On State Reduction in Linear Attention

Philipp Nazari [1 2 3] **T. Konstantin Rusch** [1 3 4 5]

Abstract

Linear attention offers a computationally efficient yet expressive alternative to softmax attention. However, recent empirical results indicate that the hidden state of trained linear attention models often exhibits a low-rank structure, suggesting that these models underexploit their capacity in practice. To illuminate this phenomenon, we provide a theoretical analysis of the role of rank in linear attention, revealing that low effective rank can amplify query noise during readout from the associative memory. In addition to these theoretical insights, we conjecture that the low-rank states can be substantially reduced post-training with only minimal performance degradation, yielding faster and more memory-efficient models. To this end, we propose a novel hardware-aware approach that structurally prunes key and query matrices, reducing the state size while retaining compatibility with existing kernels. We adapt several existing pruning strategies to fit our framework and, building on our theoretical analysis, propose a novel structured pruning method based on a rank-revealing QR decomposition. Our empirical results, evaluated across models of varying sizes and on various downstream tasks, demonstrate the effectiveness of our state reduction framework. The code for this project can be found at https://github.com/camail-official/LinearAttentionPruning

1. Introduction

Linear Attention (Katharopoulos et al., 2020; Schlag et al., 2021; Sun et al., 2023b; Peng et al., 2023; Gu & Dao, 2024; Yang et al., 2024b; Dao & Gu, 2024a; Yang et al., 2024a; Team et al., 2025) has emerged as an efficient alternative to softmax attention (Vaswani et al., 2017), enabling high-throughput chunkwise parallel training (Hua et al., 2022; Sun et al., 2023b; Lingle, 2023; Yang et al., 2023) with linear time complexity and constant memory inference. These efficiency gains have recently driven the development of large hybrid models (Lieber et al., 2024; Li et al., 2025; Blakeman et al., 2025; Team, 2025) which predominantly employ linear attention layers, interspersed with only a few softmax attention layers.

Despite their impressive performance, prior work indicates that linear attention models still underutilize their capacity in practice (Siems et al., 2025; Parnichkun et al., 2025). In particular, the matrix-valued hidden states exhibit a low-rank structure after training. In this work, we demonstrate how this structure can increase the model's sensitivity to query noise. Through the lens of *linear associative memories* (Ramsauer et al., 2020; Wang et al., 2025), the hidden state acts as a storage for sequence history. Our observation of low effective rank indicates that the model might be using its memory inefficiently, effectively wasting its capacity. This finding suggests that the state size can be reduced post-training, yielding models that are both faster and more memory-efficient.

Towards this end, we propose a structured pruning framework to reduce the size of the hidden states in linear attention models. Within this framework, our experiments reveal that we can consistently remove approximately 50% of the key and query channels at only a minor increase in perplexity, even before recovery fine-tuning (Hu et al., 2022; Ma et al., 2023; Ashkboos et al., 2024a). Crucially, our approach is compatible with causal convolutions (Chollet, 2017; So et al., 2021; Fu et al., 2022; Yang et al., 2023; Gu & Dao, 2024; Dao & Gu, 2024a; Yang et al., 2024b) by avoiding internal state-space rotations, unlike methods such as SpinQuant (Liu et al., 2024) and QuaRot (Ashkboos et al., 2024b). However, our framework remains fully compatible with the residual stream rotations employed by these methods (as well as SliceGPT (Ashkboos et al., 2024a)), allowing for further efficiency gains.

Specifically, our approach relies on the observation that linear attention models are invariant under orthogonal transformations applied jointly to the queries and keys. Build-

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany [2]Department of Computer Science, ETH Zürich, Zürich, Switzerland [3]ELLIS Institute Tübingen [4]Tübingen AI Center [5]Liquid AI. Correspondence to: Philipp Nazari <philipp.nazari@tuebingen.mpg.de>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

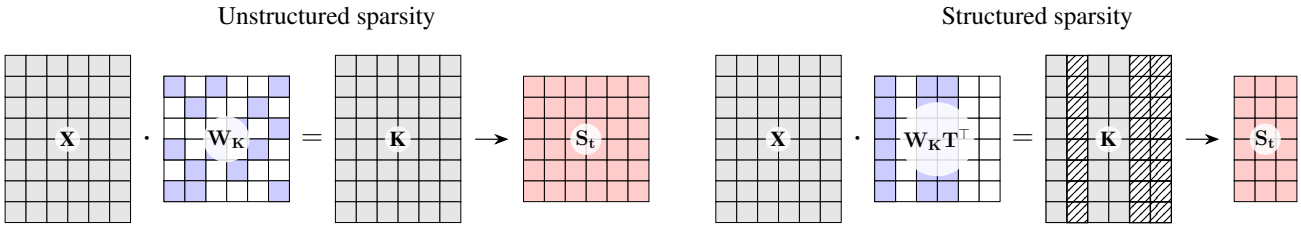

Figure 1. **Left:** Unstructured pruning yields a sparse weight matrix $\mathbf{W_K}$ yet preserves the column dimension of $\mathbf{K}$, leaving the domain of the state matrix $\mathbf{S}_t \in \mathbb{R}^{d_v, d_k}$ invariant. **Right:** Structured pruning eliminates basis vectors, mapping keys to a lower-dimensional space $\mathbb{R}^{d'_k}$ where $d'_k < d_k$. This results in a compressed state $\mathbf{S}_t \in \mathbb{R}^{d_v, d'_k}$. This reduction strictly decreases the FLOP count required to compute the recurrence. This figure is inspired by Ashkboos et al. (2024a, Figure 1).

ing on this insight, we seek transformations that select the columns of the keys and queries that contribute substantially to model performance. Within this framework, we provide a reformulation of several established pruning strategies, including those based on parameter magnitude and gradient saliency. Motivated by theoretical insights, we furthermore introduce a novel structured pruning approach that selects a subset of columns that maximizes the rank utilization of the remaining memory.

In summary, our **main contributions** are:

- We provide theoretical insights into the role of rank in linear attention (Section **??**). In particular, we show that rank utilization affects readout error and can amplify query noise.
- Motivated by the low rank utilization observed in practice, we formulate a post-training state-size reduction framework (Section 3). Specifically, we show that the state can be reduced substantially by jointly selecting subsets of channels from the keys and queries.
- We adapt several existing pruning strategies to fit our framework and, building on our theoretical analysis, propose a structured pruning method based on rank-revealing QR decompositions (Section 3.3).
- We present extensive empirical results on pre-trained DeltaNet and Gated DeltaNet models on downstream language-model evaluations, demonstrating the effectiveness of our proposed framework (Section 4.1).

2. Theoretical Insights

Linear attention (Katharopoulos et al., 2020; Schlag et al., 2021) is motivated by removing the softmax operator in self attention (Vaswani et al., 2017). Let $\mathbf{X} \in \mathbb{R}^{T,h}$ be an input that gets mapped to queries $\mathbf{Q} = \mathbf{X}\mathbf{W_Q} \in \mathbb{R}^{T,d_k}$, keys $\mathbf{K} = \mathbf{X}\mathbf{W_K} \in \mathbb{R}^{T,d_k}$, and values $\mathbf{V} = \mathbf{X}\mathbf{W}_v \in \mathbb{R}^{T,d_v}$ via linear projections. Linear attention can then be expressed as the recurrence,

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \mathbf{v}_t \mathbf{k}_t^\top, \tag{1}$$

where $\mathbf{S}_t \in \mathbb{R}^{d_v, d_k}$ is a matrix-valued hidden state used to compute the output $\mathbf{o}_t = \mathbf{S}_t \mathbf{q}_t$. This state implements a *linear associative memory* (Ramsauer et al., 2020; Wang et al., 2025) that is addressed using the queries. Throughout this work, we will refer to the update rule of $\mathbf{S}_t$ as the sequence mixer.

The mechanism defined in Equation (1) can only write *into* the associative memory. DeltaNet (Schlag et al., 2021; Yang et al., 2024b) addresses this limitation by explicitly erasing old information correlated with the current key before writing the new value to $\mathbf{S}_t$, i.e.,

$$\mathbf{S}_t = \mathbf{S}_{t-1}(\mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top) + \beta_t \mathbf{v}_t \mathbf{k}_t^\top. \tag{2}$$

Gated DeltaNet (Yang et al., 2024a) further refines this mechanism by introducing a data-dependent decay term.

2.1. On the Role of the Rank

A commonly identified weakness of linear attention compared to its softmax counterpart is its fixed-sized associative memory. However, recent empirical results indicate that these models do not manage this memory well (Siems et al., 2025; Parnichkun et al., 2025), exhibiting an effective low-rank structure in practice. We confirm this finding for DeltaNet 370M, showing that its hidden state is characteristically heavy-tailed (Figure 2). In the following, we build a framework that illuminates this phenomenon.

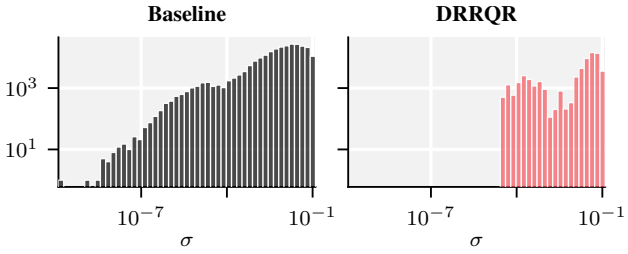

Figure 2. Singular value spectra for a single DeltaNet 370M head, aggregated over tokens (Fineweb-Edu, $T = 2048$, skipping the first 128 tokens). **Left:** Uncompressed baseline. **Right:** DRRQR at 75% compression (pre-RFT).

By construction, the subspace spanned by the associative

memory is governed by the keys and values. Formally, for a family of linear attention models, the row and column spaces of the associative memory satisfy,

$$\operatorname{row} \mathbf{S}_t \subseteq \operatorname{span}(\mathbf{k}_1, \ldots, \mathbf{k}_t), \quad \operatorname{col} \mathbf{S}_t \subseteq \operatorname{span}(\mathbf{v}_1, \ldots, \mathbf{v}_t).$$

It thus follows that,

$$\operatorname{rank} \mathbf{S}_t \leq \min\left(\operatorname{rank} \mathbf{K}_t, \operatorname{rank} \mathbf{V}_t\right) \leq t, \tag{3}$$

where $\mathbf{K}_t = [\mathbf{k}_1, \ldots, \mathbf{k}_t]$ and $\mathbf{V}_t = [\mathbf{v}_1, \ldots, \mathbf{v}_t]$ are the matrices obtained by stacking the keys and values, respectively. The proof is presented in Appendix H.2.

Equation (3) shows that the algebraic rank of the associative memory is bounded by the algebraic ranks of the keys and values. However, the algebraic rank is too rigid to serve as a meaningful measure for noisy real-world data. To address this, we consider the *effective rank* (or *stable rank*) (Rudelson & Vershynin, 2007; Tropp et al., 2015; Vershynin, 2018; Ipsen & Saibaba, 2025) instead: Given a matrix $\mathbf{S} \in \mathbb{R}^{d_v, d_k}$, its *effective rank* is defined as

$$\operatorname{er}(\mathbf{S}) \coloneqq \|\mathbf{S}\|_F^2 / \|\mathbf{S}\|_2^2 = \sum_i \sigma_i^2 / \sigma_1^2.$$

with $\sigma_1 \geq \ldots \geq \sigma_{\min(d_v, d_k)} \geq 0$ the singular values of $\mathbf{S}$. It measures the skewness of the singular value spectrum. The following proposition generalizes Equation (3), relating the effective rank of the associative memory to the conditioning of the keys. It serves as a first tool that provides control over the effective rank of the associative memory:

Proposition 2.1. *Consider the linear attention recurrence $\mathbf{S}_t = \mathbf{S}_{t-1} + \mathbf{v}_t \mathbf{k}_t^\top$. There exists a scalar quantity $\nu(\mathbf{V}_t)$ such that the effective rank of the memory is lower bounded:*

$$\nu(\mathbf{V}_t) / \kappa^2(\mathbf{K}_t) \leq \operatorname{er}(\mathbf{S}_t),$$

where κ denotes the l^2 condition number.

The proof of this proposition is presented in Appendix H.1. Although this statement holds for the specific case of plain linear attention, we use it as an approximation for (Gated) DeltaNet. It establishes that the conditioning of the keys, $\kappa(\mathbf{K}_t)$, is tightly connected to the effective rank of the hidden state. Towards our goal of improving the memory utilization of linear attention models, this proposition shows that improving the conditioning of the keys improves the effective rank of the memory.

2.1.1. Rank Utilization

While the effective rank measures the raw dimensionality of the stored information, it does not capture how *efficiently* a model uses its available memory. We thus introduce the notion of rank utilization:

Definition 2.2 (Rank Utilization)**.** Given a non-zero matrix $\mathbf{S} \in \mathbb{R}^{d_v, d_k}$, its *rank utilization* is the ratio of its effective rank and its theoretically maximal rank:

$$\mathrm{u}(\mathbf{S}) \coloneqq \operatorname{er}(\mathbf{S}) / \min(d_k, d_v).$$

Rank utilization combines effective rank and theoretically maximum capacity $d \coloneqq \min(d_k, d_v)$ and thus serves as a measure for memory utilization in linear attention models. We identify the following two edge cases:

- **Low Utilization** ($\mathrm{u} \ll 1$)**:** The memory suffers from *rank collapse*. Its energy is concentrated in a few principal components. The vast majority of information stored in the state is redundant.
- **High Utilization** ($\mathrm{u} \approx 1$)**:** The memory is *isotropic*. Energy is distributed evenly across all dimensions.

Proposition 2.1 directly relates the conditioning of the keys to the rank utilization of the associative memory:

$$\frac{\nu(\mathbf{V}_t)}{d\kappa^2(\mathbf{K}_t)} \leq \mathrm{u}(\mathbf{S}_t).$$

In particular, at a fixed capacity d, rank utilization of $\mathbf{S}_t$ can be increased by improving the conditioning of the keys.

2.1.2. Why does Rank Utilization Matter?

Let $\mathbf{S} = \sum_{i=1}^{d} \sigma_i \mathbf{u}_i \mathbf{w}_i^\top$ be the SVD of the associative memory (we drop the subscript t for brevity), with singular values $\sigma_1 \geq \cdots \geq \sigma_d \geq 0$. Moreover, consider a noisy query $\tilde{\mathbf{q}} = \mathbf{q}^* + \boldsymbol{\epsilon}$, where $\mathbf{q}^*$ is the pure query and $\boldsymbol{\epsilon}$ is noise. To analyze how the memory structure affects the output, define two coefficients that capture the alignment of noise and signal with the principal axis:

$$\delta \coloneqq |\boldsymbol{\epsilon}^\top \mathbf{w}_1| / \|\boldsymbol{\epsilon}\|_2 \quad \text{and} \quad \gamma \coloneqq |\mathbf{q}^{*\top} \mathbf{w}_1| / \|\mathbf{q}^*\|_2.$$

The following theorem establishes that the relative readout error is governed by the effective rank $\operatorname{er}(\mathbf{S})$.

Theorem 2.3 (Effective Rank Governs Readout Error)**.** *The ratio of the relative output error to the input noise-to-signal ratio is governed by the effective rank of the memory:*

$$\frac{\delta}{\sqrt{\operatorname{er}(\mathbf{S})}} \leq \frac{\|\mathbf{o} - \mathbf{o}^*\|_2 / \|\mathbf{o}^*\|_2}{\|\boldsymbol{\epsilon}\|_2 / \|\mathbf{q}^*\|_2} \leq \frac{\sqrt{\operatorname{er}(\mathbf{S})}}{\gamma}. \tag{4}$$

The proof is presented in Appendix H.3. Identifying $\mathrm{u}(\mathbf{S}) = \operatorname{er}(\mathbf{S})/d$, Theorem 2.3 reveals the influence of rank utilization on the readout error. We highlight two regimes:

- **Low Utilization** ($u(\mathbf{S}) \ll 1$)**:** The system is highly sensitive to noise, unless the noise is orthogonal to the principal component of the memory ($\delta \approx 0$). Simultaneously, the upper bound is small only if the signal is

aligned with the principal component ($\gamma \approx 1$). The rank utilization acts as a multiplier to the alignment of the noise with the principal component.

- **High Utilization ($u(\mathbf{S}) \approx 1$):** The memory becomes isotropic, all dimensions carry the same energy. Rank utilization acts only as a weak multiplier on the alignment of the noise and signal with the principal component. The model is not overly sensitive to noise along the principal component.

3. The Proposed Pruning Approach

Motivated by our theoretical insights into the role of rank utilization in linear attention, we propose a structured pruning framework to reduce the per-head key dimension of linear attention models. This yields a strictly smaller hidden state with higher throughput and lower memory requirements.

3.1. Invariant Transformations for Linear Attention

Decomposing $\mathbf{S}_t$ row-wise reveals that the sequence mixer simulates d_v parallel, independent linear time-varying (LTV) dynamical systems. Indeed, let $\mathbf{h}_t^{(i)} \in \mathbb{R}^{d_k}$ denote the transpose of the i-th row of $\mathbf{S}_t$. Each value channel $i \in \{1, \dots, d_v\}$ follows the vector-valued dynamics

$$\mathbf{h}_t^{(i)} = \mathbf{A}_t^\top \mathbf{h}_{t-1}^{(i)} + \mathbf{B}_t v_{t,i}, \tag{5}$$

where the system matrices $\mathbf{A}_t = \mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top$ and $\mathbf{B}_t = \beta_t \mathbf{k}_t$ are shared across all value channels. The readout is computed via the vector $\mathbf{C}_t = \mathbf{q}_t^\top$.

The dynamical system in Equation (5) is invariant under the choice of basis in state-space (Chahine et al., 2026; Chen, 1984). That is, every transformation

$$(\mathbf{A}_t, \mathbf{B}_t, \mathbf{C}_t) \to (\mathbf{T}^{-\top} \mathbf{A}_t \mathbf{T}^\top, \mathbf{T}\mathbf{B}_t, \mathbf{C}_t \mathbf{T}^{-1})$$

derived from an invertible matrix $\mathbf{T} \in GL(d_k)$ leaves the input-output mapping invariant. However, requiring invertibility is not sufficient, as the transformed transition matrix

$$\tilde{\mathbf{A}}_t = \mathbf{T}^{-\top} \mathbf{A}_t \mathbf{T}^\top = \mathbf{I} - (\mathbf{T}^{-\top} \mathbf{k}_t)(\mathbf{T}\mathbf{k}_t)^\top,$$

is not symmetric and can thus not be expressed as a DeltaNet. If additionally $\mathbf{T} \in O(d_k)$ is an orthogonal change of basis, this transformation can be absorbed to preserve the structure of the DeltaNet recurrence:

Proposition 3.1 (Orthogonal Invariance of Sequence Mixing)**.** *Let $\mathbf{T} \in O(d_k)$ be an orthogonal matrix. Then the (Gated) DeltaNet attention mechanism is invariant under the* simultaneous *transformation $(\mathbf{k}_t, \mathbf{q}_t) \mapsto (\mathbf{T}\mathbf{k}_t, \mathbf{T}\mathbf{q}_t)$.*

3.2. State Size Reduction Requires Structured Pruning

Existing pruning methods for Large Language Models generally rely on unstructured or semi-structured sparsity (Frantar & Alistarh, 2023; Sun et al., 2023a). However, these methods do not reduce the state dimension of the dynamical system, as they leave the query and key vectors dense (see Figure 1). We thus focus on *structured* pruning. Formally, given a *semi-orthogonal* matrix $\mathbf{T} \in \mathbb{R}^{d_k', d_k}$ with target dimension $d_k' < d_k$, we apply it *jointly* to the keys and queries before computing the dynamical system:

$$(\mathbf{k}_t, \mathbf{q}_t) \to (\mathbf{T}\mathbf{k}_t, \mathbf{T}\mathbf{q}_t) \quad \forall t = 1, \dots, T.$$

By Proposition 3.1, if $d_k' = d_k$, this transformation yields an equivalent realization of the dynamical system. Our goal is finding a $\mathbf{T}$ with $d_k' \ll d_k$ that entails a small error.

The Convolution Constraint. To achieve actual wall-clock speedup, $\mathbf{T}$ must be absorbed into the weight matrices $\mathbf{W_K}$ and $\mathbf{W_Q}$. However, linear attention models typically employ causal convolutions (Chollet, 2017; So et al., 2021; Fu et al., 2022; Yang et al., 2023; Gu & Dao, 2024; Dao & Gu, 2024a; Yang et al., 2024b) to queries and keys (see Algorithm 1 in the Appendix for a prototypical DeltaNet layer). While the sequence mixer is compatible with orthogonal transformations (Proposition 3.1), depthwise convolutions are not. A dense semi-orthogonal matrix $\mathbf{T}$ would mix channels, misaligning them with their corresponding convolution filters. Preserving the dynamics would require converting the efficient diagonal convolution filters into expensive dense matrices. Therefore, we effectively constrain our framework to *axis-aligned* transformations. The case of general orthogonal matrices is discussed in Appendix D.4.

3.3. Axis-Aligned Methods

We focus on transformations that preserve the channel-wise convolutions. Formally, this constraints the design space to *axis-aligned* semi-orthogonal matrices. Given a set $\mathcal{I} = \{i_1, \dots, i_{d_k'}\} \subseteq \{1, \dots, d_k\}$ of distinct indices with cardinality $d_k' < d_k$, we define the axis-aligned projection matrix $\mathbf{P}_\mathcal{I} \in \mathbb{R}^{d_k', d_k}$ as the matrix whose rows are the standard basis vectors corresponding to $\mathcal{I}$. This matrix is semi-orthogonal. Applying it corresponds to a structural pruning operation that selects the subset of channels $\mathcal{I}$ and discards the rest. Crucially, this operation preserves the independence of the remaining channels:

Proposition 3.2 (Compatibility with Depthwise Convolutions)**.** *Let $\mathrm{Conv1D}(\mathbf{X}, \mathbf{W})$ denote a depthwise convolution on input $\mathbf{X} \in \mathbb{R}^{T, d_k}$ with per-channel filters $\mathbf{W} \in \mathbb{R}^{d_k, l}$ of size l. Then*

$$\mathrm{Conv1D}(\mathbf{X}, \mathbf{W})\mathbf{P}_\mathcal{I}^\top = \mathrm{Conv1D}(\mathbf{X}\mathbf{P}_\mathcal{I}^\top, \mathbf{P}_\mathcal{I}\mathbf{W}).$$

This result implies that key dimensions can be pruned by simply slicing the corresponding columns from the projection matrices $\mathbf{W_K}$ and $\mathbf{W_Q}$, along with the corresponding entries of the convolution weights $\mathbf{W}$. In the following paragraphs, we present a plethora of different strategies to

select the index set $\mathcal{I}$ for each head, ranging from simple heuristics to a novel rank-revealing approach.

3.3.1. Proposed Pruning Methods

Weight-Magnitude-based ($\mathbf{L^1}$). As a baseline, we implement a weight-magnitude based pruner, defining the importance of the j-th channel as the sum of the l^1 norms of the corresponding columns in the query and key projection matrices: $s_j = \|\mathbf{W}_{\mathbf{Q},:j}\|_1 + \|\mathbf{W}_{\mathbf{K},:j}\|_1$. We rank these scores locally within each head and select the top-d'_k indices to form the retained set $\mathcal{I}$.

S-Wanda. We adapt Wanda (Sun et al., 2023a) to our structured setting (coining the method *S-Wanda*), defining the saliency of the j-th channel by aggregating the element-wise Wanda scores across the channel dimension:

$$s_j = \sum_{i=1}^{h} \left(|\mathbf{W}_{q,ij}| + |\mathbf{W}_{k,ij}|\right) \|\mathbf{X}_{:i}\|_2,$$

where $\|\mathbf{X}_{:i}\|_2$ is the l^2 norm of the i-th input feature computed over a small calibration set.

Sensitivity-based. We employ a gradient-based saliency criterion (LeCun et al., 1989; Wang et al., 2019; Ma et al., 2023) to identify dimensions that maximally influence a loss It is the only method we consider that takes into account information beyond the current layer and is task-aware. We quantify the importance of the j-th key dimension using the first-order Taylor expansion of the loss on a calibration set:

$$s_j = \sum_i \left(\left|\mathbf{W}_{\mathbf{Q},ij} \nabla_{\mathbf{W}_{\mathbf{Q},ij}} \mathcal{L}\right| + \left|\mathbf{W}_{\mathbf{K},ij} \nabla_{\mathbf{W}_{\mathbf{K},ij}} \mathcal{L}\right|\right).$$

Rank-based (DRRQR). In light of our theoretical insights into the role of rank in linear attention models (Section 2.1.2), we propose a simple algorithm that explicitly improves the conditioning of the keys. By Proposition 2.1, this increases the effective rank of the associative memory. *Deep Rank Revealing QR* (*DRRQR*) applies a *Strong Rank Revealing QR* factorization (Gu & Eisenstat, 1996) to the activation statistics $\mathbf{M} = [\mathbf{K}, \mathbf{Q}]$ to select a subset of d'_k columns that form a well-conditioned set of channels. The algorithm (see Algorithm 2) starts from a QR decomposition with column pivoting and then iteratively permutes columns between a current chosen basis and a set of candidates to satisfy numerical stability bounds. *DRRQR* optimizes the conditioning of the retained columns and thus implicitly increases rank utilization (see Proposition F.5).

4. Experiments

We use *flame* (Zhang & Yang, 2025) for recovery fine-tuning (RFT, (Ashkboos et al., 2024a)) and pre-training the 370M parameter models. RFT employs LoRA (Hu et al., 2022) on a single H100 GPU. The details on our experimental setup may be found in Appendix C. Besides the methods introduced in Section 3, we also report a *Rand* baseline which randomly selects key channels to drop.

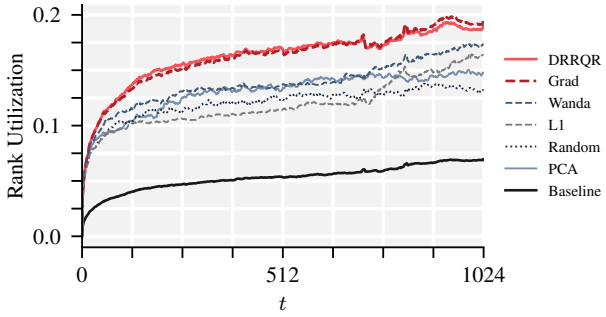

Figure 3. Rank utilization of DeltaNet 370M as a function of the token index for a random sample of Fineweb-Edu of length 1024, averaged over layers and heads, at a compression ratio of 75% pre-RFT. Baseline is not compressed.

4.1. Results

We evaluate our structured pruning methods on DeltaNet and Gated DeltaNet (Yang et al., 2024b;a) with both 370M and 1.3B parameters. Specifically, we compare the optimization-based approaches *DRRQR* (rank-optimal) and *Grad* (gradient-based) with the magnitude-based strategies.

Figure 3 shows the rank utilization during a forward pass through DeltaNet 370M at a compression ratio of 75%. We can see that the two most powerful compression methods, *Grad* and *DRRQR*, have the largest rank utilization. A fine-grained per-head analysis (Appendix E.1) reveals that rank utilization varies by an order of magnitude across heads, and that DRRQR lifts it consistently. Figure 2 furthermore shows how DRRQR removes the tail of the hidden states' spectrum (see Appendix E.3 for more results).

Language Modeling. We observe that pruning 50% of the key dimension entails only a small degradation in Wikitext and Lambada perplexity, especially when using the *Grad* and *DRRQR* pruners, even before RFT (see Table 8 for an overview of pre-RFT and post-RFT results). For instance, at a 50% compression ratio, Gated DeltaNet 1.3B's perplexity on Wikitext increases modestly from 16.8 to 17.3. Even at 75%, the perplexity only goes up by about two points. This finding suggests that these models effectively use at most half of their available capacity for next token prediction. Table 1 contains extended post-RFT results for Gated DeltaNet 1.3B (more comprehensive breakdowns are provided in Tables 8-12 in the Appendix). The zero-shot common sense reasoning scores remain robust under compression. For instance, Gated DeltaNet 1.3B maintains an average of 58.3 at 50% compression when compressed via *Grad*, around a one-point drop from the 59.4 baseline.

Table 1. Comprehensive post-RFT evaluation of Gated DeltaNet 1.3B. We report Wikitext and Lambada perplexities, the average zero-shot generation accuracy on language-model evaluations (**LM evals**, (Gao et al., 2024)) across varying compression ratios (best results in **bold**, second best underlined).

Method	75%			50%			40%			30%		
	Wiki ↓	LMB ↓	LM Evals ↑	Wiki ↓	LMB ↓	LM Evals ↑	Wiki ↓	LMB ↓	LM Evals ↑	Wiki ↓	LMB ↓	LM Evals ↑
Rand	19.3	18.1	54.6	16.9	11.5	57.8	16.5	10.7	58.3	16.1	10.2	58.7
L1	18.6	16.8	55.3	16.8	13.0	57.2	16.5	11.9	57.8	16.3	11.3	58.3
S-Wanda	18.2	15.1	55.4	16.6	11.7	57.8	16.4	11.5	58.1	16.1	11.0	58.6
Grad	**17.8**	**12.5**	**56.8**	16.4	**10.5**	**58.3**	**16.1**	**10.1**	58.6	**15.9**	**9.9**	**59.0**
DRRQR	17.9	14.7	56.2	**16.3**	12.0	58.0	**16.1**	11.0	**58.7**	**15.9**	10.7	58.8
Baseline	16.8	9.7	59.4	16.8	9.7	59.4	16.8	9.7	59.4	16.8	9.7	59.4

We generally find the rank-based method *DRRQR* to perform competitively with the gradient-saliency based *Grad*, even though it is local and task-agnostic. *Grad*, on the other hand, is non-local and removes weight in a way that entails a minimal increase in perplexity. This highlights the influence of our rank considerations on model performance. A spectral analysis of post-pruning keys reveals that DRRQR and Grad preserve nearly the full energy ($\sigma_{\max} \approx 53$), while L1 and S-Wanda retain low-energy columns ($\sigma_{\max} = 16$–24). Interestingly, L1 and S-Wanda achieve the lowest condition number κ, yet perform worst; they attain low κ by avoiding high-energy columns entirely. DRRQR's higher κ reflects its retention of high-energy columns combined with the highest $\sigma_{\min}$. The Grad-based method achieves κ comparable to DRRQR, indicating that gradient saliency implicitly selects the most informative dimensions.

Ablation on Column Selection. We investigate the impact of selecting columns based on just keys, just queries, or both. Our ablation study (detailed in Appendix E.5) reveals that magnitude-based heuristics (*L1*, *S-Wanda*) are unstable when targeting keys, performing best when restricted to queries. In contrast, the two best methods, *Grad* and *DRRQR*, consistently achieve the lowest perplexity when using a joint selection scheme. For comparability, all results in the main body of this paper use joint selection schemes for all methods. However, even when comparing the optimal configuration for magnitude based approaches (queries-only) against our proposed methods (queries and keys), *Grad* and *DRRQR* still demonstrate superior performance.

Speedup. Our structured pruning applies a joint semi-orthogonal projection to the keys and queries, reducing the per-head key dimension from d_k to $d'_k < d_k$. We benchmark the Gated DeltaNet 1.3B sequence-mixer's prefill throughput and peak VRAM on an NVIDIA H100 across multiple batch sizes (Table 2). At $B = 8$, a 50% compression yields a $1.26\times$ speedup with peak VRAM reduced to $0.73\times$, and 75% compression yields a $1.32\times$ speedup with VRAM reduced to $0.59\times$. These speedups and savings are even more pronounced from $B = 16$ onwards, enabling either larger batches or longer contexts under the same memory budget.

Table 2. Gated DeltaNet sequence-mixer speedup and peak VRAM, both relative to the uncompressed baseline at the same batch size, on an NVIDIA H100 ($T = 2048$, 16 heads, $h = 2048$, $d_v = 128$).

Batch Size	Compr.	Speedup	Mem
8	50%	1.26×	0.73×
	75%	1.32×	0.60×
16	50%	1.28×	0.73×
	75%	1.50×	0.59×
32	50%	1.28×	0.72×
	75%	1.51×	0.59×

5. Discussion

Motivated by the low-rank structure of the associative memory in linear attention models, we propose a state-size reduction framework that selects a subset of informative key and query columns. In addition to adapting several pruning strategies, we introduce a method explicitly designed to improve the conditioning of the keys. We further provide a rigorous analysis of the role of rank in linear attention, showing that low-rank structure can amplify query noise and govern readout error. Finally, we present empirical evidence demonstrating the practical efficiency and effectiveness of our structured pruning framework.

While this manuscript focuses on delta-rule models, the proposed structured pruning framework extends naturally to other linear recurrence and linear attention architectures, including Mamba-2 (Dao & Gu, 2024b). In Appendix E.2, we further validate this by reducing the state size of a 340M-parameter Mamba-2 model by 75%, and report both pre-RFT WikiText-2 perplexity and downstream task accuracies. Qualitatively, the observed behavior closely mirrors that of the DeltaNet models. These results suggest that our framework provides a broadly applicable approach to state reduction across the class of linear attention models.

References

Ashkboos, S., Croci, M. L., Nascimento, M. G. d., Hoefler, T., and Hensman, J. SliceGPT: Compress Large Language Models by Deleting Rows and Columns. *arXiv preprint arXiv:2401.15024*, 2024a.

Ashkboos, S., Mohtashami, A., Croci, M. L., Li, B., Cameron, P., Jaggi, M., Alistarh, D., Hoefler, T., and Hensman, J. Quarot: Outlier-Free 4-bit Inference in Rotated LLMs. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024b.

Blakeman, A., Basant, A., Khattar, A., Renduchintala, A., Bercovich, A., Ficek, A., Bjorlin, A., Taghibakhshi, A., Deshmukh, A. S., Mahabaleshwarkar, A. S., et al. Nemotron-H: A Family of Accurate and Efficient Hybrid Mamba-Transformer Models. *arXiv preprint arXiv:2504.03624*, 2025.

Chahine, M., Nazari, P., Rus, D., and Rusch, T. K. The Curious Case of In-Training Compression of State Space Models. In *International Conference on Learning Representations*, 2026.

Chen, C.-T. *Linear System Theory and Design*. Saunders college publishing, 1984.

Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Dao, T. and Gu, A. Transformers are SSMs: Generalized Models and Efficient Algorithms through Structured State Space Duality. *arXiv preprint arXiv:2405.21060*, 2024a.

Dao, T. and Gu, A. Mamba-2: A State Space Duality Framework. *International Conference on Machine Learning*, 2024b.

Das, R. J., Sun, M., Ma, L., and Shen, Z. Beyond Size: How Gradients Shape Pruning Decisions in Large Language Models. *arXiv preprint arXiv:2311.04902*, 2023.

Dong, P., Li, L., Tang, Z., Liu, X., Pan, X., Wang, Q., and Chu, X. Pruner-Zero: Evolving Symbolic Pruning Metric from Scratch for Large Language Models. *arXiv preprint arXiv:2406.02924*, 2024.

Dong, Y., Cordonnier, J.-B., and Loukas, A. Attention is not All Aou Need: Pure Attention Loses Rank Doubly Exponentially with Depth. In *International conference on machine learning*, pp. 2793–2803. PMLR, 2021.

Dong, Z., Yuan, J., Li, W., Zhang, H., Chen, S., Chen, H., Bian, J., and Yang, Y. MambaQuant: Quantizing the mamba family with variance aligned rotation methods. *arXiv preprint arXiv:2501.13644*, 2025.

Frantar, E. and Alistarh, D. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot. In *International conference on machine learning*, pp. 10323–10337. PMLR, 2023.

Fu, D. Y., Dao, T., Saab, K. K., Thomas, A. W., Rudra, A., and Ré, C. Hungry Hungry Hippos: Towards Language Modeling with State Space Models. *arXiv preprint arXiv:2212.14052*, 2022.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. The Language Model Evaluation Harness, 07 2024. URL https://zenodo.org/records/12608602.

Gu, A. and Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *First conference on language modeling*, 2024.

Gu, M. and Eisenstat, S. C. Efficient Algorithms for Computing a Strong Rank-Revealing QR Factorization. *SIAM Journal on Scientific Computing*, 17(4):848–869, 1996.

Horn, R. A. and Johnson, C. R. *Topics in Matrix Analysis*. Cambridge university press, 1994.

Hsu, Y.-C., Hua, T., Chang, S., Lou, Q., Shen, Y., and Jin, H. Language Model Compression with Weighted Low-Rank Factorization. *arXiv preprint arXiv:2207.00112*, 2022.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022.

Hua, W., Dai, Z., Liu, H., and Le, Q. Transformer Quality in Linear Time. In *International conference on machine learning*, pp. 9099–9117. PMLR, 2022.

Ipsen, I. C. and Saibaba, A. K. Stable Rank and Intrinsic Dimension of Real and Complex Matrices. *SIAM Journal on Matrix Analysis and Applications*, 46(3):1988–2007, 2025.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In *International conference on machine learning*, pp. 5156–5165. PMLR, 2020.

LeCun, Y., Denker, J., and Solla, S. Optimal Brain Damage. *Advances in neural information processing systems*, 2, 1989.

Li, A., Gong, B., Yang, B., Shan, B., Liu, C., Zhu, C., Zhang, C., Guo, C., Chen, D., Li, D., et al. Minimax-01: Scaling Foundation Models with Lightning Attention. *arXiv preprint arXiv:2501.08313*, 2025.

Lieber, O., Lenz, B., Bata, H., Cohen, G., Osin, J., Dalmedigos, I., Safahi, E., Meirom, S., Belinkov, Y., Shalev-Shwartz, S., et al. Jamba: A Hybrid Transformer-Mamba Language Model. *arXiv preprint arXiv:2403.19887*, 2024.

Lingle, L. D. Transformer-VQ: Linear-Time Transformers via Vector Quantization. *arXiv preprint arXiv:2309.16354*, 2023.

Liu, Z., Zhao, C., Fedorov, I., Soran, B., Choudhary, D., Krishnamoorthi, R., Chandra, V., Tian, Y., and Blankevoort, T. SpinQuant: LLM Quantization with Learned Rotations. *arXiv preprint arXiv:2405.16406*, 2024.

Ma, X., Fang, G., and Wang, X. LLM-Pruner: On the Structural Pruning of Large Language Models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

Noci, L., Anagnostidis, S., Biggio, L., Orvieto, A., Singh, S. P., and Lucchi, A. Signal Propagation in Transformers: Theoretical Perspectives and the Role of Rank Collapse. *Advances in Neural Information Processing Systems*, 35: 27198–27211, 2022.

Parnichkun, R. N., Tumma, N., Thomas, A. W., Moro, A., An, Q., Suzuki, T., Yamashita, A., Poli, M., and Massaroli, S. Quantifying Memory Utilization with Effective State-Size. *arXiv preprint arXiv:2504.19561*, 2025.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Biderman, S., Cao, H., Cheng, X., Chung, M., Grella, M., et al. RWKV: Reinventing RNNs for the Transformer Era. *arXiv preprint arXiv:2305.13048*, 2023.

Petersen, K. B., Pedersen, M. S., et al. The Matrix Cookbook. *Technical University of Denmark*, 7(15):510, 2008.

Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., et al. Hopfield Networks is All You Need. *arXiv preprint arXiv:2008.02217*, 2020.

Rudelson, M. and Vershynin, R. Sampling from Large Matrices: An Approach through Geometric Functional Analysis. *Journal of the ACM (JACM)*, 54(4):21–es, 2007.

Schlag, I., Irie, K., and Schmidhuber, J. Linear Transformers Are Secretly Fast Weight Programmers. In *International conference on machine learning*, pp. 9355–9366. PMLR, 2021.

Siems, J., Carstensen, T., Zela, A., Hutter, F., Pontil, M., and Grazzi, R. DeltaProduct: Improving State-Tracking in Linear RNNs via Householder Products. *arXiv preprint arXiv:2502.10297*, 2025.

So, D., Mańke, W., Liu, H., Dai, Z., Shazeer, N., and Le, Q. V. Searching for Efficient Transformers for Language Modeling. *Advances in neural information processing systems*, 34:6010–6022, 2021.

Sun, M., Liu, Z., Bair, A., and Kolter, J. Z. A Simple and Effective Pruning Approach for Large Language Models. *arXiv preprint arXiv:2306.11695*, 2023a.

Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., and Wei, F. Retentive Network: A Successor to Transformer for Large Language Models. *arXiv preprint arXiv:2307.08621*, 2023b.

Team, K., Zhang, Y., Lin, Z., Yao, X., Hu, J., Meng, F., Liu, C., Men, X., Yang, S., Li, Z., et al. Kimi Linear: An Expressive, Efficient Attention Architecture. *arXiv preprint arXiv:2510.26692*, 2025.

Team, Q. Qwen3-Next: Towards Ultimate Training & Inference Efficiency, 2025.

Trefethen, L. N. and Bau, D. *Numerical Linear Algebra*. SIAM, 2022.

Tropp, J. A. et al. An Introduction to Matrix Concentration Inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention Is All You Need. *Advances in neural information processing systems*, 30, 2017.

Vershynin, R. *High-Dimensional Probability: An Introduction with Applications in Data Science*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, 2018.

Wang, C., Grosse, R., Fidler, S., and Zhang, G. Eigendamage: Structured Pruning in the Kronecker-Factored Eigenbasis. In *International conference on machine learning*, pp. 6566–6575. PMLR, 2019.

Wang, K. A., Shi, J., and Fox, E. B. Test-Time Regression: a Unifying Framework for Designing Sequence Models with Associative Memory. *arXiv preprint arXiv:2501.12352*, 2025.

Xu, H.-Y., Huang, C.-E., and Wu, C.-W. Quamba: A post-training quantization recipe for selective state-space models. *arXiv preprint arXiv:2410.13229*, 2024a.

Xu, Y., Jie, Z., Dong, H., Wang, L., Lu, X., Saha, A., Savarese, S., and Sahoo, D. ThinK: Thinner Key Cache by Query-Driven Pruning. *arXiv preprint arXiv:2407.21018*, 2024b.

Yang, S., Wang, B., Shen, Y., Panda, R., and Kim, Y. Gated Linear Attention Transformers with Hardware-Efficient Training. *arXiv preprint arXiv:2312.06635*, 2023.

Yang, S., Kautz, J., and Hatamizadeh, A. Gated Delta Networks: Improving Mamba2 with Delta Rule. *arXiv preprint arXiv:2412.06464*, 2024a.

Yang, S., Wang, B., Zhang, Y., Shen, Y., and Kim, Y. Parallelizing Linear Transformers with the Delta Rule over Sequence Length. *Advances in neural information processing systems*, 37:115491–115522, 2024b.

Yang, Y., Zhen, K., Ganesh, B., Galstyan, A., Huybrechts, G., Müller, M., Kübler, J. M., Swaminathan, R. V., Mouchtaris, A., Bodapati, S. B., et al. Wanda++: Pruning Large Language Models via Regional Gradients. *arXiv preprint arXiv:2503.04992*, 2025.

Yuan, Z., Shang, Y., Song, Y., Wu, Q., Yan, Y., and Sun, G. ASVD: Activation-aware Singular Value Decomposition for Compressing Large Language Models. *arXiv preprint arXiv:2312.05821*, 2023.

Zhang, Y. and Yang, S. Flame: Flash Language Modeling Made Easy, January 2025. URL https://github.com/fla-org/flame.

A. Extended Related Work

B. Related Work

Rank Considerations. Recent empirical results imply that linear attention models do not manage their associative memory well (Siems et al., 2025; Parnichkun et al., 2025), indicating that the memory of linear attention models often exhibits a low-rank structure. Our work builds on these observations, forcing the model to operate in a lower-dimensional space at increased rank utilization and minimal decrease in effective rank (Ipsen & Saibaba, 2025). We note that rank collapse is a well-known phenomenon in transformers. (Dong et al., 2021) for instance show that skip connections help alleviate it, while (Noci et al., 2022) show that collapsed queries and keys hinder gradient flow at initialization. In this work, we tie the rank of the queries and keys to that of the hidden state in linear attention and show how a skewed spectrum of the associative memory can amplify query noise during readout.

Pruning Methods. Conventional pruning methods are usually either unstructured or semi-structured (LeCun et al., 1989; Frantar & Alistarh, 2023; Sun et al., 2023a) and thus require specialized kernels to realize sparsity-induced speedups. SparseGPT (Frantar & Alistarh, 2023) frames pruning as a local reconstruction problem. Wanda (Sun et al., 2023a) proposes a simpler, gradient-free metric based on the product of weight magnitudes and input activation norms. Several works have sought to enhance this metric by re-incorporating gradients. GBLM (Das et al., 2023) and Pruner-Zero (Dong et al., 2024) utilize gradients derived from full-model backpropagation to refine pruning scores. Wanda++ (Yang et al., 2025) introduces regional gradients to decrease the computational cost.

A challenge in structured pruning is the handling of coupled structures, where removing a neuron or head in one layer breaks dimensional consistency in subsequent layers. (Ma et al., 2023) (LLM-Pruner) address this by constructing dependency graphs to identify groups of parameters that must be excised simultaneously. Similarly, our work addresses the structural coupling of linear attention models, specifically the dependency between the projection matrices of queries and keys and the depthwise convolutions.

SliceGPT (Ashkboos et al., 2024a) prunes a model's backbone. This affects the rows of $\mathbf{W_Q}$ and $\mathbf{W_K}$. In particular, the attention mechanism operates in the same space pre- and post-slicing. Our work can be considered complementary to this approach. By pruning columns of $\mathbf{W_Q}$ and $\mathbf{W_K}$, we explicitly reduce the dimension of the attention mechanism. ThinK (Xu et al., 2024b) prunes KV cache channels in softmax Transformers based on attention weight magnitudes. While conceptually related, ThinK operates on the KV cache at inference time and does not account for the structural coupling between projections that arises in linear attention.

Quantization Methods. Recent Transformer quantization approaches like QuaRot (Ashkboos et al., 2024b) and SpinQuant (Liu et al., 2024) apply orthogonal rotations to both the residual stream and the state-space to disperse outliers. While effective for softmax Transformers, these state-space rotations are incompatible with the causal convolutions typically employed in linear attention models. Recent works on quantizing linear attention variants, including Quamba (Xu et al., 2024a), MambaQuant (Dong et al., 2025), and SSDi8 (**?**) further demonstrate the community's interest in compressing the sequence mixer. Low-rank factorization methods such as FWSVD (Hsu et al., 2022) and ASVD (Yuan et al., 2023) approximate weight matrices but do not reduce the state dimension: the recurrence still operates in the original d_k-dimensional space, yielding no speedup in the sequence mixer. Our framework instead utilizes axis-aligned transformations, which are compatible with causal convolutions and directly reduce the state size.

C. Training Details

We use the *flame* (Zhang & Yang, 2025) library for recovery fine-tuning (RFT, (Ashkboos et al., 2024a)) and pre-training the 370M parameter models. The latter uses 10 billion tokens of Fineweb-Edu. The larger DeltaNet and Gated DeltaNet models are taken from fla-hub [1] and m-a-p [2], respectively.

We perform RFT on a single H100 GPU using LoRA (Hu et al., 2022) with rank $r = 16$ $\alpha = 32$, using $32k$ samples of Fineweb-Edu. During training, we use: a batch size of 16, training on sequences of length 2048. After warming up for 5% of the total steps, we decay the learning rate from 10^{-4} down to 10^{-5}. We furthermore unfreeze the causal

[1] https://huggingface.co/collections/fla-hub/deltanet

[2] https://huggingface.co/m-a-p/1.3B-100B-GatedDeltaNet-pure

convolutions, which make up just a fraction of the total parameters. As suggested by (Ma et al., 2023), we furthermore apply knowledge-distillation during RFT using the original model.

For the compression methods requiring a calibration set, we use 128 samples of Fineweb-Edu. *DRRQR* uses a random subsample of $5k$ keys and queries each.

D. Additional Material

Algorithm 1 describes a typical DeltaNet (Yang et al., 2024b) layer.

Algorithm 1 A Typical DeltaNet Layer

Input: Hidden states $\mathbf{X} \in \mathbb{R}^{T,d}$, Previous state $\mathbf{S}_0$
Parameters: $\mathbf{W}_q, \mathbf{W}_k \in \mathbb{R}^{d_k,h}$, $\mathbf{W}_v \in \mathbb{R}^{d_v,h}$, $\mathbf{W}_o \in \mathbb{R}^{h,d_v}$, $\mathbf{W}_\beta \in \mathbb{R}^{1,h}$.

// 1. Projections and Local Mixing (Short Convolutions)
$\mathbf{q} \leftarrow \mathrm{SiLU}(\mathrm{Conv1D}(\mathbf{X}\mathbf{W}_q))$
$\mathbf{k} \leftarrow \mathrm{SiLU}(\mathrm{Conv1D}(\mathbf{X}\mathbf{W}_k))$
$\mathbf{v} \leftarrow \mathrm{SiLU}(\mathrm{Conv1D}(\mathbf{X}\mathbf{W}_v))$
$\beta \leftarrow \sigma(\mathbf{X}\mathbf{W}_\beta)$

// 2. Normalization
$\mathbf{q} \leftarrow \mathbf{q}/\|\mathbf{q}\|_2$
$\mathbf{k} \leftarrow \mathbf{k}/\|\mathbf{k}\|_2$

// 3. Delta Rule
for $t = 1$ **to** T **do**
  $\mathbf{S}_t \leftarrow \mathbf{S}_{t-1}(\mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top) + \beta_t \mathbf{v}_t \mathbf{k}_t^\top$
  $\mathbf{o}_t \leftarrow \mathbf{S}_t \mathbf{q}_t$
end for
Concatenate heads to form $\mathbf{O} \in \mathbb{R}^{T,d}$

// 4. Output Projection
$\mathbf{O} \leftarrow \mathrm{RMSNorm}(\mathbf{O})$
$\mathbf{Y} \leftarrow \mathbf{O}\mathbf{W}_o$

Return: $\mathbf{Y}$

D.1. Structured Pruning from the Test-Time Regression Perspective

Linear attention, and specifically DeltaNet, are usually interpreted as performing gradient descent on a linear regression objective (Schlag et al., 2021; Yang et al., 2024b), where the hidden state $\mathbf{S}_t$ acts as the fast weight matrix trained to map keys $\mathbf{k}_t$ (inputs) to values $\mathbf{v}_t$ (targets). In this framework, our proposed structured pruning strategy admits another interpretation, functioning as a *feature selection* step applied to the input of the online learner. By restricting the input features to the subspace spanned by the transformation matrix $\mathbf{T}$, we effectively constrain the hypothesis class of the regression. This forces the fast weights to ignore the null space of $\mathbf{T}$, acting as a form of regularization.

D.2. Details on Rank Revealing QR (RRQR)

In this section, we detail the *Strong Rank-Revealing QR* (RRQR) algorithm (see Algorithm 2) proposed by (Gu & Eisenstat, 1996), which forms the basis of our *DRRQR* pruning method.

Mathematical Formulation. Let $\mathbf{M} \in \mathbb{R}^{m,n}$ be the input matrix of concatenated keys and queries (in the main text, $m = 2N$ and $n = d_k$). We seek a permutation $\mathbf{\Pi}$ and a target rank k (in our main text denoted as d'_k) such that the QR factorization

$$\mathbf{M}\mathbf{\Pi} = \mathbf{Q} \begin{pmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{0} & \mathbf{C}_k \end{pmatrix} \tag{6}$$

satisfies specific bounds on the singular values of the leading principal submatrix $\mathbf{A}_k \in \mathbb{R}^{k,k}$ and the trailing submatrix $\mathbf{C}_k \in \mathbb{R}^{(m-k),(n-k)}$. Specifically, a Strong RRQR factorization guarantees that $\sigma_{\min}(\mathbf{A}_k)$ is bounded away from zero and

$\|\mathbf{C}_k\|_2 = \sigma_{\max}(\mathbf{C}_k)$ is small. This implies that $\mathbf{A}_k$ is well-conditioned.

Geometric Intuition. The algorithm aims to select k columns that maximize the volume of the parallelotope formed by the selected column vectors. Since $|\det(\mathbf{A}_k)| = \prod_{i=1}^{k} \sigma_i(\mathbf{A}_k)$, maximizing the determinant pushes the smallest singular values upward, thereby minimizing the condition number $\kappa(\mathbf{A}_k)$.

Algorithm 2 Deep Rank Revealing QR (*DRRQR*). Adapted from Algorithm 4 of (Gu & Eisenstat, 1996).

Input: Activation Matrix $\mathbf{A} \in \mathbb{R}^{2N, d_k}$, Target Rank d'_k, Tolerance $f \geq 1$
Output: Selected Indices $\mathcal{I}$

// 1. Initialization: QR with Column Pivoting
$[\mathbf{Q}, \mathbf{R}, \mathbf{\Pi}] \leftarrow \mathrm{QRCP}(\mathbf{A})$
Partition $\mathbf{R} = \begin{pmatrix} \mathbf{A}_{d'_k} & \mathbf{B}_{d'_k} \\ \mathbf{0} & \mathbf{C}_{d'_k} \end{pmatrix}$, where $\mathbf{A}_{d'_k} \in \mathbb{R}^{d'_k, d'_k}$
Initialize $\omega_i(\mathbf{A}_{d'_k}) = 1/\|(\mathbf{A}_{d'_k}^{-1})_{i,:}\|_2$ and $\gamma_j(\mathbf{C}_{d'_k}) = \|(\mathbf{C}_{d'_k})_{:,j}\|_2$

// 2. Iterative Swapping for Stability
while True **do**
  Compute $\mathbf{U} = \mathbf{A}_{d'_k}^{-1}\mathbf{B}_{d'_k}$

  // Calculate swap gain metric ρ_{ij} for all pairs
  // Checks if U_{ij} is large or if residual γ_j is large relative to basis ω_i
  $\rho_{ij} \leftarrow \sqrt{|U_{ij}|^2 + (\gamma_j(\mathbf{C}_{d'_k})/\omega_i(\mathbf{A}_{d'_k}))^2}$

  Let $(i^*, j^*) = \mathrm{argmax}_{i,j}\, \rho_{ij}$
  if $\rho_{i^*j^*} \leq f$ **then**
    break // Strong RRQR condition met
  end if

  Swap column i^* of $\mathbf{A}_{d'_k}$ with column j^* of $\mathbf{C}_{d'_k}$
  Update $\mathbf{R}$, $\omega(\mathbf{A}_{d'_k})$, and $\gamma(\mathbf{C}_{d'_k})$
end while

$\mathcal{I} \leftarrow \mathbf{\Pi}[1 : d'_k]$
Return: $\mathcal{I}$

The Swap Criterion. Let $\mathbf{\Pi}$ be the current permutation column permutation matrix. To determine if swapping the i-th column of the basis (where $1 \leq i \leq k$) with the j-th column of the residual (where $1 \leq j \leq n-k$) improves the factorization, we analyze the ratio of the new determinant to the current determinant.

(Gu & Eisenstat, 1996) derive an efficiently computable metric for this ratio. Let $\mathbf{U} = \mathbf{A}_k^{-1}\mathbf{B}_k$. We define:

- $\gamma_j(\mathbf{C}_k) = \|(\mathbf{C}_k)_{:,j}\|_2$: The ℓ_2-norm of the j-th column of the residual block.
- $\omega_i(\mathbf{A}_k) = 1/\|(\mathbf{A}_k^{-1})_{i,:}\|_2$: The reciprocal of the ℓ_2-norm of the i-th row of the inverse basis.

By Lemma 3.1 of (Gu & Eisenstat, 1996), the potential gain ρ_{ij} from swapping basis column i with candidate column j is given by:

$$\rho_{ij} = \sqrt{|U_{ij}|^2 + \left(\frac{\gamma_j(\mathbf{C}_k)}{\omega_i(\mathbf{A}_k)}\right)^2}. \tag{7}$$

If $\rho_{ij} > f$ for a chosen tolerance factor $f \geq 1$, swapping these columns guarantees an increase in $|\det(\mathbf{A}_k)|$ by a factor of at least ρ_{ij}. The first term, $|U_{ij}|^2$, captures the linear dependence of the candidate vector on the current basis vector, while the second term captures the magnitude of the candidate relative to the stability of the basis vector.

Algorithm and Update Rules. The *DRRQR* procedure (Algorithm 2) proceeds as follows:

1. **Initialization:** Compute an initial factorization using standard QRCP. This provides a baseline $\mathbf{\Pi}$, $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$.
2. **Identification:** Search for a pair of indices (i, j) such that $\rho_{ij} > f$. Efficient search strategies maximize over j for fixed i, or simply identify the first valid pair.
3. **Update:** If a valid pair is found:
   (a) Permute columns to swap indices i and $j + k$.
   (b) Retriangularize the matrix $\mathbf{R}$ using Givens rotations to restore the upper-triangular structure of $\mathbf{A}_k$. This costs $O(k(n-k))$ operations rather than the $O(n^3)$ of a full factorization.
   (c) Update the auxiliary vectors $\omega(\mathbf{A}_k)$ and $\gamma(\mathbf{C}_k)$ using the formulas provided in Section 4 of (Gu & Eisenstat, 1996).
4. **Termination:** The process repeats until no pair (i, j) satisfies $\rho_{ij} > f$, ensuring the matrix satisfies the strong rank-revealing condition.

D.3. Derivation of Tighter Error Bounds

In this section, we derive tighter bounds for the readout error ratio. The proof of Theorem 2.3 relies on the loose upper bound $\|\mathbf{S}\boldsymbol{\epsilon}\|_2 \leq \|\mathbf{S}\|_F \|\boldsymbol{\epsilon}\|_2$.

However, if we allow using the condition number $\kappa(\mathbf{S}) = \sigma_1/\sigma_d$ as a measure for the anisotropy of the associative memory, it follows readily from standard perturbation theory that

$$\frac{1}{\kappa(\mathbf{S})} \leq \frac{\|\mathbf{o} - \mathbf{o}^*\|_2/\|\mathbf{o}^*\|_2}{\|\boldsymbol{\epsilon}\|_2/\|\mathbf{q}^*\|_2} \leq \kappa(\mathbf{S}). \tag{8}$$

Indeed, it holds that (Trefethen & Bau, 2022, Lecture 12) $\sigma_d\|\mathbf{x}\|_2 \leq \|\mathbf{S}\mathbf{x}\|_2 \leq \sigma_1\|\mathbf{x}\|_2$ for any vector $\mathbf{x}$. Applying these inequalities to $\mathbf{S}\boldsymbol{\epsilon}$ and $\mathbf{S}\mathbf{q}^*$ yields Equation (8). Similar to Corollary H.3, this allows deriving bounds on the expected error under isotropic Gaussian noise (assuming again $\|q^*\|_2 = 1$ for simplicity):

$$\frac{1}{\kappa(\mathbf{S})}\mu \leq \|\mathbf{o} - \mathbf{o}^*\|_2/\|\mathbf{o}^*\|_2 \leq \kappa(\mathbf{S})\mu.$$

D.4. Adapting Convolutions to General Rotations

Handling the more general case of (semi-) orthogonal, non-axis-aligned transformations is more intricate than the axis-aligned one (see Section 3.3). In particular, Proposition 3.2 does not hold anymore. If one wishes to employ general orthogonal transformations $\mathbf{T} \in O(d_k)$ (such as those derived from PCA) to prune the sequence mixer, the convolution layers must be adapted.

Depthwise convolutions operate independently on each channel. When the input space is rotated via $\mathbf{T}$, the original basis-aligned filters become misaligned with the new principal components. Towards maintaining learned structures after pruning, one must find new convolution kernels that best approximate the original dynamics.

D.5. Optimal Diagonal Adaptation

We formalize this adaptation as an optimization problem: finding the optimal diagonal (channel-wise) filters in the new basis that minimize the reconstruction error of the original convolution output.

Proposition D.1 (Optimal Diagonal Adaptation)**.** *Let $\mathbf{x}_t$ be the input signal and let $\mathbf{T} \in O(d)$ be an orthogonal matrix, yielding features $\tilde{\mathbf{x}}_t = \mathbf{T}\mathbf{x}_t$. Let $\mathbf{W} \in \mathbb{R}^{d,l}$ be the original learnable filters for d channels and kernel size l.*

The optimal diagonal per-channel weights $\mathbf{W}' \in \mathbb{R}^{d,l}$ that minimize the expected squared reconstruction error:

$$\min_{\mathbf{W}'} \mathbb{E}_{\mathbf{x}} \left[\|\mathbf{W}' * \tilde{\mathbf{x}} - \mathbf{T}(\mathbf{W} * \mathbf{x})\|_F^2 \right]$$

are given by the energy-weighted projection:

$$\mathbf{W}' = (\mathbf{T} \odot \mathbf{T})\mathbf{W},$$

where $*$ denotes the depthwise convolution and $\odot$ is the Hadamard (element-wise) product.

Proof. We seek to find new depthwise separable convolutions with filter weights $\mathbf{W}'$ that minimize the error between the rotated input convolved with the new weights and the rotated original output.

Recall that a depthwise convolution with filter matrix $\mathbf{W}'$ acting on input $\tilde{\mathbf{x}}$ can be written as:

$$\mathbf{W}' * \tilde{\mathbf{x}} = \sum_{j=0}^{l-1} \mathbf{W}'^{(j)} \tilde{\mathbf{x}}_{t-j},$$

where $\mathbf{W}'^{(j)} = \mathrm{diag}(\mathbf{w}'^{(j)})$ is the diagonal filter matrix at time lag j. Similarly, the rotated original output is:

$$\mathbf{T}(\mathbf{W} * \mathbf{x}) = \mathbf{T} \sum_{j=0}^{l-1} \mathbf{W}^{(j)} \mathbf{x}_{t-j} = \sum_{j=0}^{l-1} \mathbf{T}\mathbf{W}^{(j)}\mathbf{T}^\top \tilde{\mathbf{x}}_{t-j},$$

where we used $\mathbf{x} = \mathbf{T}^\top \tilde{\mathbf{x}}$.

The error term is then $\sum_{j=0}^{l-1} (\mathbf{W}'^{(j)} - \mathbf{T}\mathbf{W}^{(j)}\mathbf{T}^\top)\tilde{\mathbf{x}}_{t-j}$. Thus, the ideal filter in the new basis is the dense matrix $\mathbf{M}^{(j)} := \mathbf{T}\mathbf{W}^{(j)}\mathbf{T}^\top$. However, to maintain the efficiency of depthwise convolutions, we are constrained to approximate this dense matrix with a diagonal matrix $\mathbf{W}'^{(j)}$.

Let us focus on the error for a specific lag j (omitting j for brevity) and channel k. The error vector is $\mathbf{e} = (\mathbf{W}' - \mathbf{M})\tilde{\mathbf{x}}$. The k-th component is:

$$e_k = (W'_{kk} - M_{kk})\tilde{x}_k - \sum_{n \neq k} M_{kn} \tilde{x}_n.$$

Squaring and taking the expectation, assuming the features in the rotated basis $\tilde{\mathbf{x}}$ are decorrelated (which is true if $\mathbf{T}$ is the PCA transformation matrix) such that $\mathbb{E}[\tilde{x}_k \tilde{x}_n] = 0$ for $n \neq k$:

$$\mathbb{E}[e_k^2] = (W'_{kk} - M_{kk})^2 \mathbb{E}[\tilde{x}_k^2] + \sum_{n \neq k} M_{kn}^2 \mathbb{E}[\tilde{x}_n^2].$$

To minimize this error with respect to the diagonal weight W'_{kk}, we must set the first term to zero:

$$W'_{kk} = M_{kk} = (\mathbf{T}\,\mathrm{diag}(\mathbf{w})\mathbf{T}^\top)_{kk}.$$

Expanding this matrix multiplication:

$$W'_{kk} = \sum_{m=1}^{d} T_{km} w_m T_{km} = \sum_{m=1}^{d} (T_{km})^2 w_m = ((\mathbf{T} \odot \mathbf{T})\mathbf{w})_k.$$

Extending this to all channels and lags, we obtain the matrix form $\mathbf{W}' = (\mathbf{T} \odot \mathbf{T})\mathbf{W}$. □

Intuitively, the new kernel for a principal component is a weighted average of the original kernels, weighted by the energy (squared contribution) each original dimension contributes to that component.

D.6. Shared Convolutions

An alternative approach to facilitate general rotations is to constrain the model architecture itself. If we enforce that the convolution filters are shared across all channels within a head, the convolution operation becomes a scalar multiplication at each lag, which commutes with any linear transformation.

Lemma D.2 (Commutativity of Shared Convolutions)**.** *Let $\mathbf{w} \in \mathbb{R}^l$ be a filter shared across all d channels, such that the convolution kernel matrix $\mathbf{W} \in \mathbb{R}^{d,l}$ has identical rows $\mathbf{W}_{k,:} = \mathbf{w}$ for all k. For any linear transformation matrix $\mathbf{T} \in \mathbb{R}^{d,d}$ (including orthogonal rotations), the convolution commutes with the transformation:*

$$\mathbf{W} * (\mathbf{T}\mathbf{x}) = \mathbf{T}(\mathbf{W} * \mathbf{x}).$$

Proof. For a shared filter, the convolution operation on the vector $\mathbf{x}_t$ can be written as a scalar convolution applied element-wise: $(\mathbf{W} * \mathbf{x})_t = \sum_{j=0}^{l-1} w_j \mathbf{x}_{t-j}$. Applying the transformation $\mathbf{T}$ first:

$$\mathbf{W} * (\mathbf{T}\mathbf{x})_t = \sum_{j=0}^{l-1} w_j (\mathbf{T}\mathbf{x}_{t-j}) = \mathbf{T}\left(\sum_{j=0}^{l-1} w_j \mathbf{x}_{t-j}\right) = \mathbf{T}(\mathbf{W} * \mathbf{x})_t.$$

□

This lemma implies that for models trained with shared convolutions, the optimal filter in the rotated basis $\mathbf{W}'$ is identical to the original filter $\mathbf{W}$. This architectural choice would render the model naturally robust to basis changes, enabling rotation-based pruning methods like PCA-based truncation without the need for filter adaptation or approximation errors.

E. Additional Experimental Results

E.1. Per-Head Rank Utilization

We compute the rank utilization of each (layer, head) pair for DeltaNet 370M at 75% compression (pre-RFT), averaged over the sequence- and batch-axis for a sample from Fineweb-Edu. Table 3 summarizes the results across all (layer, head) pairs. Three findings emerge. First, heads are heterogeneous: rank utilization varies by an order of magnitude across heads (0.03 to 0.40 post-pruning vs. 0.01 to 0.16 at baseline), suggesting that the model allocates memory capacity unevenly. Second, DRRQR lifts rank utilization consistently across heads ($\rho = 0.96$), while magnitude-based methods such as L1 exhibit weaker correlation with the baseline per-head structure ($\rho = 0.68$). Third, every method increases rank utilization at the reduced capacity, confirming the theoretical prediction of Proposition 2.1.

Table 3. Per-head rank utilization statistics across all (layer, head) pairs for DeltaNet 370M at 75% compression (pre-RFT). ρ denotes the Pearson correlation between baseline and post-pruning per-head rank utilization.

Method	Mean	Std	Min	Max	ρ
Baseline ($d_k = 128$)	0.053	0.028	0.01	0.16	1.00
DRRQR ($d_k = 32$)	0.170	0.060	0.03	0.40	0.96
L1 ($d_k = 32$)	0.124	0.036	0.03	0.30	0.68

The strong heterogeneity suggests a natural extension: per-head adaptive compression ratios, where low-utilization heads are pruned more aggressively and high-utilization heads are preserved. We allocate per-head budgets $d'_k(\ell, h)$ proportionally to baseline rank utilization, subject to the constraint that the total budget matches uniform compression ($\sum_{\ell,h} d'_k(\ell, h) = L \cdot H \cdot d'_k$), and apply DRRQR independently per head. On DeltaNet 370M at 75% compression ($d'_k = 32$), adaptive allocation achieves a Wikitext perplexity of 41.8 and Lambada perplexity of 68.0, improving over standard DRRQR (42.0 and 75.1, respectively). This confirms that the per-head heterogeneity is actionable and suggests a promising direction for future work.

E.2. Generalization to Mamba-2

To assess the generality of our framework beyond DeltaNet, we apply our pruning methods to a Mamba-2 370M model (Dao & Gu, 2024b), reducing the state dimension from 128 to 32 (75% compression, pre-RFT). In the Mamba-2 recursion, $\mathbf{B}$

and $\mathbf{C}$ take the role of keys and queries, respectively. Table 4 reports the results. The two methods identified as strong for (Gated) DeltaNet—*Grad* and *DRRQR*—also perform best here, confirming that the rank-based perspective generalizes. These results are pre-RFT; based on our DeltaNet experiments, recovery fine-tuning would be expected to further close the gap to the baseline.

Table 4. Mamba-2 370M at 75% state reduction (128 → 32). Zero-shot accuracy and Wikitext perplexity (pre-RFT).

METHOD	ARC-E	ARC-C	HELLAS	WINOG	PIQA	WIKI ↓
BASELINE	49.8	27.1	40.2	52.4	65.7	29.2
GRAD	47.0	26.7	36.8	50.6	64.1	44.6
DRRQR	44.2	26.1	36.1	51.0	63.2	47.3
S-WANDA	41.7	25.3	35.4	53.0	62.2	52.8
L1	40.3	24.9	34.6	51.7	62.0	54.3
RAND	41.0	25.0	34.7	50.6	61.5	52.8

E.3. Singular Value Spectrum

Figure 4 illustrates the singular value spectrum of a randomly selected head's hidden state, aggregated across all tokens. We compare the uncompressed baseline against *DRRQR*, *Grad* and *L1* methods at a 75% compression ratio (prior to recovery fine-tuning). Notably, while the uncompressed model exhibits a heavy tail of singular values, the compressed models display a much sharper spectral truncation. For some heads, we observe the emergence of a spectral gap in certain heads. This gap can also develop during recovery fine-tuning (see Figure 5). We consider studying this phenomenon an interesting branch for future research.

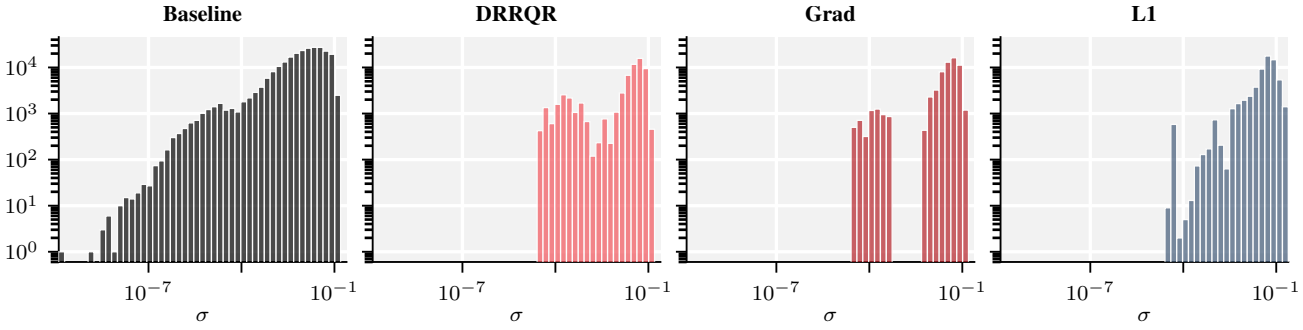

Figure 4. Singular value spectrum of a DeltaNet 370M head's hidden state (Fineweb-Edu, $T = 2048$, first 128 tokens skipped). We compare the uncompressed *Baseline* against compressed models at a 75% compression ratio (pre-RFT).

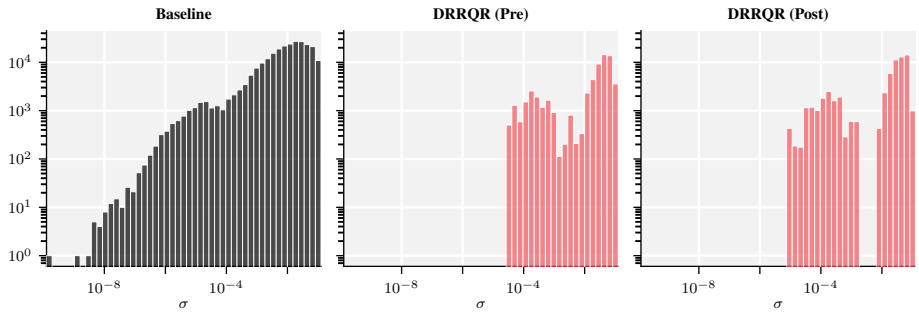

Figure 5. Impact of Recovery Fine-Tuning (RFT) on the singular value spectrum ($T = 2048$, first 128 tokens skipped). We compare the uncompressed *Baseline* with *DRRQR* (75% compression) both before (*pre*) and after (*post*) RFT.

E.4. On PCA and Convolutions

In this section, we provide extended experimental results on the PCA-based pruning strategy. We furthermore analyze the impact of depthwise convolutions on the applicability and performance of semi-orthogonal structured pruning.

As mentioned in Section 3, depthwise convolutions are generally not invariant under orthogonal transformations. When pruning via PCA, the features are rotated, causing a misalignment with the per-channel convolution filters. In Appendix D.4, we lay out a framework to fix this misalignment, either by introducing *shared convolutions* or by *mixing filters* (see Proposition D.1).

E.4.1. Shared Convolutions

To understand the impact of those two approaches, we train DeltaNet 370M variants using *Shared Convolutions*, where the convolution filter is tied across all channels within a head. As shown in Lemma D.2, this architecture is equivariant under rotations.

Table 5. DeltaNet 370M models trained on $10B$ tokens evaluated on common sense zero-shot reasoning tasks. "Shared Conv" indicates whether convolution filters are shared across heads and/or channels (for example, $\times\checkmark$ means that filters are shared across channels inside of a head but not across heads).

Shared Conv	Wiki. ppl ↓	LMB. ppl ↓	ARC-e acc_n ↑	ARC-c acc_n ↑	Hella. acc_n ↑	Wino. acc ↑	PIQA acc_n ↑	LMB. acc ↑	Avg ↑
$\times\times$	29.8	37.0	51.1	27.6	38.1	52.2	65.0	31.3	44.2
$\times\checkmark$	29.5	40.4	49.7	27.1	38.3	52.4	64.7	30.4	43.8
$\checkmark\checkmark$	29.3	40.4	49.9	26.4	37.8	49.9	65.2	29.9	43.2

Filter Similarity. Figure 6 visualizes the learned filters of a standard DeltaNet (non-shared). We observe high similarity between filters within specific heads, suggesting that the model naturally learns to share dynamics across channels. This could serve as a justification for sharing filters among channels inside each head.

Performance of Shared Convolutions. We include results on pre-trained DeltaNet models with and without shared convolutions in Table 5. It shows that, while models with shared convolutions are competitive (especially when just shared inside of a head and not across heads), there is a slight drop-off.

Table 6. Comparison of post-compression (pre-RFT) perplexity on Wikitext-2 for DeltaNet 370M. We compare standard (Non-Shared) vs. Shared Convolutions under different pruning strategies. "PCA (adv.)" removes the highest variance components (keeping noise), while "PCA (prop.)" removes the lowest variance components (keeping signal).

Comp.	Method	Non Shared	Shared
75%	PCA (adv.)	$3.1 \cdot 10^5$	$2.1 \cdot 10^5$
	PCA (prop.)	$3.3 \cdot 10^5$	184.48
	Grad	46.7	69.78
50%	PCA (adv.)	$2.4 \cdot 10^5$	$6.1 \cdot 10^4$
	PCA (prop.)	$3.2 \cdot 10^5$	155.76
	Grad	31.75	33.73
0%	–	29.83	29.55

E.4.2. Impact of Convolutions on PCA-based Pruning

Towards quantifying the two approaches of handling the per-channel convolutions, we compare a "Proper PCA" method, where we retain the principal components with the highest variance, against an "Adversarial PCA" baseline, where we deliberately retain the dimensions with the lowest variance. In a system robust to rotation, proper PCA should outperform the adversarial baseline.

Table 6 presents the perplexity on Wikitext-2 for DeltaNet 370M (pre-RFT). For the shared convolutions, proper PCA generally yields way better perplexity than the adversarial baseline. However, in the standard, non-shared case, the filter averaging does not suffice to make up for the misalignment post-transformation. Furthermore, the *Grad* pruning methods still outperforms PCA even when using shared convolutions.

Table 7. Consolidated ablation study on DeltaNet and Gated DeltaNet models (370M and 1.3B) measuring Wikitext-2 perplexity. We compare the impact of picking just keys ($\mathbf{K}$), just queries ($\mathbf{Q}$), or both ($\mathbf{K}$, $\mathbf{Q}$) for feature selection at a 50% compression ratio.

	DeltaNet						**Gated DeltaNet**					
	370M			**1.3B**			**370M**			**1.3B**		
Method	$\mathbf{K}, \mathbf{Q}$	$\mathbf{K}$	$\mathbf{Q}$	$\mathbf{K}, \mathbf{Q}$	$\mathbf{K}$	$\mathbf{Q}$	$\mathbf{K}, \mathbf{Q}$	$\mathbf{K}$	$\mathbf{Q}$	$\mathbf{K}, \mathbf{Q}$	$\mathbf{K}$	$\mathbf{Q}$
L1	3843.5	425218.3	33.0	66.1	1596.3	34.9	41.8	61.0	49.9	26.5	29.75	24.5
DRRQR	31.4	286035.7	32.1	20.5	566.9	22.9	31.6	56.9	43.5	17.3	21.9	22.0
Grad	31.7	17709.6	31.9	18.3	20.3	19.5	33.3	36.2	34.6	17.4	18.3	17.7
S-Wanda	915.9	376074.0	33.0	60.0	1604.4	29.1	40.0	61.3	43.5	21.9	22.5	21.5
Baseline		*29.8*			*16.7*			*28.8*			*16.8*	

E.5. On Coupled Selection

We compare selecting indices based on (i) the sum of scores derived from both projections ($\mathbf{K}, \mathbf{Q}$), (ii) keys only ($\mathbf{K}$), and (iii) queries only ($\mathbf{Q}$). Results are reported in Table 7.

We observe that selecting columns based on query projections ($\mathbf{Q}$) consistently outperforms selection based solely on keys ($\mathbf{K}$) for magnitude-based methods. Since queries govern readout, pruning based on $\mathbf{Q}$ ensures we discard dimensions with minimal contribution to the output. In contrast, pruning based on $\mathbf{K}$ risks removing information that the model attempts to access with a strong query, leading to significant readout errors.

Curiously, magnitude-based methods seem to perform better when selecting just based on queries ($\mathbf{Q}$) than when selecting based on both keys and queries ($\mathbf{K}, \mathbf{Q}$). Since they add the scores of keys and queries ($s_j = \|\mathbf{W}_{k,:j}\| + \|\mathbf{W}_{q,:j}\|$), this suggests the model contains large key weights whose corresponding query weights have lower magnitude.

DRRQR avoids this by targeting the effective rank of the joint subspace.

The results in Table 7 reveal a clear difference between the studied pruning methods. Magnitude-based methods (*L1*, *Wanda*) are unstable when targeting keys, performing best when restricted to queries. In contrast, optimization-based methods (*Grad*, *DRRQR*) consistently achieve the lowest perplexity using joint selection ($\mathbf{K}, \mathbf{Q}$). This indicates that the associative memory's effective rank requires accounting for the coupled interaction between keys and queries, rather than treating them in isolation.

F. The Effective Rank

In this section we present some properties of the effective rank. For more details, please refer to (Ipsen & Saibaba, 2025).

Definition F.1 (effective rank)**.** For a non-zero matrix $\mathbf{A} \in \mathbb{R}^{m,n}$, the effective rank is defined as:

$$\mathrm{er}(\mathbf{A}) \coloneqq \frac{\|\mathbf{A}\|_F^2}{\|\mathbf{A}\|_2^2} = \frac{\sum_i \sigma_i^2(\mathbf{A})}{\sigma_{\max}^2(\mathbf{A})}.$$

Unlike the algebraic rank, which is discontinuous, the effective rank is a continuous function of the matrix entries. This implies that small perturbations to the memory state $\mathbf{S}_t$ (e.g., from gradient noise or quantization) result in bounded changes to $\mathrm{er}(\mathbf{S}_t)$.

Proposition F.2 (Invariance under Transposition)**.** *The effective rank is invariant under transposition. For any matrix* $\mathbf{A}$*:*

$$\mathrm{er}(\mathbf{A}) = \mathrm{er}(\mathbf{A}^\top).$$

Proposition F.3 (Invariance under Unitary Transformations and Scaling)**.** *The effective rank is invariant under unitary transformations and scalar multiplication. For any unitary matrices* $\mathbf{U}, \mathbf{V}$ *and non-zero scalar* $c \in \mathbb{R}$*:*

$$\mathrm{er}(c\mathbf{U}\mathbf{A}\mathbf{V}^\top) = \mathrm{er}(\mathbf{A}).$$

Proposition F.4 (Bounds and Relation to Algebraic Rank)**.** *The effective rank is bounded by the algebraic rank:*

$$1 \leq \mathrm{er}(\mathbf{A}) \leq \mathrm{rank}(\mathbf{A}) \leq \min(m, n).$$

The lower bound is achieved if and only if $\mathbf{A}$ *has rank 1. The upper bound is achieved if and only if all non-zero singular values are equal.*

Proposition F.5 (Relation to Condition Number)**.** *Let* $\mathbf{A}$ *be a (non-zero) matrix. Then*

$$\kappa^2(\mathbf{A}) \geq \frac{\mathrm{rank}(\mathbf{A})}{\mathrm{er}(\mathbf{A})}.$$

In the specific case where $\mathbf{A}$ *has full rank, this implies*

$$\kappa^2(\mathbf{A}) \geq \frac{1}{\mathrm{u}(\mathbf{A})},$$

where u *is the rank utilization (see Definition 2.2).*

Edge Case (Isotropy): In the specific case where the matrix is perfectly conditioned on its support (i.e., $\kappa(\mathbf{A}) = 1$), the inequality becomes an equality:

$$1 \geq \frac{r}{\mathrm{er}(\mathbf{A})} \implies \mathrm{er}(\mathbf{A}) \geq r.$$

Since we known $\mathrm{er}(\mathbf{A}) \leq r$, this forces $\mathrm{er}(\mathbf{A}) = \mathrm{rank}(\mathbf{A})$. This confirms that for isotropic matrices (where all non-zero singular values are equal), the effective rank and algebraic rank coincide. Conversely, a large gap between r and $\mathrm{er}(\mathbf{A})$ is a sufficient condition for ill-conditioning.

G. Dynamical Systems Perspective

In this section, we derive the DeltaNet recurrence rule from a continuous-time dynamical systems perspective. We show that the sequence mixer can be interpreted as a discretization of a continuous gradient flow minimizing a linear regression objective.

Continuous-Time Dynamics. Consider a time-continuous associative memory $\mathbf{S}(t) \in \mathbb{R}^{d_v, d_k}$ receiving a stream of keys $\mathbf{k}(t) \in \mathbb{R}^{d_k}$ and values $\mathbf{v}(t) \in \mathbb{R}^{d_v}$. We define the instantaneous regression loss at time t as:

$$\mathcal{L}(\mathbf{S}(t)) = \frac{1}{2}\|\mathbf{v}(t) - \mathbf{S}(t)\mathbf{k}(t)\|_2^2.$$

The dynamics of the state $\mathbf{S}(t)$ are governed by the gradient flow minimizing this objective:

$$\dot{\mathbf{S}}(t) = -\nabla_{\mathbf{S}}\mathcal{L}(\mathbf{S}(t)) \tag{9}$$

$$= -(\mathbf{S}(t)\mathbf{k}(t) - \mathbf{v}(t))\mathbf{k}(t)^\top \tag{10}$$

$$= \mathbf{v}(t)\mathbf{k}(t)^\top - \mathbf{S}(t)\mathbf{k}(t)\mathbf{k}(t)^\top. \tag{11}$$

Equation (11) represents a linear time-varying (LTV) ordinary differential equation (ODE) of the form $\dot{\mathbf{S}}(t) = \mathbf{S}(t)\mathbf{A}(t) + \mathbf{B}(t)$, where $\mathbf{A}(t) = -\mathbf{k}(t)\mathbf{k}(t)^\top$ is the state-transition matrix acting on the right, and $\mathbf{B}(t) = \mathbf{v}(t)\mathbf{k}(t)^\top$ is the input forcing term.

Euler Discretization. To obtain the discrete-time update rule employed by DeltaNet, we apply the forward Euler method to Equation (11). Let Δt be the step size, which corresponds to the gating factor β_t in the DeltaNet formulation. The discretization yields:

$$\frac{\mathbf{S}_t - \mathbf{S}_{t-1}}{\beta_t} \approx \mathbf{v}_t\mathbf{k}_t^\top - \mathbf{S}_{t-1}\mathbf{k}_t\mathbf{k}_t^\top.$$

Rearranging the terms to solve for the next state $\mathbf{S}_t$:

$$\begin{aligned}\mathbf{S}_t &= \mathbf{S}_{t-1} + \beta_t(\mathbf{v}_t\mathbf{k}_t^\top - \mathbf{S}_{t-1}\mathbf{k}_t\mathbf{k}_t^\top) \\ &= \mathbf{S}_{t-1}(\mathbf{I} - \beta_t\mathbf{k}_t\mathbf{k}_t^\top) + \beta_t\mathbf{v}_t\mathbf{k}_t^\top.\end{aligned}$$

This exactly recovers the DeltaNet update rule (Equation (2)).

System Stability. The stability of this dynamical system is determined by the spectral properties of the transition matrix $(\mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top)$. For the system to be stable (non-divergent), the eigenvalues of this operator must lie within the unit circle. This implies the condition $|1 - \beta_t \|\mathbf{k}_t\|_2^2| \leq 1$. In standard DeltaNet implementations, keys are normalized ($\|\mathbf{k}_t\|_2 = 1$), and β_t is the output of a sigmoid function ($\beta_t \in (0, 1)$), strictly satisfying the stability condition and ensuring the memory decays appropriately over time.

H. Proofs

In this section, we provide proofs of statements presented in the main paper.

H.1. Proof of Proposition 2.1

We consider the matrix form of the associative memory $\mathbf{S} = \mathbf{V}^\top \mathbf{K}$, where $\mathbf{V} \in \mathbb{R}^{T, d_v}$ and $\mathbf{K} \in \mathbb{R}^{T, d_k}$.

To handle potential misalignment between the subspaces of values and keys, we decompose the values $\mathbf{V}$ into two orthogonal components relative to the column space of the keys $\mathbf{K}$:

$$\mathbf{V} = \mathbf{V}_\parallel + \mathbf{V}_\perp,$$

where the columns of $\mathbf{V}_\parallel$ lie in $\mathrm{col}(\mathbf{K})$, and the columns of $\mathbf{V}_\perp$ are orthogonal to it. Consequently, $\mathbf{V}_\perp^\top \mathbf{K} = \mathbf{0}$, and the memory state simplifies to:

$$\mathbf{S} = (\mathbf{V}_\parallel + \mathbf{V}_\perp)^\top \mathbf{K} = \mathbf{V}_\parallel^\top \mathbf{K}.$$

We define the scalar quantity $\nu(\mathbf{V})$ appearing in the main text as the effective rank of the projected values:

$$\nu(\mathbf{V}) \coloneqq \mathrm{er}(\mathbf{V}_\parallel) = \frac{\|\mathbf{V}_\parallel\|_F^2}{\|\mathbf{V}_\parallel\|_2^2}.$$

Since the columns of $\mathbf{V}_\parallel$ lie entirely within the column space of $\mathbf{K}$, and assuming $\mathbf{K}$ has full column rank, the matrix multiplication acts as a bijection on the row space of $\mathbf{V}_\parallel^\top$. We now derive the lower bound for the effective rank $\mathrm{er}(\mathbf{S}) = \|\mathbf{S}\|_F^2 / \|\mathbf{S}\|_2^2$.

First, we bound the numerator (Frobenius norm) from below. We use the property $\|\mathbf{AB}\|_F \geq \|\mathbf{A}\|_F \sigma_{\min}(\mathbf{B})$, which holds strictly here because the rows of $\mathbf{V}_\parallel^\top$ align with the range of $\mathbf{K}$:

$$\|\mathbf{S}\|_F^2 = \|\mathbf{V}_\parallel^\top \mathbf{K}\|_F^2 \geq \|\mathbf{V}_\parallel\|_F^2 \sigma_{\min}^2(\mathbf{K}).$$

Next, we bound the denominator (Spectral norm) from above using the standard sub-multiplicative property $\|\mathbf{AB}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{B}\|_2$:

$$\|\mathbf{S}\|_2^2 = \|\mathbf{V}_\parallel^\top \mathbf{K}\|_2^2 \leq \|\mathbf{V}_\parallel\|_2^2 \|\mathbf{K}\|_2^2.$$

Combining these inequalities yields the lower bound:

$$\mathrm{er}(\mathbf{S}) = \frac{\|\mathbf{S}\|_F^2}{\|\mathbf{S}\|_2^2} \geq \frac{\|\mathbf{V}_\parallel\|_F^2 \sigma_{\min}^2(\mathbf{K})}{\|\mathbf{V}_\parallel\|_2^2 \|\mathbf{K}\|_2^2} = \mathrm{er}(\mathbf{V}_\parallel) \frac{1}{\kappa^2(\mathbf{K})} = \frac{\nu(\mathbf{V})}{\kappa^2(\mathbf{K})}.$$

□

H.2. Proof of Algebraic Rank

We first prove the following proposition:

Proposition H.1. *Let $\mathbf{S}_0 = 0$ and $\mathbf{S}_t$ be a matrix satisfying the recursion*

$$\mathbf{S}_t = \mathbf{S}_{t-1}(\alpha_t \mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top) + \gamma_t \mathbf{v}_t \mathbf{k}_t^\top$$

for some non-zero scalars $\alpha_t, \beta_t, \gamma_t$ and some vectors $\mathbf{v}_t$, $\mathbf{k}_t$. Then

$$\mathrm{row}\, \mathbf{S}_t \subseteq \mathrm{span}\langle \mathbf{k}_1, \ldots, \mathbf{k}_t \rangle \quad \textit{and} \quad \mathrm{col}\, \mathbf{S}_t \subseteq \mathrm{span}\langle \mathbf{v}_1, \ldots, \mathbf{v}_t \rangle.$$

Consequently,

$$\operatorname{rank} \mathbf{S}_t \leq \min\left(\operatorname{rank} \mathbf{K}_t, \operatorname{rank} \mathbf{V}_t\right)$$

where $\mathbf{K}_t = (\mathbf{k}_1, \ldots, \mathbf{k}_t)$ *and* $\mathbf{V}_t = (\mathbf{v}_1, \ldots, \mathbf{v}_t)$ *are the matrices obtained by stacking the* $\mathbf{k}$*- and* $\mathbf{v}$*-vectors, respectively.*

Proof. We start by showing the first claim by induction. For $t = 1$, we have $\mathbf{S}_1 = \gamma_1 \mathbf{v}_1 \mathbf{k}_1^\top$. Consider $t > 1$ and assume the claim is true for every $s < t$. Then

$$\begin{aligned}
\mathbf{S}_t &= \mathbf{S}_{t-1}(\alpha_t \mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top) + \gamma_t \mathbf{v}_t \mathbf{k}_t^\top \\
&= \alpha_t \mathbf{S}_{t-1} + (\gamma_t \mathbf{v}_t - \beta_t \mathbf{S}_{t-1} \mathbf{k}_t) \mathbf{k}_t^\top \\
&= \alpha_t \mathbf{S}_{t-1} + \mathbf{u}_t \mathbf{k}_t^\top,
\end{aligned}$$

where we defined $\mathbf{u}_t \coloneqq \gamma_t \mathbf{v}_t - \beta_t \mathbf{S}_{t-1} \mathbf{k}_t$. Next, we use that for two matrices $\mathbf{A}$ and $\mathbf{B}$,

$$\text{row}(\mathbf{A} + \mathbf{B}) \subseteq \text{span}(\text{row}(\mathbf{A}), \text{row}(\mathbf{B}))$$

and thus

$$\begin{aligned}
\text{row}(\alpha_t \mathbf{S}_{t-1} + \mathbf{u}_t \mathbf{k}_t^\top) &\subseteq \text{span}(\text{row}(\alpha_t \mathbf{S}_{t-1}), \text{row}(\mathbf{u}_t \mathbf{k}_t^\top)) \\
&= \text{span}(\text{row}(\mathbf{S}_{t-1}), \mathbf{k}_t) \\
&\subseteq \text{span}(\mathbf{k}_1, ..., \mathbf{k}_t).
\end{aligned}$$

This concludes the first claim.

For the second claim, we proceed analogously.

Indeed, we again show this claim by induction. The case $t = 1$ is clear. For any $t > 1$, we compute, using $\text{col}(\mathbf{AB}) \subseteq \text{col}(\mathbf{A})$,

$$\begin{aligned}
\text{col}(\mathbf{S}_t) &= \text{col}(\mathbf{S}_{t-1}(\alpha_t \mathbf{I} - \beta_t \mathbf{k}_t \mathbf{k}_t^\top), \gamma_t \mathbf{v}_t \mathbf{k}_t^\top) \\
&\subseteq \text{col}(\mathbf{S}_{t-1}, \gamma_t \mathbf{v}_t \mathbf{k}_t^\top) \\
&\subseteq \text{col}(\mathbf{v}_1, \ldots, \mathbf{v}_t).
\end{aligned}$$

This concludes the proof. □

Interestingly, PCA-based transformations are guaranteed to not decrease the rank of the keys:

Lemma H.2 (Monotonicity of Rank Utilization)**.** *Pruning the low-variance directions via PCA strictly increases (or maintains) rank utilization. That is, for any* $d_k' < d_k$*:*

$$u(\mathbf{K}') \geq u(\mathbf{K}).$$

Proof. Assume towards a contradiction that the utilization decreases, i.e., $u(\mathbf{K}') < u(\mathbf{K})$. This means that

$$\frac{1}{d_k'} \sum_{i=1}^{d_k'} \sigma_i^2 < \frac{1}{d_k} \sum_{i=1}^{d_k} \sigma_i^2.$$

In words, the average energy of the top d_k' principal components is strictly less than the average energy of the full spectrum. This is a contradiction, as the singular values are non-increasing ($\sigma_1 \geq \cdots \geq \sigma_{d_k}$). Thus, our assumption must have been wrong. □

H.3. Proof of Theorem 2.3

We analyze the error amplification ratio R, defined as the relative output error divided by the input noise-to-signal ratio:

$$R = \frac{\|\mathbf{o} - \mathbf{o}^*\|_2 / \|\mathbf{o}^*\|_2}{\|\boldsymbol{\epsilon}\|_2 / \|\mathbf{q}^*\|_2} = \frac{\|\mathbf{S}\boldsymbol{\epsilon}\|_2}{\|\boldsymbol{\epsilon}\|_2} \cdot \frac{\|\mathbf{q}^*\|_2}{\|\mathbf{S}\mathbf{q}^*\|_2}.$$

This expression represents the Rayleigh quotient of the noise divided by the Rayleigh quotient of the signal.

We start by showing the lower bound. To find the minimum error, we start by projecting the response to the noise onto $\mathbf{u}_1$:

$$\|\mathbf{S}\boldsymbol{\epsilon}\|_2 \geq |\mathbf{u}_1^\top \mathbf{S}\boldsymbol{\epsilon}| = \sigma_1 |\mathbf{w}_1^\top \boldsymbol{\epsilon}| = \sigma_1 \delta \|\boldsymbol{\epsilon}\|_2.$$

Next, we upper bound the response to the true signal, using Cauchy-Schwarz:

$$\|\mathbf{S}\mathbf{q}^*\|_2 \leq \|\mathbf{S}\|_F \|\mathbf{q}^*\|_2.$$

Substituting these into R:

$$R \geq \frac{\sigma_1 \delta \|\boldsymbol{\epsilon}\|_2}{\|\boldsymbol{\epsilon}\|_2} \cdot \frac{\|\mathbf{q}^*\|_2}{\|\mathbf{S}\|_F \|\mathbf{q}^*\|_2} = \delta \frac{\sigma_1}{\|\mathbf{S}\|_F} = \delta \frac{1}{\sqrt{\mathrm{er}(\mathbf{S})}}.$$

Using the definition $\mathrm{er}(\mathbf{S}) = d \cdot u(\mathbf{S})$, we obtain the lower bound:

$$R \geq \frac{\delta}{\sqrt{d \cdot u(\mathbf{S})}}.$$

Next, we show the upper bound. To find the maximum error, we first upper bound the response of the system to the noise:

$$\|\mathbf{S}\boldsymbol{\epsilon}\|_2 \leq \|\mathbf{S}\|_F \|\boldsymbol{\epsilon}\|_2.$$

Next, we lower bound the response to the true query by projecting onto $\mathbf{u}_1$:

$$\|\mathbf{S}\mathbf{q}^*\|_2 \geq |\mathbf{u}_1^\top \mathbf{S}\mathbf{q}^*| = \sigma_1 |\mathbf{w}_1^\top \mathbf{q}^*| = \sigma_1 \gamma \|\mathbf{q}^*\|_2.$$

Again, substituting these into R:

$$R \leq \frac{\|\mathbf{S}\|_F \|\boldsymbol{\epsilon}\|_2}{\|\boldsymbol{\epsilon}\|_2} \cdot \frac{\|\mathbf{q}^*\|_2}{\sigma_1 \gamma \|\mathbf{q}^*\|_2} = \frac{1}{\gamma} \frac{\|\mathbf{S}\|_F}{\sigma_1} = \frac{\sqrt{\mathrm{er}(\mathbf{S})}}{\gamma}.$$

Using $\mathrm{er}(\mathbf{S}) = d \cdot u(\mathbf{S})$, we obtain the upper bound:

$$R \leq \frac{\sqrt{d \cdot u(\mathbf{S})}}{\gamma}.$$

□

H.4. Expected Error Bounds

Corollary H.3 (Expected Error Bounds)**.** *Assume the noise is isotropic Gaussian, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \xi^2 \mathbf{I})$. Assume furthermore, for simplicity, that $\|q^*\|_2 = 1$. Then*

$$\sqrt{\frac{2}{\pi \,\mathrm{er}(\mathbf{S})}} \xi \leq \mathbb{E}\left[\frac{\|\mathbf{o} - \mathbf{o}^*\|_2}{\|\mathbf{o}^*\|_2}\right] \leq \frac{\sqrt{\mathrm{er}(\mathbf{S})}}{\gamma} \xi \mu,$$

where $\mu \coloneqq \sqrt{2}\Gamma(\frac{d+1}{2})/\Gamma(\frac{d}{2})$.

Proof. Follows by computing the bounds derived in Theorem 2.3. The noise follows $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \xi^2 \mathbf{I})$. Note that the expected norm of an isotropic Gaussian satisfies

$$\mu = \mathbb{E}[\|\boldsymbol{\epsilon}\|_2] = \xi \sqrt{2} \frac{\Gamma(\frac{d+1}{2})}{\Gamma(\frac{d}{2})}.$$

Furthermore, the expected alignment $z = \mathbf{w}_1^\top \boldsymbol{\epsilon}$ of an isotropic Gaussian $\boldsymbol{\epsilon}$ with a vector $\mathbf{w}_1$ can be computed as

$$\mathbb{E}[\delta \|\boldsymbol{\epsilon}\|_2] = \mathbb{E}[|z|] = \xi \sqrt{\frac{2}{\pi}}.$$

□

H.5. Calculus

We first need to show that

$$\partial_{\text{vec}(\mathbf{B})}\mathbf{ABx} = \mathbf{x}^\top \otimes \mathbf{A}$$

for matrices $\mathbf{A}, \mathbf{B}$ and a vector $\mathbf{x}$. But this follows immediately from the matrix identity (Petersen et al., 2008, Equation (520))

$$\mathbf{ABx} = (\mathbf{x}^\top \otimes \mathbf{A})\text{vec}(\mathbf{B})$$

and then taking the derivative.

Next, we need to show that

$$\kappa(\mathbf{x} \otimes \mathbf{A}) = \kappa(\mathbf{A}).$$

But this follows from Horn & Johnson (1994, Theorem 4.2.15) and writing κ as a fraction of singular values, so that

$$\kappa(\mathbf{x} \otimes \mathbf{A}) = \kappa(\mathbf{x})\kappa(\mathbf{A}) = \kappa(\mathbf{A}).$$

□

I. Extended Results

I.1. Language Modeling

This subsection contains the extensive zero-shot (Gao et al., 2024) task evaluations. Table 8 contains averaged results for all models at a fixed compression ratio of 50%.

Table 8. Comparison of compression methods at a 50% compression ratio across all model sizes, both pre- and post-RFT. We report Perplexity for WikiText (**Wiki**) and Lambada (**LMB**), as well as the average accuracy for LM Evals (**LM Evals**) tasks (best results in **bold**, second best in underlined).

		Pre-RFT			Post-RFT		
Model	Method	Wiki ↓	LMB ↓	LM Evals Avg ↑	Wiki ↓	LMB ↓	LM Evals Avg ↑
DeltaNet 370M	L1	3843.5	57304.8	33.0	32.5	56.8	43.0
	Rand	1032.6	2882.8	34.3	32.6	49.9	42.6
	DRRQR	**31.4**	36.0	44.7	**29.4**	36.6	**44.5**
	Grad	31.7	**33.3**	**44.9**	**29.4**	**36.3**	44.4
	S-Wanda	915.9	6118.0	35.1	32.1	53.2	43.4
	Baseline	29.8	37.0	44.2	29.8	37.0	44.2
DeltaNet 1.3B	L1	66.1	298.7	41.6	19.0	16.1	48.2
	Rand	41.2	68.0	44.4	19.4	14.4	48.2
	DRRQR	20.5	47.2	45.7	17.5	11.8	49.7
	Grad	**18.3**	**15.4**	**48.8**	**17.2**	**11.3**	**50.3**
	S-Wanda	60.0	361.3	40.8	18.5	15.9	48.3
	Baseline	16.7	11.9	50.0	16.7	11.9	50.0
Gated DeltaNet 370M	L1	41.8	156.6	40.4	29.2	44.6	43.5
	Rand	35.5	50.7	43.4	29.3	46.5	43.6
	DRRQR	**31.6**	39.4	**44.0**	**28.7**	40.6	**43.8**
	Grad	33.3	**39.2**	43.8	**28.7**	**39.4**	43.7
	S-Wanda	40.0	141.5	40.9	29.2	44.6	43.5
	Baseline	28.8	35.9	44.4	28.8	35.9	44.4
Gated DeltaNet 1.3B	L1	26.5	34.0	53.3	16.8	13.0	57.2
	Rand	18.8	14.4	56.5	16.9	11.5	57.8
	DRRQR	**17.3**	14.2	57.5	**16.3**	12.0	58.0
	Grad	17.4	**10.1**	**58.6**	16.4	**10.5**	**58.3**
	S-Wanda	21.9	24.1	55.1	16.6	11.7	57.8
	Baseline	16.8	9.7	59.4	16.8	9.7	59.4

Table 9. Zero-shot performance of DeltaNet 370M models, evaluated using the *lm-eval-harness* (Gao et al., 2024), given different compression ratios. Pre- and post RFT.

RFT	Method	Compr.	Wiki. ppl ↓	LMB. ppl ↓	ARC-e acc_n ↑	ARC-c acc_n ↑	Hella. acc_n ↑	Wino. acc ↑	PIQA acc_n ↑	LMB. acc ↑	Avg ↑
×	L1	75%	100083.0	9661610.6	27.2	28.0	27.7	52.2	50.9	0.0	31.0
×	Rand	75%	136955.3	5879873.3	27.0	27.4	26.6	48.9	51.4	0.0	30.2
×	DRRQR	75%	**42.0**	74.1	51.9	27.5	38.7	50.9	64.6	25.3	43.1
×	Grad	75%	46.7	**58.0**	52.0	27.0	38.5	49.8	64.7	29.1	**43.5**
×	S-Wanda	75%	33512.5	4955134.6	28.5	26.8	27.7	51.1	48.9	0.0	30
✓	L1	75%	36.3	146.7	46.0	26.2	36.1	51.0	63.0	17.1	39.9
✓	Rand	75%	40.3	129.9	46.8	26.1	35.8	51.9	62.8	17.7	40.2
✓	DRRQR	75%	**31.4**	43.7	52.0	27.5	38.1	52.4	64.6	29.7	44.0
✓	Grad	75%	31.5	**42.5**	52.1	28.0	38.0	51.6	64.8	30.2	**44.1**
✓	S-Wanda	75%	35.4	101.7	46.5	26.7	37.1	50.9	63.3	20.3	40.8
×	L1	50%	3843.5	57304.8	35.8	23.6	31.2	50.8	55.0	1.6	33.0
×	Rand	50%	1032.6	2882.8	37.8	23.7	30.3	50.5	56.6	6.9	34.3
×	DRRQR	50%	**31.4**	36.0	52.1	27.6	38.7	52.0	65.1	32.7	44.7
×	Grad	50%	31.7	**33.3**	51.8	27.9	38.7	52.2	65.5	33.4	**44.9**
×	S-Wanda	50%	915.9	6118.0	39.9	24.3	33.5	50.1	57.4	5.4	35.1
✓	L1	50%	32.5	56.8	49.2	27.0	37.7	53.0	64.6	26.4	43.0
✓	Rand	50%	32.6	49.9	49.2	27.9	37.6	49.0	64.9	27.3	42.6
✓	DRRQR	50%	**29.4**	36.6	51.8	27.7	38.1	52.1	65.3	31.7	**44.5**
✓	Grad	50%	**29.4**	**36.3**	51.0	27.7	38.1	52.1	65.5	32.0	44.4
✓	S-Wanda	50%	32.1	53.2	51.1	26.1	37.9	52.6	65.0	27.7	43.4
×	L1	40%	1232.0	5431.6	39.6	23.9	34.0	50.2	56.4	5.7	34.9
×	Rand	40%	121.3	329.1	42.8	25.4	34.2	49.8	61.5	15.0	38.1
×	DRRQR	40%	**30.2**	33.4	52.6	28.0	38.7	51.9	65.2	33.1	**44.9**
×	Grad	40%	30.4	**32.7**	52.2	27.7	38.7	51.8	65.3	33.4	**44.9**
×	S-Wanda	40%	276.8	603.4	45.3	25.4	35.8	50.3	60.3	13.4	38.4
✓	L1	40%	31.7	53.9	50.8	26.7	37.7	53.0	65.1	26.5	43.3
✓	Rand	40%	31.0	43.6	49.1	27.6	38.0	52.5	65.2	29.2	43.6
✓	DRRQR	40%	**29.0**	36.3	51.4	27.8	38.2	52.0	65.4	32.0	**44.5**
✓	Grad	40%	29.1	**35.7**	51.5	27.4	38.2	51.5	65.5	32.1	44.4
✓	S-Wanda	40%	31.2	48.3	50.7	26.9	38.0	52.4	64.8	28.8	43.6
×	L1	30%	270.1	307.0	45.6	24.7	36.2	50.2	60.1	18.5	39.2
×	Rand	30%	52.7	103.2	48.4	26.8	36.1	50.4	62.9	22.7	41.2
×	DRRQR	30%	**29.3**	**32.7**	52.4	27.7	38.6	52.7	65.0	33.5	**45.0**
×	Grad	30%	29.4	33.0	51.7	27.9	38.6	52.5	64.7	33.4	44.8
×	S-Wanda	30%	138.9	133.9	46.3	26.0	37.1	50.0	62.0	22.7	40.7
✓	L1	30%	30.8	50.9	51.1	27.0	37.9	53.6	64.7	27.4	43.6
✓	Rand	30%	30.0	40.7	50.1	27.6	38.1	52.2	65.5	30.3	44.0
✓	DRRQR	30%	**28.8**	**36.0**	51.3	27.6	38.2	52.1	65.1	32.3	44.4
✓	Grad	30%	**28.8**	**36.0**	51.3	27.7	38.1	51.9	65.5	32.3	**44.5**
✓	S-Wanda	30%	30.7	45.4	51.2	27.2	38.0	52.6	64.9	29.6	43.9
Baseline	–	0%	29.8	37.0	51.1	27.6	38.1	52.2	65.0	31.3	44.2

Table 10. Zero-shot performance of DeltaNet 1.3B models, evaluated using the *lm-eval-harness* (Gao et al., 2024), given different compression ratios. Pre- and post RFT.

RFT	Method	Compr.	Wiki. ppl ↓	LMB. ppl ↓	ARC-e acc_n ↑	ARC-c acc_n ↑	Hella. acc_n ↑	Wino. acc ↑	PIQA acc_n ↑	LMB. acc ↑	Avg ↑
×	L1	75%	2860.8	6728.9	36.0	26.4	41.5	51.9	62.4	2.6	36.8
×	Rand	75%	53664.2	3585.3	37.4	26.9	36.5	50.2	60.3	3.6	35.8
×	DRRQR	75%	43.9	375.5	43.6	27.7	49.1	51.3	69.5	9.9	41.8
×	Grad	75%	**27.0**	**61.5**	49.7	27.0	49.3	52.3	69.8	22.4	**45.1**
×	S-Wanda	75%	3160.0	6856.0	37.0	27.4	40.5	51.8	64.3	2.0	37.2
✓	L1	75%	20.2	26.6	48.1	25.8	47.3	51.1	68.7	33.1	45.7
✓	Rand	75%	22.9	22.9	47.8	27.1	46.5	51.5	69.7	35.3	46.3
✓	DRRQR	75%	19.2	17.5	50.5	27.3	48.7	53.5	69.6	40.5	48.4
✓	Grad	75%	**19.0**	**14.7**	50.7	26.8	49.1	52.4	69.9	43.6	**48.8**
✓	S-Wanda	75%	20.0	23.1	48.6	26.4	47.2	51.9	69.5	35.6	46.5
×	L1	50%	66.1	298.7	42.4	27.2	47.9	54.4	67.1	10.6	41.6
×	Rand	50%	41.2	68.0	46.9	27.8	47.9	53.7	69.5	20.4	44.4
×	DRRQR	50%	20.5	47.2	46.5	27.8	51.3	53.0	69.9	25.5	45.7
×	Grad	50%	**18.3**	**15.4**	48.8	27.5	50.9	53.4	70.2	42.2	**48.8**
×	S-Wanda	50%	60.0	361.3	40.8	27.0	47.1	53.7	66.4	9.9	40.8
✓	L1	50%	19.0	16.1	48.9	27.1	48.6	53.7	69.7	41.1	48.2
✓	Rand	50%	19.4	14.4	48.5	26.2	48.6	52.5	69.9	43.4	48.2
✓	DRRQR	50%	17.5	11.8	51.0	26.6	49.7	53.3	69.7	48.2	49.7
✓	Grad	50%	**17.2**	**11.3**	51.2	26.5	49.9	54.5	70.7	48.7	**50.3**
✓	S-Wanda	50%	18.5	15.9	49.7	27.5	48.5	52.0	70.2	41.5	48.3
×	L1	40%	45.2	194.4	44.1	27.3	48.2	53.8	68.5	13.6	42.6
×	Rand	40%	23.9	30.0	49.3	27.6	49.1	52.3	69.3	30.6	46.4
×	DRRQR	40%	19.3	37.8	47.8	28.3	52.0	54.4	69.5	28.4	46.7
×	Grad	40%	**17.9**	**13.0**	48.8	26.9	51.0	53.5	70.3	45.6	**49.4**
×	S-Wanda	40%	44.7	189.0	42.6	27.6	47.8	54.1	68.3	13.7	42.4
✓	L1	40%	18.5	13.7	49.8	27.1	48.8	52.2	69.8	44.6	48.7
✓	Rand	40%	18.6	13.0	49.6	26.4	48.9	52.8	70.2	44.8	48.8
✓	DRRQR	40%	**17.1**	11.1	51.5	27.0	49.9	53.9	69.7	49.2	50.2
✓	Grad	40%	**17.1**	**10.4**	50.8	26.8	50.1	54.5	70.7	50.8	**50.6**
✓	S-Wanda	40%	18.2	13.7	50.7	27.2	48.9	51.8	70.3	45.0	49.0
×	L1	30%	37.6	128.6	44.5	27.1	48.7	53.4	68.6	16.6	43.2
×	Rand	30%	19.8	22.9	48.9	27.3	49.3	53.7	69.6	36.7	47.6
×	DRRQR	30%	18.3	27.4	48.5	29.0	52.3	55.2	70.4	33.2	48.1
×	Grad	30%	**17.2**	**11.9**	50.3	26.8	50.8	54.6	70.8	47.5	**50.1**
×	S-Wanda	30%	36.9	119.0	44.4	26.1	48.6	52.6	68.7	17.4	43.0
✓	L1	30%	18.1	12.8	50.0	27.0	49.3	52.9	70.0	45.9	49.2
✓	Rand	30%	17.4	12.0	50.4	26.9	49.6	53.9	70.3	46.6	49.6
✓	DRRQR	30%	**16.8**	10.5	51.1	27.4	50.0	53.9	70.0	50.2	50.4
✓	Grad	30%	16.9	**10.1**	51.3	26.5	50.4	54.8	70.6	51.1	**50.8**
✓	S-Wanda	30%	18.0	13.0	50.5	27.0	49.2	54.0	70.3	45.4	49.4
Baseline	–	0%	16.7	11.9	51.3	26.1	50.6	53.3	70.5	48.4	50.0

Table 11. Zero-shot performance of Gated DeltaNet 370M models, evaluated using the *lm-eval-harness* (Gao et al., 2024), given different compression ratios. Pre- and post RFT.

RFT	Method	Compr.	Wiki. ppl ↓	LMB. ppl ↓	ARC-e acc_n ↑	ARC-c acc_n ↑	Hella. acc_n ↑	Wino. acc ↑	PIQA acc_n ↑	LMB. acc ↑	Avg ↑
×	L1	75%	116.4	1399.7	36.8	27.6	34.7	50.3	59.7	8.5	36.3
×	Rand	75%	135.3	1078.1	41.5	26.5	34.4	51.2	60.4	11.2	37.5
×	DRRQR	75%	**45.9**	**97.8**	49.7	27.8	38.4	50.1	63.7	23.8	**42.3**
×	Grad	75%	79.6	215.6	41.5	27.9	37.4	50.0	62.9	20.1	40.0
×	S-Wanda	75%	96.6	1205.5	37.2	26.5	35.5	50.4	60.6	8.8	36.5
✓	L1	75%	32.2	60.0	49.6	27.4	38.2	49.3	64.6	25.2	42.4
✓	Rand	75%	33.5	70.9	49.5	26.4	37.9	51.9	65.0	22.5	42.2
✓	DRRQR	75%	**31.4**	53.5	50.8	27.1	38.5	49.9	65.3	26.3	43.0
✓	Grad	75%	**31.4**	**49.1**	50.3	27.4	38.5	50.6	65.8	27.1	**43.3**
✓	S-Wanda	75%	32.1	58.2	50.0	27.0	38.0	49.3	64.5	25.6	42.4
×	L1	50%	41.8	156.6	45.6	28.3	38.8	47.8	64.0	17.9	40.4
×	Rand	50%	35.5	50.7	47.1	27.0	39.1	53.1	64.9	29.4	43.4
×	DRRQR	50%	**31.6**	39.4	49.8	27.9	39.9	49.5	65.2	31.5	**44.0**
×	Grad	50%	33.3	**39.2**	48.2	27.6	40.0	49.2	65.8	31.8	43.8
×	S-Wanda	50%	40.0	141.5	45.5	28.7	38.9	49.6	64.0	18.9	40.9
✓	L1	50%	29.2	44.6	50.2	27.8	39.0	49.6	65.3	29.0	43.5
✓	Rand	50%	29.3	46.5	50.3	26.6	38.9	51.9	66.1	28.2	43.6
✓	DRRQR	50%	**28.7**	40.6	50.9	27.5	39.2	50.4	65.3	29.8	**43.8**
✓	Grad	50%	**28.7**	**39.4**	50.3	27.4	39.0	50.6	65.7	29.6	43.7
✓	S-Wanda	50%	29.2	44.6	50.3	27.3	39.1	50.2	65.0	29.0	43.5
×	L1	40%	36.9	96.4	46.1	28.3	38.7	49.0	65.0	21.5	41.4
×	Rand	40%	32.2	42.2	48.8	27.3	39.2	53.0	65.3	30.5	44.0
×	DRRQR	40%	**30.1**	38.3	50.5	28.0	40.0	49.6	65.4	31.7	44.2
×	Grad	40%	30.7	**34.6**	49.5	27.5	40.2	50.4	65.8	33.0	**44.4**
×	S-Wanda	40%	35.9	95.8	46.7	28.0	39.2	50.2	64.1	22.1	41.7
✓	L1	40%	28.6	40.6	50.8	28.2	39.1	50.1	65.4	30.8	44.0
✓	Rand	40%	28.6	43.1	50.7	26.5	38.9	50.9	65.9	29.5	43.7
✓	DRRQR	40%	**28.2**	39.5	51.1	27.8	39.4	50.0	65.4	30.5	44.0
✓	Grad	40%	**28.2**	**38.0**	50.6	27.7	39.2	50.7	65.6	30.5	**44.1**
✓	S-Wanda	40%	38.5	40.6	50.9	27.8	38.8	49.6	65.2	30.8	43.9
×	L1	30%	33.9	65.1	47.7	28.2	39.4	49.3	64.9	25.1	42.4
×	Rand	30%	29.9	38.4	48.9	26.5	39.6	51.3	65.2	31.3	43.8
×	DRRQR	30%	**29.0**	36.6	51.1	28.2	39.9	50.4	65.3	32.4	**44.5**
×	Grad	30%	**29.0**	**32.4**	50.6	27.0	40.2	50.4	64.7	33.7	44.4
×	S-Wanda	30%	33.2	66.5	47.4	27.7	39.5	50.0	65.2	25.3	42.5
✓	L1	30%	28.1	38.5	50.6	27.9	39.1	49.9	65.7	31.1	44.1
✓	Rand	30%	28.2	40.9	50.8	27.0	39.3	50.5	65.4	29.7	43.8
✓	DRRQR	30%	**27.8**	38.0	51.2	27.9	39.4	50.0	65.5	31.4	44.2
✓	Grad	30%	**27.8**	**35.8**	51.3	27.7	39.4	50.6	65.5	31.7	**44.4**
✓	S-Wanda	30%	28.0	38.8	50.5	28.0	39.1	49.0	65.3	30.7	43.8
Baseline	–	0%	28.8	35.9	51.5	28.2	39.6	50.4	65.5	31.5	44.4

Table 12. Zero-shot performance of Gated DeltaNet 1.3B models, evaluated using the *lm-eval-harness* (Gao et al., 2024), given different compression ratios. Pre- and post RFT.

RFT	Method	Compr.	Wiki. ppl ↓	LMB. ppl ↓	ARC-e acc_n ↑	ARC-c acc_n ↑	Hella. acc_n ↑	Wino. acc ↑	PIQA acc_n ↑	LMB. acc ↑	Avg ↑
×	L1	75%	68.8	279.5	54.4	32.8	47.2	55.4	66.2	15.9	45.3
×	Rand	75%	32.8	37.8	56.0	34.3	51.8	54.5	70.2	30.8	49.6
×	DRRQR	75%	22.5	23.6	62.6	36.5	55.7	59.8	72.1	37.0	54.0
×	Grad	75%	**22.3**	**17.7**	64.0	38.3	57.1	60.2	71.4	42.5	**55.6**
×	S-Wanda	75%	30.6	77.0	59.7	36.3	54.3	58.9	69.9	24.6	50.6
✓	L1	75%	18.6	16.8	66.1	37.8	57.0	58.0	72.4	40.7	55.3
✓	Rand	75%	19.3	18.1	64.7	36.5	56.6	57.8	72.5	39.5	54.6
✓	DRRQR	75%	17.9	14.7	64.0	37.8	58.4	61.1	73.1	43.1	56.2
✓	Grad	75%	**17.8**	**12.5**	65.1	37.5	58.4	60.9	72.9	45.8	**56.8**
✓	S-Wanda	75%	18.2	15.1	63.9	37.7	57.5	59.0	72.6	41.7	55.4
×	L1	50%	26.5	34.0	64.8	36.5	56.3	58.2	71.6	32.3	53.3
×	Rand	50%	18.8	14.4	65.5	38.3	58.8	59.7	73.2	43.7	56.5
×	DRRQR	50%	**17.3**	14.2	66.2	39.2	59.9	62.5	73.1	44.0	57.5
×	Grad	50%	17.4	**10.1**	65.9	39.8	60.5	61.2	72.7	51.3	**58.6**
×	S-Wanda	50%	21.9	24.1	65.1	38.5	58.7	60.0	72.1	36.1	55.1
✓	L1	50%	16.8	13.0	66.0	39.2	59.2	59.8	73.6	45.3	57.2
✓	Rand	50%	16.9	11.5	66.6	38.9	58.8	61.2	73.6	47.8	57.8
✓	DRRQR	50%	**16.3**	12.0	66.8	39.4	59.4	61.1	73.7	47.4	58.0
✓	Grad	50%	16.4	**10.5**	65.7	39.3	59.6	61.1	73.9	49.9	**58.3**
✓	S-Wanda	50%	16.6	11.7	66.6	39.2	59.4	60.5	73.8	47.3	57.8
×	L1	40%	23.7	25.3	64.3	38.5	57.8	58.6	71.7	36.1	54.5
×	Rand	40%	18.3	11.0	65.7	39.3	59.7	60.5	73.9	49.7	58.1
×	DRRQR	40%	**16.7**	11.9	67.2	39.4	60.5	60.9	73.5	47.8	58.2
×	Grad	40%	**16.7**	**9.4**	67.8	40.5	60.6	62.3	73.5	52.6	**59.6**
×	S-Wanda	40%	20.7	21.6	65.4	39.6	59.5	60.3	72.4	37.4	55.8
✓	L1	40%	16.5	11.9	67.2	38.8	59.6	59.8	74.0	47.5	57.8
✓	Rand	40%	16.5	10.7	66.4	39.2	59.6	61.3	73.8	49.5	58.3
✓	DRRQR	40%	**16.1**	11.0	67.2	40.0	59.8	61.3	73.9	49.8	**58.7**
✓	Grad	40%	**16.1**	**10.1**	66.7	39.4	59.9	61.2	74.2	50.5	58.6
✓	S-Wanda	40%	16.4	11.5	66.8	39.5	59.9	60.8	74.2	47.7	58.1
×	L1	30%	22.2	24.3	64.5	38.1	58.4	61.0	71.9	35.4	54.9
×	Rand	30%	16.7	11.4	65.2	39.2	60.2	60.2	73.9	48.7	57.9
×	DRRQR	30%	**16.3**	11.6	67.0	39.7	60.5	61.0	73.7	47.5	58.2
×	Grad	30%	**16.3**	**9.2**	68.5	40.4	60.7	62.4	73.7	52.8	**59.7**
×	S-Wanda	30%	19.8	18.4	65.5	39.8	60.1	61.5	73.1	39.7	56.6
✓	L1	30%	16.3	11.3	67.5	39.2	59.7	61.2	74.1	48.3	58.3
✓	Rand	30%	16.1	10.2	66.5	39.6	59.8	61.6	74.4	50.4	58.7
✓	DRRQR	30%	**15.9**	10.7	67.3	39.9	59.9	61.8	74.3	49.7	58.8
✓	Grad	30%	**15.9**	**9.9**	66.9	40.4	60.0	61.6	74.4	50.9	**59.0**
✓	S-Wanda	30%	16.1	11.0	67.6	40.3	60.0	60.5	74.0	49.0	58.6
Baseline	–	0%	16.8	9.7	67.7	41.0	60.1	62.2	74.0	51.5	59.4

J. More Plots

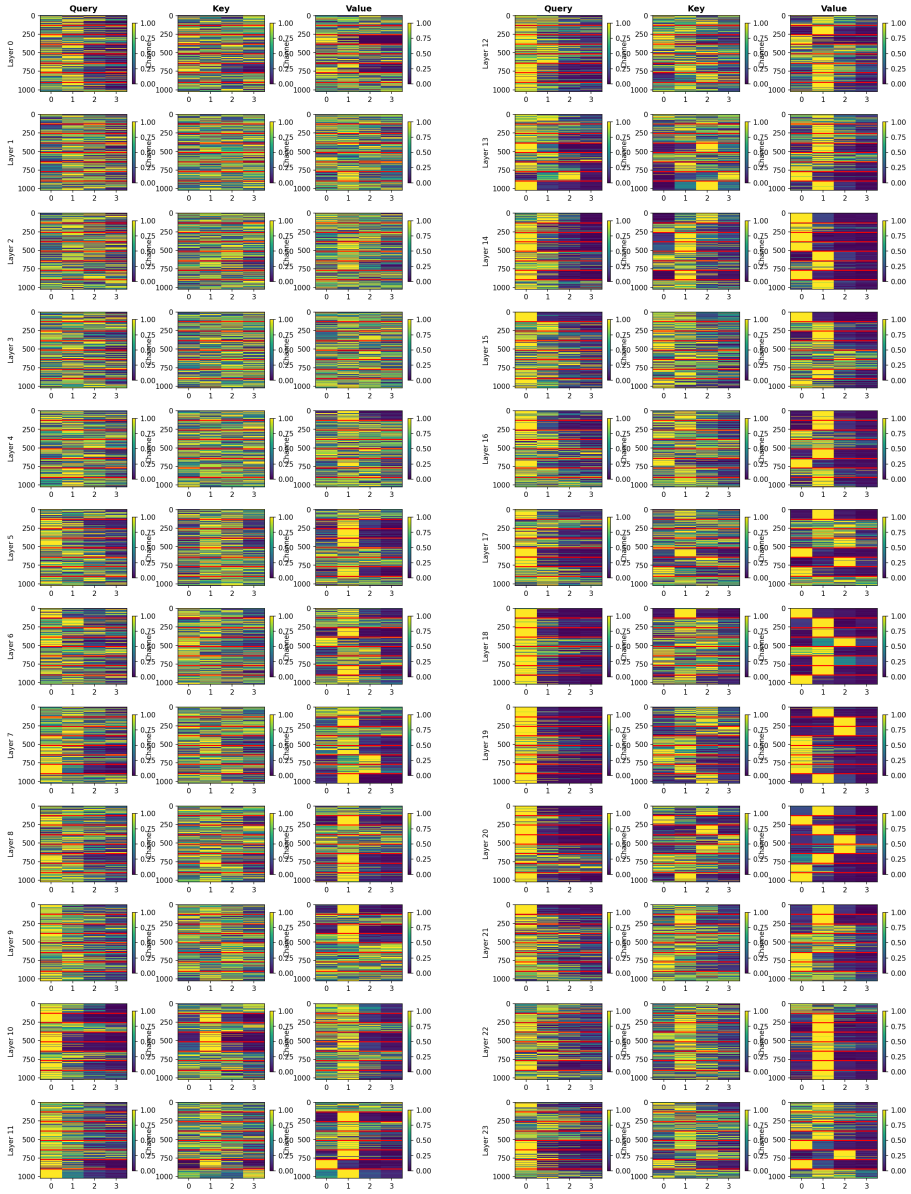

Figure 6. Convolution filters for queries, keys, and values of a standard DeltaNet 370M (non-shared). High similarity within heads (separated by red lines) suggests implicit sharing.

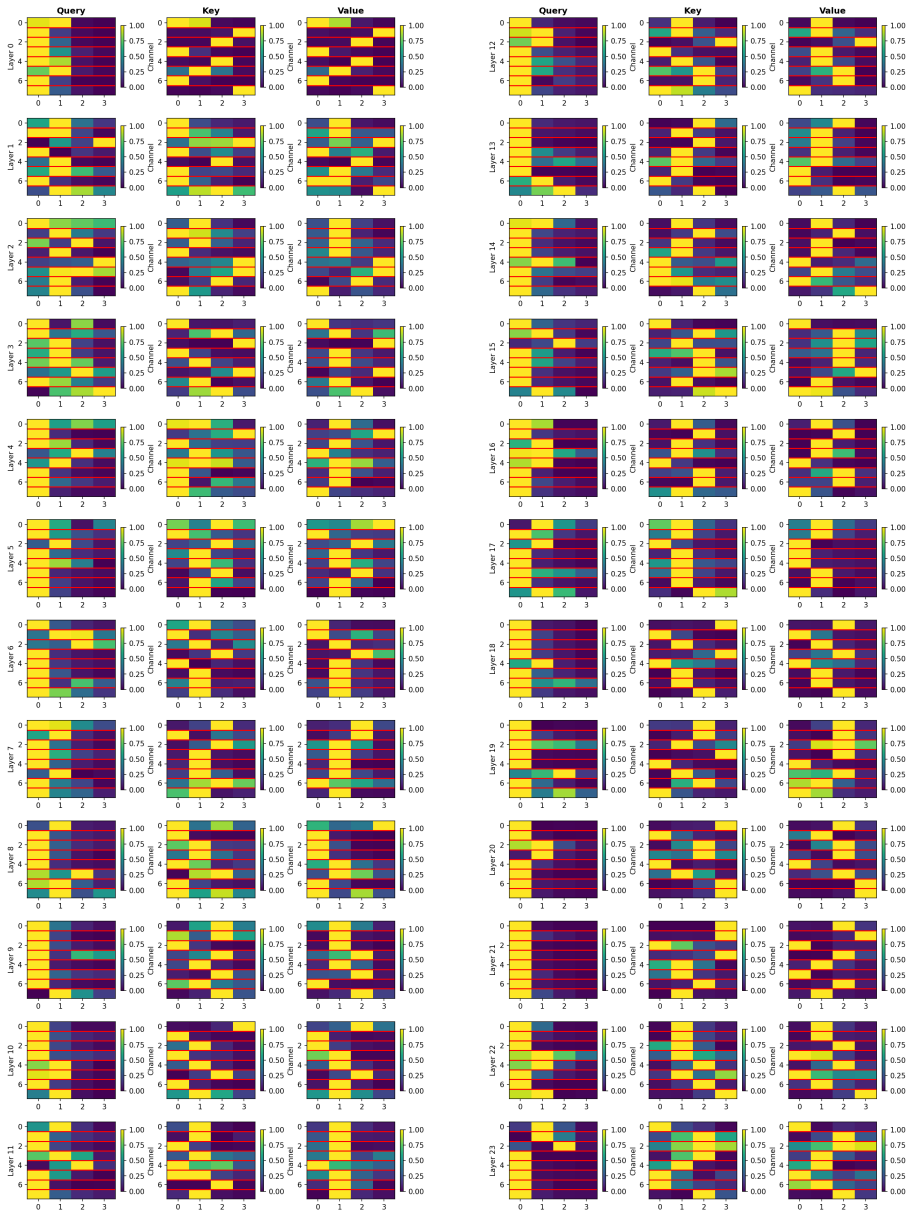

Figure 7. Convolution filters for DeltaNet 370M with explicitly shared convolutions. The model learns distinct patterns for Q, K, and V.