You Only Cache Once: Decoder-Decoder Architectures for Language Models

Yutao Sun
Tsinghua University
syt23@mails.tsinghua.edu.cn

Intelligence as Infrastructure

清華大學 Tsinghua University

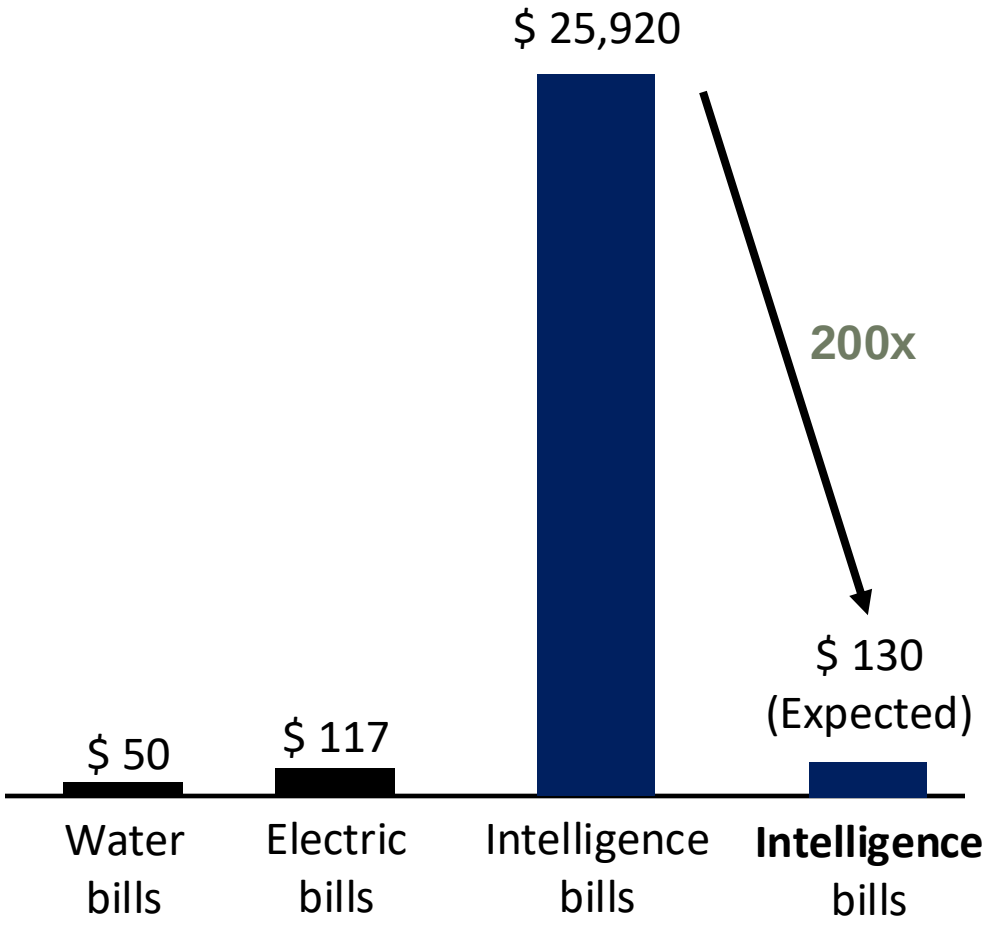
Rough estimation based on current GPT-4 services

Average cost per GPT-4 call ($ 0.06) x

Estimated average calls per house-hold (24h x 60m x 10 = 14,400) x 30d = $ 25,920

Why are Transformer LLMs High-Cost?

- Speed/Throughput/Latency: Memory access is much slower than computation
- Energy: Memory access needs much more energy than computation
- Number of GPUs to host a model: Memory capacity of one GPU is insufficient to host a model



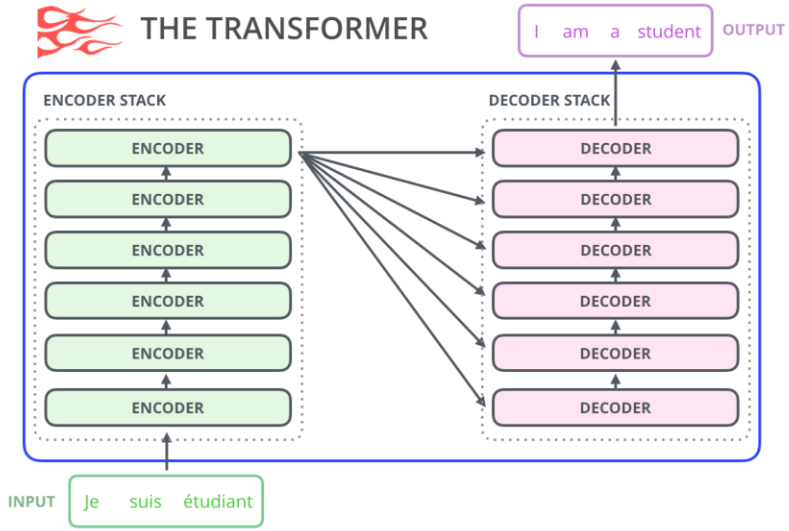
* Average monthly bills per house-hold in the US

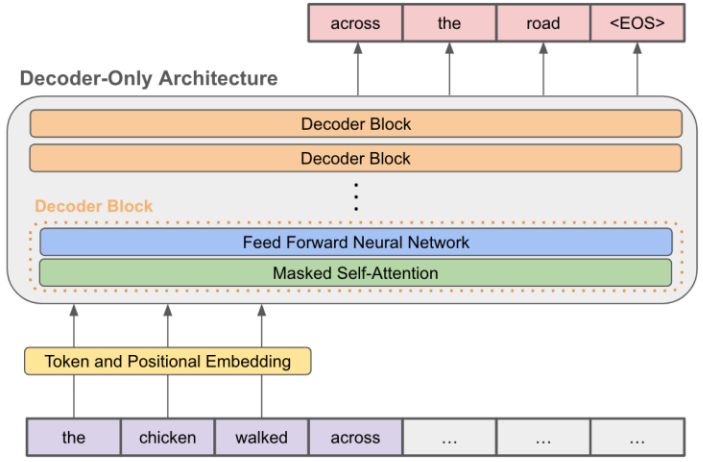
YOCO
Motivation

- Hybridization does not sound a great solution
  - 4x maximal acceleration and KV cache saving
  - Still quadratic complexity
- Towards the optimal acceleration for lossless sequence modeling
  - $O(N)$ KV cache is compulsory, maybe just one piece?
  - $O(N)$ single-step inference is essential for token retrieval
- **Cache once** with linear-complexity **pre-filling**!

YOCO
Model Layout

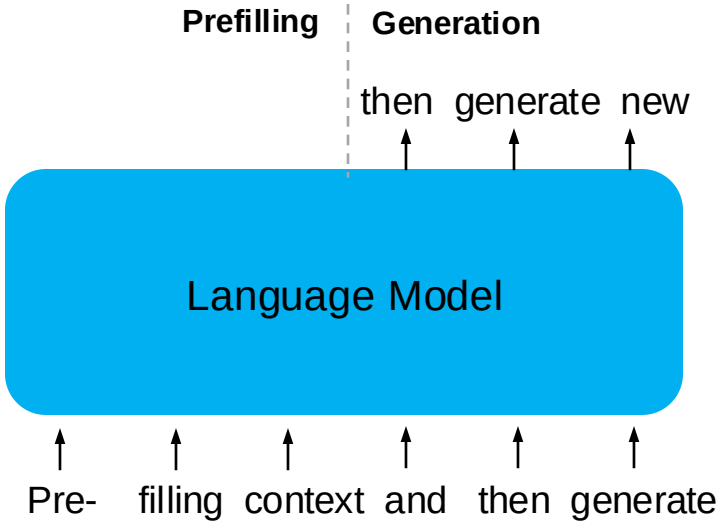
- Encoder-Decoder:
  - Bidirectional modeling in the encoder part
  - Save layer-wise KV cache
  - Struggle to implement efficient pre-training
- Decoder-Only:
  - Default architecture in modern LLMs
  - Heavy KV cache and prefilling cost
- Decoder-Decoder:
  - Efficient pre-filling and KV cache from Eec-Dec
  - Next Token Prediction from Dec-Only





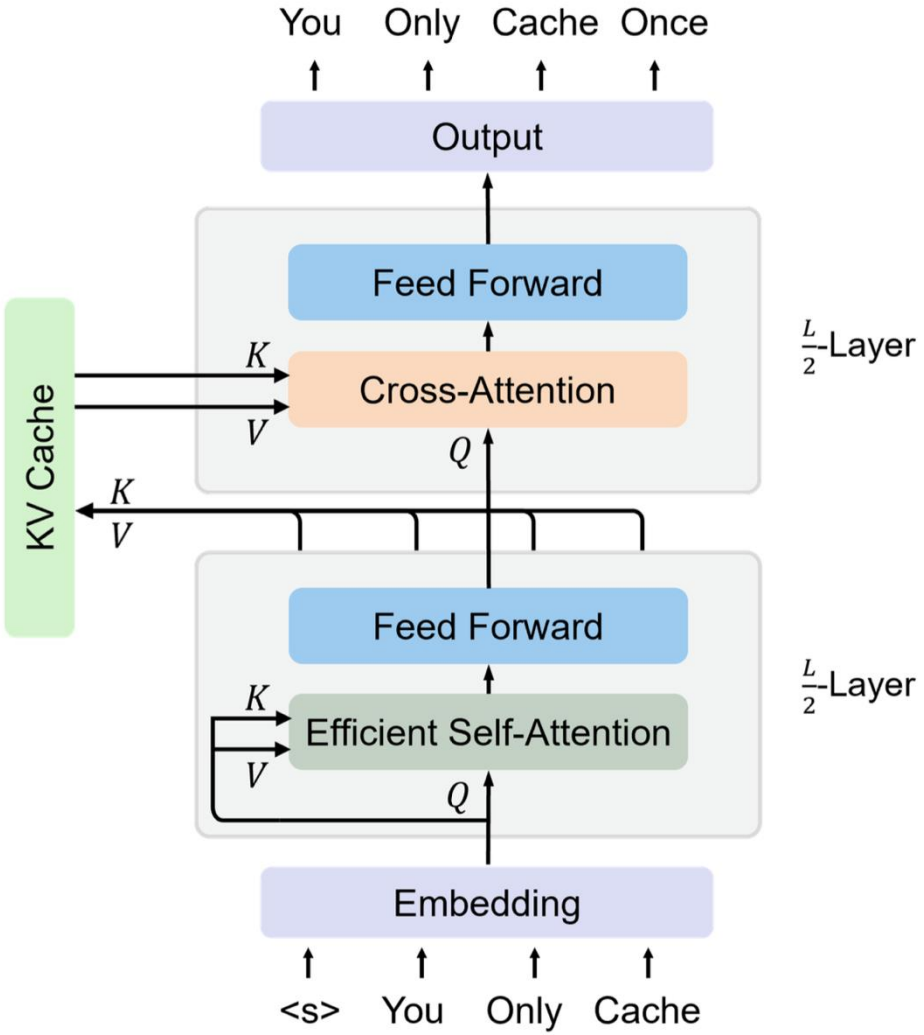
YOCO
Generation Pipeline

- Prefilling:
  - Encode all the user query into KV cache for generation
  - $O(N^2)$ complexity where N is sequence length
- Generation:
  - Decode the next token each step with the previous $O(LN)$ KV cache
  - Memory bounded



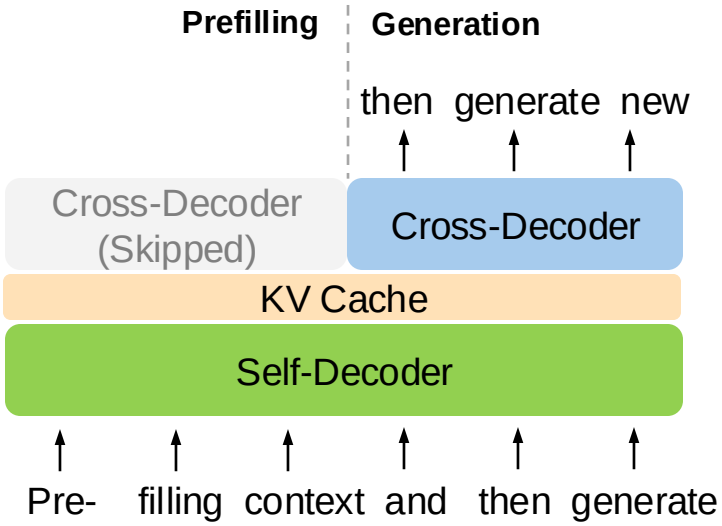
YOCO
Architecture

- Disentangle prefilling and generation stage
- (Self-)Decoder-(Cross)-Decoder architecture
- RetNet and other linear architectures are still valuable!
- You Only Cache Once (YOCO) global KV cache
- Shared keys and values with Cross-Attention
- Stacked connection rather than Encoder-Decoder style



YOCO
Inference Advantage

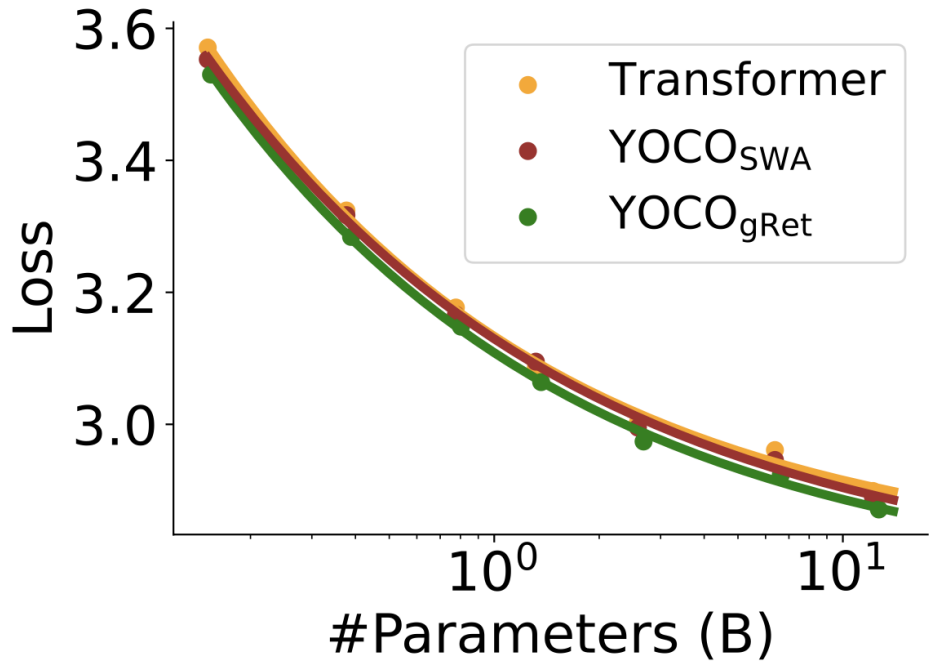
- Prefilling:
  - Transformer requires $O(LN^2D)$ computation to encode KV cache with Self-Attention
  - YOCO only needs $O(LND)$ computation due to efficient Self-Decoder
- Context Memory:
  - Transformer saves KV cache layer-wisely with $O(LND)$ GPU memory
  - YOCO only saves KV cache once where the memory usage is only $O\big((L+N)D\big)$



YOCO

Language Modeling Performance

- Better than standard Transformer
- Gains come from hybrid architectures of attention and retention
- Verified with strong open-source Transformer models including StableLM

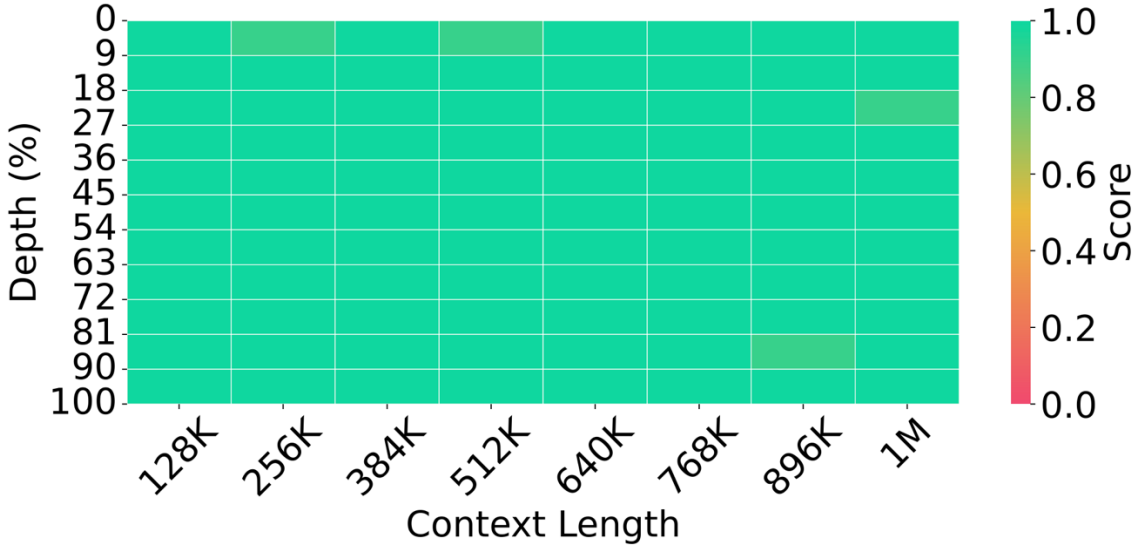


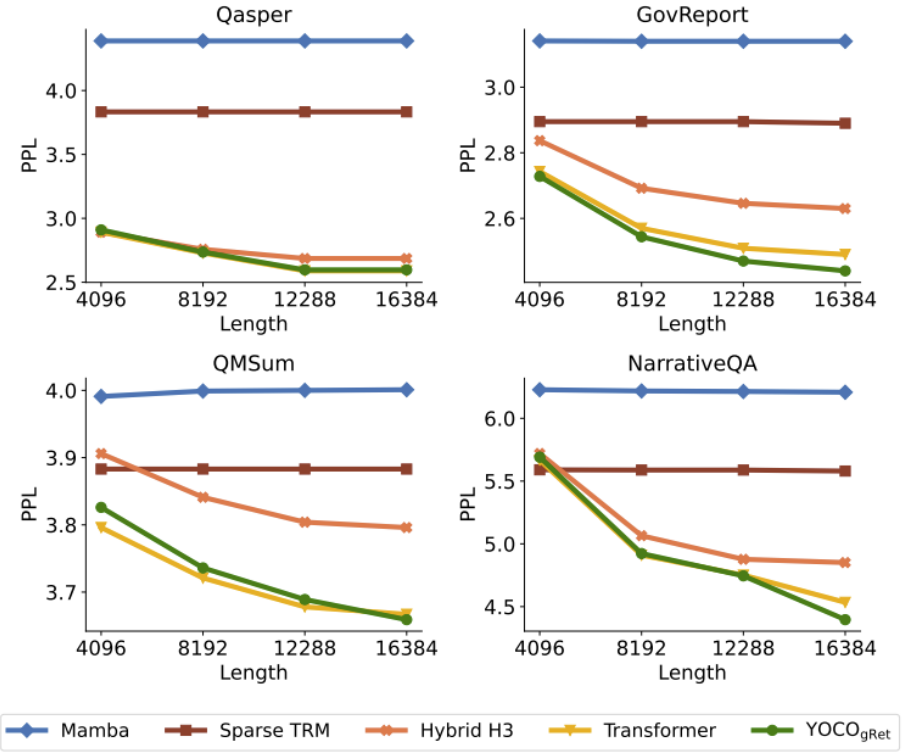
Model	ARC-C	ARC-E	BoolQ	Hellaswag	OBQA	PIQA	Winogrande	SciQ	Avg
Training with 1T tokens									
OpenLLaMA-3B-v2	0.339	0.676	**0.657**	**0.700**	0.260	0.767	0.629	**0.924**	0.619
StableLM-base-alpha-3B-v2	0.324	0.673	0.646	0.686	0.264	0.760	0.621	0.921	0.612
StableLM-3B-4E1T	—	0.666	—	—	—	**0.768**	0.632	0.914	—
YOCO-3B	**0.379**	**0.731**	0.645	0.689	**0.298**	0.763	**0.639**	**0.924**	**0.634**
Training with 1.6T tokens									
StableLM-3B-4E1T	—	0.688	—	—	—	0.762	0.627	0.913	—
YOCO-3B	0.396	0.733	0.644	0.698	0.300	0.764	0.631	0.921	0.636
Extending context length to 1M tokens									
YOCO-3B-1M	0.413	0.747	0.638	0.705	0.300	0.773	0.651	0.932	0.645

YOCO

Long Sequence Modeling

- Continue training to 1 million length
- Achieving perfect accuracy on Needle-in-Haystack experiments
- Comparable with well-known Transformer models including MiniCPM and ChatGLM

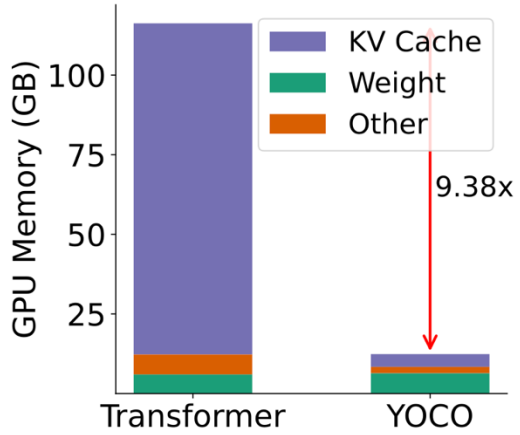


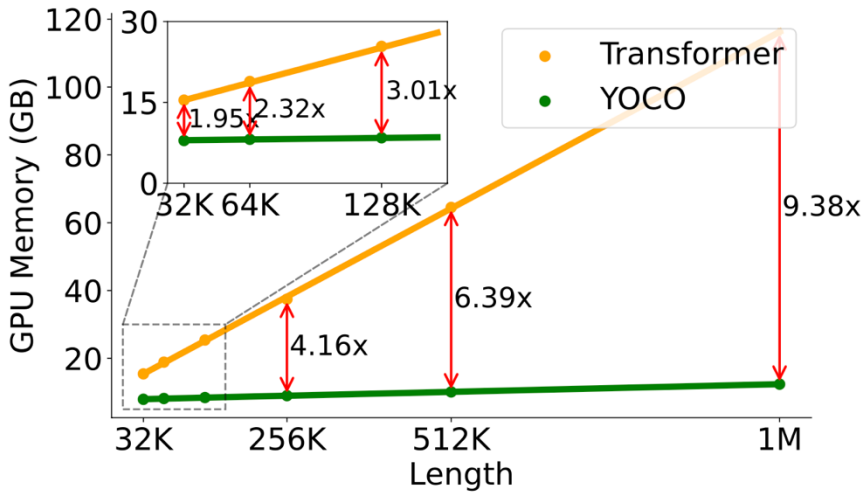


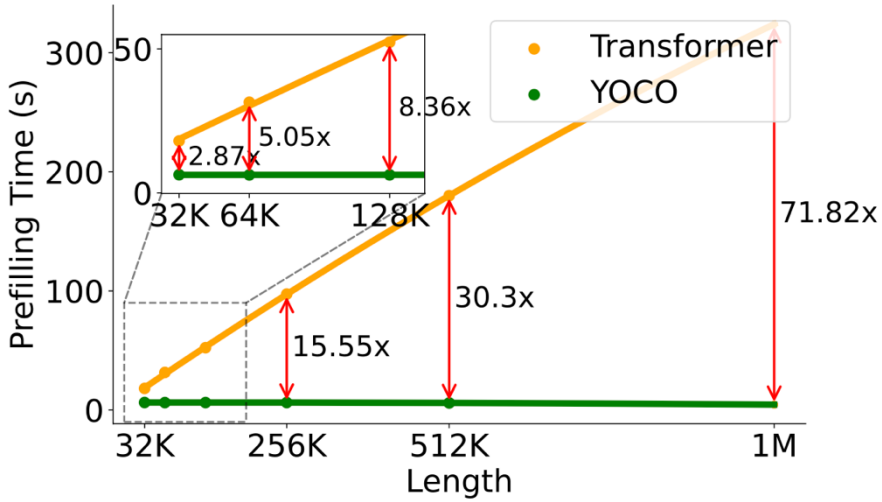
Model	Size	$N = 2$	$N = 4$	$N = 8$
YaRN-Mistral-128K [27]	7B	0.12	0.08	0.20
LWM-1M-text [23]	7B	0.90	0.76	0.62
MiniCPM-128K [15]	2.4B	1.00	0.54	0.56
ChatGLM3-128K [45]	6B	0.72	0.52	0.44
YOCO-3B-1M	3B	0.98	0.84	0.56

YOCO
Inference Performance

- 9.4x memory saving at 512k length
- Prefilling latency: 180s -> 6s
- KV cache is almost negligible
- Make long sequence deployment practical!







Conclusion

- Why YOCO will be the default backbone in the future?
  - Comparable and better performance in almost every aspects
  - Huge efficiency advantages
  - Long sequence demand grows
- Code is available at https://aka.ms/YOCO

Future

- Multi-Modality Fusion
  - Video is a natural scenario for long sequence modeling
  - Real-time video understanding, low-cost generation, embodied agent…
- Sparse Attention Diagram
  - Building index for key-value retrieval
  - YOCO enables only one index rather than index for each layer