ACL 2025 VIENNA

RoToR: Towards More Reliable Responses for Order-Invariant Inputs

Paper Code

Soyoung Yoon[1*], Dongha Ahn[12], Youngwon Lee[1], Minkyu Jung[2], HyungJoo Jang[2], Seung-won Hwang[1†]

[1]Seoul National University [2]Channel Corporation

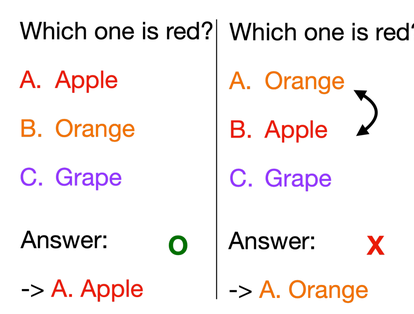
[†]Corresponding Author *Work done during an internship at Channel Corporation

Overview

We introduce **RoToR**: a **zero-shot** order-invariant Language Model, with (1) Global Sorting + Circular Position IDs and (2) Selective Routing for Mixed Inputs, which achieve SOTA robustness on 3 benchmarks & 25-45% lower FLOPs v.s. Baselines (PINE)

Motivation: Positional Bias for Listwise Inputs

- Lost-in-the Middle (RAG)
- First-choice bias (75%) in LLM-as-a-judge
- MMLU rank shifts by 8 with shuffle -> need neutral handling for sets, tables, multiple-choice questions
- Zero-shot invariant LMs have been proposed as a solution, but had 2 limitations:



Limitations of previous Zero-shot order-invariant LMs

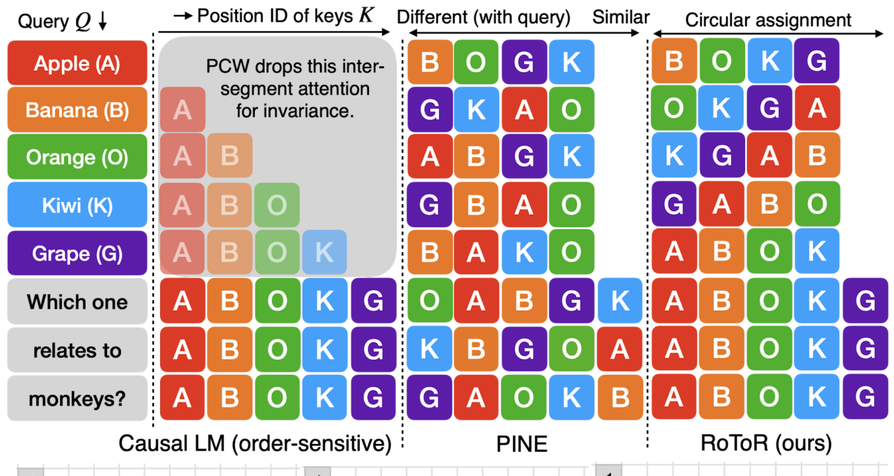
- **Limitation 1: Training and inference distribution mismatch**
  - PCW, Set-based prompting: No cross-segment content
  - PINE: per-query sort -> $O(O(n^2)$ + instability
  - Frequent ID changes cause **OOD behavior** -> drops its ability
- **Limitation 2: Fail to extend to real-life scenarios** (order-invariant + order-sensitive)
  - Does not consider, cannot be applied to hybrid cases (e.g., MMLU)

Self-attention patterns (x = query, y = key) across order-invariant models →

Example of order-sensitive and order-invariant cases

In 8085 name/names of the 16 bit registers is/are:

Order sensitive	Order invariant
A. stack pointer	A. stack pointer
B. program counter	B. program counter
C. both A and B.	C. accumulator
D. none of these	D. microprocessor



PINE: Bidirectional processing with Q-K similarity

- Has to obtain the same attention representation, regardless of initial ordering of segments
- Places query IDs last, sorts other segments in a order-invariant way

Challenges of PINE

- Frequent alterations on position IDs (layer, head, suffix, generation tokens)
- Computationally expensive
- Numerical Instability (arising from attention assignment)

Proposed Method: RoToR

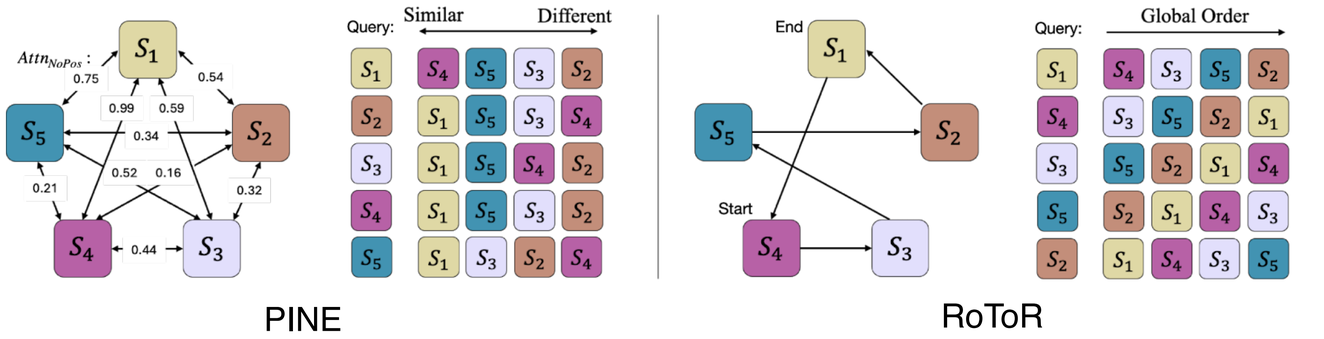
- Keep the bidirectional structure, but alter the position assignment in a simple and stable way!
- Define a single global ordering + circular assignment
- We propose three different global ordering methodology: (1) Lexical: Hierarchical sort on tokenized ID numbers, (2) Reranking (monot5), (3) Frequency-normalized

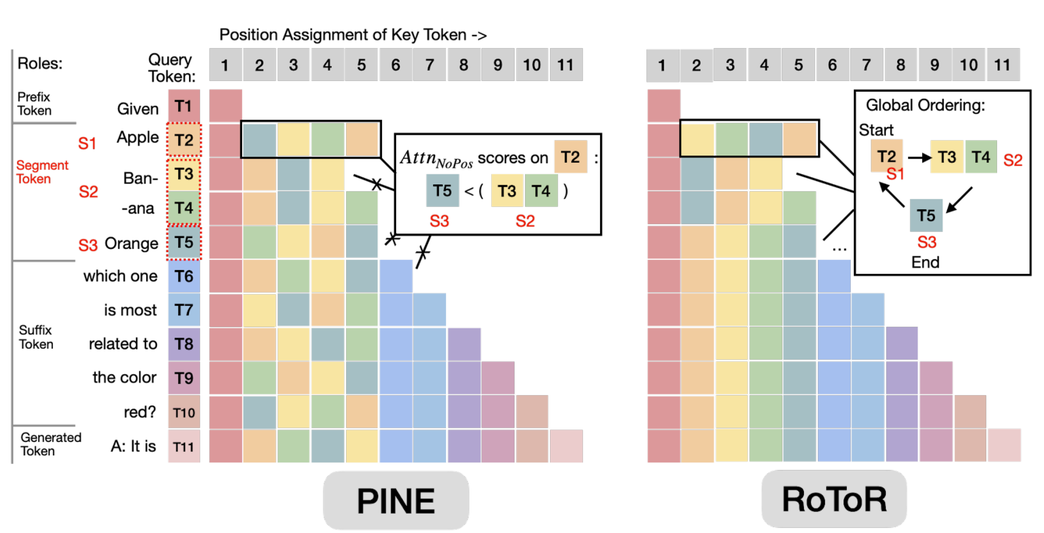
1. Training and inference distribution mismatch

- Stable, order-invariant solution (RoToR)
- Query-agnostic global ordering with minimal positional ID modifications

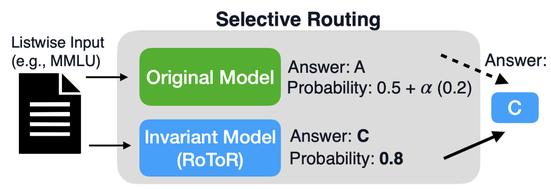
2. Fail to extend to hybrid cases

- Selective Routing, which switches between original / invariant LMs based on confidence





PINE: query-dependent grid, **RoToR:** fixed order, rotate per query
-> Stable IDs, zero collisions, less computation



Experimental Setup

- **Benchmarks:**
  - Lost-in-the-Middle (LitM)
  - Knowledge Graph QA (KGQA): Mintaka
  - MMLU: selective routing cases
  - LongBench: long context scenarios (Appendix)
- **Model backbones:**
  - Llama-3.1-8B/70B
  - Qwen-1.5-4/7/72B-Chat
- **Metrics:** best_subspan_em (LitM), EM, F1, Acc. (KGQA), Acc. (MMLU)
- **Methods:** Original (order-sensitive), PCW, Set-based prompting, PINE, RoToR

KGQA

Method	Llama-3.1-8B-Instr.			Llama-3.1-70B-Instr.			Qwen1.5-4B-Chat			Qwen1.5-7B-Chat			Qwen1.5-72B-Chat		
	Acc.	EM	F1	Acc.	EM	F1	Acc.	EM	F1	Acc.	EM	F1	Acc.	EM	F1
$N=30$															
Initial, no shuffling of segments															
Original	50.2	44.0	51.9	61.6	57.7	63.6	30.7	27.9	34.9	31.5	27.8	35.4	41.4	37.7	43.7
PINE	51.5	45.0	52.6	63.1	58.7	64.8	31.6	28.7	35.6	32.3	28.8	36.4	46.7	42.9	49.0
RoToR	**53.1**	**46.5**	**54.1**	**63.6**	**59.1**	**65.2**	32.0	29.0	35.7	**34.3**	**29.8**	**37.7**	**47.5**	**43.2**	**49.2**
RoToR-MonoT5	51.6	45.0	52.5		–		**32.3**	29.1	**36.2**	32.9	28.4	36.3		–	
RoToR-Freq.	52.6	46.1	53.7		–		**32.3**	**29.2**	36.0	33.7	29.5	37.2		–	
After shuffling segments, averaged over 3 seeds															
Original	49.5	43.3	51.0	62.1	57.8	64.0	30.1	27.5	34.7	31.4	27.3	35.0	41.0	37.6	43.6
↪ stdev. (±)	0.07 / 0.14 / 0.17			0.37 / 0.40 / 0.27			0.41 / 0.34 / 0.43			0.26 / 0.28 / 0.29			0.75 / 0.40 / 0.33		
PINE	51.8	45.2	52.8	63.3	58.8	64.9	31.5	28.7	35.6	32.3	28.8	35.7	46.9	**43.3**	**49.2**
↪ stdev. (±)	0.05 / 0.07 / 0.16			0.13 / 0.04 / 0.10			0.20 / 0.18 / 0.13			0.17 / 0.20 / 0.13			0.18 / 0.20 / 0.20		
RoToR	**52.8**	**46.2**	**53.8**	**63.5**	**59.1**	**65.3**	31.8	28.8	35.5	**34.2**	**29.9**	**37.7**	**47.4**	43.1	49.1
↪ stdev. (±)	0.05 / 0.05 / 0.02			0.11 / 0.07 / 0.08			0.05 / 0.02 / 0.09			0.09 / 0.07 / 0.06			0.06 / 0.04 / 0.07		
RoToR-MonoT5	51.6	45.0	52.6		–		**32.4**	29.2	**36.3**	33.0	28.8	36.5		–	
↪ stdev. (±)	0.12 / 0.06 / 0.10				–		0.04 / 0.02 / 0.13			0.12 / 0.09 / 0.07				–	
RoToR-Freq.	52.5	45.9	53.5		–		32.3	**29.3**	36.0	33.8	29.6	37.4		–	
↪ stdev. (±)	0.10 / 0.15 / 0.11				–		0.13 / 0.16 / 0.09			0.04 / 0.00 / 0.09				–	
$N=50$															
Initial, no shuffling of segments															
Original	50.0	44.0	51.7	62.6	58.5	64.5	31.6	28.6	35.8	31.7	28.0	35.7	42.1	38.7	44.5
PINE	51.6	45.1	52.6	64.1	59.8	65.8	31.6	28.8	35.3	32.0	28.5	35.9	48.0	44.1	49.9
RoToR	52.9	46.0	53.6	**64.6**	**60.0**	**66.2**	**32.7**	**29.6**	**36.2**	**34.3**	**30.1**	**38.0**	**48.4**	**44.3**	**50.3**
RoToR-MonoT5	52.4	45.4	52.8		–		32.3	29.3	35.9	32.9	28.9	36.6		–	
RoToR-Freq.	**53.1**	**46.4**	**53.7**		–		32.3	29.2	36.1	33.5	29.5	37.2		–	
After shuffling segments, averaged over 3 seeds															
Original	49.7	43.5	51.0	62.8	58.5	64.5	30.3	27.6	35.0	31.6	27.9	35.5	42.1	38.9	44.7
↪ stdev. (±)	0.34 / 0.28 / 0.46			0.29 / 0.28 / 0.05			0.26 / 0.24 / 0.35			0.40 / 0.56 / 0.42			0.30 / 0.40 / 0.35		
PINE	51.8	45.3	52.7	64.3	59.8	65.9	31.5	28.7	35.3	31.7	28.2	35.7	**48.0**	**44.3**	50.0
↪ stdev. (±)	0.15 / 0.16 / 0.19			0.16 / 0.15 / 0.14			0.17 / 0.20 / 0.21			0.18 / 0.16 / 0.14			0.02 / 0.04 / 0.05		
RoToR	52.7	45.9	53.5	**64.5**	**60.0**	**66.1**	**32.5**	**29.6**	**36.1**	**34.2**	**30.1**	**38.0**	**48.3**	**44.3**	**50.3**
↪ stdev. (±)	0.05 / 0.09 / 0.04			0.02 / 0.02 / 0.01			0.11 / 0.06 / 0.09			0.06 / 0.05 / 0.04			0.05 / 0.09 / 0.05		
RoToR-MonoT5	52.2	45.2	52.8		–		32.3	29.4	35.9	32.8	28.8	36.5		–	
↪ stdev. (±)	0.16 / 0.18 / 0.18				–		0.16 / 0.13 / 0.07			0.16 / 0.09 / 0.07				–	
RoToR-Freq.	**53.1**	**46.4**	**53.7**		–		32.4	29.3	**36.1**	33.7	29.6	37.4		–	
↪ stdev. (±)	0.02 / 0.07 / 0.03				–		0.09 / 0.04 / 0.06			0.04 / 0.16 / 0.22				–	

- Top-30 and Top-50 knowledge triples per query
- Test before / after shuffling segments to see robustness
- RoToR obtains lower stdev (better stability) + higher performance than PINE
- Trend persists for > 70B model variants
  - Single use of Order-invariant models fail, but selective routing restores accuracy & improves performance. →
- On Llama-3.1-8B-Inst.
- Original model fluctuates performance, while ours (RoToR) maintains stable

Efficiency

- Less computation:
  - Overhead FLOPs ↓ 98 % (72B)
- Faster:
  - E2E Latency ↓ 23-43 % on LitM
- Reduces OOD:
  - Perplexity ↓; collision rate 0 %

Model	Benchmark	PINE	RoToR	Reduction
(a) Overhead FLOPs, relative to original model				
Llama-3.1-8B-Instruct	MMLU, $N=4$	0.59×	**0.55**×	7.6%
	LitM, $N=10$	7.07×	**4.81**×	31.9%
	LitM, $N=30$	22.43×	**15.05**×	32.9%
Llama-3.1-70B-Instruct	KGQA, $N=30$	1.27×	**0.94**×	26.0%
	KGQA, $N=50$	1.82×	**1.29**×	29.0%
Qwen1.5-72B-Chat	KGQA, $N=30$	0.45×	**0.01**×	98.0%
	KGQA, $N=50$	0.58×	**0.03**×	94.8%
(b) End-to-end latency (s)				
Llama-3.1-70B-Instruct	LitM, $N=10$	57,352	**44,219**	22.9%
	LitM, $N=20$	87,091	**58,680**	32.6%
Llama-3.1-8B-Instruct	MMLU, $N=4$	7,371	**6,608**	10.4%
	LitM, $N=10$	18,551	**14,264**	23.1%
	LitM, $N=30$	41,664	**23,569**	43.4%
(c) Perplexity & Collision rate, (on LitM)				
Llama-3.1-8B-Instruct	Perplexity ($N=20$)	6.91	**6.65**	–
	Collision rate ($N=30$)	42.3%	**0 (None)**	–

MMLU

Method	Llama-3.1-8B-Instruct			Qwen1.5-4B-Chat			Qwen1.5-7B-Chat		
	Init.	Rev.	Avg.	Init.	Rev.	Avg.	Init.	Rev.	Avg.
Orig.	68.3	64.8	65.5 ± 1.0	53.6	51.9	52.6 ± 0.6	**60.1**	56.6	58.6 ± 0.9
PCW	57.0	55.1	56.1 ± 1.1		–			–	
Set-Based Prompting	31.1	33.0	31.6 ± 0.8		–			–	
PINE	64.8	63.3	63.6 ± 0.7	50.5	49.3	49.4 ± 0.5	57.0	54.4	55.8 ± 0.9
RoToR	63.2	62.6	62.8 ± 0.7	49.6	47.7	48.3 ± 0.7	56.5	55.8	56.2 ± 0.6
↪ + S.R.	**68.5**	65.1	65.7 ± 0.9	53.7	51.8	**52.6 ± 0.6**	**60.1**	**57.4**	**58.8 ± 0.7**
RoToR - MonoT5	64.2	62.9	63.5 ± 0.5	49.7	47.6	48.7 ± 0.7	56.2	54.4	55.5 ± 0.7
↪ + S.R.	68.4	65.2	65.8 ± 0.9	**53.8**	51.9	**52.6 ± 0.6**	**60.1**	57.3	58.7 ± 0.8
RoToR - Freq.	64.3	63.6	63.8 ± 0.6	49.9	47.6	48.7 ± 0.5	56.4	54.7	55.7 ± 0.7
↪ + S.R.	**68.5**	**65.3**	**65.8 ± 0.8**	53.7	**52.3**	**52.6 ± 0.6**	60.0	57.3	58.6 ± 0.8
RoToR + S.R. (Oracle)	75.0	71.9	72.7 ± 1.0	61.8	60.1	61.1 ± 1.0	68.1	66.2	67.2 ± 0.7

LitM