A SURROGATE DISTRIBUTION PERSPECTIVE ON KEY-VALUE CACHE MERGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Merging entries in the key-value (KV) cache of large language models is a promising approach for managing GPU memory and throughput, and hence scaling to larger context windows. However, naively merging entries changes the scaled dot-product attention output at each layer, and if done aggressively seriously degrades accuracy on retrieval and reasoning tasks. In this paper we observe that the exact contribution of a merged cluster to attention is the cumulant generating function of the distribution of its constituent key norms, together with the mean value under the exponential tilting of that distribution induced by the query. Compensating for a merge is therefore a question of which surrogate distribution to substitute for the cluster, and we show that a two-atom surrogate matching the cluster's first two key-norm moments yields closed-form corrections to both the attention score and the merged value, at a cost of two scalars and one vector per merged cache slot. We include in our analysis the Gaussian surrogate implicit in a second-order Taylor expansion of the attention score function. The resulting corrections are exact for two-token merges and for every query, and reduce exactly to standard attention for entries that have never been merged. Moreover, the statistics that our corrections depend on are invariant to the order in which merges are performed, so clusters can grow online as new tokens are processed. Experiments validate our theory by measuring fidelity of the induced corrections and sweeping performance on standard reasoning and retrieval tasks against merge threshold.

1 INTRODUCTION

Running a large language model over a long context window is increasingly a memory problem rather than a compute problem. The key-value (KV) cache, a per-layer record of every processed token's key and value, grows linearly with context length, and at the context windows now routine for reasoning and retrieval tasks it dominates GPU memory and decode throughput. A large body of literature therefore attempts to manage the size of the cache. Eviction schemes discard entries outright, which is simple but irrecoverable; merging schemes instead replace a set of similar entries by a single summary, retaining a lossy trace of every token. Merging, however, has a correctness problem: the non-linearity of softmax means that a merged cache entry which simply averages constituent keys and values changes the output of the attention computation itself. Specifically, it assumes the probability mass of a single token while representing many, and returns a fixed average value where the true computation is query dependent. When merging is performed aggressively, this significantly degrades accuracy on retrieval and reasoning tasks.

This paper starts from an exact identity rather than a heuristic. The contribution of a merged cluster to attention can be written in closed form: its score is, up to an additive $\log m$, the cumulant generating function of the distribution of its constituent key norms, and its value is the mean value under that distribution exponentially tilted by the query. Compensating for a merge is therefore not a matter of inventing an ad hoc correction term but of choosing a surrogate distribution to stand in for the set of merged samples. Every choice of surrogate yields a closed form correction. This framing organizes existing corrections and ours into a single ladder. A one-atom surrogate yields the count correction of KeepKV (Tian et al., 2026), the strongest published correction of this kind. A Gaussian surrogate turns out to be exactly the second-order Taylor expansion of the score function. Whereas a two-atom surrogate matching the same moments as the Gaussian yields $\log\cosh$ and $\tanh$ corrections that are

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

exact for two-token merges under every query. Applying the same two-atom construction in logit space drops the aligned-keys assumption at no additional storage.

Concretely, we make the following contributions.

- **A surrogate-distribution view of cache merging.** We show that the exact merged-slot computation is a cumulant generating function of the cluster's key-norm distribution, so that every analytic correction corresponds to a surrogate for that distribution (Sec. 3). This unifies previously published count corrections, Taylor series expansions, and our corrections. We state what each assumes, when it is exact, what it costs and what it saves.
- **Closed-form score and value corrections.** Our two-atom surrogates give corrections at the cost of two scalars and one d-vector per merged slot. They reduce exactly to standard attention for never-merged entries and are exact for two-token merges under *every* query (Props. 3.2 and 3.3), not only at the query observed when the merge is made.
- **Order-invariant online merging.** The statistics maintained for our correction formulae are additive power sums, so clusters can grow online as tokens are processed. Moreover, the resulting cache is independent of the order in which merges are performed (Prop. A.12).
- **Error analysis.** We derive error bounds on the score and value corrections. Importantly, the score error is bounded cubically in the cluster's key-norm spread, and errors provably do not accumulate across clusters (App. A.6).
- **Empirical validation.** Measured on real activations, the fidelity ordering that the ladder predicts holds, and the logit-space surrogate reduces approximation error severalfold over every key-norm-space surrogate. On sweeps of accuracy versus cache-size over two models and three tasks it is the strongest on both reasoning and retrieval (Sec. 4).

The corrections applied to merged clusters are analytic and principled. They do not require training, calibration or fine-tuning, and can be implemented inside fused FlashAttention-style kernels (Dao et al., 2022). The main body of the paper focuses on the core theory and contributions; proofs of results and extended analysis are collected in the appendix.

2 Related Work

Eviction. The dominant approach to bounding KV-cache growth is eviction: H2O (Zhang et al., 2023) retains "heavy hitter" tokens measured on accumulated attention, StreamingLLM (Xiao et al., 2024) keeps initial sink tokens and a window of recent tokens, discarding the rest, SnapKV (Li et al., 2024) selects tokens per layer based on attention weight from a short window at the end of the prompt, TOVA (Oren et al., 2024) evicts whichever token the current query attends to least, and KVzip (Kim et al., 2025) scores entries query-agnostically by their ability to reconstruct the context. A newer line of work selects tokens to evict by output sensitivity rather than attention alone (Feng et al., 2026; Goel et al., 2025). This is close to us in spirit, but it decides what to discard, whereas we ask what to compute once entries have been combined.

Merging. Merging replaces a set of entries by a summary, retaining a lossy trace of every token. Most works focus on which tokens to merge and how to weight their average, e.g., key cosine similarity (Wang et al., 2024), folding entries into retained ones (Zhang et al., 2024; Wan et al., 2025), values only (Yuan et al., 2025), within semantic chunks (Wu et al., 2026), per-head composite tokens (Akulov et al., 2025), online key clustering (Zandieh et al., 2024). GRKV (Peng et al., 2026) chooses merge weights by regressing the compressed attention output onto the full one; DMC (Nawrot et al., 2024) and Still (O'Neill et al., 2026) instead apply learning with no closed form correction. KVSlimmer (Liu et al., 2026) argues different treatment for keys and values, an asymmetry that we make explicit. In all of these, the summary is fixed at merge time, so clusters cannot adjust to unseen future queries. Outside the KV cache, ToMe (Bolya et al., 2023) merges patch tokens within a single forward pass, where the query dependency does not arise.

Correcting attention after merging. We are not the first to observe that merged entries need correction. SelKV (Bouyahiaoui et al., 2026) uses the term "attention sag" for the gap between computations on the (single) convex combination of merged keys and the true score. KeepKV (Tian et al., 2026) compensate for merging by adding $\log(1+m)$ to the logit of an entry that has absorbed m tokens whereas SelKV replaces the count by an attention ratio estimated from prefill, and gates

merging on value similarity. ResKV (Zhan et al., 2026) applies the same accounting to eviction, adding residual entries that restore the discarded attention mass inside the softmax. We similarly apply corrections to merged slots. However, our starting position is to treat the merged cluster's score and value as functions of the query and approximate the functions themselves. Our corrections are exact for two-token merges on *any* query and the approximation recovers the KeepKV construction as its zeroth-order case.

Rather than proposing yet another correction, we place them in a family. The exact merged score is a cumulant generating function of the group's key distribution, so every correction amounts to choosing a surrogate for that distribution, and the choice determines the correction in closed form: count correction is a one-atom surrogate, ours a two-atom surrogate, and the second-order Taylor expansion gives a Gaussian member of the same family. This unified approach makes clear what each correction assumes, when it is exact and what it costs to store, rather than leaving the compensation term to be chosen empirically.

Quantization (Hooper et al., 2024; Zandieh et al., 2026) and low-rank projection act on the representation of each entry rather than the number of entries, and compose with merging; see Jiang et al. (2026) for these and the system-level techniques we do not consider in this paper.

3 KV Merging with Surrogate Distribution Models

Let $k_1, \ldots, k_n$ and $v_1, \ldots, v_n$ be key and value vectors already processed and stored in the KV cache. For a new query $q \in \mathbb{R}^d$, scaled dot-product attention (Vaswani et al., 2017) computes

$$a = \text{softmax}\left(\frac{q^T k_1}{\sqrt{d}}, \frac{q^T k_2}{\sqrt{d}}, \cdots, \frac{q^T k_n}{\sqrt{d}}\right) \tag{1}$$

$$y = a_1 v_1 + a_2 v_2 + \ldots + a_n v_n \tag{2}$$

where the softmax is taken over the n logits $\ell_i = q^T k_i/\sqrt{d}$ and $a \in \mathbb{R}^n$ represents the (normalized) attention weight over the value vectors for the given query.

Consider merging a set of $m < n$ key-value pairs $\{(k_i, v_i)\}_{i=1}^m$ into a single cluster. We can rewrite the above expression for scaled dot-product attention exactly as

$$\tilde{a} = \text{softmax}\left(s^\star(q), \frac{q^T k_{m+1}}{\sqrt{d}}, \cdots, \frac{q^T k_n}{\sqrt{d}}\right) \tag{3}$$

$$y = \tilde{a}_1 v^\star(q) + \tilde{a}_2 v_{m+1} + \ldots + \tilde{a}_{n-m+1} v_n \tag{4}$$

where

$$s^\star(q) = \log \sum_{i=1}^m \exp\left(\frac{q^T k_i}{\sqrt{d}}\right) \quad \text{and} \quad v^\star(q) = \sum_{i=1}^m \exp\left(\frac{q^T k_i}{\sqrt{d}} - s^\star(q)\right) v_i. \tag{5}$$

The term $s^\star(q)$ is the log of the unnormalized probability mass associated with the cluster, and $v^\star(q)$ is the attention-weighted mean of its values under query q. To show that Eqn. 2 and Eqn. 4 are equivalent simply substitute in the expressions for $s^\star(q)$ and $v^\star(q)$ (see Sec. A.2). The extension to multiple clusters and multi-head attention is straightforward. However, for clarity of exposition we focus on a single cluster (and remaining set of unmerged key-value pairs) here.

We call the cache entry representing a merged cluster a *slot*. Each slot holds one key vector, one value vector, and a constant number of statistics. A token that has never been merged occupies a slot of its own with m=1, otherwise a slot represents several tokens ($m \geq 2$). Compression is reported as either ratios of number of slots to tokens or slot-equivalent bytes to token-equivalent bytes.

The main difficulty after merging keys and values into a cluster is in approximating $s^\star(q)$ and $v^\star(q)$ from the data stored in the slot. Ideally, the result computed after merging is close to the result computed had the keys and values not been merged in the first place for any future query q. A typical criterion for clustering is that the constituent keys are aligned as measured by cosine similarity, i.e., $\hat{k}_i^T \hat{k}_j \geq t$ for any two keys in the cluster, where $\hat{\cdot}$ denotes a normalized vector and $t \in [-1, 1]$ is some predetermined merge threshold. As $t \to 1$ we get perfect alignment, i.e., $\hat{k}_i = \hat{k}_j$, giving rise to the following assumption.[1]

[1]When applied to multi-head attention, the criterion is for worst-head alignment, i.e., $\min_h \{\hat{k}_{h,i}^T \hat{k}_{h,j}\} \geq t$.

Table 1: Surrogate models $\tilde{p}$ of a cluster's key-norm distribution, and the closed-form corrections implied by Eqn. 7. The top four rows use only the first two moments of $\tilde{p}$ and differ only in how they place the mass. The final row leaves key-norm space and matches moments against constituent logits, dropping the key alignment assumption. See text for details; App. A.1 for symbol definitions.

SURROGATE $\tilde{p}$	SCORE $s(z)$	VALUE $v(z)$	EXACT WHEN
one atom at μ_r	$\log m + \mu_r z$	μ_v	all r_i equal
Gaussian $\mathcal{N}(\mu_r, \sigma_r^2)$	$\log m + \mu_r z + \frac{1}{2}\sigma_r^2 z^2$	$\mu_v + z\,\Sigma_{rv}$	only as $m \to \infty$
two atoms at $\mu_r \pm \sigma_r$	$\log m + \mu_r z + \log\cosh(\sigma_r z)$	$\mu_v + \frac{\tanh(\sigma_r z)}{\sigma_r}\Sigma_{rv}$	$m = 2$
two atoms, scaled	$\log m + \mu_r z + a\log\cosh(bz)$	$\mu_v + \frac{\tanh(bz)}{b}\Sigma_{rv}$	$m = 2$
two atoms, logit-space	$\log m + \|\bar{k}\| z + \log\cosh(\tilde{x})$	$\mu_v + d^{-\frac{1}{2}}\Sigma_{vk} q$	$m = 2$, in dist.

Assumption 3.1 (Key Alignment): Let k_i and k_j be two keys merged into the same cluster. Then $\hat{k}_i \approx \hat{k}_j$, which we idealize as exact. Writing r for a key's norm, so that $r_i = \|k_i\|$ and $r_j = \|k_j\|$, we take $k_i = r_i\tilde{k}$ and $k_j = r_j\tilde{k}$ where $\tilde{k}$ is the unit key vector representing the cluster.

With this assumption in hand, we can define $z = q^T\tilde{k}/\sqrt{d}$ for any arbitrary query q, and write $s^\star$ and $v^\star$ as functions of the scalar quantity z. Namely,

$$s^\star(z) = \log\sum_{i=1}^{m}\exp(r_i z) \quad \text{and} \quad v^\star(z) = \sum_{i=1}^{m}\exp(r_i z - s^\star(z))\, v_i. \tag{6}$$

Everything a merged slot must serve is contained in these two scalar-argument functions, and they depend on the cluster only through the set $\{(r_i, v_i)\}$ of constituent key norms and value vectors—the individual high-dimensional keys are no longer needed. Our strategy is therefore to retain a small number of low-order moments of this set and reconstruct $s^\star$ and $v^\star$ from them in closed form.

The two functions in Eqn. 6 are not arbitrary. Writing p for the empirical distribution of the pairs (r_i, v_i) within the cluster, the functions can be expressed as

$$s^\star(z) = \log m + \log\mathbf{E}[e^{rz}] \quad \text{and} \quad v^\star(z) = \mathbf{E}[v\,e^{rz}]\;/\,\mathbf{E}[e^{rz}] \tag{7}$$

where $\mathbf{E}[\cdot]$ is the (unweighted) average over the m constituent elements of a cluster. Hence up to the constant $\log m$ the score correction is the cumulant generating function (CGF) of the cluster's key-norm distribution, and the value correction is the mean value under distribution p tilted by e^{rz}. The object to be approximated is therefore not really a pair of functions but the distribution p itself. By choosing a surrogate distribution $\tilde{p}$, described by a handful of scalars, we can evaluate Eqn. 7 under $\tilde{p}$ in closed-form exactly. All approximation error is then attributable to a single modeling decision, and the sufficient statistics are dictated by the surrogate model. To distinguish between the true and approximate $s^\star$ and $v^\star$, we use s and v for the quantities derived using the surrogate.

Tab. 1 summarizes the surrogate distributions $\tilde{p}$ investigated in this paper. We restrict attention to surrogates determined by the first two moments of the key norms, together with the value statistics that they induce. For example, the first row places a single atom (i.e., point mass) at $(\mu_r, \mu_v) = (\mathbf{E}[r], \mathbf{E}[v])$, while the second row models the cluster as a Gaussian with mean μ_r and variance $\sigma_r^2 = \mathbf{E}\left[(r-\mu_r)^2\right]$. The term $\Sigma_{rv} = \mathbf{E}[(r-\mu_r)(v-\mu_v)] \in \mathbb{R}^d$ captures the covariance between key-norm r and value vector v. The one atom surrogate discards the norm spread entirely and yields the count correction of Tian et al. (2026), the strongest published correction of this kind and the baseline we improve on. The remaining rows define two-atom variants with the final row applying the same two-atom construction to the per-constituent logits ℓ_i rather than the norm r_i by dropping the aligned-keys assumption (see Sec. 3.2).

Before going into further details we illustrate important properties of each surrogate in Fig. 1. The plot is constructed from a synthetically generated cluster ($m = 8$) with key-norms r_i sampled in the range $[r_{\min}, r_{\max}]$. The left panel compares the second-order correction of each surrogate distribution against the exact score (solid black line) assuming aligned keys. Note that the one-atom surrogate fails to capture curvature whereas the Gaussian and two-atom surrogates manage to approximate the exact score to a greater or lesser extent. Observe also that the exact score is not symmetric in z, which the last two surrogates are able to model.

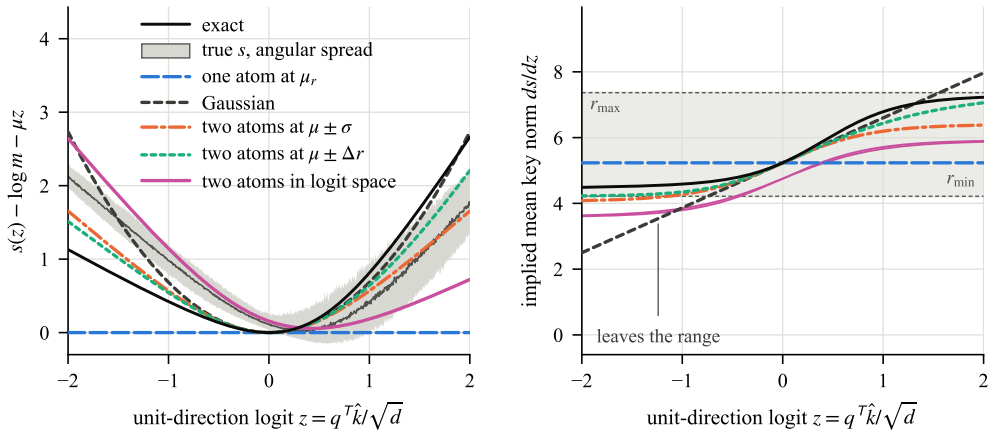

Figure 1: Which surrogate should stand in for a merged cluster? We illustrate properties on a single synthetic cluster whose construction is described in App. A.11. The left panel shows score correction net the one-atom term as a function of the unit-direction logit z. The gray band depicts the true correction once the constituents carry angular spread (ignored by *exact* which assumes aligned keys). The right panel shows the mean key norm ds/dz implied by the surrogate against the admissible range of norms appearing in the cluster $[r_{\min}, r_{\max}]$ (shaded).

In the same (left) panel, we also show an example of a true cluster, where keys do not perfectly align so that there exists some angular spread. The gray band depicts deviation due to query components orthogonal to the cluster's representative key direction. In this example, the response profile shifts to the right, which only the logit space surrogate (bottom row in Tab. 1) is able to capture.

The second panel in Fig. 1 shows the derivative of score function $s^\star$ with respect to z. This is the implied key-norm mean at z, i.e., $\mu_r(z) = \mathbf{E}\left[re^{rz}\right]/\mathbf{E}\left[e^{rz}\right]$. Notice that the two atom surrogates remain inside the observed range $[r_{\min}, r_{\max}]$ with the scaled variant asymptotically approaching its boundary. The logit space curve (which includes a small query-dependent band) legitimately exits below $r_{\min}$ since the surrogate corrects for non-aligned keys (i.e., violations of Assumption 3.1).

3.1 Score and Value Correction

Score Correction. The score function induced by each surrogate distribution is shown in Tab. 1. All surrogates recover standard attention in the singleton case ($m = 1$) since $\sigma_r = 0$. The one-atom surrogate is degenerate and only exact when all keys are equal, in which case every surrogate is exact. The two-atom surrogate, which places equal mass at each of $\mu_r \pm \sigma_r$, gives

$$s(z) = \log m + \log \tfrac{1}{2}\left(e^{(\mu_r+\sigma_r)z} + e^{(\mu_r-\sigma_r)z}\right) = \log m + \mu_r z + \log\cosh(\sigma_r z), \tag{8}$$

and is exact for any two aligned keys as the following proposition shows.[2]

Proposition 3.2 (Exactness for Aligned Keys): Let $\hat{k} \in \mathbb{R}^d$ be an arbitrary unit vector and let $k_1 = r_1\hat{k}$ and $k_2 = r_2\hat{k}$ for scalars $r_1, r_2 > 0$. Let $\mu_r = (r_1 + r_2)/2$, $\sigma_r = (r_1 - r_2)/2$, and $z = q^T\hat{k}/\sqrt{d}$ for some $q \in \mathbb{R}^d$. Then

$$\exp\Big(\log(2) + \mu_r z + \log\cosh(\sigma_r z)\Big) = \exp\left(\frac{q^T k_1}{\sqrt{d}}\right) + \exp\left(\frac{q^T k_2}{\sqrt{d}}\right).$$

The score function induced by the Gaussian surrogate is precisely the second-order Taylor expansion of $s^\star$ about $z = 0$ (App. A.4). Moreover, every surrogate in Tab. 1 reproduces the same first- and second-order statistics $(\mu_r, \sigma_r^2, \Sigma_{rv})$, and $\log\cosh(\sigma_r z) = \frac{1}{2}\sigma_r^2 z^2 - \frac{1}{12}\sigma_r^4 z^4 + O(\sigma_r^6 z^6)$ agrees with the Gaussian to second order. What distinguishes the surrogate distributions is entirely the behavior

[2]To avoid numerical overflow $\log\cosh(x)$ is implemented as $\mathrm{logaddexp}(x, -x) - \log(2)$.

away from the expansion point. In this sense the corrections we adopt are second-order in the moments they consume, not in the number of terms they retain in the Taylor expansion.

Value Correction. The value correction follows the same pattern as score correction through the choice of surrogate distribution. Under the symmetric two-atom surrogate the cluster consists of two sub-populations with key norms $\mu_r \pm \sigma_r$ and key-norm–value covariance Σ_{rv}, giving

$$v(z) = \mu_v + \alpha\,\Sigma_{rv} = (1 - \alpha\mu_r)\,\mu_v + \alpha\Phi, \qquad \alpha = \frac{\tanh(\sigma_r z)}{\sigma_r}, \qquad \Phi = \mathbf{E}\,[rv]\,. \tag{9}$$

Here $\tanh$ results from the derivative of the score correction, $\frac{d}{dz}\log\cosh(\sigma_r z)/\sigma_r$, as required by Eqn. 7. The second form of $v(z)$ involving the mixed moment Φ rather than the covariance Σ_{rv} allows us to maintain an additive statistic at merge time. Note that, unlike score correction, value correction beyond the one-atom surrogate requires maintaining a d-vector statistic per merge slot. This will become significant when we measure compression ratios in our experiments.

Three properties follow immediately. First, the correction vanishes when all constituent norms are equal, since then r is deterministic and $\Sigma_{rv} = 0$. Thus, a cluster of (aligned) equal-norm keys is exactly modeled by the plain merge value for every query. Second, the correction for the two-atom surrogate is bounded since $|\alpha| \le 1/\sigma_r$, unlike the Gaussian case where the term $z\Sigma_{rv}$ is unbounded for non-trivial covariance. Last, the correction is exact for any two aligned keys.

Proposition 3.3 (Exactness of the Value Correction for Aligned Keys): Let $k_1 = r_1\hat{k}$, $k_2 = r_2\hat{k}$, μ_r, σ_r and z be as in Prop. 3.2, and let $v_1, v_2 \in \mathbb{R}^d$ be arbitrary. Let $\mu_v = \frac{1}{2}(v_1 + v_2)$ and $\Phi = \frac{1}{2}(r_1 v_1 + r_2 v_2)$. Then, with $\alpha = \tanh(\sigma_r z)/\sigma_r$, we have

$$(1 - \alpha\mu_r)\,\mu_v + \alpha\Phi = \frac{\exp\left(q^T k_1/\sqrt{d}\right) v_1 + \exp\left(q^T k_2/\sqrt{d}\right) v_2}{\exp\left(q^T k_1/\sqrt{d}\right) + \exp\left(q^T k_2/\sqrt{d}\right)} = v^\star(z).$$

Together, Props. 3.2 and 3.3 give the merged cache a two-token exactness guarantee that holds for every query, not just on average. This is in contrast to the result proposed by Tian et al. (2026), which is only exact at the point of merge (i.e., current q).[3] Beyond two-token merges, we present error bounds in App. A.6 for larger clusters and violations of the aligned key assumption.

3.2 Extensions

Scaling. The symmetric two-atom surrogate discussed thus far places its atoms at $\mu_r \pm \sigma_r$, so the score's slope saturates there, whereas Fig. 1 suggests that the exact slope saturates at $r_{\min}$ and $r_{\max}$. For a cluster whose norms are genuinely spread as $\mu_r \pm \sigma_r$ these coincide, but real clusters are typically skewed with many near-duplicate tokens and a handful of dominant outliers much farther from the mean than σ_r alone suggests. Thus, the surrogate saturates prematurely. We can recover the exact saturation level at no cost in the second-order data by introducing scaling factors a and b,

$$s(z) = \log m + \mu_r z + a\log\cosh(bz) \tag{10}$$

where matching the second derivative at the origin now requires only $ab^2 = \sigma_r^2$; Eqn. 8 is the particular member $a = 1$, $b = \sigma_r$. Choosing $ab = \Delta r$ where Δr is the observed extremal deviation, $\Delta r = r_{\max} - \mu_r$ if $z \ge 0$ and $\mu_r - r_{\min}$ otherwise, pins the asymptotic slope to saturate exactly the limits (see Fig. 1). Solving for a and b gives

$$s(z) = \log m + \mu_r z + \frac{\Delta r^2}{\sigma_r^2}\log\cosh\left(\frac{\sigma_r^2 z}{\Delta r}\right) \tag{11}$$

which requires tracking $r_{\max}$ and $r_{\min}$ as two additional scalar statistics. The corresponding value correction is Eqn. 9 with σ_r replaced by the same b,

$$v(z) = (1 - \beta\mu_r)\,\mu_v + \beta\Phi, \qquad \beta = \frac{\tanh(bz)}{b}. \tag{12}$$

[3] KeepKV cleverly tracks logits via a per-slot exponential moving average in an attempt to correct the averaging post merge. We found this to be of limited utility in our experiments, and it involves additional per-token computation.

Repositioning the atoms retains exactness for two aligned keys, i.e., Props. 3.2 and 3.3 still hold.

Logit-space. Relaxing the aligned key assumption, we now consider the final surrogate in Tab. 1. The CGF identity in Eqn. 7 can be generalized beyond the setting in which it was presented. Indeed, for any key configuration, $s^\star(q) = \log m + \log \mathbf{E}\left[e^{\ell}\right]$ where $\ell_i = q^T k_i/\sqrt{d}$ is the per-constituent logit. So the score is always $\log m$ plus the log-moment-generating function of some distribution. Until now we have considered the norm distribution for aligned keys through the single scalar z, and hence a special case, $\ell_i = r_i z$. In the more general setting we consider moments over ℓ,

$$\mathbf{E}\left[\ell\right] = \frac{1}{m}\sum_{i=1}^{m} q^T k_i \,/\sqrt{d} = q^T \bar{k}\,/\sqrt{d} \quad \text{and} \quad \mathrm{Var}(\ell) = q^T \Sigma_{kk} q \,/\, d, \tag{13}$$

where $\bar{k} = \frac{1}{m}\sum_{i=1}^{m} k_i$ is the mean key vector and $\Sigma_{kk} = \mathbf{E}\left[(k-\bar{k})(k-\bar{k})^T\right]$. Note that the latter has shape $d \times d$ and hence intractable to store. Instead, we split q into its component along $\hat{k}$ and the orthogonal remainder $q_\perp$, and use the isotropic approximation

$$\mathrm{Var}(\ell) \;\approx\; \sigma_r^2 z^2 \;+\; \frac{\mathrm{tr}\,\Sigma_{kk}}{d-1}\cdot\frac{\|q_\perp\|^2}{d} \quad \text{with} \quad \mathrm{tr}\,\Sigma_{kk} = \tfrac{1}{m}\textstyle\sum_i \|k_i\|^2 - \|\bar{k}\|^2, \tag{14}$$

which we denote by $\tilde{x}^2$ in Tab. 1. See App. A.8 for the full derivation.

The value correction upgrades along the same lines to give,

$$v(q) = \mu_v + \Sigma_{vk} q\,/\sqrt{d}. \tag{15}$$

where $\Sigma_{vk} = \mathbf{E}\left[(v-\mu_v)(k-\bar{k})^T\right]$. However, this is once again a $d \times d$ quantity and therefore too expensive to store. As such we resort to simply using μ_v in our experiments.

3.3 Order-Invariant Online Merging

For merging to be usable during inference, clusters must be able to grow one token at a time, and the result must not depend on the order in which tokens happen to arrive. Both properties are apparent when writing the correction statistics as being derived from unnormalised power sums.[4] For a cluster with constituent index set $\mathcal{I}$, define three scalars and two d-vectors

$$S_0 = \sum_{i\in\mathcal{I}} 1, \quad S_r = \sum_{i\in\mathcal{I}} r_i, \quad S_{rr} = \sum_{i\in\mathcal{I}} r_i^2, \quad S_v = \sum_{i\in\mathcal{I}} v_i, \quad S_{rv} = \sum_{i\in\mathcal{I}} r_i v_i, \tag{16}$$

from which every quantity the corrections consume follows by normalisation,

$$m = S_0, \quad \mu_r = \frac{S_r}{S_0}, \quad \sigma_r^2 = \frac{S_{rr}}{S_0} - \mu_r^2, \quad \mu_v = \frac{S_v}{S_0}, \quad \Phi = \frac{S_{rv}}{S_0}. \tag{17}$$

Expressed this way, merging is simply addition. Pseudo-code for our merge algorithm and a proof of merge order invariance are included in App. A.9.

4 Experimental Results

We evaluate on three tasks to validate our theory: **LongHealth** (Adams et al., 2024), multiple-choice question answering over long clinical documents; **MATH500**, a 500-problem subset of MATH (Hendrycks et al., 2021), short-context mathematical reasoning; and **RULER** (Hsieh et al., 2024) needle-in-a-haystack retrieval, at context length 16384. The first aggregates evidence over many tokens, the second is decode heavy, and the last resolves on a single token. Models are **Qwen2.5-7B-Instruct** and **Llama-3.1-8B-Instruct**, both with $d = 128$, running on a single NVIDIA H100 GPU with 96GB of VRAM. Merges are done online by the criterion of Sec. 3.3 with a key-cosine threshold swept per task, so every point in the figures below is one threshold.

We report cache size in *token-equivalent* bytes so that an unmerged slot measures one unit. The cost of a merged slot is shown in Tab. 2 for each variant taking into account stored scalar and vector statistics. The first row of the table corresponds to merging without correction (Wang et al., 2024) while the second row compensates for the cluster size but not variance and is the strongest published baseline (Tian et al., 2026).

[4] In our implementation we store normalized quantities for better numerical robustness, and in practice perform chunked updates over a group of tokens rather than a single token at a time.

Table 2: The variants evaluated in this paper and their relative storage per merged slot. †Despite the cluster size m not appearing in the correction formulae in the first row it is still needed for computing running means, μ_r and μ_v. The mean norm μ_r is not counted for variants 1–5 since the stored key is reconstructed at magnitude μ_r (App. A.9). See Tab. 1 and App. A.1 for full definition of terms.

ID	Score correction $s(z)$	Value correction $v(z)$	Slot cost
1	$\mu_r z$	μ_v	$(2d+1)/2md$ †
2	$\log(m) + \mu_r z$	μ_v	$(2d+1)/2md$
3	$\log(m) + \mu_r z + \frac{1}{2}\sigma_r^2 z^2$	$(1-\alpha\mu_r)\mu_v + \alpha\Phi$	$(3d+2)/2md$
4	$\log(m) + \mu_r z + \log\cosh(\sigma_r z)$	$(1-\alpha\mu_r)\mu_v + \alpha\Phi$	$(3d+2)/2md$
5	$\log(m) + \mu_r z + a\log\cosh(bz)$	$(1-\beta\mu_r)\mu_v + \beta\Phi$	$(3d+4)/2md$
6	$\log(m) + \|\tilde{k}\|z + \log\cosh(\tilde{x})$	μ_v	$(2d+3)/2md$

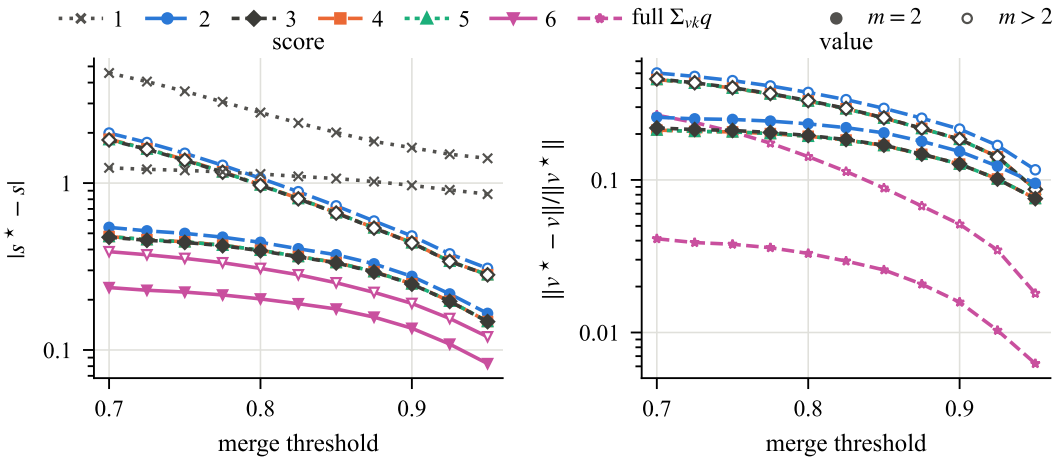

Figure 2: Plots of approximation error against merge threshold for Llama-3.1-8B on RULER averaged over three prompts (lower is better). Left: absolute score error $|s^\star - s|$, measured in the log domain, against the true merged attention. Right: relative value error, $\|v^\star - v\|/\|v^\star\|$. Curves are labelled by the variant IDs of Tab. 2 and separated into cluster sizes $m = 2$ and $m > 2$.

4.1 Approximation Fidelity

We evaluate the quality of our corrections on real activations against $s^\star$ and $v^\star$. To collect data we perform one forward pass per dataset example saving raw keys, values and queries. We then merge on a clone of the KV cache at each threshold and report the approximated score and value against the exact log-sum-exp and softmax-weighted value over the cluster's raw constituents shown in Fig. 2.

Three observations validate the theory. First, the ordering predicted by our ladder holds everywhere. Specifically, at every threshold the two-atom variants improve over the one-atom variant, and the logit-space variant improves over all others. Second, the two-atom variants (IDs 4 and 5) coincide exactly at $m = 2$ as required by Prop. A.10. Beyond $m = 2$ they separate only slightly. Third, the curves highlight violation of the key alignment assumption. From Propositions 3.2 and 3.3 any residual at $m = 2$ for these two-atom variants is attributable to angular spread alone. Indeed, we can see the residual falling monotonically as alignment tightens (higher cosine similarity threshold). The logit-space variant (ID 6) recovers most of that residual on the score side. The value-side curve shows the corresponding correction for the value using the full cross-covariance Σ_{vk}. However, as discussed above, this quantity is too large to store, and only included here to validate the theory.

4.2 Accuracy against Cache Size

Fig. 3 reports task accuracy versus KV-cache compression by sweeping over the merge threshold for selected variants from Tab. 2. We include the eviction baseline TOVA (Oren et al., 2024), a state-of-the-art query-aware eviction scheme, swept over its own compression parameter for calibration. Reading the curves vertically at a fixed relative cache size gives a like-for-like comparison.

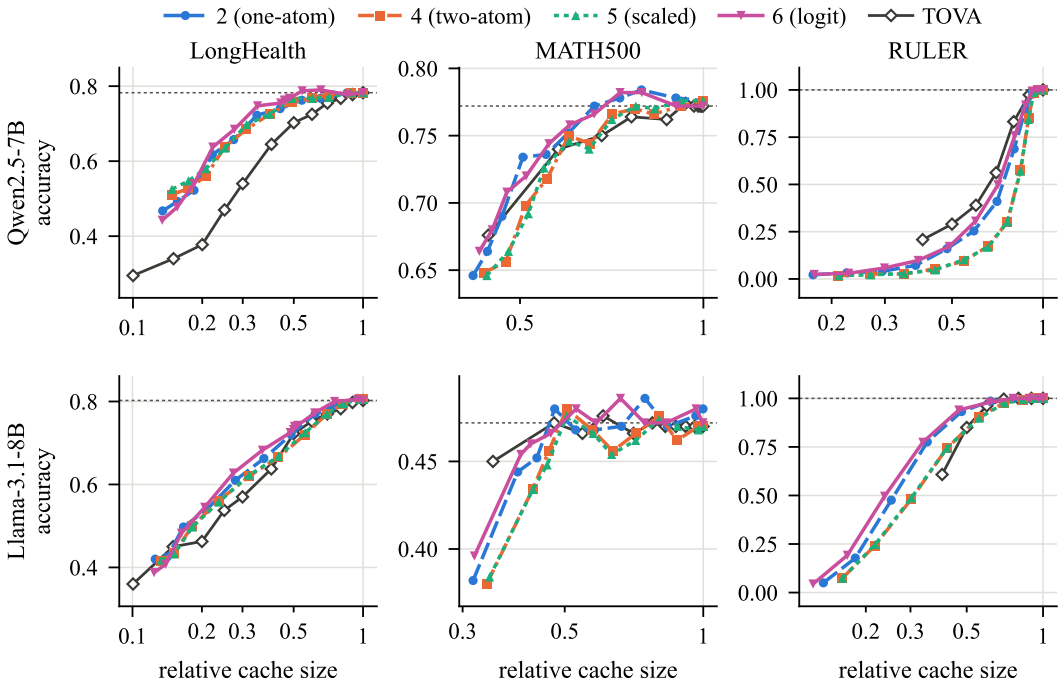

Figure 3: Accuracy on three different benchmark tasks and two models as a function of cache compression measured in token relative bytes. Curves are the merging variants of Tab. 2, labelled by ID; each marker is one merge threshold evaluated over the full benchmark (400 questions for LongHealth, 500 problems for MATH500, 500 samples for RULER). The dotted line marks uncompressed accuracy. We include a SoTA eviction scheme, TOVA, as a reference accuracy calibration.

The evidence-aggregating and decode-heavy tasks (LongHealth and MATH500) tolerate merging far more readily than retrieval (RULER), which resolves on a single token. This is most evident on Qwen2.5-7B, where merging on RULER degrades sharply when compressed. We highlight two further observations: First, on the whole, merging is competitive with state-of-the-art eviction (TOVA) across the range of tasks and models. Second, the token equivalent axis compensates for the storage cost of each variant. As such, the two-atom variants, despite producing better score and value approximations (see Sec. 4.1) are less competitive when measured against cache size.

5 Discussion and Conclusion

In this paper we have unified merging approaches to KV cache management, putting them under the perspective of choosing a surrogate probability distribution for modeling the cluster of merged keys and values. This gives rise to closed form correction formulae that can be applied at inference time rather than ad hoc rules or specialized trained models. Importantly, the corrections are exact for two-token merges of aligned keys, under every query. Beyond clusters of two tokens, error bounds have been derived, and empirical results are reported that support the theory.

5.1 Limitations

Angular spread. Real clusters formed by a cosine similarity criterion have non-aligned keys so violate the assumption upon which the one- and two-atom corrections are based. Our logit-space variant (Sec. 3.2) corrects for angular spread isotropically, which reduces much of the score residual; the corresponding value correction remains infeasible to store.

Kernel modification. The corrections are analytic and need no retraining, but they must be applied inside the attention computation requiring modification to inference code. The calculations are compatible with the online-softmax structure of FlashAttention, so fast, but the method cannot wrap an unmodified inference stack and is unavailable for closed-weight models served behind an API.

AI USE STATEMENT

In this work, we used generative AI tools for helping with experiments, including coding of methods and collating results. Generative AI tools were also used for rephrasing select parts of the paper and for assisting with mathematical proofs. We have not used generative AI tools for ideation or production of the main mathematical insights. All AI produced work was reviewed. Specifically, AI generated code was checked for correctness against the formulae derived in the paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper provides theoretical insights into KV cache merging. All experiments are conducted on publicly available datasets and used in a manner consistent with the licensing terms. Likewise for all software. To avoid any doubt, no human subjects or sensitive/copyright data was used in this research.

REPRODUCIBILITY STATEMENT

Sufficient details have been provided in the main body of the paper and appendices (including full proofs for all propositions) to allow reproduction of the results by researchers familiar with the field. Moreover, software source code will be released upon publication of the paper.

REFERENCES

Lisa Adams, Felix Busch, Tianyu Han, Jean-Baptiste Excoffier, Matthieu Ortala, Alexander Löser, Hugo JWL Aerts, Jakob Nikolas Kather, Daniel Truhn, and Keno Bressem. LongHealth: A question answering benchmark with long clinical documents. *arXiv preprint arXiv:2401.14490*, 2024.

Dmitry Akulov, Mohamed Sana, Antonio De Domenico, Tareq Si Salem, Nicola Piovesan, and Fadhel Ayed. KVCompose: Efficient structured KV cache compression with composite tokens. *arXiv preprint arXiv:2509.05165*, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2023.

Soumia Bouyahiaoui, Manel Kara laouar, Aicha Boutorh, and Mohamed Hadj Ameur. SelKV: Selective KV cache merging with per-token merge-or-drop and attention compensation. *arXiv preprint arXiv:2607.16213*, 2026.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yuan Feng, Junlin Lv, Haoyu Guo, Yukun Cao, S. Kevin Zhou, and Xike Xie. CriticalKV: Optimizing KV cache eviction from an output perturbation perspective. *arXiv preprint arXiv:2502.03805*, 2026.

Raghavv Goel, Junyoung Park, Mukul Gagrani, Dalton Jones, Matthew Morse, Harper Langston, Mingu Lee, and Chris Lott. CAOTE: KV cache selection for LLMs via attention output error-based token eviction. *arXiv preprint arXiv:2504.14051*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun Sophia Shao, Kurt Keutzer, and Amir Gholami. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. *arXiv preprint arXiv:2401.18079*, 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling (COLM)*, 2024.

Jiantong Jiang, Peiyu Yang, Rui Zhang, and Feng Liu. Towards efficient large language model serving: A survey on system-aware KV cache optimization. In *Findings of the Association for Computational Linguistics (ACL Findings)*, 2026.

Jang-Hyun Kim, Jinuk Kim, Sangwoo Kwon, Jae W. Lee, Sangdoo Yun, and Hyun Oh Song. KVzip: Query-agnostic KV cache compression with context reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Lianjun Liu, Hongli An, Weiqi Yan, Xin Du, Shengchuan Zhang, Huazhong Liu, and Yunshan Zhong. KVSlimmer: Theoretical insights and practical optimizations for asymmetric KV merging. *arXiv preprint arXiv:2603.00907*, 2026.

Piotr Nawrot, Adrian Łańcucki, Marcin Chochowski, David Tarjan, and Edoardo M. Ponti. Dynamic memory compression: Retrofitting LLMs for accelerated inference. In *Proc. of the International Conference on Machine Learning (ICML)*, 2024.

Charles O'Neill, Alex Sandomirsky, Harry Partridge, Mudith Jayasekara, and Max Kirkby. Still: Amortized KV cache compaction in a single forward pass. *arXiv preprint arXiv:2606.07878*, 2026.

Matanel Oren, Michael Hassid, Nir Yarden, Yossi Adi, and Roy Schwartz. Transformers are multi-state RNNs. In *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Junjie Peng, You Wu, Haoyi Wu, Jialong Han, Xiaohua Xie, Kewei Tu, and Jianhuang Lai. GRKV: Global regression for training-free KV cache compression in long-context LLMs. *arXiv preprint arXiv:2605.31105*, 2026.

Yuxuan Tian, Zihan Wang, Yebo Peng, Aomufei Yuan, Zhiming Wang, Bairen Yi, Xin Liu, Yong Cui, and Tong Yang. KeepKV: Achieving periodic lossless KV cache compression for efficient LLM inference. In *Proc. of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 33259–33267, 2026. doi: 10.1609/aaai.v40i39.40611.

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

Zhongwei Wan, Xinjian Wu, Yu Zhang, Yi Xin, Chaofan Tao, Zhihong Zhu, Xin Wang, Siqi Luo, Jing Xiong, Longyue Wang, and Mi Zhang. D2O: Dynamic discriminative operations for efficient long-context inference of large language models. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2025.

Zheng Wang, Boxiao Jin, Zhongzhi Yu, and Minjia Zhang. Model tells you where to merge: Adaptive KV cache merging for LLMs on long-context tasks. *arXiv 2407.08454*, 2024.

Shunlong Wu, Hai Lin, Shaoshen Chen, Tingwei Lu, Yongqin Zeng, Shaoxiong Zhan, Hai-Tao Zheng, and Hong-Gee Kim. SemantiCache: Efficient KV cache compression via semantic chunking and clustered merging. *arXiv preprint arXiv:2603.14303*, 2026.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2024.

Jian Yuan, Ziwei He, Haoli Bai, Jingwen Leng, and Bo Jiang. WeightedKV: Attention scores weighted key-value cache merging for large language models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

Amir Zandieh, Insu Han, Vahab Mirrokni, and Amin Karbasi. SubGen: Token generation in sublinear time and memory. *arXiv preprint arXiv:2402.06082*, 2024.

Amir Zandieh, Majid Daliri, Majid Hadian, and Vahab Mirrokni. TurboQuant: Online vector quantization with near-optimal distortion rate. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2026.

Yuhang Zhan, Lisi Chen, and Shuo Shang. ResKV: Reconstructing omitted attention contributions for fixed-budget KV cache compression. *arXiv preprint arXiv:2607.29591*, 2026.

Yuxin Zhang, Yuxuan Du, Gen Luo, Yunshan Zhong, Zhenyu Zhang, Shiwei Liu, and Rongrong Ji. CaM: Cache merging for memory-efficient LLMs inference. In *Proc. of the International Conference on Machine Learning (ICML)*, pp. 58840–58850, 2024.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

A Derivations and Proofs

A.1 Symbol Definitions

Tab. 3 lists the cluster statistics used throughout the paper. All expectations are the unweighted average over the m constituents of a cluster; $\hat{k}$ is the stored unit-key direction and $\bar{k} = \mathbf{E}[k]$ the plain mean key. We store raw moment Φ as it is additive. The identity $\Sigma_{rv} = \Phi - \mu_r \mu_v$ is an algebraic fact, not an implementation rule, which would invite numerical overflow. Our implementation never materializes Σ_{rv}. The value correction when needed is evaluated in the equivalent form $(1 - \alpha\mu_r)\mu_v + \alpha\Phi$ as shown in Tab. 2. Any offline analysis computes Σ_{rv} using the centered two-pass form $\mathbf{E}[(r - \mu_r)(v - \mu_v)]$.

Table 3: Cluster statistics. The centered quantities are derived at evaluation time or used only in analysis. The final column shows quantities stored in a merge slot.

Symbol	Definition	Shape	Stored
μ_r	$\mathbf{E}[r]$, mean key norm	scalar	yes
σ_r^2	$\mathbf{E}\left[(r - \mu_r)^2\right]$, key-norm variance	scalar	yes
μ_v	$\mathbf{E}[v]$, mean value	d-vector	yes
Φ	$\mathbf{E}[rv]$, raw norm–value correlation	d-vector	when Σ_{rv} is used
Σ_{rv}	$\Phi - \mu_r\mu_v$, norm–value covariance	d-vector	no (derived)
Σ_{pv}	$\mathbf{E}[(v - \mu_v)(p - \bar{p})],\ p = \hat{k}^T k$	d-vector	no
Σ_{kk}	$\mathbf{E}\left[(k - \bar{k})(k - \bar{k})^T\right]$, key covariance	$d \times d$	no ($\operatorname{tr}\Sigma_{kk}$ derivable)
Σ_{vk}	$\mathbf{E}\left[(v - \mu_v)(k - \bar{k})^T\right]$, value–key cross-covariance	$d \times d$	no

To simplify expressions throughout the paper we often write $\alpha = \tanh(\sigma_r z) \,/\, \sigma_r$, $\beta = \tanh(bz) \,/\, b$, and $\tilde{x}^2 = \sigma_r^2 z^2 \ + \ \frac{\operatorname{tr}\Sigma_{kk}}{d-1} \cdot \frac{\|q_\perp\|^2}{d}$ where $\|q_\perp\|^2 = \|q\|^2 - (q^T\hat{k})^2$ by Pythagoras' theorem.

A.2 Equivalence of Equations 2 and 4

For brevity, write $\ell_i = q^T k_i / \sqrt{d}$. Substituting the attention weights $\tilde{a}$ into Eqn. 4 we have

$$
\begin{aligned}
y &= \tilde{a}_1 v^\star(q) + \tilde{a}_2 v_{m+1} + \ldots + \tilde{a}_{n-m+1} v_n && (18)\\
&= \frac{1}{Z}\exp(s^\star(q)) v^\star(q) + \frac{1}{Z}\exp(\ell_{m+1}) v_{m+1} + \ldots + \frac{1}{Z}\exp(\ell_n) v_n && (19)\\
&= \frac{1}{Z}\exp(s^\star(q)) \underbrace{\left(\sum_{i=1}^{m} \exp\left(\ell_i - s^\star(q)\right) v_i\right)}_{v^\star(q)} + \frac{1}{Z}\exp(\ell_{m+1}) v_{m+1} + \ldots + \frac{1}{Z}\exp(\ell_n) v_n && (20)\\
&= \frac{1}{Z}\left(\sum_{i=1}^{m} \exp(\ell_i) v_i\right) + \frac{1}{Z}\exp(\ell_{m+1}) v_{m+1} + \ldots + \frac{1}{Z}\exp(\ell_n) v_n && (21)\\
&= \frac{1}{Z}\sum_{i=1}^{n} \exp(\ell_i) v_i && (22)\\
&= a_1 v_1 + \ldots + a_n v_n && (23)
\end{aligned}
$$

where $Z = \exp(s^\star(q)) + \sum_{i=m+1}^{n} \exp(\ell_i) = \sum_{i=1}^{n} \exp(\ell_i)$.

A.3 Deferred Proofs

The proof of Prop. 3.2 is a direct computation.

Proof.

$$\text{LHS} = 2e^{\mu_r z}\cosh(\sigma_r z) = e^{\mu_r z}\left(e^{\sigma_r z} + e^{-\sigma_r z}\right) = e^{(\mu_r+\sigma_r)z} + e^{(\mu_r-\sigma_r)z} = e^{r_1 z} + e^{r_2 z} = \text{RHS}$$

□

The proof of Prop. 3.3 is likewise a direct computation.

Proof. For the left-hand side, $\Phi - \mu_r\mu_v = \frac{1}{2}\left((r_1-\mu_r)v_1 + (r_2-\mu_r)v_2\right) = \frac{\sigma_r}{2}(v_1 - v_2)$, so

$$(1-\alpha\mu_r)\,\mu_v + \alpha\Phi = \mu_v + \alpha\,(\Phi - \mu_r\mu_v) = \mu_v + \tanh(\sigma_r z)\,\tfrac{1}{2}(v_1 - v_2).$$

For the right-hand side, using $q^T k_i/\sqrt{d} = r_i z$ and dividing numerator and denominator by $e^{\mu_r z}$, we have

$$\begin{aligned} v^\star(z) = \frac{e^{\sigma_r z}v_1 + e^{-\sigma_r z}v_2}{2\cosh(\sigma_r z)} &= \frac{\cosh(\sigma_r z)(v_1+v_2) + \sinh(\sigma_r z)(v_1 - v_2)}{2\cosh(\sigma_r z)} \\ &= \mu_v + \tanh(\sigma_r z)\,\tfrac{1}{2}(v_1 - v_2), \end{aligned}$$

which coincides with the left-hand side. □

A.4 Taylor Expansion of the Score Correction $s^\star(z)$

Under the Key Alignment Assumption, $q^T k_i/\sqrt{d} = r_i z$ for $z = q^T\tilde{k}/\sqrt{d}$, so the exact score and value corrections become functions of the scalar z alone,

$$s^\star(z) = \log\sum_{i=1}^m \exp(r_i z), \qquad v^\star(z) = \sum_{i=1}^m w_i(z)\,v_i, \qquad w_i(z) = \exp\left(r_i z - s^\star(z)\right). \tag{24}$$

Note $w_i(z) \geq 0$ and $\sum_i w_i(z) = 1$, so for every z, $w(z)$ is itself a valid probability distribution over the m cluster elements. We can derive a Taylor expansion of $s^\star$ around an arbitrary reference point z_0. This supplies the moments for every surrogate of Tab. 1, and is the form of the correction obtained from the Gaussian surrogate specifically.

Define the tilted mean and variance of the cluster's key norms under $w(z)$,

$$\mu_r(z) = \sum_{i=1}^m w_i(z)\,r_i, \qquad \sigma_r^2(z) = \sum_{i=1}^m w_i(z)\big(r_i - \mu_r(z)\big)^2. \tag{25}$$

The derivatives of $s^\star$ are exactly these moments. Writing $Z(z) = \exp(s^\star(z)) = \sum_i \exp(r_i z)$, so that $w_i(z) = \exp(r_i z)/Z(z)$, we obtain

$$\frac{ds^\star}{dz} = \frac{Z'(z)}{Z(z)} = \sum_{i=1}^m r_i\,w_i(z) = \mu_r(z), \qquad \frac{dw_i}{dz} = \frac{r_i e^{r_i z}}{Z(z)} - \frac{e^{r_i z}Z'(z)}{Z(z)^2} = w_i(z)\big(r_i - \mu_r(z)\big). \tag{26}$$

Taking second derivatives,

$$\frac{d^2 s^\star}{dz^2} = \frac{d\mu_r}{dz} = \sum_{i=1}^m r_i\,w_i(z)\big(r_i - \mu_r(z)\big) = \sum_{i=1}^m w_i(z)\big(r_i - \mu_r(z)\big)^2 = \sigma_r^2(z), \tag{27}$$

the last step because $\sum_{i=1}^m w_i(z)(r_i - \mu_r(z)) = 0$. By Taylor's theorem, for any reference point z_0,

$$s^\star(z) = s^\star(z_0) + \mu_r(z_0)(z - z_0) + \tfrac{1}{2}\sigma_r^2(z_0)(z - z_0)^2 + R_2(z), \tag{28}$$

where the remainder $R_2(z) = \frac{1}{6}s^{\star\prime\prime\prime}(\xi)(z - z_0)^3$ for some ξ between z_0 and z and $s^{\star\prime\prime\prime}(\xi)$ is the third central moment of the tilted distribution $w(\xi)$. Truncating Eqn. 28 gives the second-order score correction

$$s^\star(z) \approx s^\star(z_0) + \mu_r(z_0)(z - z_0) + \tfrac{1}{2}\sigma_r^2(z_0)(z - z_0)^2, \tag{29}$$

which is the Gaussian-surrogate correction of Tab. 1.

Remark A.1: At $z_0 = 0$, $w_i(0) = 1/m$ for all i, so $\mu_r(0)$ and $\sigma_r^2(0)$ reduce to the plain (untilted) mean and variance of the cluster's key norms, matching the constants μ_r, σ_r of Prop. 3.2.

Remark A.2: The two-atom surrogate of Sec. 3 can be read off this series. When the cluster's norms are $r_i \in \{\mu_r \pm \sigma_r\}$, every odd central moment beyond the first vanishes and every even central moment is determined entirely by σ_r^2, so the series sums *exactly* to $\mu_r z + \log\cosh(\sigma_r z)$. The surrogate view of Sec. 3 reaches the same closed form without truncating anything, which is why it also has the exactness of Prop. 3.2 directly.

A.5 Taylor Expansion of the Value Correction $v^\star(z)$

Reusing the notation of App. A.4, we can write,

$$\Sigma_{rv}(z) = \sum_{i=1}^{m} w_i(z)\big(r_i - \mu_r(z)\big)\, v_i, \qquad \Psi(z) = \sum_{i=1}^{m} w_i(z)\big(r_i - \mu_r(z)\big)^2\big(v_i - v^\star(z)\big). \tag{30}$$

These are the first two derivatives of $v^\star$. Using $dw_i/dz = w_i(z)(r_i - \mu_r(z))$ from App. A.4,

$$\frac{dv^\star}{dz} = \sum_{i=1}^{m} \frac{dw_i}{dz}\, v_i = \sum_{i=1}^{m} w_i(z)\big(r_i - \mu_r(z)\big)v_i = \Sigma_{rv}(z), \tag{31}$$

and differentiating $\Sigma_{rv}(z)$ term by term, we arrive at

$$\frac{d^2v^\star}{dz^2} = \frac{d\Sigma_{rv}}{dz} = \sum_{i=1}^{m} w_i(z)\Big(\big(r_i - \mu_r(z)\big)^2 - \sigma_r^2(z)\Big)\, v_i \tag{32}$$

$$= \sum_{i=1}^{m} w_i(z)\big(r_i - \mu_r(z)\big)^2 v_i - \sigma_r^2(z)\, v^\star(z) = \Psi(z), \tag{33}$$

the last step because $\sum_{i=1}^{m} w_i(z)(r_i - \mu_r(z))^2\, v^\star(z) = \sigma_r^2(z)\, v^\star(z)$. So by Taylor's theorem, for any reference point z_0,

$$v^\star(z) = v^\star(z_0) + \Sigma_{rv}(z_0)(z - z_0) + \tfrac{1}{2}\Psi(z_0)(z - z_0)^2 + R_2(z), \tag{34}$$

with remainder $R_2(z) = \frac{1}{6}v^{\star\prime\prime\prime}(\xi)(z - z_0)^3$ for some ξ between z_0 and z. Truncating gives the second-order value correction

$$v^\star(z) \approx v^\star(z_0) + \Sigma_{rv}(z_0)(z - z_0) + \tfrac{1}{2}\Psi(z_0)(z - z_0)^2. \tag{35}$$

Remark A.3: At $z_0 = 0$, $\Sigma_{rv}(0) = \Sigma_{rv} = \Phi - \mu_r\mu_v$, the plain (untilted) covariance between key norm and value across the cluster, which is the quantity the value correction of Sec. 3.1 restores, and the Gaussian-surrogate value correction of Tab. 1 is exactly this first-order term.

Remark A.4 (Centering, and the relationship to KeepKV): Nothing requires the expansion point to be the origin. Matching a surrogate to the *tilted* moments $\mu_r(z_0)$, $\sigma_r^2(z_0)$, $\mu_v(z_0)$, $\Sigma_{rv}(z_0)$ centers its accurate region at z_0, and these are computable from the same additive sums with each constituent weighted by $e^{r_i z_0}$. At the expansion point itself the centered corrections collapse onto the exact values, $s = s^\star(z_0)$ and $v = v^\star(z_0)$: a slot centered at z_0 and queried exactly there is served *losslessly*. This is precisely the construction of Tian et al. (2026), with their per-entry moving average of observed attention playing the role of z_0. Their correction is the zeroth-order case of ours, exact at the predicted operating point and degrading as the query drifts away from it, while the terms we retain in $z - z_0$ are an analytic model of that drift.

A.6 Error Bounds

We now bound the approximation quality of merging. Write $\varepsilon_s = s(z) - s^\star(z)$ for the score error and $\varepsilon_v = v(z) - v^\star(z)$ for the value error of a single cluster, and let y and $\tilde{y}$ be the attention outputs of Eqn. 2 computed over the unmerged and merged caches. The first result says that these two errors are the only ones there are, and identifies exactly what multiplies each.

Proposition A.5 (Error Decomposition): Let $\mathcal{I}$ be a subset of indices for a merged cluster that is not the whole cache, and let

$$u = \frac{\sum_{j \notin \mathcal{I}} e^{\ell_j} v_j}{\sum_{j \notin \mathcal{I}} e^{\ell_j}}$$

be the attention-weighted mean value of the rest of the cache, where $\ell_j = q^T k_j / \sqrt{d}$. Let Δ and $\tilde{\Delta}$ denote the fraction of attention mass the cluster receives before and after merging (they are equal when $\varepsilon_s = 0$). Then

$$\tilde{y} - y = \tilde{\Delta}\, \varepsilon_v + \Delta \left(1 - \tilde{\Delta}\right) (e^{\varepsilon_s} - 1)\, (v^\star(z) - u) \tag{36}$$

and consequently

$$\|\tilde{y} - y\| \leq \tilde{\Delta}\, \|\varepsilon_v\| + \Delta \left(1 - \tilde{\Delta}\right) |e^{\varepsilon_s} - 1|\, \|v^\star(z) - u\|\,. \tag{37}$$

Proof. Merging alters neither the logits ℓ_j of the unmerged entries nor their relative weights, so u is the same before and after and the output is a two-way mixture, $y = \Delta v^\star(z) + (1 - \Delta)u$ and $\tilde{y} = \tilde{\Delta} v(z) + (1 - \tilde{\Delta})u$. Subtracting and substituting $v(z) = v^\star(z) + \varepsilon_v$,

$$\tilde{y} - y = \tilde{\Delta}\varepsilon_v + (\tilde{\Delta} - \Delta)\, (v^\star(z) - u)\,.$$

Writing $A = e^{s^\star(z)}$, $B = e^{s(z)} = Ae^{\varepsilon_s}$ and $Z = \sum_{j \notin \mathcal{I}} e^{\ell_j}$, so that $\Delta = A/(A + Z)$ and $\tilde{\Delta} = B/(B + Z)$,

$$\tilde{\Delta} - \Delta = \frac{Z\,(B - A)}{(B + Z)\,(A + Z)} = \frac{A}{A + Z} \cdot \frac{Z}{B + Z} \cdot \frac{B - A}{A} = \Delta \left(1 - \tilde{\Delta}\right) (e^{\varepsilon_s} - 1)\,,$$

which gives Eqn. 36. Eqn. 37 then follows by the triangle inequality. □

Two consequences are worth noting before bounding ε_s and ε_v themselves. First, both terms carry a factor of the cluster's own attention mass, so a cluster that the query does not attend to contributes nothing however wrong its statistics are. This makes the negative-logit regime benign since a cluster whose keys are anti-aligned with the query has $\Delta \to 0$. Second, the score error (i.e., second term governed by e^{ε_s}) is multiplied by $\|v^\star(z) - u\|$. Merging leaves the relative weight of unmerged entries untouched, so the effect of misjudging the cluster's mass is to transfer weight between $v^\star(z)$ and u, so constant when the two agree, however wrong the individual attention weights may be. Moreover, both are convex combinations of cached values, so $\|v^\star(z) - u\|$ never exceeds the diameter of the cached value set.

We now bound the two error terms ε_s and ε_v under assumptions of cluster tightness. Concretely, let the cluster's key norms lie within ϵ of each other and, let the values lie within δ of each other,

$$|r_i - r_j| \leq \epsilon \quad \text{and} \quad \|v_i - v_j\| \leq \delta \qquad \text{for all } i, j \in \mathcal{I}. \tag{38}$$

Proposition A.6 (Score Error): Under Eqn. 38, the two-atom score correction Eqn. 8 satisfies

$$|s^\star(z) - s(z)| \;\leq\; \min\left\{ \tfrac{1}{16}\, \epsilon^3 |z|^3,\;\; \epsilon|z| \right\}. \tag{39}$$

The error is therefore cubic in the norm spread for small $\epsilon|z|$, and at worst linear in the tails.

Proof. Let $X = r - \mu_r$ under the cluster's empirical distribution, so $\mathbb{E}[X] = 0$ and X lies in an interval of width at most ϵ, whence $|X| \leq \epsilon$ and, by Popoviciu's inequality, $\sigma_r^2 = \mathrm{Var}[X] \leq \epsilon^2/4$. Subtracting the one-atom terms, define $L(z) = s^\star(z) - \log m - \mu_r z = \log \mathbf{E}\left[e^{Xz}\right]$ and $G(z) = \log\cosh(\sigma_r z)$, so that $\varepsilon_s = G(z) - L(z)$.

For the cubic bound, note $L(0) = G(0) = 0$, $L'(0) = G'(0) = 0$ and $L''(0) = G''(0) = \sigma_r^2$: the two agree to second order by construction, so Taylor's theorem gives $|G(z) - L(z)| \leq \frac{1}{6}|z|^3 \big(\sup_\xi |L'''(\xi)| + \sup_\xi |G'''(\xi)|\big)$. Now $L'''(\xi)$ is the third central moment of the exponentially tilted law p_ξ, which is supported in the same width-ϵ interval, so

$|L'''(\xi)| \le \epsilon \operatorname{Var}_\xi(X) \le \epsilon^3/4$ by Popoviciu again. For the surrogate, $|(\log\cosh)'''| \le 1$ gives $|G'''| \le \sigma_r^3 \le \epsilon^3/8$. Summing, $|G - L| \le \frac{1}{6}\left(\frac{1}{4} + \frac{1}{8}\right)\epsilon^3|z|^3 = \frac{1}{16}\epsilon^3|z|^3$. For the tail bound, $L(z) \le \max_i(r_i - \mu_r)|z| \le \epsilon|z|$ and $G(z) \le \sigma_r|z| \le \epsilon|z|/2$. □

Proposition A.7 (Value Error): The value correction Eqn. 9 is a signed combination of the constituent values, $v(z) = \sum_{i\in\mathcal{I}} c_i v_i$ with

$$c_i = \frac{1 + \alpha\,(r_i - \mu_r)}{m}, \qquad \sum_{i\in\mathcal{I}} c_i = 1, \tag{40}$$

and under Eqn. 38 its error satisfies

$$\|v^\star(z) - v(z)\| \;\le\; \tfrac{1}{2}\,\|c - w(z)\|_1\,\delta \;\le\; \tfrac{1}{2}\,\delta\left(\epsilon|z| + e^{\epsilon|z|} - 1\right), \tag{41}$$

where $w(z)$ are the true intra-cluster attention weights. In particular the value error vanishes when the norms are equal, for any δ.

Proof. Expanding Eqn. 9 with $\mu_v = \frac{1}{m}\sum_i v_i$ and $\Phi = \frac{1}{m}\sum_i r_i v_i$ gives $v(z) = \frac{1}{m}\sum_i(1 + \alpha(r_i - \mu_r))v_i$, and $\sum_i c_i = 1$ because $\sum_i(r_i - \mu_r) = 0$. Since $v^\star(z) = \sum_i w_i(z)v_i$ and both coefficient vectors sum to one, the discrepancy $c - w$ is zero-sum. Split it into its positive and negative parts, $(c-w)_+$ and $(c-w)_-$, which therefore carry equal total mass $T = \frac{1}{2}\|c-w\|_1$. Then

$$\varepsilon_v = \sum_i (c_i - w_i)v_i = T\left(\frac{1}{T}\sum_i (c_i - w_i)_+ v_i - \frac{1}{T}\sum_i (c_i - w_i)_- v_i\right) = T\,(v_+ - v_-)\,,$$

where $v_\pm$ are convex combinations of $\{v_i\}$ and hence both lie in its convex hull. Any two points of that hull are at distance at most δ, so $\|\varepsilon_v\| \le T\delta$, which is the first inequality. For the second, $|\alpha| = |\tanh(\sigma_r z)/\sigma_r| \le |z|$ and $|r_i - \mu_r| \le \epsilon$ give $\sum_i |c_i - 1/m| \le \epsilon|z|$, while $mw_i \in [e^{-\epsilon|z|}, e^{\epsilon|z|}]$ gives $\sum_i |w_i - 1/m| \le e^{\epsilon|z|} - 1$; the triangle inequality combines them. If all norms are equal then $\epsilon = 0$, $w_i \equiv 1/m$ and $c_i \equiv 1/m$, so $\varepsilon_v = 0$. □

Combining Prop. A.5 with Prop. A.6 and Prop. A.7 bounds the end-to-end output error of a merge in terms of the tightness of the cluster alone. Two corollaries follow.

Proposition A.8 (Multi-Cluster Decomposition): Partition the cache into clusters $\mathcal{I}_1, \ldots, \mathcal{I}_C$, counting every unmerged entry as a singleton cluster with zero error. Writing Δ_c and $\tilde{\Delta}_c$ for the attention mass on cluster c before and after merging, and $\varepsilon_v^{(c)}$, $\varepsilon_s^{(c)}$ for its value and score errors,

$$\tilde{y} - y = \sum_c \tilde{\Delta}_c\,\varepsilon_v^{(c)} + \sum_c \left(\tilde{\Delta}_c - \Delta_c\right) v_c^\star, \tag{42}$$

where $v_c^\star$ is the exact value of cluster c. Both Δ and $\tilde{\Delta}$ are probability vectors, so the second sum has zero-sum coefficients.

Proof. Both outputs are convex combinations of the per-cluster exact values, $y = \sum_c \Delta_c v_c^\star$ and $\tilde{y} = \sum_c \tilde{\Delta}_c\big(v_c^\star + \varepsilon_v^{(c)}\big)$; subtracting and collecting terms gives Eqn. 42. □

Proposition A.9 (Error Does Not Accumulate Across Clusters): In the setting of Prop. A.8, let D be the diameter of the cached value set. Then

$$\|\tilde{y} - y\| \;\le\; \max_c \left\|\varepsilon_v^{(c)}\right\| \;+\; \max_c \left|e^{\varepsilon_s^{(c)}} - 1\right| \cdot D.$$

Neither term grows with the number of clusters, so compressing a cache into many merged slots does not multiply the bound by their number.

Proof. The value term of Eqn. 42 is a sub-convex combination, $\sum_c \tilde{\Delta}_c \|\varepsilon_v^{(c)}\| \le \max_c \|\varepsilon_v^{(c)}\|$, since $\sum_c \tilde{\Delta}_c = 1$. For the score term, the coefficients $\tilde{\Delta}_c - \Delta_c$ sum to zero, so by the argument of Prop. A.7 the sum is at most $\frac{1}{2}\|\tilde{\Delta} - \Delta\|_1 D$. Write $x_c = e^{\varepsilon_s^{(c)}}$ and $\kappa = \sum_c \Delta_c x_c$, so that $\tilde{\Delta}_c = \Delta_c x_c/\kappa$ and $\frac{1}{2}\|\tilde{\Delta} - \Delta\|_1 = \frac{1}{2\kappa}\mathbf{E}_\Delta[|x - \kappa|]$. With x_c confined to $[\ell, u] \ni 1$ and $\mathbf{E}_\Delta[x] = \kappa$ fixed, the mean absolute deviation is maximised by a two-point law on the endpoints; maximising the result over κ gives $\kappa = \sqrt{\ell u}$ and value $(\sqrt{u} - \sqrt{\ell})^2/(2\sqrt{\ell u})$. Setting $M = \max(u - 1, 1 - \ell)$, this is at most M, which for $M \ge 1$ because $(\sqrt{u} - \sqrt{\ell})^2 \le u - \ell \le 2M$, and for $M < 1$ because the expression is bounded by $2 - 2\sqrt{1 - M^2} \le 2M^2$. Hence $\frac{1}{2}\|\tilde{\Delta} - \Delta\|_1 \le \max_c |e^{\varepsilon_s^{(c)}} - 1|$. □

A.7 The Slope-Matched Family

We refer to Eqns. 8 and 9 as the symmetric two-atom corrections and to Eqns. 11 and 12 as their scaled variants, and ablate both in Sec. 4. Note that Δr is selected by the sign of z, i.e., the Gibbs weights concentrate on whichever tail z falls.

The family Eqn. 10 is more than just a way to fix the saturation level. It spans the whole (two-atom) ladder. Expanding for small b under the constraint $ab^2 = \sigma_r^2$,

$$a \log\cosh(bz) = \frac{\sigma_r^2}{b^2}\left(\frac{b^2 z^2}{2} - \frac{b^4 z^4}{12} + O(b^6 z^6)\right) \longrightarrow \tfrac{1}{2}\sigma_r^2 z^2 \quad \text{as } b \to 0, \tag{43}$$

so the two ends of the one-parameter family are the two atomic and non-atomic surrogates of Tab. 1. Setting $b = \sigma_r$ gives the symmetric two-atom correction, and setting $b \to 0$ gives the Gaussian one. The scaled variant is therefore not a third rung standing beside the others but a continuous interpolation between them, and the rule $b = \sigma_r^2/\Delta r$ chooses a position on that path from statistics the slot already stores.

Proposition A.10 (The Scaled Variant Retains Two-Token Exactness): For a cluster of two aligned key-value pairs, labelled so that $r_1 \ge r_2$ and hence $\sigma_r \ge 0$, we have $r_{\max} - \mu_r = \mu_r - r_{\min} = \sigma_r$, so $a = 1$ and $b = \sigma_r$ for either sign of z: Eqn. 11 and Eqn. 12 coincide with Eqn. 8 and Eqn. 9, and Propositions 3.2 and 3.3 apply unchanged.

Proof. With $r_1 \ge r_2$, $\mu_r = (r_1 + r_2)/2$ and $\sigma_r = (r_1 - r_2)/2 \ge 0$ we have $r_{\max} - \mu_r = \sigma_r$ and $\mu_r - r_{\min} = \sigma_r$, so $\Delta r = \sigma_r$ regardless of the sign of z. Hence $a = \Delta r^2/\sigma_r^2 = 1$ and $b = \sigma_r^2/\Delta r = \sigma_r$. The choice of which key is labelled k_1 is without loss of generality since swapping the labels only flips the sign of σ_r, and both corrections depend on σ_r through even functions. □

A.8 Logit Space: Dropping the Aligned-Keys Assumption

Every construction in the body of the paper models the distribution of the constituent key norms r_i and converts it to a score through the single scalar quantity $z = q^T\hat{k}/\sqrt{d}$. The underlying random variable, however, is the per-constituent logit $\ell_i = q^T k_i/\sqrt{d}$, and the exact contribution of the cluster is $s^\star = \log\sum_i e^{\ell_i}$ for any key configuration. So for a pair of tokens the two-atom identity

$$s^\star = \log 2 + \bar{\ell} + \log\cosh\left(\tfrac{1}{2}(\ell_1 - \ell_2)\right), \qquad \bar{\ell} = \tfrac{1}{2}(\ell_1 + \ell_2),$$

holds exactly, with no alignment assumption at all. The aligned model of the main text is the special case $\ell_i = r_i z$, under which the logit moments factorise into stored norm statistics times scalar quantity z. This appendix shows that much of the factorisation survives when the alignment assumption is dropped.

First moment. The cluster mean logit is available exactly as $\frac{1}{m}\sum_i \ell_i = q^T\bar{k}/\sqrt{d}$, which requires only one dot product on the mean key $\bar{k} = \frac{1}{m}\sum_i k_i$. This vector statistic is additive over disjoint clusters and hence order-invariant in the sense of Prop. A.12. Note that $\|\bar{k}\|$ is different to μ_r maintained from the aligned key assumption, i.e., $\|\bar{k}\| \le \frac{1}{m}\sum_i \|k_i\| = \mu_r$.

Second moment. The logit variance over the cluster is $\mathrm{Var}_i(\ell_i) = q^T \Sigma_{kk}\, q/d$ with $\Sigma_{kk} = \mathbf{E}\left[(k-\bar{k})(k-\bar{k})^T\right]$, which is a $d \times d$ matrix and hence too big to store. Splitting q into its component along $\hat{k}$ and the orthogonal remainder $q_\perp$, we can compute term along $\hat{k}$ as $\sigma_r^2 z^2$ under key alignment, the cross term $2z\,\hat{k}^T \Sigma_{kk}\, q_\perp/\sqrt{d}$, which has zero mean over the directions of $q_\perp$ and is therefore dropped, and the orthogonal term modelled isotropically at the full trace, to give

$$\mathrm{Var}_i(\ell_i) \;\approx\; \sigma_r^2 z^2 \;+\; \frac{\mathrm{tr}\,\Sigma_{kk}}{d-1} \cdot \frac{\|q_\perp\|^2}{d}, \qquad \mathrm{tr}\,\Sigma_{kk} = \tfrac{1}{m}\textstyle\sum_i \|k_i\|^2 - \|\bar{k}\|^2. \tag{44}$$

The new ingredients cost one additional scalar per slot. The trace needs no new statistic since $\mathrm{tr}\,\Sigma_{kk} = \frac{1}{m}\sum_i \|k_i\|^2 - \|\bar{k}\|^2 = \sigma_r^2 + \mu_r^2 - \|\bar{k}\|^2$. The trace is zero for singletons and inherits Prop. A.12 from the statistics it is built from. Every surrogate of Tab. 1 then upgrades by consuming the argument $\tilde{x} = \mathrm{Var}_i(\ell_i)^{1/2}$ in place of $\sigma_r|z|$. The surrogate family is unchanged, and the moments matched are now those of the logit distribution rather than of the norm distribution scaled by z. What is lost is the direction of $q_\perp$ relative to the eigenvectors of Σ_{kk}.

Remark A.11 (Exactness): Singletons remain exact ($\Sigma_{kk} = 0$). Pair exactness changes in its kind. Prop. 3.2 is pointwise under alignment, whereas in logit space the two-atom form is exact given the true logit gap $q^T(k_1 - k_2)/\sqrt{d}$, which Eqn. 44 estimates only in distribution over the direction of $q_\perp$. The exactness is therefore in distribution rather than pointwise.

A.9 The Merge Algorithm

The corrections are maintained through the power sums of Eqn. 16. Let $\mathcal{I}$ be the subset of indices of cluster constituents. Writing $S(\mathcal{I}) = (S_0, S_r, S_{rr}, S_v, S_{rv})$ for the tuple of sums over index set $\mathcal{I}$, disjoint index sets satisfy

$$S(\mathcal{I}_1 \cup \mathcal{I}_2) = S(\mathcal{I}_1) + S(\mathcal{I}_2) \tag{45}$$

componentwise, so a merge is a single vector addition and a fresh token is the tuple $S(\{t\}) = (1, r_t, r_t^2, v_t, r_t v_t)$. The scaled variant of Sec. 3.2 additionally tracks $r_{\max}$ and $r_{\min}$, which are not sums but combine by $\max$ and $\min$, likewise associative and commutative.

The algorithms below use a bar for quantities as they are *stored*, to keep them distinct from the idealised objects of Sec. 3. A slot holds the key $\bar{k}_i = \mu_{r,i}\tilde{k}_i$ at its true magnitude rather than the unit direction $\tilde{k}_i$, together with the value $\mu_{v,i}$. Correspondingly, $\bar{z}_i = q^T\bar{k}_i/\sqrt{d} = \mu_{r,i} z_i$ is the logit an unmodified matrix product produces, a factor of $\mu_{r,i}$ larger than the unit-direction logit z_i in which the formulas of Sec. 3.1 are written. A cache entry is created by CacheAddNew, entries are combined by CacheMerge, and attention is computed by MergedAttention.

function CacheAddNew(k_t, v_t)
  $r_t \leftarrow \|k_t\|$
  $\bar{K} \leftarrow \bar{K} \cup \{k_t\}, \quad \bar{V} \leftarrow \bar{V} \cup \{v_t\}$
  $S \leftarrow S \cup \{(1,\ r_t,\ r_t^2,\ v_t,\ r_t v_t)\}, \quad r_{\max,t} = r_{\min,t} \leftarrow r_t$
end function

function CacheMerge(i, j)
  $S \leftarrow S_i + S_j$ ▷ componentwise, Eqn. 45
  $r_{\max} \leftarrow \max\{r_{\max,i}, r_{\max,j}\}, \quad r_{\min} \leftarrow \min\{r_{\min,i}, r_{\min,j}\}$
  $\bar{k} \leftarrow (S_r/S_0) \cdot \mathrm{normalize}\left(S_{0,i}\bar{k}_i + S_{0,j}\bar{k}_j\right)$ ▷ magnitude μ_r, direction from the merge
  replace slots i and j by the single slot $(\bar{k},\ S_v/S_0,\ S,\ r_{\max},\ r_{\min})$
end function

function MERGEDATTENTION($q, \bar{K}, \bar{V}, S$)
  recover $(m_i, \mu_{r,i}, \sigma^2_{r,i}, \Phi_i)$ from S_i by Eqn. 17
  compute stored-key logits, $\bar{z}^T = q^T \bar{K}/\sqrt{d}$ ▷ $\|\bar{k}_i\| = \mu_{r,i}$, so $\bar{z}_i = \mu_{r,i} z_i$
  $c_i \leftarrow \sigma_{r,i} \bar{z}_i / \mu_{r,i}$ ▷ $= \sigma_{r,i} z_i$, the unit-direction logit of Sec. 3
  recalibrate scores, $\tilde{z}_i \leftarrow \log(m_i) + \bar{z}_i + \log\cosh(c_i)$
  recalibrate values, $\tilde{v}_i \leftarrow (1 - \alpha_i \mu_{r,i})\, \mu_{v,i} + \alpha_i \Phi_i$ with $\alpha_i = \tanh(c_i)/\sigma_{r,i}$
  return $y = \mathrm{softmax}(\tilde{z})\, \tilde{V}$ ▷ $\tilde{V}$ has rows $\tilde{v}_i$
end function

Storing the key as $\bar{k} = \mu_r \tilde{k}$, rather than keeping the direction $\tilde{k}$ and the magnitude μ_r separately, lets the logit be computed by the unmodified matrix product; one division by $\mu_{r,i}$ per slot then recovers the formulas of Sec. 3.1. Note that $\mu_r = S_r/S_0$ is obtained from the sums and the key is then *reconstructed* at that magnitude. Taking instead the norm of the merged direction, $\|S_{0,i}\bar{k}_i + S_{0,j}\bar{k}_j\|/S_0$, would understate μ_r whenever the merged keys are not exactly parallel (triangle inequality), and because μ_r feeds every subsequent merge into the same slot the bias would compound.

Proposition A.12 (Order-Invariant Merging): Let a cluster with constituent index set $\mathcal{I}$ be produced by any sequence of CACHEADDNEW and CACHEMERGE operations. Then its statistics $(m, \mu_r, \sigma_r^2, r_{\max}, r_{\min}, \mu_v, \Phi)$ equal those computed in one batch from $\{(r_i, v_i)\}_{i\in\mathcal{I}}$, independent of the order of operations. Consequently both corrections, Eqn. 8 and Eqn. 9, depend on $\mathcal{I}$ alone. In particular, two clusters that have themselves already been merged, over mutually exclusive index sets $\mathcal{I}_1$ and $\mathcal{I}_2$, may be merged with each other, and the result agrees with merging $\mathcal{I}_1 \cup \mathcal{I}_2$ in one step.

Proof. CACHEMERGE accumulates the sums Eqn. 16, and by Eqn. 45 the tuple $S(\mathcal{I})$ for any assembly order is the same componentwise sum over $\mathcal{I}$; addition is commutative and associative, and $\max$ and $\min$ likewise, so $S(\mathcal{I})$, $r_{\max}$ and $r_{\min}$ depend only on $\mathcal{I}$. The moments follow from $S(\mathcal{I})$ by Eqn. 17, and the corrections from the moments. □

Remark A.13 (The key direction): The stored key *direction* is the one quantity that is not a sum, and correspondingly the one that is not exactly order-invariant: CACHEMERGE renormalises it at every merge, so for keys that are not exactly parallel the direction obtained by merging three or more constituents depends weakly on the order. Under the Key Alignment assumption the renormalisation is a no-op and invariance is exact; more generally the discrepancy is of second order in the angular spread within the cluster, which is precisely the quantity the merge threshold controls.

A.10 RUNNING TIME

The correction formulae require changes to the attention kernel. Merging is therefore a trade-off between the compute saved by a smaller cache and the additional compute needed for correction. We measure throughput in Fig. 4 of the prefill and decode phases versus the merge threshold for variant 6 of Tab. 2, for Llama-3.1-8B on RULER at four context lengths. Merging is chunked, with chunk and search window of 4096 tokens, and each point is the mean over repeated runs. The implementation modifies a HuggingFace model with our fused Triton kernels, and all measurements were collected on a single H100 GPU with 96 GB of VRAM.

Two trends are clearly visible in the figure. First, as expected, throughput falls as the threshold rises since a higher threshold results in fewer merges and hence a larger cache. Second, at aggressive thresholds the decode throughput is roughly constant in context length: the compressed cache the decoder reads grows far more slowly than the context it represents. The dashed reference lines show baseline throughput of the unmodified HuggingFace code. At aggressive thresholds the merged cache decodes faster than the unmodified model at every context length measured, and its prefill overtakes at the longest contexts, so merging becomes a throughput gain as well as a memory saving.

A.11 CONSTRUCTION OF THE CLUSTER IN FIGURE 1

The cluster use to produce the curves in both panels of Fig. 1 is synthetic but shaped to match a measured property of real merged slots.

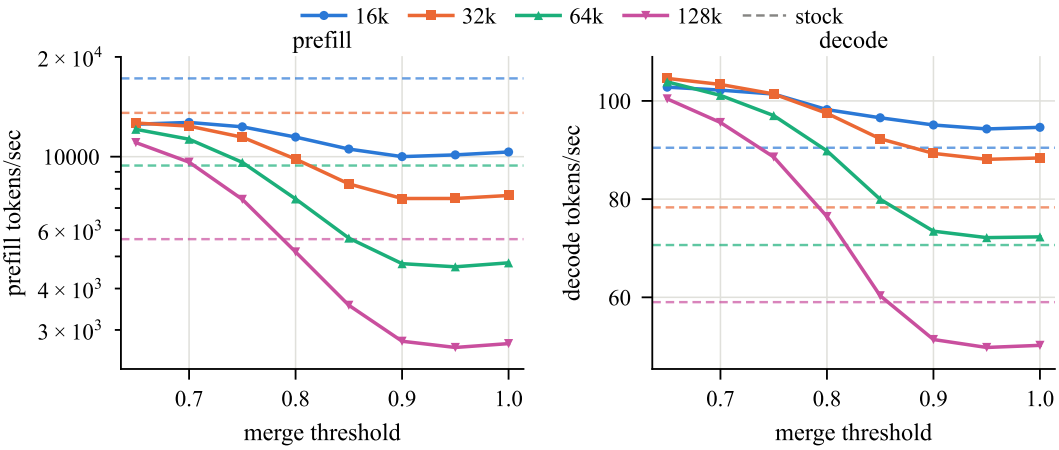

Figure 4: End-to-end throughput against merge threshold for variant 6 (Llama-3.1-8B on RULER), one curve per context length; dashed lines mark the unmodified model at the same context length. Left: prefill tokens/sec (log scale). Right: decode tokens/sec.

The cluster has $m = 8$ constituent keys sampled from two distributions. Two of the keys (a fraction $p = 0.25$) are drawn from $\mathcal{N}(7.4, 0.15^2)$ and the remainder from $\mathcal{N}(4.6, 0.15^2)$ to give a minority of high-norm tokens merged with near-duplicate fillers. This choice gives $r_{\max} - \mu_r = 1.82\,\sigma_r$, which is within the 1.42–1.87 range of medians that we measured for real merged slots of $m \in [5, 32]$.

For the left panel's grey band the constituents additionally receive directions, i.e., unit vectors drawn from a spherically cap around a common centre at $d = 128$. The width of the cap is determined by a minimum pairwise cosine target of 0.75. Keys are $k_i = r_i \hat{k}_i$ with the norms above; the slot direction is the norm-weighted mean of the $\hat{k}_i$, renormalised, exactly as the merge recursion builds it. Queries are drawn at fixed total norm $\|q\| = 1.3\, z_{\max}\sqrt{d}$ with the on-axis component set by the panel's abscissa ($z = q^T \hat{k}/\sqrt{d}$) and the orthogonal remainder uniform on the sphere of the remaining norm. The band is the 5–95% range of the true correction over 400 such queries per z. The logit-space curve is computed from the same keys and same queries.

All samples use fixed random seeds and the panels span $z \in [-2, 2]$. The aligned-keys curves of the left panel are computed from the norms alone whereas the grey band and the logit-space curve additionally use the directions and queries above.