BRIDGING KV-CACHE QUANTIZATION AND LINEAR ATTENTION: FROM THEORY TO PRETRAINED WEIGHT MIGRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

KV-cache quantization and linear attention are two representative approaches to tackling the storage and computational costs of Transformers. KV-cache quantization compresses individual KV entries into discrete codes but retains all entries, whereas linear attention recurrently aggregates multiple historical KV contributions into a fixed-size continuous state but can introduce interference. This contrast raises the question of whether per-KV compression and multi-KV aggregation can be bridged within a single mechanism for efficient attention. We identify RAM-Net as such a bridge through soft assignments over a discrete address space. These assignments determine recurrent updates to the continuous slot state associated with each address. Under a restricted RAM-Net construction, we prove that soft address assignments extend hard quantized matching to a separable read-write overlap that locally approximates full-attention similarity and supports recurrent aggregation. These connections further enable Transformer-to-RAM-Net weight migration through a new path based on a soft-quantized intermediate construction. Across nine pretrained Transformer models from 0.3B to 7B parameters, RAM-Net recovers an average of 87.1% of the teachers' accuracy gains over random guessing across six commonsense and knowledge tasks using only a 500M-token budget per model.

1 INTRODUCTION

Transformer models have become the dominant architecture for language modeling (Vaswani et al., 2017; Brown et al., 2020). However, the memory use and per-query computation of full attention grow with context length because all historical keys and values are stored in the KV cache and accessed for each query. Two representative approaches reduce these costs in different ways. KV-cache quantization compresses individual KV entries into discrete codes (Liu et al., 2024; Savkin et al., 2025), but retains all entries, so memory still grows with context length. Linear attention instead recurrently aggregates multiple historical KV contributions into a fixed-size continuous state (Katharopoulos et al., 2020; Choromanski et al., 2021), while this aggregation can cause interference among them. This contrast raises the question of whether per-KV compression and multi-KV aggregation can be bridged within a single mechanism for efficient attention. We identify RAM-Net (Xiao et al., 2026) as such a bridge, combining a discrete address space with recurrent continuous slot states. This bridge provides a new view of efficient attention and a new path for weight migration.

We theoretically explain how RAM-Net's soft address assignments connect discrete-code quantization, full attention, and linear attention within a restricted subset of RAM-Net. These assignments generalize hard code selection in KV quantization, recovering quantized-key matching in the hard-write limit while reads remain soft. We further prove that the overlap between soft read and write assignments locally approximates the exponential dot-product similarity of full attention without reconstructing keys from discrete codes. Moreover, the address overlap is exactly separable into read and write terms. This separability enables historical KV contributions to be recurrently aggregated into continuous slot states, yielding the linear-style continuous state construction.

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.

These theoretical connections further enable practical Transformer-to-RAM-Net migration across a broad range of pretrained Transformers. Unlike prior Transformer-to-linear migration methods (Goldstein et al., 2025; Bick et al., 2024), our migration introduces a soft-quantized intermediate attention model that replaces exponential dot-product matching with separable address overlap. To construct the soft-quantized intermediate attention model, we first compress and align the Transformer's query and key projections to match its attention patterns. We then fold a codebook into the aligned projections to implement soft quantization, while copying the remaining shared parameters from the Transformer. Finally, we initialize RAM-Net with the intermediate model's parameters and progressively fine-tune it to compensate for approximation errors and architectural differences.

We evaluate the migration design on nine publicly available pretrained Transformer models covering five model families, multi-head attention (MHA) and grouped-query attention (GQA) (Ainslie et al., 2023) configurations, and scales from 0.3B to 7B parameters. With a migration budget of only 500M tokens per model, RAM-Net recovers an average of 87.1% of the teachers' accuracy gains over random guessing across six commonsense and knowledge tasks. Ablations on the intermediate construction show that the theory-guided design improves attention alignment with the teacher and generally leads to lower output error during head-level recovery. Together with the downstream gains from staged recovery under the same total token budget, these findings support the practical value of our theory-guided migration design.

2 Background

In sequence modeling, attention uses learnable projections to map an input hidden state $\mathbf{h}_t$ to a query $\mathbf{q}_t$, a key $\mathbf{k}_t$, and a value $\mathbf{v}_t$. Under the test-time training (TTT) view (Sun et al., 2025), attention constructs a query-to-value predictor f_t from historical KV samples $\{(\mathbf{k}_s, \mathbf{v}_s)\}_{s\le t}$ and evaluates it on the current query as $\mathbf{o}_t = f_t(\mathbf{q}_t)$. The predictor f_t is therefore defined over the query-key (QK) matching space.

For causal full attention (Vaswani et al., 2017), its predictor forms a weighted sum of historical values using softmax-normalized dot-product similarities between the query and historical keys:

$$f_t(\mathbf{q}) = \sum_{s\le t} \alpha_{t,s}(\mathbf{q})\mathbf{v}_s, \quad \alpha_{t,s}(\mathbf{q}) = \frac{\exp(\mathbf{q}^\top \mathbf{k}_s/\tau)}{\sum_{j\le t}\exp(\mathbf{q}^\top \mathbf{k}_j/\tau)}. \tag{1}$$

Here $\tau > 0$ is the softmax temperature, and $\alpha_{t,s}(\mathbf{q})$ are the softmax scores. Since softmax assigns larger weights to keys with larger dot-product similarities, full attention can be viewed as a **soft nearest-neighbor (Soft-NN) predictor** in QK space. This Soft-NN predictor uses the complete set of historical KV samples, so its storage and per-query computation grow with context length. Fig. 1 illustrates how full attention is modified by KV-cache quantization and linear attention, and how RAM-Net connects these two directions, as detailed in the following three subsections.

2.1 Quantized KV Caches: Discrete-Code Soft-NN Predictor

KV-cache quantization compresses historical KV samples into a finite set of numerical levels or codewords. Quantizing keys partitions QK space into regions whose keys share the same reconstruction. For example, scalar quantization encodes individual coordinates (Liu et al., 2024; Micikevicius et al., 2022), whereas vector quantization jointly encodes coordinate groups (Savkin et al., 2025; Son et al., 2025). Let $\mathcal{Q}_\mathrm{K}$ and $\mathcal{Q}_\mathrm{V}$ denote the quantize-dequantize operators for keys and values. This yields a **discrete-code Soft-NN predictor**:

$$f_t(\mathbf{q}) = \sum_{s\le t} \alpha_{t,s}(\mathbf{q})\mathcal{Q}_\mathrm{V}(\mathbf{v}_s), \quad \alpha_{t,s}(\mathbf{q}) = \frac{\exp\big(\mathbf{q}^\top \mathcal{Q}_\mathrm{K}(\mathbf{k}_s)/\tau\big)}{\sum_{j\le t}\exp(\mathbf{q}^\top \mathcal{Q}_\mathrm{K}(\mathbf{k}_j)/\tau)}. \tag{2}$$

This predictor still stores all historical KV samples, but quantization reduces the storage required for each sample. The resulting matching quality depends on how accurately the reconstructed keys preserve the original similarities between query and keys.

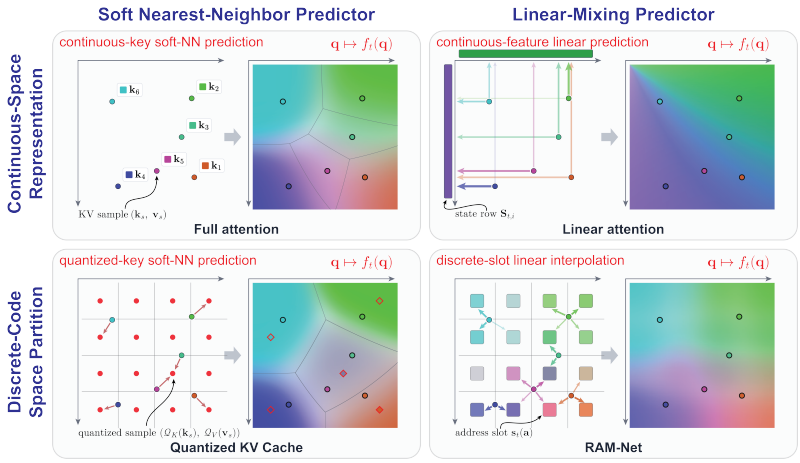

Figure 1: **Comparison of attention predictors in QK space.** For each sub-plot, the left shows how historical KV information is stored or aggregated, while the right visualizes the resulting $f_t(\mathbf{q})$ over QK space. RAM-Net retains the discrete-code representation of KV-cache quantization and the continuous-state aggregation of linear attention.

2.2 Linear Attention: Linear Predictor with Continuous Features

Kernelized linear attention replaces the exponential dot-product similarity in Eq. (1) with a separable query-key similarity given by $\phi(\mathbf{q})\phi(\mathbf{k})^\top$ (Katharopoulos et al., 2020; Choromanski et al., 2021). This separable form allows historical KV contributions to be recurrently aggregated into a fixed-size and continuous-feature state $\mathbf{S}_t$. It yields a **linear predictor**:

$$f_t(\mathbf{q}) \propto \phi(\mathbf{q})\mathbf{S}_t, \quad \mathbf{S}_t = \Lambda_t \mathbf{S}_{t-1} + \phi(\mathbf{k}_t)^\top \mathbf{v}_t. \tag{3}$$

Here, Λ_t controls how the previous state is retained or modified (Yang et al., 2024; Gu & Dao, 2024; Yang et al., 2025). Maintaining this fixed-size state makes storage and per-query computation independent of context length. However, the fixed state size limits the predictor's capacity for fine-grained matching to historical keys in the QK space.

2.3 RAM-Net: Discrete-Address Linear-Interpolation Predictor

RAM-Net (Xiao et al., 2026) consists of a finite set of discrete addresses $\mathbf{a} \in \mathcal{A}$, each indexing a slot that stores a continuous state $\mathbf{s}_t(\mathbf{a})$. Keys and queries produce soft write and read assignments over discrete addresses, respectively, determining which slots are updated and read. The discrete addresses act as codes, and the assignments softly partition QK space. Each slot stores a continuous state that recurrently aggregates historical KV contributions as in linear attention. Specifically, $\mathbf{k}_t$ and $\mathbf{q}_t$ are each split into U groups of address logits, and Product Softmax combines their group-wise softmax probabilities into write and read distributions over addresses. Only the Top-K probabilities are retained for efficiency, which yields sparse write weights $w_t(\mathbf{a})$ and read weights $r_t(\mathbf{a})$. Gated Sparse Update (GSU) then recurrently updates the selected slots with $\mathbf{v}_t$. The predictor and GSU rule are given by

$$f_t(\mathbf{q}_t) \propto \sum_{\mathbf{a}\in\mathcal{A}} r_t(\mathbf{a})\mathbf{s}_t(\mathbf{a}), \quad \begin{cases} m_t(\mathbf{a}) = \sigma(\mathrm{logit}(1 - w_t(\mathbf{a})) - \gamma_t) \cdot m_{t-1}(\mathbf{a}) + w_t(\mathbf{a}) \\ \mathbf{s}_t(\mathbf{a}) = \mathbf{s}_{t-1}(\mathbf{a}) + \frac{w_t(\mathbf{a})}{m_t(\mathbf{a})+\epsilon}\left(\mathbf{v}_t - \mathbf{s}_{t-1}(\mathbf{a})\right) \end{cases}. \tag{4}$$

Here, $f_t(\mathbf{q}_t)$ is defined by the address overlap of write and read weights, γ_t is an input-dependent scalar that modulates the forgetting tendency across all slots, $m_t(\mathbf{a})$ is a per-slot scalar that tracks the slot's inertia against new writes, and ϵ stabilizes the denominator. The read weights linearly combine the continuous states stored at discrete addresses to produce the output, defining a **finite discrete-address linear-interpolation predictor**. RAM-Net therefore combines the discrete-code representation of KV quantization with the recurrent aggregation of linear attention. Sec. 3 establishes this bridge theoretically, showing how a separable address kernel locally approximates Soft-NN similarity and how GSU extends cumulative averaging with forgetting while retaining linear aggregation.

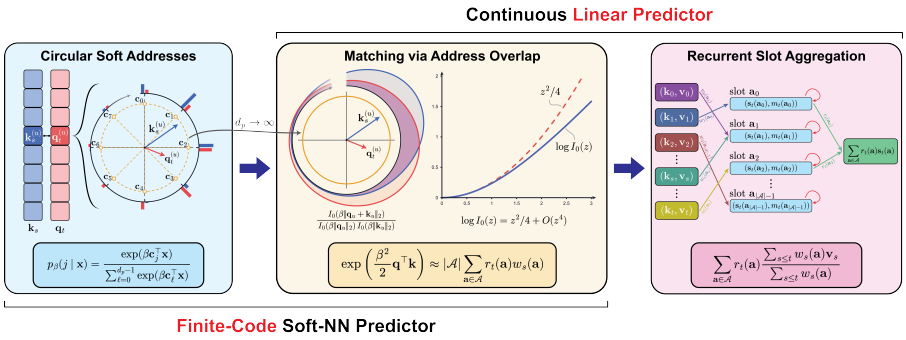

Figure 2: **Overview of Sec. 3.** RAM-Net connects discrete-code Soft-NN and linear predictors through the separable overlap between read and write distributions built on circular soft addressing, which locally approximates Soft-NN matching.

3 Bridging Discrete-Code Soft-NN and Continuous Linear Predictors

This section establishes how the address space of RAM-Net connects discrete-code Soft-NN predictors and continuous-feature linear predictors, as illustrated in Fig. 2. Sec. 3.1 first introduces soft assignment distributions over a finite address space as a **discrete-code representation** of queries and keys and adopts a restricted circular construction for theoretical analysis. Sec. 3.2 then shows that the overlap between soft read and write distributions locally approximates exponential dot-product similarity, enabling discrete-code Soft-NN matching in the address space without explicit reconstruction. The address overlap is also separable into read and write terms. Building on this separability, Sec. 3.3 shows how a **linear predictor** aggregates historical KV contributions into recurrent slot states and how GSU extends the cumulative updates with retention.

3.1 Discrete-Code Representation via Circular Soft Addressing

Address space. In RAM-Net, we define the address space as $\mathcal{A} = \{0, \dots, d_p - 1\}^U$, where each address $\mathbf{a} = (a^{(0)}, \dots, a^{(U-1)})$ consists of U factors, and each factor $a^{(u)}$ takes one of d_p possible code indices. Each address $\mathbf{a} \in \mathcal{A}$ indexes a slot. The read and write projections $\mathbf{W}_r$ and $\mathbf{W}_w$ each map the hidden state $\mathbf{h}_t$ to U groups of logits, one for each factor. Product Softmax normalizes each group and multiplies the resulting factor-wise probabilities to form a distribution over $\mathcal{A}$.

Restricted projection. For the u-th factor, let $\mathbf{W}_r^{(u)}$ and $\mathbf{W}_w^{(u)}$ denote the read and write projection blocks that produce the corresponding logits in Product Softmax. We restrict these projection blocks to the forms $\mathbf{W}_r^{(u)} = \beta \mathbf{C}^\top \widetilde{\mathbf{W}}_Q^{(u)}$ and $\mathbf{W}_w^{(u)} = \beta \mathbf{C}^\top \widetilde{\mathbf{W}}_K^{(u)}$, respectively. Here, $\mathbf{C} \in \mathbb{R}^{2 \times d_p}$ is a shared and fixed codebook, and the scalar $\beta \geq 0$ controls the concentration of the code assignments. This restricted parameterization defines a special case of RAM-Net in which each read and write projection block has rank at most two. This parameterization introduces two-dimensional intermediate query and key vectors, with $\widetilde{\mathbf{W}}_Q^{(u)}, \widetilde{\mathbf{W}}_K^{(u)} \in \mathbb{R}^{2 \times d_{\text{model}}}$ and $\mathbf{q}_t^{(u)}, \mathbf{k}_s^{(u)} \in \mathbb{R}^2$. $\widetilde{\mathbf{W}}_Q^{(u)}$ and $\widetilde{\mathbf{W}}_K^{(u)}$ map the hidden states to $\mathbf{q}_t^{(u)} = \widetilde{\mathbf{W}}_Q^{(u)} \mathbf{h}_t$ and $\mathbf{k}_s^{(u)} = \widetilde{\mathbf{W}}_K^{(u)} \mathbf{h}_s$. This two-dimensional structure also supports an approximate correspondence between RoPE rotations and RAM-Net's cyclic address shifts (CAPE), as described in Appendix A.6 and used for weight migration in Sec. 4.

Circular code with soft assignments. For our theoretical construction, we choose $\mathbf{C}$ as a circular codebook with d_p codes uniformly spaced on the unit circle. Each intermediate vector $\mathbf{q}_t^{(u)}, \mathbf{k}_s^{(u)} \in \mathbb{R}^2$ is scored against these shared codes. For any $\mathbf{x} \in \mathbb{R}^2$, the code vectors and the corresponding soft assignment are defined as

$$\mathbf{c}_j = \left(\cos \frac{2\pi j}{d_p}, \sin \frac{2\pi j}{d_p} \right)^\top, \quad p_\beta(j \mid \mathbf{x}) = \frac{\exp(\beta \mathbf{c}_j^\top \mathbf{x})}{\sum_{\ell=0}^{d_p-1} \exp(\beta \mathbf{c}_\ell^\top \mathbf{x})}, \quad j = 0, \dots, d_p - 1. \tag{5}$$

The circular codebook is therefore $\mathbf{C} = \begin{bmatrix} \mathbf{c}_0 & \cdots & \mathbf{c}_{d_p-1} \end{bmatrix}$, with each scaled codeword $\beta \mathbf{c}_j$ having norm β. Product Softmax combines the assignments from the U factors to define the read and write

distributions over $\mathcal{A}$ as $r_t(\mathbf{a}) = \prod_{u=0}^{U-1} p_\beta(a^{(u)} \mid \mathbf{q}_t^{(u)})$ and $w_s(\mathbf{a}) = \prod_{u=0}^{U-1} p_\beta(a^{(u)} \mid \mathbf{k}_s^{(u)})$. As $\beta \to \infty$, each factor-wise assignment concentrates all probability mass on its highest-scoring code. Product Softmax therefore concentrates on the address formed by these factor-wise maximizing code indices. Thus, hard code assignment in discrete-code quantization is a limiting case of our soft address construction, as shown in Appendix A.5.

3.2 Approximation to Soft-NN Predictors via Address Overlap

The circular soft addressing distribution in Sec. 3.1 provides a discrete-code representation of queries and keys. We write $\mathbf{q}_t = (\mathbf{q}_t^{(0)}, \ldots, \mathbf{q}_t^{(U-1)})$ and $\mathbf{k}_s = (\mathbf{k}_s^{(0)}, \ldots, \mathbf{k}_s^{(U-1)})$ for the $2U$-dimensional representations. We show how the address distribution overlap can approximate the exponential dot-product similarity $\exp\big(\mathbf{q}_t^\top \mathbf{k}_s/\tau\big)$ used by the Soft-NN predictors of full attention and KV-cache quantization. The following theorem establishes this approximation in a local regime.

Theorem 1 (Local approximation of Soft-NN matching)**.** *Consider the circular soft-addressing construction in Sec. 3.1. For fixed U and $d_p \geq 4$, in the local regime $\max_u \beta(\|\mathbf{q}_t^{(u)}\|_2 + \|\mathbf{k}_s^{(u)}\|_2) \to 0$, the address overlap satisfies*

$$\exp\left(\frac{\beta^2}{2}\mathbf{q}_t^\top \mathbf{k}_s\right) \approx \kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s) = |\mathcal{A}| \sum_{\mathbf{a}\in\mathcal{A}} r_t(\mathbf{a}) w_s(\mathbf{a}). \tag{6}$$

Proof sketch. Define $Z_{d_p}(\mathbf{x}) = \frac{1}{d_p}\sum_{j=0}^{d_p-1} \exp(\mathbf{c}_j^\top \mathbf{x})$. Using the factorized Product Softmax distributions defined in Sec. 3.1, the address overlap can be written as

$$\kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s) = \prod_{u=0}^{U-1}\left[d_p \sum_{j=0}^{d_p-1} p_\beta(j \mid \mathbf{q}_t^{(u)}) p_\beta(j \mid \mathbf{k}_s^{(u)})\right] = \prod_{u=0}^{U-1} \frac{Z_{d_p}\Big(\beta(\mathbf{q}_t^{(u)} + \mathbf{k}_s^{(u)})\Big)}{Z_{d_p}(\beta\mathbf{q}_t^{(u)}) Z_{d_p}(\beta\mathbf{k}_s^{(u)})}. \tag{7}$$

As $d_p \to \infty$, the uniformly spaced circular codes approach a continuous unit circle, giving $Z_{d_p}(\mathbf{x}) \to I_0(\|\mathbf{x}\|_2)$, where I_0 is the modified Bessel function of the first kind of order zero. In the local regime, substituting $\log I_0(z) = z^2/4 + O(z^4)$ into the logarithm of Eq. (7) in this limit and expanding the squared norms gives

$$\log \kappa_\infty(\mathbf{q}_t, \mathbf{k}_s) = \lim_{d_p\to\infty} \log \kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s) = \frac{\beta^2}{2}\mathbf{q}_t^\top \mathbf{k}_s + O\left(\beta^4 \sum_{u=0}^{U-1} \big(\|\mathbf{q}_t^{(u)}\|_2 + \|\mathbf{k}_s^{(u)}\|_2\big)^4\right). \tag{8}$$

Appendix A.1 provides explicit error bounds and shows that, the finite-codebook correction is absorbed by the remainder in Eq. (8). The same expansion therefore holds for $\log \kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s)$. Exponentiating yields Eq. (6). □

The scaling factor $|\mathcal{A}|$ is introduced for the kernel comparison in the proof and does not affect the normalized attention weights. Intuitively, even after rescaling, the address overlap remains bounded by $|\mathcal{A}|$, whereas the exponential dot-product kernel is unbounded, motivating a local approximation. Matching the local expansion to the target kernel gives $\beta^2/2 = 1/\tau$ and hence $\beta^\star = \sqrt{2/\tau}$. For $\tau = \sqrt{d_{\text{head}}}$, this becomes $\beta^\star = \sqrt{2}\, d_{\text{head}}^{-1/4}$. We use $\beta^\star$ to guide weight migration in Sec. 4. Together, the local approximation in Theorem 1 and the separability of the address overlap establish RAM-Net's connections to continuous Soft-NN, discrete-code Soft-NN, and linear predictors. The local approximation allows RAM-Net to preserve Soft-NN matching in the discrete-code address space without the key reconstruction used in KV-cache quantization. The address overlap is exactly separable into read and write terms, enabling the recurrent linear aggregation developed in Sec. 3.3.

3.3 Linear Predictor via Slot Aggregation

Using the local approximation in Theorem 1 over historical keys and the separable feature form for linear predictors provided by the address overlap in Eq. (6), we aggregate historical KV contributions by address, giving

$$\frac{\sum_{s\le t}\exp\left(\beta^2/2\cdot\mathbf{q}_t^\top\mathbf{k}_s\right)\mathbf{v}_s}{\sum_{s\le t}\exp\left(\beta^2/2\cdot\mathbf{q}_t^\top\mathbf{k}_s\right)}\approx\frac{\sum_{s\le t}\kappa_{\mathcal{A}}(\mathbf{q}_t,\mathbf{k}_s)\mathbf{v}_s}{\sum_{s\le t}\kappa_{\mathcal{A}}(\mathbf{q}_t,\mathbf{k}_s)}=\frac{\sum_{\mathbf{a}}r_t(\mathbf{a})m_t(\mathbf{a})\mathbf{s}_t(\mathbf{a})}{\sum_{\mathbf{a}}r_t(\mathbf{a})m_t(\mathbf{a})}. \tag{9}$$

Here, $m_t(\mathbf{a})=\sum_{s\le t}w_s(\mathbf{a})$ is the cumulative write mass of slot $\mathbf{a}$, and $\mathbf{s}_t(\mathbf{a})=\frac{\sum_{s\le t}w_s(\mathbf{a})\mathbf{v}_s}{m_t(\mathbf{a})}$ is its cumulative weighted mean of values, detail in Appendix A.2. The rightmost expression defines a mass-aware readout whose read coefficients are proportional to $r_t(\mathbf{a})m_t(\mathbf{a})$, so the contribution of each slot depends on both its query read score and accumulated write mass.

The GSU rule in Eq. (4) extends this cumulative form by introducing γ_t to control the retention of historical contributions. With $m_0(\mathbf{a})=0$, $\mathbf{s}_0(\mathbf{a})=\mathbf{0}$, and $\epsilon=0$, taking $\gamma_t\to-\infty$ at every step removes this forgetting effect and reduces GSU to cumulative averaging:

$$m_t(\mathbf{a})=m_{t-1}(\mathbf{a})+w_t(\mathbf{a}),\qquad \mathbf{s}_t(\mathbf{a})=\frac{m_{t-1}(\mathbf{a})\mathbf{s}_{t-1}(\mathbf{a})+w_t(\mathbf{a})\mathbf{v}_t}{m_t(\mathbf{a})}. \tag{10}$$

For finite γ_t, GSU retains its forgetting mechanism, where m_t and $\mathbf{s}_t$ track the retained write mass and its corresponding weighted value mean. The resulting predictor remains linear in the read-address features, while GSU updates the slot states online with controlled retention.

Original RAM-Net reads these slot states using $\sum_{\mathbf{a}}r_t(\mathbf{a})\mathbf{s}_t(\mathbf{a})$, omitting the write-mass reweighting. The mass-aware readout ablation in Appendix B.3(Tables 7 and 8) shows that the mass-aware readout generally matches the Transformer's attention more closely before head-level recovery, but tends to yield slightly higher perplexity after migration than the original readout. It may be related to the coupling of read weights to forgetting dynamics and a potential mismatch between mass reweighting and Top-K selection based only on $r_t(\mathbf{a})$.

4 Transformer-to-RAM-Net Weight Migration

The connections established in Sec. 3 provide a basis for migrating pretrained Transformer weights to RAM-Net. The migration proceeds in two stages. First, we adapt the teacher's query-key matching to soft-quantized intermediate attention model, aiming to preserve its token-to-token attention patterns. Second, we enable RAM-Net's recurrent updates and progressively fine-tune the model to reduce the remaining output discrepancies.

4.1 Migration to Soft-Quantized Intermediate Attention

Soft-quantized intermediate attention uses the overlap of circular read and write address assignments in place of exponential dot-product similarity (Secs. 3.1 and 3.2), allowing us to align token-to-token matching before GSU. We construct it by performing low-rank and positional alignment of the teacher's query and key projections, then folding them into read and write projections.

Low-rank and positional alignment. The circular construction in Sec. 3.1 uses U two-dimensional factors. Each two-dimensional factor can serve as a RoPE rotation plane, with rotations at frequencies $\omega_u=2\pi/d_p^{u+1}$ for $u=0,\dots,U-1$ approximately corresponding to CAPE's cyclic address shifts (Appendix A.6). Since the address count d_p^U grows exponentially with U, we compress each teacher head's query and key vectors from their original dimension d_k to $2U$, motivated by evidence that these representations admit low-rank approximations (Saxena et al., 2024). We introduce adapters $\mathbf{P}_Q,\mathbf{P}_K\in\mathbb{R}^{2U\times d_k}$ to implement this reduction, aiming to preserve position-dependent token matching. For each head, we consider both RoPE with the target frequencies ω_u and no positional encoding (NoPE). Using the Fourier representation of rotary matching (Hua et al., 2025), we obtain closed-form adapter initializations from calibration statistics through frequency projection and low-rank factorization (Appendix A.7). After refining the RoPE and NoPE candidates against the teacher's attention maps using forward Kullback-Leibler divergence (KL), we retain RAM-Net's original CAPE pattern across layers and assign heads to CAPE or NoPE based on their relative alignment errors.

Circular soft quantization. After selecting the adapters and positional modes, we map the aligned query and key representations to soft circular address assignments. Let $\widetilde{\mathbf{C}}_U = \mathrm{Diag}(\mathbf{C}, \ldots, \mathbf{C})$ denote the block-diagonal expansion of the circular codebook with U copies. Given the teacher projections $\mathbf{W}_Q^t$ and $\mathbf{W}_K^t$, we combine low-rank alignment, the circular construction, and temperature matching to obtain the student projections $\mathbf{W}_r^s = \beta^\star \widetilde{\mathbf{C}}_U^\top \mathbf{P}_Q \mathbf{W}_Q^t$ and $\mathbf{W}_w^s = \beta^\star \widetilde{\mathbf{C}}_U^\top \mathbf{P}_K \mathbf{W}_K^t$, where $\beta^\star$ is the assignment concentration defined in Sec. 3.2 with $\tau = \sqrt{2U}$. Product Softmax then yields address assignments whose overlap defines the intermediate attention similarity.

4.2 RAM-Net Conversion and Progressive Recovery

Recovery proceeds at the head, attention-block, and full-model levels. We convert the intermediate model in Sec. 4.1 to RAM-Net and optimize its RAM-Net-specific parameters to compensate for output differences introduced by Top-K slot selection and GSU. Notably, we optimize the read and write projection matrices $\mathbf{W}_r^s$ and $\mathbf{W}_w^s$ without the low-rank and shared-codebook constraints used in Sec. 4.1. For **head-level recovery**, we adapt address routing and GSU updates to match the teacher's per-head outputs. For **attention-block recovery**, we train the attention branch to match the teacher's residual updates. Both stages use normalized mean squared error (NMSE), computed on head outputs before output projection and on attention-branch outputs before residual addition, respectively. For **full-model recovery**, we unfreeze and fine-tune all student parameters using next-token cross-entropy as the main objective. Appendix B.2 provides the complete training details.

5 Experiments

5.1 Experimental Setup

We evaluate Transformer-to-RAM-Net weight migration on pretrained models ranging from 340M to 7B parameters. We use nine publicly available pretrained teacher models, whose architectures are summarized in Table 5. Each RAM-Net matches its teacher's architecture and tokenizer, except for the attention mechanism. Unless otherwise specified, we use $d_p = 4$, $U = 5$, and Top-$K = 32$. For GQA (Ainslie et al., 2023), each KV head is repeated over its query-head group to form the equivalent multi-head attention (MHA) weights used for migration. All migration experiments use the same total budget of 500M tokens, with per-stage token allocations, learning rates, and other training details provided in Table 6 and Appendix B.2. The tokens come from the 10BT sample of the FineWeb-Edu dataset HuggingFaceFW (2024). Full-model recovery uses two NVIDIA H200 GPUs with FSDP2, while all other stages use a single GPU.

5.2 Language Model Evaluation Benchmarks

Table 1 reports language modeling performance and accuracy on six common-knowledge tasks. Across the nine teachers, the Wiki PPL gap between RAM-Net and its teacher ranges from 1.22 to 5.09, while the average-accuracy gap across the six tasks ranges from 1.8 to 4.5 percentage points. We use the relative score $R = \frac{s-r}{t-r}$ to avoid overstating retention when both models perform near chance; s, t, and r denote RAM-Net, teacher, and chance accuracies. Across teachers, R ranges from 79.2% to 95.0%, with a mean of 87.1%. These results show that the migration procedure retains most of the teachers' performance on these benchmarks across attention types, model families, and scales. We also report the mass-aware readout variant (Sec. 3.3) in Appendix B.3, which shows similar overall trends. Table 3 and Table 10 in the appendix further show that RAM-Net partially recovers teacher performance on code completion and summarization tasks from LongBench.

5.3 Ablation Studies

Low-rank alignment and circular codebook. We ablate the low-rank alignment and circular codebook in Sec. 4.1, measuring the Jensen–Shannon divergence (JS) between teacher and RAM-Net attention distributions after intermediate attention construction and before head-level recovery. The *r/w projection* and *kq projection adapter* omit learned low-rank alignment, learning projections from scratch and transferring teacher QK weights through random adapters, respectively. The codebook baselines retain learned low-rank alignment to isolate the effects of codebook design. The *random*

Table 1: Zero-shot language modeling and commonsense reasoning performance of Teacher-to-RAM-Net migration.

Family	Teacher	Model	Tokens	Ctx.	Wiki PPL ↓	ARC-E ↑	ARC-C ↑	PIQA ↑	SciQ ↑	COPA ↑	Hella. ↑	Avg. ↑	Avg. R ↑
FLA	`transformer-340M-10B`	Teacher	10B	4K	30.05	57.4	24.1	65.6	82.5	73.0	32.1	55.8	86.5
		RAM-Net	0.5B	2K	35.14	54.7	23.7	64.7	80.2	65.0	31.0	53.2	
	`transformer-1.3B-100B`	Teacher	100B	2K	17.66	56.1	23.9	70.1	85.3	75.0	38.5	58.1	86.4
		RAM-Net	0.5B	2K	21.93	52.7	24.8	68.3	78.8	71.0	35.6	55.2	
	`transformer-2.7B-100B`	Teacher	100B	2K	15.23	59.4	26.5	71.9	87.9	75.0	42.0	60.4	89.2
		RAM-Net	0.5B	2K	19.82	56.5	24.6	70.3	79.9	74.0	38.3	57.3	
Qwen2.5	`0.5B-Base`	Teacher	–	32K	18.46	64.4	29.3	70.6	93.1	74.0	40.7	62.0	79.2
		RAM-Net	0.5B	4K	23.35	63.6	26.4	68.5	81.9	73.0	36.9	58.4	
	`1.5B-Base`	Teacher	–	32K	12.60	75.2	41.1	75.6	94.2	83.0	50.2	69.9	95.0
		RAM-Net	0.5B	4K	16.33	75.3	41.5	74.3	89.8	82.0	45.6	68.1	
	`3B-Base`	Teacher	–	32K	12.41	77.6	45.0	78.6	96.1	86.0	55.0	73.0	91.5
		RAM-Net	0.5B	4K	13.63	77.2	43.7	77.0	92.9	79.0	50.7	70.1	
RedPajama	`INCITE-Base-3B-v1`	Teacher	800B	2K	11.76	68.0	31.7	74.8	90.4	77.0	47.6	64.9	79.7
		RAM-Net	0.5B	2K	16.65	63.6	27.7	72.2	80.6	77.0	41.9	60.5	
Mistral	`7B-v0.1`	Teacher	–	8K	8.22	80.3	50.9	80.9	96.4	92.0	61.5	77.0	89.4
		RAM-Net	0.5B	4K	10.14	77.2	44.1	79.7	91.5	87.0	58.0	72.9	
LLaMA	`2-7B-hf`	Teacher	2T	4K	8.76	75.3	42.7	77.7	94.8	88.0	57.1	72.6	87.2
		RAM-Net	0.5B	4K	11.84	71.6	39.0	76.2	89.2	81.0	51.6	68.1	

Avg. is the arithmetic mean of the six task accuracies. Relative score Avg. R gives $\mathrm{Avg.}((s-r)/(t-r))$ over tasks for which $t > r$ with 95% confidence.

codebook tests the circular construction against shared random codes. Inspired by FAVOR+ (Choromanski et al., 2021), the *orthogonal random codebook* replaces circular codes with randomly scaled two-dimensional directions obtained by QR decomposition of Gaussian matrices. We also compare learned codebooks (Alonso et al., 2026; van den Dool et al., 2026) with the fixed circular construction: *shared k-means* learns one shared d_p-centroid codebook from calibration using query and key across factors, while *per-factor PQ* learns one per factor (Jégou et al., 2011). Table 2 shows that the circular codebook yields lower JS than all alternatives across the nine teachers. In the appendix, Fig. 7 further shows that the circular codebook generally yields lower head NMSE during recovery; Fig. 6 and Table 11 evaluate the effects of varying d_p and U.

Code assignment concentration β. The concentration β controls how the address overlap approximates exponential dot-product similarity. We sweep $\beta/\beta^\star$, where $\beta^\star$ is derived in Sec. 3.2. For each teacher, all sweep configurations use the same low-rank adapters. We report normalized JS (Fig. 3) and normalized head NMSE (Fig. 5 in the appendix) before head-level recovery, each normalized by its value at $\beta^\star$ for the same teacher. Across teachers, $\beta^\star$ yields near-minimal JS and head NMSE, although some minima occur around $1.58\beta^\star$. For well-aligned query-key pairs, higher-order terms can reduce the overlap relative to the local approximation, so a slightly larger β may partly compensate by sharpening code assignments.

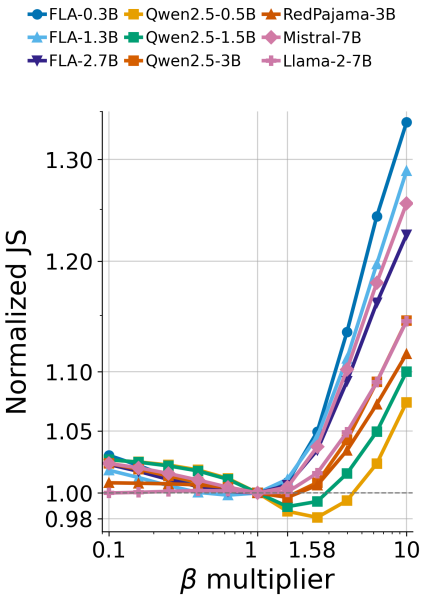

Figure 3: Attn. JS under different circular assignment concentrations β.

Migration pipeline ablation. We assess the value of staged recovery in Sec. 4 on Qwen2.5-1.5B under a fixed 500M-token budget (Table 4). The *complete pipeline* achieves lower Wiki PPL and higher downstream accuracy than *full-model only*, which allocates the entire budget to full-model recovery. Removing either *head-level recovery* or *attention-block recovery* lowers downstream accuracy, even when its tokens are reallocated to full-model recovery. To assess the preset CAPE pattern, *pattern-free CAPE selection* determines CAPE head counts solely from relative RoPE alignment errors. It slightly lowers logit KL and Wiki PPL but also lowers downstream accuracy relative to the preset pattern. Appendix Table 12 and Fig. 8 provide intermediate results and stage-wise errors, respectively.

Table 2: Attention JS divergence of intermediate attention construction.

Method	FLA-0.3B	FLA-1.3B	FLA-2.7B	Qwen-0.5B	Qwen-1.5B	Qwen-3B	Red.-3B	Mist.-7B	LLaMA-7B
r/w projection	0.561	0.563	0.568	0.625	0.617	0.601	0.611	0.619	0.629
QK projection adapter	0.570	0.558	0.571	0.627	0.629	0.623	0.601	0.615	0.615
Random codebook	0.441	0.440	0.437	0.486	0.476	0.498	0.443	0.430	0.389
Orth. Rand. codebook	0.487	0.515	0.525	0.574	0.566	0.560	0.563	0.568	0.586
Shared k-means codebook	0.415	0.412	0.419	0.466	0.453	0.481	0.421	0.411	0.377
Per-factor PQ codebook	0.412	0.423	0.431	0.487	0.476	0.487	0.446	0.445	0.407
Circular codebook	**0.398**	**0.394**	**0.402**	**0.452**	**0.444**	**0.470**	**0.412**	**0.389**	**0.359**

Table 3: LongBench results.

Model	Method	LCC	MultiN.	SAMSum	Avg. ↑
FLA-1.3B	Teacher	48.3	6.6	6.2	20.3
	RAM-Net	36.4	7.2	9.3	17.6
Qwen-3B	Teacher	14.1	24.5	45.5	28.0
	RAM-Net	15.5	15.9	29.5	20.3
Mistral-7B	Teacher	70.0	15.5	28.2	37.9
	RAM-Net	33.4	12.1	22.1	22.5

Table 4: Pipeline ablations on Qwen2.5-1.5B.

Method	Logit KL ↓	Wiki PPL ↓	Avg. ↑	R ↑
Teacher	–	12.60	69.9	–
Complete	0.164	16.33	**68.1**	**95.0**
Pattern-free CAPE select.	**0.162**	**16.17**	67.6	93.7
w/o Head Rec.	0.166	16.31	66.9	90.8
w/o Attn.-Block Rec.	0.181	16.64	66.9	90.2
Full-Model only	0.197	16.82	66.6	89.7

6 Related Work

6.1 Linear-Time Attention via Quantization and Kernel Approximation

Vector-quantized attention reduces attention cost by replacing individual keys with a finite codebook. Transformer-VQ (Lingle, 2024) uses quantized keys with recurrent per-code statistics, whose connection to our framework through hard writes and soft reads is derived in Appendix A.5. AVQ-Attention (van den Dool et al., 2026) adaptively refines the codebook according to attention importance, while online VQ attention (Alonso et al., 2026) uses sparse online updates to maintain a larger fixed-size memory. Kernelized attention instead uses separable features to approximate softmax matching. Performer/FAVOR+ (Choromanski et al., 2021) approximates the exponential dot-product kernel with positive orthogonal random features. Our construction connects these two views by using discrete-code soft assignments whose separable read and write overlap locally approximates exponential dot-product similarity and supports recurrent aggregation.

6.2 Transformer-to-Linear Weight Migration

Prior work has migrated pretrained Transformers to various linear or recurrent architectures, including recurrent attention in SUPRA (Mercat et al., 2024), kernelized linear attention in Hedgehog (Zhang et al., 2024) and LoLCATs (Zhang et al., 2025), Mamba models in MOHAWK (Bick et al., 2024) and Llamba (Bick et al., 2025), RWKV variants in RADLADS (Goldstein et al., 2025), gated recurrent models in Liger (Lan et al., 2025), and linearization for Gated DeltaNet (Kuzina et al., 2026). These methods generally convert directly to the target architectures. Attention-to-Mamba (Moudgil et al., 2026) instead uses kernelized linear attention as an intermediate before conversion to Mamba. Unlike these Transformer-to-linear methods, our migration proposes a distinct path by introducing an intermediate construction derived from quantized Transformers.

7 Conclusion and Limitations

In this paper, we establish RAM-Net as a theoretical bridge between discrete-code quantization and linear attention, and use this connection to enable efficient Transformer-to-RAM-Net weight migration. One limitation is that our derivation is local and assumes a fixed circular codebook with projection blocks of rank at most two, while practical RAM-Net uses unrestricted projections. Our migration experiments are also conducted under a fixed token budget without further tuning. We leave these limitations for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and polishing, reviewing mathematical proofs, and limited code editing and refactoring. We did not use generative AI tools for synthetic data generation, translation, dataset cleaning or reformatting, qualitative or thematic data analysis, or interpretation of experimental results. We also used generative AI tools to discuss alternative formulations during research development. We developed the research ideas, theoretical models and claims, method and experimental design, and core code implementation ourselves. We reviewed and verified all AI-assisted text, mathematical feedback, and code. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, user studies, personal or sensitive data, or the collection or release of new datasets. All models and datasets used in our experiments are publicly available.

REPRODUCIBILITY STATEMENT

We provide complete derivations and proofs for the theoretical results in Appendix A. We report the experimental details required for reproducibility in Sec. 5.1 and Appendix B.2. We use only publicly available datasets and teacher models, as listed in Appendix B.1. We provide the code for reproducing the experiments at `https://anonymous.4open.science/r/Transformer2RAMNet-608D`.

REFERENCES

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 4895–4901, 2023.

Nick Alonso, Tomas Figliolia, and Beren Millidge. Online vector quantized attention. *arXiv preprint arXiv:2602.03922*, 2026. URL https://arxiv.org/abs/2602.03922.

Aviv Bick, Kevin Y. Li, Eric P. Xing, J. Zico Kolter, and Albert Gu. Transformers to SSMs: Distilling quadratic knowledge to subquadratic models. *arXiv preprint arXiv:2408.10189*, 2024. URL https://arxiv.org/abs/2408.10189.

Aviv Bick, Tobias Katsch, Nimit Sohoni, Arjun Desai, and Albert Gu. Llamba: Scaling distilled recurrent models for efficient language processing. *arXiv preprint arXiv:2502.14458*, 2025. URL https://arxiv.org/abs/2502.14458.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ua6zuk0WRH.

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

FLA Hub. FLA Hub Models. https://huggingface.co/fla-hub/models. Accessed: 2026-09-26.

Daniel Goldstein, Eric Alcaide, Janna Lu, and Eugene Cheah. RADLADS: Rapid attention distillation to linear attention decoders at scale. *arXiv preprint arXiv:2505.03005*, 2025. URL https://arxiv.org/abs/2505.03005.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Ermo Hua, Che Jiang, Xingtai Lv, Kaiyan Zhang, Youbang Sun, Yuchen Fan, Xuekai Zhu, Biqing Qi, Ning Ding, and Bowen Zhou. Fourier position embedding: Enhancing attention's periodic extension for length generalization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24932–24949. PMLR, 2025. URL https://proceedings.mlr.press/v267/hua25b.html.

HuggingFaceFW. fineweb-edu (revision 22b0aca), 2024. URL https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu.

Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011. doi: 10.1109/TPAMI.2010.57.

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

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

Anna Kuzina, Paul N. Whatmough, and Babak Ehteshami Bejnordi. The key to going linear: Analysis-driven transformer linearization. *arXiv preprint arXiv:2607.07706*, 2026. URL https://arxiv.org/abs/2607.07706.

Disen Lan, Weigao Sun, Jiaxi Hu, Jusen Du, and Yu Cheng. Liger: Linearizing large language models to gated recurrent structures. *arXiv preprint arXiv:2503.01496*, 2025. URL https://arxiv.org/abs/2503.01496.

Lucas D. Lingle. Transformer-VQ: Linear-time transformers via vector quantization. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=oDdzXQzP2F.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344, 2024. URL https://proceedings.mlr.press/v235/liu24bz.html.

Jean Mercat, Igor Vasiljevic, Sedrick Keh, Kushal Arora, Achal Dave, Adrien Gaidon, and Thomas Kollar. Linearizing large language models. *arXiv preprint arXiv:2405.06640*, 2024. URL https://arxiv.org/abs/2405.06640.

Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, Naveen Mellempudi, Stuart Oberman, Mohammad Shoeybi, Michael Siu, and Hao Wu. FP8 formats for deep learning. *arXiv preprint arXiv:2209.05433*, 2022. doi: 10.48550/arXiv.2209.05433. URL https://arxiv.org/abs/2209.05433.

L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960. doi: 10.1093/qmath/11.1.50.

Abhinav Moudgil, Ningyuan Huang, Eeshan Gunesh Dhekane, Pau Rodríguez, Luca Zappella, and Federico Danieli. Attention to mamba: A recipe for cross-architecture distillation. *arXiv preprint arXiv:2604.14191*, 2026.

R. Penrose. On best approximate solutions of linear matrix equations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 52(1):17–19, 1956. doi: 10.1017/S0305004100030929.

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

Semyon Savkin, Eitan Porat, Or Ordentlich, and Yury Polyanskiy. NestQuant: Nested lattice quantization for matrix products and LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53042–53062. PMLR, 2025. URL https://proceedings.mlr.press/v267/savkin25a.html.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. Eigen attention: Attention in low-rank space for KV cache compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 15332–15344. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.899.

Donghyun Son, Euntae Choi, and Sungjoo Yoo. NSNQuant: A double normalization approach for calibration-free low-bit vector quantization of KV cache. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1436. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/3d8ee933c215fbb7b4d1948ff4906299-Abstract-Conference.html.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. URL https://arxiv.org/abs/2104.09864.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (Learn at test time): RNNs with expressive hidden states. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57503–57522. PMLR, 2025. URL https://proceedings.mlr.press/v267/sun25h.html.

Together Computer. RedPajama-INCITE-Base-3B-v1. https://huggingface.co/togethercomputer/RedPajama-INCITE-Base-3B-v1, 2023. Hugging Face model repository.

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

Winfried van den Dool, Patrick Forré, Amir Habibian, Yuki M. Asano, and Max Welling. AVQ-attention: Adaptive vector-quantized attention. *arXiv preprint arXiv:2607.12789*, 2026. URL https://arxiv.org/abs/2607.12789.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Kaicheng Xiao, Haotian Li, Liran Dong, and Guoliang Xing. RAM-Net: Expressive linear attention with selectively addressable memory. *arXiv preprint arXiv:2602.11958*, 2026. URL https://arxiv.org/abs/2602.11958.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56501–56523, 2024. URL https://proceedings.mlr.press/v235/yang24ab.html.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=r8H7xhYPwz.

Michael Zhang, Kush Bhatia, Hermann Kumbong, and Christopher Ré. The hedgehog & the porcupine: Expressive linear attentions with softmax mimicry. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2402.04347.

Michael Zhang, Simran Arora, Rahul Chalamala, Alan Wu, Benjamin Spector, Aaryan Singhal, Krithik Ramesh, and Christopher Ré. LoLCATs: On low-rank linearizing of large language models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.10254.

A Additional Proofs and Analysis

A.1 Full Proof and Extensions of Theorem 1

We provide a complete and generalized proof of Theorem 1, allowing separate read and write assignment concentrations (i.e., scaled codeword norms) and deriving an explicit approximation error bound stated in Theorem 2.

Theorem 2 (Error bound with separate read and write concentrations and finite codebooks)**.** *Under the circular-codebook and factorized read and write setup of Theorem 1, let $\beta_r, \beta_w \geq 0$ denote the read and write codeword norms. With $\|\mathbf{c}_j\|_2 = 1$, the corresponding codewords are $\beta_r \mathbf{c}_j$ and $\beta_w \mathbf{c}_j$, respectively. Let $M = \max_{0 \leq u < U} \max\{\beta_r \|\mathbf{q}_t^{(u)}\|_2, \beta_w \|\mathbf{k}_s^{(u)}\|_2\}$ and $\xi_{d_p,M} = 2e^M M^{d_p}/d_p!$. The log-overlap satisfies*

$$\left| \log \kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s) - \frac{\beta_r \beta_w}{2} \mathbf{q}_t^\top \mathbf{k}_s \right| \leq U \left[\frac{M^4}{4} + \frac{3\xi_{d_p,M}}{1 - \xi_{d_p,M}} \right], \qquad \xi_{d_p,M} < 1. \tag{11}$$

Proof. We first derive the continuous-codebook limit and its dot-product approximation, and then bound the dot-product approximation error and finite-codebook error separately.

For the uniform circular codewords $\mathbf{c}_j = (\cos(2\pi j/d_p), \sin(2\pi j/d_p))^\top$, introduce the partition function

$$Z_{d_p}(\mathbf{x}) = \frac{1}{d_p} \sum_{j=0}^{d_p - 1} \exp(\mathbf{c}_j^\top \mathbf{x}).$$

Here the read and write scales are absorbed into the argument of Z_{d_p}. Since $\mathcal{A} = \{0, \ldots, d_p - 1\}^U$, the product structure of the read and write distributions allows the overlap to factorize over the U

factors:

$$\begin{aligned}
\kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s) &= |\mathcal{A}| \sum_{\mathbf{a}\in\mathcal{A}} r_t(\mathbf{a}) w_s(\mathbf{a}) \\
&= d_p^U \sum_{\mathbf{a}\in\mathcal{A}} \prod_{u=0}^{U-1} \frac{\exp(\beta_r \mathbf{c}_{a^{(u)}}^\top \mathbf{q}_t^{(u)})}{\sum_{j=0}^{d_p-1} \exp(\beta_r \mathbf{c}_j^\top \mathbf{q}_t^{(u)})} \cdot \prod_{u=0}^{U-1} \frac{\exp(\beta_w \mathbf{c}_{a^{(u)}}^\top \mathbf{k}_s^{(u)})}{\sum_{j=0}^{d_p-1} \exp(\beta_w \mathbf{c}_j^\top \mathbf{k}_s^{(u)})} \\
&= \prod_{u=0}^{U-1} \left[d_p \sum_{i=0}^{d_p-1} \frac{\exp(\beta_r \mathbf{c}_i^\top \mathbf{q}_t^{(u)})}{\sum_{j=0}^{d_p-1} \exp(\beta_r \mathbf{c}_j^\top \mathbf{q}_t^{(u)})} \frac{\exp(\beta_w \mathbf{c}_i^\top \mathbf{k}_s^{(u)})}{\sum_{j=0}^{d_p-1} \exp(\beta_w \mathbf{c}_j^\top \mathbf{k}_s^{(u)})} \right] \\
&= \prod_{u=0}^{U-1} \frac{d_p \sum_{j=0}^{d_p-1} \exp\Big(\mathbf{c}_j^\top (\beta_r \mathbf{q}_t^{(u)} + \beta_w \mathbf{k}_s^{(u)})\Big)}{d_p Z_{d_p}(\beta_r \mathbf{q}_t^{(u)}) \cdot d_p Z_{d_p}(\beta_w \mathbf{k}_s^{(u)})} \\
&= \prod_{u=0}^{U-1} \frac{Z_{d_p}(\beta_r \mathbf{q}_t^{(u)} + \beta_w \mathbf{k}_s^{(u)})}{Z_{d_p}(\beta_r \mathbf{q}_t^{(u)}) Z_{d_p}(\beta_w \mathbf{k}_s^{(u)})}.
\end{aligned} \tag{12}$$

Let $Z_\infty(\mathbf{x}) = \lim_{d_p\to\infty} Z_{d_p}(\mathbf{x})$ and $\kappa_\infty(\mathbf{q}_t, \mathbf{k}_s) = \lim_{d_p\to\infty} \kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s)$. The circular Riemann sum converges to an integral; substituting this limit into Eq. (12) gives

$$\begin{aligned}
Z_\infty(\mathbf{x}) &= \lim_{d_p\to\infty} \frac{1}{d_p} \sum_{j=0}^{d_p-1} \exp\Big(\|\mathbf{x}\|_2 \cos\Big(\tfrac{2\pi j}{d_p} - \arg \mathbf{x}\Big)\Big) \\
&= \frac{1}{2\pi} \int_0^{2\pi} \exp(\|\mathbf{x}\|_2 \cos\theta)\, d\theta \\
&= I_0(\|\mathbf{x}\|_2) \\
\kappa_\infty(\mathbf{q}_t, \mathbf{k}_s) &= \prod_{u=0}^{U-1} \frac{I_0\Big(\|\beta_r \mathbf{q}_t^{(u)} + \beta_w \mathbf{k}_s^{(u)}\|_2\Big)}{I_0\Big(\|\beta_r \mathbf{q}_t^{(u)}\|_2\Big)\, I_0\Big(\|\beta_w \mathbf{k}_s^{(u)}\|_2\Big)}.
\end{aligned} \tag{13}$$

Here I_0 is the modified Bessel function of the first kind of order zero.

To identify the leading term of the log-kernel, use $I_0(\rho) = 1 + \frac{\rho^2}{4} + O(\rho^4)$ and $\log(1+x) = x - \frac{x^2}{2} + O(x^3)$ as $x \to 0$:

$$\log I_0(\rho) = \Big(\tfrac{\rho^2}{4} + O(\rho^4)\Big) - \tfrac{1}{2}\Big(\tfrac{\rho^2}{4} + O(\rho^4)\Big)^2 + O(\rho^6) = \frac{\rho^2}{4} + O(\rho^4).$$

Substituting this expansion into the logarithm of Eq. (13) as $M \to 0$ and using the squared-norm identity $\|\beta_r \mathbf{q}_t^{(u)} + \beta_w \mathbf{k}_s^{(u)}\|_2^2 = \beta_r^2 \|\mathbf{q}_t^{(u)}\|_2^2 + 2\beta_r\beta_w {\mathbf{q}_t^{(u)}}^\top \mathbf{k}_s^{(u)} + \beta_w^2 \|\mathbf{k}_s^{(u)}\|_2^2$ yields

$$\begin{aligned}
\log \kappa_\infty(\mathbf{q}_t, \mathbf{k}_s) &= \sum_{u=0}^{U-1} \Big[\log I_0\Big(\|\beta_r \mathbf{q}_t^{(u)} + \beta_w \mathbf{k}_s^{(u)}\|_2\Big) \\
&\qquad - \log I_0\Big(\|\beta_r \mathbf{q}_t^{(u)}\|_2\Big) - \log I_0\Big(\|\beta_w \mathbf{k}_s^{(u)}\|_2\Big) \Big] \\
&= \frac{1}{4} \sum_{u=0}^{U-1} \Big[\|\beta_r \mathbf{q}_t^{(u)} + \beta_w \mathbf{k}_s^{(u)}\|_2^2 - \beta_r^2 \|\mathbf{q}_t^{(u)}\|_2^2 - \beta_w^2 \|\mathbf{k}_s^{(u)}\|_2^2 \Big] \\
&\qquad + O\left(\sum_{u=0}^{U-1} \big(\beta_r \|\mathbf{q}_t^{(u)}\|_2 + \beta_w \|\mathbf{k}_s^{(u)}\|_2\big)^4 \right) \\
&= \frac{\beta_r\beta_w}{2} \mathbf{q}_t^\top \mathbf{k}_s + O\Big(\big(\beta_r \|\mathbf{q}_t\|_2 + \beta_w \|\mathbf{k}_s\|_2\big)^4 \Big)
\end{aligned} \tag{14}$$

The last step uses a conservative remainder bound in terms of the norms of the full query and key vectors by the triangle inequality. Eq. (14) gives the continuous-codebook version of the dot-product

approximation in Theorem 1, generalized to separate read and write assignment concentrations β_r and β_w.

We now replace this asymptotic remainder with an explicit bound. With M as defined in Theorem 2, the triangle inequality gives $\|\beta_r \mathbf{q}_t^{(u)} + \beta_w \mathbf{k}_s^{(u)}\|_2 \le 2M$ for every u. It therefore suffices to control the scalar approximation error of $\log I_0$ on these argument ranges.

For $\rho \ge 0$, set $t = \rho^2/4$, giving

$$\left(1 + \frac{t}{2}\right)^2 = 1 + t + \frac{t^2}{4} \le I_0(\rho) = \sum_{m=0}^{\infty} \frac{t^m}{(m!)^2} \le \sum_{m=0}^{\infty} \frac{t^m}{m!} = e^t.$$

To obtain a lower bound for $\log I_0$, we use the following integral representation and the inequality $1 - s \le 1/(1+s)$ for $s \ge 0$:

$$(t/2) - \frac{(t/2)^2}{2} = \int_0^{\frac{t}{2}} (1 - s)\, ds \le \int_0^{\frac{t}{2}} \frac{1}{1+s}\, ds = \log(1 + \frac{t}{2}).$$

Taking logarithms of the power-series bounds and applying this inequality yields

$$t - \frac{t^2}{4} \le 2 \log\left(1 + \frac{t}{2}\right) \le \log I_0(\rho) \le t.$$

Rearranging isolates the quadratic approximation error $0 \le t - \log I_0(\rho) \le t^2/4$. Define this scalar remainder by

$$\eta(\rho) = \frac{\rho^2}{4} - \log I_0(\rho)$$

and substitute $t = \rho^2/4$ to obtain

$$0 \le \eta(\rho) \le \frac{\rho^4}{64}. \tag{15}$$

Substituting $\log I_0(\rho) = \frac{\rho^2}{4} - \eta(\rho)$ into the logarithm of Eq. (13) gives the exact remainder:

$$\log \kappa_\infty(\mathbf{q}_t, \mathbf{k}_s) - \frac{\beta_r \beta_w}{2} \mathbf{q}_t^\top \mathbf{k}_s = \sum_{u=0}^{U-1} \Big[\eta\Big(\beta_r \|\mathbf{q}_t^{(u)}\|_2\Big) + \eta\Big(\beta_w \|\mathbf{k}_s^{(u)}\|_2\Big) - \eta\Big(\|\beta_r \mathbf{q}_t^{(u)} + \beta_w \mathbf{k}_s^{(u)}\|_2\Big) \Big].$$

By Eq. (15), the first two terms in each summand lie in $[0, M^4/64]$, while the subtracted term lies in $[0, (2M)^4/64]$. Summing these bounds gives

$$-\frac{UM^4}{4} \le \log \kappa_\infty(\mathbf{q}_t, \mathbf{k}_s) - \frac{\beta_r \beta_w}{2} \mathbf{q}_t^\top \mathbf{k}_s \le \frac{UM^4}{32}. \tag{16}$$

Thus the continuous-kernel approximation error satisfies

$$\left| \log \kappa_\infty(\mathbf{q}_t, \mathbf{k}_s) - \frac{\beta_r \beta_w}{2} \mathbf{q}_t^\top \mathbf{k}_s \right| \le \frac{UM^4}{4}. \tag{17}$$

Now consider the error between the finite and continuous log-kernels, $|\log \kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s) - \log \kappa_\infty(\mathbf{q}_t, \mathbf{k}_s)|$. Set $\rho = \|\mathbf{x}\|_2$, and let I_n denote the modified Bessel function of the first kind of order n. Applying the Fourier-Bessel expansion $\exp(\rho \cos\phi) = I_0(\rho) + 2\sum_{n=1}^{\infty} I_n(\rho)\cos(n\phi)$ to $Z_{d_p}(\mathbf{x})$ shows that only frequencies divisible by d_p survive the discrete average:

$$\begin{aligned} Z_{d_p}(\mathbf{x}) &= I_0(\rho) + 2\sum_{n=1}^{\infty} I_n(\rho) \left[\frac{1}{d_p} \sum_{j=0}^{d_p - 1} \cos\Big(n \left(\tfrac{2\pi j}{d_p} - \arg \mathbf{x}\right)\Big) \right] \\ &= Z_\infty(\mathbf{x}) + 2\sum_{\ell=1}^{\infty} I_{\ell d_p}(\rho) \cos(\ell d_p \arg \mathbf{x}). \end{aligned}$$

These additional modes account for the finite-codebook error. Dividing by $Z_\infty(\mathbf{x}) = I_0(\rho) > 0$ and using $|\cos\theta| \le 1$ gives

$$\left|\frac{Z_{d_p}(\mathbf{x})}{Z_\infty(\mathbf{x})} - 1\right| \le \frac{2\sum_{\ell=1}^{\infty} I_{\ell d_p}(\rho)}{I_0(\rho)}.$$

To bound this tail explicitly, compare the power series of I_n and use $(m+n)! \ge m!n!$:

$$I_n(\rho) = \left(\frac{\rho}{2}\right)^n \sum_{m=0}^{\infty} \frac{(\rho^2/4)^m}{m!(m+n)!} \le \frac{(\rho/2)^n}{n!} \sum_{m=0}^{\infty} \frac{(\rho^2/4)^m}{(m!)^2} = \frac{(\rho/2)^n}{n!} I_0(\rho).$$

Apply the resulting bound $I_n(\rho)/I_0(\rho) \le (\rho/2)^n/n!$, and use $(d_p+m)! \ge d_p!m!$:

$$\begin{aligned}\frac{2\sum_{\ell=1}^{\infty} I_{\ell d_p}(\rho)}{I_0(\rho)} &\le 2\sum_{\ell=1}^{\infty} \frac{(\rho/2)^{\ell d_p}}{(\ell d_p)!} \le 2\sum_{n=d_p}^{\infty} \frac{(\rho/2)^n}{n!} = 2\left(\frac{\rho}{2}\right)^{d_p} \sum_{m=0}^{\infty} \frac{(\rho/2)^m}{(d_p+m)!} \\ &\le \frac{2(\rho/2)^{d_p}}{d_p!} \sum_{m=0}^{\infty} \frac{(\rho/2)^m}{m!} = \frac{2e^{\rho/2}(\rho/2)^{d_p}}{d_p!}.\end{aligned}$$

Since every argument norm in the kernel factors is at most $2M$, combining the preceding bounds gives, uniformly for $\|\mathbf{x}\|_2 \le 2M$,

$$\frac{2\sum_{\ell=1}^{\infty} I_{\ell d_p}(\rho)}{I_0(\rho)} \le \xi_{d_p,M}, \qquad \xi_{d_p,M} = \frac{2e^M M^{d_p}}{d_p!}. \tag{18}$$

To compare the log-kernels, we convert this relative error bound into a logarithmic one. Assume $\xi_{d_p,M} < 1$ and $\|\mathbf{x}\|_2 \le 2M$, and set $s = Z_{d_p}(\mathbf{x})/I_0(\|\mathbf{x}\|_2) - 1$. Eq. (18) gives $|s| \le \xi_{d_p,M}$, and every v between 0 and s satisfies $1 + v \ge 1 - \xi_{d_p,M} > 0$. Consequently,

$$|\log(1+s)| = \left|\int_0^s \frac{1}{1+v}\, dv\right| \le \left|\int_0^s \frac{1}{1-\xi_{d_p,M}}\, dv\right| \le \frac{\xi_{d_p,M}}{1-\xi_{d_p,M}}.$$

Substituting the definition of s yields, uniformly for $\|\mathbf{x}\|_2 \le 2M$,

$$\left|\log \frac{Z_{d_p}(\mathbf{x})}{Z_\infty(\mathbf{x})}\right| \le \frac{\xi_{d_p,M}}{1-\xi_{d_p,M}}. \tag{19}$$

By Eqs. (12) and (13), the logarithm of each factor of $\kappa_{\mathcal{A}}/\kappa_\infty$ contains one numerator correction and two denominator corrections, all with argument norms at most $2M$. Applying Eq. (19) to these $3U$ terms and using the triangle inequality gives

$$\left|\log \frac{\kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s)}{\kappa_\infty(\mathbf{q}_t, \mathbf{k}_s)}\right| \le \frac{3U\xi_{d_p,M}}{1-\xi_{d_p,M}}. \tag{20}$$

Combining the finite-codebook bound in Eq. (20) with the continuous-kernel approximation bound in Eq. (17) proves Eq. (11):

$$\begin{aligned}&\left|\log \kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s) - \frac{\beta_r\beta_w}{2}\mathbf{q}_t^\top \mathbf{k}_s\right| \\ \le& \left|\log \kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s) - \log \kappa_\infty(\mathbf{q}_t, \mathbf{k}_s)\right| + \left|\log \kappa_\infty(\mathbf{q}_t, \mathbf{k}_s) - \frac{\beta_r\beta_w}{2}\mathbf{q}_t^\top \mathbf{k}_s\right| \\ \le& U\left[\frac{M^4}{4} + \frac{3\xi_{d_p,M}}{1-\xi_{d_p,M}}\right]\end{aligned}$$

For fixed d_p, $\xi_{d_p,M} = O(M^{d_p})$ as $M \to 0$, so $\xi_{d_p,M} < 1$ holds for sufficiently small M. The total error is therefore $O\big(U(M^4 + M^{d_p})\big)$, and hence $O(UM^4)$ whenever $d_p \ge 4$. For fixed U, setting $\beta_r = \beta_w = \beta$ and exponentiating gives

$$\kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s) = \exp\left(\frac{\beta^2}{2}\mathbf{q}_t^\top \mathbf{k}_s\right)\left[1 + O(UM^4)\right], \qquad M \to 0,$$

which recovers Theorem 1. □

A.2 From Softmax Attention to Mass-Aware Recurrent Readout

We approximate causal softmax attention by a mass-aware readout over address states. When the local regime holds uniformly over $s \le t$, the first step follows from Theorem 2; the remaining equalities are exact. Canceling $|\mathcal{A}|$ and exchanging the sums over tokens and addresses yields a readout whose address weights are proportional to the read probability multiplied by the accumulated write mass:

$$\begin{aligned}
\frac{\sum_{s\le t}\exp\left(\frac{\beta_r\beta_w}{2}\mathbf{q}_t^\top\mathbf{k}_s\right)\mathbf{v}_s}{\sum_{s\le t}\exp\left(\frac{\beta_r\beta_w}{2}\mathbf{q}_t^\top\mathbf{k}_s\right)} &\approx \frac{\sum_{s\le t}\kappa_{\mathcal{A}}(\mathbf{q}_t,\mathbf{k}_s)\mathbf{v}_s}{\sum_{s\le t}\kappa_{\mathcal{A}}(\mathbf{q}_t,\mathbf{k}_s)} \\
&= \frac{|\mathcal{A}|\sum_{s\le t}\sum_{\mathbf{a}\in\mathcal{A}} r_t(\mathbf{a})w_s(\mathbf{a})\mathbf{v}_s}{|\mathcal{A}|\sum_{s\le t}\sum_{\mathbf{b}\in\mathcal{A}} r_t(\mathbf{b})w_s(\mathbf{b})} \\
&= \frac{\sum_{\mathbf{a}\in\mathcal{A}} r_t(\mathbf{a})\sum_{s\le t} w_s(\mathbf{a})\mathbf{v}_s}{\sum_{\mathbf{b}\in\mathcal{A}} r_t(\mathbf{b})\sum_{s\le t} w_s(\mathbf{b})} \\
&= \sum_{\mathbf{a}\in\mathcal{A}} \frac{r_t(\mathbf{a})m_t(\mathbf{a})}{\sum_{\mathbf{b}\in\mathcal{A}} r_t(\mathbf{b})m_t(\mathbf{b})}\frac{\sum_{s\le t} w_s(\mathbf{a})\mathbf{v}_s}{m_t(\mathbf{a})} \\
&= \sum_{\mathbf{a}\in\mathcal{A}} \frac{r_t(\mathbf{a})m_t(\mathbf{a})}{\sum_{\mathbf{b}\in\mathcal{A}} r_t(\mathbf{b})m_t(\mathbf{b})}\mathbf{s}_t(\mathbf{a}).
\end{aligned} \tag{21}$$

Here $m_t(\mathbf{a})$ is the cumulative write mass, and $\mathbf{s}_t(\mathbf{a})$ is the write-weighted mean of values at address $\mathbf{a}$:

$$m_t(\mathbf{a}) = \sum_{s\le t} w_s(\mathbf{a}), \qquad \mathbf{s}_t(\mathbf{a}) = \frac{\sum_{s\le t} w_s(\mathbf{a})\mathbf{v}_s}{m_t(\mathbf{a})}.$$

Appendix A.3 shows how GSU recovers these cumulative statistics in its cumulative-averaging limit.

A.3 GSU Dynamics and the Cumulative-Averaging Limit

We show that GSU recovers the cumulative write mass and weighted value mean derived in Appendix A.2. For fixed soft write weights $0 < w_t(\mathbf{a}) < 1$, taking $\gamma_t \to -\infty$ at every step makes the retention gate tend to one:

$$\sigma(\mathrm{logit}(1 - w_t(\mathbf{a})) - \gamma_t) \to 1$$

In this limit, the mass update retains all previous mass and adds the current write weight. Starting from $m_0(\mathbf{a}) = 0$ and $\mathbf{s}_0(\mathbf{a}) = \mathbf{0}$, the accumulated mass is therefore

$$m_t(\mathbf{a}) = m_{t-1}(\mathbf{a}) + w_t(\mathbf{a}) = \sum_{s\le t} w_s(\mathbf{a}).$$

We set $\epsilon = 0$ in the state update. Substituting the mass recurrence then shows by induction that the state is the cumulative write-weighted mean. The fourth equality uses the induction hypothesis; the base case follows from the zero initialization:

$$\begin{aligned}
\mathbf{s}_t(\mathbf{a}) &= \mathbf{s}_{t-1}(\mathbf{a}) + \frac{w_t(\mathbf{a})}{m_t(\mathbf{a})}\left(\mathbf{v}_t - \mathbf{s}_{t-1}(\mathbf{a})\right) \\
&= \frac{\left(m_t(\mathbf{a}) - w_t(\mathbf{a})\right)\mathbf{s}_{t-1}(\mathbf{a}) + w_t(\mathbf{a})\mathbf{v}_t}{m_t(\mathbf{a})} \\
&= \frac{m_{t-1}(\mathbf{a})\mathbf{s}_{t-1}(\mathbf{a}) + w_t(\mathbf{a})\mathbf{v}_t}{m_t(\mathbf{a})} \\
&= \frac{\sum_{s<t} w_s(\mathbf{a})\mathbf{v}_s + w_t(\mathbf{a})\mathbf{v}_t}{m_t(\mathbf{a})} \\
&= \frac{\sum_{s\le t} w_s(\mathbf{a})\mathbf{v}_s}{\sum_{s\le t} w_s(\mathbf{a})}.
\end{aligned}$$

Thus GSU recovers cumulative weighted averaging in this limit. Combined with the mass-aware readout in Eq. (21), it exactly reproduces the normalized address-kernel attention.

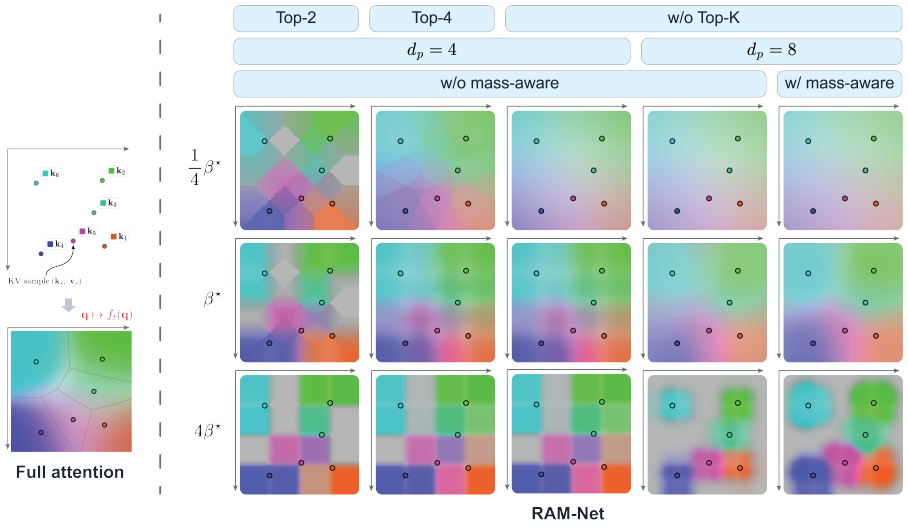

Figure 4: Value fields under different RAM-Net configurations. Keys and queries lie on $S^1 \times S^1 \subset \mathbb{R}^4$, with the plotted axes representing the angular coordinates of the two factors. Marker colors encode the stored values, while the background shows the predicted value $f_t(\mathbf{q})$. Full attention provides the reference on the left. Rows vary the codeword norm around the reference setting $\beta^\star$, and columns vary Top-K, d_p, and mass-aware readout.

A.4 Effects of Codebook Size, Sparsity, and Readout

Building on the attention approximation in Appendix A.1, we examine how codebook resolution, sparse address selection, and readout weighting affect the resulting value field. Fig. 4 compares these configurations with full attention, using a shared read and write codeword norm $\beta_r = \beta_w = \beta$.

The codeword norm β controls how strongly reads and writes concentrate on nearby addresses. Top-K truncation introduces larger artifacts at small β, where diffuse address weights leave more probability mass outside the selected slots. Increasing d_p refines the angular grid and expands the address space to d_p^U slots. At moderate β, the finer codebook reduces coarse grid artifacts and produces smoother transitions that more closely resemble the full-attention reference.

Standard RAM-Net uses $\sum_{\mathbf{a}} r_t(\mathbf{a})\mathbf{s}_t(\mathbf{a})$, whereas the mass-aware readout in Eq. (21) replaces the coefficients with $r_t(\mathbf{a})m_t(\mathbf{a})/\sum_{\mathbf{b}} r_t(\mathbf{b})m_t(\mathbf{b})$. The two agree when the masses are equal and positive on the read support and $\sum_{\mathbf{a}} r_t(\mathbf{a}) = 1$. Otherwise, mass weighting changes the relative influence of slots and shifts the boundaries between value regions. When these masses differ, mass weighting changes the relative influence of slots and can shift the boundaries between value regions. Because the address masses depend jointly on the codebook factors, this weighting can introduce oblique transitions, as illustrated by the last two columns.

Although mass-aware readout recovers normalized address-kernel attention in the cumulative-averaging limit, its practical benefit may be limited by interactions with forgetting and sparse selection. Appendix B.3 compares the two readouts empirically. First, forgetting can reduce a slot's mass relative to other slots, lowering its mass-aware readout weight even when the read scores remain unchanged. This couples forgetting to both the state dynamics and the readout, potentially suppressing slots that remain relevant to the query. Second, the efficient Top-K decoder selects addresses using the factorized score, whereas the history-dependent mass generally breaks this factorization in $r_t(\mathbf{a})m_t(\mathbf{a})$. Reweighting only the selected addresses can therefore miss slots that would rank highly under the mass-aware score.

A.5 Hard-Write Limit and Vector-Quantized Attention

We derive the hard-write limit from the exact address overlap in Eq. (12). We keep the read codeword norm β_r fixed and finite while taking the write codeword norm $\beta_w \to \infty$, yielding hard write assignments with soft reads.

For each key block, assume a unique maximizing code and define

$$j_s^{(u)} = \arg\max_{0 \le j < d_p} \mathbf{c}_j^\top \mathbf{k}_s^{(u)} = \arg\min_{0 \le j < d_p} \|\mathbf{k}_s^{(u)} - \mathbf{c}_j\|_2^2.$$

The second equality follows from $\|\mathbf{c}_j\|_2 = 1$, so the assignment is nearest-code vector quantization. For nonzero key blocks with continuously distributed angles, ties occur with probability zero, so the maximizing code is unique with probability one.

Dividing the numerator and denominator of each factor-wise write probability by the largest exponential makes every nonmaximizing term vanish as $\beta_w \to \infty$. Consequently,

$$\frac{\exp(\beta_w \mathbf{c}_j^\top \mathbf{k}_s^{(u)})}{\sum_{i=0}^{d_p-1} \exp(\beta_w \mathbf{c}_i^\top \mathbf{k}_s^{(u)})} \longrightarrow \mathbf{1}_{\{j = j_s^{(u)}\}}.$$

Let $\mathbf{a}_s = (j_s^{(0)}, \ldots, j_s^{(U-1)})$ be the selected product address; then $w_s(\mathbf{a}) \longrightarrow \mathbf{1}_{\{\mathbf{a} = \mathbf{a}_s\}}$. Define the reconstructed key by $\widehat{\mathbf{k}}_s = (\mathbf{c}_{j_s^{(0)}}^\top, \ldots, \mathbf{c}_{j_s^{(U-1)}}^\top)^\top$. The hard write selects a single term in the overlap, giving

$$\lim_{\beta_w \to \infty} \kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s) = d_p^U \prod_{u=0}^{U-1} \frac{\exp(\beta_r \mathbf{c}_{j_s^{(u)}}^\top \mathbf{q}_t^{(u)})}{\sum_{j=0}^{d_p-1} \exp(\beta_r \mathbf{c}_j^\top \mathbf{q}_t^{(u)})} = \frac{\exp(\beta_r \mathbf{q}_t^\top \widehat{\mathbf{k}}_s)}{\prod_{u=0}^{U-1} Z_{d_p}(\beta_r \mathbf{q}_t^{(u)})}. \tag{22}$$

The denominator depends only on the query and therefore cancels when the attention weights are normalized over tokens:

$$\lim_{\beta_w \to \infty} \frac{\sum_{s \le t} \kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s) \mathbf{v}_s}{\sum_{s \le t} \kappa_{\mathcal{A}}(\mathbf{q}_t, \mathbf{k}_s)} = \frac{\sum_{s \le t} \exp(\beta_r \mathbf{q}_t^\top \widehat{\mathbf{k}}_s) \mathbf{v}_s}{\sum_{s \le t} \exp(\beta_r \mathbf{q}_t^\top \widehat{\mathbf{k}}_s)}. \tag{23}$$

Eq. (23) gives softmax attention over product-quantized keys (Jégou et al., 2011), with each of the U blocks quantized using a fixed circular codebook of d_p codewords. Taking this hard-write limit after the cumulative-averaging limit in Appendix A.3 makes each occupied slot's mass and state the count and mean of its assigned values. With the mass-aware readout in Eq. (21), this recovers the per-code aggregation underlying Transformer-VQ (Lingle, 2024).

A.6 Correspondence between RoPE and CAPE

Consider the full soft address distributions of the circular construction in Sec. 3.1 before Top-K truncation. We flatten each address $\mathbf{a} \in \mathcal{A}$ into a slot index $i \in \{0, \ldots, d_p^U - 1\}$, with $a^{(0)}$ as the least significant digit:

$$i(\mathbf{a}) = \sum_{u=0}^{U-1} a^{(u)} d_p^u, \qquad a^{(u)} = \left\lfloor \frac{i}{d_p^u} \right\rfloor \bmod d_p, \qquad u = 0, \ldots, U-1. \tag{24}$$

For each two-dimensional input $\mathbf{x}$, the circular assignment has the following log probability, using the partition function Z_{d_p} defined in Appendix A.1:

$$\begin{aligned} \log p_\beta(j \mid \mathbf{x}) &= \beta \mathbf{c}_j^\top \mathbf{x} - \log\big(d_p Z_{d_p}(\beta \mathbf{x})\big) \\ &= \beta \|\mathbf{x}\|_2 \cos\left(\frac{2\pi}{d_p} j - \arg \mathbf{x}\right) - \log\big(d_p Z_{d_p}(\beta \mathbf{x})\big). \end{aligned}$$

For a vector $\mathbf{x} = (\mathbf{x}^{(0)}, \ldots, \mathbf{x}^{(U-1)})$ with two-dimensional blocks, we remove the normalization term, which is independent of i, to define the factor logit waveform:

$$\Phi_{\mathbf{x}}^{(u)}(i) = \log p_\beta(a^{(u)} \mid \mathbf{x}^{(u)}) + \log\Big(d_p Z_{d_p}(\beta \mathbf{x}^{(u)})\Big) = \beta \|\mathbf{x}^{(u)}\|_2 \cos\left(\frac{2\pi}{d_p} a^{(u)} - \arg \mathbf{x}^{(u)}\right) \tag{25}$$

The read and write distributions are obtained by superposing their respective factor logit waves and applying a softmax over addresses:

$$r_t(\mathbf{a}) \propto \exp\left(\sum_{u=0}^{U-1} \Phi^{(u)}_{\mathbf{q}_t}(i(\mathbf{a}))\right) \qquad w_s(\mathbf{a}) \propto \exp\left(\sum_{u=0}^{U-1} \Phi^{(u)}_{\mathbf{k}_s}(i(\mathbf{a}))\right)$$

By Eq. (24), each factor waveform has period d_p^{u+1} and is constant over blocks of d_p^u slots. It can be obtained by sampling and holding the following continuous wave, where $s \in \mathbb{R}$ denotes a continuous slot coordinate:

$$\widetilde{\Phi}^{(u)}_{\mathbf{x}}(s) = \beta\|\mathbf{x}^{(u)}\|_2 \cos\left(\frac{2\pi}{d_p^{u+1}} s - \arg \mathbf{x}^{(u)}\right). \tag{26}$$

Evaluating this continuous wave at the left endpoint of each block of d_p^u slots exactly recovers the discrete waveform in Eq. (25):

$$\Phi^{(u)}_{\mathbf{x}}(i) = \widetilde{\Phi}^{(u)}_{\mathbf{x}}\left(d_p^u \left\lfloor \frac{i}{d_p^u} \right\rfloor\right), \qquad u = 0, \ldots, U-1. \tag{27}$$

RoPE preserves each block's norm and rotates it by $\nu^{(u)}t$, where $\nu^{(u)}$ is the angular frequency of factor u. Substituting this change into Eq. (26) gives

$$\begin{aligned}
&\beta\|\mathbf{x}^{(u)}\|_2 \cos\left(\frac{2\pi}{d_p^{u+1}} s - \arg \mathbf{x}^{(u)} - \nu^{(u)} t\right) \\
&= \beta\|\mathbf{x}^{(u)}\|_2 \cos\left[\frac{2\pi}{d_p^{u+1}}\left(s - \frac{d_p^{u+1}\nu^{(u)}}{2\pi} t\right) - \arg \mathbf{x}^{(u)}\right] \\
&= \widetilde{\Phi}^{(u)}_{\mathbf{x}}\left(s - \frac{d_p^{u+1}\nu^{(u)}}{2\pi} t\right).
\end{aligned}$$

This corresponds to translation at speed $d_p^{u+1}\nu^{(u)}/(2\pi)$ slots per token. Under the cyclic-shift convention $f(i) \mapsto f((i-t) \bmod d_p^U)$, setting the translation speed to one slot per token gives

$$\nu^{(u)} = \frac{2\pi}{d_p^{u+1}}, \qquad u = 0, \ldots, U-1. \tag{28}$$

With Eq. (28), every continuous factor wave becomes $\widetilde{\Phi}^{(u)}_{\mathbf{x}}(s-t)$, so their superposition also translates by t slots. Since softmax commutes with cyclic slot permutations, CAPE equivalently shifts the sampled-and-held logits, whereas RoPE translates the continuous waves before sampling and holding. By Eq. (27), the two operations give the following factor logits at slot i:

$$\begin{aligned}
\text{CAPE:} \quad & \widetilde{\Phi}^{(u)}_{\mathbf{x}}\left(d_p^u \left\lfloor \frac{i-t}{d_p^u} \right\rfloor\right), \\
\text{RoPE:} \quad & \widetilde{\Phi}^{(u)}_{\mathbf{x}}\left(d_p^u \left\lfloor \frac{i}{d_p^u} \right\rfloor - t\right).
\end{aligned}$$

This difference in the order of operations introduces a discretization error into the resulting address distributions.

A.7 Low-rank Adapter Construction

We derive the initialization in Sec. 4.1 by matching the query-key interactions induced by the teacher projections. Consider one attention head and omit layer and head indices. Under the column-vector convention used in the main text, the teacher projections are $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d_k \times d_{\text{model}}}$, and the target projections are $\widetilde{\mathbf{W}}_Q, \widetilde{\mathbf{W}}_K \in \mathbb{R}^{2U \times d_{\text{model}}}$. Let $\Delta = i - j$ be the key position i minus the query position j, and let $\mathbf{R}_T(\Delta)$ and $\mathbf{R}_S(\Delta)$ denote the relative teacher and target rotations. The unscaled teacher logit is $\mathbf{h}_j^\top \mathbf{B}_T(\Delta)\mathbf{h}_i$, where

$$\mathbf{B}_T(\Delta) = \mathbf{W}_Q^\top \mathbf{R}_T(\Delta)\mathbf{W}_K, \qquad \mathbf{B}_S(\Delta) = \widetilde{\mathbf{W}}_Q^\top \mathbf{R}_S(\Delta)\widetilde{\mathbf{W}}_K. \tag{29}$$

These bilinear operators describe the interactions that determine attention logits. Matching them therefore targets the joint behavior of the query and key projections rather than approximating each projection independently.

Let $p(\Delta)$ be the empirical relative-position distribution weighted by teacher attention. Define the hidden-state second moments $\mathbf{\Sigma}_Q = \mathbb{E}_j[\mathbf{h}_j\mathbf{h}_j^\top]$ and $\mathbf{\Sigma}_K = \mathbb{E}_i[\mathbf{h}_i\mathbf{h}_i^\top]$ to weight operator errors. Here, $\mathbf{\Sigma}_Q$ and $\mathbf{\Sigma}_K$ reflect the different positional sampling distributions of queries and keys under causal attention. Approximating squared logit error with factorized query and key second moments that do not depend on Δ gives the objective

$$\mathcal{L} = \mathbb{E}_{\Delta\sim p}\left\|\mathbf{\Sigma}_Q^{1/2}\big[\mathbf{B}_T(\Delta) - \mathbf{B}_S(\Delta)\big]\mathbf{\Sigma}_K^{1/2}\right\|_F^2. \tag{30}$$

Write $\widehat{\mathbf{B}}_T(\Delta) = \mathbf{\Sigma}_Q^{1/2}\mathbf{B}_T(\Delta)\mathbf{\Sigma}_K^{1/2}$ for the weighted teacher operator.

CAPE initialization. Each teacher RoPE plane contributes cosine and sine terms to $\mathbf{B}_T(\Delta)$ (Su et al., 2021). We approximate these positional interactions using the target frequencies $\nu^{(u)}$, $u = 0, \dots, U-1$, specified in Eq. (28). Collect their cosine and sine functions in $\boldsymbol{\phi}(\Delta) = (\cos(\nu^{(0)}\Delta), \sin(\nu^{(0)}\Delta), \dots, \cos(\nu^{(U-1)}\Delta), \sin(\nu^{(U-1)}\Delta))^\top$. We first relax the two-dimensional projection structure by allowing $2U$ unrestricted coefficient matrices $\mathbf{X}_a \in \mathbb{R}^{d_{\text{model}}\times d_{\text{model}}}$. The relaxed objective is

$$\mathcal{L} = \mathbb{E}_{\Delta\sim p}\left\|\widehat{\mathbf{B}}_T(\Delta) - \sum_{a=0}^{2U-1}\phi_a(\Delta)\mathbf{X}_a\right\|_F^2.$$

Differentiating the least-squares objective gives

$$\frac{\partial\mathcal{L}}{\partial\mathbf{X}_a} = 2\left(\sum_b \mathbb{E}_{\Delta\sim p}[\phi_a(\Delta)\phi_b(\Delta)]\mathbf{X}_b - \mathbb{E}_{\Delta\sim p}[\phi_a(\Delta)\widehat{\mathbf{B}}_T(\Delta)]\right).$$

Let $G_{ab} = \mathbb{E}_{\Delta\sim p}[\phi_a(\Delta)\phi_b(\Delta)]$ denote the entries of the Gram matrix $\mathbf{G}$. A solution of the normal equations is

$$\mathbf{X}_a^\star = \sum_{b=0}^{2U-1}(\mathbf{G}^\dagger)_{ab}\mathbb{E}_{\Delta\sim p}[\phi_b(\Delta)\widehat{\mathbf{B}}_T(\Delta)]. \tag{31}$$

Here, $\mathbf{G}^\dagger$ is the Moore-Penrose pseudoinverse. This gives the minimum-norm solution of the unconstrained frequency projection (Penrose, 1956), with $\mathbf{G}$ accounting for correlations among the target basis functions. We next seek target projections $\widetilde{\mathbf{W}}_Q$ and $\widetilde{\mathbf{W}}_K$ whose induced weighted operator approximates the frequency expansion from Eq. (31):

$$\mathbf{\Sigma}_Q^{1/2}\widetilde{\mathbf{W}}_Q^\top\mathbf{R}_S(\Delta)\widetilde{\mathbf{W}}_K\mathbf{\Sigma}_K^{1/2} \approx \sum_{a=0}^{2U-1}\phi_a(\Delta)\mathbf{X}_a^\star.$$

For each target frequency $\nu^{(u)}$, we approximate the corresponding cosine and sine coefficients using a single rotation plane. Let the rows of $\widetilde{\mathbf{W}}_Q^{(u)}\mathbf{\Sigma}_Q^{1/2}$ and $\widetilde{\mathbf{W}}_K^{(u)}\mathbf{\Sigma}_K^{1/2}$ be $\mathbf{a}_{u,0}^\top, \mathbf{a}_{u,1}^\top$ and $\mathbf{b}_{u,0}^\top, \mathbf{b}_{u,1}^\top$, respectively. Then

$$\begin{aligned}
&\mathbf{\Sigma}_Q^{1/2}\left(\widetilde{\mathbf{W}}_Q^{(u)}\right)^\top\mathbf{R}_S^{(u)}(\Delta)\widetilde{\mathbf{W}}_K^{(u)}\mathbf{\Sigma}_K^{1/2} \\
&= \begin{bmatrix}\mathbf{a}_{u,0}^\top\\ \mathbf{a}_{u,1}^\top\end{bmatrix}^\top\begin{bmatrix}\cos(\nu^{(u)}\Delta) & -\sin(\nu^{(u)}\Delta)\\ \sin(\nu^{(u)}\Delta) & \cos(\nu^{(u)}\Delta)\end{bmatrix}\begin{bmatrix}\mathbf{b}_{u,0}^\top\\ \mathbf{b}_{u,1}^\top\end{bmatrix} \\
&\approx \cos(\nu^{(u)}\Delta)\mathbf{X}_{2u}^\star + \sin(\nu^{(u)}\Delta)\mathbf{X}_{2u+1}^\star.
\end{aligned} \tag{32}$$

Based on Eq. (32), define $\mathbf{a}_u = \mathbf{a}_{u,0} + i\mathbf{a}_{u,1}$, $\mathbf{b}_u = \mathbf{b}_{u,0} + i\mathbf{b}_{u,1}$, and $\mathbf{M}_u = \mathbf{X}_{2u}^\star + i\mathbf{X}_{2u+1}^\star$, where i is the imaginary unit:

$$\begin{aligned}
\text{Re}[e^{-i\nu_u\Delta}\mathbf{a}_u\mathbf{b}_u^H] &= \cos(\nu^{(u)}\Delta)(\mathbf{a}_{u,0}\mathbf{b}_{u,0}^\top + \mathbf{a}_{u,1}\mathbf{b}_{u,1}^\top) + \sin(\nu^{(u)}\Delta)(\mathbf{a}_{u,1}\mathbf{b}_{u,0}^\top - \mathbf{a}_{u,0}\mathbf{b}_{u,1}^\top)\\
\text{Re}[e^{-i\nu_u\Delta}\mathbf{M}_u] &= \cos(\nu^{(u)}\Delta)\mathbf{X}_{2u}^\star + \sin(\nu^{(u)}\Delta)\mathbf{X}_{2u+1}^\star
\end{aligned}$$

Minimizing the sum of the squared Frobenius errors of the cosine and sine coefficients is equivalent to approximating $\mathbf{M}_u$ by $\mathbf{a}_u\mathbf{b}_u^H$, whose rank is at most one. By the Eckart-Young-Mirsky

Table 5: Architectures of the teacher models.

Family	Teacher	Layer	Hidden size	Head	KV Head	d_{head}	Attn.	Native ctx.
FLA	transformer-340M-10B	24	1024	32	32	32	MHA	4K
	transformer-1.3B-100B	24	2048	32	32	64	MHA	2K
	transformer-2.7B-100B	32	2560	20	20	128	MHA	2K
Qwen2.5	0.5B-Base	24	896	14	2	64	GQA	32K
	1.5B-Base	28	1536	12	2	128	GQA	32K
	3B-Base	36	2048	16	2	128	GQA	32K
RedPajama	INCITE-Base-3B-v1	32	2560	32	32	80	MHA	2K
Mistral	7B-v0.1	32	4096	32	8	128	GQA/SWA	8K
LLaMA	Llama-2-7b-hf	32	4096	32	32	128	MHA	4K

theorem (Eckart & Young, 1936; Mirsky, 1960), the best rank-one approximation of $\mathbf{M}_u$ in Frobenius norm is given by its leading singular component. We apply singular value decomposition (SVD) to $\mathbf{M}_u$ and retain its leading singular component, $\mathbf{M}_u \approx \sigma_u \mathbf{l}_u \mathbf{r}_u^H$. Taking $\mathbf{a}_u = \sqrt{\sigma_u}\mathbf{l}_u$ and $\mathbf{b}_u = \sqrt{\sigma_u}\mathbf{r}_u$ gives

$$\widetilde{\mathbf{W}}_Q^{(u)} = \sqrt{\sigma}_u \begin{bmatrix} (\operatorname{Re}\mathbf{l}_u)^\top \\ (\operatorname{Im}\mathbf{l}_u)^\top \end{bmatrix} \Sigma_Q^{-1/2}, \qquad \widetilde{\mathbf{W}}_K^{(u)} = \sqrt{\sigma}_u \begin{bmatrix} (\operatorname{Re}\mathbf{r}_u)^\top \\ (\operatorname{Im}\mathbf{r}_u)^\top \end{bmatrix} \Sigma_K^{-1/2}. \tag{33}$$

Stacking the U two-row blocks in Eq. (33) gives the $2U$-dimensional projections required by Sec. 3.1. The frequency projection is optimal before imposing the plane structure, and each rank-one truncation is optimal for its coefficient matrix. Because the basis functions can be correlated and unequally weighted under $p(\Delta)$, the separate truncations do not generally minimize the joint constrained objective in Eq. (30). They instead provide the structured initialization refined by forward KL alignment in Sec. 4.1.

NoPE initialization. For NoPE, $\mathbf{R}_S(\Delta) = \mathbf{I}$, so the target operator is independent of relative position. Under Eq. (30), its optimal unconstrained value in the weighted space is $\widehat{\mathbf{B}}_0 = \mathbb{E}_{\Delta\sim p}[\widehat{\mathbf{B}}_T(\Delta)]$: expanding the squared error around this mean leaves a constant term plus the squared distance to $\widehat{\mathbf{B}}_0$. Since the target operator has rank at most $2U$, its optimal constrained approximation is the truncated SVD of this mean. Let $\mathbf{U}_r\mathbf{S}_r\mathbf{V}_r^\top$ denote the truncation with $r = \min(2U, d_k)$. The corresponding projections are

$$\widetilde{\mathbf{W}}_Q = \mathbf{S}_r^{1/2}\mathbf{U}_r^\top \mathbf{\Sigma}_Q^{-1/2}, \qquad \widetilde{\mathbf{W}}_K = \mathbf{S}_r^{1/2}\mathbf{V}_r^\top \mathbf{\Sigma}_K^{-1/2}. \tag{34}$$

If $r < 2U$, zero rows complete the required dimensions. This construction averages the teacher's positional interactions under $p(\Delta)$ rather than simply removing its RoPE rotations.

Both constructions are compatible with the adapter in Sec. 4.1. The weighted teacher operators have column space contained in that of $\mathbf{\Sigma}_Q^{1/2}\mathbf{W}_Q^\top$ and row space contained in that of $\mathbf{W}_K\mathbf{\Sigma}_K^{1/2}$. Frequency projection preserves these containments, as do the retained nonzero singular components. After inverse weighting, the constructed query and key projections therefore lie in the row spaces of $\mathbf{W}_Q$ and $\mathbf{W}_K$, respectively. Consequently, adapters realizing these projections are $\mathbf{P}_Q = \widetilde{\mathbf{W}}_Q\mathbf{W}_Q^\dagger$ and $\mathbf{P}_K = \widetilde{\mathbf{W}}_K\mathbf{W}_K^\dagger$, which satisfy $\widetilde{\mathbf{W}}_Q = \mathbf{P}_Q\mathbf{W}_Q$ and $\widetilde{\mathbf{W}}_K = \mathbf{P}_K\mathbf{W}_K$.

Finally, the derivation matches unscaled interactions. Including the factor $(2U/d_k)^{1/4}$ in both adapters preserves the teacher's dot-product coefficient because $(2U/d_k)^{1/2}/\sqrt{2U} = 1/\sqrt{d_k}$. The resulting projections provide the initialization for forward KL alignment and subsequent circular address folding.

B Experiments

B.1 Teacher Models

We evaluate the generality of our migration method on nine teacher models from five families with different architectural configurations, as summarized in Table 5. The five families are FLA FLA Hub, Qwen2.5 Qwen et al. (2025), RedPajama Together Computer (2023), Mistral Jiang et al.

Table 6: Overview of Transformer-to-RAM-Net migration. The total training budget is 500M tokens per teacher.

Phase	Stage	Main Loss	Trainable Parameters	Tokens	Warmup	LR
Intermediate	Low-rank & positional alignment	Attention KL	QK adapters, log-scales	7.209M	0.360M	3×10^{-4}
	Circular soft quantization	–	–	0	–	–
Recovery	Head-level	Head-output NMSE	W_r, W_w, scales, γ_t	49.152M	1.638M	1×10^{-4}
	Attn.-block local	Residual-update NMSE	Attention branch	61.440M	2.048M	1×10^{-4}
	Attn.-block stack	Residual-update NMSE	Attention branch	36.864M	1.638M	3×10^{-5}
	Full-model	0.25 Logit KL + CE	All parameters	345.375M	0.819M	1×10^{-5}

Table 7: Mass-aware readout results. Values in parentheses indicate the score differences relative to RAM-Net in Table 1.

Family	Teacher	Wiki PPL ↓	ARC-E ↑	ARC-C ↑	PIQA ↑	SciQ ↑	COPA ↑	Hella. ↑	Avg. ↑	R ↑
FLA	transformer-340M-10B	35.28 (+0.1)	**55.8 (+1.1)**	**23.9 (+0.2)**	64.3 (-0.4)	**80.3 (+0.1)**	**66.0 (+1.0)**	**31.1 (+0.1)**	**53.6 (+0.4)**	**87.7 (+1.2)**
	transformer-1.3B-100B	**21.83 (-0.1)**	52.3 (-0.4)	23.7 (-1.1)	67.7 (-0.6)	78.0 (-0.8)	**72.0 (+1.0)**	**35.8 (+0.2)**	54.9 (-0.3)	86.4 (-0.1)
	transformer-2.7B-100B	20.09 (+0.3)	**56.8 (+0.3)**	**25.4 (+0.8)**	69.6 (-0.7)	79.4 (-0.5)	**76.0 (+2.0)**	37.8 (-0.5)	**57.5 (+0.2)**	**89.6 (+0.4)**
Qwen2.5	0.5B-Base	23.55 (+0.2)	**64.1 (+0.5)**	**28.8 (+2.4)**	68.4 (-0.1)	81.9 (0.0)	69.0 (-4.0)	36.8 (-0.1)	58.2 (-0.2)	**86.1 (+6.9)**
	1.5B-Base	17.30 (+1.0)	74.8 (-0.5)	40.2 (-1.3)	**74.7 (+0.4)**	88.4 (-1.4)	77.0 (-5.0)	44.7 (-0.9)	66.6 (-1.5)	90.2 (-4.8)
	3B-Base	13.89 (+0.3)	77.2 (0.0)	**44.2 (+0.5)**	77.0 (0.0)	91.6 (-1.3)	**81.0 (+2.0)**	50.3 (-0.4)	**70.2 (+0.1)**	**92.3 (+0.8)**
RedPajama	INCITE-Base-3B-v1	17.71 (+1.1)	**63.9 (+0.3)**	**29.8 (+2.1)**	71.4 (-0.8)	**81.1 (+0.5)**	74.0 (-3.0)	41.0 (-0.9)	60.2 (-0.3)	**82.1 (+2.4)**
Mistral	7B-v0.1	10.32 (+0.2)	76.7 (-0.5)	**44.7 (+0.6)**	79.5 (-0.2)	**92.3 (+0.8)**	**88.0 (+1.0)**	57.4 (-0.6)	**73.1 (+0.2)**	**89.8 (+0.4)**
LLaMA	Llama-2-7b-hf	12.98 (+1.1)	71.2 (-0.4)	**39.4 (+0.4)**	76.2 (0.0)	89.0 (-0.2)	76.0 (-5.0)	50.4 (-1.2)	67.0 (-1.1)	84.6 (-2.6)

(2023), and LLaMA-2 Touvron et al. (2023). All teachers are loaded from their corresponding publicly available pretrained checkpoints on Hugging Face. We use the 10BT sample of the FineWeb-Edu dataset HuggingFaceFW (2024) for training. Their different head dimensions, numbers of query and KV heads, and attention configurations lead to different low-rank adapter shapes and head-wise CAPE/NoPE assignments during migration.

B.2 Migration Details

RAM-Net configurations. Unless otherwise specified, RAM-Net uses $d_p = 4$, $U = 5$, and Top-$K = 32$, and the retained address probabilities are not renormalized after Top-K. Output normalization is disabled throughout migration because the Transformer teacher has no corresponding operation. The sigmoid output gate is introduced only at the beginning of attention-block recovery. We initialize its projection weight to zero and bias to $b_g = 3$, giving a constant gain $g_0 = \sigma(3)$, and rescale the copied output projection as $W^o \leftarrow W^o/g_0$. This preserves the attention output at insertion while allowing the gate to become learnable in subsequent recovery.

Optimizer. Table 6 summarizes the complete training configuration. All stages use AdamW with zero weight decay and gradient clipping at 1. Forward and backward computation use BF16 with FP32 master weights. Each optimizer step processes 16K tokens. Low-rank alignment, head-level recovery, and attention-block recovery use linear warmup followed by cosine decay, while full-model recovery uses a short warmup followed by a constant learning rate. After each training stage, we restore the checkpoint with the lowest validation loss before proceeding to the next stage.

Overall workflow. During low-rank and positional alignment, only the adapters and head-wise log-scales are optimized, while the teacher and backbone remain frozen. Circular soft quantization analytically folds the selected adapters into RAM-Net read and write projections and requires no training. Head-level recovery then trains the RAM routing parameters and dynamic forgetting parameters on the native Top-K path while keeping the copied V/O projections and backbone frozen.

Attention-block recovery contains two consecutive steps. The local step feeds each student block the corresponding teacher hidden state, while the stack step propagates student hidden states through the complete stack. Both optimize the residual-update objective $\mathcal{L}_{\text{block}} = \frac{1}{L}\sum_{\ell=0}^{L-1} \text{NMSE}\big(\Delta z_\ell^S, \Delta z_\ell^T\big)$, where $\Delta z_\ell = z_\ell - u_\ell$, and u_ℓ and z_ℓ denote the block input and

Table 8: JS divergence with and without mass-aware readout after enabling GSU.

Readout rule	FLA-0.3B	FLA-1.3B	FLA-2.7B	Qwen-0.5B	Qwen-1.5B	Qwen-3B	Red.-3B	Mist.-7B	LLaMA-7B
w/o	0.277	0.286	0.308	0.292	**0.293**	0.312	**0.312**	0.283	**0.253**
w/	**0.259**	**0.273**	**0.300**	**0.290**	0.297	**0.311**	0.319	**0.280**	0.257

Table 9: Train-from-scratch results for 340M models with and without mass-aware readout.

	Wiki PPL ↓	ARC-E ↑	ARC-C ↑	PIQA ↑	SciQ ↑	COPA ↑	Hella. ↑	Avg. ↑
w/o	26.89	59.1	24.7	66.9	82.5	67.0	33.4	55.6
w/	27.10	59.1	26.1	67.5	82.1	73.0	33.3	56.8

output. Both steps train the complete attention branch, including RAM routing, dynamic forgetting, V/O projections, and the output gate, while MLP, normalization, embedding, and language-model-head parameters remain frozen. The copied V/O projections use $0.1\times$ the base learning rate.

Full-model recovery finally unfreezes all parameters and optimizes $\mathcal{L}_{\text{full}}$. RAM-specific parameters use the base learning rate, while parameters copied from the teacher use $0.1\times$ the base learning rate.

For both attention-block recovery steps, copied V/O parameters use $0.1\times$ the listed learning rate. During full-model recovery, all parameters copied from the teacher similarly use $0.1\times$ the listed rate, while RAM-specific parameters use the full rate.

B.3 Performance of Mass-Aware Readout

Tables 7 and 8 compare mass-aware and standard RAM-Net readouts. Consistent with the theoretical derivation in Sec. 3.3, mass-aware readout more closely matches the Transformer before head-level recovery, yielding lower attention JS in six of nine models. However, after migration, it generally gives higher Wiki PPL while maintaining comparable common-knowledge performance. This gap may be related to the coupling of read weights to forgetting dynamics and a potential mismatch between mass reweighting and Top-K selection based only on $r_t(\mathbf{a})$.

Table 9 further compares the two readouts when training a 340M model from scratch with the same FineWeb-Edu 10BT dataset. Mass-aware readout again gives slightly higher Wiki PPL, while improving the average common-knowledge score from 55.6 to 56.8. This trend is consistent with the migration results.

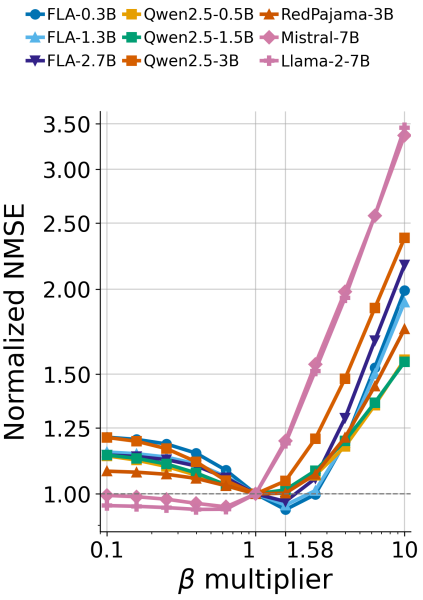

Figure 5: Head NMSE under different circular assignment concentrations β.

B.4 Additional Experimental Results

Detailed results on LongBench. Table 10 reports code completion and summarization results in LongBench for all teacher and RAM-Net pairs. RAM-Net recovers teacher performance more closely for smaller models, while the gap generally becomes larger at higher model scales. Since all models use the same 500M-token migration budget, these results suggest that larger teachers may require a larger recovery budget to better transfer their downstream capabilities.

Head NMSE with concentration β. Fig. 5 shows a trend consistent with the attention JS results in Fig. 3. Head NMSE is minimized near $\beta^\star$ across teachers, with some minima around $1.58\beta^\star$, further supporting $\beta^\star$ as a balanced default setting.

Table 10: LongBench results for all teacher and RAM-Net pairs.

Metric	Method	FLA-0.3B	FLA-1.3B	FLA-2.7B	Qwen-0.5B	Qwen-1.5B	Qwen-3B	Red.-3B	Mist.-7B	LLaMA-7B
LCC ↑	Teacher	17.0	48.3	50.7	10.0	14.7	14.1	44.7	70.0	66.7
	RAM-Net	19.6	36.4	36.4	18.5	12.7	15.5	32.7	33.4	24.8
MultiNews ↑	Teacher	10.5	6.6	6.3	21.4	25.2	24.5	20.1	15.5	13.8
	RAM-Net	9.6	7.2	8.4	12.9	13.9	15.9	8.6	12.1	11.5
SAMSum ↑	Teacher	6.6	6.2	4.9	37.3	42.4	45.5	0.0	28.2	31.4
	RAM-Net	8.0	9.3	10.3	20.3	28.7	29.5	0.1	22.1	23.9
Avg. ↑	Teacher	11.3	20.3	20.6	22.9	27.4	28.0	21.6	37.9	37.3
	RAM-Net	12.4	17.6	18.3	17.3	18.4	20.3	13.8	22.5	20.1

Table 11: Effect of order U on Qwen2.5-1.5B. Logit KL is measured after full-model recovery.

Order	Low-rank KL ↓	Quant. JS ↓	Head NMSE ↓	Logit KL ↓	Wiki PPL ↓	ARC-E ↑	ARC-C ↑	PIQA ↑	SciQ ↑	COPA ↑	Hella. ↑	Avg. ↑	R ↑
4 (256 slots)	1.957	**0.253**	**0.177**	0.500	16.58	73.8	38.9	73.6	87.4	**82.0**	45.6	66.9	90.8
5 (1024 slots)	1.748	0.293	0.197	**0.465**	16.33	**75.3**	41.5	**74.3**	**89.8**	**82.0**	**45.6**	**68.1**	**95.0**
6 (4096 slots)	**1.555**	0.339	0.209	0.476	**16.01**	74.4	**41.6**	74.0	89.2	77.0	45.4	66.9	91.8

Head-recovery loss curves. Fig. 7 extends the low-rank alignment and circular codebook comparison in Table 2 to head-level recovery. Consistent with the lower attention JS reported in the main text, the circular codebook also starts from the lowest NMSE and shows the fastest error reduction during recovery. Together, these results show that the circular construction improves both the initial attention alignment and the efficiency of subsequent recovery.

Per-stage breakdown in migration. Using FLA-2.7B as an example, Fig. 8 shows how different migration stages affect attention matching and output recovery. Circular soft quantization introduces approximation error, while Top-K truncation causes the largest increase in attention KL. This is expected because the address-overlap approximation in Sec. 3.2 is defined over the complete soft read and write distributions, whereas Top-K discards part of the overlap and sparsifies the induced token-to-token matching. Despite this large attention mismatch, head-level recovery rapidly reduces the output NMSE while the attention KL remains high. This indicates that RAM-Net can recover teacher head outputs through its sparse address routing and recurrent updates without reproducing the teacher attention distributions exactly. Attention-block and full-model recovery further reduce the remaining output discrepancy. The slight increase in NMSE during full-model recovery reflects the shift from intermediate-output matching to the final logit KL and cross-entropy objectives.

Different d_p and U. Fig. 6 evaluates the effect of d_p and U during intermediate construction. Increasing U generally reduces the forward KL and head NMSE by preserving a higher-dimensional QK representation, while a larger d_p provides finer angular resolution for the circular codebook. However, lower intermediate error does not necessarily translate into better final migration performance. Since the number of RAM-Net slots grows as d_p^U, increasing either parameter also enlarges the address space and can reduce overlap mass and effective slot utilization, as observed in the original RAM-Net study. Table 11 reflects this trade-off, where $U = 5$ achieves the best final Avg. and relative score despite $U = 6$ having lower low-rank KL. We therefore use $d_p = 4$ and $U = 5$ as a balanced setting between intermediate approximation quality and final language-model performance.

Ablations on staged migration. Table 12 provides a detailed comparison of the migration pipeline. *Complete pipeline* denotes the full staged procedure, while *full-model only* performs only full-model recovery. The "matched" checkpoints use the same cumulative token budgets as the corresponding stages of the complete pipeline. *Pattern-free CAPE selection* determines CAPE heads directly from their relative alignment errors without the preset pattern. *w/o head-level recovery* and *w/o attention-block recovery* omit the corresponding recovery stage and reallocate its training tokens to full-model recovery.

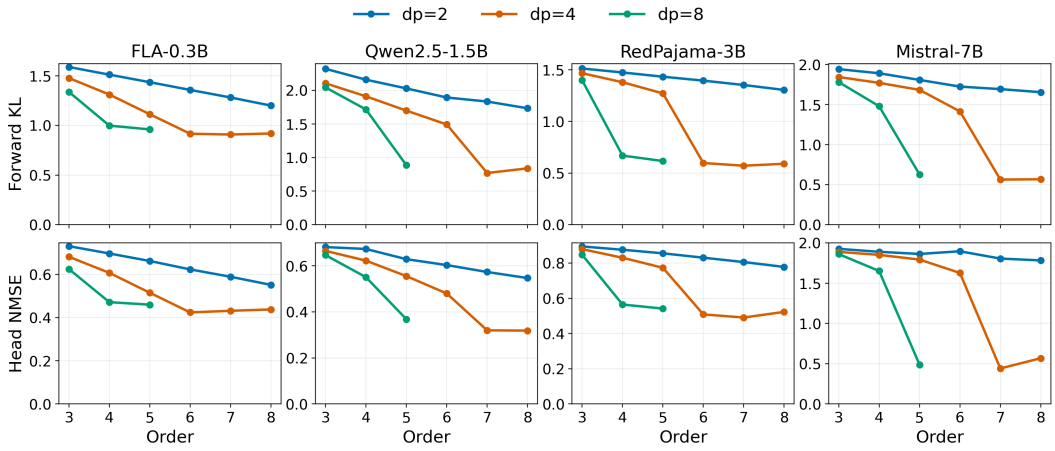

Figure 6: Effect of d_p and order U on intermediate forward KL and head NMSE.

Table 12: Ablation of the staged migration pipeline on Qwen2.5-1.5B.

Method	Checkpoint	Tokens (M)	Logit KL ↓	Wiki ↓	ARC-E	ARC-C	PIQA	SciQ	COPA	Hella.	Avg. ↑	R ↑
Teacher	Original	–	–	12.60	75.2	41.1	75.6	94.2	83.0	50.2	69.9	–
Complete	Head-Rec. end	56.4	–	77.56	47.1	25.1	64.2	69.9	69.0	35.2	51.7	43.8
	Block-Rec. end	154.7	–	19.44	72.1	37.6	73.6	87.2	79.0	44.1	65.6	86.3
	Final	500.0	0.164	16.33	75.3	41.5	74.3	89.8	82.0	45.6	68.1	95.0
Full-Model only	Matched head	56.4	0.419	23.90	65.1	31.5	70.2	83.0	77.0	40.7	61.2	71.1
	Matched block	154.7	0.278	19.35	70.1	35.7	73.6	87.3	78.0	43.2	64.7	82.6
	Final	500.0	0.197	16.82	71.8	39.2	74.3	89.5	80.0	44.6	66.6	89.7
Pattern-free CAPE	Final	500.0	0.162	16.17	74.3	41.1	74.3	89.1	81.0	45.8	67.6	93.7
w/o head Rec.	Final	500.0	0.166	16.31	72.8	39.7	74.4	89.2	80.0	45.2	66.9	90.8
w/o block Rec.	Final	500.0	0.181	16.64	74.3	39.2	73.9	89.4	79.0	45.4	66.9	90.2

The *full-model only* baseline achieves higher task scores at the early matched-head budget, but the *complete pipeline* overtakes it after attention-block recovery and reaches higher final Avg. and relative score R under the same 500M-token budget. Removing either head-level or attention-block recovery also lowers the final Avg. and R, even when Wiki PPL remains similar. These results show that the intermediate recovery stages improve teacher-relative downstream performance beyond what can be achieved by allocating the same training budget directly to full-model recovery.

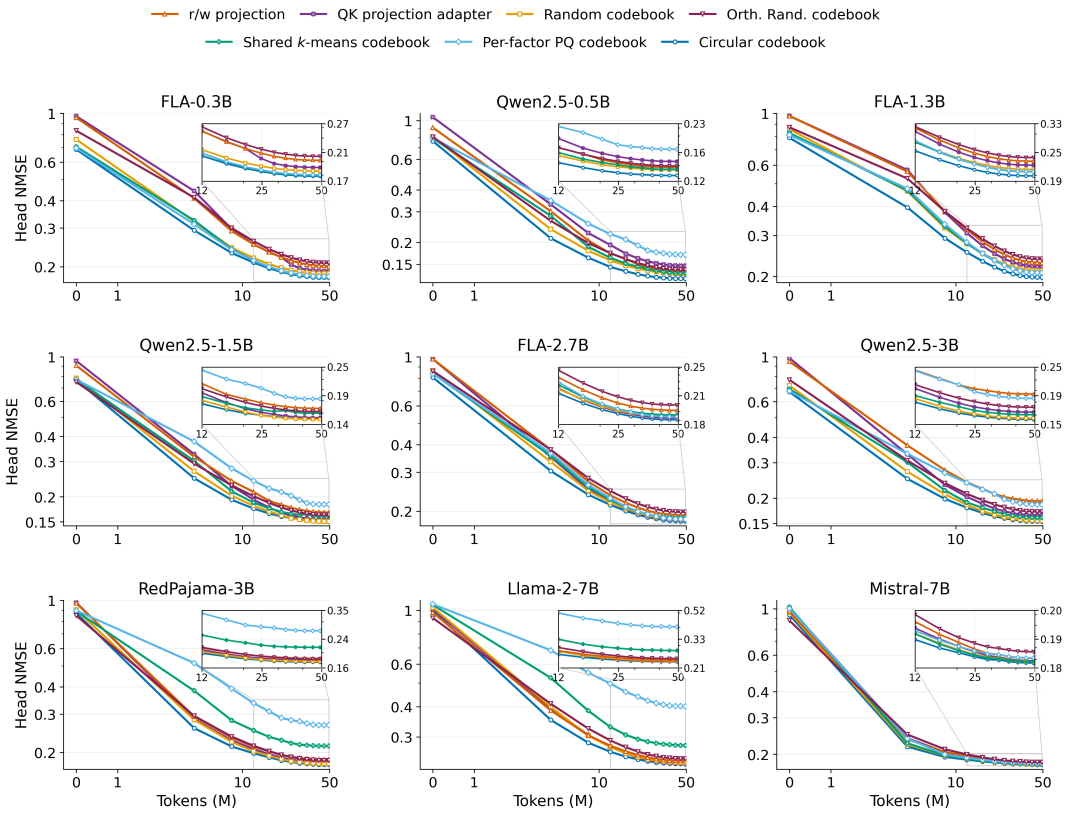

Figure 7: Head-level recovery NMSE under different low-rank adapter and codebook constructions.

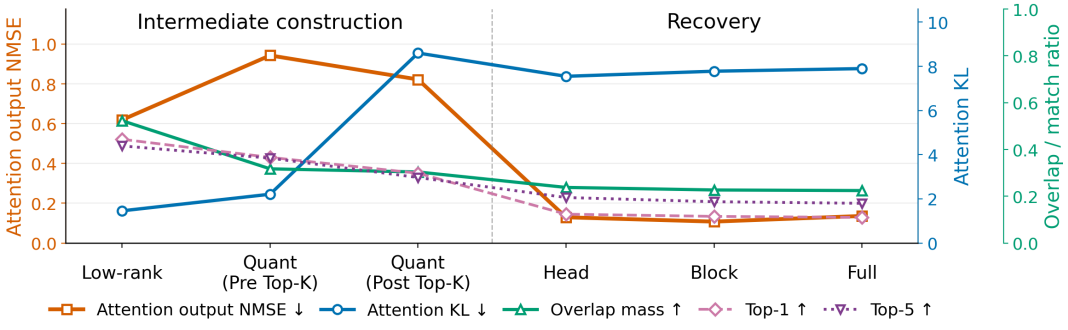

Figure 8: Stage-wise evolution throughout the migration pipeline of FLA-2.7B.