SVEQ: Closed-Form Per-Channel Bit Allocation for 2-Bit Key Cache Quantization

Anonymous ACL submission

Abstract

The key-value (KV) cache dominates memory in long-context LLM serving. 2-bit quantization is the standard operating point for aggressive cache compression, but a measurable FP16 gap remains, driven primarily by per-channel range non-uniformity in the key cache. Recent per-channel mixed-precision methods exploit this non-uniformity but each requires calibration data, online profiling, or stored bit-width indices beyond the min-max metadata that any quantizer maintains. We frame per-channel bit allocation as the minimization of a worst-case attention-output error bound, and derive a closed-form solution $b_c = B + \log_2(R_c/\bar{R}_{\text{geo}})$ that depends only on the per-channel range R_c and its within-group geometric mean. We realize this allocation as **SVEQ**, which requires no calibration, no profiling, and no auxiliary indices. Across four LLMs and six tasks, SVEQ at $\sim$2.09 effective bits consistently matches or exceeds KIVI and stays within 2 absolute points of FP16, including FP16-level Needle-in-a-Haystack retrieval up to 128K context. It also matches the accuracy of the strongest prior mixed-precision baseline at a lower effective bit rate. A Triton prototype scales to 5.3$\times$ FP16's peak batch size and 4.4$\times$ its decoding throughput on a single A800 80GB GPU.

1 Introduction

The key-value (KV) cache stores attention keys and values during autoregressive generation to avoid recomputation, and its memory footprint grows linearly with both sequence length and batch size. This makes the KV cache the dominant memory bottleneck in long-context, high-throughput LLM serving (Liu et al., 2024). Quantization is the most direct way to reduce this footprint. At 2-bit precision the cache shrinks by $8\times$ over FP16, enabling substantially larger batches and longer contexts within the same memory budget, while remaining the most aggressive bit-width that current methods reach without major accuracy loss.

KIVI (Liu et al., 2024) combines per-channel quantization for keys with per-token quantization for values, and successfully extends KV cache compression into the 2-bit regime. It has since become the de facto standard in current serving stacks. On recent capable models, however, the FP16–KIVI gap remains non-trivial across both language modeling and downstream tasks. A primary cause is a structural property of the key cache. Within a single quantization group, per-channel key ranges vary by roughly $5\times$ (§3). Uniform b-bit quantization error scales as $R_c^2/(2^b-1)^2$, so this range spread translates into a steep error imbalance. Across three models, the widest 25% of channels carry about half of the total quantization error under uniform allocation, and uniform bit-widths cannot correct this asymmetry.

A natural fix is to give those high-range channels more bits. Several recent key cache methods, including Kitty (Xia et al., 2025), MixKVQ (Zhang et al., 2025b), and ChanMix (Liao and Wen, 2026), show that per-channel mixed precision can close part of the residual gap. Each of these methods, however, requires auxiliary deployment artifacts beyond the standard min-max metadata that any quantizer already maintains, in the form of online profiling, calibration data, or stored per-channel indices (we contrast each method in detail in §2). This raises an obvious question. Can per-channel mixed precision be deployed with zero such overhead?

Our answer is yes. We frame per-channel bit allocation as a minimax over a worst-case attention-output error bound, and specialize the classical log-variance rate-distortion rule to this objective. The resulting closed-form allocation is

$$\boxed{b_c \;=\; B \;+\; \log_2\!\left(\frac{R_c}{\bar{R}_{\text{geo}}}\right)}$$

where R_c is the per-channel range and $\bar{R}_{\text{geo}}$ is

the geometric mean of ranges within the group. The sensitivity proxy that drives this allocation collapses to R_c alone, a quantity already maintained as min-max metadata. We instantiate this allocation as **SVEQ**, the first per-channel key cache mixed-precision quantizer that requires *no calibration data, no online profiling, and no auxiliary indices*.

Contributions.

1. We derive a closed-form per-channel key cache bit allocation $b_c = B + \log_2(R_c/\bar{R}_{\text{geo}})$ from a minimax attention-output bound, identifying the per-channel range R_c as the optimal sensitivity in this setting (§4).
2. We instantiate this allocation as SVEQ, the first per-channel key cache quantizer requiring *no* calibration data, online profiling, or auxiliary index storage beyond the standard min-max metadata that any quantizer already maintains (§5).
3. At $\sim$2.09 effective bits per element, SVEQ stays within 2 absolute points of FP16 across six benchmarks (WikiText-2 perplexity, MMLU, GSM8K, HumanEval, Needle-in-a-Haystack, LongBench) on four LLMs (Llama-3-8B/70B, Qwen2-7B, Mistral-7B), including FP16-level Needle-in-a-Haystack retrieval up to 128K context (§6).

2 Related Work

Uniform-bit KV cache quantization. The standard line of KV cache quantization assigns a single bit-width to every channel and token. KIVI (Liu et al., 2024) establishes that the key cache must be quantized per-channel rather than per-token because key channels exhibit persistent outliers. Its design pairs per-channel keys with per-token values under group-wise scaling, with an FP16 residual buffer for the most recent tokens. It has since become the standard 2-bit baseline in current serving stacks. KVQuant (Hooper et al., 2024) adds non-uniform datatypes and a dense-plus-sparse decomposition for outlier channels. QuaRot (Ashkboos et al., 2024) applies Hadamard rotations to equalize channel distributions before quantization, at the cost of online matrix multiplications. QServe (Lin et al., 2024) migrates outlier magnitudes into projection weights via SmoothAttention. GEAR (Kang et al., 2024) separates outliers via low-rank plus sparse decomposition. All of these methods commit to a single bit-width across channels; the present work targets the orthogonal axis of per-channel allocation and is composable with their preprocessing or outlier-handling schemes.

Per-channel mixed-precision quantization for key cache. Three recent methods relax uniform allocation at the per-channel level for the key cache, and are our most direct point of comparison. Kitty (Xia et al., 2025) computes a per-channel magnitude average $\frac{1}{T}\sum_t |x_{t,c}|$ online and promotes a small fraction (12.5% or 25%) to 4-bit via a stored per-group boost mask. MixKVQ (Zhang et al., 2025b) maintains a sliding query window to compute a salience score $A_c = I_c \cdot S_c$ (query-activation magnitude times key scale), and assigns each channel to one of three tiers ({INT2, INT4, BF16}) via tuned thresholds. ChanMix (Liao and Wen, 2026) clusters channels by the per-channel range R_c (the same signal we use) via k-means over 1000 calibration samples, assigning outlier / normal / subnormal classes $\{3, 2, 1\}$ bits plus a 4-bit tier for retrieval-head channels, with bit-widths recorded in a channel-reorder index. All three rely on hand-designed allocation rules without an explicit optimality criterion, and each requires calibration data, online profiling, or stored indices beyond the standard min-max metadata that any quantizer already maintains. Among them, Kitty (12.5% boost, $b_{\text{eff}} = 2.125$) is the closest budget-comparable point and serves as our primary head-to-head comparison. Coarser-granularity schemes such as layer-wise Pareto search (Li et al., 2025) are orthogonal to and composable with per-channel allocators including ours.

Closed-form bit allocation in rate-distortion theory. Optimal bit allocation across parallel quantization channels under a mean-squared-error objective is a classical rate-distortion result (Shannon, 1959; Jayant and Noll, 1984; Segall, 1976), with the log-variance rule giving a closed-form solution. The same structure has been adapted to LLM weight quantization. HAWQ (Dong et al., 2019) uses Hessian eigenvalues as per-layer sensitivities for mixed-precision search, and BAQ (Zhang et al., 2025a) derives an analogous Hessian-weighted closed form. Our derivation specializes this rule to the per-channel key cache, where the appropriate sensitivity proxy reduces to R_c alone, a quantity already stored as min-max metadata, requiring neither Hessian information nor calibration. Concurrent work RateQuant (Zuo et al., 2026) develops a

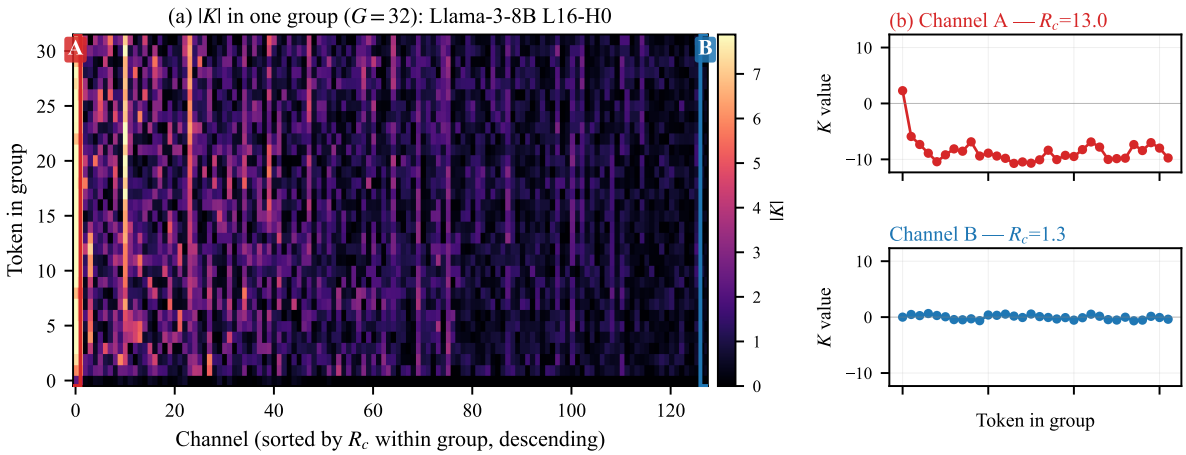


Figure 1: Per-channel range non-uniformity in Llama-3-8B Layer 16, Head 0. (a) $|K|$ over the 32 tokens of one quantization group, with channels sorted by R_c in descending order. The bright vertical stripes among the high-R_c channels on the left, set against the uniformly dark low-R_c channels on the right, show that high-R_c channels carry visibly larger values than low-R_c channels in the same group. (b) Token values within the same group for channel A (widest, $R_c = 13.0$) and channel B (narrowest, $R_c = 1.3$) on the same vertical scale. In this group $\max R_c / \min R_c \approx 10.0$ and $\sigma(\log_2 R_c) = 0.68$; the pattern holds across all 32 layers with median $\sigma(\log_2 R_c) = 0.40$.

rate-distortion formulation for KV cache at the *per-head* granularity, with a gradient-norm sensitivity that requires a calibration pass. Ours operates per-channel within a group, and the two granularities are composable.

3 Motivation: Empirical Analysis

Before deriving our allocation rule, we establish two empirical observations about the key cache on Llama-3-8B (Grattafiori et al., 2024) with WikiText-2 (Merity et al., 2017). First, per-channel ranges within a single quantization group vary by roughly $5\times$, and quadratic error scaling concentrates most of the quantization error on a small fraction of high-range channels. Second, upgrading only the highest-range channels to 3 bits recovers most of the 2-to-3-bit perplexity gap, while the same budget spent on low-range channels recovers very little. Together these two observations motivate the search for a principled allocation rule. We use Llama-3-8B in the main text and defer the corresponding Qwen2-7B and Mistral-7B analysis to Appendix A.

3.1 Per-Channel Range Non-Uniformity

We compute the per-channel range $R_c = k_{\max,c} - k_{\min,c}$ within each quantization group of $G = 32$ consecutive tokens and examine its distribution across the $d = 128$ channels of a representative head, head 0 of layer 16. The key cache is extracted on a 200-token WikiText-2 sample.

Figure 1 visualizes one such group. Panel (a) is the $|K|$ heatmap with channels sorted by R_c in descending order, where the bright stripes among the leftmost high-R_c channels, contrasted with the uniformly dark low-R_c channels on the right, make the within-group non-uniformity immediately apparent. Panel (b) zooms into the widest- and narrowest-range channels on the same vertical scale, where channel A spans roughly an order of magnitude in K value while channel B is tightly concentrated near zero. Under uniform b-bit allocation, the quantization error on channel c scales as $R_c^2/(2^b-1)^2$, so a small fraction of high-range channels dominates the overall error. Across the three models, the widest 25% of channels account for 47–51% of the total $\sum_c R_c^2$ (Appendix A).

3.2 Disproportionate Benefit of Targeted Bit Upgrade

The range non-uniformity above suggests that allocating more bits to high-range channels may help, and we verify this directly with a pilot experiment. Starting from a uniform 2-bit key cache baseline (per-channel scaling as in KIVI), we upgrade a fraction α of channels from 2 bits to 3 bits and measure WikiText-2 perplexity. We compare three selection strategies. Top-range upgrades the highest-R_c channels, bottom-range upgrades the lowest-R_c channels, and random selects channels uniformly

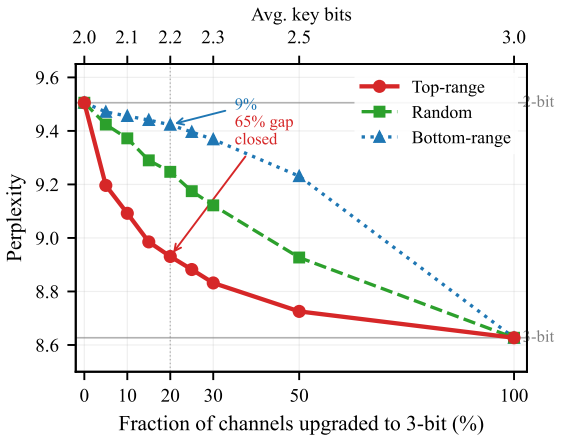


Figure 2: Pilot experiment on Llama-3-8B. WikiText-2 perplexity as a function of the fraction of channels upgraded from 2 to 3 bits, under three channel-selection strategies. At a 20% upgrade fraction (effective average key bits = 2.2), top-range recovers 65% of the 2-to-3-bit gap while bottom-range recovers only 9%.

at random (averaged over 3 seeds). Perplexity is evaluated on the full WikiText-2 test set with non-overlapping 2048-token windows.

Figure 2 shows the result. At $\alpha = 20\%$, an effective average key budget of 2.2 bits (a 10% bit-budget increase over uniform 2-bit), the top-range strategy recovers 65% of the 2-to-3-bit perplexity gap, random recovers 29%, and bottom-range recovers only 9%. The asymmetry is stark, and indicates that bit allocation guided by channel range is qualitatively more efficient than uniform allocation.

This raises the natural question of what the *optimal* per-channel allocation is. We answer it in the next section.

4 Theoretical Framework

In this section, we formalize per-channel bit allocation as a minimax problem over the worst-case attention score error variance and derive a closed-form solution. We begin with notation and preliminaries on KV cache quantization, analyze how key quantization error propagates through the attention mechanism, and finally state the allocation that minimizes the worst-case output degradation.

4.1 Preliminaries

Attention Mechanism. We follow the standard multi-head attention formulation (Vaswani et al., 2017). Consider a single attention head with per-head dimension d. At a given decoding step, the query vector $\mathbf{q} \in \mathbb{R}^d$ attends over all T cached keys $K = [\mathbf{k}_1, \ldots, \mathbf{k}_T]^\top \in \mathbb{R}^{T\times d}$ and values $V = [\mathbf{v}_1, \ldots, \mathbf{v}_T]^\top \in \mathbb{R}^{T\times d}$, where $t \in \{1, \ldots, T\}$ indexes cached tokens. The attention scores and output are computed as:

$$s_t = \frac{\mathbf{q}^\top \mathbf{k}_t}{\sqrt{d}}, \quad w_t = \frac{\exp(s_t)}{\sum_j \exp(s_j)}, \quad \mathbf{o} = \sum_t w_t \mathbf{v}_t \tag{1}$$

We use $c \in \{1, \ldots, d\}$ to index channels (dimensions) within a head.

Per-Channel Group Quantization. Following KIVI (Liu et al., 2024), the key cache is quantized per-channel with group size G along the sequence dimension. For a group of G consecutive tokens, each channel c is independently quantized using uniform min-max quantization. Let $k_{\min,c}$ and $k_{\max,c}$ denote the minimum and maximum values of channel c within the group. The channel range is $R_c = k_{\max,c} - k_{\min,c}$, and the quantization step size is $\Delta_c = R_c/(2^{b_c} - 1)$, where b_c is the bit-width assigned to channel c. KIVI stores the metadata pair $(k_{\min,c}, \Delta_c)$ per channel per group and uses a fixed $b_c = b$ for all channels.

Quantization Error Model. Modeling the quantization error $\delta k_{t,c}$ as approximately uniform over $[-\Delta_c/2, +\Delta_c/2]$, its variance is:

$$\sigma_c^2 \triangleq \mathrm{Var}(\delta k_{t,c}) = \frac{\Delta_c^2}{12} = \frac{R_c^2}{12(2^{b_c} - 1)^2} \tag{2}$$

We further assume that quantization errors are independent across channels and tokens.

4.2 Score Error Propagation

We now analyze how per-channel key quantization errors affect the attention output. When the key $\mathbf{k}_t$ is replaced by its quantized version $\hat{\mathbf{k}}_t = \mathbf{k}_t + \delta\mathbf{k}_t$, the attention score incurs an error:

$$\delta s_t = \frac{\mathbf{q}^\top \delta\mathbf{k}_t}{\sqrt{d}} = \frac{1}{\sqrt{d}} \sum_{c=1}^{d} q_c \cdot \delta k_{t,c} \tag{3}$$

Proposition 1 (Output Error via Score Perturbation)**.** *Under first-order Taylor expansion of the softmax, the attention output error is* $\delta\mathbf{o} \approx \sum_t w_t(\delta s_t - \bar{\delta s})\,\mathbf{v}_t$, *where* $\bar{\delta s} = \sum_j w_j \delta s_j$ *is the attention-weighted mean score error. This follows from* $\delta w_t = w_t(\delta s_t - \bar{\delta s})$, *the standard softmax Jacobian.*

Corollary 1. *The squared output error is bounded by:*

$$\|\delta \mathbf{o}\|^2 \leq \|\mathbf{v}_{\max}\|^2 \cdot \mathrm{Var}_w(\delta s) \tag{4}$$

where $\mathrm{Var}_w(\delta s) = \sum_t w_t(\delta s_t - \bar{\delta s})^2$ *is the attention-weighted variance of score errors, and* $\|\mathbf{v}_{\max}\|^2 = \max_t \|\mathbf{v}_t\|^2$. *This follows by applying the triangle inequality to Proposition 1, then Jensen's inequality for* $x \mapsto x^2$ *to the weighted average* $\sum_t w_t|\delta s_t - \bar{\delta s}|$.

This establishes that minimizing the score error variance minimizes the upper bound on attention output degradation. We therefore adopt $\mathbb{E}[\mathrm{Var}_w(\delta s)]$, the expectation taken over the quantization noise, as our optimization objective.

4.3 Optimal Bit Allocation

Objective. Given d channels and a total bit budget $B \cdot d$ (where B is the target average bits per channel), we seek the allocation $\{b_c\}_{c=1}^d$ that minimizes the score error variance. Because the keys are quantized before the attending queries are known, we optimize for the worst-case query:

$$\min_{\{b_c\}} \max_{\|\mathbf{q}\|=1} \mathbb{E}\big[\mathrm{Var}_w(\delta s)\big] \quad \text{s.t.} \quad \sum_{c=1}^{d} b_c = B \cdot d \tag{5}$$

Decomposition. Since quantization errors are independent across channels (§4.1), the variance of the score error (Eq. 3) for a single token decomposes as:

$$\mathrm{Var}(\delta s_t) = \frac{1}{d} \sum_{c=1}^{d} q_c^2 \cdot \sigma_c^2 \tag{6}$$

Because the errors are also independent across tokens, $\mathrm{Var}_w(\delta s) = \sum_t w_t(\delta s_t - \bar{\delta s})^2$ has expectation $(1 - \sum_t w_t^2)\,\mathrm{Var}(\delta s_t)$. Because the attention weights satisfy $w_t \geq 0$ and $\sum_t w_t = 1$, we have $\sum_t w_t^2 \leq 1$, hence $\mathbb{E}[\mathrm{Var}_w(\delta s)] \leq \mathrm{Var}(\delta s_t)$ and it suffices to minimize this upper bound. Substituting the error model (Eq. 2) gives:

$$\mathrm{Var}(\delta s_t) \propto \sum_{c=1}^{d} q_c^2 \cdot \frac{R_c^2}{(2^{b_c} - 1)^2} \tag{7}$$

The minimax optimum equalizes all channel terms regardless of the precise error model (shown below). To turn this equalization condition into a closed form, we adopt the standard high-resolution approximation $(2^{b_c} - 1)^2 \approx 4^{b_c}$, exact as $b_c \to \infty$ and differing by a factor of 0.56 at $b_c = 2$:

$$\mathrm{Var}(\delta s_t) \propto \sum_{c=1}^{d} q_c^2 \cdot \frac{R_c^2}{4^{b_c}} \tag{8}$$

Worst-case query. The inner maximization over unit-norm queries yields:

$$\max_{\|\mathbf{q}\|=1} \sum_c q_c^2 \cdot \frac{R_c^2}{4^{b_c}} = \max_c \frac{R_c^2}{4^{b_c}} \tag{9}$$

since the maximum of a linear function over the simplex $\{q_c^2 : \sum_c q_c^2 = 1\}$ is attained at the vertex corresponding to the largest coefficient.

Solution. Combining the bound on $\mathbb{E}[\mathrm{Var}_w(\delta s)]$ with the worst-case query, the allocation problem becomes

$$\min_{\{b_c\}} \max_c \frac{R_c^2}{4^{b_c}} \quad \text{s.t.} \quad \sum_c b_c = Bd, \tag{10}$$

which we solve over continuous $b_c \in \mathbb{R}$. The minimum of the maximum is achieved when all terms are equal, since otherwise transferring bits to the largest term from a smaller one strictly reduces the maximum. Setting $R_c^2/4^{b_c} = \lambda$ for all c gives $b_c = \log_2 R_c - \frac{1}{2}\log_2 \lambda$; substituting into the budget $\sum_c b_c = Bd$, solving for λ, and substituting back yields the closed-form allocation

$$\boxed{b_c = B + \log_2\left(\frac{R_c}{\bar{R}_{\text{geo}}}\right)} \tag{11}$$

where $\bar{R}_{\text{geo}} = \left(\prod_{c=1}^{d} R_c\right)^{1/d}$ is the geometric mean of per-channel ranges.

The allocation is *sum-preserving* in the continuous regime ($\sum_c \log_2(R_c/\bar{R}_{\text{geo}}) = 0$ by the geometric-mean definition), *range-adaptive* (channels with $R_c > \bar{R}_{\text{geo}}$ receive $b_c > B$, matching the empirical signal in §3.1), and *zero-overhead* (it depends only on $\{R_c\}$, already maintained as min-max metadata).

5 Method: SVEQ

We now instantiate the optimal allocation (Eq. 11) as a practical quantizer. This requires rounding the continuous b_c to integer bit-widths and, more importantly, deploying the per-channel allocation without storing a bit-width index. We describe the algorithm and then this zero-overhead deployment mechanism.

Algorithm 1 SVEQ Key Cache Quantization (per group)

Require: Key group $K \in \mathbb{R}^{G \times d}$, target B, bounds $[b_{\min}, b_{\max}] = [2, 3]$
Ensure: Packed quantized keys K^Q and per-channel metadata $\{(k_{\min,c}, k_{\max,c})\}$
for each channel $c = 1, \ldots, d$ **do**
  $k_{\min,c} \leftarrow \min_t K_{t,c}$; $k_{\max,c} \leftarrow \max_t K_{t,c}$
  $R_c \leftarrow k_{\max,c} - k_{\min,c}$
end for
$\bar{R}_{\text{geo}} \leftarrow \exp\big(\frac{1}{d} \sum_c \ln R_c\big)$
for each channel $c = 1, \ldots, d$ **do**
  $b_c^* \leftarrow B + \log_2(R_c / \bar{R}_{\text{geo}})$
  $b_c \leftarrow \text{round}(\text{clamp}(b_c^*, b_{\min}, b_{\max}))$
  $\Delta_c \leftarrow R_c \,/\, (2^{b_c} - 1)$
  $K^Q_{:,c} \leftarrow \text{round}((K_{:,c} - k_{\min,c}) \,/\, \Delta_c)$ // pack into b_c bits
end for
Store: K^Q, $\{(k_{\min,c}, k_{\max,c})\}_{c=1}^d$

5.1 Algorithm

Algorithm 1 shows the quantization procedure. Following KIVI, we partition the key cache along the sequence dimension into groups of $G = 32$ tokens and keep the most recent 128 tokens (residual buffer) in FP16. Within each group, we compute the per-channel range R_c (already required for any min-max quantizer), apply the allocation formula with target $B = 2$ and clamping to $[b_{\min}, b_{\max}] = [2, 3]$, and quantize each channel at its assigned bit-width. We keep $b_{\min} = 2$ rather than 1 because demoting low-range channels to 1-bit costs more accuracy than the added 3-bit channels recover.

Bit-widths are recomputed, not stored. SVEQ stores two FP16 scalars per channel, the same count as any min-max quantizer; we store $(k_{\min,c}, k_{\max,c})$ rather than KIVI's $(k_{\min,c}, \Delta_c)$ because the former lets the dequantizer recover b_c. The dequantizer recomputes $R_c = k_{\max,c} - k_{\min,c}$, then $\bar{R}_{\text{geo}}$, then b_c and Δ_c via the same formulas used at quantization time. Because $\bar{R}_{\text{geo}}$ depends only on $\{R_c\}_{c=1}^d$ within the group, the recomputation is deterministic and exact, and the $O(d)$ overhead is negligible relative to the $O(T \cdot d)$ attention computation. **No per-channel bit-width index is stored.**

Value cache. SVEQ modifies only the key cache quantization. The value cache uses standard per-token 2-bit quantization with group size $G = 32$ along the head dimension.

5.2 Effective Storage Cost

The metadata stored per group is two FP16 scalars per channel, identical in size to KIVI's $(k_{\min,c}, \Delta_c)$ pair, plus the variable-width packed integers. Clamping the continuous allocation to integer $\{2, 3\}$ breaks the strict sum-preservation of the unclamped rule. Channels with $R_c < \bar{R}_{\text{geo}}$ would receive $b_c < B$ in the continuous regime but are clamped upward to 2, so the realized per-channel average $\bar{b}_K$ exceeds the target $B = 2$.

Asymptotic effective bit rate. In the long-context limit, where the FP16-retained tokens become negligible relative to the quantized region, the per-element bit rate, averaged across K and V cache and counting only the algorithm's payload (following the KIVI / Kitty / MixKVQ / ChanMix convention of excluding the shared per-group min-max metadata), reduces to

$$b_{\text{eff}} = \tfrac{1}{2}\big(\bar{b}_K + b_V\big) = \tfrac{1}{2}\bar{b}_K + 1, \qquad (12)$$

with $b_V = 2$ in our protocol. Because $\bar{b}_K$ is determined by the empirical channel-range distribution rather than a closed-form quantity, we measure it directly over the full WikiText-2 test split ($\geq 10^8$ channel–group cells per model).

The realized rate clusters between $\bar{b}_K = 2.176$ and 2.220 across the four models, corresponding to $b_{\text{eff}} = 2.088$–2.110, with 17.6–22.0% of channels clamp-rounded to 3-bit. Under identical accounting, KIVI 2-bit is exactly 2.00 and Kitty (12.5% INT4) is 2.125, so SVEQ sits within the same effective-bit-rate band as the uniform 2-bit baseline and prior per-channel methods; the accuracy comparisons in §6 are therefore at matched budget. This b_{eff} is also SVEQ's complete per-channel cost. Kitty additionally stores a per-group boost mask ($\sim 1/G$ bit per K element), MixKVQ a per-channel tier label, and ChanMix a channel-reorder index, whereas SVEQ recomputes b_c from R_c at dequantization time and stores nothing beyond the min-max metadata.

6 Experiments

6.1 Setup

Models and tasks. We evaluate on four open-source LLMs: **Llama-3-8B**, **Qwen2-7B** (Yang et al., 2024), **Mistral-7B** (v0.2) (Jiang et al., 2023),

and **Llama-3-70B**. Tasks requiring instruction-following (GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021), Needle-in-a-Haystack (Kamradt, 2023), LongBench (Bai et al., 2024)) use the corresponding Instruct variants. WikiText-2 perplexity and MMLU (Hendrycks et al., 2021) use the base models. Long-context retrieval extends to each model's maximum context window: 8K (Llama-3-8B-Instruct and Llama-3-70B-Instruct), 32K (Mistral-7B-Instruct-v0.2), and 128K (Qwen2-7B-Instruct). GSM8K and HumanEval are scored through lm-evaluation-harness following Kitty's official protocol, so the comparison is like-for-like. GSM8K uses the 8-shot chain-of-thought task `gsm8k_cot_llama` with flexible-extract exact match, and HumanEval the 0-shot `humaneval_instruct` task with code-fence answer extraction. Both decode greedily.

Baselines. For fair comparison, all four configurations share the same protocol, with group size $G = 32$, a 32-token attention sink, and a 128-token FP16 residual buffer for the most recent tokens; they differ only in the key quantization rule. We compare (i) **FP16** (uncompressed); (ii) **KIVI 2-bit** (Liu et al., 2024), the standard 2-bit baseline with per-channel keys and per-token values; (iii) **Kitty (12.5% INT4)** (Xia et al., 2025), our primary mixed-precision baseline, which ranks key channels by magnitude and promotes 12.5% to INT4; and (iv) our **SVEQ 2-bit** with $b_c \in \{2, 3\}$ at target $B = 2$. The effective bit rate (Eq. 12) is 2.000 for KIVI, 2.125 for Kitty, and 2.088/2.090/2.092/2.110 for SVEQ on Llama-3-8B / Qwen2-7B / Mistral-7B / Llama-3-70B, so SVEQ is compared at a budget at or below Kitty's throughout.

Implementation. The accuracy experiments use a fake-quantization framework on top of the KIVI codebase: K and V are quantized in the cache update path and dequantized to FP16 before attention. Experiments are run on NVIDIA A800 80GB GPUs; the 70B model uses NF4 weight quantization (Dettmers et al., 2023) for fit. End-to-end efficiency is measured separately with the Triton kernel of §6.4.

6.2 Short-Context Accuracy

Table 1 reports accuracy on the four short-context benchmarks. On **WikiText-2**, SVEQ closes 38–48% of the FP16–KIVI perplexity gap on every model and reaches Kitty's accuracy within ± 0.1 PPL, at a lower effective bit rate. On **MMLU**, SVEQ closes roughly 49–60% of the FP16–KIVI gap on every model, also at lower b_{eff} than Kitty. On **GSM8K**, SVEQ stays at or above KIVI on every model and exactly matches FP16 on Qwen2-7B (84.46); on Llama-3-70B the three compressed methods all land within 1 point of FP16, within sampling noise. On **HumanEval**, SVEQ outperforms Kitty on all four models at a lower bit rate, and reaches or exceeds FP16 on Llama-3-8B (59.15 vs. 57.93) and Qwen2-7B (68.29, tied). It trails FP16 by only 1.8 and 1.2 points on Mistral-7B and Llama-3-70B; greedy decoding makes the evaluation deterministic, so a gap of one or two of the 164 problems carries no statistical weight.

Model	Method	PPL	MMLU	GSM8K	HE
Llama-3-8B	FP16	8.284	65.08	77.71	57.93
	KIVI 2-bit	8.819	63.51	75.59	57.32
	SVEQ 2-bit	8.600	64.29	77.18	59.15
	Kitty (12.5%)	8.642	64.07	77.71	56.71
Qwen2-7B	FP16	7.602	68.79	84.46	68.29
	KIVI 2-bit	8.290	67.30	82.34	67.07
	SVEQ 2-bit	8.024	68.06	84.46	68.29
	Kitty (12.5%)	8.043	68.52	83.78	66.46
Mistral-7B	FP16	5.942	57.70	41.85	43.90
	KIVI 2-bit	6.234	56.74	40.41	42.07
	SVEQ 2-bit	6.095	57.32	40.79	42.07
	Kitty (12.5%)	6.126	57.21	41.85	41.46
Llama-3-70B	FP16	5.768	76.98	90.52	75.61
	KIVI 2-bit	6.328	76.12	90.83	72.56
	SVEQ 2-bit	6.117	76.54	90.90	74.39
	Kitty (12.5%)	6.200	76.61	91.51	73.17

Table 1: Accuracy on short-context benchmarks. WikiText-2 perplexity uses the non-overlapping 2048-token protocol on base models; MMLU is 5-shot on base models; GSM8K is 8-shot CoT and HumanEval (HE, pass@1) is 0-shot, both on Instruct variants. Effective bit rates are listed in §6.1.

6.3 Long-Context Accuracy

Table 2 reports long-context retrieval and downstream accuracy. On **Needle-in-a-Haystack**, SVEQ matches FP16 within 0.25 points at every model's maximum context, recovering $\sim$1.2 points over KIVI at 32K and 128K. This suggests range-adaptive allocation preserves the high-range key channels that mediate long-range retrieval, where uniform 2-bit allocation begins to break down. On **LongBench**, SVEQ matches FP16 within ± 0.1 points on all four models and improves over KIVI on every one. SVEQ reaches Kitty's overall accuracy at a lower effective bit rate, and surpasses

Model / Max Ctx	Method	NIAH	LongBench
Llama-3-8B-Inst. / 8K	FP16	99.25	42.46
	KIVI 2-bit	99.25	42.01
	SVEQ 2-bit	99.25	42.43
	Kitty (12.5%)	99.25	42.16
Mistral-7B-Inst. / 32K	FP16	100.0	38.25
	KIVI 2-bit	98.83	38.03
	SVEQ 2-bit	100.0	38.27
	Kitty (12.5%)	100.0	38.28
Qwen2-7B-Inst. / 128K	FP16	100.0	43.45
	KIVI 2-bit	98.50	42.93
	SVEQ 2-bit	99.75	43.47
	Kitty (12.5%)	99.88	43.34
Llama-3-70B-Inst. / 8K	FP16	99.75	45.85
	KIVI 2-bit	99.75	45.74
	SVEQ 2-bit	99.75	45.93
	Kitty (12.5%)	99.75	45.88

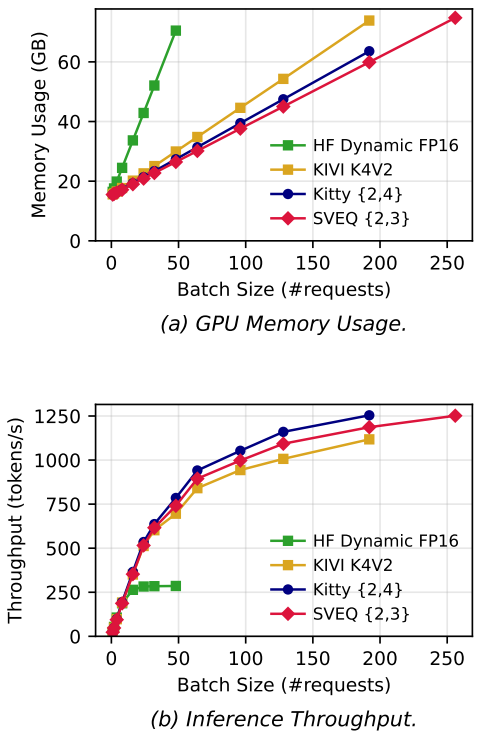
Table 2: Long-context accuracy. NIAH is Needle-in-a-Haystack retrieval accuracy (%), averaged over all (length, depth) cells up to each model's maximum context window. LongBench is the average over 12 standard tasks (NarrativeQA, Qasper, MultiFieldQA-en, 2WikiMQA, HotpotQA, MuSiQue, GovReport, QMSum, MultiNews, TREC, TriviaQA, SAMSum); per-task scores are in Appendix B.

Kitty on three of four models.

6.4 End-to-End Efficiency

To evaluate the system-level efficiency of SVEQ, we run end-to-end Qwen3-8B inference with a prototype engine that quantizes and dequantizes the KV cache through custom Triton kernels. A batch of example prompts is fed to the engine, which generates tokens until it reaches the maximum sequence length of 8192. We compare against three baselines under the same engine. HF Dynamic FP16 is the default Hugging Face KV cache, which grows dynamically as tokens are cached. KIVI K4V2 is implemented on the same Kitty kernel with all key channels assigned to 4 bit. Kitty $\{2, 4\}$ is the per-channel mixed-precision baseline of §2. KIVI K4V2, Kitty $\{2, 4\}$, and SVEQ $\{2, 3\}$ all use a per-token 2-bit V cache, so the figure isolates the key cache strategy. For each KV-cache implementation we raise the batch size until out-of-memory, and report peak memory usage and token-generation throughput. The hardware is a single NVIDIA A800 80GB GPU.

Figure 3 reports the sweep. SVEQ has the lowest memory footprint at every batch size, and on the 80GB budget it scales to a batch of 256, which is 1.33× the largest batch Kitty sustains and 5.3× that of FP16. Its throughput in the large-batch regime reaches 1252 tokens/s, 4.4× the FP16 baseline and on par with Kitty's peak, while its memory stays below Kitty's throughout the sweep. We implemented the SVEQ kernels in Triton as a proof-of-concept, so a lower-level CUDA implementation with finer-grained optimization would improve efficiency further, which we leave to future work.



(a) GPU Memory Usage.

(b) Inference Throughput.

Figure 3: End-to-end Qwen3-8B inference at an 8192-token sequence length on a single A800 80GB GPU. Panel (a) shows peak GPU memory and panel (b) token-generation throughput, with the batch size swept up to each method's out-of-memory point.

7 Conclusion

We derived a closed-form per-channel bit allocation $b_c = B + \log_2(R_c/\bar{R}_{\text{geo}})$ from a minimax bound on attention-output error, and instantiated it as SVEQ, a practical key cache quantizer. The allocation depends only on the per-channel range R_c that a min-max quantizer already stores, and therefore needs no calibration data, online profiling, or auxiliary index. At ∼2.09 effective bits per element, SVEQ stays within 2 absolute points of FP16 across six benchmarks on four LLMs, matching the accuracy of heuristic per-channel methods at a comparable bit rate without their deployment overhead. SVEQ thus offers a principled, zero-overhead alternative to the heuristic allocators currently used for per-channel key cache mixed precision.

Limitations

- **Magnitude of improvement.** SVEQ closes 38–48% of the FP16–KIVI perplexity gap and roughly 49–60% of the MMLU gap, so a residual gap to FP16 remains at the $\sim$2.09-bit budget. Closing it further requires a higher bit budget, an operating point we do not target.

- **Storage overhead over KIVI.** SVEQ's realized $b_{\text{eff}} = 2.09$–2.11 carries an overhead of about 0.1 bit over KIVI's exact 2.00. Reaching exactly 2.00 would require demoting low-range channels to 1-bit (e.g., via a largest-remainder allocator over $\{1, 2, 3\}$); we explored this and found the 1-bit demotion harmful, costing 0.45 MMLU and 2.44 HumanEval points on Llama-3-70B relative to our $\{2, 3\}$ clamp, which motivates $b_{\min} = 2$.

- **Empirical comparison scope.** Our head-to-head experiments compare against KIVI and the closest budget-comparable per-channel method, Kitty. We do not benchmark MixKVQ (Zhang et al., 2025b), ChanMix (Liao and Wen, 2026), or concurrent per-head RateQuant (Zuo et al., 2026) directly, as they operate at different granularities or budgets; we contrast them qualitatively in §2.

- **Composition with rotation.** SVEQ's gain is contingent on per-channel range non-uniformity. Applying a Hadamard rotation first (e.g., QuaRot (Ashkboos et al., 2024)) equalizes R_c and degenerates SVEQ to uniform 2-bit allocation; per-channel allocation and rotation address the same cause, and combining them is left as future work.

- **Kernel evaluation scope.** The efficiency study (§6.4) benchmarks our Triton kernel on a single model and a single sequence length. A broader sweep and a lower-level CUDA implementation are left to future work.

Potential Risks

SVEQ is a quantization method that reduces the deployment cost of existing large language models without altering their behavior or introducing new capabilities. The technique inherits any risks already associated with the underlying models, including the generation of harmful, biased, or misleading content. By lowering the memory footprint of long-context serving, the method may also reduce the cost barrier to large-scale LLM deployment. This is neutral with respect to deployment intent. We are not aware of risks that are specific to SVEQ beyond those already present in the underlying models.

References

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. 2024. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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. 2024. LongBench: A bilingual, multi-task benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2019. HAWQ: Hessian AWare quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 293–302.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. 2024. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Nuggehally S Jayant and Peter Noll. 1984. *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Prentice Hall.

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. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Greg Kamradt. 2023. Needle in a haystack – pressure testing LLMs. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`.

Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. 2024. GEAR: An efficient KV cache compression recipe for near-lossless generative inference of LLM. *arXiv preprint arXiv:2403.05527*.

Xing Li, Zeyu Xing, Yiming Li, Linping Qu, Hui-Ling Zhen, Wulong Liu, Yiwu Yao, Sinno Jialin Pan, and Mingxuan Yuan. 2025. KVTuner: Sensitivity-aware layer-wise mixed-precision KV cache quantization for efficient and nearly lossless LLM inference. *arXiv preprint arXiv:2502.04420*.

Chengxi Liao and Zeyi Wen. 2026. Channel-aware mixed-precision quantization for efficient long-context inference. In *International Conference on Learning Representations (ICLR)*.

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. 2024. QServe: W4A8KV4 quantization and system co-design for efficient LLM serving. *arXiv preprint arXiv:2405.04532*.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*.

Adrian Segall. 1976. Bit allocation and encoding for vector sources. *IEEE Transactions on Information Theory*, 22(2):162–169.

Claude E Shannon. 1959. Coding theorems for a discrete source with a fidelity criterion. *IRE National Convention Record*, 7:142–163.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Haojun Xia, Xiaoxia Wu, Jisen Li, Robert Wu, Junxiong Wang, Jue Wang, Chenxi Li, Aman Singhal, Alay Dilipbhai Shah, Alpay Ariyak, Donglin Zhuang, Zhongzhu Zhou, Ben Athiwaratkun, Zhen Zheng, and Shuaiwen Leon Song. 2025. Kitty: Accurate and efficient 2-bit KV cache quantization with dynamic channel-wise precision boost. *arXiv preprint arXiv:2511.18643*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Chao Zhang, Li Wang, Samson Lasaulce, and Merouane Debbah. 2025a. BAQ: Efficient bit allocation quantization for large language models. *arXiv preprint arXiv:2506.05664*.

Tao Zhang, Ziqian Zeng, Hao Peng, Huiping Zhuang, and Cen Chen. 2025b. MixKVQ: Query-aware mixed-precision KV cache quantization for long-context reasoning. *arXiv preprint arXiv:2512.19206*.

Fei Zuo, Zikang Zhou, Hao Cong, Xiaoyan Xi, and Ho Fai Leung. 2026. RateQuant: Optimal mixed-precision KV cache quantization via rate-distortion theory. *arXiv preprint arXiv:2605.06675*.

A Range Non-Uniformity Across Models and Layers

Section 3.1 establishes per-channel range non-uniformity on a single representative head (Llama-3-8B, Layer 16, Head 0). Here we verify that the property holds across every layer of three models. For Head 0 of each layer of Llama-3-8B, Qwen2-7B, and Mistral-7B, we compute the per-channel range R_c over the full WikiText-2 sample and summarize its spread by $\sigma(\log_2 R_c)$, the standard deviation of the log-ranges, and by the channel-range ratio $\max_c R_c / \min_c R_c$. Because uniform b-bit error scales as R_c^2, we additionally report the share of the total $\sum_c R_c^2$ contributed by the widest-range 25% of channels, a direct measure of how concentrated the quantization error is.

Granularity. The main-text analysis (§3.1) measures R_c within a single $G = 32$ quantization group, whereas the per-layer statistics here are computed over the full sample. The two granularities agree closely. On Llama-3-8B Layer 16 Head 0, the full-sample $\sigma(\log_2 R_c) = 0.66$ lies within 0.01 of the group-wise median 0.67 taken over the six groups of that head. The full-sample statistic is therefore a faithful and convenient proxy for the per-group quantity that the allocation rule actually consumes.

Results. Figure 4 plots $\sigma(\log_2 R_c)$ against layer index for all three models, and Table 3 reports per-layer medians. Range non-uniformity is present in every layer of every model. The per-layer median $\sigma(\log_2 R_c)$ is 0.40 for Llama-3-8B and Qwen2-7B and 0.47 for Mistral-7B, and no layer falls below 0.29. The widest-to-narrowest channel-range ratio has a per-layer median of 4.7–$5.8\times$ across the three models, and reaches $47.7\times$ in individual layers. Because uniform b-bit error scales as R_c^2, this range spread concentrates the quantization error. At the per-layer median, the widest 25% of channels carry 47–51% of the total $\sum_c R_c^2$, while individual layers reach 65–81%. A quarter of the key channels therefore account for roughly half of all quantization error. This confirms that the range-adaptive allocation of SVEQ is motivated by a structural property shared across models and layers, rather than an artifact of the single head visualized in Figure 1.

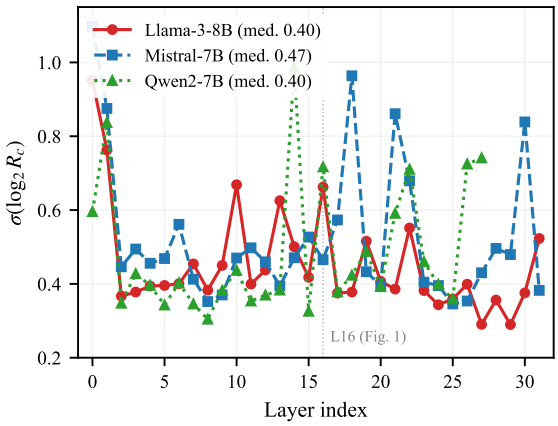


Figure 4: Per-channel range spread $\sigma(\log_2 R_c)$ for Head 0 of every layer of three models, computed over the full WikiText-2 sample. Every layer of every model exhibits substantial non-uniformity; the dotted line marks Llama-3-8B Layer 16, the head visualized in Figure 1.

Model	Layers	$\sigma(\log_2 R_c)$	Range ratio	Err. share
Llama-3-8B	32	0.40	$5.5\times$	47%
Mistral-7B	32	0.47	$5.8\times$	51%
Qwen2-7B	28	0.40	$4.7\times$	48%

Table 3: Per-layer range-spread statistics over Head 0 of each layer, computed on the full WikiText-2 sample; each cell is the median across layers. "Range ratio" is the within-layer $\max_c R_c / \min_c R_c$; "Err. share" is the fraction of the total $\sum_c R_c^2$ (proportional to uniform-quantization error) carried by the widest-range 25% of channels. Cross-layer ranges are reported in the text.

B Complete LongBench Results

Table 4 reports the per-task LongBench scores for all 12 datasets across the four models and four KV-cache configurations, expanding the LongBench column of Table 2.

Method	Single-Doc QA			Multi-Doc QA			Summarization			Few-shot			
	Nrtv	Qsp	MFld	2Wik	Hpot	Msq	GvRp	QMSm	MNws	TREC	Triv	SmSm	**Avg**
Llama-3-8B-Instruct													
FP16	20.92	42.96	47.78	39.10	47.32	23.28	30.46	22.71	27.56	74.50	90.23	42.66	42.46
KIVI 2-bit	20.69	42.31	44.95	39.11	47.38	22.18	30.36	22.44	27.46	74.50	90.38	42.35	42.01
SVEQ 2-bit	21.25	42.50	47.39	39.76	47.31	22.90	30.61	22.35	27.36	74.50	90.31	42.92	42.43
Kitty (12.5%)	20.71	43.00	46.37	38.98	47.28	22.61	30.22	22.32	27.32	74.50	90.21	42.37	42.16
Mistral-7B-Instruct													
FP16	21.95	29.30	47.65	21.89	37.52	18.36	31.41	23.91	26.72	71.00	86.23	43.01	38.25
KIVI 2-bit	21.83	29.03	46.72	20.90	36.64	18.08	31.37	23.92	26.82	71.00	86.41	43.60	38.03
SVEQ 2-bit	21.47	29.59	46.24	22.53	37.79	17.87	31.50	24.48	26.84	71.00	86.35	43.58	38.27
Kitty (12.5%)	22.54	29.84	46.68	21.31	37.44	18.22	31.38	24.16	26.92	71.00	86.42	43.44	38.28
Qwen2-7B-Instruct													
FP16	24.47	46.56	46.81	43.34	43.64	25.39	32.44	22.97	24.60	77.00	89.41	44.80	43.45
KIVI 2-bit	24.19	45.92	45.43	42.48	43.52	23.57	31.73	21.92	24.09	77.50	89.72	45.12	42.93
SVEQ 2-bit	23.91	46.69	46.68	42.72	44.43	25.73	31.73	23.08	24.26	77.00	90.06	45.35	43.47
Kitty (12.5%)	24.03	45.89	46.27	43.80	44.36	23.63	31.94	22.91	24.29	77.00	90.06	45.89	43.34
Llama-3-70B-Instruct													
FP16	27.74	48.60	49.86	58.15	48.96	28.26	31.62	22.36	26.81	72.00	91.21	44.63	45.85
KIVI 2-bit	27.82	48.34	49.79	58.40	47.78	28.67	31.10	21.99	26.83	72.00	91.16	45.01	45.74
SVEQ 2-bit	28.35	48.05	51.41	58.77	48.29	28.25	31.19	22.02	26.67	72.00	90.99	45.22	45.93
Kitty (12.5%)	28.23	48.53	50.82	58.63	48.21	28.34	31.08	21.98	26.69	72.00	91.16	44.86	45.88

Table 4: Complete per-task LongBench results across all 12 datasets, for the four models and four KV-cache configurations of §6.3. The **Avg** column is the 12-task mean and reproduces the LongBench column of Table 2. Task abbreviations: Nrtv (NarrativeQA), Qsp (Qasper), MFld (MultiFieldQA-en), 2Wik (2WikiMQA), Hpot (HotpotQA), Msq (MuSiQue), GvRp (GovReport), QMSm (QMSum), MNws (MultiNews), TREC, Triv (TriviaQA), SmSm (SAMSum).