Distilling Sequential Computation in Transformer Language Models

Zixuan Lan[1] **Jessica Yang**[2] **Yanhong Li**[3] **Karen Livescu**[2] **Jiawei Zhou**[4]

[1]The University of Chicago [2]Toyota Technological Institute at Chicago
[3]Independent Researcher [4]Stony Brook University

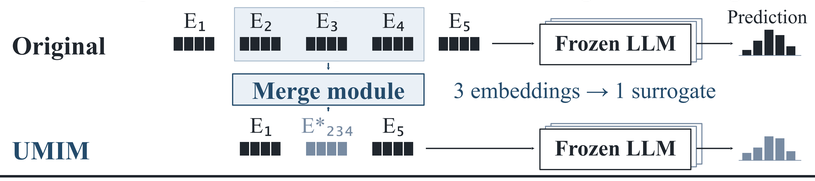
Code on GitHub

0 Motivation & Overview

LLMs generate one token at a time. Longer contexts increase attention computation and KV-cache memory requirements.

Can we compress prompts and generated contexts while preserving predictions, with the backbone LLM frozen?

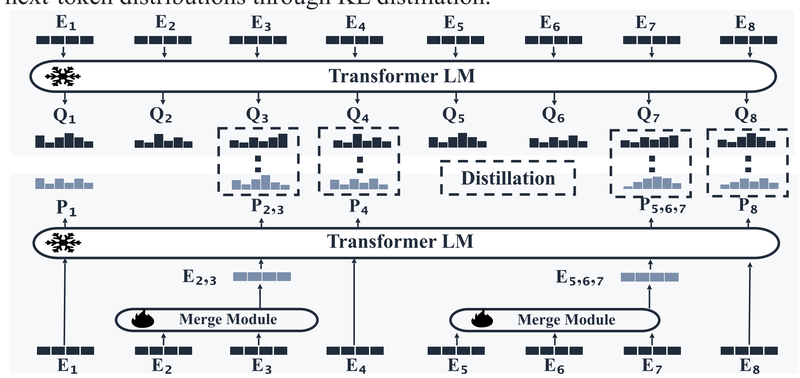
UMIM (Universal Multi-step Input Merging) uses a lightweight merge module to replace multi-token spans with reusable surrogate embeddings.



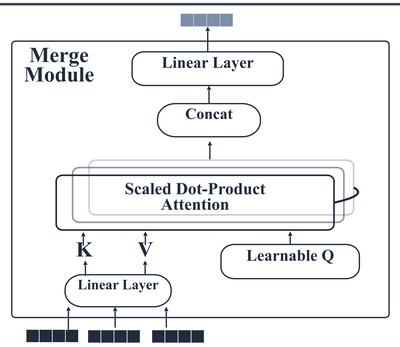
1 Training and Merge Module

We extract frequent 2–4-grams from WikiText-103 as merge candidates, providing repeated examples for training the merge module.

Train only the merge module to match the frozen LLM's next-token distributions through KL distillation.



Original embeddings / teacher Merged embeddings / student



Shared across token spans

A linear layer projects span embeddings into keys and values.

Learnable queries pool information through multiple attention heads.

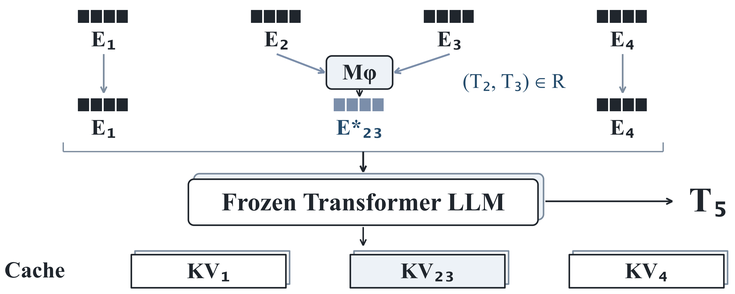
Concatenate the head outputs, then project them into one surrogate embedding.

2 Inference

At Inference time, we merge tokens from the embedding level. We can merge tokens at both Prefill and Decoding Stage.

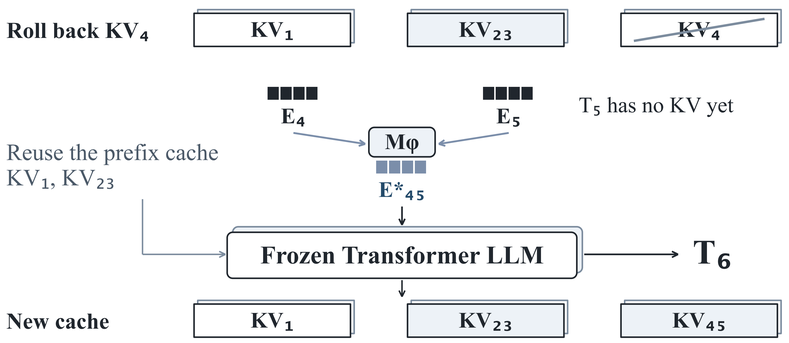
A Prefill: Input tokens merging

E_i: token embeddings. R: the mined set of merge rules. T: tokens ids



B Decode: match, KV cache roll back, replace

The sampled T_5 completes a matching suffix (T_4, T_5).



The output text retains the original tokens T_5 and T_6.

3 Downstream Deployment

Deployment mode	Merge rules	Merge module
Direct reuse	WikiText-103 rules	Frozen base module
Rules only	Re-mine task rules	Frozen base module
Task adaptation	**Re-mine task rules**	**Base → SFT → DPO**

The backbone stays frozen. We only train the merge module for the whole project. UMIM can also be considered a method for parameter fine-tuning.

Takeaway: Learn a reusable surrogate for token spans. Shorten prompt and decoding contexts with a frozen LLM.

4 Key Results

On WikiText-103, Llama-3-8B uses 36.1% fewer tokens with UMIM, while perplexity (PPL) changes from 13.4 to 13.8.

Further tests evaluate QA, summarization, and task-specific adaptation.

Language modeling

PPL: Base → UMIM (lower is better). TR: token reduction.

Backbone	WikiText-103	BookCorpus	OpenWebText
Llama-3-8B	13.4 → 13.8 TR 36.1%	15.3 → 15.3 TR 14.8%	9.2 → 9.8 TR 13.8%
Llama-3.2-1B	20.1 → 23.4 TR 36.1%	21.2 → 22.0 TR 14.8%	13.4 → 14.7 TR 13.8%
GPT2-XL	28.6 → 37.9 TR 35.1%	27.1 → 30.0 TR 14.1%	13.1 → 14.4 TR 12.5%

Zero-shot QA

Llama-3.1-8B

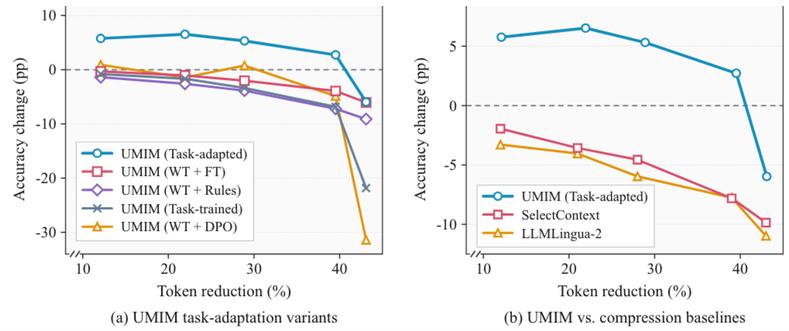
Acc. (%)	PIQA	COPA	OpenBookQA	ARC-E	ARC-C
Base	79.6	76.8	43.6	76.5	49.5
UMIM	**78.4**	**76.8**	**42.4**	**73.7**	**44.8**
TR (%)	10	6	10	14	14

CNN/DailyMail summarization

Method	ROUGE-L ↑	BERTScore ↑	TR (%)
Base	24.3	86.8	0
UMIM	**23.1**	**86.4**	**16**

Task adaptation

HellaSwag · Llama-3.1-8B



(a) UMIM task-adaptation variants (b) UMIM vs. compression baselines

Generation example

Llama-3.1-8B

Blue highlights indicate merged spans.

Prompt	Traditional Chinese literary criticism emphasized the life of the author when interpreting a work.
Base	The author's life was seen as the key to understanding the work.
UMIM	The author's life was seen as the key to understanding the work.