NeurIPS 2025 – Large Language Models

Extending the Context of Pretrained LLMs by Dropping their Positional Embeddings

Yoav Gelberg[1,2], Koshi Eguchi[1],
Takuya Akiba[1], Edoardo Cetin[1]

[1]Sakana AI
[2]University of Oxford

December 16, 2025

Code: https://github.com/SakanaAI/DroPE

Lecture Structure

Overview

01

Introduction & Motivation

The context extension challenge and current limitations

02

Background

Transformers, self-attention, and positional embeddings

03

Role of Positional Embeddings

Why explicit positional information is crucial for training

04

Why RoPE Scaling Fails

Fundamental limitations of frequency scaling approaches

05

The DroPE Method

Dropping positional embeddings for zero-shot extension

06

Experimental Validation

Comprehensive evaluation across model sizes and tasks

Key Finding: DroPE enables seamless zero-shot context extension without expensive long-context finetuning

01

Introduction & Motivation

The challenge of extending language model context beyond pretraining sequence length

The Core Challenge

The Context Extension Problem

Why Pretraining is Limited

Quadratic complexity bottleneck: Self-attention requires $O(n^2)$ token-to-token operations, making pretraining at long sequence lengths computationally intractable at scale.

Modern LMs are typically pretrained on contexts of 2K-4K tokens, but many real-world applications require 10K+ tokens.

The Performance Degradation

When inference exceeds pretraining context, **performance drops sharply** due to out-of-distribution positional embeddings.

RoPE rotations become out-of-distribution at unseen sequence lengths, causing attention mechanisms to fail.

Impact on Real-World Applications

Document Summarization
Legal documents, research papers often exceed 10K tokens

Long-form Conversation
Multi-turn dialogue with extensive context history

Code Understanding
Large codebases with complex dependencies

Knowledge-intensive QA
Retrieval over extensive knowledge bases

Research Goal: Enable models to use contexts beyond pretraining length without additional long-context finetuning — achieving zero-shot context extension.

State of the Art

Current Solutions & Their Limitations

Long-Context Finetuning

Method

Continue training model on sequences of target length

Cost

Requires full training budget on long sequences

Limitation: Computationally expensive and requires access to training infrastructure

RoPE Scaling Methods

Examples

PI, NTK-RoPE, YaRN

Method

Rescale RoPE frequencies to avoid OOD rotations

Limitation: Still require finetuning and struggle with zero-shot generalization

Alternative Architectures

Examples

ALiBi, RNoPE-SWA, NoPE

Trade-offs

Performance and stability compromises

Limitation: Not widely adopted due to performance gaps

The Fundamental Gap

All existing approaches require additional training and fail to generalize out-of-the-box to long-context downstream tasks.

Even popular RoPE scaling methods like YaRN need expensive finetuning on long sequences and don't enable true zero-shot extension — they essentially crop the effective context window rather than truly extending it.

Our Contribution

DroPE: A Simple Yet Powerful Solution

Core Idea

Drop positional embeddings after pretraining to unlock zero-shot context extension without compromising original capabilities.

Challenge the conventional role of PEs as permanent components — treat them as transient training scaffolds that can be removed.

Observation 1:

PEs provide crucial inductive bias, significantly accelerating pretraining convergence

Observation 2:

Over-reliance on PEs prevents test-time generalization to unseen sequence lengths

Observation 3:

DroPE Procedure

1. **Standard Pretraining**
   Train transformer normally with RoPE embeddings
2. **Remove PEs**
   Drop all positional embeddings from every layer
3. **Brief Recalibration**
   Continue training at original context length
4. **Zero-Shot Extension**
   Generalize to unseen sequence lengths

Key Advantage

Cost

02

Background

Foundations of transformers,
self-attention, and positional embeddings

Foundations

Self-Attention Mechanism

Core Operations

Given representations $\mathbf{h}_1, \ldots, \mathbf{h}_t$, self-attention computes queries, keys, and values via linear projections:

$$q_i = W_Q h_i,\ k_i = W_K h_i,\ v_i = W_V h_i$$

Attention scores and weights are computed between all position pairs, then used to weight value vectors.

Mathematical Formulation

Attention Scores:

$$s_{ij} = (1/\sqrt{d_k})\ q_i^{\top} k_j$$

Attention Weights:

$$\alpha_{ij} = \mathrm{softmax}(s_{i1}, \ldots, s_{ii})_j$$

Output:

$$z_i = \sum_{j \le i} \alpha_{ij} v_j$$

Attention Flow

Q Queries → Determine what information to retrieve

K Keys → Label information for retrieval

V Values → Contain actual information to aggregate

Causal Masking

The softmax is taken over $j \le i$, implementing a causal mask that prevents tokens from attending to future positions.

This ensures autoregressive generation — each token can only depend on previous tokens, not future ones.

Rotary Position Embeddings (RoPE)

Core Idea

RoPE injects **relative positional information** by rotating queries and keys in 2D subspaces before computing attention scores.

Modified Attention Score:

$$s^{\text{RoPE}}_{ij} = (R_i q_i)^T (R_j k_j)$$

$$= q_i^T R_{j-i} k_j$$

Rotation Matrix

Each $R(\omega_m)$ is a 2×2 planar rotation acting on coordinate pair $(2m, 2m+1)$:

$$R(\omega) = [\cos(\omega)\ -\sin(\omega);\ \sin(\omega)\ \cos(\omega)]$$

The complete rotation matrix is block-diagonal with $d_k/2$ independent 2D rotations.

Frequency Parameterization

$$\omega_m = b^{-2(m-1)/d_k}$$

where $b = 10{,}000$ (standard base)

- ↓ **Low frequencies ($m \approx d_k/2$):** $\omega_m \approx 1/b$, slow rotations
- ↑ **High frequencies ($m \approx 1$):** $\omega_m \approx 1$, fast rotations

Why RoPE Became Standard

1. **Relative positions:** Attention depends on relative distance $j-i$, not absolute positions
2. **Linear interpolation:** Easy to extend to longer contexts
3. **Efficiency:** No additional parameters or memory overhead
4. **Stability:** Well-behaved during optimization

Context Extension

RoPE Scaling Methods

The Scaling Problem

Given Ctrain and target Ctest, define s = Ctest/Ctrain.

At position Δ relative distance, RoPE phase is $\phi_m(\Delta) = \omega_m \Delta$. Scaling introduces new frequencies $\omega'_m = \gamma_m \omega_m$.

Extension Factor:

$$s = C_{test} / C_{train}$$

e.g., 2×, 4×, 8× extension

Position Interpolation

$$\gamma^{PI}_m = 1/s$$

Uniform scaling: All frequencies scaled equally by 1/s.

Simple but degrades high-frequency patterns.

NTK-RoPE

$$\gamma^{NTK}_m = (1/s)^{\alpha_m}$$

$$\alpha_m = 2^{2(m-1)/d_k}$$

Non-uniform: Low frequencies scaled like PI, high frequencies preserved.

Better preservation of positional patterns.

YaRN

$$\gamma^{YaRN}_m = (1-\kappa_m)/s + \kappa_m$$

Interpolation: Smoothly interpolates between 1/s and 1 based on frequency.

Most popular method, requires finetuning.

Common Limitation: All scaling methods require additional finetuning on long sequences and still struggle with reliable zero-shot generalization to downstream tasks.

03

Role of Positional Embeddings in Training

Why explicit positional information is crucial for effective pretraining

Empirical Evidence

RoPE vs. NoPE: Performance Gap

The Paradox

NoPE transformers are **theoretically expressive enough** for sequence modeling, but **consistently underperform** RoPE.

Kazemnejad et al. (2023) proved NoPE can perfectly reconstruct positions, yet the empirical gap persists.

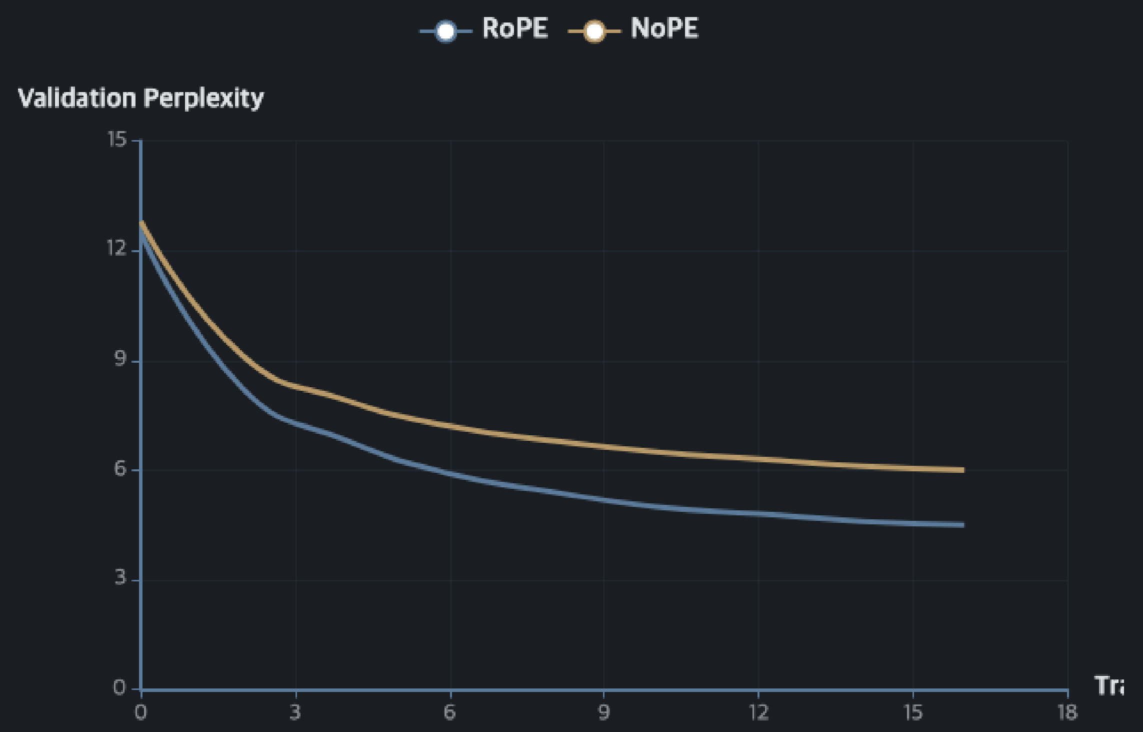
Experimental Setup

Architecture
500M parameter transformers

Dataset
16B FineWeb tokens

Context Length
2048 tokens

Training Loss Curves



Key Finding: NoPE maintains visibly worse perplexity throughout t

Theoretical Framework

Attention Positional Bias

Definition

Let **c_{ij}** be centered positional weights with $\Sigma_{j \leq i} c_{ij} = 0$.

$$A_c(\alpha) = (1/T) \Sigma_{i=1}^{T} \Sigma_{j \leq i} c_{ij}\alpha_{ij}$$

This measures **non-uniformity** in attention patterns — how much attention deviates from uniform distribution.

Common Patterns

Diagonal Head

$c_{ij} = 1$ if $i=j$, else $-1/(i-1)$

Focuses mass on current token position

Previous Token

$c_{ij} = 1$ if $j=i-1$, else $-1/(i-1)$

Captures immediate previous context

Why This Matters

$\|\nabla_\theta A_c\|$ bounds the rate at which non-uniform attention patterns emerge during training.

Higher gradients ⇒ faster development of positional bias ⇒ better learning of structured attention patterns.

Interpretation

Empirical Validation

Gradient Analysis: Why RoPE Learns Faster

Experimental Validation

Compare **gradient norms** of attention positional bias at initialization.

- ✓ **Diagonal heads** – placing mass on current token
- ✓ **Off-diagonal heads** – previous token attention

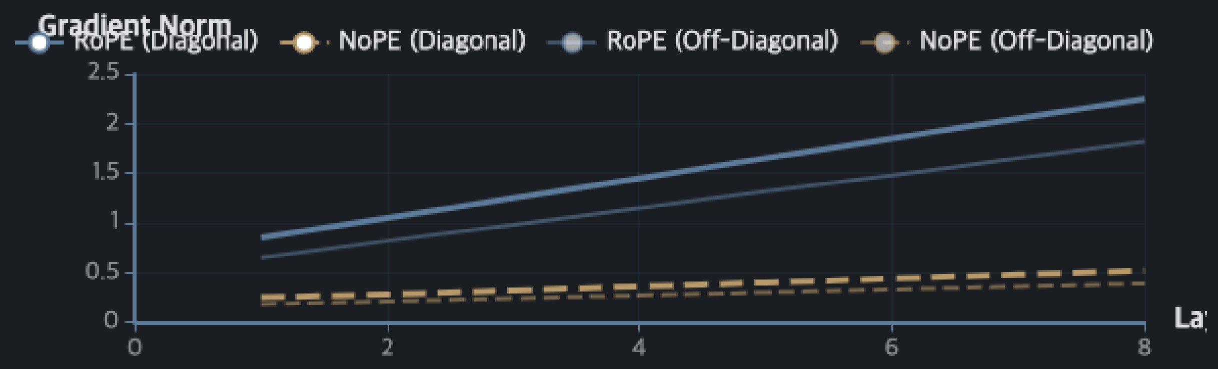
Key Finding

NoPE gradients are far lower than RoPE, with the gap growing in deeper layers.

Diagonal/off-diagonal heads develop slower in NoPE, reflecting difficulty recovering positional information.

Implication: NoPE can learn positional bias, but attention non-uniformity develops slowly due to bounded A_c gradients.

Gradient Norm Comparison



Diagonal Head Bias

RoPE shows consistently higher gradient norms across all layers

Enables rapid learning of current-token focus

Off-Diagonal Head Bias

Similar pattern for previous-token attention

Critical for capturing local context patterns

Early Training

RoPE: 3–5× higher gradients

Deep Layers

Gap increases with depth

Implication

Slower pattern learning

Theoretical Analysis: The NoPE Problem

1 Constant Sequences

On **constant sequences** ($x_1 = \ldots = x_t$):

NoPE Behavior:

- All attention heads are uniform
- Query/key gradients vanish
- $A_c = 0$, $\nabla_\theta A_c = 0$
- Output is constant

RoPE Behavior:

Non-zero A_c and gradients even on constant sequences

2 Embedding Uniformity

At initialization, embeddings have small variance ($\sigma^2 = 0.02$).

Define **prefix-spread**:

$$\Delta^{(l)}_h = \max \| h^{(l)}_i - h^{(l)}_j \|$$

For NoPE, if $\Delta^{(1)}_h \leq \varepsilon$, then $\Delta^{(l)}_h \leq C_1\varepsilon$ for all layers.

Theorem: Uniformity persists through the network

3 Bounded Gradients

For NoPE transformers with small initial spread:

$|A_c| \leq C_2\varepsilon$

$\|\nabla A_c\| \leq C_3\varepsilon$

Constants depend on architecture, not sequence length.

Intuition: Uniform embeddings ⇒ uniform attention ⇒ bounded gradients

Summary of Theoretical Results

NoPE Challenge: At initialization, embeddings are near-uniform with small variance. This uniformity persists through the network.

Uniform mixing cannot increase prefix spread, so attention remains approximately uniform.

RoPE Advantage: Circuits this issue by injecting explicit positional information, enabling non-zero gradients even on constant sequences.

Result: Faster development of positional bias and better training dynamics.

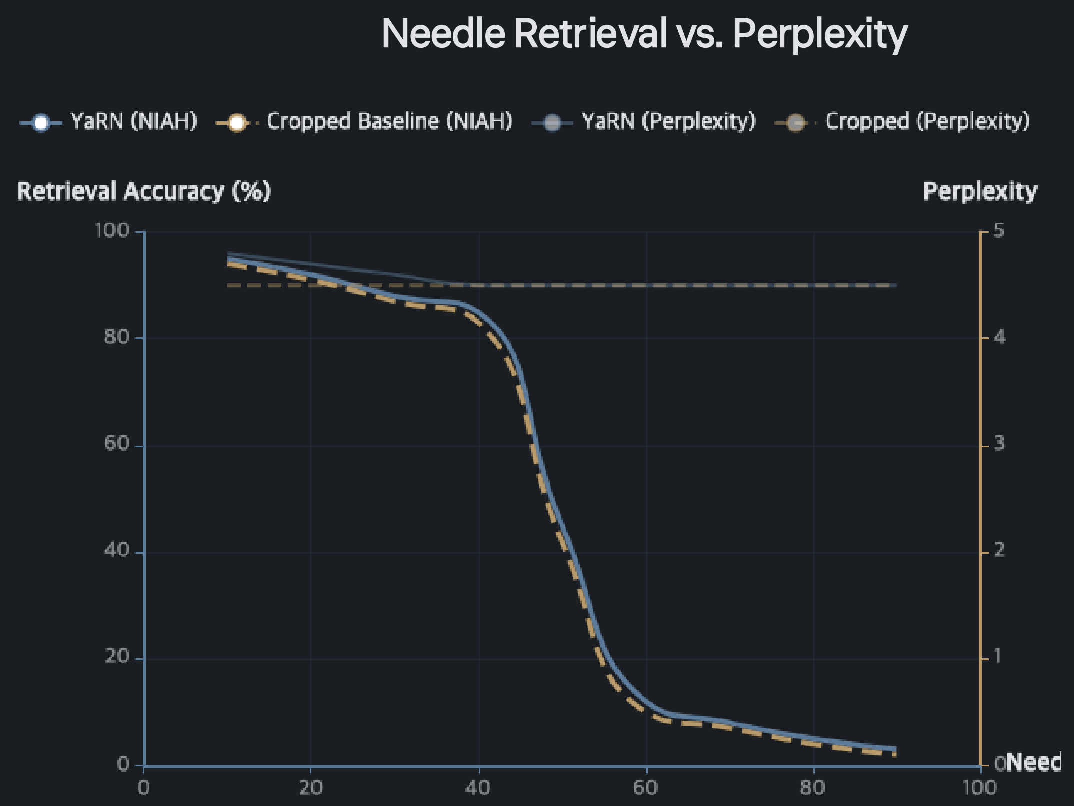
04

Why RoPE Scaling Fails

The fundamental limitations of frequency scaling approaches

Empirical Failure

YaRN's Failure Mode: Perplexity vs. Retrieval



The Paradox

YaRN **maintains perplexity** on long sequences but **fails at retrieval** when needles are beyond training context.

Performance closely matches simply cropping the input to the training context length.

Needle-in-a-Haystack

Task

Retrieve specific fact placed at varying depths

Metric

Success rate over 500 trials

Context

2× training length (e.g., 4K → 8K)

Why Scaling Fails

Frequency Scaling Analysis

Two Types of Attention Heads

Positional Heads

Use high frequencies

Patterns based on relative token positions (diagonal, previous-token)

Semantic Heads

Use low frequencies

Attend based on query/key content similarity

Scaling Impact

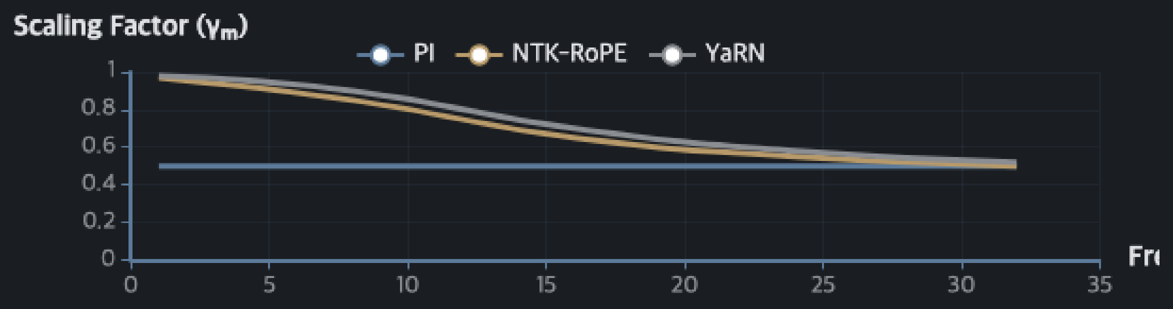
↑ **High Frequencies**

$\gamma_m \approx 1$ (nearly unchanged)

↓ **Low Frequencies**

$\gamma_m \approx 1/s$ (aggressively compressed)

Scaling Factor by Frequency



Positional Heads

Largely unaffected by scaling

High frequencies dominate, attention profiles remain si
milar

Semantic Heads

Substantially affected at long range

Low frequency compression warps semantic matchin
g

Critical Insight

Scaling warps low-frequency phases, shifting long-range attention in subspaces most used for seman
tic matching. This explains why YaRN fails at retrieval despite maintaining perplexity.

Theoretical Constraint

Why Compression is Inevitable

The Phase Constraint

In standard RoPE, **low-frequency phases never complete a full cycle** over the original context.

$$\varphi_m(C_{train}) = \omega_m C_{train} < 2\pi$$

For $b=10^4$, $d_k=64$, ≥ 5 low frequencies have $\phi_m(C_{train}) < 2\pi$ even at $C_{train}=32K$.

The Inevitable Trade-off

1. **Keep unchanged**
   $\phi_m(C_{test})$ becomes OOD
2. **Compress frequencies**

The Fundamental Dilemma

> **Any scaling method must compress low frequencies.**
>
> But this compression, in turn, shifts attention weights at long relative distances.

The relative phase $\phi_m(\Delta)$ is larger for distant tokens, so the $1/s$ scaling factor has a **greater effect**.

Mathematical Analysis

To keep phases in range:

$$\gamma_m \leqslant C_{train} / C_{test}$$

As extension factor s grows, this bound becomes increasingly small, requiring more aggressive compression.

Conclusion: This is a fundamental constraint of post-hoc RoPE scaling, not an implementation issue.

05

The DroPE Method

Dropping positional embeddings after pretraining for zero-shot context extension

The Solution

DroPE: Core Idea & Procedure

Key Insight

PEs are a transient training scaffold that can be removed aft er serving their purpose.

Harness the inductive bias exclusively during training, then drop for zero-shot generalization.

Why This Works

1. Model learns useful representations with RoPE
2. Removing PEs eliminates length constraints
3. Brief recalibration adapts to NoPE architecture
4. Zero-shot extension to unseen lengths

Step-by-Step Procedure

1. **Standard Pretraining**
   Train transformer normally with RoPE to convergence
2. **Remove PEs**
   Drop all positional embeddings from every layer
3. **Brief Recalibration**
   Continue training at original context length
4. **Zero-Shot Extension**
   Generalize to unseen sequence lengths

Works With

Any pretrained RoPE model in the wild

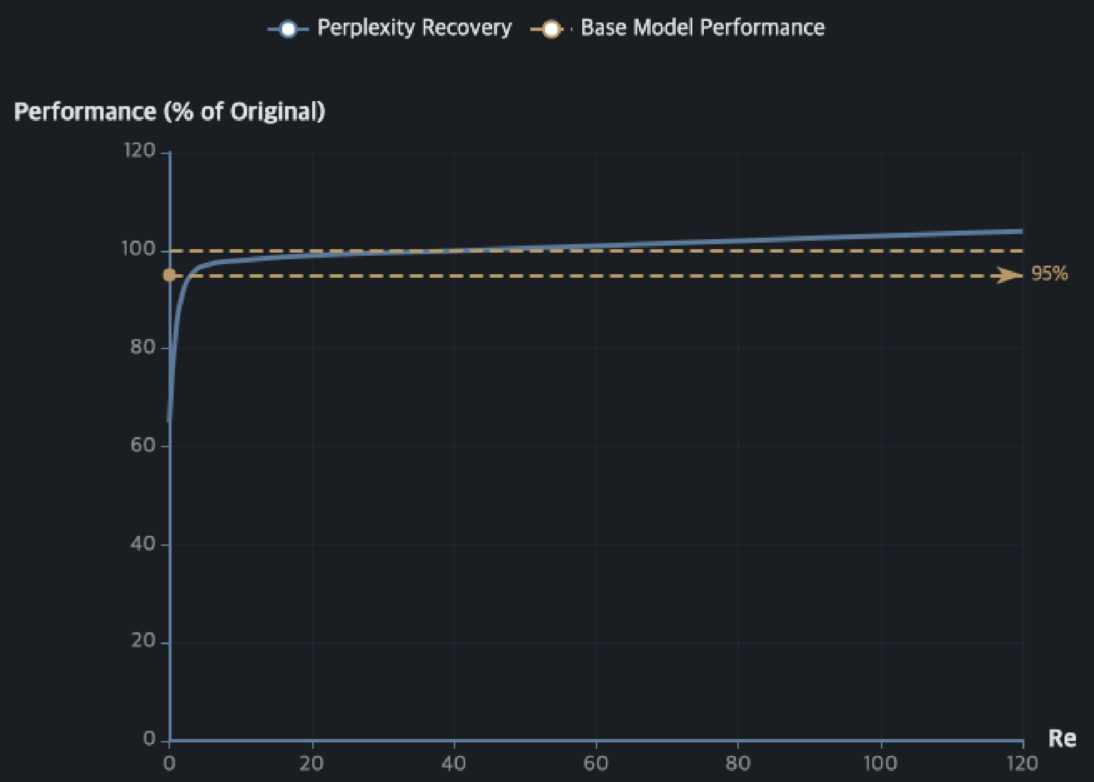
Cost

Minimal recalibration (2-5% of pretraining)

Training Dynamics

Recalibration Phase Analysis

Performance Recovery During Recalibration



Rapid Recovery

DroPE **recovers 95% of original performance in under 5B tokens** — jus t 0.8% of SmolLM's original 600B budget.

With extended recalibration (120B tokens), DroPE exceeds original performan ce.

Implementation Details

Stability

QKNorm added for training stability

Learning Rate

Schedule adjusted for extended training

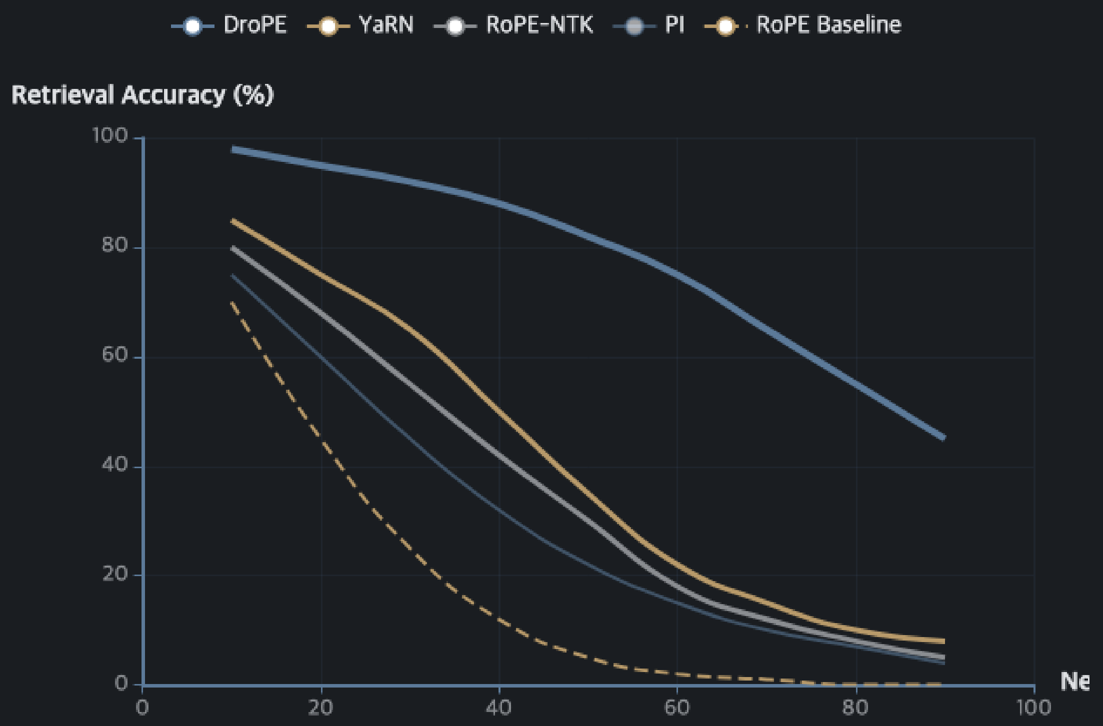
Data

Same data & hyperparameters as original

Breakthrough Result

Zero-Shot Context Extension Results

Needle Retrieval at 2× Context Length



Dramatic Improvement

DroPE achieves **seamless zero-shot context extension** without an
y long-context training.

Needle-in-a-haystack accuracy at 2× context length shows substantial gai
ns over all RoPE scaling methods.

Key Advantages

True Zero-Shot

No long-context finetuning required

Far Extension

Works at 2×, 4×, even 8× training length

Preserves Capabilities

No degradation in original context

Contrast with Scaling

06

Experimental Validation

Comprehensive evaluation across model sizes and tasks

Small-Scale Validation

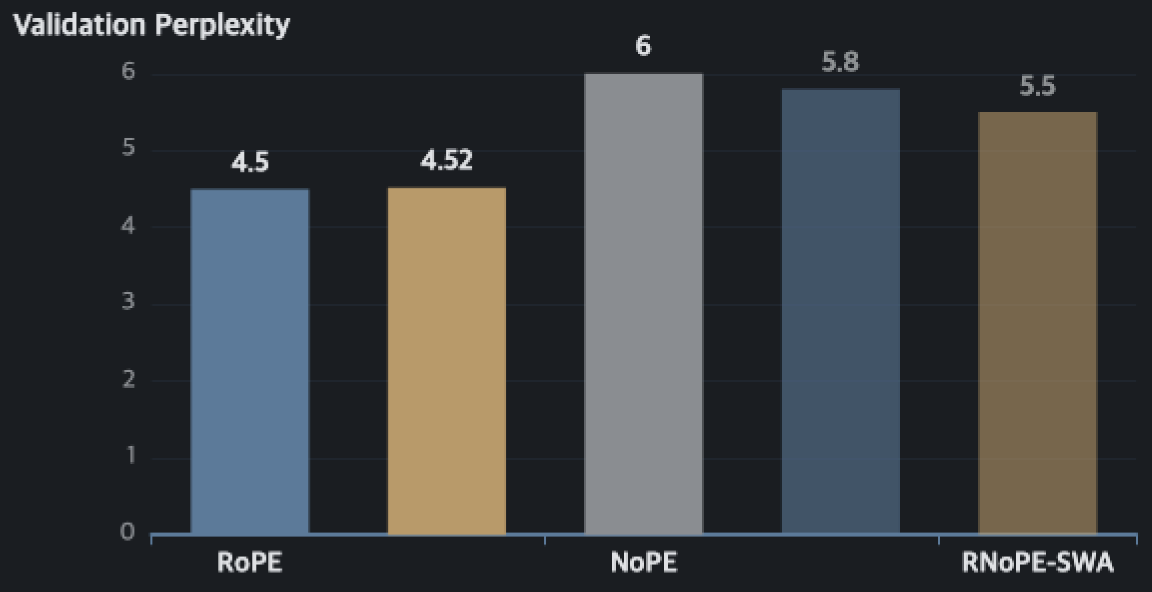
Small-Scale Experiments: 500M Models

Model Size
500M

Training Data
16B

- ✓ RoPE, NoPE, ALiBi, RNoPE-SWA
- ✓ DroPE: 14B RoPE + 2B recalibration

Final Validation Perplexity



DroPE Result
Matches RoPE's final perplexity

vs. NoPE
Clear edge over NoPE baseline

Real-World Application

Medium-Scale: Adapting SmolLM (360M)

Experiment Setup

Base Model

SmolLM-360M

600B pretraining tokens

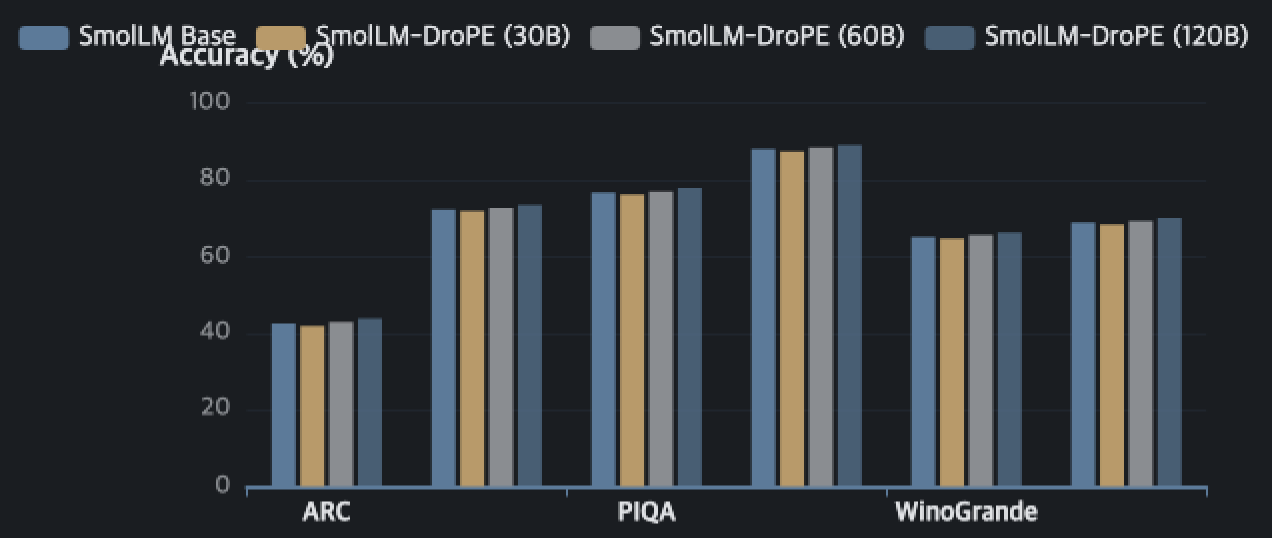
Context Length

2048 tokens

Recalibration Budgets

30B tokens	5% of original
60B tokens	10% of original
120B tokens	20% of original

LM Benchmark Performance Recovery

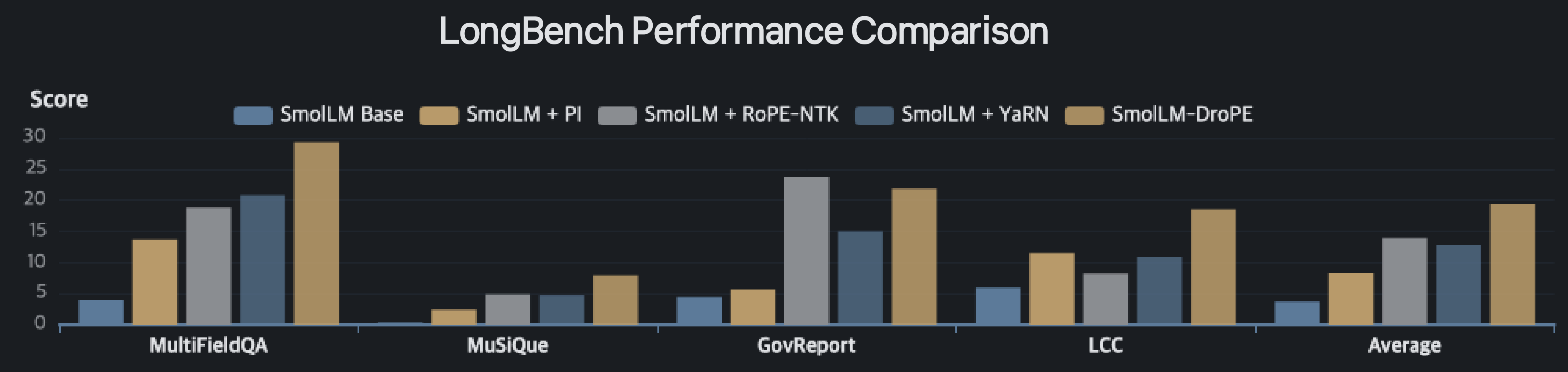


Key Result

Even with the shortest schedule, SmolLM-DroPE almost entirely matches base SmolLM on eve ry task. With the longest schedule, it **exceeds original performance**.

Long Context Benchmark

LongBench Evaluation Results



Tested Tasks

MultiFieldQA: Multi-document QA

MuSiQue: Multi-hop reasoning

GovReport: Document summarization

LCC: Legal case classification

Challenge

Tasks longer than **80× pretraining context** (160K+ tokens).

Even challenging for closed-source LMs.

Outcome

DroPE displays **clear edge** over prior approaches.

Gains far beyond all prior zero-shot RoPE extensions.

Scalability

Large-Scale: SmolLM-1.7B & Llama2-7B

Model Specifications

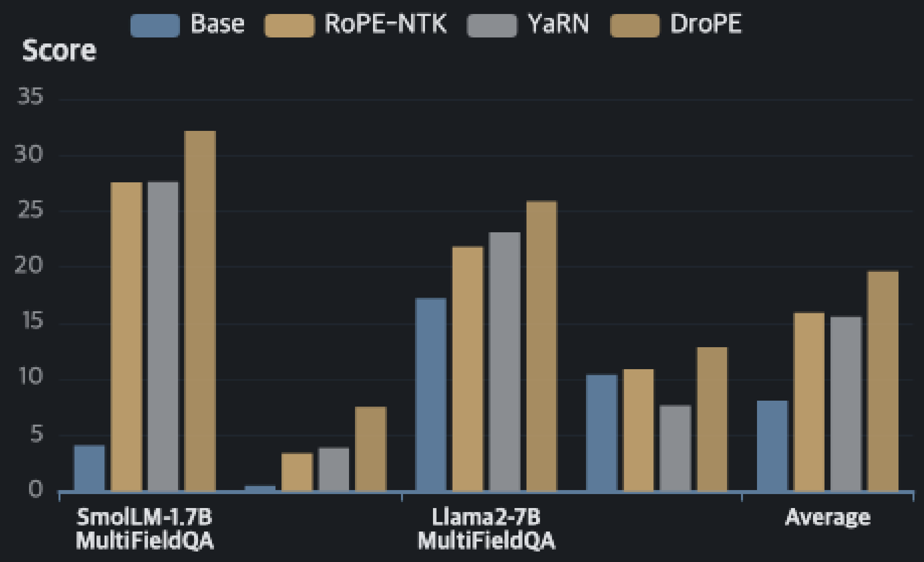
SmolLM-1.7B

Pretraining	1T tokens
Recalibration	20B tokens
Cost	2% of original

Llama2-7B

Pretraining	4T tokens
Recalibration	20B tokens
Cost	0.5% of original

Long Context QA Performance



Scalability Confirmed

DroPE consistently outperforms state-of-the-art RoPE scaling methods on long-context question-answering and summarization, providing strong evidence towards its **scalability and immediate potential**.

Retrieval Tasks

RULER Benchmark: Needle Retrieval Tasks

RULER Tasks

Multi-Query
Retrieve needles for several listed keys

Multi-Key
Retrieve needle for one specified key

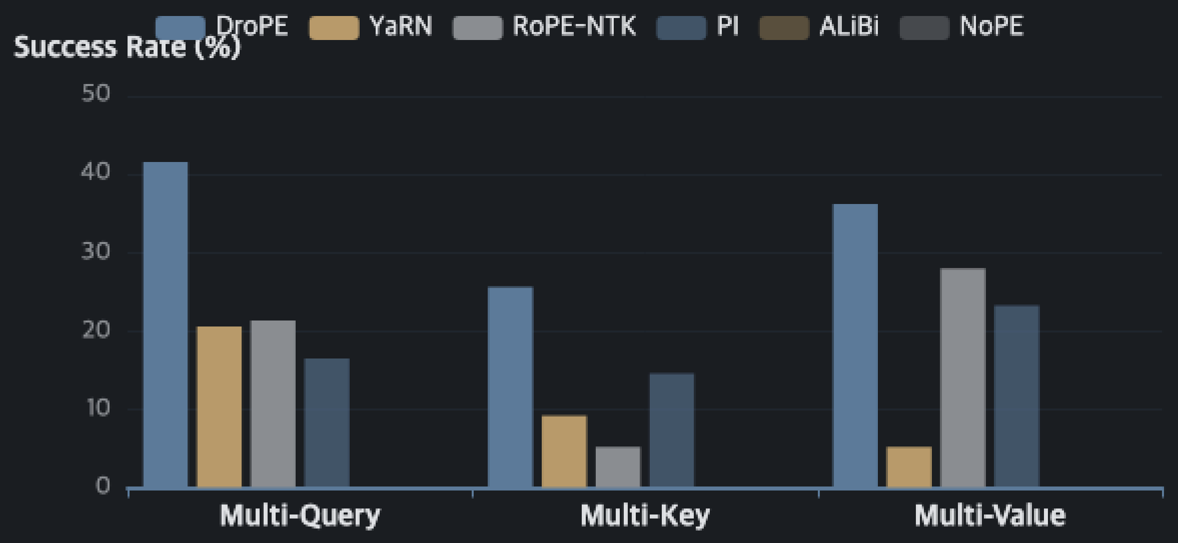
Multi-Value
Retrieve all needles for one key

500 Trials

Success rate measured over **500 trials per task** at 2× training c ontext length.

Evaluates robustness of retrieval across different needle placements an d query types.

Success Rate at 2× Context



Multi-Query
DroPE: 41.6% (vs. 20.6% YaRN)

Multi-Key
DroPE: 25.6% (vs. 9.2% YaRN)

Multi-Value
DroPE: 36.2% (vs. 5.2% YaRN)

Key Contributions & Future Directions

Reimagining Positional Embeddings

1 Training vs. Inference

PEs are **crucial for training** but **constrain zero-shot extension**. DroPE reconciles this by using different architectural choices at different stages.

2 Why Scaling Fails

RoPE scaling fails due to **inevitable frequency compression**. This fundamental constraint cannot be overcome by better scaling factors.

3 DroPE Solution

DroPE enables **effective zero-shot extension** at **minimal cost**, empowering arbitrary pretrained models.

Broader Implications

This work demonstrates that **canonical trade-offs in LM architecture design can be reconciled** by employing different architectural choices for different stages of training and inference pipelines.

Thank you

Code: https://github.com/SakanaAI/DroPE