LinearARD: linear-memory Attention Distillation for RoPE Restoration

Ning Yang[1,†*] **Hengyu Zhong**[2,†] **Wentao Wang**[3,†] **Baoliang Tian**[4]
Yuan Zhou[4] **Haijun Zhang**[5] **Jun Wang**[6]
[1]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]Southwest University, Chongqing, China
[3]Dalian University of Technology, Dalian, Liaoning, China
[4]Tianjin University, Tianjin, China
[5]University of Science and Technology Beijing, Beijing, China
[6]University College London, London, United Kingdom

Abstract

The extension of context windows in Large Language Models is typically facilitated by scaling positional encodings followed by Continued Pre-Training (CPT). While effective, this paradigm is notoriously data-hungry and computationally expensive, requiring massive long-text corpora to recalibrate the model to the shifted positional distribution. We propose LinearARD, a restoration distillation method that restores Rotary Position Embedding (RoPE)-scaled students through attention-structure consistency with a frozen native-RoPE teacher. Rather than next-token prediction or opaque hidden-state matching, LinearARD aligns row-wise distributions of dense Q/Q, K/K, and V/V self-relation matrices from the final attention layer to directly supervise attention dynamics. To remove the quadratic memory bottleneck of $n \times n$ relation maps, we introduce a linear-activation-memory kernel that stores only per-token log-sum-exp statistics and recomputes logits in the backward pass to obtain exact Kullback-Leibler divergence gradients. Across LLaMA2-7B, LLaMA3-8B, and Mistral-7B-v0.1 extended to 32K context, LinearARD recovers 93.1%/94.2%/94.3% of native short-context performance and achieves strong long-context robustness on RULER using only **4.25M** tokens—amounting to just 1.6% (a $\sim$60$\times$ reduction) of the 256M-token budget required by state-of-the-art baselines. Under this severely constrained budget, CPT and LongReD remain near zero on RULER, demonstrating that relation-level restoration provides a substantially better efficiency-quality tradeoff. Our code is available at `https://github.com/gracefulning/LinearARD`.

1 Introduction

The practical utility of Large Language Models (LLMs) is increasingly defined by their long-context capabilities, which are essential for advanced tasks such as retrieval-augmented generation Asai et al. [2024], multi-step agentic workflows Yao et al. [2022], and document-level reasoning Liu et al. [2024]. However, training LLMs from scratch on extended context windows is prohibitively expensive, which motivates the development of post-hoc context extension methods that leverage pretrained models Chen et al. [2023], Peng et al. [2023], Ding et al. [2024], Zhu et al. [2024].

A widely used family of techniques extends the context window at inference time by modifying Rotary Position Embeddings (RoPE) Su et al. [2024]. These methods include position interpolation and

*Corresponding author: Ning Yang (`ning.yang@ia.ac.cn`).
†Ning Yang, Hengyu Zhong, and Wentao Wang contributed equally to this work.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

related scaling schedules such as linear interpolation Chen et al. [2023], NTK-aware scaling bloc97 [2023], Yet another RoPE extension method (YaRN)Peng et al. [2023], and LongRoPEDing et al. [2024]. While effective for long-context inference, such scaling often degrades short-context accuracy. This is because altering the rotary frequency schedule shifts the relative positional relationships between tokens and disrupts the learned attention patterns, which consequently impairs performance on shorter sequences.

To mitigate this trade-off, various approaches have been proposed. Continued Pre-Training (CPT) Ke et al. [2023] adapts the model by further training on long-context data under the scaled configuration. However, this approach typically requires substantial computational resources and high-quality long-context corpora, which are often scarce, and it runs the risk of catastrophic forgetting regarding the original short-context capabilities. Consequently, restoration distillation was introduced in subsequent works Gu et al. [2023], Dong et al. [2025], which treats the original model as a teacher and the RoPE-scaled model as a student, fine-tuning the student to match the teacher on sequences within the teacher's native length while retaining the expanded maximum context. Nevertheless, existing restoration methods primarily focus on hidden-state matching. Because hidden states are aggregated outputs, they provide only indirect constraints that fail to precisely rectify the fine-grained positional distortions in the attention mechanism, limiting the efficiency and accuracy of the restoration.

Aligning distributional quantities within the attention module presents a conceptually appealing solution to the limitations of indirect constraints. Since RoPE scaling acts directly on queries and keys, it fundamentally alters attention logits; therefore, directly supervising the resulting distributions offers a more precise path to restoration. However, this approach faces a significant system-level bottleneck. Attention-level objectives typically incur quadratic memory overhead, and materializing full attention maps for backpropagation quickly exhausts GPU memory even at moderate sequence lengths. This burden is further exacerbated in a distillation setting, where both teacher and student models must be maintained simultaneously. Consequently, prior attention distillation methods Sun et al. [2020], Wang et al. [2020], Jiao et al. [2020] have been typically restricted to short sequences. Although various targets have been investigated, such as attention maps Sun et al. [2020], Jiao et al. [2020], relation matrices Wang et al. [2020, 2021], and output logits Hinton et al. [2015], Sanh et al. [2019], scaling these to long contexts often necessitates selective or sparse objectives He et al. [2024], which sacrifices exact distribution matching.

Following the discussion on attention distillation and full attention maps, **LinearARD** is proposed. This approach targets the root cause of performance degradation by enforcing structural consistency on the dense self-relation matrices, specifically Q/Q, K/K, and V/V, within each attention head. To overcome the associated memory bottleneck, we design an IO-aware tiled distillation kernel with linear activation memory. By recomputing relation logits on-chip during the backward pass, the kernel evaluates exact Kullback-Leibler (KL) divergence and its gradients without materializing full probability matrices, thereby enabling high-fidelity structural distillation on long sequences.

Our contributions are summarized as follows:

- We introduce an IO-aware gradient fusion kernel that computes the exact KL divergence with linear activation memory complexity $\mathcal{O}(n)$. By bypassing the quadratic memory bottleneck, this kernel enables direct structural supervision on ultra-long sequences, where standard methods would otherwise fail due to GPU memory exhaustion.
- We propose LinearARD, a framework that enforces structural consistency across Q/Q, K/K, and V/V self-relations. This objective directly rectifies the positional misalignments induced by RoPE scaling and stabilizes representation drift, demonstrating robust transfer across materially different scaling schemes, including Position Interpolation and NTK-aware scaling.
- Theoretical analysis shows that the proposed kernel scales linearly in activation memory (Theorem 3.2) while remaining exactly equivalent to standard full-matrix backpropagation (Proposition 3.3), providing a foundation for high-fidelity distillation.
- Extensive evaluations on LLaMA2-7B, LLaMA3-8B, and Mistral-7B-v0.1 show that LinearARD attains strong long-context performance on both RULER (63.0/67.8/60.2) and LongBench-E (22.65/21.18/17.17) using only 4.25M training tokens, while retaining 93.1%/94.2%/94.3% of native short-context performance. This corresponds to roughly a 60$\times$ token reduction relative to 256M-token restoration baselines.

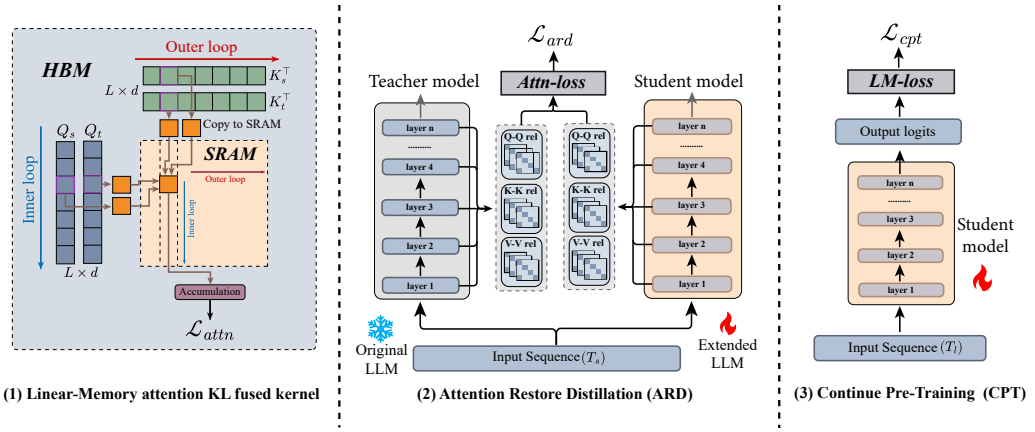

Figure 1: LinearARD pipeline. A frozen native-RoPE teacher provides final-layer dense row-wise relation distributions (Q/Q, K/K, and V/V), and the RoPE-scaled student is restored by minimizing relation KL with an exact linear-activation-memory kernel.

2 Related Work

Efficient Attention and Memory Optimization. The quadratic memory complexity of self-attention ($O(n^2)$) constitutes the primary bottleneck for long-context modeling. Seminal works like FlashAttention Dao et al. [2022], Dao [2023] and Ring Attention Liu et al. [2023] address this by tiling computations in GPU SRAM and utilizing online statistics to compute the attention output without materializing the full attention matrix. While these kernels reduce the memory footprint of standard forward and backward passes to linear scale ($O(n)$), they do not support the computation of distributional alignment objectives. In a distillation setting, minimizing the KL divergence between teacher and student distributions typically necessitates instantiating full $n \times n$ probability matrices to calculate loss gradients, which reintroduces the quadratic memory bottleneck. Our work bridges this gap by proposing an IO-aware kernel that fuses the KL divergence calculation into the backward pass. Unlike prior kernels that execute self-attention forward and backward passes, our kernel enables exact, linear-activation-memory structural supervision, making dense relation alignment feasible at long sequence lengths where it was previously memory-prohibitive.

Structural Distillation in Transformers. Knowledge Distillation (KD) Hinton et al. [2015] is a standard paradigm for transferring capabilities from a teacher to a student model. Beyond logit and hidden-state distillation, aligning attention mechanisms is critical for capturing linguistic dependencies. Approaches such as TinyBERT Jiao et al. [2020] and MobileBERT Sun et al. [2020] align attention maps directly, while MiniLM Wang et al. [2020, 2021] enhances stability by distilling self-relation modules. However, the efficacy of these methods is strictly limited by their spatial complexity; enforcing consistency on dense matrices is feasible only for short context windows. Consequently, recent long-context works resort to sparse supervision or token-level logit matching He et al. [2024], sacrificing structural fidelity for memory efficiency. By overcoming this memory barrier, our framework revisits and scales these dense objectives. Full-context alignment of Q, K, and V self-relation distributions is enabled on ultra-long sequences, ensuring the student retains the precise structural priors of the teacher.

RoPE Scaling and Context Restoration. Techniques such as Position Interpolation (PI)Chen et al. [2023], NTK-aware scalingbloc97 [2023], YaRN Peng et al. [2023], and LongRoPE Ding et al. [2024] extend the context window of pretrained LLMs by modifying RoPE frequencies or position indices. While effective for extension, this rescaling distorts the rotation-sensitive geometric relationships established during pretraining, leading to a collapse in short-context performance. CPT Ke et al. [2023] mitigates this but requires extensive compute and risks catastrophic forgetting. More recently, restoration distillation methods like LongReD Dong et al. [2025] have attempted to recover performance by matching hidden states. However, hidden states are aggregated features derived after the attention operation; they provide only a coarse, indirect signal that fails to isolate the root cause of the degradation. Since RoPE scaling directly perturbs the dot-product attention logits,

restoration must target these internal relational structures directly Su et al. [2024]. **LinearARD** corrects these fine-grained geometric distortions at their source, achieving significantly higher data efficiency and restoration accuracy than indirect hidden-state matching.

3 Methodology

RoPE scaling modifies the rotary frequency schedule, altering the distance-dependent phase between queries and keys and causing the attention mechanism to drift relative to the original model. This redistribution of attention can degrade the short-context behavior established during pretraining, thereby reducing performance on short-context tasks. This issue is addressed by formulating restoration as a restoration distillation problem between an unscaled teacher model and a RoPE-scaled student model. The teacher provides stable supervision on sequences within its native context range, and the student is optimized to recover the teacher's short-context behavior while retaining the extended context window enabled by RoPE scaling.

This section first defines the restoration objective by specifying which internal structures are matched between the teacher and the RoPE-scaled student (Sec. 3.2). It then introduces an exact linear-activation-memory operator, a dedicated Kernel for KL distillation on dense $n \times n$ relation maps that bypasses the quadratic-memory bottleneck (Sec. 3.3). Finally, the discussion details the training procedure and implementation choices, including a parameter-efficient restoration recipe and an optional lightweight CPT stage (Sec. 3.4).

Method Overview. As illustrated in Fig. 1, the approach consists of three components in the above order. First, the teacher model's final-layer relational structure is distilled by aligning row-wise relation distributions induced by Q/Q, K/K, and V/V self-relations (Sec. 3.2). Second, the quadratic-memory bottleneck is eliminated with an exact linear-activation-memory Kernel for KL distillation of dense $n \times n$ relation maps (Sec. 3.3). Third, a practical and parameter-efficient restoration procedure is adopted to optimize the student under these objectives, optionally augmented with a lightweight CPT stage (Sec. 3.4).

3.1 Problem Setup

This work considers a decoder-only Transformer with L layers and H attention heads per layer. Let B denote the batch size, n the sequence length, and d the per-head dimension. The original model with native RoPE serves as a frozen teacher, while the student shares the same architecture but employs a scaled RoPE configuration to support a larger maximum context length. During distillation, both models are evaluated on the same token sequence of length n within the teacher's supported range. The teacher remains fixed to provide supervision, while the student uses scaled RoPE during both training and inference. Unless explicitly stated as an ablation, LinearARD applies the relation-alignment loss only to the final Transformer layer, denoted by $\ell^\star = L$.

The model employs masked causal self-attention with an additive mask $\mathbf{M} \in \mathbb{R}^{n\times n}$. Query positions are indexed by $i \in \{1, \dots, n\}$ and key positions by $j \in \{1, \dots, n\}$. This mask enforces causality and padding: $\mathbf{M}(i, j) = -\infty$ if position j is not visible to query i, and $\mathbf{M}(i, j) = 0$ otherwise.

For model index $m \in \{t, s\}$, where t and s denote the teacher and student respectively, the query, key, and value tensors in the selected final layer $\ell^\star$ are represented as $\mathbf{Q}_m \in \mathbb{R}^{B\times H_Q\times n\times d}$ and $\mathbf{K}_m, \mathbf{V}_m \in \mathbb{R}^{B\times H_{KV}\times n\times d}$, where H_Q and H_{KV} denote the number of query and key-value heads. To simplify the definition of the distillation objective, we formulate the loss for a single attention head and batch element, treating $\mathbf{Q}_m, \mathbf{K}_m, \mathbf{V}_m$ as matrices in $\mathbb{R}^{n\times d}$ and omitting head and batch indices when the context is unambiguous.

3.2 Distillation Objectives

RoPE scaling perturbs the relative geometry of the projected representations, changing the relation distributions induced inside each attention head. The student is restored by matching these row-wise relation distributions to those of the frozen teacher.

Relation distributions. Fix the selected final layer, an attention head, and a batch element. Let $\mathbf{X}_m, \mathbf{Y}_m \in \mathbb{R}^{n\times d}$ be two sequences of projected vectors from model $m \in t, s$. A masked similarity

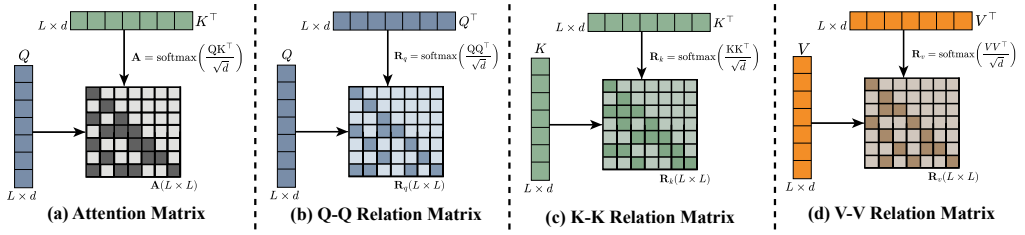

Figure 2: Final-layer standard attention vs. QKV self-relations. (a) Attention matrix $\mathbf{A}$ computed from $\mathbf{QK}^\top$ (shown for reference), and (b–d) Q/Q, K/K, and V/V relation matrices computed from $\mathbf{QQ}^\top$, $\mathbf{KK}^\top$, and $\mathbf{VV}^\top$ (Eq. 4). LinearARD distills the row-wise relation distributions in (b–d) from the final Transformer layer by aligning teacher vs. student rows via forward KL (Eq. 3) under the same mask $\mathbf{M}$.

logit matrix is defined

$$\mathbf{Z}_m \;=\; \frac{1}{\sqrt{d}}\,\mathbf{X}_m\mathbf{Y}_m^\top \;+\; \mathbf{M} \quad \in \mathbb{R}^{n\times n}. \tag{1}$$

Applying a row-wise softmax yields a row-wise relation distribution

$$\mathbf{R}_m \;=\; \mathrm{softmax}(\mathbf{Z}_m), \tag{2}$$

where $\mathbf{R}_m(i,:)$ is a categorical distribution over key positions $j \in \{1,\dots,n\}$ for each fixed query position i.

For a fixed query position i, let $\mathcal{J} \subseteq \{1,\dots,n\}$ denote the visible key positions. For each $j \in \mathcal{J}$, write $r_{t,j} \triangleq \mathbf{R}_t(i,j)$ and $r_{s,j} \triangleq \mathbf{R}_s(i,j)$ for the teacher and student probabilities, respectively, and $z_{s,j} \triangleq \mathbf{Z}_s(i,j)$ for the corresponding student logit.

KL objective. The distillation objective matches teacher and student relation distributions by minimizing the average forward KL divergence between corresponding rows:

$$\mathcal{L}_{\mathrm{KL}}(\mathbf{R}_t,\mathbf{R}_s) = \frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}\big(\mathbf{R}_t(i,:)\,\|\,\mathbf{R}_s(i,:)\big), \tag{3}$$

In practice, this loss is computed on the final Transformer layer $\ell^\star = L$ and averaged over heads and batch elements; lower layers are not directly aligned in the default LinearARD objective. Layer-scope variants are studied only as ablations in Appendix E.

Proposition 3.1 (Gradient Behavior in Sparse Regimes)**.** *Consider a query row over visible positions $\mathcal{J}$. Let $r_{t,j}$ and $r_{s,j} = \frac{\exp(z_{s,j})}{\sum_{a\in\mathcal{J}}\exp(z_{s,a})}$ denote the teacher and student distributions, with row losses:*

$$\mathcal{L}^{\mathrm{row}}_{\mathrm{MSE}} = \frac{1}{2}\sum_{j\in\mathcal{J}}(r_{s,j}-r_{t,j})^2, \qquad \mathcal{L}^{\mathrm{row}}_{\mathrm{KL}} = \sum_{j\in\mathcal{J}} r_{t,j}\log\frac{r_{t,j}}{r_{s,j}}.$$

For any $k \in \mathcal{J}$, the derivatives with respect to logit $z_{s,k}$ satisfy:

$$\frac{\partial\mathcal{L}^{\mathrm{row}}_{\mathrm{MSE}}}{\partial z_{s,k}} = r_{s,k}\Big[(r_{s,k}-r_{t,k}) - \sum_{j\in\mathcal{J}} r_{s,j}(r_{s,j}-r_{t,j})\Big], \qquad \frac{\partial\mathcal{L}^{\mathrm{row}}_{\mathrm{KL}}}{\partial z_{s,k}} = r_{s,k} - r_{t,k}.$$

Consequently, as $r_{s,k} \to 0$ with fixed $r_{t,k} > 0$:

$$\frac{\partial\mathcal{L}^{\mathrm{row}}_{\mathrm{MSE}}}{\partial z_{s,k}} \to 0, \qquad \frac{\partial\mathcal{L}^{\mathrm{row}}_{\mathrm{KL}}}{\partial z_{s,k}} \to -r_{t,k}.$$

The proof is provided in Appendix H.

This analytical distinction motivates the choice of forward KL over probability MSE because attention-like distributions are typically sparse and peaked. When the student assigns near-zero mass to a dependency supported by the teacher, MSE gradients can vanish due to the softmax Jacobian, whereas forward KL provides a first-order correction signal.

Algorithm 1 Linear-Activation-Memory KL Distillation for Relation Distributions (Forward)

Input: Student $(\mathbf{X}_s, \mathbf{Y}_s)$, teacher $(\mathbf{X}_t, \mathbf{Y}_t)$, and $\mathcal{O}(n)$ mask metadata $\mathcal{C}$.
Output: Loss $\mathcal{L}$.
Note: For QKV relations, set $(\mathbf{X}_m, \mathbf{Y}_m) \leftarrow (\mathbf{Q}_m, \mathbf{Q}_m)$, $(\mathbf{K}_m, \mathbf{K}_m)$, or $(\mathbf{V}_m, \mathbf{V}_m)$ as in Eq. 4.
Note: $\mathrm{ComputeLSE}(\mathbf{X}_m, \mathbf{Y}_m; \mathcal{C})$ computes $\mathrm{LSE}_m(a) = \log \sum_{b \in \mathcal{J}_a} \exp(\mathbf{Z}_m(a, b))$ via masked tiling.
PHASE 1: GLOBAL STATISTICS (LINEAR ACTIVATION MEMORY)
$\mathrm{LSE}_s \leftarrow \mathrm{ComputeLSE}(\mathbf{X}_s, \mathbf{Y}_s; \mathcal{C})$
$\mathrm{LSE}_t \leftarrow \mathrm{ComputeLSE}(\mathbf{X}_t, \mathbf{Y}_t; \mathcal{C})$
PHASE 2: FUSED FORWARD PASS VIA TILING
Initialize loss $\mathcal{L} \leftarrow 0$
for query blocks $\mathbf{X}_s^{(I)}, \mathbf{X}_t^{(I)}$ loaded to SRAM **do**
  for key blocks $\mathbf{Y}_s^{(J)}, \mathbf{Y}_t^{(J)}$ loaded to SRAM **do**
    // Generate mask tile on-the-fly in SRAM
    $\mathbf{M}^{(I,J)} \leftarrow \mathrm{GenerateMask}(I, J; \mathcal{C}), \quad \mathcal{V} \leftarrow \{(r, c) \mid \mathbf{M}^{(I,J)}(r, c) = 0\}$
    $\mathbf{Z}_s \leftarrow \frac{1}{\sqrt{d}} \mathbf{X}_s^{(I)} (\mathbf{Y}_s^{(J)})^\top + \mathbf{M}^{(I,J)}$
    $\mathbf{Z}_t \leftarrow \frac{1}{\sqrt{d}} \mathbf{X}_t^{(I)} (\mathbf{Y}_t^{(J)})^\top + \mathbf{M}^{(I,J)}$
    // Exclude masked entries before difference to avoid NaN
    $\boldsymbol{\ell}_s \leftarrow \big(\mathbf{Z}_s - \mathrm{LSE}_s^{(I)}\big)[\mathcal{V}], \quad \boldsymbol{\ell}_t \leftarrow \big(\mathbf{Z}_t - \mathrm{LSE}_t^{(I)}\big)[\mathcal{V}]$
    $\mathbf{p}_t \leftarrow \exp(\boldsymbol{\ell}_t)$
    $\mathcal{L} \leftarrow \mathcal{L} + \sum \big[\mathbf{p}_t \odot (\boldsymbol{\ell}_t - \boldsymbol{\ell}_s)\big]$
  end for
end for
$\mathcal{L} \leftarrow \mathcal{L}/n$

QKV self-relation targets. Eq. 1 and Eq. 2 are instantiated using Q/Q, K/K, and V/V self-relations. Concretely, for each model $m \in \{t, s\}$ the following are defined:

$$\mathbf{R}_m^Q = \mathrm{softmax}\left(\frac{\mathbf{Q}_m \mathbf{Q}_m^\top}{\sqrt{d}} + \mathbf{M}\right), \tag{4a}$$

$$\mathbf{R}_m^K = \mathrm{softmax}\left(\frac{\mathbf{K}_m \mathbf{K}_m^\top}{\sqrt{d}} + \mathbf{M}\right), \tag{4b}$$

$$\mathbf{R}_m^V = \mathrm{softmax}\left(\frac{\mathbf{V}_m \mathbf{V}_m^\top}{\sqrt{d}} + \mathbf{M}\right). \tag{4c}$$

While V/V is not a direct RoPE correction, it preserves the geometry of aggregated content and limits value drift under joint QKV training (see ablation in Table 3, Appendix A). Although self-relations do not algebraically reconstruct cross-attention $\mathbf{Q}\mathbf{K}^\top$, minimizing them empirically restores attention dynamics with higher stability than direct attention matching. Fig. 2 visualizes the standard attention matrix alongside the Q/Q, K/K, and V/V self-relation distributions.

Overall loss. Eq. 3 is applied to each of $\mathbf{R}^Q$, $\mathbf{R}^K$, and $\mathbf{R}^V$:

$$\mathcal{L} = \lambda_q \, \mathcal{L}_{\mathrm{KL}}(\mathbf{R}_t^Q, \mathbf{R}_s^Q) + \lambda_k \, \mathcal{L}_{\mathrm{KL}}(\mathbf{R}_t^K, \mathbf{R}_s^K) + \lambda_v \, \mathcal{L}_{\mathrm{KL}}(\mathbf{R}_t^V, \mathbf{R}_s^V). \tag{5}$$

Here $\lambda_q, \lambda_k, \lambda_v \geq 0$ are scalar weights; unless otherwise noted, $\lambda_q = \lambda_k = \lambda_v = 1$ is used. The loss in Eq. 5 is evaluated only for the selected final layer $\ell^\star$ in the main experiments.

3.3 linear-activation-memory KL Distillation Kernel

Relational distillation requires matching dense $n \times n$ relation maps, whose naive implementation has quadratic activation memory. An IO-aware tiled gradient fusion kernel is proposed to compute the exact KL loss and gradients using $\mathcal{O}(n)$ memory by avoiding materialization of any $n \times n$ probability matrix.

Key identity. Let $\mathbf{Z}_s, \mathbf{Z}_t \in \mathbb{R}^{n\times n}$ be the masked similarity logits of the student and teacher, and let $\mathbf{R}_s = \mathrm{softmax}(\mathbf{Z}_s)$ and $\mathbf{R}_t = \mathrm{softmax}(\mathbf{Z}_t)$ be the corresponding row-wise distributions. Differentiating Eq. 3 with respect to the student logits yields

$$\frac{\partial \mathcal{L}_{\mathrm{KL}}}{\partial \mathbf{Z}_s(i,j)} = \frac{1}{n}\Big(\mathbf{R}_s(i,j) - \mathbf{R}_t(i,j)\Big). \tag{6}$$

Eq. 6 shows that each gradient entry depends only on the local probabilities for the same (i, j). Therefore, if $\mathbf{R}_m(i, j)$ can be reconstructed on the fly without storing the full matrix, exact gradients can be computed with linear activation memory.

Two-pass tiled execution. A two-pass tiled strategy inspired by FlashAttention Dao [2023] is adopted. First, the row-wise log-sum-exp statistics are computed and stored

$$\mathrm{LSE}_m(i) = \log \sum_{k=1}^{n} \exp(\mathbf{Z}_m(i,k)),$$

for both models $m \in \{t, s\}$. This requires $\mathcal{O}(n)$ storage in High Bandwidth Memory (HBM).

Second, query and key tiles of size $T_r \times T_c$ are iterated. For each tile, $\mathbf{Z}_s$ and $\mathbf{Z}_t$ are recomputed and local probabilities are reconstructed using the stored $\mathrm{LSE}_m(i)$ values:

$$\mathbf{R}_m(i,j) = \exp\big(\mathbf{Z}_m(i,j) - \mathrm{LSE}_m(i)\big).$$

All intermediate tile tensors reside in fast on-chip SRAM, and the accumulated gradients are streamed to HBM. Optionally, the scalar KL loss is computed in the same tiled pass via the decomposition

$$\mathcal{L}_{\mathrm{KL}} = \frac{1}{n}\sum_{i,j} \mathbf{R}_t(i,j)\Big[(\mathbf{Z}_t(i,j) - \mathrm{LSE}_t(i)) - (\mathbf{Z}_s(i,j) - \mathrm{LSE}_s(i))\Big]. \tag{7}$$

Complexity and guarantees. The efficiency and correctness of the kernel are established through the following statements.

Theorem 3.2 (Linear Activation Memory Complexity)**.** *The proposed KL distillation Kernel reduces activation memory complexity from $\mathcal{O}(BHn^2)$ to $\mathcal{O}(BHnd)$, which scales linearly with sequence length n for fixed d.*

Proposition 3.3 (Mathematical Exactness)**.** *The gradients computed by the tiled formulation in Algorithm 2 (Appendix K) are mathematically equivalent to the analytical gradients obtained from standard full-matrix backpropagation.*

While activation memory scales as $\mathcal{O}(BHnd)$, arithmetic complexity remains quadratic ($\mathcal{O}(BHn^2d)$) since all pairwise relations are evaluated on-chip. The proofs for Theorem 3.2 and Proposition 3.3 are provided in Appendix I and Appendix J. Figure 3 provides empirical diagnostics verifying memory scaling and numerical exactness, and Appendix C reports wall-clock throughput. By choosing $(\mathbf{X}, \mathbf{Y})$ as $(\mathbf{Q}, \mathbf{Q})$, $(\mathbf{K}, \mathbf{K})$, or $(\mathbf{V}, \mathbf{V})$, this Kernel enables exact QKV relation distillation described in Sec. 3.2 for long sequences.

3.4 Training Recipe and Parameter Efficiency

With the distillation targets and losses defined and the exact linear-activation-memory KL kernel established, the training procedure used to optimize the RoPE-scaled student is described next. The goal is to restore short-context behavior while retaining the extended context window enabled by RoPE scaling.

Stage 1: Attention distillation. Distillation is performed on sequences within the teacher's native context range. By optimizing Eq. 5 on the final attention layer, the Q/Q, K/K, and V/V relation distributions in Eq. 4 are aligned, directly correcting relational distortions introduced by RoPE scaling while avoiding the cost of supervising every layer.

Stage 2: Context adaptation. After relation distillation, a lightweight CPT stage is optionally run under the scaled RoPE configuration. This stage uses the standard language modeling objective to adapt the student to the expanded context window. In this setting, Stage 1 already restores short-context behavior, so this optional stage can be kept compute-efficient.

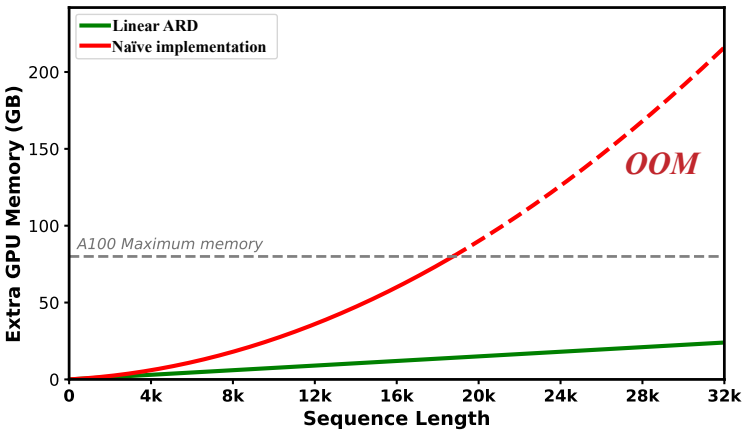

Figure 3: Memory scaling of the linear-activation-memory KL distillation Kernel in Sec. 3.3 as a function of sequence length.

Parameter efficiency. Most of the model is frozen and only the attention projection matrices are optimized, focusing updates on components most sensitive to RoPE perturbations. To further reduce training cost, LoRA or QLoRA adapters Hu et al. [2022], Dettmers et al. [2023] are optionally employed during both stages.

4 Experiments

4.1 Experimental Setup

Models. We study LLaMA2-7B Touvron et al. [2023], LLaMA3-8B Grattafiori et al. [2024], and Mistral-7B-v0.1, using native training windows of 4K, 8K, and 8K, respectively. The main experiments extend these windows to 32K using Position Interpolation (PI) Chen et al. [2023]: the scaling factor is eight for LLaMA2 and four for LLaMA3 and Mistral. LLaMA2 uses 32 query and 32 key/value heads, while the two GQA backbones use 32 query and eight key/value heads. To test sensitivity beyond PI, we additionally evaluate fixed-theta NTK-aware scaling bloc97 [2023] on LLaMA2-7B. The selected positional configuration is held fixed during restoration, adapter merging, CPT, and evaluation; the teacher retains native RoPE. Appendix N specifies the exact PI and NTK configurations.

Training and optimization. For each backbone, the PI-scaled model acts as the student and the native-RoPE model as a frozen teacher. The student is trained on sequences within the teacher's native context range using our final-layer Q/Q, K/K, and V/V relation distillation objective (Eq. 5). Unless otherwise stated, only the attention-module $Q/K/V$ projection weights are updated and all remaining parameters are kept fixed. In addition to this distillation phase, a short continued-pretraining phase is run at the extended context length to further adapt the student to long sequences. All models are optimized with AdamW, using a base learning rate of 2×10^{-5}, linear warmup followed by cosine decay, gradient clipping at 5.0, and gradient checkpointing for the student.

Baselines. Our comparisons include PI-scaled models without restoration, CPT under PI, and LongReD Dong et al. [2025], which restores performance via hidden-state distillation. For CPT and LongReD, the training budgets reported for this setting (256M tokens) are followed, while the restoration uses only a few million tokens and adds a lightweight 2M-token CPT phase.

Evaluation Benchmarks. Short-context performance is evaluated on MMLU Hendrycks et al. [2020] and seven standard short-text benchmarks, including LAMBADA Paperno et al. [2016], MathQA Amini et al. [2019], BoolQ Clark et al. [2019], OpenBookQA Mihaylov et al. [2018], PIQA Bisk et al. [2020], SIQA-CA Sap et al. [2019], and ARC-Challenge Clark et al. [2018]. The mean over these eight scores is reported, denoted Avg. Long-context robustness is measured with

Table 1: Main results on short-text benchmarks and long-context robustness. Short-text accuracy in percent is reported across eight benchmarks and their mean, Avg. CW: context window; PE: RoPE scaling; Rec.: recovery relative to native; RULER: average over 8K, 16K, and 32K, with per-length breakdowns in App. G. Token budgets use units of 2^{20} tokens across all stages, detailed in App. B. Bold marks the best trained result per backbone and budget group. LAMB.: LAMBADA; OBQA: OpenBookQA; ARC-C: ARC-Challenge.

				Short-text accuracy (%)									Summary		
Model	CW	PE	Method	MMLU	LAMB.	MathQA	BoolQ	OBQA	PIQA	SIQA	ARC-C	Avg.	Rec.	RULER	Tokens
LLaMA2-7B	4K	–	–	45.99	71.18	29.92	78.23	44.00	78.73	40.28	49.23	54.69	100.0	–	–
	32K	PI 8×	–	24.65	6.18	19.73	58.29	36.20	69.31	34.60	24.91	34.23	62.6	–	–
	32K	PI 8×	CPT	37.35	67.30	27.60	77.52	44.00	**78.73**	36.18	48.89	52.20	95.4	59.6	256M
	32K	PI 8×	LongReD	38.40	67.57	27.60	**78.35**	44.00	**78.73**	36.39	50.94	52.75	96.4	59.7	256M
	32K	PI 8×	LinearARD	**39.54**	**67.82**	**29.31**	76.82	**44.20**	78.40	**39.82**	**51.11**	**53.38**	**97.6**	**65.1**	256M
	32K	PI 8×	CPT	25.41	27.87	21.04	65.54	39.40	73.78	35.58	39.68	41.03	75.0	0.040	4.25M
	32K	PI 8×	LongReD	25.71	27.34	20.77	66.54	42.40	74.32	34.90	39.51	41.44	75.8	0.035	4.25M
	32K	PI 8×	LinearARD	**34.50**	**63.40**	**27.70**	**76.54**	**43.90**	**77.30**	**35.98**	**48.10**	**50.93**	**93.1**	**63.0**	4.25M
LLaMA3-8B	8K	–	–	65.42	72.22	42.08	81.71	44.80	80.85	41.61	59.38	61.01	100.0	–	–
	32K	PI 4×	–	24.17	4.01	26.53	55.84	28.60	67.19	34.70	26.96	33.50	54.9	–	–
	32K	PI 4×	CPT	60.74	69.38	**39.87**	80.24	43.60	**80.36**	39.00	57.42	58.83	96.4	81.3	256M
	32K	PI 4×	LongReD	**61.26**	68.98	38.83	**80.98**	**45.20**	80.00	39.82	**57.59**	59.08	96.9	67.9	256M
	32K	PI 4×	LinearARD	61.22	**71.76**	39.33	79.54	43.60	79.71	**41.90**	57.34	**59.30**	**97.2**	**81.6**	256M
	32K	PI 4×	CPT	35.20	48.67	29.92	71.01	36.20	73.94	34.70	43.69	46.66	76.5	0.043	4.25M
	32K	PI 4×	LongReD	41.18	44.74	31.79	73.82	37.00	74.05	34.85	46.04	47.93	78.6	0.044	4.25M
	32K	PI 4×	LinearARD	**56.70**	**66.52**	**37.12**	**80.46**	**43.60**	**80.03**	**38.18**	**57.00**	**57.45**	**94.2**	**67.8**	4.25M
Mistral-7B-v0.1	8K	–	–	62.54	72.74	36.58	84.62	45.80	82.81	41.91	60.66	60.96	100.0	–	–
	32K	PI 4×	–	23.59	2.63	21.07	59.30	32.40	62.40	33.83	28.67	32.99	54.1	–	–
	32K	PI 4×	CPT	53.53	67.94	33.70	83.43	43.60	79.71	39.82	56.57	57.29	94.0	55.3	256M
	32K	PI 4×	LongReD	58.82	69.13	35.24	**84.50**	**45.80**	**81.61**	40.99	58.98	59.38	97.4	**62.3**	256M
	32K	PI 4×	LinearARD	**58.92**	**72.52**	**36.45**	84.43	45.20	81.56	**42.98**	**59.39**	**60.18**	**98.7**	62.0	256M
	32K	PI 4×	CPT	24.83	46.32	23.12	64.62	37.00	70.62	35.94	37.46	42.49	69.7	0.010	4.25M
	32K	PI 4×	LongReD	24.20	28.78	23.45	63.94	37.40	69.26	35.37	39.08	40.19	65.9	0.010	4.25M
	32K	PI 4×	LinearARD	**51.77**	**63.30**	**34.20**	**87.21**	**43.56**	**81.83**	**39.36**	**58.65**	**57.49**	**94.3**	**60.2**	4.25M

RULER Hsieh et al. [2024]. For all three backbones, evaluation is conducted at 8K, 16K, and 32K tokens and averaged across tasks and lengths (Appendix G). Additionally, we evaluate all three backbones across 13 tasks on the 32K-compatible subset of LongBench-E Bai et al. [2024]; Appendix M details the tasks, length groupings, and complete breakdown scores.

4.2 Main Results

Short-context restoration. Table 1 shows that PI scaling creates a large native-to-scaled mismatch: the short-text Avg. drops to 34.23/33.50/32.99 on LLaMA2-7B, LLaMA3-8B, and Mistral-7B-v0.1. With only 4.25M tokens, final-layer LinearARD recovers 93.1%, 94.2%, and 94.3% of the native Avg., respectively, while budget-matched CPT and LongReD recover only 65.9–78.6%. The gains are especially visible on geometry-sensitive tasks. For LLaMA2-7B, LAMBADA rises from 6.18 after PI scaling to 63.40 after LinearARD, and ARC-Challenge rises from 24.91 to 48.10. On LLaMA3-8B, LinearARD improves the 4.25M-token Avg. from 46.66/47.93 for CPT/LongReD to 57.45, narrowing the gap to the 61.01 native score. Similar recoveries appear on Mistral, where LinearARD reaches 57.49 Avg. while using roughly $60\times$ fewer tokens than the 256M-token baselines.

Long-context robustness and realistic evaluation. Under the constrained 4.25M-token budget, LinearARD achieves strong RULER performance across all backbones, scoring 63.0 on LLaMA2-7B, 67.8 on LLaMA3-8B, and 60.2 on Mistral-7B-v0.1. This matches or exceeds the 256M-token LongReD baseline, whereas budget-matched controls remain near zero, showing that standard language-modeling updates or hidden-state distillation fail to activate long-range retrieval at this small data scale. At the hardest 32K context, LinearARD consistently outperforms LongReD, for example lifting LLaMA2-7B from 41.8 to 47.4. Table 2 broadens this assessment to realistic document comprehension on LongBench-E: 4.25M LinearARD substantially improves overall scores from 13.22 to 22.65 on LLaMA2-7B, from 12.80 to 21.18 on LLaMA3-8B, and from 4.78 to 17.17 on Mistral-7B-v0.1, matching 256M-token CPT baselines across both MHA and GQA architectures. While overall gains are substantial, minor length-dependent trade-offs occur, such as a slight dip from

Table 2: LongBench-E scores and training GPU-hours under PI scaling. Bold marks the best LongBench-E score per budget. Detailed breakdowns appear in Appendices M and C.

Backbone	Budget	LongBench-E (↑)		GPU-Hours (↓)	
		CPT	**LinearARD**	CPT	**LinearARD**
LLaMA2-7B	4.25M	13.220	**22.650**	0.545	0.661
	256M	22.940	**23.800**	31.593	30.998
LLaMA3-8B	4.25M	12.799	**21.184**	0.652	0.453
	256M	21.392	**22.799**	23.924	27.120
Mistral-7B-v0.1	4.25M	4.780	**17.167**	0.429	0.400
	256M	20.663	**21.796**	19.591	23.847

22.82 to 22.43 in the 4–8K group for LLaMA3-8B. Detailed task and length breakdowns appear in Appendices G and M.

Training efficiency and scaling transfer. As reported in Table 2, training GPU-hours under equal token budgets remain comparable to direct CPT, varying within -30.5% to $+21.7\%$ across all runs. Evaluated on a time-to-quality basis, however, LinearARD delivers dramatic compute savings: on LLaMA2-7B with Position Interpolation, 4.25M LinearARD reaches a LongBench-E score of 22.65 in 0.661 GPU-hours, whereas direct CPT requires 256M tokens and 31.593 GPU-hours to attain 22.94. This represents an empirical **47.8**$\times$ reduction in training GPU-hours. Furthermore, LinearARD transfers effectively beyond Position Interpolation. Under fixed-θ NTK-aware scaling on LLaMA2-7B, 4.25M LinearARD lifts RULER from 32.85 to 64.84 and improves LongBench-E from 21.36 to 22.11 over budget-matched CPT, with consistent gains holding at 256M tokens. These results confirm that relation-level restoration generalizes robustly across distinct RoPE scaling formulations. Complete compute breakdowns and scaling comparisons appear in Appendices C and N.

4.3 Ablation Studies

Table 3 compares distillation targets on LLaMA2-7B under PI $8\times$ without CPT. Full $Q/Q + K/K + V/V$ alignment achieves 56.05 Avg. and 93.64% recovery, outperforming attention-map distillation at 53.84. Omitting V/V drops Avg. to 54.27, confirming that value relations curb representation drift, while output-logit KL yields only a marginal gain to 57.00 at heavy memory cost. Layerwise diagnostics in Appendix E show that restoration concentrates in upper layers: aligning only the final layer Last-1 captures 89.83% recovery while running 10.9$\times$ faster than all-layer distillation, reducing step time from 252.61s to 23.22s at 32K context.

Tables 4 and 5 confirm a clear functional division between restoration and context activation. On LLaMA2-7B, ARD alone recovers short-context Avg. to 51.7, close to the 51.9 reached with CPT, but achieves only 36.8 on RULER. Adding the 2M-token CPT stage then surges RULER to 63.2. Consistent trends across GQA backbones show that restoring internal attention geometry first enables context extension with minimal long-sequence tokens.

5 Conclusion

In this paper, we presented LinearARD. By distilling Q/Q, K/K, and V/V relational distributions from a native-RoPE teacher, LinearARD directly rectifies the fine-grained positional distortions in the attention mechanism. Our analysis of training dynamics reveals a strong causal link between internal attention drift and output distribution shifts: aligning the internal relational structure implicitly restores the model's output proficiency, rendering explicit logit-level supervision redundant. To overcome the quadratic memory barrier, an IO-aware, linear-activation-memory kernel is introduced to enable exact dense distillation on ultra-long sequences. Empirical evaluations on LLaMA2-7B, LLaMA3-8B, and Mistral-7B-v0.1 show that LinearARD restores short-context performance with about $60\times$ fewer tokens than standard continued pre-training and provides strong long-context robustness. This work underscores the importance of maintaining structural integrity within the attention mechanism for efficient and effective context window extension.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under Grant 62301559, and the Beijing High-Creation Young Talent Support Project.

The authors declare no competing interests.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 2357–2367, 2019.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. 2024.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020.

bloc97. NTK-aware scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation. 2023. URL `https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have/`.

Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35: 16344–16359, 2022.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. Longrope: Extending llm context window beyond 2 million tokens. In *International Conference on Machine Learning*, pages 11091–11104. PMLR, 2024.

Zican Dong, Junyi Li, Jinhao Jiang, Mingyu Xu, Wayne Xin Zhao, Bingning Wang, and Weipeng Chen. Longred: Mitigating short-text degradation of long-context large language models via restoration distillation. *arXiv preprint arXiv:2502.07365*, 2025.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023.

Jinghan He, Haiyun Guo, Kuan Zhu, Zihan Zhao, Ming Tang, and Jinqiao Wang. Seekr: Selective attention-guided knowledge retention for continual learning of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3254–3266, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 4163–4174, 2020.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. Continual pre-training of language models. *arXiv preprint arXiv:2302.03241*, 2023.

Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, 2018.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1525–1534, 2016.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, 2020.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, 2021.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Dawei Zhu, Nan Yang, Liang Wang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. Pose: Efficient context window extension of llms via positional skip-wise training, 2024.

Appendix Roadmap

This supplement is organized to make it easy to locate experimental details, theory derivations, and diagnostic evidence:

- **Appendix A:** Detailed ablation results for ARD components and the CPT stage.
- **Appendix B:** Training configuration, exact token accounting, and evaluation protocol.
- **Appendix C:** Kernel benchmarks and complete training GPU-hour audit.
- **Appendix D:** Auxiliary objectives used in ablation studies.
- **Appendix E:** Layerwise distillation results and interpretation.
- **Appendix F:** Extreme-scaling short-context stress tests.
- **Appendix G:** RULER per-length and per-task breakdowns.
- **Appendix H:** Proof of Proposition 3.1.
- **Appendix I:** Proof of Theorem 3.2.
- **Appendix J:** Proof of Proposition 3.3.
- **Appendix K:** KL distillation backward-kernel pseudocode.
- **Appendix L:** Supplementary analyses and numerical verification.
- **Appendix M:** LongBench-E protocol and detailed results.
- **Appendix N:** PI and fixed-theta NTK configuration and results.
- **Appendix O:** Broader impacts.

A Detailed Ablation Results

This section provides comprehensive breakdowns of the ablation studies summarized in Sec. 4.3. To isolate the effects of individual algorithmic components without confounding variables, we conduct controlled ablations using fixed random seeds and identical training checkpoints.

Table 3: ARD component ablation on LLaMA2-7B under PI $8\times$ without CPT across all layers. Avg. is the mean of MMLU, LAMBADA, BoolQ, and OpenBookQA. Rec. is the recovery ratio relative to the native model.

PE	Variant	MMLU	LAMB.	BoolQ	OBQA	Avg.	Rec.
–	Native	45.99	71.18	78.23	44.00	59.85	100.00
PI $8\times$	w/o restore	24.65	6.18	58.29	36.20	31.33	52.35
	ARD (attn. variant)	31.63	65.17	76.76	41.80	53.84	89.95
	ARD (QKV rel.)	**36.81**	64.72	79.05	**43.60**	**56.05**	**93.64**
	ARD (+logit KL)	36.38	**68.85**	**79.17**	**43.60**	57.00	95.24
	ARD (full FT QKV)	30.42	58.66	75.38	43.60	52.02	86.91
	ARD (full FT LLM)	31.03	60.02	76.36	42.60	52.50	87.72
	ARD (w/o V/V)	33.44	61.15	79.11	43.40	54.27	90.68
	ARD (LoRA QKVO+MLP)	34.13	63.30	77.74	43.20	54.59	91.21

Table 4: Short-context ablation of the two training stages on LLaMA2-7B under PI $8\times$. ARD denotes relation distillation on PG19, and CPT denotes subsequent lightweight adaptation on SlimPajama.

		Short-text accuracy (%)								
PE	Method	MMLU	LAMB.	MathQA	BoolQ	OBQA	PIQA	SIQA	ARC-C	Avg.
–	Native	46.0	71.2	29.9	78.2	44.0	78.7	40.3	49.2	54.7
PI $8\times$	ARD (no CPT)	36.3	63.2	27.2	79.4	43.8	78.1	36.3	49.4	51.7
PI $8\times$	ARD + CPT	36.8	64.7	27.9	77.5	43.8	78.6	36.0	49.6	51.9

Table 5: Comprehensive stage attribution across model backbones under PI scaling. We report the 8-benchmark short-text accuracy (Short Avg.) and per-length RULER performance across 8K, 16K, and 32K context windows. Stage 1 (ARD) drives short-context restoration, while Stage 2 (CPT) unlocks long-range context retrieval.

Backbone	Condition	Tokens	Short Avg.	RULER Accuracy (%)			
				8K	16K	32K	Avg.
LLaMA2-7B	CPT only	4.25M	41.03	0.06	0.02	0.04	0.04
	ARD only	2.25M	51.70	46.50	42.50	21.40	36.80
	ARD + CPT	4.25M	51.90	74.90	67.40	47.30	63.20
LLaMA3-8B	CPT only	4.25M	46.66	0.05	0.04	0.04	0.04
	ARD only	2.00M	56.58	62.08	61.29	46.91	56.76
	ARD + CPT	4.25M	57.45	79.70	71.70	51.90	67.80
Mistral-7B-v0.1	CPT only	4.25M	42.49	0.01	0.01	0.01	0.01
	ARD only	2.00M	57.28	52.81	36.74	7.99	32.51
	ARD + CPT	4.25M	57.49	79.40	64.10	37.00	60.20

Stage attribution across backbones. To evaluate how Stage 1 (ARD) and Stage 2 (CPT) interact across diverse architectures, Table 5 provides a comprehensive stage-wise evaluation across LLaMA2-7B, LLaMA3-8B, and Mistral-7B-v0.1 under PI scaling. We compare Stage 1 alone (ARD only), Stage 2 alone (CPT only), and the complete two-stage pipeline (ARD + CPT). Across both standard MHA and Grouped-Query Attention (GQA) architectures, ARD alone recovers the overwhelming majority of short-context capabilities, while the lightweight CPT stage activates full long-range multi-needle retrieval across extended contexts. Conversely, applying CPT alone under this severely constrained token budget yields near-zero retrieval on RULER, demonstrating that internal relation restoration is the critical driver of sample-efficient context extension.

B Training Efficiency and Token Accounting

Tokens are reported as the number of non-padding tokens consumed during training. The training pipeline streams text data and packs it into fixed-length blocks at the stage context length, so each sample contains entirely valid tokens. The token budget is computed deterministically as:

$$N_{\text{tok}} = \sum_s L_s \cdot B_{\text{global},s} \cdot A_s \cdot U_s, \tag{8}$$

where L_s is the stage context length, $B_{\text{global},s}$ is the global batch size ($B_{\text{global},s} = G_s \cdot B_{\text{device},s}$ for distributed ARD across G_s GPUs, and set directly for CPT), A_s is the gradient accumulation factor, and U_s is the number of optimizer steps.

Budget labels use binary units (1M $= 2^{20} = 1{,}048{,}576$ tokens). The 4.25M budget corresponds to exactly $4{,}456{,}448$ tokens, which decomposes into $2{,}359{,}296$ tokens for Stage 1 (ARD) and $2{,}097{,}152$ tokens for Stage 2 (CPT), strictly matching the budget of direct CPT. The 256M baseline budget corresponds to $268{,}435{,}456$ tokens, yielding a $\sim 60.2\times$ token reduction for LinearARD. Table 6 provides the complete training token accounting across models and budgets.

Table 6: Exact token accounting for LinearARD and CPT baselines across budgets. Stage 1 uses PG19 and Stage 2 uses SlimPajama-627B. The combined two-stage LinearARD tokens strictly match the CPT baseline.

Model	Stage	Context	GPUs	Batch Size	Accum.	Steps	Exact Tokens
LLaMA2-7B (4.25M)	Stage 1 (ARD)	4,096	2	2 / device	1	144	2,359,296
	Stage 2 (CPT)	32,768	4	4 global	1	16	2,097,152
	Direct CPT	32,768	8	17 global	1	8	4,456,448
LLaMA3 / Mistral (4.25M)	Stage 1 (ARD)	8,192	2	1 / device	1	144	2,359,296
All Models (256M)	Stage 1 (ARD)	4K / 8K	2	8 / device	8 / 4	256	134,217,728
	Stage 2 (CPT)	32,768	4	4 global	16	64	134,217,728

Table 7: Single-operator latency for the naive dense implementation and the IO-aware kernel. The naive implementation runs out of memory (OOM) at 16K and above.

Seq. Len.	Naive (ms)	Kernel (ms)	Speedup	Naive Status	Kernel Status
1K	5.30	3.32	1.60×	OK	OK
2K	20.74	11.83	1.75×	OK	OK
4K	81.25	43.82	1.85×	OK	OK
8K	326.93	172.56	1.90×	OK	OK
16K	OOM	683.54	–	OOM	OK
32K	OOM	2549.54	–	OOM	OK

Table 8: End-to-end ARD training step time and throughput under different layer scopes. All-32 distills all layers, while Last-k distills only the final k layers.

Context	Step Time (s)						Throughput (tokens/s)					
	All-32	Last-16	Last-8	Last-4	Last-2	Last-1	All-32	Last-16	Last-8	Last-4	Last-2	Last-1
1K	0.77	0.62	0.54	0.50	0.49	0.48	1327.2	1655.6	1882.2	2034.6	2070.9	2144.0
2K	1.84	1.30	1.02	0.88	0.81	0.78	1113.8	1581.4	2005.7	2320.1	2519.2	2629.0
4K	5.50	3.47	2.45	1.94	1.68	1.56	744.7	1181.9	1672.5	2108.4	2431.5	2628.1
8K	19.08	11.09	7.11	5.11	4.11	3.61	429.4	738.4	1152.9	1602.2	1991.3	2267.1
16K	76.32	38.35	22.53	14.64	10.69	8.70	214.7	427.2	727.2	1119.3	1532.6	1882.9
32K	252.61	143.53	78.03	46.70	31.04	23.22	129.7	228.3	419.9	701.7	1055.7	1411.4

Training and Evaluation Settings. Stage 1 (ARD) trains QKV LoRA ($r = 1024, \alpha = 2048$) using AdamW (base LR 2×10^{-5}, cosine decay, warmup of 4 steps, clipping at 5.0) with $\lambda_q = \lambda_k = \lambda_v = 1.0$ and temperature 1.0. Stage 2 merges the adapter and runs 32K context continuation on SlimPajama using FlashAttention 2 and zigzag ring attention. Downstream short-context evaluations use lm-evaluation-harness (MMLU 5-shot, MathQA 4-shot, BoolQ 3-shot, ARC-Challenge 25-shot, others 0-shot), and long-context evaluations use RULER 0-shot at 8K, 16K, and 32K across all 12 tasks with truncation disabled.

C Kernel and Training Throughput

This section provides single-operator kernel benchmarks, layer-scope training throughput, and the complete training GPU-hour audit supporting Section 4.2 and Table 2. All latency and throughput benchmarks are evaluated on NVIDIA RTX PRO 6000 GPUs (96GB memory) paired with Intel Xeon Platinum 8470Q processors, averaged over 20 training steps.

Operator Latency and Memory Feasibility. Table 7 compares the single-operator execution latency between a standard dense implementation and our IO-aware tiled kernel. While the naive implementation exhausts GPU memory at sequence lengths of 16K and beyond, our kernel scales to 32K context with linear activation memory. Furthermore, by avoiding the materialization and repeated HBM transfer of $n \times n$ probability matrices, the fused kernel achieves a $1.60\times$–$1.90\times$ speedup across all sequence lengths where dense calculation fits in memory.

Layer-Scope Training Throughput. Table 8 reports end-to-end training step times and throughput across context lengths and layer scopes on LLaMA2-7B. Distilling dense relations across all 32 layers incurs heavy computational overhead, with the relation kernel accounting for up to 94.5% of training step time at 32K. Aligning only the final layer (Last-1) reduces step latency from 252.61s to 23.22s at 32K context—a **10.9**$\times$ throughput improvement—making final-layer distillation the highly efficient default.

Training GPU-Hour Audit and Compute Efficiency. For a training pipeline comprising stages s with G_s GPUs, U_s optimizer updates, and mean step latency τ_s seconds, total device compute is measured as:

$$\text{GPU-hours} = \frac{1}{3600} \sum_s G_s \cdot U_s \cdot \tau_s. \tag{9}$$

Table 9: Complete training GPU-hour audit across model backbones and budgets under PI scaling. Costs correspond to the LongBench-E evaluations in Table 2.

Backbone	Budget	LinearARD (GPU-h)	Direct CPT (GPU-h)	Cost Difference
LLaMA2-7B	4.25M	0.661	0.545	+21.3%
	256M	30.998	31.593	–1.9%
LLaMA3-8B	4.25M	0.453	0.652	–30.5%
	256M	27.120	23.924	+13.4%
Mistral-7B-v0.1	4.25M	0.400	0.429	–6.8%
	256M	23.847	19.591	+21.7%

Table 9 reports the exact compute expenditure across models and budgets under PI scaling, corresponding to Table 2 in the main text. On LLaMA2-7B, the 4.25M LinearARD pipeline consumes 0.661 GPU-hours (Stage 1: 144 steps on 2 GPUs at 3.340s/step; Stage 2: 16 steps on 4 GPUs at 22.163s/step), whereas 256M direct CPT consumes 31.593 GPU-hours (128 steps on 4 GPUs at 222.139s/step).

As shown in Table 9, when evaluated under equal token budgets, LinearARD incurs compute requirements comparable to direct CPT (ranging from -30.5% to $+21.7\%$). When evaluated on a time-to-quality basis, however, 4.25M LinearARD achieves a LongBench-E score of 22.65 in 0.661 GPU-hours, matching the 22.94 score achieved by 256M CPT in 31.593 GPU-hours. This represents an empirical **47.8**$\times$ compute reduction ($31.593 \,/\, 0.661 \approx 47.8$).

D Auxiliary Objectives (Ablations)

To evaluate alternative distillation targets, two auxiliary objectives are defined for ablation studies.

Attention-Map KL. For a given attention head, let $\mathbf{q}_{m,i} \in \mathbb{R}^d$ and $\mathbf{k}_{m,j} \in \mathbb{R}^d$ denote the query and key representations at positions i and j from model $m \in \{t, s\}$. The scaled dot-product logits and row-wise attention probabilities are:

$$\mathbf{S}_m(i,j) = \frac{\mathbf{q}_{m,i}\,\mathbf{k}_{m,j}^\top}{\sqrt{d}} + \mathbf{M}(i,j), \tag{10a}$$

$$\mathbf{A}_m(i,j) = \frac{\exp(\mathbf{S}_m(i,j))}{\sum_{k=1}^{n} \exp(\mathbf{S}_m(i,k))}. \tag{10b}$$

The attention-map distillation loss aligns these distributions via forward KL: $\mathcal{L}_{\text{attn}} = \mathcal{L}_{\text{KL}}(\mathbf{A}_t, \mathbf{A}_s)$. In Table 3, the "ARD (attention)" variant replaces the Q/Q and K/K terms in Eq. (5) with $\mathcal{L}_{\text{attn}}$, while retaining the V/V relation constraint.

Output-Logit KL. Distillation on vocabulary logits $\mathbf{z}_m \in \mathbb{R}^V$ applies temperature-scaled KL divergence:

$$\mathcal{L}_{\text{logits}} = D_{\text{KL}}(\text{softmax}(\mathbf{z}_t/T) \parallel \text{softmax}(\mathbf{z}_s/T))\,. \tag{11}$$

In Table 3, "+logit KL" incorporates $\mathcal{L}_{\text{logits}}$ alongside the default relation objective in Eq. (5), while "w/o V/V" removes the value-side relation term.

E Layerwise Distillation

Figure 4 evaluates single-layer ARD across individual Transformer layers on LLaMA2-7B under PI $8\times$ without CPT. The earliest layers (1–4) provide minimal recovery signal, whereas restoration gains surge through layers 5–8 and plateau in the final layers. Aligning only the final layer ($\ell^\star = L$) captures 89.83% recovery in this setting (compared to 93.64% for all 32 layers). As documented in Table 8, restricting distillation to the final layer reduces step time from 252.61s to 23.22s at 32K context—yielding a **10.9**$\times$ training speedup while retaining the vast majority of the restoration benefit.

Table 10: Short-context evaluation under extreme PI scaling factors on LLaMA2-7B without CPT. ARD alone restores over 88%–91% of native accuracy across extreme context extensions.

PE	Context	Method	MMLU	LAMB.	MathQA	BoolQ	OBQA	PIQA	SIQA	ARC-C	Avg.	Rec. (%)
–	4K	Native	45.99	71.18	29.92	78.23	44.00	78.73	40.28	49.23	54.69	100.00
PI 8×	32K	w/o restore	24.65	6.18	19.73	58.29	36.20	69.31	34.60	24.91	34.23	62.60
PI 8×	32K	ARD alone	36.81	64.72	27.94	77.54	43.80	78.62	35.98	49.57	51.87	94.85
PI 16×	64K	w/o restore	23.22	0.66	19.53	44.04	32.60	63.71	34.54	24.83	30.39	55.57
PI 16×	64K	ARD alone	32.72	59.89	26.26	77.55	42.20	77.53	36.30	49.15	50.20	91.79
PI 32×	128K	w/o restore	22.98	0.35	18.99	38.44	32.40	61.86	34.08	24.32	29.18	53.35
PI 32×	128K	ARD alone	30.05	55.75	25.76	74.34	43.00	77.42	36.15	46.84	48.66	88.98

Table 11: Per-length RULER scores for budget-matched CPT and LongReD controls (4.25M tokens) under PI scaling. The exact length-wise scores explain the collapsed retrieval behavior, and their aggregate mean (Avg. All) strictly corresponds to the main results in Table 1.

Backbone	Method (Budget)	8K	16K	32K	Avg. All
LLaMA2-7B	CPT (4.25M)	0.0625	0.0192	0.0383	0.040
	LongReD (4.25M)	0.0458	0.0292	0.0300	0.035
LLaMA3-8B	CPT (4.25M)	0.0475	0.0442	0.0375	0.043
	LongReD (4.25M)	0.0408	0.0400	0.0508	0.044
Mistral-7B-v0.1	CPT (4.25M)	0.0100	0.0108	0.0100	0.010
	LongReD (4.25M)	0.0150	0.0075	0.0075	0.010

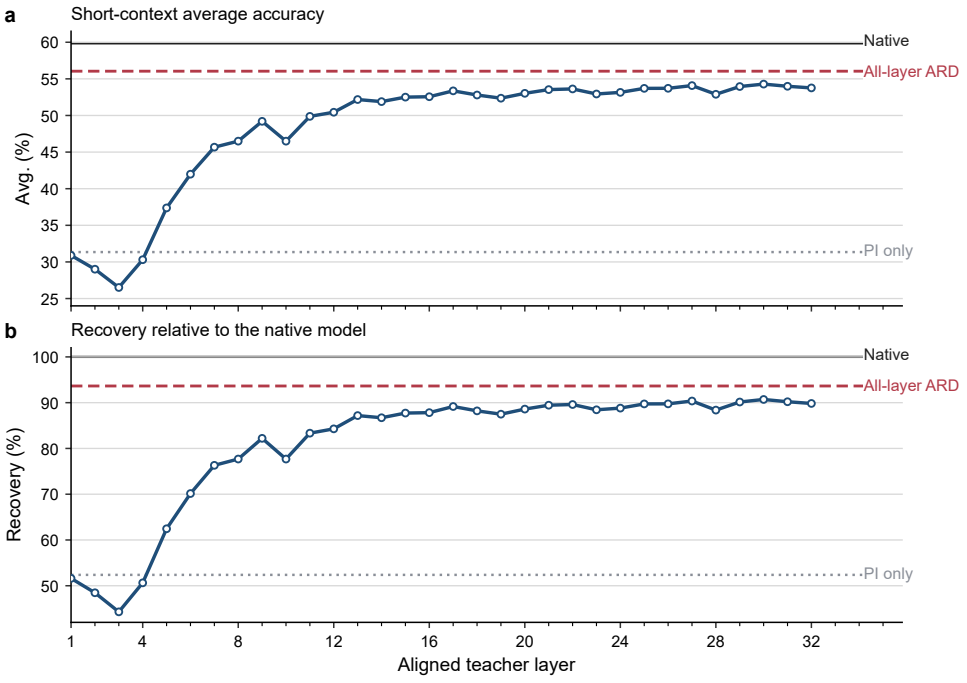

Figure 4: Layerwise restoration contribution of ARD on LLaMA2-7B under PI 8× without CPT. Deeper layers capture the dominant relational recovery signal, enabling highly efficient single-layer distillation.

Table 12: Per-task RULER accuracy (%) on LLaMA2-7B under PI $8\times$ evaluated at 8K, 16K, and 32K. Avg. reports the subtask mean at fixed length, while Avg. All reports the aggregate score across all tasks and context lengths matching Table 1.

Task	CPT (256M)			LongReD (256M)			LinearARD (4.25M)			LinearARD (no CPT)		
	8K	16K	32K	8K	16K	32K	8K	16K	32K	8K	16K	32K
NIAH-S1	95.0	91.0	82.0	95.0	94.0	88.0	100.0	100.0	100.0	97.0	90.0	76.0
NIAH-S2	95.0	96.0	78.0	95.0	97.0	86.0	100.0	100.0	96.0	80.0	74.0	30.0
NIAH-S3	91.0	84.0	55.0	95.0	91.0	60.0	88.0	88.0	43.0	45.0	52.0	15.0
NIAH-K1	93.0	90.0	62.0	95.0	94.0	63.0	96.0	96.0	76.0	66.0	58.0	25.0
NIAH-K2	76.0	68.0	23.0	76.0	60.0	20.0	74.0	70.0	29.0	47.0	43.0	13.0
NIAH-K3	22.0	10.0	0.0	26.0	11.0	0.0	27.0	0.0	0.0	1.0	0.0	1.0
NIAH-MQ	90.8	87.2	52.0	91.0	89.0	57.0	92.5	86.9	49.3	65.0	54.3	28.0
NIAH-MV	82.8	60.0	32.0	83.0	57.0	23.0	96.6	88.8	59.7	47.3	28.0	15.8
VT	84.8	37.6	30.4	86.0	31.0	13.0	59.5	43.0	39.1	10.8	11.2	0.0
CWE	36.0	34.0	41.4	36.0	25.0	36.0	43.7	32.3	16.6	52.2	45.1	27.9
FWE	67.3	61.0	29.0	68.0	60.0	39.0	63.5	66.1	43.9	15.7	21.7	7.7
QA(SQuAD)	48.1	44.1	15.4	50.0	42.0	16.0	53.5	34.9	16.1	31.0	33.0	18.0
Avg.	73.5	63.6	41.7	74.7	62.6	41.8	**74.5**	**67.2**	**47.4**	46.5	42.5	21.4
Avg. All		59.6			59.7			**63.0**			36.8	

F Extreme Scaling Short-Context Stress Test

To assess how relational distillation performs under extreme positional scaling factors, Table 10 evaluates short-context accuracy on LLaMA2-7B extended to 64K ($16\times$) and 128K ($32\times$) contexts using Stage 1 ARD alone (288 steps, 4.5M distillation tokens, without CPT). Even under extreme frequency compression, ARD restores short-context recovery from 55.57% to 91.79% at 64K and from 53.35% to 88.98% at 128K, demonstrating the structural stability of relation-level restoration across substantial scaling regimes.

G RULER Breakdown

Table 11 summarizes the per-length RULER scores for budget-matched CPT and LongReD controls. Under the restricted 4.25M-token budget, both baselines remain near zero across all evaluation lengths, confirming that language modeling loss or hidden-state distillation alone is insufficient to activate retrieval under severe data constraints.

Tables 12, 13, and 14 report detailed subtask breakdowns across 8K, 16K, and 32K context windows for LLaMA2-7B, LLaMA3-8B, and Mistral-7B-v0.1. On LLaMA2-7B, 4.25M LinearARD achieves strong 32K retrieval (47.4 vs. 41.8 for 256M LongReD), with major gains on multi-value retrieval (NIAH-MV: 59.7 vs. 23.0). On LLaMA3-8B, LinearARD attains a 32K average of 51.9, outperforming 256M LongReD (50.0).

H Proof of Proposition 3.1 (Gradient Behavior in Sparse Regimes)

Proof. Let $\mathcal{J} \subseteq \{1, \dots, n\}$ denote the visible key positions for a given query position i. For simplicity of notation, let $r_{t,j} \triangleq \mathbf{R}_t(i,j)$ and $r_{s,j} \triangleq \mathbf{R}_s(i,j)$ represent the row-wise teacher and student relation probabilities, where $r_{s,j} = \exp(z_{s,j}) / \sum_{m \in \mathcal{J}} \exp(z_{s,m})$ and $z_{s,j} \triangleq \mathbf{Z}_s(i,j)$.

The Jacobian of the row-wise softmax mapping satisfies:

$$\frac{\partial r_{s,j}}{\partial z_{s,k}} = r_{s,j}(\mathbf{1}_{j=k} - r_{s,k}). \tag{12}$$

Table 13: Per-task RULER accuracy (%) on LLaMA3-8B under PI $4\times$ evaluated at 8K, 16K, and 32K context lengths.

Task	CPT (256M)			LongReD (256M)			LinearARD (4.25M)		
	8K	**16K**	**32K**	**8K**	**16K**	**32K**	**8K**	**16K**	**32K**
NIAH-S1	99.0	100.0	99.0	89.0	89.0	91.0	99.0	99.0	99.0
NIAH-S2	100.0	100.0	99.0	86.0	84.0	62.0	96.0	96.0	88.0
NIAH-S3	100.0	98.0	94.0	81.0	73.0	63.0	66.0	71.0	64.0
NIAH-K1	99.0	95.0	93.0	84.0	82.0	56.0	98.0	90.0	73.0
NIAH-K2	99.0	80.0	60.0	92.0	83.0	66.0	95.0	96.0	61.0
NIAH-K3	90.0	55.0	20.0	75.0	57.0	23.0	29.0	22.0	11.0
NIAH-MQ	98.5	94.8	85.3	77.8	63.3	36.3	89.7	62.9	24.4
NIAH-MV	98.8	95.3	85.0	70.5	58.8	37.0	88.4	53.0	25.8
VT	99.8	96.2	28.6	98.0	97.0	62.2	93.1	86.6	70.0
CWE	78.6	62.5	35.4	73.2	55.4	10.2	79.2	66.7	29.1
FWE	88.7	85.0	72.3	80.7	85.0	66.6	67.3	70.1	58.5
QA(SQuAD)	53.1	60.1	29.3	49.4	59.1	26.7	56.1	47.2	18.4
Avg.	**92.0**	**85.1**	**66.7**	79.7	73.9	50.0	79.7	71.7	**51.9**
Avg. All		**81.3**			67.9			67.8	

Table 14: Per-task RULER accuracy (%) on Mistral-7B-v0.1 under PI $4\times$ evaluated at 8K, 16K, and 32K context lengths.

Task	CPT (256M)			LongReD (256M)			LinearARD (4.25M)		
	8K	**16K**	**32K**	**8K**	**16K**	**32K**	**8K**	**16K**	**32K**
NIAH-S1	100.0	100.0	73.0	94.0	88.0	50.0	93.0	91.0	51.0
NIAH-S2	100.0	100.0	60.0	91.0	90.0	63.0	96.0	89.0	64.0
NIAH-S3	89.0	68.0	53.0	85.0	70.0	46.0	73.0	49.0	53.0
NIAH-K1	98.0	93.0	62.0	95.0	90.0	45.0	92.0	88.0	54.0
NIAH-K2	80.0	40.0	3.0	96.0	93.0	49.0	94.0	92.0	54.0
NIAH-K3	29.0	4.0	0.0	64.0	13.0	4.0	34.0	22.0	3.0
NIAH-MQ	97.5	75.8	34.0	95.0	63.2	21.2	88.8	66.2	25.2
NIAH-MV	99.5	57.2	27.5	91.0	37.2	12.2	93.7	37.3	16.8
VT	90.2	53.4	9.4	93.4	63.4	43.4	100.0	61.0	41.2
CWE	48.6	28.5	0.1	77.1	51.1	26.5	65.3	50.6	24.5
FWE	7.7	72.6	20.3	79.7	92.7	59.0	72.3	82.8	40.6
QA(SQuAD)	58.1	40.1	17.4	52.4	40.4	16.3	49.8	40.8	17.1
Avg.	74.8	61.0	30.0	**84.5**	**66.0**	36.3	79.4	64.1	**37.0**
Avg. All		55.3			**62.3**			60.2	

Applying the chain rule to the row-averaged Mean Squared Error $\mathcal{L}_{\text{MSE}}^{\text{row}} = \frac{1}{2}\sum_{j\in\mathcal{J}}(r_{s,j} - r_{t,j})^2$ with respect to student logit $z_{s,k}$:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{MSE}}^{\text{row}}}{\partial z_{s,k}} &= \sum_{j\in\mathcal{J}}(r_{s,j} - r_{t,j})\frac{\partial r_{s,j}}{\partial z_{s,k}} = \sum_{j\in\mathcal{J}}(r_{s,j} - r_{t,j})r_{s,j}(\mathbf{1}_{j=k} - r_{s,k}) \\ &= r_{s,k}(r_{s,k} - r_{t,k}) - r_{s,k}\sum_{j\in\mathcal{J}} r_{s,j}(r_{s,j} - r_{t,j}) \\ &= r_{s,k}\left[(r_{s,k} - r_{t,k}) - \sum_{j\in\mathcal{J}} r_{s,j}(r_{s,j} - r_{t,j})\right]. \end{aligned} \tag{13}$$

For forward Kullback-Leibler divergence $\mathcal{L}_{\text{KL}}^{\text{row}} = \sum_{j\in\mathcal{J}} r_{t,j}\log(r_{t,j}/r_{s,j})$, since teacher probabilities $r_{t,j}$ are constant and $\sum_{j\in\mathcal{J}} r_{t,j} = 1$:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{KL}}^{\text{row}}}{\partial z_{s,k}} &= -\sum_{j\in\mathcal{J}} r_{t,j}\frac{\partial \log r_{s,j}}{\partial z_{s,k}} = -\sum_{j\in\mathcal{J}}\frac{r_{t,j}}{r_{s,j}}r_{s,j}(\mathbf{1}_{j=k} - r_{s,k}) \\ &= -r_{t,k} + r_{s,k}\sum_{j\in\mathcal{J}} r_{t,j} = r_{s,k} - r_{t,k}. \end{aligned} \tag{14}$$

Because $r_{s,j}, r_{t,j} \in [0,1]$, the bracketed term in Eq. (13) is strictly bounded. Thus, in the sparse regime where the teacher places probability mass on a critical token dependency ($r_{t,k} > 0$) but the student assigns vanishing probability ($r_{s,k} \to 0$):

$$\lim_{r_{s,k}\to 0} \frac{\partial \mathcal{L}_{\text{MSE}}^{\text{row}}}{\partial z_{s,k}} = 0, \qquad \lim_{r_{s,k}\to 0} \frac{\partial \mathcal{L}_{\text{KL}}^{\text{row}}}{\partial z_{s,k}} = -r_{t,k}. \tag{15}$$

Averaging over all n query positions scales both gradients by a constant factor of $1/n$, preserving this analytical property. □

Algorithm 2 Linear-Activation-Memory KL Distillation for Relation Distributions (Backward)

Input: Student $(\mathbf{X}_s, \mathbf{Y}_s)$, Teacher $(\mathbf{X}_t, \mathbf{Y}_t)$.
Output: Gradients $\nabla_{\mathbf{X}_s}\mathcal{L}, \nabla_{\mathbf{Y}_s}\mathcal{L}$.
Note: For QKV self-relations, set $(\mathbf{X}_m, \mathbf{Y}_m) \leftarrow (\mathbf{Q}_m, \mathbf{Q}_m)$, $(\mathbf{K}_m, \mathbf{K}_m)$, or $(\mathbf{V}_m, \mathbf{V}_m)$.
PHASE 1: GLOBAL STATISTICS (LINEAR HBM FOOTPRINT)
$\text{LSE}_s \leftarrow \text{ComputeLSE}(\mathbf{X}_s, \mathbf{Y}_s)$
$\text{LSE}_t \leftarrow \text{ComputeLSE}(\mathbf{X}_t, \mathbf{Y}_t)$
PHASE 2: FUSED BACKWARD PASS VIA TILING
Initialize gradient buffers in HBM: $d\mathbf{X}_s \leftarrow \mathbf{0}, d\mathbf{Y}_s \leftarrow \mathbf{0}$
for blocks of queries $\mathbf{X}_s^{(i)}, \mathbf{X}_t^{(i)}$ loaded to SRAM **do**
  for blocks of keys $\mathbf{Y}_s^{(j)}, \mathbf{Y}_t^{(j)}$ loaded to SRAM **do**
    $\mathbf{Z}_s \leftarrow \frac{1}{\sqrt{d}}\mathbf{X}_s^{(i)}(\mathbf{Y}_s^{(j)})^\top + \mathbf{M}^{(i,j)}$
    $\mathbf{Z}_t \leftarrow \frac{1}{\sqrt{d}}\mathbf{X}_t^{(i)}(\mathbf{Y}_t^{(j)})^\top + \mathbf{M}^{(i,j)}$
    $\mathbf{R}_s \leftarrow \exp(\mathbf{Z}_s - \text{LSE}_s^{(i)})$
    $\mathbf{R}_t \leftarrow \exp(\mathbf{Z}_t - \text{LSE}_t^{(i)})$
    $d\mathbf{Z} \leftarrow (\mathbf{R}_s - \mathbf{R}_t)/n$
    $d\mathbf{X}_s^{(i)} \mathrel{+}= \frac{1}{\sqrt{d}}\, d\mathbf{Z} \cdot \mathbf{Y}_s^{(j)}$
    $d\mathbf{Y}_s^{(j)} \mathrel{+}= \frac{1}{\sqrt{d}}\, d\mathbf{Z}^\top \cdot \mathbf{X}_s^{(i)}$
  end for
end for
// For self-relations ($\mathbf{X}_s = \mathbf{Y}_s$), total gradient is $d\mathbf{X}_s + d\mathbf{Y}_s$

I Proof of Theorem 3.2 and Complexity Analysis

Proof. We analyze the High Bandwidth Memory (HBM) consumption of the backward pass across sequence length n, batch size B, attention heads H, and head dimension d.

Standard Implementation: Standard autograd backpropagation retains full intermediate activation maps $\mathbf{R}_s \in \mathbb{R}^{B\times H\times n\times n}$ in HBM to evaluate loss gradients. At $n = 131{,}072$, storing a single full BF16 probability matrix requires $131{,}072^2 \times 2 \approx 34.36$ GB. Across multiple attention heads, batches, teacher/student copies, and three relation targets ($Q/Q, K/K, V/V$), the required activation storage scales as:

$$\mathcal{M}_{\text{std}} = \mathcal{O}(BHn^2), \tag{16}$$

which immediately exceeds the HBM physical capacity of modern enterprise accelerators (e.g., NVIDIA A100/H100 80GB).

IO-Aware Tiled Kernel: The proposed tiled kernel computes gradients on-the-fly in SRAM across blocks of size $T_r \times T_c$ (e.g., 128×128). In global HBM, the only retained activation buffers are:

- Row-wise log-sum-exp statistics ($\text{LSE}_s, \text{LSE}_t$): Size $2 \times B \times H \times n$.
- Accumulated tensor gradients ($d\mathbf{X}, d\mathbf{Y}$): Size $2 \times B \times H \times n \times d$.

All intermediate logit tiles $\mathbf{Z}$ and probability distributions $\mathbf{R}$ reside strictly in on-chip SRAM. Consequently, the HBM activation memory footprint is:

$$\mathcal{M}_{\text{LinearARD}} = \mathcal{O}(BHn + BHnd) = \mathcal{O}(BHnd), \tag{17}$$

which scales strictly linearly with sequence length n for fixed head dimension d. □

Arithmetic Complexity. While activation memory is reduced to $\mathcal{O}(n)$, arithmetic operations remain quadratic ($\mathcal{O}(BHn^2d)$, or $\mathcal{O}(B(H_Q + 2H_{KV})n^2d)$ for native GQA heads). Because LinearARD computes mathematically exact full-distribution KL divergence without sparse pruning or low-rank factorizations, both the forward log-sum-exp reduction and the backward recomputation visit every query–key tile block. The kernel renders exact dense relation distillation memory-feasible on ultra-long sequences, while preserving quadratic arithmetic complexity.

J Proof of Proposition 3.3 (Exactness)

Proof. For a fixed batch element and attention head, let:

$$\mathbf{Z}_s = \frac{\mathbf{X}_s \mathbf{Y}_s^\top}{\sqrt{d}} + \mathbf{M}, \qquad \mathcal{L}_{\mathrm{KL}} = \frac{1}{n} \sum_{i=1}^{n} D_{\mathrm{KL}}(\mathbf{R}_t(i,:) \parallel \mathbf{R}_s(i,:)) \,. \tag{18}$$

Differentiating the row-averaged loss with respect to student logits yields the exact gradient entry:

$$\mathbf{G}(i,j) \equiv \frac{\partial \mathcal{L}_{\mathrm{KL}}}{\partial \mathbf{Z}_s(i,j)} = \frac{\mathbf{R}_s(i,j) - \mathbf{R}_t(i,j)}{n}. \tag{19}$$

By the chain rule for the scaled bilinear form, the analytical gradients with respect to factor matrices are:

$$\frac{\partial \mathcal{L}_{\mathrm{KL}}}{\partial \mathbf{X}_s} = \frac{\mathbf{G}\mathbf{Y}_s}{\sqrt{d}}, \qquad \frac{\partial \mathcal{L}_{\mathrm{KL}}}{\partial \mathbf{Y}_s} = \frac{\mathbf{G}^\top \mathbf{X}_s}{\sqrt{d}}. \tag{20}$$

Algorithm 2 reconstructs local tile probabilities via $\mathbf{R}_m(i,j) = \exp(\mathbf{Z}_m(i,j) - \mathrm{LSE}_m(i))$, which shares the identical row-wise normalization as global softmax. For tile block (I, J), the kernel computes local gradient $\mathbf{G}^{I,J}$ and accumulates:

$$d\mathbf{X}_s^I \mathrel{+}= \frac{\mathbf{G}^{I,J}\mathbf{Y}_s^J}{\sqrt{d}}, \qquad d\mathbf{Y}_s^J \mathrel{+}= \frac{(\mathbf{G}^{I,J})^\top \mathbf{X}_s^I}{\sqrt{d}}. \tag{21}$$

By the distributive property of matrix multiplication, summing these partial tile updates across all blocks J and I yields the exact full-matrix products.

Furthermore, for self-relation targets where $\mathbf{X}_s = \mathbf{Y}_s = \mathbf{U}_s$ (as in $Q/Q, K/K, V/V$), both input branches accumulate into the identical representation tensor:

$$\frac{\partial \mathcal{L}_{\mathrm{KL}}}{\partial \mathbf{U}_s} = \frac{(\mathbf{G} + \mathbf{G}^\top)\mathbf{U}_s}{\sqrt{d}}. \tag{22}$$

Summing both accumulated gradient buffers yields exact analytical equivalence to full-matrix back-propagation. □

K KL Distillation Backward Kernel

Algorithm 2 outlines the fused backward pass for linear-activation-memory relation distillation. During Phase 1, row-wise log-sum-exp vectors $\mathrm{LSE}_s, \mathrm{LSE}_t \in \mathbb{R}^n$ are precomputed. Phase 2 executes tiled recomputation in SRAM without materializing full $n \times n$ matrices in HBM. For self-relation distillation ($Q/Q, K/K, V/V$), gradients flowing through both query and key argument roles are accumulated to form the full symmetric gradient.

L Supplementary Analyses and Numerical Verification

L.1 Attention–Logit Coupling

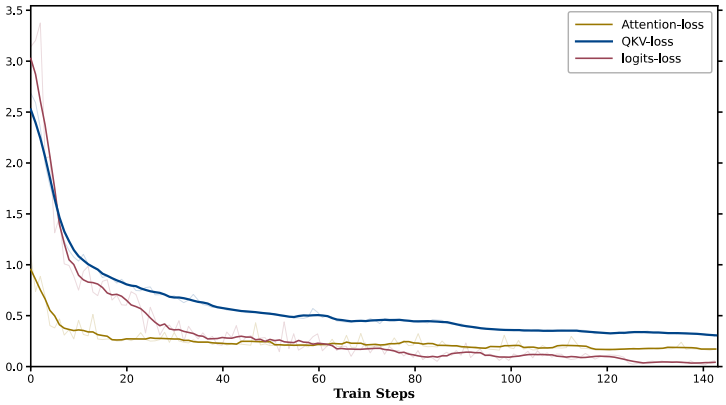

Figure 5: Training dynamics of distillation objectives. We optimize relation alignment (**QKV-loss**, blue) while monitoring unoptimized metrics (**Attention-loss**, gold; **Logit-loss**, maroon). Their concurrent decline provides strong empirical evidence that output distribution drift is closely associated with internal attention geometry distortion.

To investigate the internal restoration process, Figure 5 monitors intermediate representations during training. When optimizing exclusively on self-relation distributions (**QKV-loss**, decreasing from 2.69 to 0.30), unoptimized metrics decline concurrently: the standard attention-map loss drops from 1.09 to 0.16, and the output-logit KL collapses from 3.14 to approximately 0.05. This demonstrates a strong empirical coupling between internal relational geometry and output distribution alignment. Rectifying internal relational distortions within the attention mechanism enables the model to recover native output behavior without requiring explicit vocabulary logit supervision.

L.2 Kernel Numerical Verification

Table 15: Kernel numerical verification against a materialized dense baseline. Forward loss relative MAE is measured in FP32; backward gradient errors are evaluated in BF16 normalized by mean tensor magnitude.

Length	**Forward Loss Error**	**Backward Gradient Error**	
	Relative MAE ($\times 10^{-7}$)	Mean Relative ($\times 10^{-4}$)	Max Relative ($\times 10^{-2}$)
256	4.9	1.8	0.6
512	4.9	1.7	0.7
1024	4.7	1.5	0.8
2048	4.6	1.2	0.9
4096	4.9	1.0	1.0

While Proposition 3.3 proves analytical equivalence, block tiled reductions alter floating-point summation associativity. Table 15 quantifies numerical deviations against a materialized dense implementation across sequence lengths. The forward FP32 loss deviation remains bounded at 10^{-7} (on the order of machine epsilon $u_{32} = 2^{-24} \approx 5.96 \times 10^{-8}$), while BF16 backward gradient deviations remain strictly governed by the 7-bit fraction precision floor ($u_{b16} = 2^{-8} \approx 3.9 \times 10^{-3}$). These empirical measurements confirm that the tiled kernel achieves numerical agreement up to target floating-point rounding precision.

Table 16: LongBench-E performance across input-length intervals on GQA backbones extended from 8K to 32K ($4\times$ PI).

Backbone	Method	Tokens	Overall	<4K	4–8K	8K+
LLaMA3-8B	Direct CPT	4.25M	12.799	13.584	13.385	12.745
	LinearARD	4.25M	**21.184**	**22.245**	**21.658**	**20.805**
	Direct CPT	256M	21.392	20.522	**22.816**	21.179
	LinearARD	256M	**22.799**	**22.942**	22.433	**23.024**
Mistral-7B-v0.1	Direct CPT	4.25M	4.780	5.443	5.079	4.443
	LinearARD	4.25M	**17.167**	**15.861**	**16.439**	**17.440**
	Direct CPT	256M	20.663	17.877	21.169	20.689
	LinearARD	256M	**21.796**	**18.965**	**22.885**	**21.415**

M LongBench-E Protocol and Detailed Results

To evaluate broad long-context comprehension beyond synthetic retrieval, we benchmark models on 13 tasks from LongBench-E within the 32K context window: 2WikiMQA, GovReport, HotpotQA, LCC, MultiNews, MultiFieldQA-en, PassageCount, PassageRetrieval-en, Qasper, RepoBench-P, SAMSum, TREC, and TriviaQA. This suite spans single- and multi-document question answering, document summarization, code completion, passage retrieval and counting, and classification.

Following standard practice, evaluations are conducted zero-shot using official datasets. Input prompts are tokenized with each model's native tokenizer, and examples are retained when the prompt length plus the official maximum generation allowance fits within the 32,768-token limit. Table 17 details sample retention for LLaMA2-7B, where 3,572 of 3,668 examples (97.38%) are evaluated. Performance is macro-averaged across tasks and grouped into three input-length intervals: below 4,000 tokens, 4,000–7,999 tokens, and 8,000+ tokens.

Table 17: LongBench-E sample retention under the 32,768-token total sequence length limit for LLaMA2-7B.

Task	Total Examples	Evaluated (32K)	Excluded
2WikiMQA	300	300	0
GovReport	300	299	1
HotpotQA	300	300	0
LCC	300	269	31
MultiNews	294	289	5
MultiFieldQA-en	150	150	0
PassageCount	300	300	0
PassageRetrieval-en	300	300	0
Qasper	224	224	0
RepoBench-P	300	242	58
SAMSum	300	300	0
TREC	300	300	0
TriviaQA	300	299	1
Total	3668	3572	96

Table 18: LLaMA2-7B LongBench-E performance across input-length intervals under PI and NTK-aware scaling.

Scaling	Method	Tokens	Overall	<4K	4–8K	8K+
PI	Direct CPT	4.25M	13.22	10.25	13.94	13.21
PI	LinearARD	4.25M	**22.65**	**22.62**	**22.50**	**22.82**
PI	Direct CPT	256M	22.94	22.79	23.78	22.82
PI	LinearARD	256M	**23.80**	**26.19**	**23.86**	**23.72**
NTK	Direct CPT	4.25M	21.36	21.95	**24.33**	19.91
NTK	LinearARD	4.25M	**22.11**	**23.05**	22.98	**21.43**
NTK	Direct CPT	256M	22.96	22.45	**24.59**	21.97
NTK	LinearARD	256M	**23.19**	**24.96**	24.39	**22.03**

Table 18 and Table 16 present length-grouped evaluations. Across all three model architectures, 4.25M LinearARD delivers substantial gains over budget-matched CPT (LLaMA2-7B: 22.65 vs. 13.22; LLaMA3-8B: 21.18 vs. 12.80; Mistral-7B: 17.17 vs. 4.78), effectively matching or exceeding 256M direct CPT baselines with approximately $60\times$ fewer tokens.

M.1 LLaMA2-7B Task-Level Performance Breakdowns

Table 20 consolidates task-level performance across all 13 benchmarks for LLaMA2-7B. Under the 4.25M budget, LinearARD achieves remarkable recoveries on tasks severely degraded by linear interpolation, notably surging on TriviaQA ($12.35 \rightarrow 68.44$), TREC ($13.33 \rightarrow 55.67$), and SAMSum ($4.72 \rightarrow 22.03$).

N PI and Fixed-Theta NTK Scaling

To evaluate whether LinearARD transfers beyond linear Position Interpolation (PI), we benchmark against fixed-θ NTK-aware scaling on LLaMA2-7B extended from 4K to 32K context ($8\times$ extension).

Positional Configuration. For PI scaling, we apply linear position compression with factor 8.0 and maximum sequence length 32,768, scaling token position indices as $p/8$ while keeping base frequency $\theta_0 = 10{,}000$ constant. For fixed-θ NTK scaling, position indices remain unscaled while the rotary base frequency is adjusted statically based on head dimension $d = 128$:

$$\theta' = \theta_0 \times 8^{\frac{d}{d-2}} = 10{,}000 \times 8^{\frac{128}{126}} \approx 82{,}684.6226. \tag{23}$$

The maximum position embedding is configured to 32,768. The designated positional configuration is held constant throughout relation distillation, adapter merging, continuation training, and evaluation. The native teacher model strictly maintains unscaled RoPE.

Table 19: Comprehensive performance comparison between position interpolation (PI) and fixed-θ NTK-aware scaling on LLaMA2-7B across token budgets. Short Avg. reports the 8-benchmark short-context mean, LongBench-E denotes macro-averaged task accuracy on the 32K-compatible subset, and RULER reports aggregate retrieval accuracy across 8K, 16K, and 32K context windows.

Scaling Method	Training Method	Tokens	Short Avg.	LongBench-E	RULER
PI $8\times$	Direct CPT	4.25M	41.03	13.22	0.040
	LinearARD	4.25M	**50.93**	**22.65**	**63.00**
	Direct CPT	256M	52.20	22.94	59.60
	LinearARD	256M	**53.38**	**23.80**	**65.10**
NTK $8\times$	Direct CPT	4.25M	52.37	21.36	32.85
	LinearARD	4.25M	**53.75**	**22.11**	**64.84**
	Direct CPT	256M	53.40	22.96	64.53
	LinearARD	256M	**53.99**	**23.19**	**67.97**

Analysis of Scaling Transfer. As reported in Table 19, unscaled models modified with NTK scaling retain higher initial short-context accuracy than PI (51.66 vs. 34.23 without restoration) due to preserving high-frequency rotary dynamics. Despite these distinct mathematical starting points, LinearARD consistently improves performance across both scaling methods and training budgets. Under 4.25M tokens, LinearARD lifts NTK RULER performance from 32.85 to 64.84, while simultaneously enhancing LongBench-E from 21.36 to 22.11 and short-context accuracy from 52.37 to 53.75. At the 256M budget, LinearARD maintains consistent advantages across all benchmarks. These results confirm that internal attention relation restoration generalizes robustly across materially distinct RoPE scaling schedules.

Table 20: Task-level LongBench-E evaluation on LLaMA2-7B across positional scaling configurations, training methods, and token budgets. LinearARD substantially improves accuracy on retrieval, question answering, and reasoning tasks under constrained budgets while maintaining competitive performance at scale.

Task	**Position Interpolation (PI $8\times$)**				**NTK-Aware Scaling (NTK $8\times$)**			
	4.25M Budget		**256M Budget**		**4.25M Budget**		**256M Budget**	
	CPT	**LinearARD**	**CPT**	**LinearARD**	**CPT**	**LinearARD**	**CPT**	**LinearARD**
2WikiMQA	4.91	8.59	8.14	7.90	8.85	8.01	8.04	7.21
GovReport	22.75	25.24	28.14	26.60	26.72	25.87	24.26	26.31
HotpotQA	6.18	8.26	8.43	8.35	7.35	8.44	8.75	8.49
LCC	34.54	26.83	29.54	31.86	33.07	28.25	23.09	27.05
MultiNews	16.90	21.01	20.36	19.92	18.74	18.26	19.24	19.95
MultiFieldQA-en	11.56	15.72	15.64	16.44	12.86	15.23	14.54	15.63
PassageCount	4.25	4.43	3.65	4.18	3.86	2.69	5.37	3.26
PassageRetrieval-en	4.19	5.55	6.77	8.14	4.03	3.44	6.30	6.69
Qasper	6.58	8.82	9.45	9.09	7.47	8.52	8.62	9.68
RepoBench-P	29.67	23.86	21.77	21.65	28.59	26.98	18.60	19.33
SAMSum	4.72	22.03	21.97	22.76	29.18	21.15	29.78	26.21
TREC	13.33	55.67	52.44	57.33	49.17	54.44	59.17	57.00
TriviaQA	12.35	68.44	71.91	75.16	47.74	66.10	72.74	74.66
Macro Avg	**13.22**	**22.65**	**22.94**	**23.80**	**21.36**	**22.11**	**22.96**	**23.19**

O Broader Impacts

LinearARD's primary broader impact is to make long-context adaptation more efficient and reproducible. By reducing the memory and token budget required to restore RoPE-scaled LLMs, the method can lower compute and energy costs, broaden access for academic groups and smaller organizations, and support useful applications such as long-document assistance, retrieval-augmented reasoning, technical-document analysis, and educational or scientific workflows. The main risks are those inherited from adapting capable base LLMs: improved efficiency could also be used in harmful deployments, and restored models may retain biases, privacy risks, or security weaknesses from the teacher model. Responsible use should follow the licenses and safeguards of the underlying models and data, avoid sensitive or personally identifiable training data without an appropriate basis, and include downstream checks for bias, toxicity, privacy leakage, and unsafe generation. This work does not release a new high-risk pretrained model, image generator, or scraped dataset.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

Justification: The abstract and introduction state the final-layer relation-alignment objective, the linear-activation-memory KL kernel, the evaluated backbones, token budgets, and main short- and long-context results.

Guidelines:

- The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [No].

Justification: The experimental design is intended to validate LinearARD on mainstream open-source model families (LLaMA and Mistral) and the representative 32K context-extension setting. While this focus on industry-standard settings clearly demonstrates the core algorithmic advantages in compute efficiency and performance restoration, this paper does not include a separate limitations section discussing generalization to other RoPE scaling variants such as NTK-Aware or YaRN, non-RoPE positional-encoding architectures, or more extreme lengths such as 128K+; these experimental boundaries are instead specified in Sec. 4.1.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: The theoretical statements are numbered in Sec. 3.2 and Sec. 3.3, with proofs provided in Appendices H, I, and J.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: The paper specifies model families, RoPE scaling factors, objectives, token accounting, optimization settings, benchmarks, and RULER evaluation lengths in Sec. 4.1 and Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The method, evaluation protocol, and training-token accounting are described in the paper and supplemental material; the code has been anonymously submitted in the supplemental material.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Sec. 4.1 reports the backbones, context-window settings, objectives, optimizer, learning rate, warmup/decay schedule, clipping, checkpointing, and benchmark suite.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No].

   Justification: The main LLM restoration experiments report deterministic benchmark scores and budget-matched baselines, but do not include multi-seed error bars or formal statistical significance tests because of the cost of repeating 7B/8B-scale long-context restoration runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Appendix C reports operator latency, end-to-end training throughput, kernel-time share, and memory-feasibility diagnostics across context lengths and layer scopes, which characterize the compute requirements of the reported restoration setting.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work uses open pretrained language models and public evaluation benchmarks, preserves anonymity in the manuscript, and does not introduce human-subject or safety-critical deployment experiments.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

Justification: Appendix O emphasizes efficiency, accessibility, and useful long-context applications, while also covering inherited LLM risks and responsible-use mitigations.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: The paper does not release a new high-risk pretrained model, image generator, or scraped dataset; therefore no additional release safeguards are applicable beyond the underlying assets' access terms.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No].

Justification: The existing models, methods, and benchmarks used in the experiments are credited through citations in Secs. 2 and 4.1, but this paper does not explicitly list the license and terms of use for each asset.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes].

    Justification: The anonymized supplemental code is a new submitted asset, and the paper documents the LinearARD objective, kernel algorithms, token accounting, and empirical behavior in the main text and appendices.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: The study does not involve crowdsourcing or human-subject experiments, so there are no participant instructions, screenshots, or compensation details to report.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A].

Justification: The study does not involve human-subject experiments or participant data, so IRB approval is not required for the reported work.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: The core method studies and restores pretrained LLM backbones, which are fully described as experimental subjects, but it does not use an LLM as an original or non-standard component of the research process.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.