.3	**16:1**	**32:1**
		KVgrad		**92.4**	**89.8**	89.0	88.8	**88.7**	2:1	**32:1**
	SAM	KVzip	40.8	**39.4**	35.3	37.8	37.7	35.0	1:1	2:1
		KVzip+		39.3	37.9	39.1	36.5	32.8	1:1	8:1
		KVgrad		38.2	**39.8**	**39.4**	**39.4**	**38.2**	1:1	**16:1**
Synth	PCnt	KVzip	10.7	11.2	11.8	9.3	5.0	2.5	4:1	4:1
		KVzip+		11.3	**12.2**	9.5	5.5	2.5	4:1	4:1
		KVgrad		**12.4**	11.2	**10.8**	**9.0**	**8.0**	**8:1**	**8:1**
	PRet	KVzip	100.0	**100.0**	**100.0**	93.5	66.0	14.5	4:1	4:1
		KVzip+		**100.0**	**100.0**	98.0	84.0	12.5	8:1	8:1
		KVgrad		**100.0**	99.5	**99.5**	**99.5**	**96.5**	**16:1**	**32:1**
Code	LCC	KVzip	53.3	54.1	49.3	47.4	34.6	23.8	2:1	2:1
		KVzip+		55.0	52.0	45.5	31.1	22.9	2:1	4:1
		KVgrad		**56.4**	**56.6**	**53.4**	**48.0**	**37.2**	**8:1**	**8:1**
	RBP	KVzip	47.2	48.4	48.3	52.1	54.4	54.4	**32:1**	**32:1**
		KVzip+		48.3	**49.5**	**53.2**	**55.1**	**54.6**	**32:1**	**32:1**
		KVgrad		**48.5**	49.2	49.6	50.6	50.7	**32:1**	**32:1**