Untie the Knots: An Efficient Data Augmentation Strategy for Long-Context Pre-Training in Language Models

Junfeng Tian[1*], Da Zheng[1*], Yang Chen[2], Rui Wang[3], Colin Zhang[1], Debing Zhang[1†]

rednote | hi lab

EAST CHINA NORMAL UNIVERSITY

DECILION

ACL 2025 VIENNA JULY 27 - AUGUST 1

Introduction

Why do we want long-context models

1. Long ducument QA / summarization
2. Retrieval-augmented generation (RAG)
3. Long Chain-of-Thought (CoT)
4. Agent (Repo-level code / Deep Research / Too Use)
5. MultiModal (Text / Image / Video / Audio)

Feature	GPT-4.1	Gemini 2.5 Pro
Context Window	1M tokens	1M tokens (2M coming soon)
Maximum Output Length	32K tokens	64K tokens
Multimodal Capabilities	Text, Images	Text, Images Audio, Video

1M tokens ≈ 1 hour Video ≈ 10 hour Audio ≈ 0.7M words Text

Source: Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context

How to obtain long-context models

Category	Examples	Limitations
Training-Free Extrapolation	(1) PI, YaRN (2) LM-infinite, StreamingLLM	Performance degrades significantly beyond the original training length. "Effective context" is limited.
Efficient Training Architectures	(1) Linear Attn, Sparse Attn (2) Ulysses, Ring Attn, CP	Computationally intensive. Still face a fundamental challenge...
The Core Challenge: The Data Bottleneck		(1) Scarcity: High-quality, naturally long documents (>32k) are rare. (2) Distribution Shift: Simple upsampling alters the data distribution, potentially harming short-context performance.
Our Solution: Untie the Knots (UtK)	`Chunking` -> `Shuffling` -> `Untying`	Efficient & Effective: Creates complex, pseudo-long contexts from existing data, forcing the model to learn long-range dependencies.

Motivation

Takeaway:
Instead of searching for scarce long data, we should augment existing data to explicitly train the model's ability for long-range information retrieval.

Problem: Data Scarcity

Naturally long, high-quality documents are extremely rare.

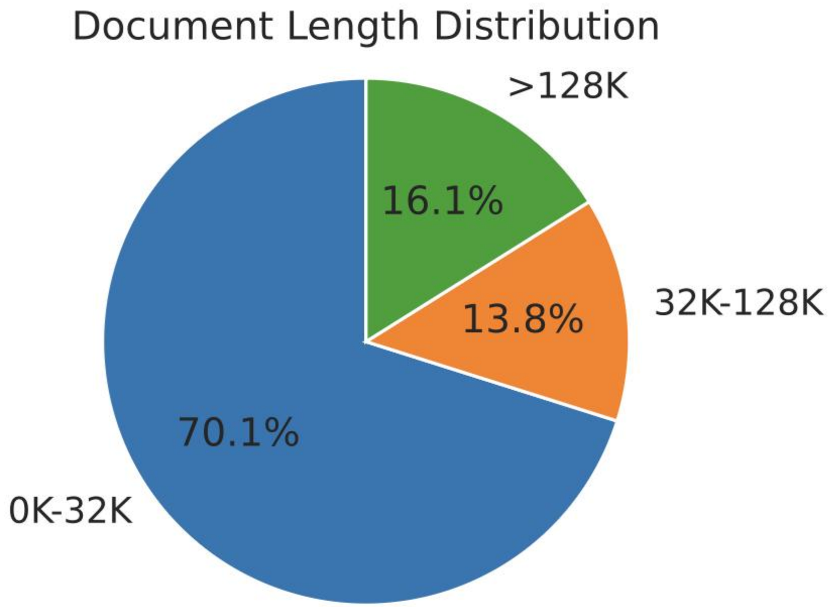


Figure: Document Length Distribution. Over 70% of tokens originate from documents shorter than 32K.

Goal: Train for Long-Range Attn

The key is to teach the model to attend to relevant information across long, distractor-filled contexts.

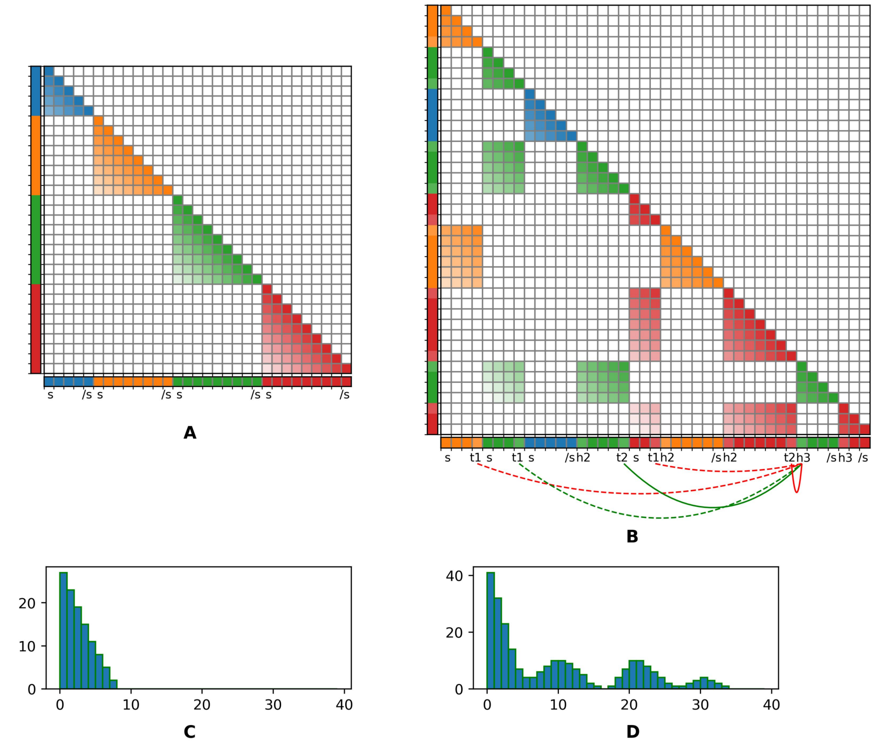
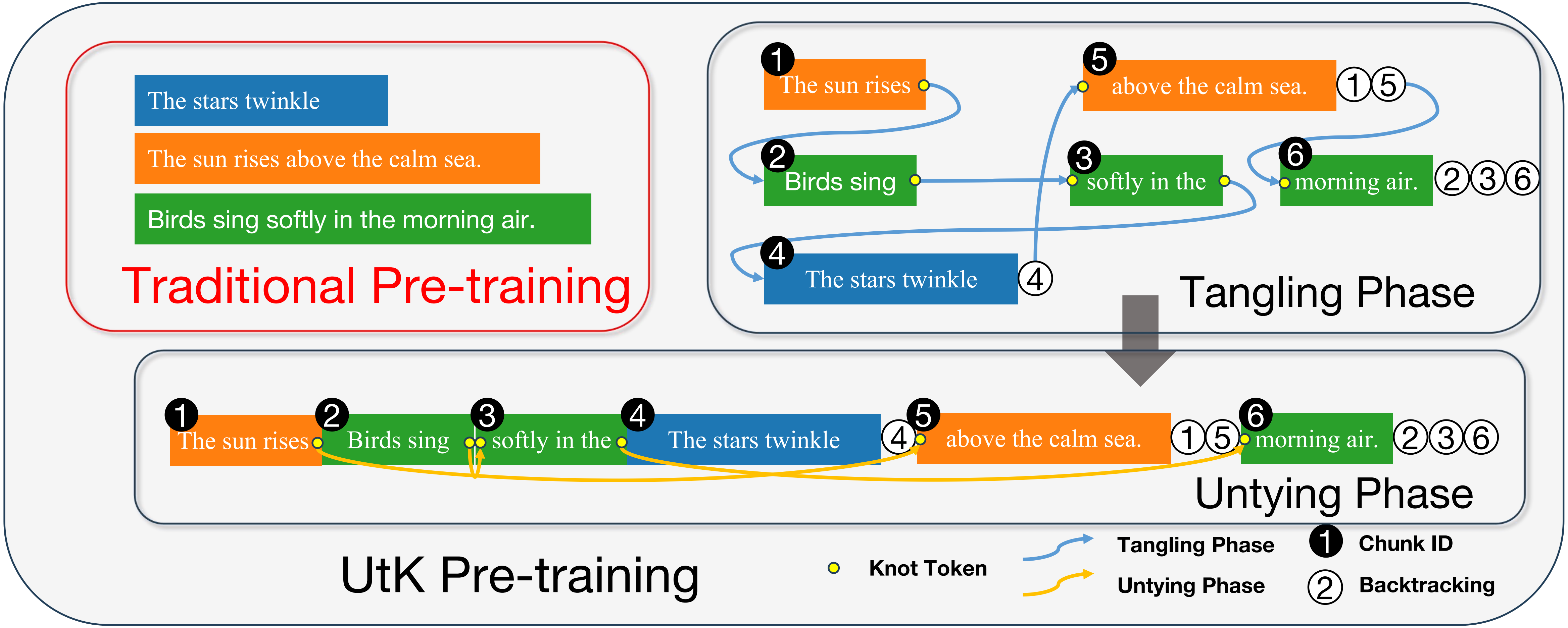


Figure: The Target Attention Pattern. We want to train the model to learn this complex, non-local attention, effectively 'untying' shuffled information.

Method Overview: Untie the Knots (UtK) 1/3

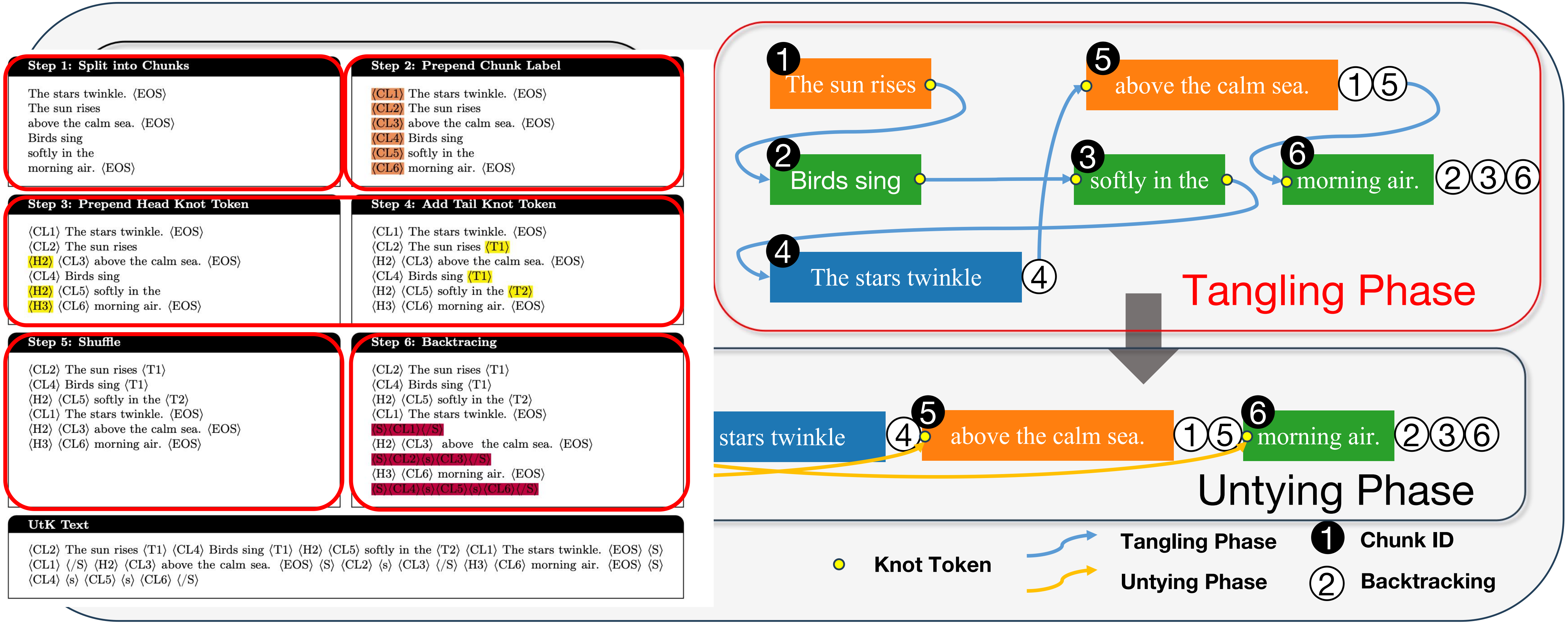


Traditional Pre-training:
Input: [Document A] [Document B], Learns local dependencies.

Original Text

The stars twinkle. ⟨EOS⟩ The sun rises above the calm sea. ⟨EOS⟩ Birds sing softly in the morning air. ⟨EOS⟩

Method Overview: Untie the Knots (UtK) 2/3



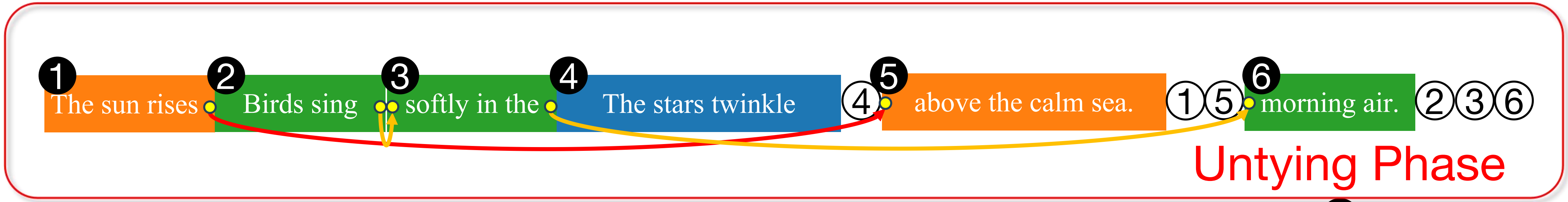
Tangling Phase (Data Augmentation):

A. Chunking: Documents are split into several segments.

B. Tying: Chunks from different documents are shuffled and "knotted" together into a single long sequence. Special [Knot] tokens and [Chunk IDs] are inserted.

C. Backtracing: the model must explicitly output the correct order of the chunks for a document.

Method Overview: Untie the Knots (UtK) 3/3



UtK Pre-training

Knot Token

Untying Phase

1 Chunk ID

2 Backtracking

Untying Phase (Training Task)

The model is trained on this "knotted" sequence. Its objective is three-fold:

A. Next Token Prediction: Standard language modeling.

B. Untie the Knots: The model must learn to attend to the correct preceding chunk (e.g., the red line) across thousands of distractor tokens.

C. Backtracing: Predict the original sequence of [Chunk IDs] at the end, forcing it to learn the full document structure.
②③⑥

Takeaway: UtK transforms standard pre-training into a challenging long-context reasoning task, forcing the model to develop robust long-range attention capabilities.

Results - UtK Delivers Superior and Scalable Performance

UtK Outperforms Existing Strategies

On the RULER benchmark, we compared UtK against various long-context training strategies on the Qwen2-7B model.

Models	32K	64K	128K
LONG-CONTEXT			
Qwen2-base (7B)[‡]	81.1	73.2	65.2
Qwen2-CT (7B)	78.2	73.6	54.8
Qwen2-ABF (7B)	78.9	75.2	65.9
Qwen2-Upsampling (7B)	80.7	76.4	67.4
Qwen2-AttnMask (7B)	80.4	75.6	72.0
Qwen2-Synthetic (7B)	83.2	80.5	72.7
Qwen2-CIP (7B)	80.5	76.3	71.0
Qwen2-UtK-base (7B)	80.5	79.2	**75.0**

At 128K context, our UtK model achieves 75.0%, significantly outperforming other methods like Upsampling (67.4%) and Synthetic data (72.7%).

UtK is a Scalable Solution

The benefits of UtK scale effectively across different model sizes (7B to 72B) and architectures (Qwen2, Llama).

7B MODEL Models	32K	64K	128K
Llama3.1-base (8B)[†]	**90.2**	80.4	66.1
Qwen2-base (7B)[‡]	81.1	73.2	65.2
Llama3.1-UtK-base (8B)[†]	88.8	**83.6**	73.8
Qwen2-UtK-base (7B)	80.5	79.2	**75.0**
70B MODEL			
Llama3.1 (70B)[§]	**94.8**	88.4	66.6
Qwen2 (72B)[§]	94.1	79.8	53.7
Llama3.1-base (70B)[†]	91.7	84.6	66.0
Qwen2-base (72B)[‡]	93.3	85.9	78.0
Qwen2-UtK-base (72B)	93.3	**90.6**	**84.5**

UtK boosts the 72B model to an impressive 84.5%. It also improves the already strong Llama3.1, showing its general applicability.

Discussion: What About Short-Context Performance?

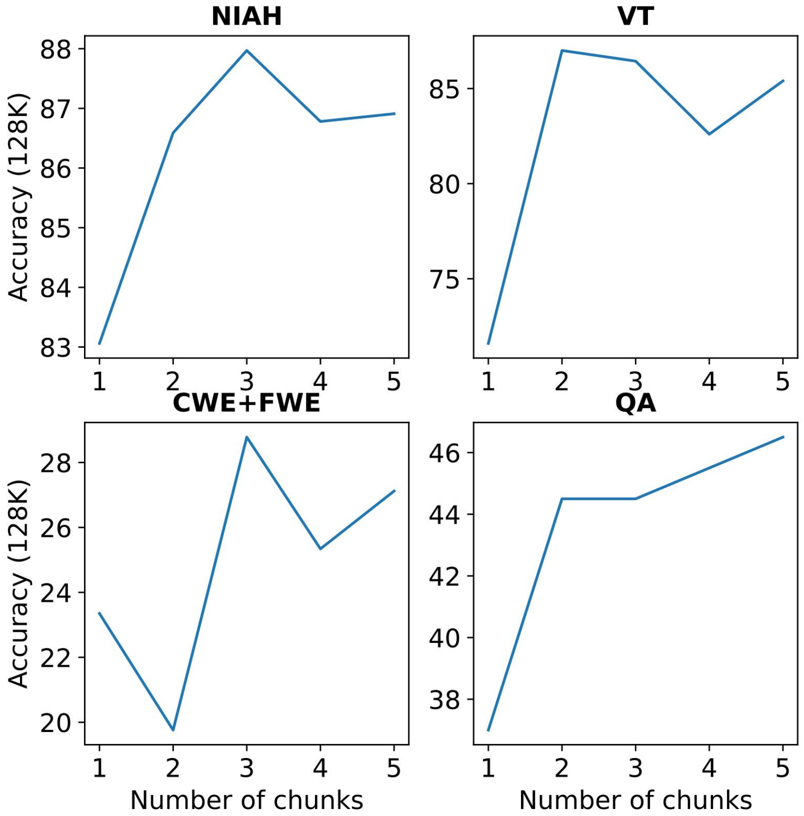
A critical question: Does training for long context hurt performance on standard, short-context tasks?

Models	Understanding			Code		Math	Avg.	Δ
	BBH 3shot	NQ 5shot	TriviaQA 5shot	HumanEval 0shot	MBPP 3shot	GSM8K 4shot		
Llama3.1-base (8B)	**63.9**	33.5	**80.2**	35.4	54.5	58.0	54.2	-
Llama3.1-UtK-base (8B)	61.9	**34.0**	79.6	38.4	54.6	59.3	54.6	+0.7%
Qwen2-base (7B)	61.4	30.3	70.2	**46.3**	64.6	**80.9**	59.0	-
Qwen2-CT-base (7B)	61.1	29.6	70.3	44.5	**66.2**	77.6	58.1	-1.5%
Qwen2-UtK-base (7B)	61.6	29.5	70.2	45.1	64.2	78.1	58.2	-1.4%
Llama3.1-base (70B)	**81.0**	**49.3**	**91.2**	59.2	72.8	82.3	72.6	-
Qwen2-base (72B)	79.8	45.6	88.0	**61.6**	**76.9**	**88.8**	**73.3**	-
Qwen2-UtK-base (72B)	80.6	45.0	87.6	61.0	75.9	87.8	73.0	-0.4%

No. UtK preserves short-context capabilities.

Ablation Study – Why Does UtK Work?

How many chunks are optimal?



Performance generally peaks at 2 or 3 chunks. Too few chunks (1) is just standard training. Too many chunks can make the task overly difficult, hindering learning.

Which components of UtK matter most?

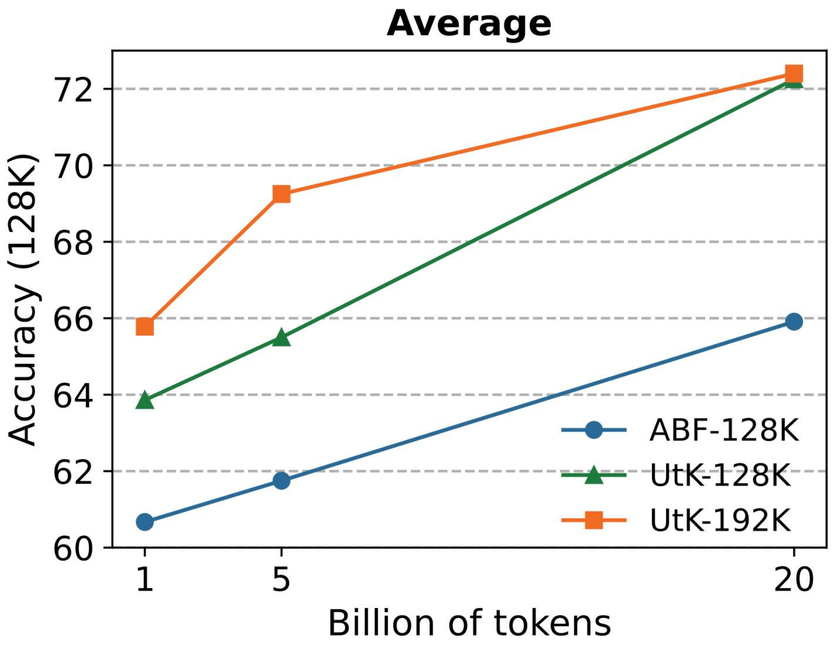
Models	Average	NIAH	VT	CWE+ FWE	QA
Qwen2-UtK-80%	**75.0**	90.3	**97.6**	**29.9**	**48.0**
- Disrupt order	73.0	90.4	97.8	17.4	46.0
- W/o backtracing	74.3	**91.3**	94.8	23.5	46.5
Qwen2-UtK-30%	73.1	88.8	94.8	28.5	44.0
- Disrupt order	72.3	89.0	89.8	21.7	47.0
- W/o backtracing	70.8	88.5	86.2	21.2	42.0

1. Backtracing is critical: Removing it causes a significant performance drop.
2. Preserving Order is important: Completely disrupting the order hurts performance, especially on aggregation tasks (CWE+FWE).
3. More is Better: A higher UtK probability (80% vs. 30%) yields better results.

Analysis - Efficiency and Interpretability

Training Efficiency: Faster and Better

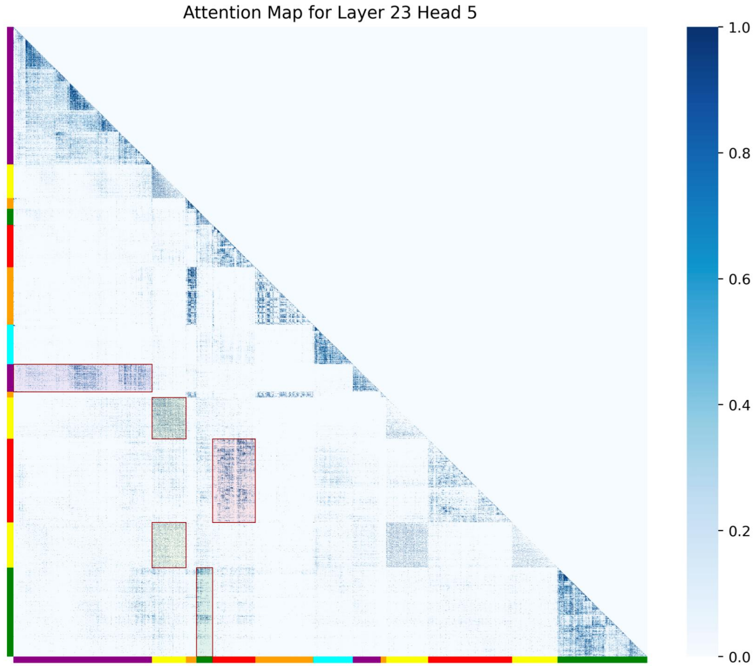
We tracked model performance on RULER 128K as a function of the number of training tokens.



With only 1 billion tokens, our UtK-192K strategy already matches the performance of the baseline trained on 20 billion tokens—a 20x improvement in data efficiency.

Attention Visualization: Learning to Untie

To understand how the model handles knotted context, we visualized its attention patterns.



The attention map clearly shows that the model learns to attend to non-adjacent but related chunks of the same document (highlighted in red boxes). It effectively 'untying the knots' as intended.

Conclusion

The path to better long-context models may not be through finding more long data, but by making better use of the data we already have.

1. Effective: SOTA Performance on RULER, LV-Eval, and InfiniteBench.
2. Efficient: 20x Data Efficiency
3. Safe: No Degradation on Short-Context Tasks

Models & Code Available at: https://github.com/rgtjf/Untie-the-Knots

We also trained and open sourced dots.llm1.base and dots.llm1.inst based on UtK.

1. A MoE model with 14B activated and 142B total parameters trained on high-quality corpus.
2. Intermediate model checkpoints are released spanning the entire training process.

dots.llm1

rednote | hi lab

THANKS