Microsoft
Stanford University

Decoder-Hybrid-Decoder Architecture for Efficient Reasoning with Long Generation

Liliang Ren[1], Congcong Chen[1], Haoran Xu[1], Young Jin Kim[1], Adam Atkinson[1], Zheng Zhan[1], Jiankai Sun[2], Baolin Peng[1], Liyuan Liu[1], Shuohang Wang[1], Hao Cheng[1], Jianfeng Gao[1], Weizhu Chen[1], Yelong Shen[1]
[1] Microsoft [2] Stanford University

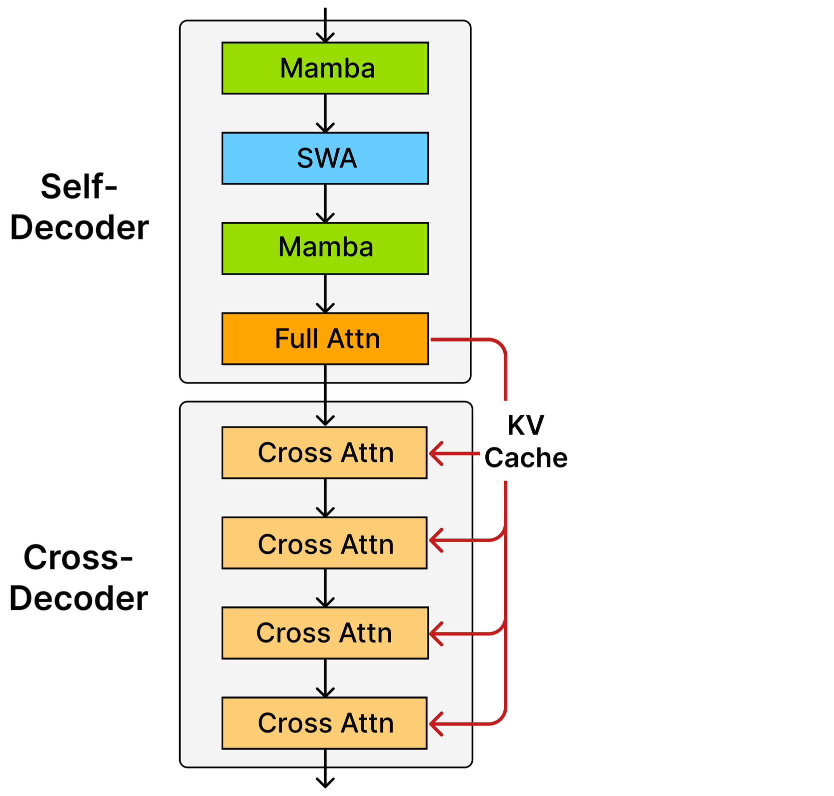
ChatPhi4Flash Github

Q1. How to Build a Decoding-Efficient Model with **Linear Prefilling** Time?
Q2. Does Hybrid Models Scale Effectively with Compute and Data?
Q3. Can Hybrid Models Reason Efficiently with Long Generation?

A1: Gated Memory Unit + YOCO
A2: Higher data-efficiency, same asymptotic limit
A3: Yes! **10x throughput** for 32K generation

Motivation

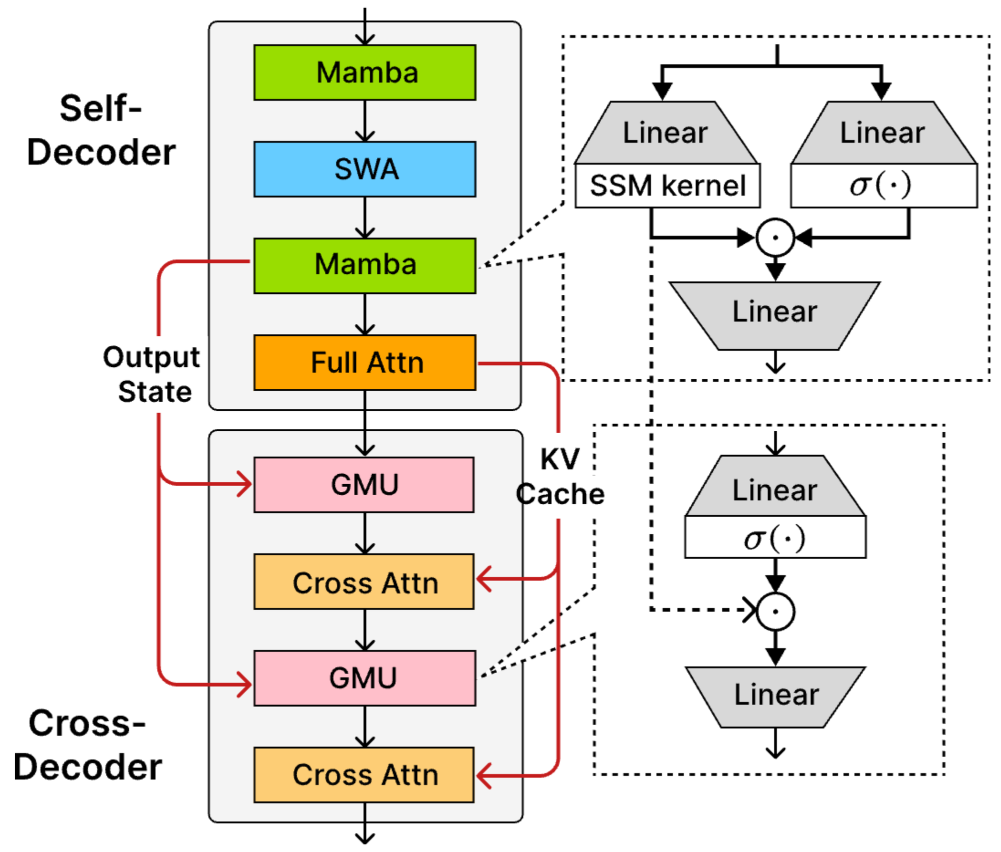
- Decoder-decoder architecture, YOCO (You Only Cache Once) [SDZ+24], has **linear** prefill complexity.



- Memory I/O cost = Full attention for cross-decoder.
  Slow for long sequence generation!
- Idea: Replace Cross Attention with linear modules.

Decoder-Hybrid-Decoder Architecture

- **Linear** prefill complexity with half cross-attention layers replaced with Gated Memory Units (GMUs).



SambaY = GMU+YOCO+Samba

Gated Memory Unit (GMU)

- Token mixing as a matrix operator at layer l':

$$M^{(l')} = A^{(l')}\, V^{(l')}$$

- GMU at layer $l > l'$:

$$Y_l = (M^{(l')} \odot \sigma(X_l W_1^T))W_2$$

- Fine-grained reweighting of the previous layer's token mixing matrix with current layer input:

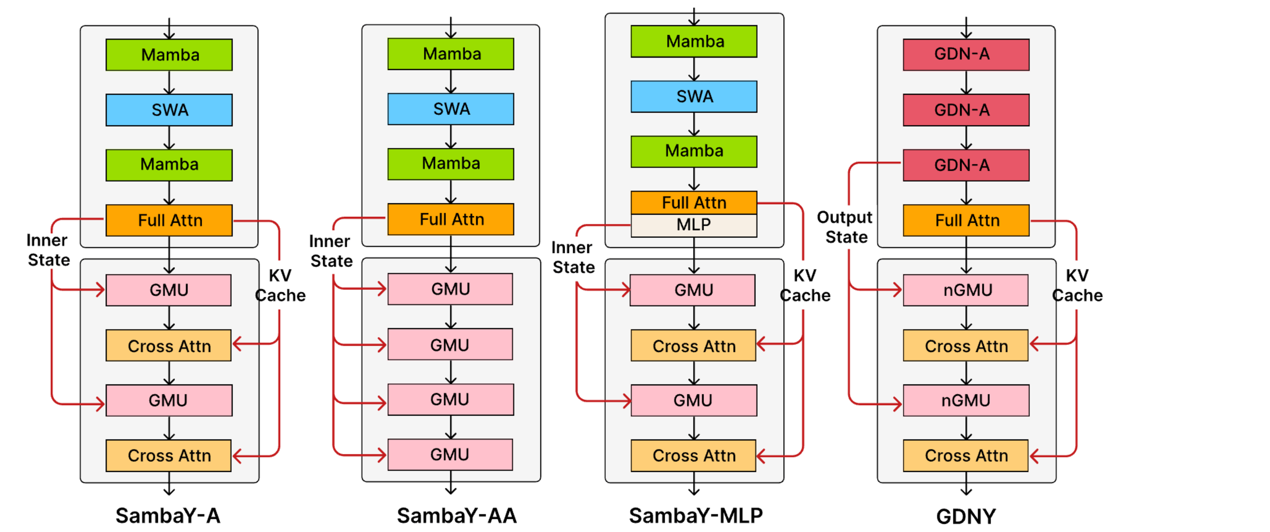
$$H_{ik} = G_{ik}^{(l)} \sum_j A_{ij}^{(l')}\, V_{jk}^{(l')} = \sum_j G_{ik}^{(l)}\, A_{ij}^{(l')}\, V_{jk}^{(l')} = \sum_j \underbrace{A_{ij}^{(l')}\, G_{ik}^{(l)}}_{\tilde{A}_{ijk}}\, V_{jk}^{(l')}$$

where $G^{(l)} = \sigma(X_l W_1^T)$.

- GMU keeps linearity on the previous value $V^{(l')}$ so the original signal is not distorted.
- We can add RMSNorm after gating => **nGMU** to stabilize training of memory from linear attention.
- GMU can also be applied to Attention and MLP!

Ablation Study

- We pretrain 1B models on ProLong-64K [GWYC24] dataset with 32K context



- GDNY > SambaY > SambaY-MLP > SambaY-A > SambaY-AA on PhoneBook [JBKM24] with multi-key-value retrieval from 32K context.

Model	Speed mtps ↑	Wiki. ppl ↓	PB-32K acc ↑	LMB. acc ↑	ARC-c acc_n ↑	ARC-e acc ↑	Hella. acc_n ↑	PIQA acc ↑	Wino. acc ↑	Avg. acc ↑
SambaY	1.10	16.89	78.13	50.22	28.58	59.18	**49.07**	70.84	**55.09**	52.16
MambaY	0.94	17.29	12.50	50.24	28.84	59.64	48.27	71.44	52.80	51.87
SambaY-2	**1.43**	17.17	40.63	48.96	28.84	59.18	48.01	70.18	50.83	51.00
MambaY-2	1.38	18.63	50.78	49.58	28.24	58.75	48.29	70.13	51.07	51.01
S-GDNY	1.34	**16.78**	83.59	**50.94**	29.61	58.96	48.93	71.55	51.85	51.97
GDNY	1.22	16.92	**89.84**	50.38	28.84	60.61	48.01	71.27	51.38	51.75
SambaY-A	1.11	18.12	58.59	49.85	30.29	59.60	48.41	71.33	54.06	52.26
SambaY-AA	1.25	17.03	46.88	49.93	28.50	59.05	48.69	**72.25**	53.91	52.06
SambaY-MLP	1.15	18.70	64.84	50.16	**30.38**	**60.69**	48.46	71.44	54.78	**52.65**

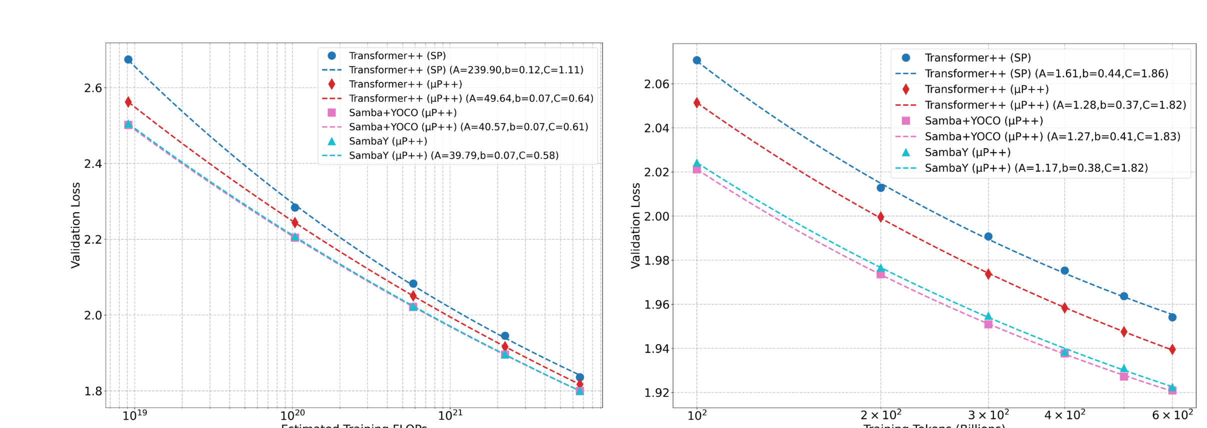
Scaling Experiments

How to have fair scaling comparisons between architectures?

- Tie number of parameters by adjusting model width
- Stable and effective scaling with μP++
  = μP + Depth-μP + zero weight decay on vector-like parameters.

Parameter	Scheme	LR mult.	Initialization	Res. mult.	Weight mult.	WD
Embedding	SP	$\propto 1$	$\mathcal{N}(0,\sigma^2)$	—	$\propto 1$	$\propto 1$
	μP	$\propto 1$	$\mathcal{N}(0,\sigma^2)$	—	$\propto 1$	$\propto 1$
	μP++	$\propto 1$	$\mathcal{N}(0,\sigma^2)$	—	$\propto 1$	0
Unembedding	SP	$\propto 1$	0 or tied	—	$\propto 1$	$\propto 1$
	μP	$\propto 1$	0 or tied	—	$\propto 1/w$	$\propto 1$
	μP++	$\propto 1$	0 or tied	—	$\propto 1/w$	0
Hidden Weights	SP	$\propto 1$	$\mathcal{N}(0,\tau^2)$	1	$\propto 1$	$\propto 1$
	μP	$\propto 1/w$	$\mathcal{U}(\frac{-\beta}{\sqrt{\text{fan\_in}}}, \frac{\beta}{\sqrt{\text{fan\_in}}})$	1	$\propto 1$	$\propto 1$
	μP++	$\propto 1/w$	$\mathcal{U}(\frac{-\beta}{\sqrt{\text{fan\_in}}}, \frac{\beta}{\sqrt{\text{fan\_in}}})$	$1/\sqrt{2d}$	$\propto 1$	$\propto 1$

- Power Law: $L(D_{\text{FLOPs}}) = A \cdot D_{\text{FLOPs}}^{-b} + C$

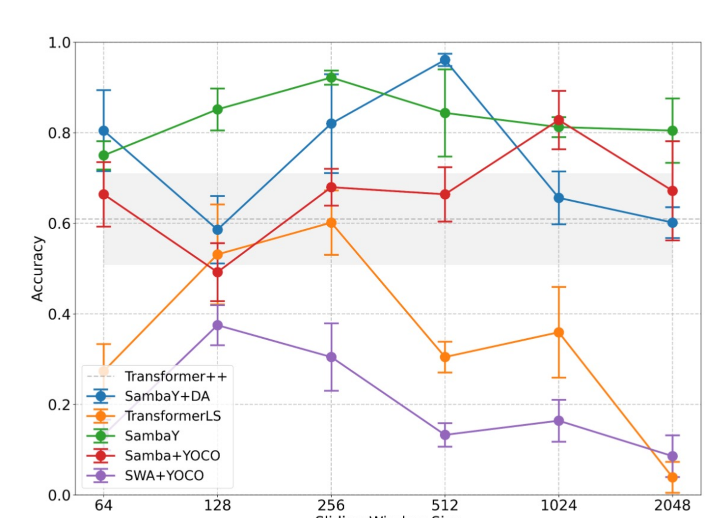


(a) Compute scaling comparisons (b) Data scaling comparisons

- SambaY has **same learning efficiency (b) & better scaling convergence (C)** than Transformer for compute scaling (with 5x Chinchilla Optimal).
- Why? **Higher data efficiency and same data scaling convergence.**

Long-context Performance

- We pretrain 1B models with various SWA size on ProLong-64K with 32K context and evaluate them on PhoneBook-32K.



- Better long context performance with tuned SWA size on RULER :

Model	SWA	MK-1	MK-2	MK-3	MQ	MV	S-1	S-2	S-3	Avg.
Transformer++	-	36.4	3.8	0.0	27.9	24.1	94.8	66.0	31.0	35.5
TransformerLS	256	42.8	6.0	0.0	**29.8**	**27.5**	91.8	49.6	23.4	33.9
SWA+YOCO	128	24.2	6.8	0.2	10.2	14.7	81.2	32.6	48.4	27.3
Samba+YOCO	1024	49.0	**28.0**	**2.6**	12.8	18.3	**100.0**	63.2	23.6	37.2
SambaY	256	54.6	27.8	0.4	12.7	19.4	83.2	81.2	63.8	42.9
SambaY+DA	512	**64.6**	27.6	0.2	12.8	19.9	99.8	**86.4**	**69.6**	**47.6**

- Why? Hybrid models have higher data efficiency and long context data is limited.
- SambaY can also extrapolate 2x context length out-of-box on PhoneBook due to NoPE.

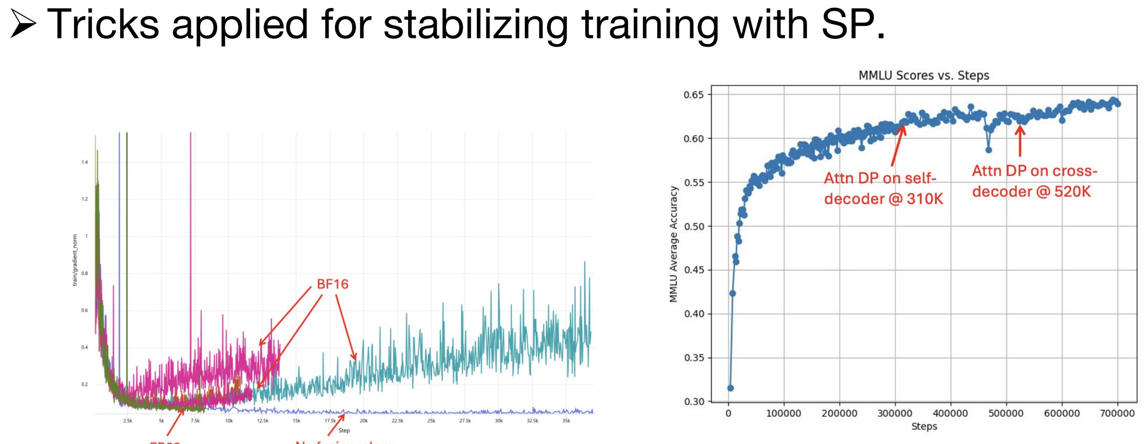
Model	SWA Size	32K Acc. (%)	64K Acc. (%)	128K Acc. (%)
Transformer++	–	60.94 ± 10.00	0.00 ± 0.00	0.00 ± 0.00
TransformerLS	256	60.16 ± 7.12	17.19 ± 5.63	0.78 ± 1.35
Samba+YOCO	1024	82.81 ± 6.44	67.97 ± 11.13	20.31 ± 8.41
SambaY	256	92.19 ± 1.56	96.09 ± 2.59	0.00 ± 0.00
SambaY+DA	512	96.09 ± 1.35	84.38 ± 3.12	5.47 ± 2.59

Large-scale Pretraining

- Phi4-mini-Flash: 3.8B SambaY+Differential Attention model pretrained, mid-trained and post-trained on total 5T tokens.

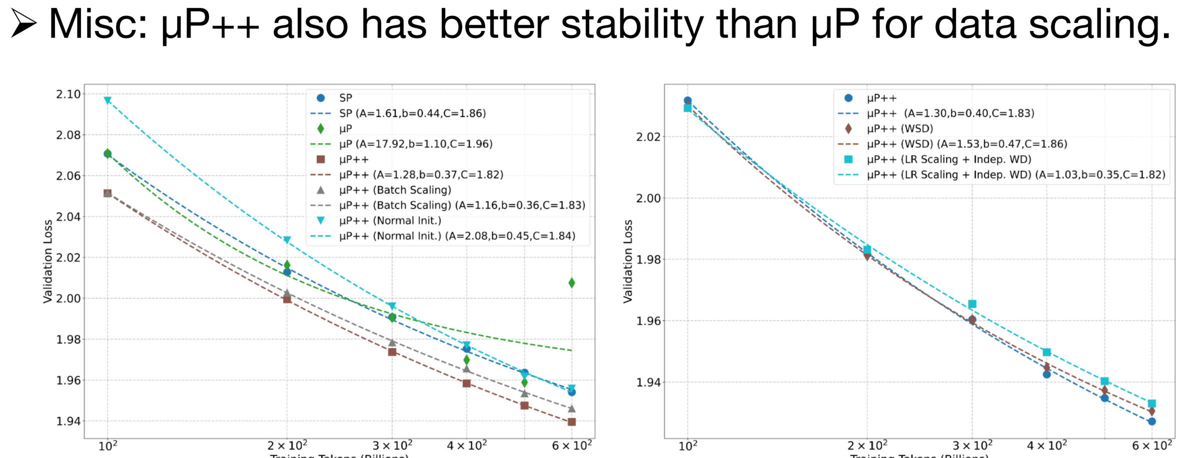
Benchmark	Metric	Phi4-mini	Phi4-mini-Flash
MMLU [HBB+21]	5-shot	67.3	**71.9**
MMLU-Pro [WMZ+24]	0-shot, CoT	52.8	**54.7**
Arena Hard [LCF+24]	Win Rate	32.8	**34.9**
GSM8K [CKB+21]	0-shot, CoT	88.6	**89.5**
Qasper [DLB+21]	F1	**40.4**	40.2
SummScreenFD [CCWG22]	ROUGE-L	16.0	**17.0**
BigCodeBench [ZVC+25]	pass@1	43.0	**44.5**
MBPP [AON+21]	pass@1	65.3	**69.8**

- Tricks applied for stabilizing training with SP.



(a) Fused FP32 LM Head with Cross-Entropy Loss (b) Adding Attention Dropout

- Misc: μP++ also has better stability than μP for data scaling.

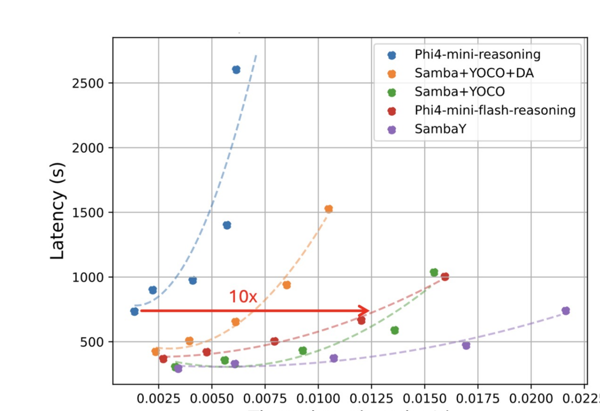


(a) Tied Embedding (b) Untied Embedding

Efficient Long Reasoning

- We further train Phi4-mini-Flash on 150B long CoT data with SFT.
- Pass@1 avg. over 64 runs for AIME 24/25, over 8 for MATH 500 and GPQA-Diamond.

Model	AIME24	AIME25	Math500	GPQA Diamond
DeepSeek-R1-Distill-Qwen-1.5B	29.58	20.78	84.50	37.69
DeepSeek-R1-Distill-Qwen-7B	53.70	35.94	93.03	47.85
DeepSeek-R1-Distill-Llama-8B	43.96	27.34	87.48	45.83
Bespoke-Stratos-7B	21.51	18.28	80.73	38.51
OpenThinker-7B	29.69	24.32	87.25	41.60
Phi4-mini-Reasoning (3.8B)	48.13	31.77	91.20	44.51
Phi4-mini-Flash-Reasoning (3.8B)	**52.29**	**33.59**	**92.45**	**45.08**



(b) Prompt: 2000, Generation: 32000

- 10x Throughput & 2.4x Latency speed-up **even with slow** Differential Attention (DA) and **sub-optimal** vLLM implementation.