

Mixture-of-Experts in the Era of LLMs

A New Odyssey



Tianlong Chen

*The University of North
Carolina at Chapel Hill*



Yu Cheng

*Chinese University of
Hong Kong*



Beidi Chen

*Carnegie Mellon
University*



Minjia Zhang

*University of Illinois
Urbana-Champaign*



Mohit Bansal

*The University of North
Carolina at Chapel Hill*

Sessions	Title	Speakers
1:00-1:25	Overview & Key Challenges in MoEs and their Crucial Roles in LLMs	Tianlong Chen
1:25-1:55	MoE Architecture Variance, Building MoE from Dense LLMs, and MoE Beyond Efficiency	Yu Cheng
1:55-2:10	How to Train a Superior MoE from a System View?	Minjia Zhang
2:10-2:25	Key Extension - Multi-Modal MoE; Multi-Agent Communications	Mohit Bansal, Tianlong Chen
2:25-3:00	Panel - MoE Designs, Multi-Modal Multi-Task MoE, Multi-Agent MoE	Tianlong Chen (Moderator), Yu Cheng, Beidi Chen, Minjia Zhang, Mohit Bansal

Overview & Key Challenges in MoEs and their Crucial Roles in LLMs

TIANLONG CHEN

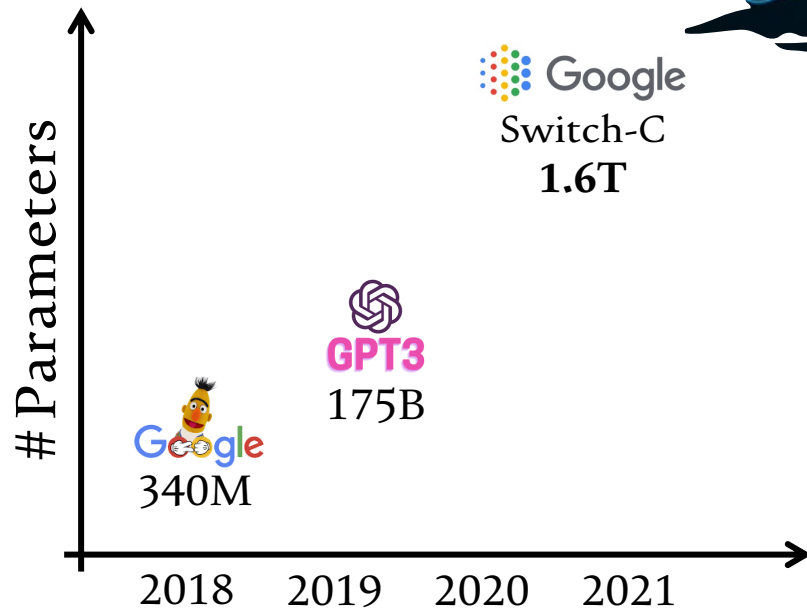
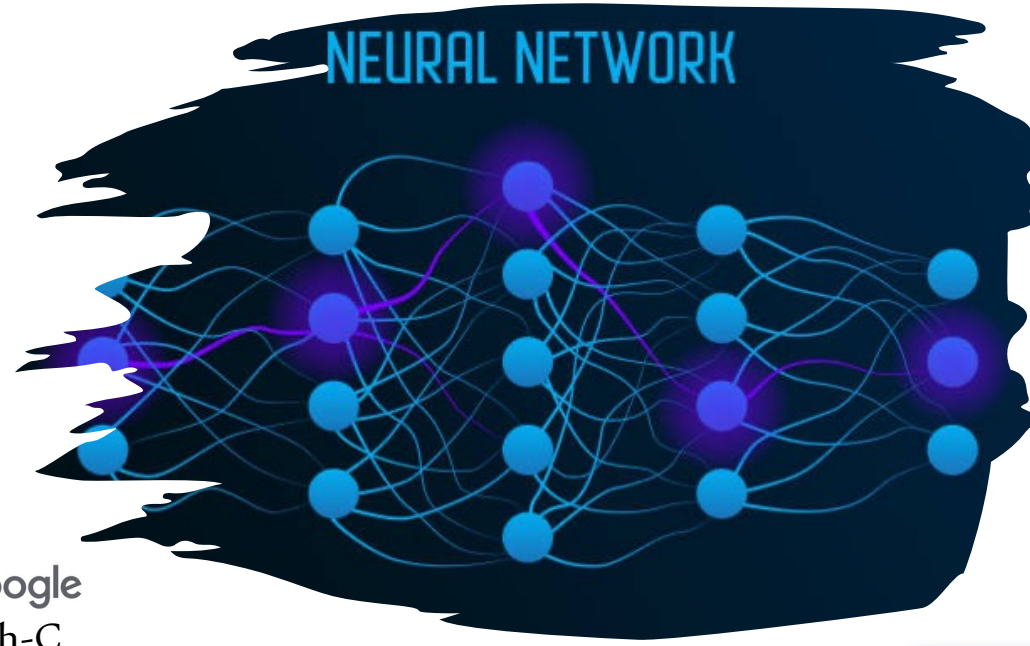
Ph.D., Assistant Professor

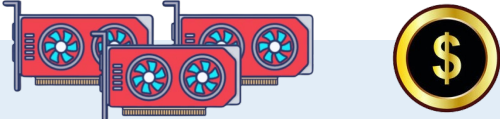
Computer Science

University of North Carolina at Chapel Hill



Current AI models are Dense and Gigantic




Costs >\$10M



What is Mixture-of-Experts?

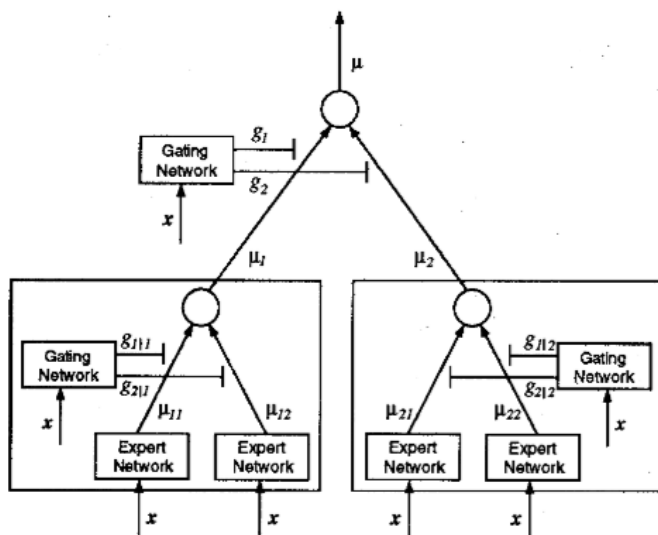


Figure 1: A two-level hierarchical mixture of experts.

Hierarchical Mixtures of Experts
for the EM Algorithm, 1993

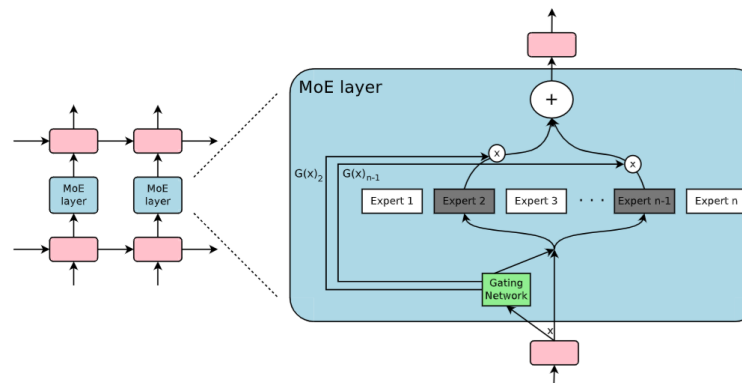
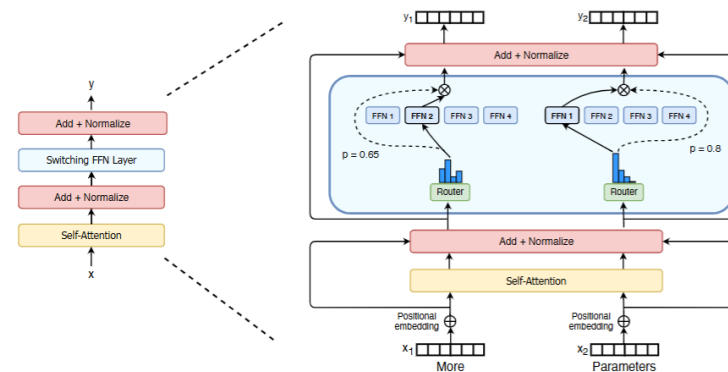


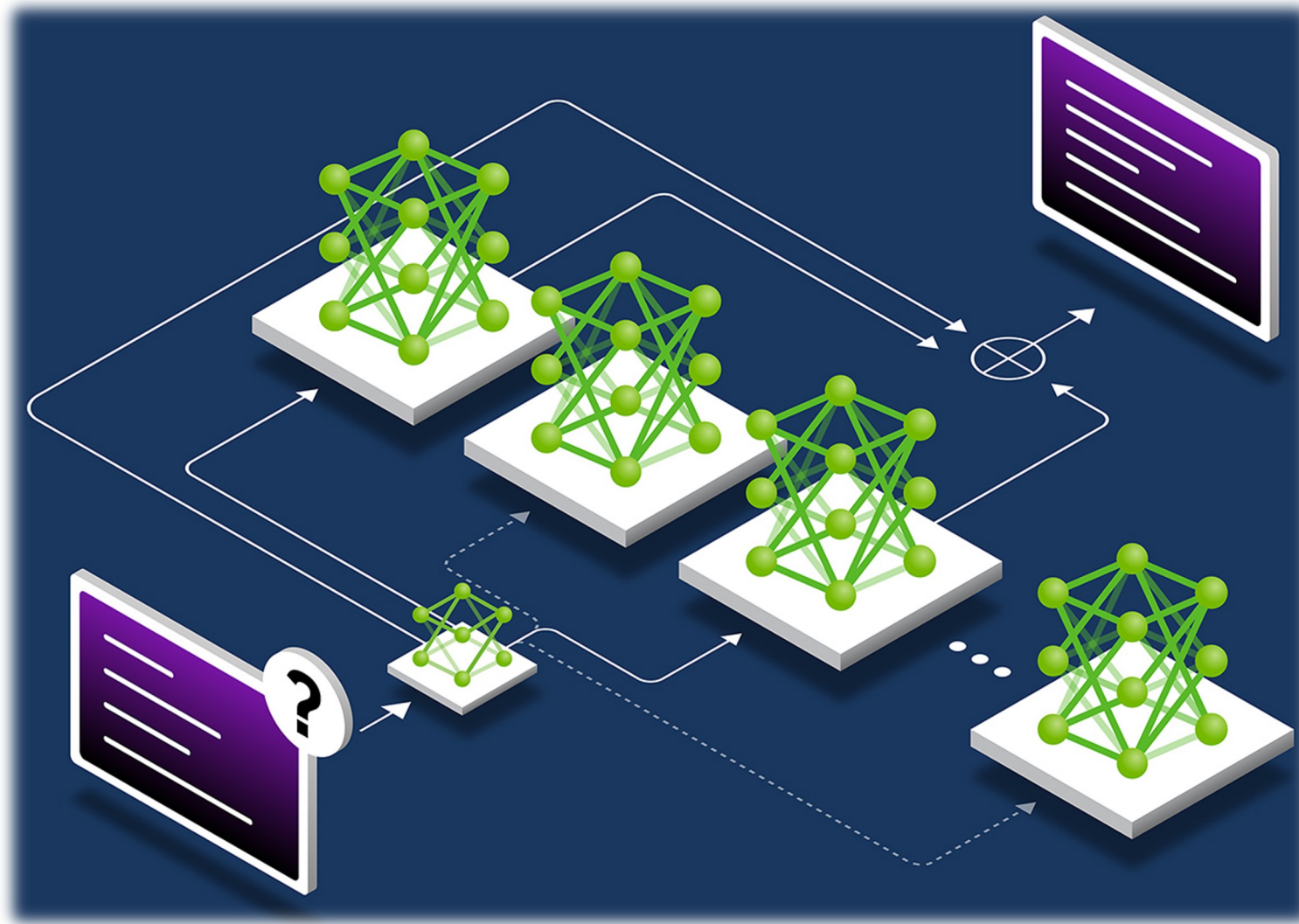
Figure 1: A Mixture of Experts (MoE) layer embedded within a recurrent language model. In this case, the sparse gating function selects two experts to perform computations. Their outputs are modulated by the outputs of the gating network.

Sparse-Gated Mixture of Experts in LSTM, 2017



Sparse-Gated Mixture of Experts in Transformer, 2021

Overview and Key Challenges in MoEs



Unbalanced Routing

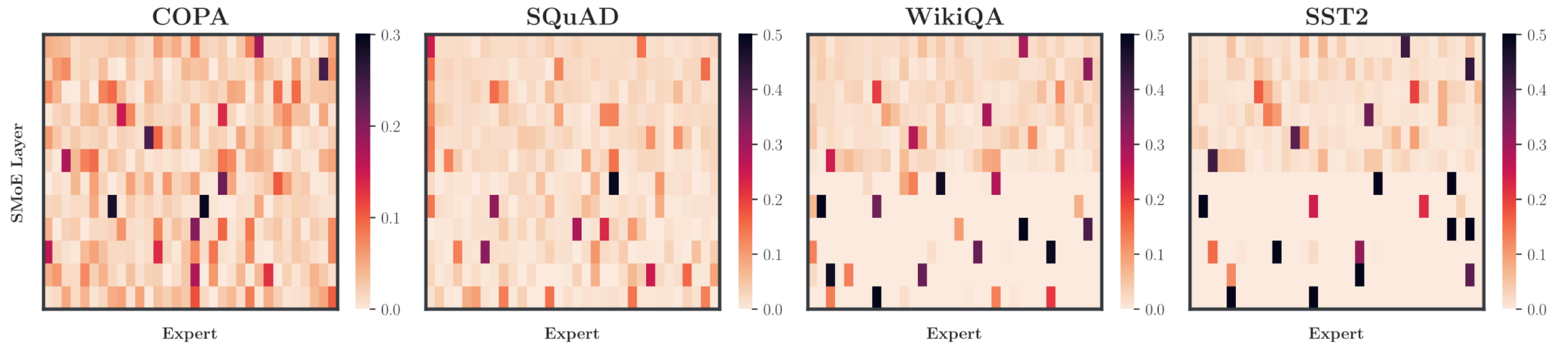
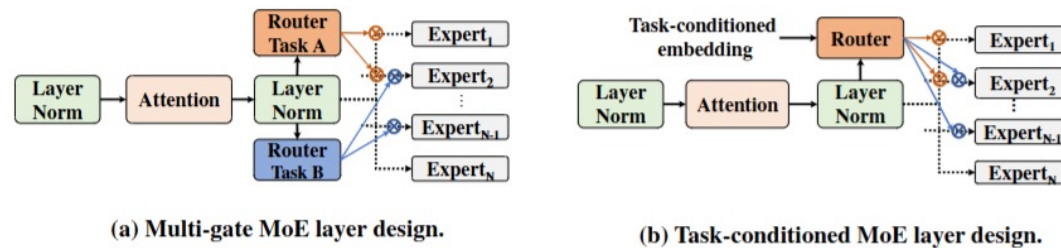
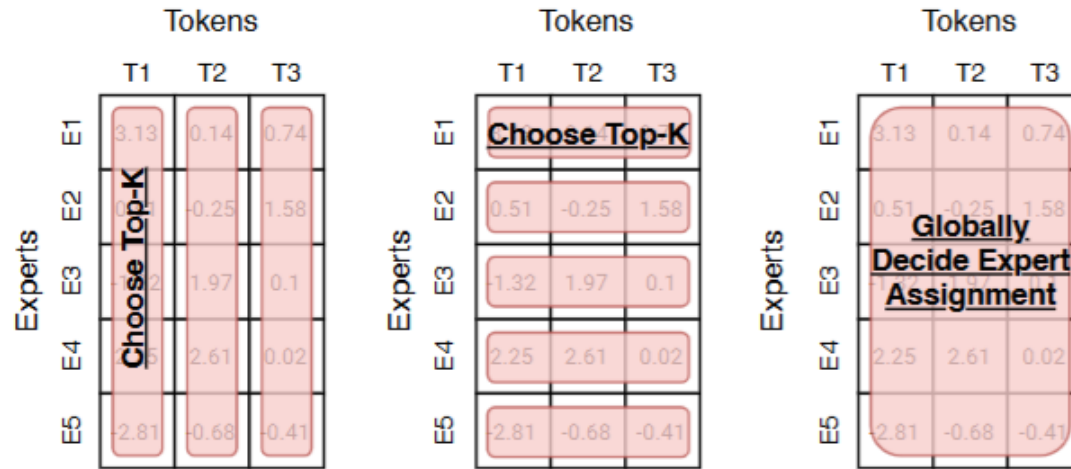
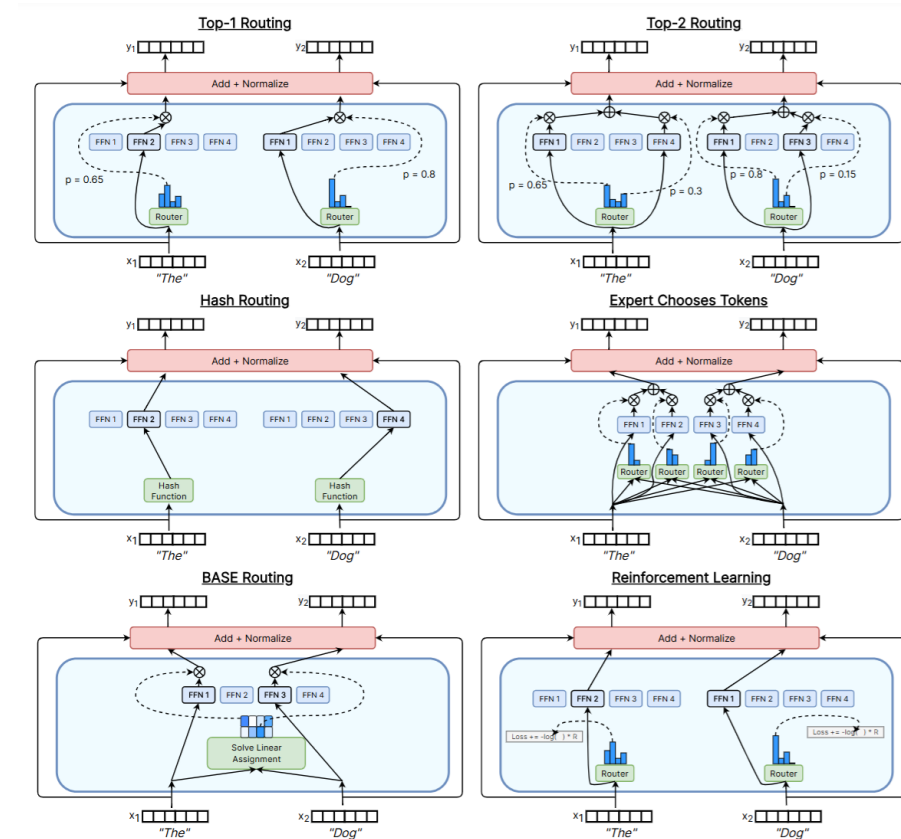


Figure 3: **Distribution of expert activation frequencies** in the *switch-base-32* model, encompassing 12

Routing Algorithms

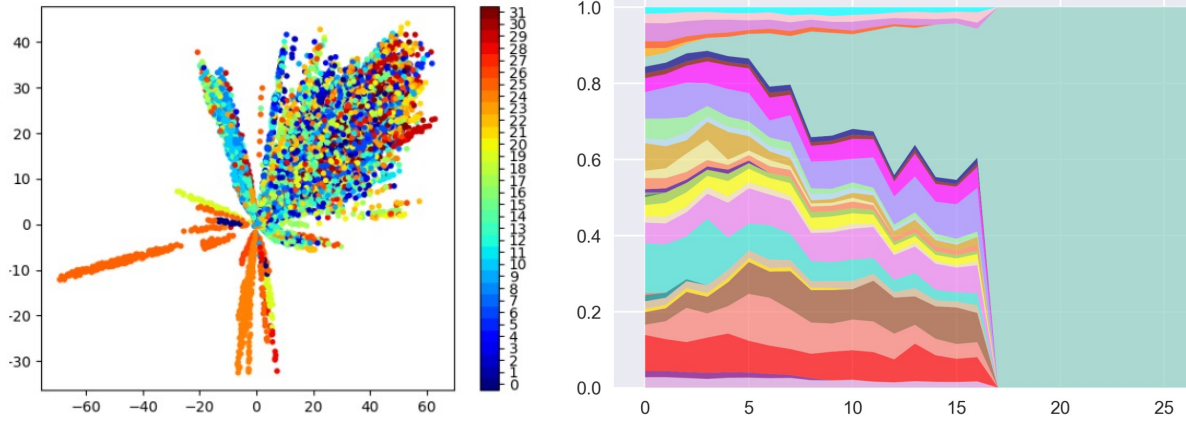


Modality & Task Specific Routing Policy

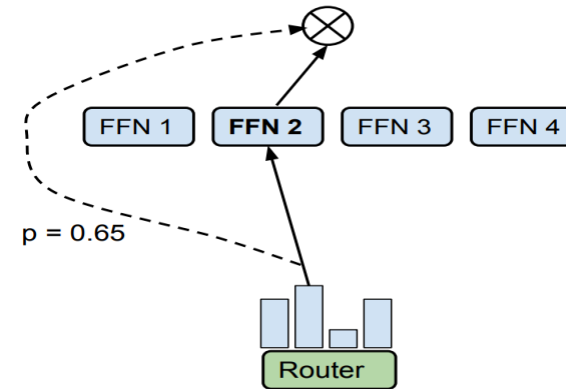


Expert Selection Algorithms

Optimization Artifacts



Representation Collapse & Imbalanced Routing
→ **Redundant Experts** [Chi et. al, 2022]

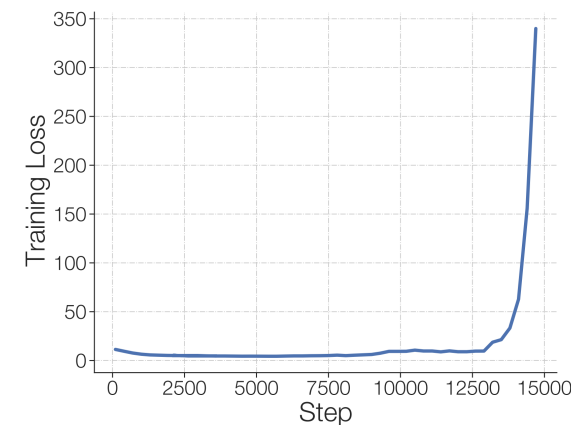
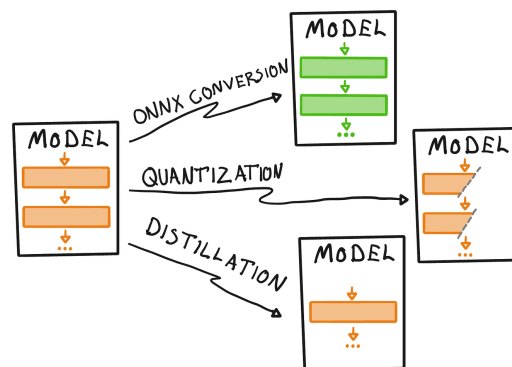


Overfitting the Number of Activated Experts
→ **Poor Scalability** [Riquelme et al. 2022]



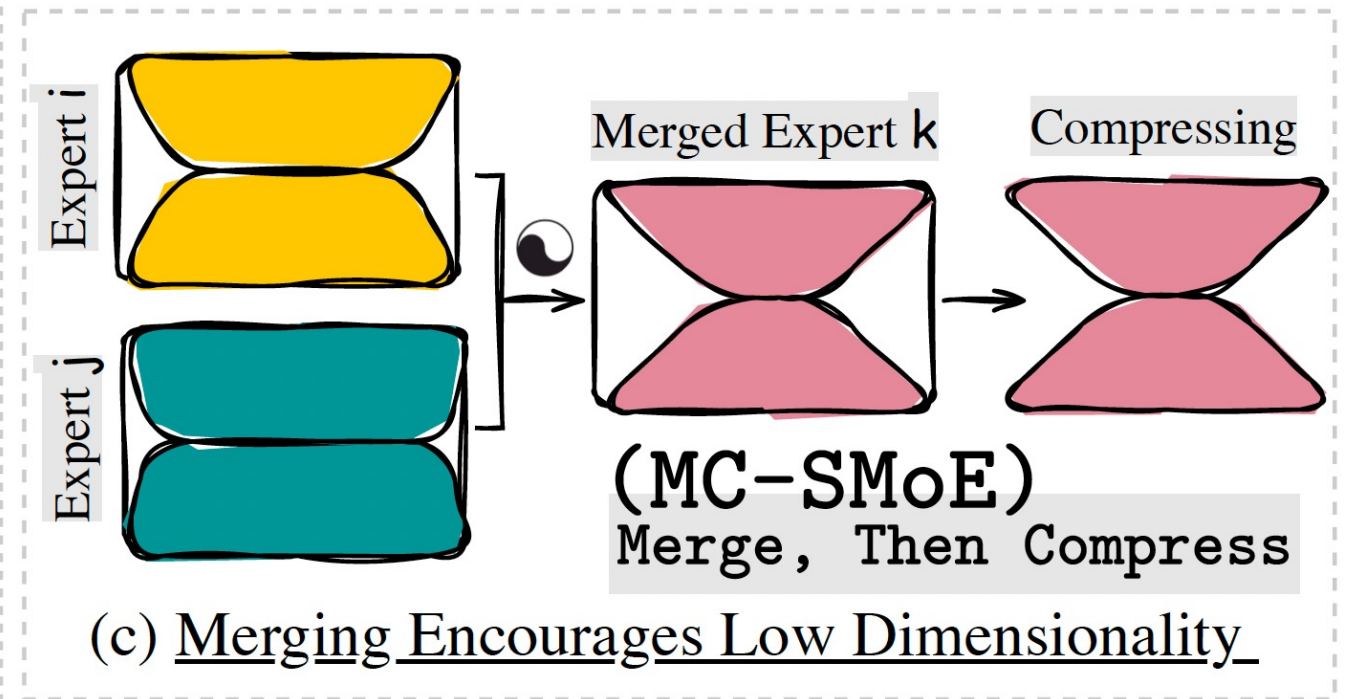
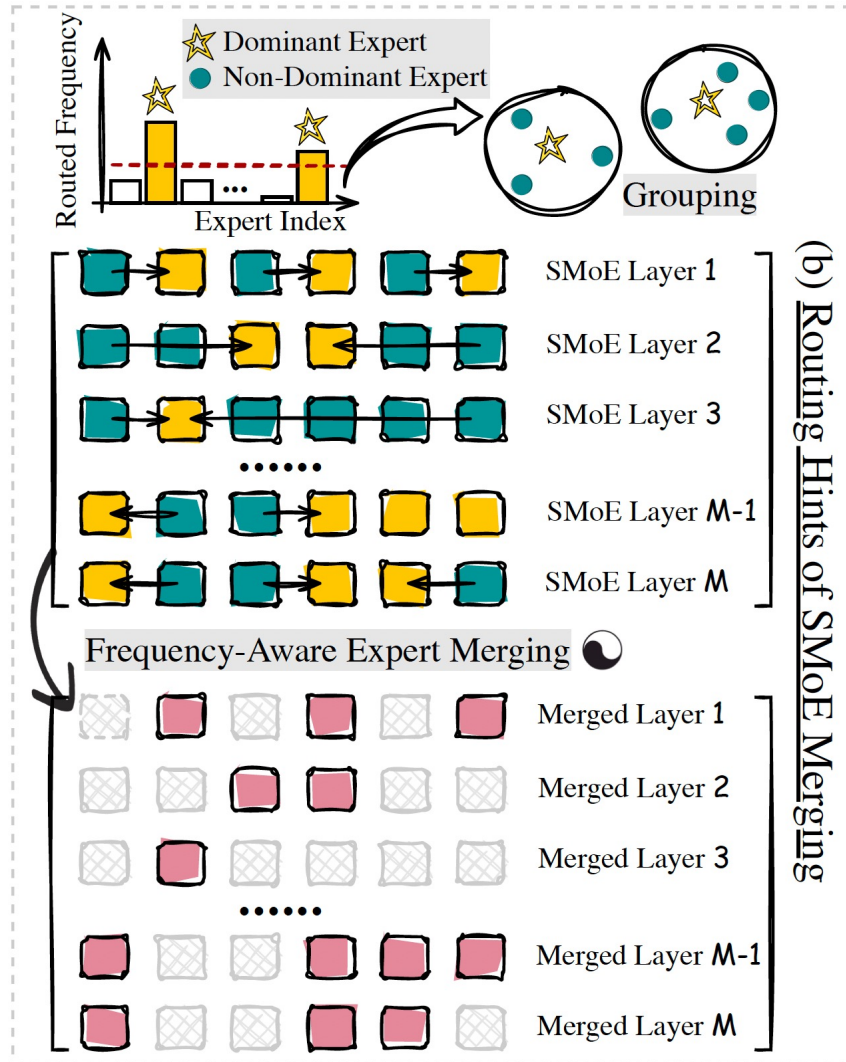
Opportunity for
Compression!

MODEL COMPRESSION



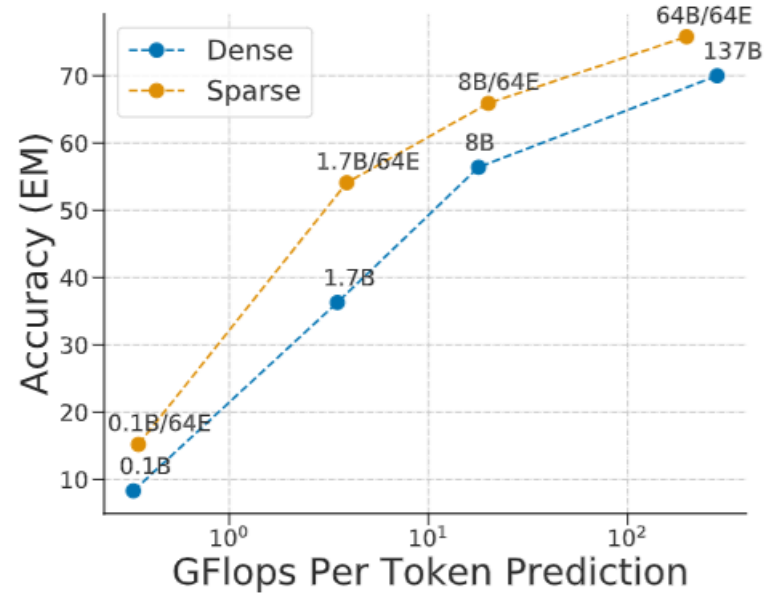
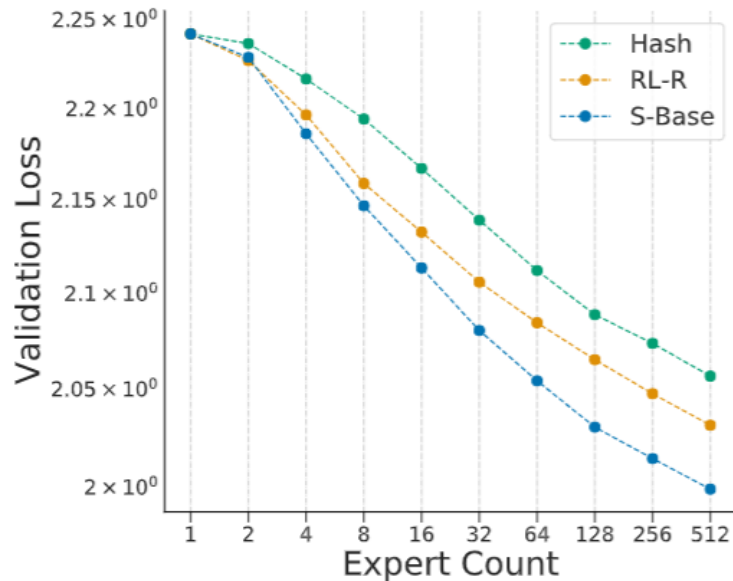
Training Instability
[Barret et. al, 2022]

Merge, Then Compress in SMoE



Scalability

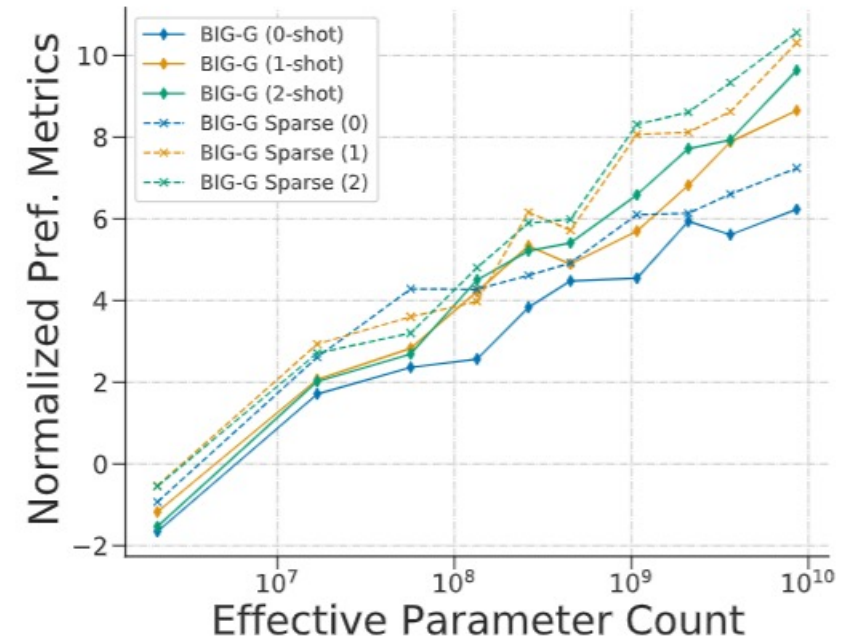
Upstream Performance
w.r.t Expert Count



Downstream Performance
w.r.t Inference Overhead

Superior and more
complex scalability
than dense models!

Downstream Performance
w.r.t Parameter Count



Interpretability and Specialization



Figure 8: Text samples where each token is colored with the first expert choice. The selection of experts appears to be more aligned with the syntax rather than the domain, especially at the initial and final layers.

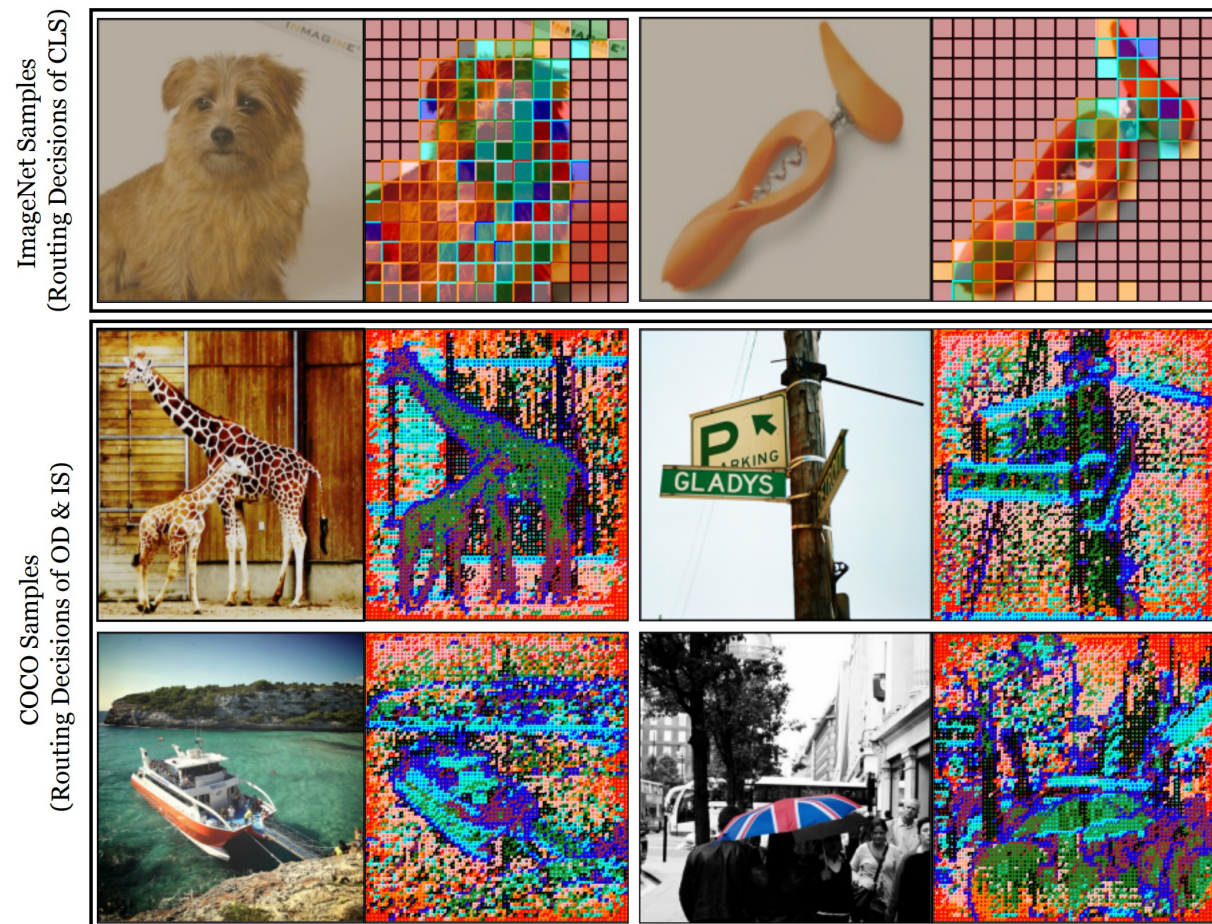
First Expert Choice of Text Samples Mixtral[Arxiv, 2024]

Patch Choices of Each Expert LIMO E [NeurIPS, 2022]



Figure 2: Token routing examples for Coco. Image examples of how patches are routed at the MoE layer placed in the 18-th encoder block –i.e. middle of the network– for the LIMoE-H/14 model.

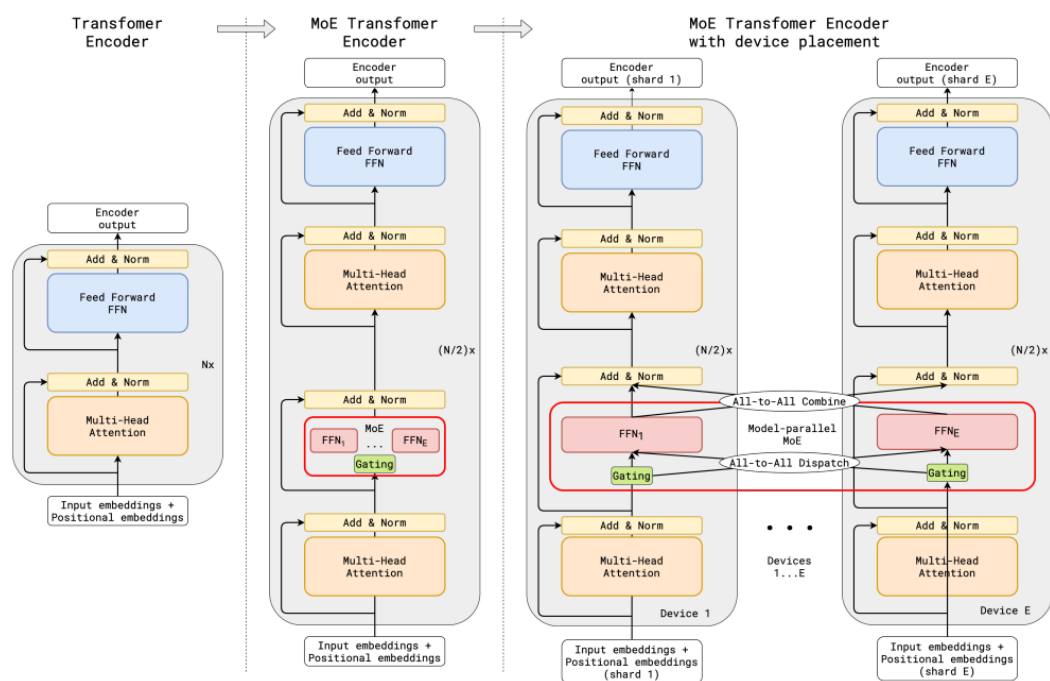
Interpretability and Specialization



Our ICCV'23

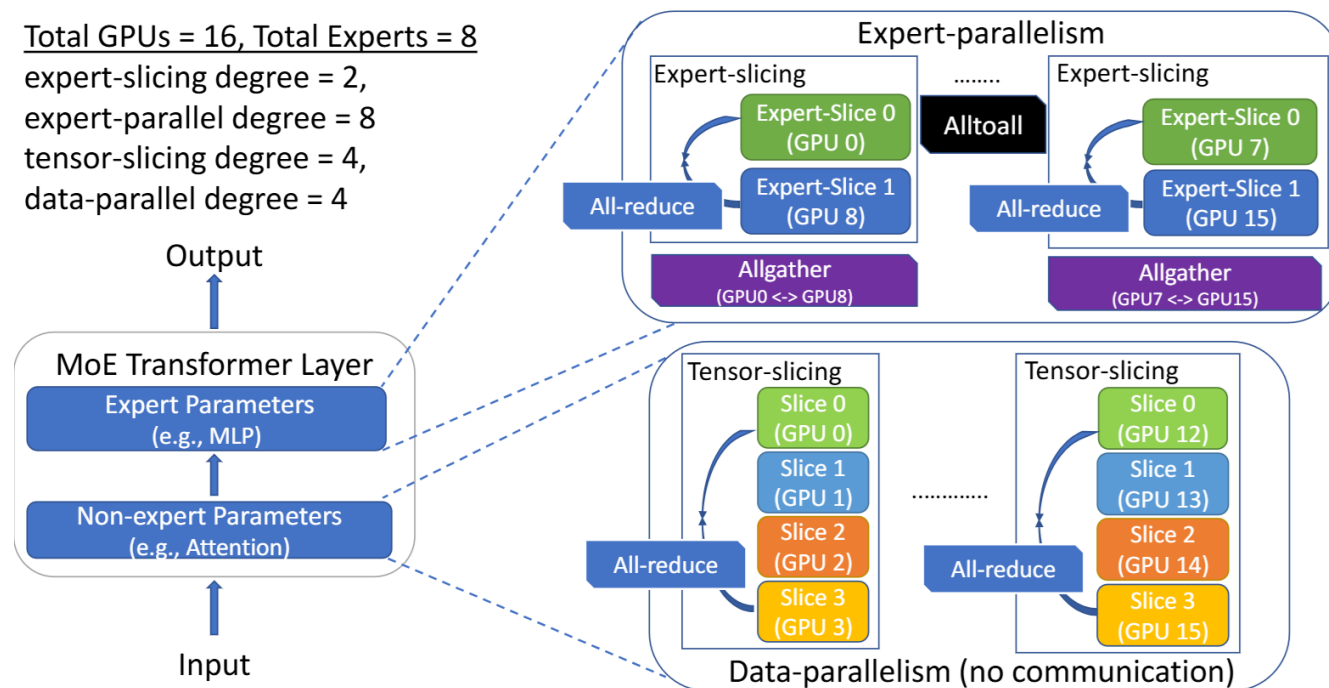
Hardware Support

Q: How to efficiently place different experts across multiple accelerators?



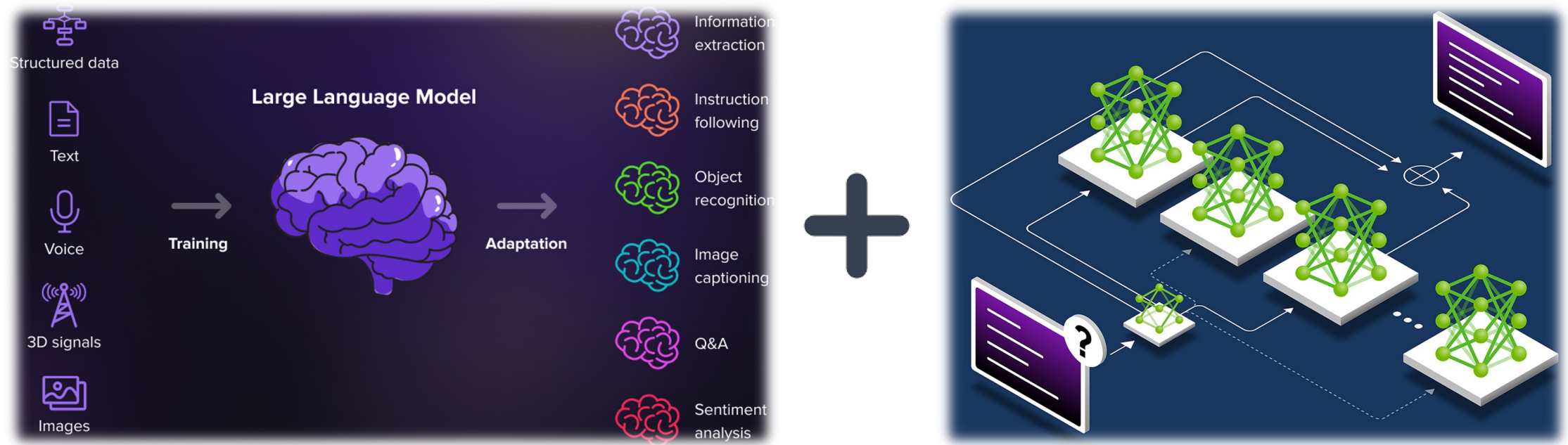
Shared MoE Layers Between Devices
GShard [ICLR'21]

Total GPUs = 16, Total Experts = 8
expert-slicing degree = 2,
expert-parallel degree = 8
tensor-slicing degree = 4,
data-parallel degree = 4



Multi-Dimensional Parallelism
DeepSpeed-MoE [Arxiv'24]

Crucial Roles of MoE in LLMs





MoE LLMs are Impressively Efficient!



8x7

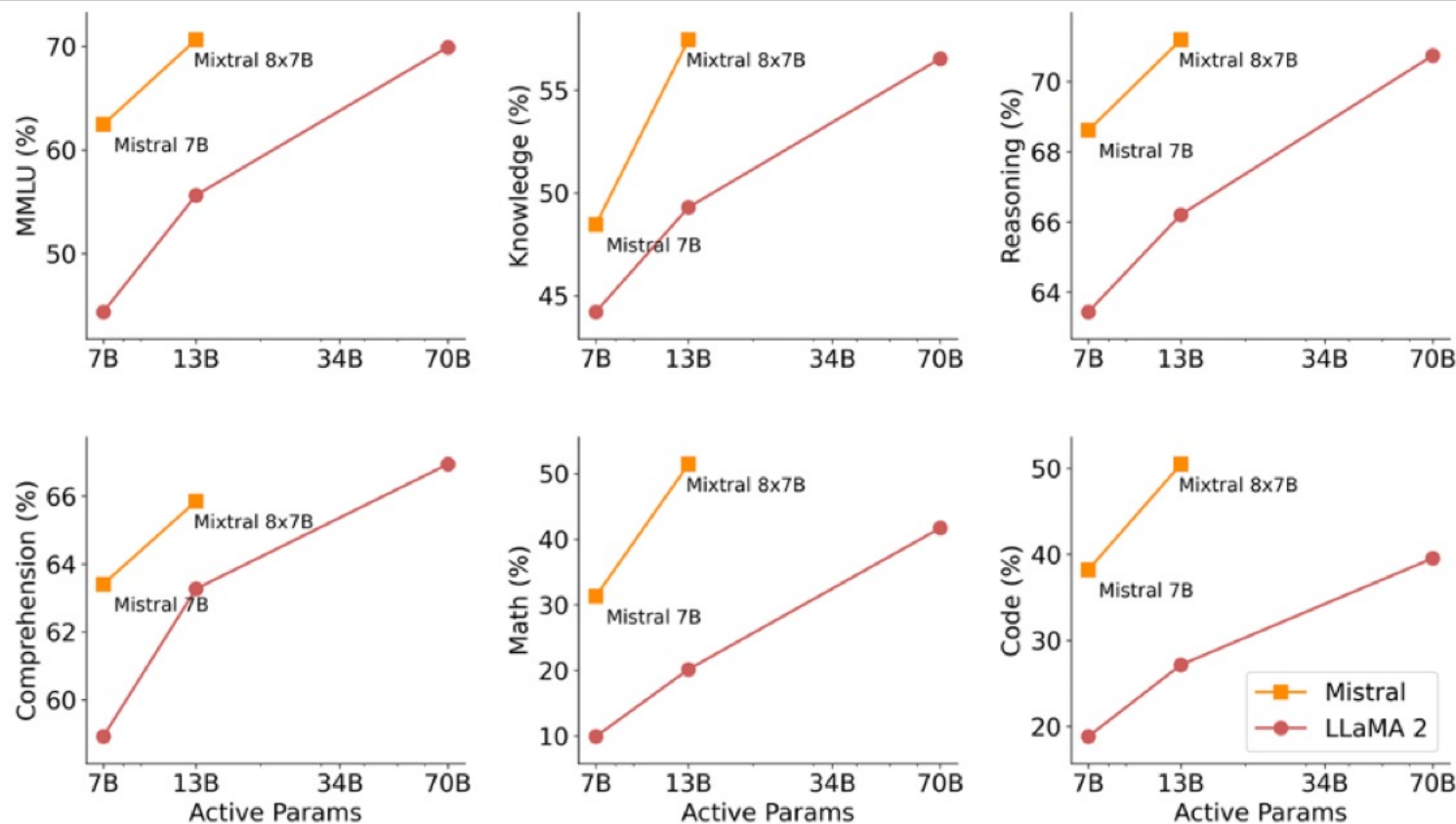
deepseek-ai/
DeepSeek-MoE

DeepSeekMoE: Towards Ultimate Expert
Specialization in Mixture-of-Experts Language
Models

1 Contributor 10 Issues 909 Stars 39 Forks



Qwen1.5-MoE

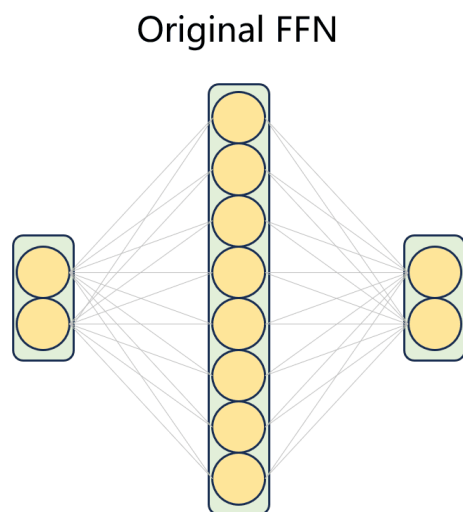


Popular MoE LLMs

Mixtral 8x7b [Jiang et. al, 2024]

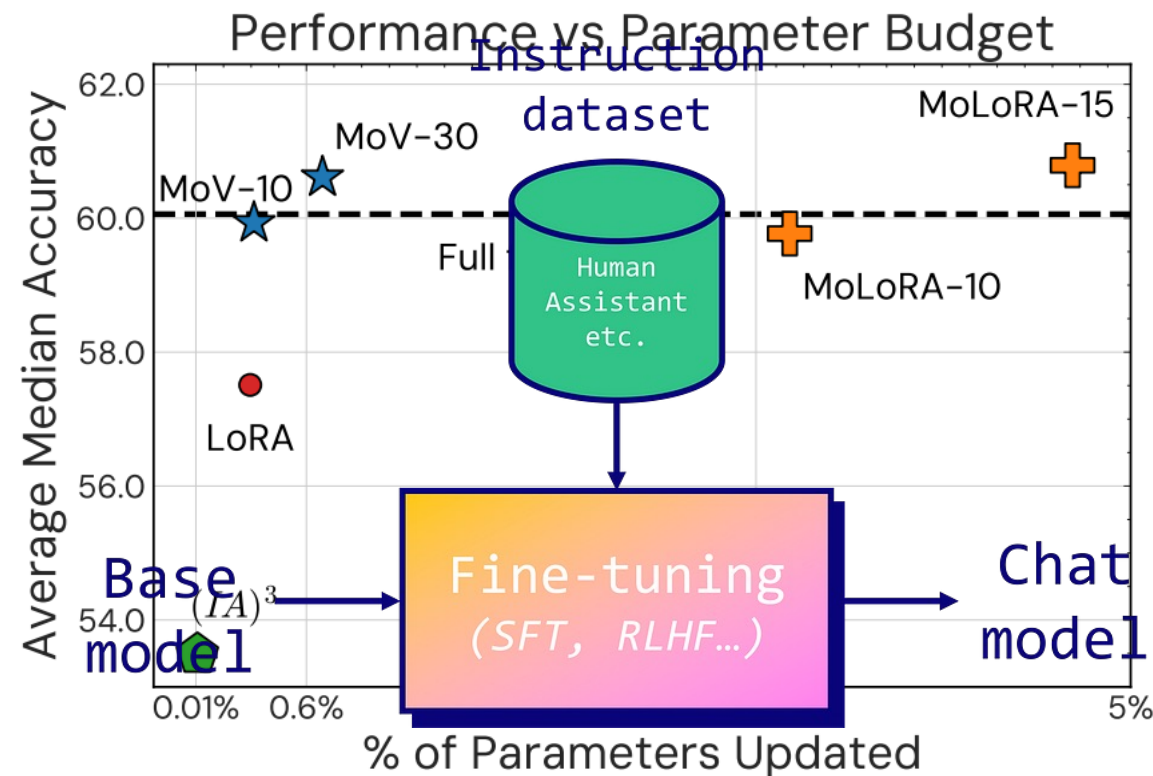
- ✓ at inference, 6x faster than Llama 2 70B
- ✓ matches or outperforms GPT3.5 & Llama 2 70B

Efficiently Training and Deployment of MoE LLMs



Build up MoE from dense LLMs

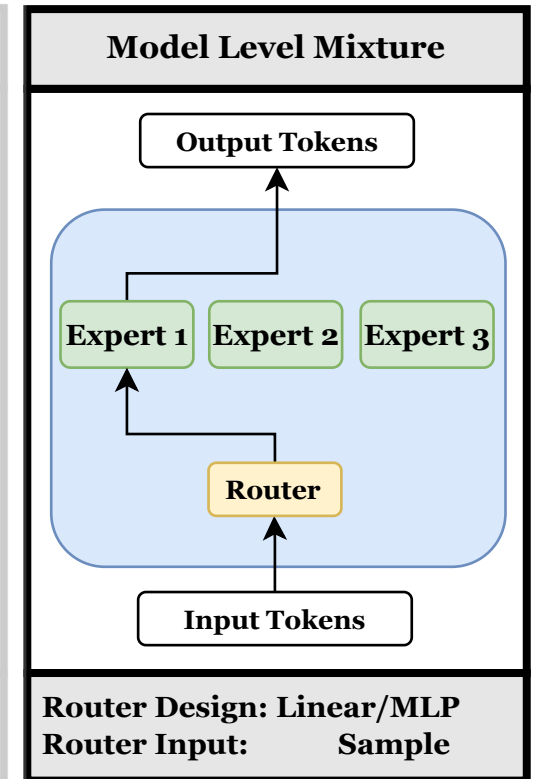
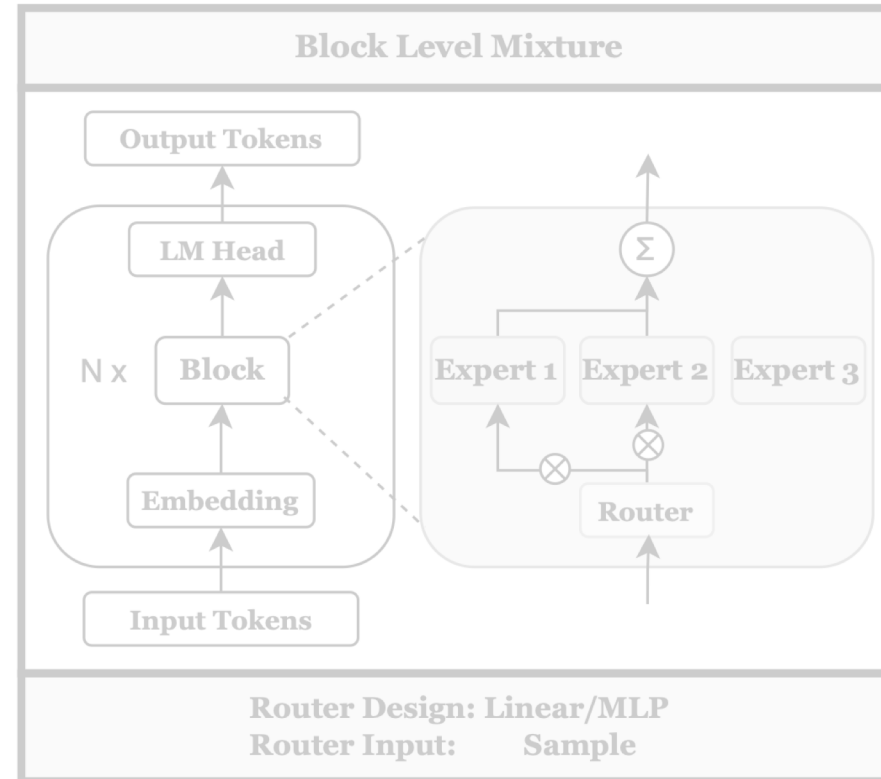
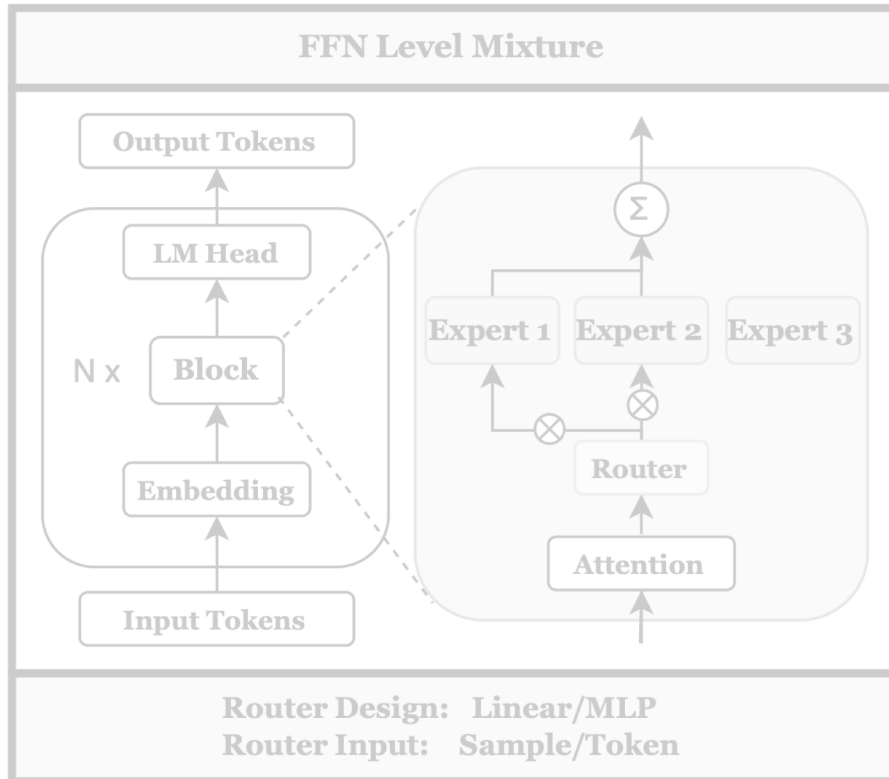
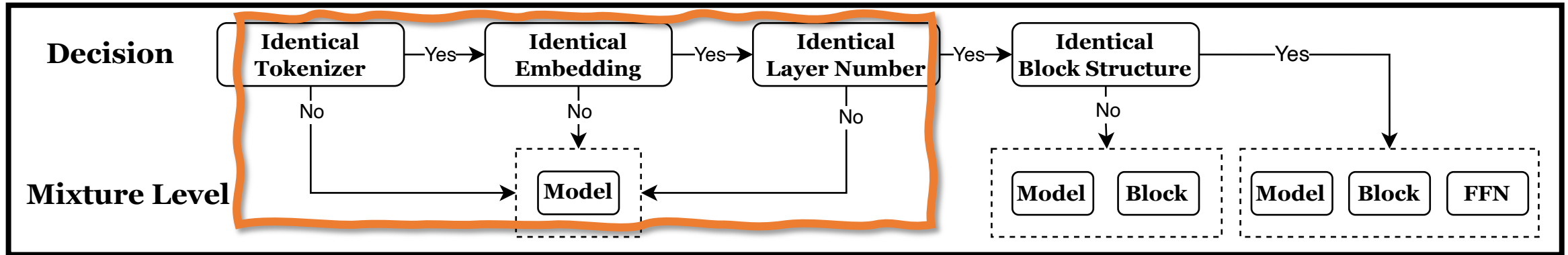
- ✓ Expert construction and continuous pre-training [LLaMA-MoE Team, 2023]



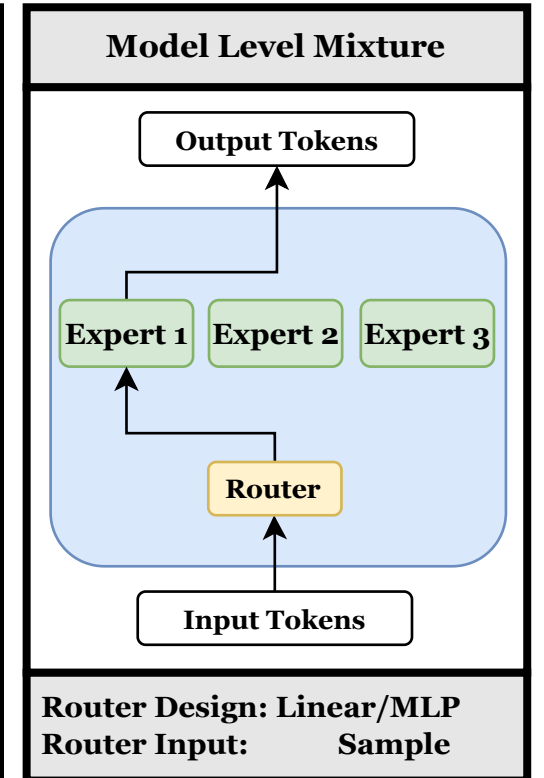
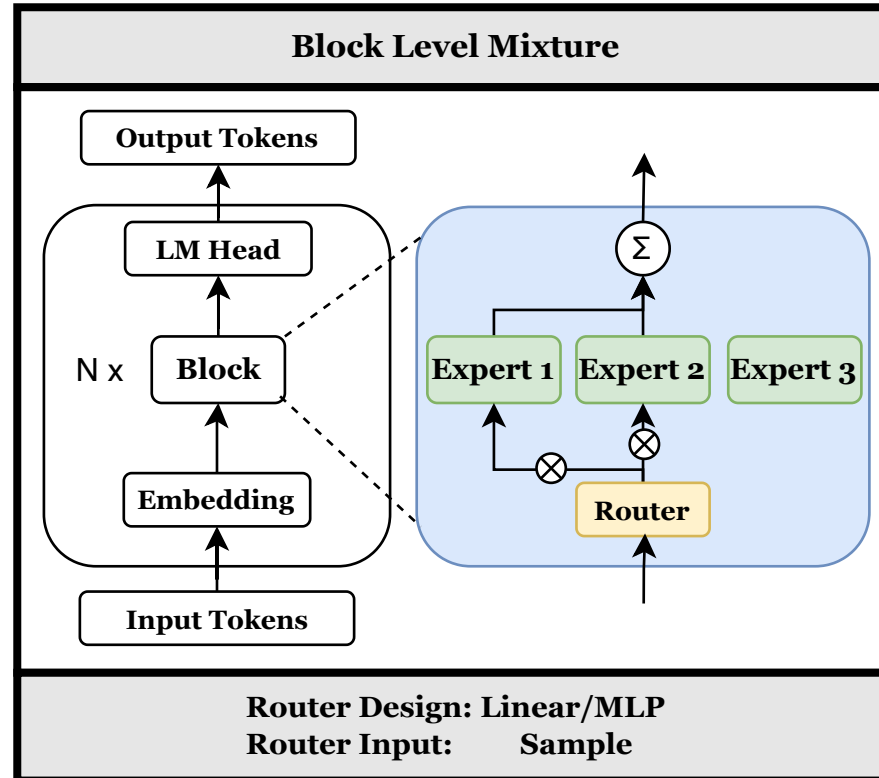
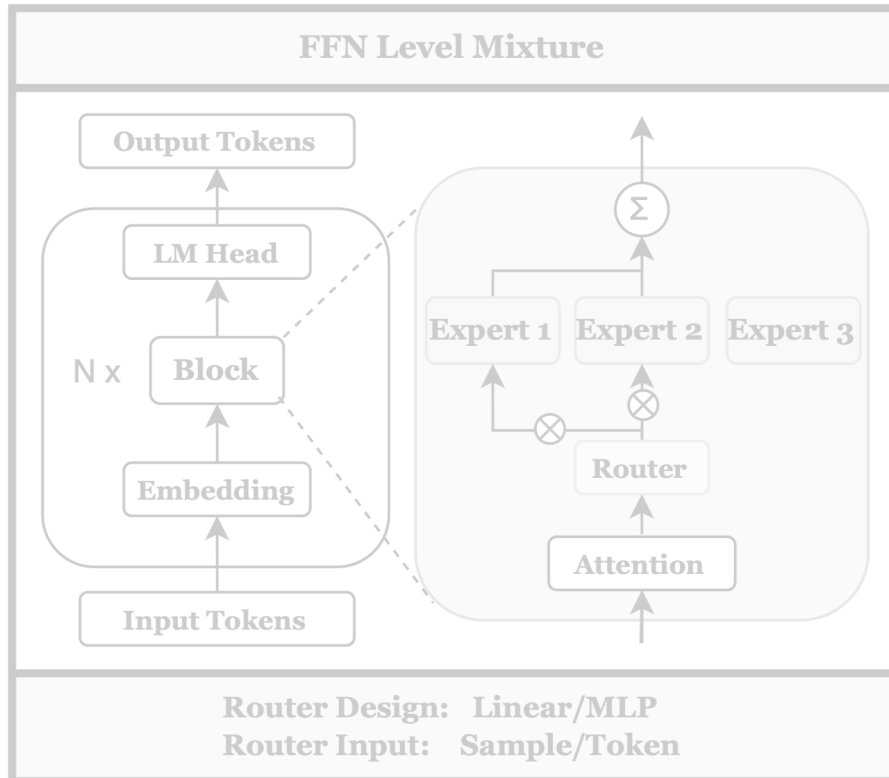
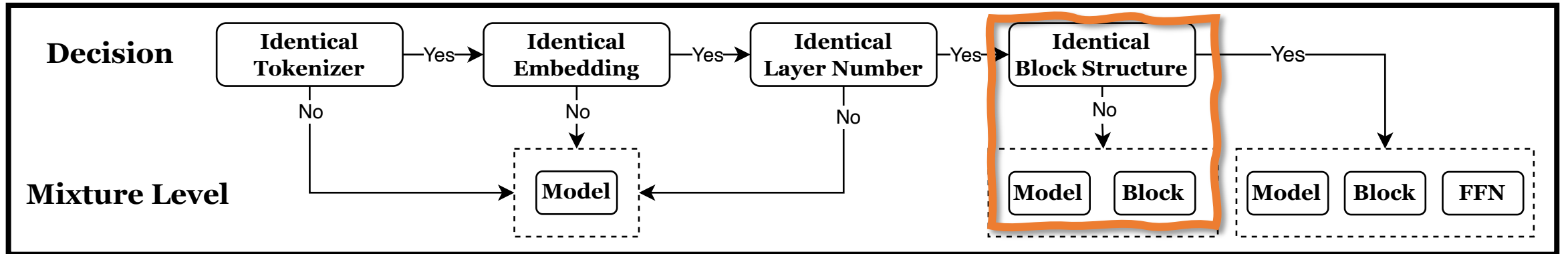
Downstream adaptation of MoE

- ✓ Instruction tuning [Shen et. al, 2023]
- ✓ Parameter-efficient fine-tuning: MoLoRA & MoV [Zadouri et. al, 2023]

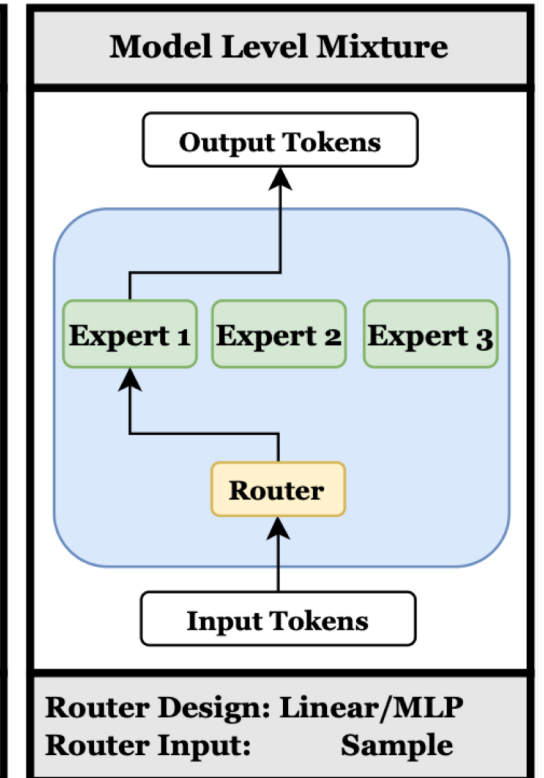
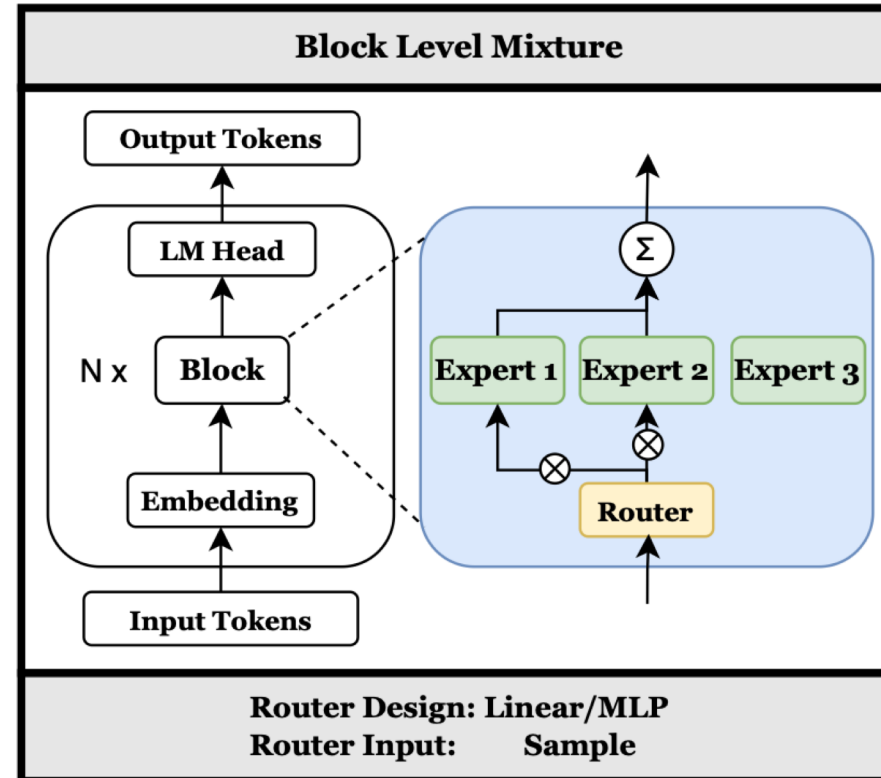
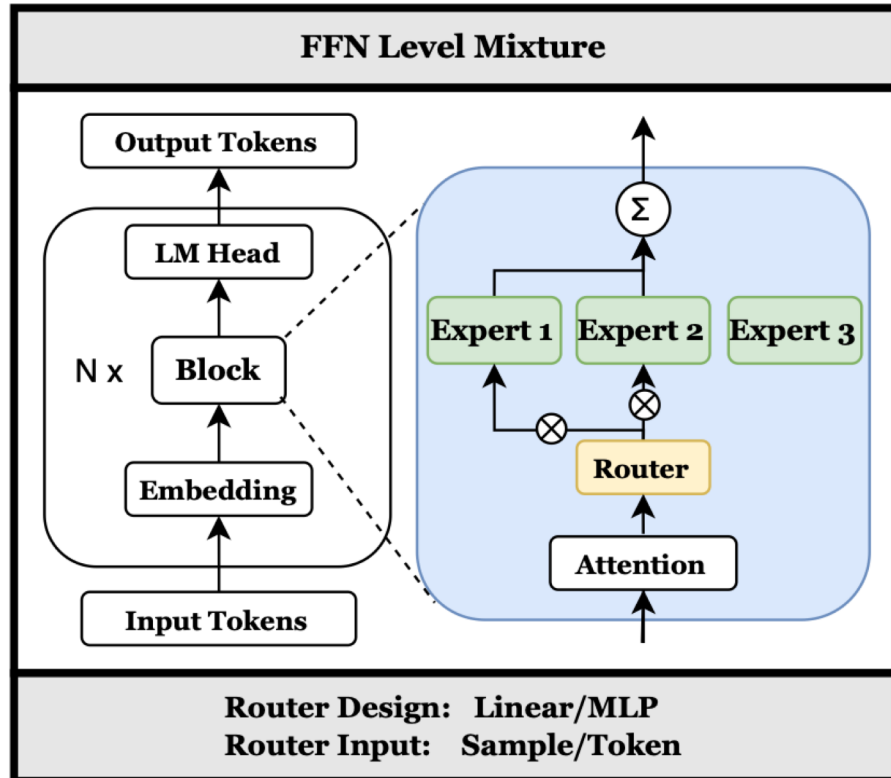
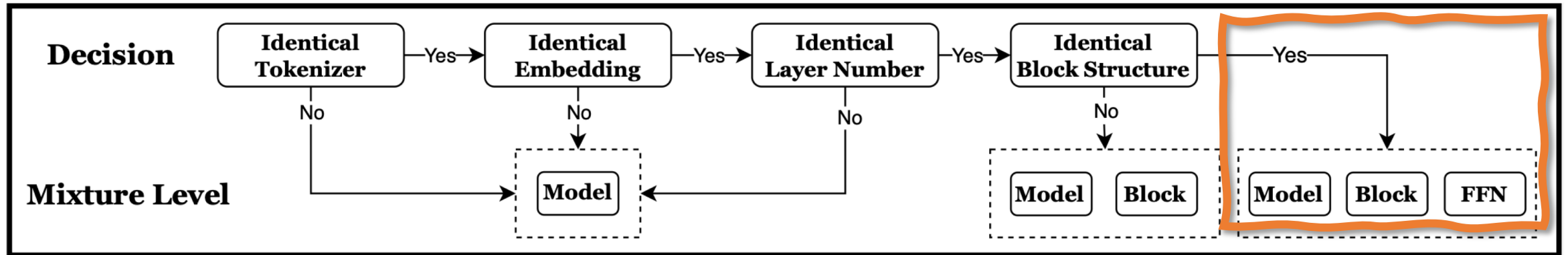
Mix diverse LLMs like MoE!



Mix diverse LLMs like MoE!



Mix diverse LLMs like MoE!



Empirical Investigations

RQ1: *At which level does the model mixture manifest its utmost effectiveness?*

A1: *Model level mixture is consistently better.*

Table 1: Model mixture methods and their abbreviations used in our study. Methods applicable for models with distinct architectures are highlighted in gray .

Abbreviation	Mix. Level	Router	Router Input
F-L-T	FFN	Linear	Token
F-L-S	FFN	Linear	Sample
F-M-S	FFN	MLP	Sample
B-L-S	Block	Linear	Sample
B-M-S	Block	MLP	Sample
M-L-S	Model	Linear	Sample

Table 2: Comparison of different mixture levels. For each task in each model zoo, we highlight the performance best in each model zoo in **bold**.

Model	ARC	WinoGrande	MMLU	GSM8K	MBPP	HumanEval	Average
Which2							
Best Single Model	54.27%	71.51%	47.24%	21.30%	18.00%	13.06%	37.68%
F-L-S	52.82%	70.80%	50.04%	23.12%	19.00%	17.68%	38.91%
B-L-S	52.73%	70.01%	49.90%	19.94%	18.84%	15.85%	37.88%
M-L-S	54.44%	72.38%	50.51%	22.21%	20.00%	20.73%	40.04%
Which4							
Best Single Model	55.03%	73.72%	48.33%	24.26%	17.80%	13.41%	38.70%
F-L-S	53.75%	73.88%	47.97%	34.87%	21.80%	23.17%	42.57%
B-L-S	52.65%	74.66%	47.05%	21.15%	20.40%	14.63%	38.42%
M-L-S	49.06%	72.14%	41.81%	60.05%	17.60%	15.24%	42.65%

✓ **Which2:** *Llama-2-7b-chat-hf, vicuna-7b-v1.5*

✓ **Which4:** *Synthia-7B-v1.2, Llama-2-7b-evolvealpaca, pygmalion-2-7b, MetaMath-7B-V1.0*

Empirical Investigations

RQ2: *Does more complex router design brings better results?*

A2: *Not necessary, as the linear router outperforms the MLP router.*

Table 3: Comparison between linear and MLP routers on Which2 setting. We highlight better performance within each pair in **bold**.

Model	ARC	WinoGrande	MMLU	GSM8K	MBPP	HumanEval	Average
F-L-T	53.41%	70.48%	50.74%	23.28%	20.80%	16.46%	39.20%
F-M-T	53.58%	72.06%	50.01%	21.92%	17.40%	17.68%	38.78%
B-L-S	52.73%	70.01%	49.90%	19.94%	18.84%	15.85%	37.88%
B-M-S	51.53%	70.56%	49.41%	19.94%	16.60%	14.02%	37.01%

- ✓ **Linear Router:** initialized from the prompt vector following [Beyond](#), training-free.
- ✓ **MLP Router:** randomly initialized, and fine-tuned on GPT4All by only updating the router.

Empirical Investigations

RQ3: Which router input is better, token-level or sample-level?

A3: Not quite different. Token input suits a mixture of the same domain models.

Table 5: Comparison of different router input designs. **Which4** includes one group with chatting models (**Chat**) and another with different domain models (**Domain**) . For each task in each model zoo, we highlight the performance best among all model mixture methods in **bold**.

Model	ARC	WinoGrande	MMLU	GSM8K	MBPP	HumanEval	Average
Which2							
Best Single Model	54.27%	71.51%	47.24%	21.30%	18.00%	13.06%	37.68%
F-L-T	53.41%	70.48%	50.74%	23.28%	20.80%	16.46%	39.20%
F-L-S	52.82%	70.80%	50.04%	23.12%	19.00%	17.68%	38.91%
Which4							
Best Single Model	55.03%	73.72%	48.33%	24.26%	17.80%	13.41%	38.70%
Chat F-L-T	55.63%	72.77%	50.28%	23.88%	20.00%	22.56%	40.85%
Chat F-L-S	53.75%	70.96%	49.78%	20.32%	20.40%	20.12%	39.22%
Domain F-L-T	55.72%	74.11%	48.32%	30.17%	22.00%	20.12%	41.74%
Domain F-L-S	53.75%	73.88%	47.97%	34.87%	21.80%	23.17%	42.57%

- ✓ **Which4 Chat:** *Synthia-7B-v1.2, OpenHermes-7B, Llama-2-7b-chat-hf, vicuna-7b-v1.5*
- ✓ **Which4 Domain:** *Synthia-7B-v1.2, Llama-2-7b-evolvealpaca, pygmalion-2-7b, MetaMath-7B-V1.0*

Empirical Investigations

RQ4: *Is it feasible for hybrid mixtures to provide enhancements?*

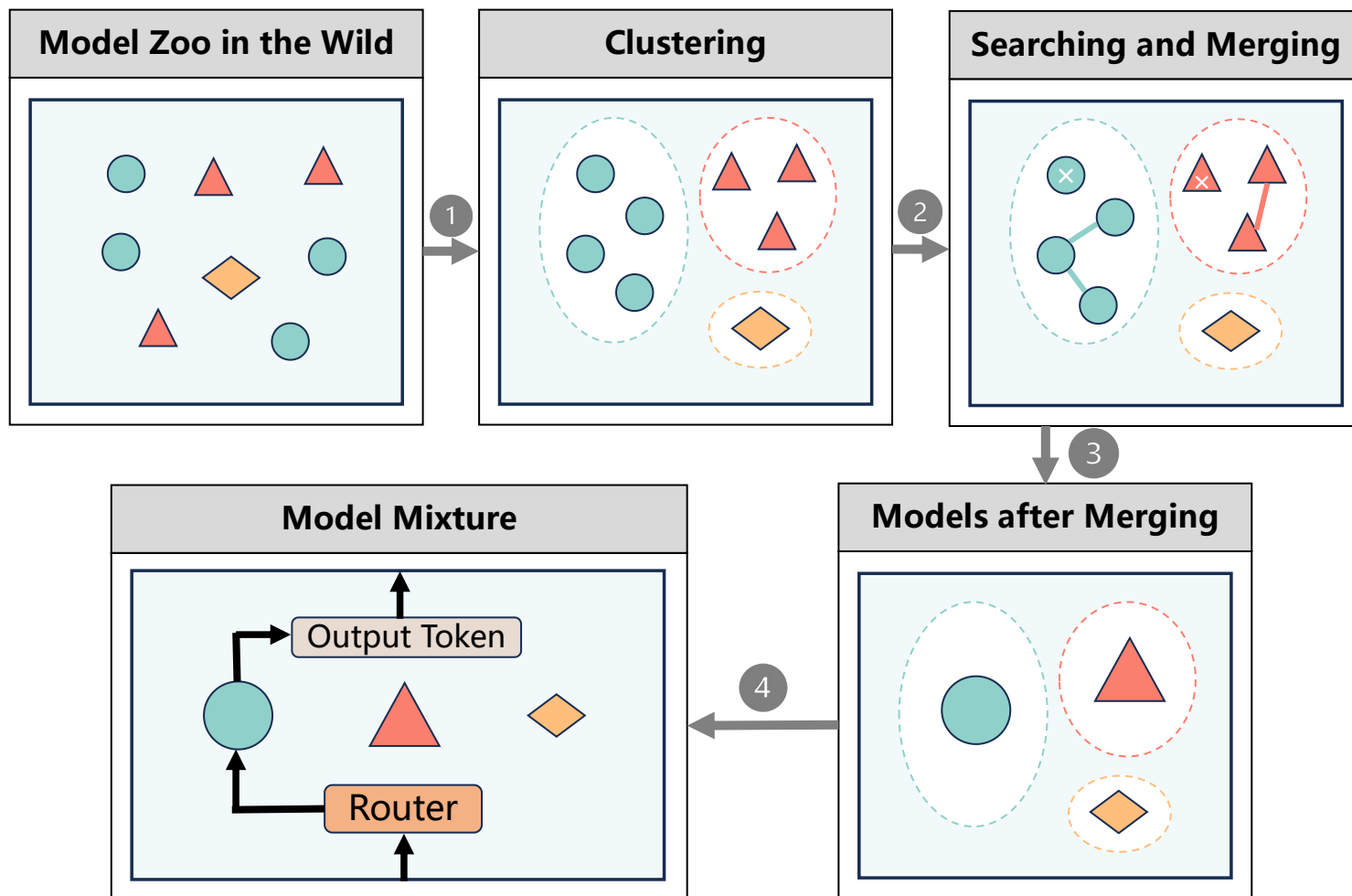
A4: *Yes, hybrid mixture significantly improves on math and code tasks.*

Table 6: Comparison between F-L-T methods with and without hybrid mixture technique. For each task in each model zoo, we highlight the performance best among all model mixture methods in **bold**.

Model	ARC	WinoGrande	MMLU	GSM8K	MBPP	HumanEval	Average
Which2							
Best Single Model	54.27%	71.51%	47.24%	21.30%	18.00%	13.06%	37.68%
F-L-T	53.41%	70.48%	50.74%	23.28%	20.80%	16.46%	39.20%
Hybrid F-L-T	54.44%	71.19%	50.45%	23.96%	21.80%	18.29%	40.02%
Which4							
Best Single Model	55.03%	73.72%	48.33%	24.26%	17.80%	13.41%	38.70%
F-L-T	55.72%	74.11%	48.32%	30.17%	22.00%	20.12%	41.74%
Hybrid F-L-T	54.86%	73.80%	48.23%	37.53%	24.30%	23.17%	43.65%

- ✓ **Hybrid Mixture:** the bottom 16 layers of all single LLMs are merged, and then the rest layers follow any of the mixture level designs.

Efficient LLM Scaling with Model Merging and Mixture



Our Best LLM scaling Recipe:

- ① **Model Clustering** based on model architecture and weight's cosine similarity;
- ② **Model Filtering and Searching**;
 - Heuristic Strategy to search models to merged and merging coefficients
 - Evolutionary Strategy for fine-tune the coefficients
- ③ **Model Merging** within each cluster;
 - Merging Method: Linear merging
- ④ **Model Level Mixture** of merged clusters

Empirical Investigations

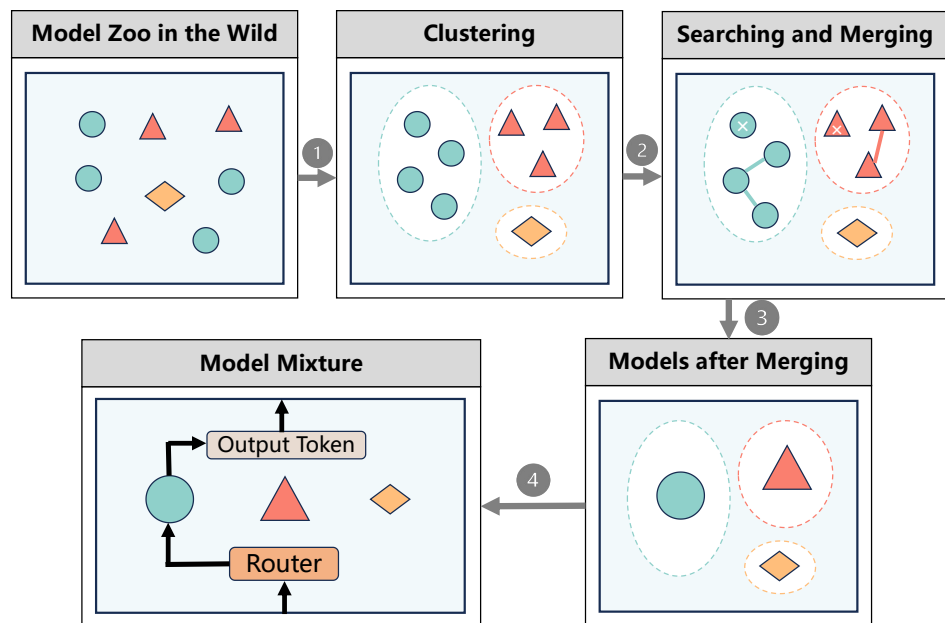


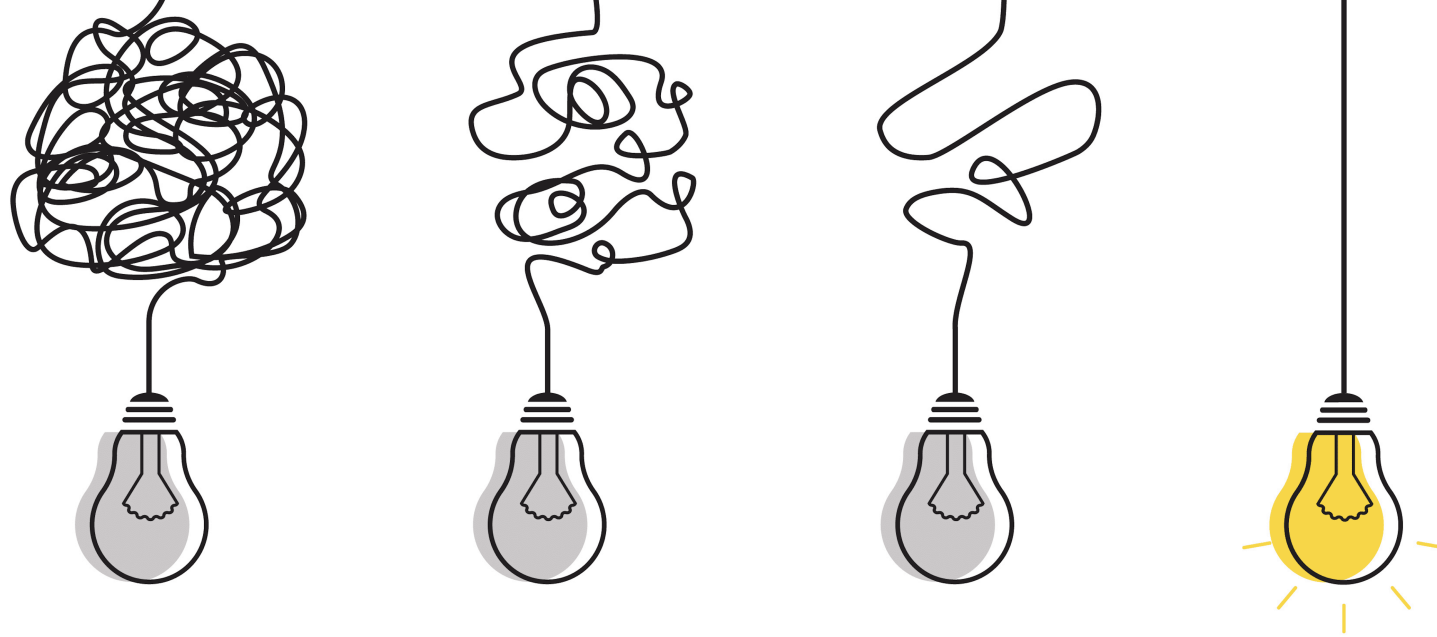
Table 8: Comparison between the best single model, Full Merging, Full Mixture and our Model-GLUE.

Model	ARC	WinoGrande	MMLU	GSM8K	MBPP	HumanEval	Average
Best Single Model	46.76%	64.33%	46.33%	62.40%	42.00%	31.10%	48.82%
Full Merging	55.12%	73.64%	50.13%	39.35%	21.80%	21.34%	43.56%
F-L-T Mixture	54.69%	73.32%	48.74%	35.18%	22.60%	21.34%	42.65%
Model-GLUE	51.62%	70.56%	51.85%	53.53%	47.20%	51.83%	54.43%

Models: 12 *Llama-2*-based LLMs fine-tuned towards different domains (Chatting, Mathematic reasoning, Coding ...)

Baselines

- ✓ **Full Merging:** progressive model merging without mixture (①②③)
- ✓ **F-L-T Mixture:** FFN-level mixture of models selected by Full merging



Q&A

Tianlong Chen, Assistant Professor

CS@UNC Chapel Hill

Web: <https://tianlong-chen.github.io/>

Mixture-of-Experts in the Era of LLMs: A New Odyssey

Yu Cheng

The Chinese University of Hong Kong

July 22, 2024

Outline

1. MoE Design

- Architecture, Auxiliary Loss, Routing

1. Building MoE from Dense LLMs

- Upcycling
- Sparse Splitting

2. MoE Beyond Efficiency

- Scaling Law, Fine-tuning MoE
- Other derivatives in this era

Outline

1. MoE Design

- Architecture, Auxiliary Loss, Routing

1. Building MoE from Dense LLMs

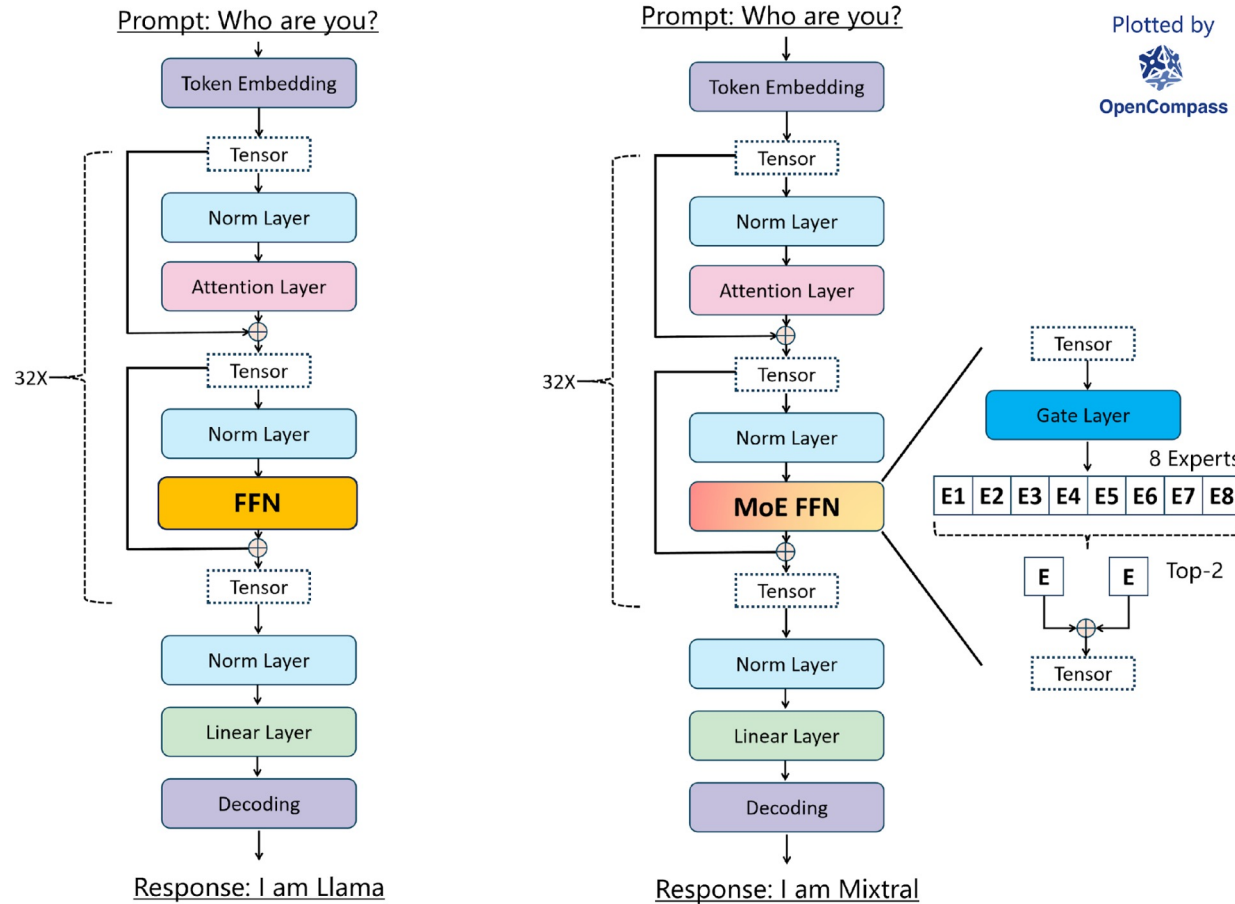
- Upcycling
- Sparse Splitting

2. MoE Beyond Efficiency

- Scaling Law, Fine-tuning MoE
- Other derivatives in this era

Recap: Mixture-of-Experts

■ Model Architectures



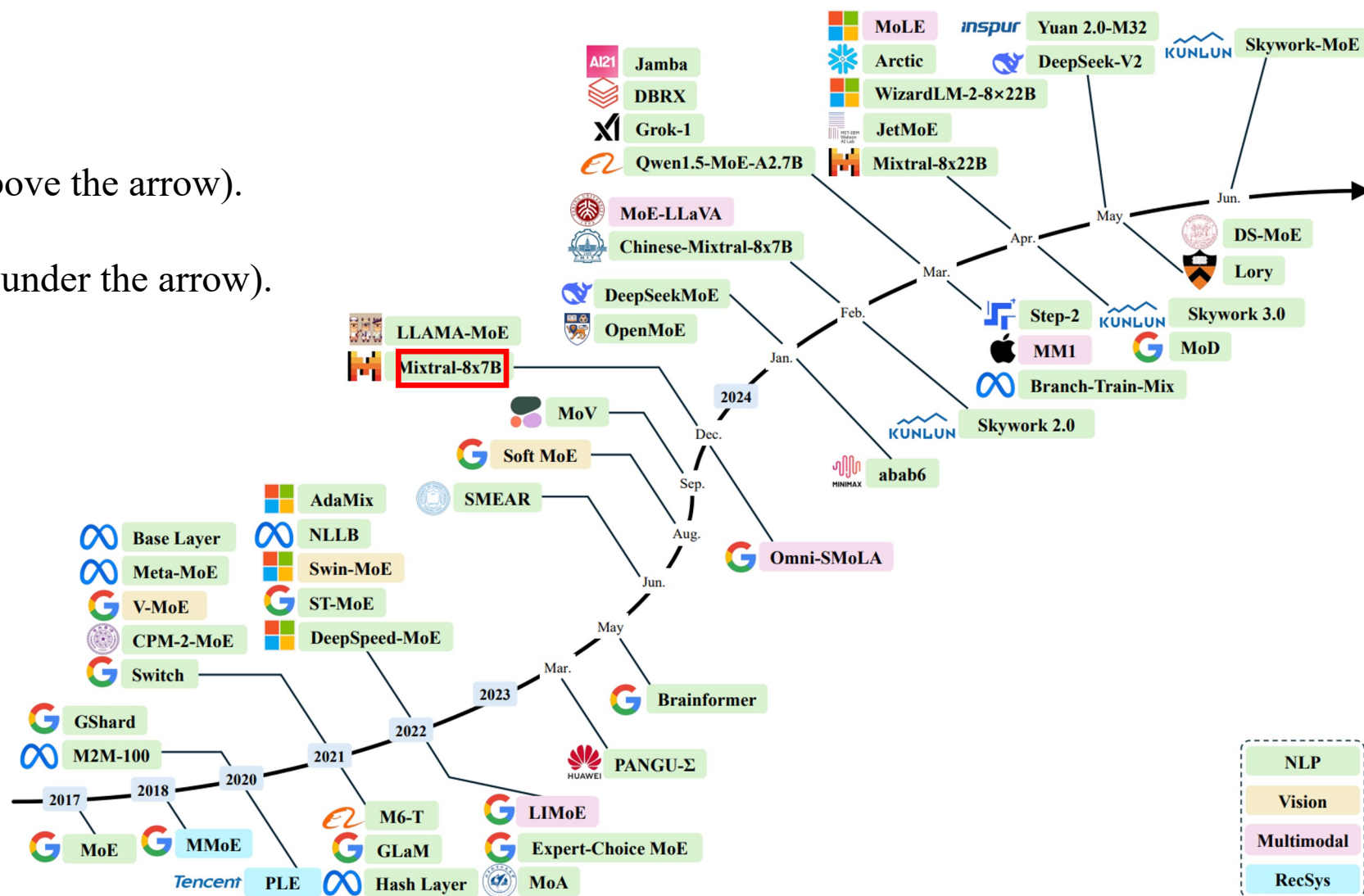
Key points:

- Activate different experts parameters for each input token.
- Sparse activation. Not all parameters are activated.

Recap: Mixture-of-Experts

Road Map

- Open-source (above the arrow).
- Private models (under the arrow).



What should we care when designing a MoE?

Network types	FFN, Attention
Fine-grained experts	64 experts/128 experts/...
Shared experts	Isolated experts
Activation Function	ReLU/GEGLU/SwiGLU
MoE frequency	Every two layer/Each layer/...
Training auxiliary loss	Auxiliary loss/Z-loss/...

Fine-grained and Shared Experts

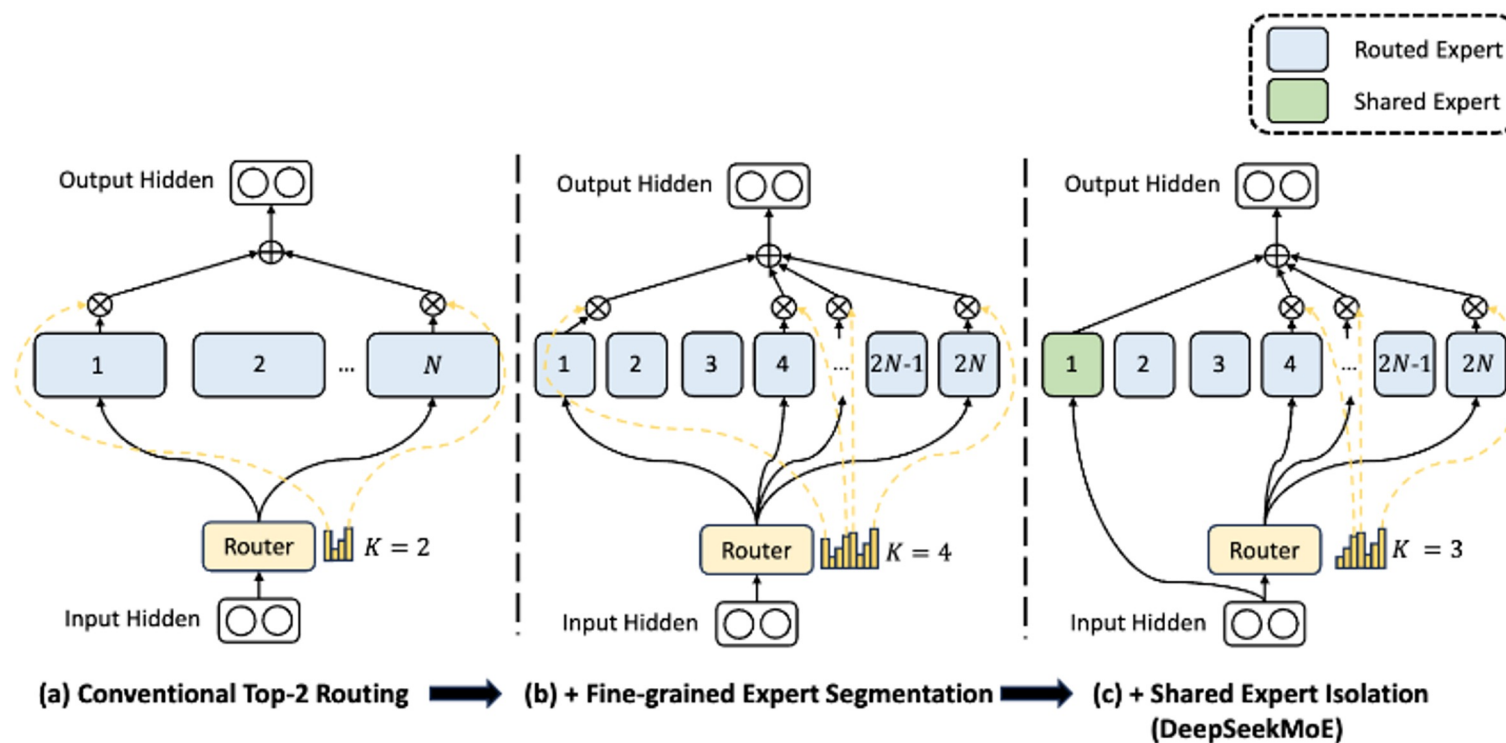


Figure 2 | Illustration of DeepSeekMoE. Subfigure (a) showcases an MoE layer with the conventional top-2 routing strategy. Subfigure (b) illustrates the fine-grained expert segmentation strategy. Subsequently, subfigure (c) demonstrates the integration of the shared expert isolation strategy, constituting the complete DeepSeekMoE architecture. It is noteworthy that across these three architectures, the number of expert parameters and computational costs remain constant.

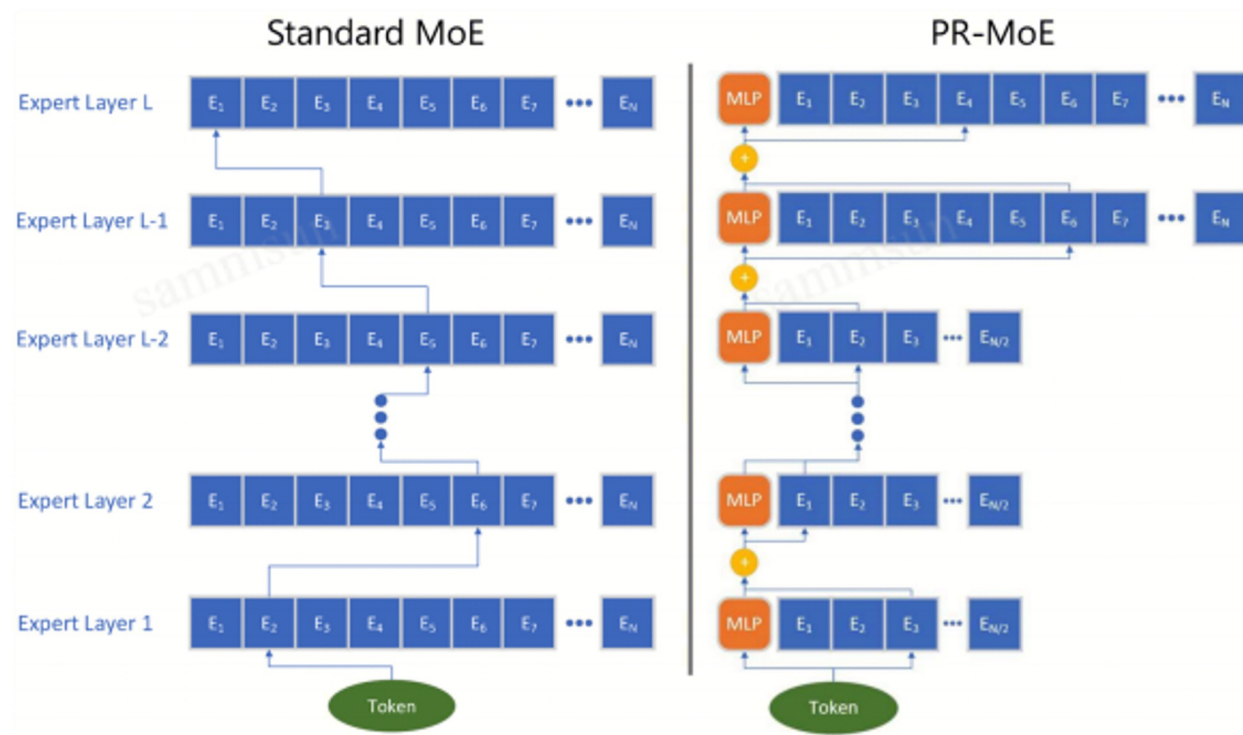
MoE Experts Design

Reference	Models	Expert Count (Activ./Total)	d_{model}	d_{ffn}	d_{expert}	#L	#H	d_{head}	Placement Frequency	Activation Function	Share Expert Count
GShard [86] (2020)	600B	2/2048	1024	8192	d_{ffn}	36	16	128	1/2	ReLU	0
	200B	2/2048	1024	8192	d_{ffn}	12	16	128	1/2	ReLU	0
	150B	2/512	1024	8192	d_{ffn}	36	16	128	1/2	ReLU	0
	37B	2/128	1024	8192	d_{ffn}	36	16	128	1/2	ReLU	0
Switch [49] (2021)	7B	1/128	768	2048	d_{ffn}	12	12	64	1/2	GEGLU	0
	26B	1/128	1024	2816	d_{ffn}	24	16	64	1/2	GEGLU	0
	395B	1/64	4096	10240	d_{ffn}	24	64	64	1/2	GEGLU	0
	1571B	1/2048	2080	6144	d_{ffn}	15	32	64	1	ReLU	0
GLaM [44] (2021)	0.1B/1.9B	2/64	768	3072	d_{ffn}	12	12	64	1/2	GEGLU	0
	1.7B/27B	2/64	2048	8192	d_{ffn}	24	16	128	1/2	GEGLU	0
	8B/143B	2/64	4096	16384	d_{ffn}	32	32	128	1/2	GEGLU	0
	64B/1.2T	2/64	8192	32768	d_{ffn}	64	128	128	1/2	GEGLU	0
DeepSpeed-MoE [121] (2022)	350M/13B	2/128	1024	$4d_{model}$	d_{ffn}	24	16	64	1/2	GeLU	0
	1.3B/52B	2/128	2048	$4d_{model}$	d_{ffn}	24	16	128	1/2	GeLU	0
	PR-350M/4B	2/32-2/64	1024	$4d_{model}$	d_{ffn}	24	16	64	1/2, 10L-32E, 2L-64E	GeLU	1
	PR-1.3B/31B	2/64-2/128	2048	$4d_{model}$	d_{ffn}	24	16	128	1/2, 10L-64E, 2L-128E	GeLU	1
ST-MoE [197] (2022)	0.8B/4.1B	2/32	1024	2816	d_{ffn}	27	16	64	1/4, add extra FFN	GEGLU	0
	32B/269B	2/64	5120	20480	d_{ffn}	27	64	128	1/4, add extra FFN	GEGLU	0
Mixtral [74] (2023)	13B/47B	2/8	4096	14336	d_{ffn}	32	32	128	1	SwiGLU	0
	39B/141B	2/8	6144	16384	d_{ffn}	56	48	128	1	SwiGLU	0
LLAMA-MoE [149] (2023)	3.0B/6.7B	2/16	4096	11008	688	32	32	128	1	SwiGLU	0
	3.5B/6.7B	4/16	4096	11008	688	32	32	128	1	SwiGLU	0
	3.5B/6.7B	2/8	4096	11008	1376	32	32	128	1	SwiGLU	0
DeepSeekMoE [30] (2024)	0.24B/1.89B	8/64	1280	-	$\frac{1}{4}d_{ffn}$	9	10	128	1	SwiGLU	1
	2.8B/16.4B	8/66	2048	10944	1408	28	16	128	1, except 1st layer	SwiGLU	2
	22B/145B	16/132	4096	-	$\frac{1}{8}d_{ffn}$	62	32	128	1, except 1st layer	SwiGLU	4
OpenMoE [172] (2024)	339M/650M	2/16	768	3072	d_{ffn}	12	12	64	1/4	SwiGLU	1
	2.6B/8.7B	2/32	2048	8192	d_{ffn}	24	24	128	1/6	SwiGLU	1
	6.8B/34B	2/32	3072	12288	d_{ffn}	32	24	128	1/4	SwiGLU	1
Qwen1.5-MoE [151] (2024)	2.7B/14.3B	8/64	2048	5632	1408	24	16	128	1	SwiGLU	4
DBRX [34] (2024)	36B/132B	4/16	6144	10752	d_{ffn}	40	48	128	1	SwiGLU	0
Jamba [94] (2024)	12B/52B	2/16	4096	14336	d_{ffn}	32	32	128	1/2, 1:7 Attention:Mamba	SwiGLU	0
Skywork-MoE [154] (2024)	22B/146B	2/16	4608	12288	d_{ffn}	52	36	128	1	SwiGLU	0
Yuan 2.0-M32 [166] (2024)	3.7B/40B	2/32	2048	8192	d_{ffn}	24	16	256	1	SwiGLU	0

Key points:

- Most recent models place MoE each layer.
- Some of recent models apply Shared experts.

Pyramid Design of Experts



- Utilizes more experts in the last few layers as compared to previous layers
- Positive results compared with the baseline MoE

Model (num. params)	LAMBADA	PIQA	BoolQ	RACE-h	TriviaQA	WebQs
350M+MoE-128 (13B)	62.70	74.59	60.46	35.60	16.58	5.17
350M+PR-MoE-32/64 (4B)	63.65	73.99	59.88	35.69	16.30	4.73
1.3B+MoE-128 (52B)	69.84	76.71	64.92	38.09	31.29	7.19
1.3B+PR-MoE-64/128 (31B)	70.60	77.75	67.16	38.09	28.86	7.73

Auxiliary Loss

Training with different auxiliary loss:

Reference	Auxiliary Loss	Coefficient
Shazeer et al.[135], V-MoE[128]	$L_{importance} + L_{load}$	$w_{importance} = 0.1, w_{load} = 0.1$
GShard[86], Switch-T[49], GLaM[44], Mixtral-8x7B[74], DBRX[34], Jamba[94], DeepSeekMoE[30], DeepSeek-V2[36], Skywork-MoE[154]	L_{aux}	$w_{aux} = 0.01$
ST-MoE[197], OpenMoE[172], MoA[182], JetMoE [139]	$L_{aux} + L_z$	$w_{aux} = 0.01, w_z = 0.001$
Mod-Squad[21], Moduleformer[140], DS-MoE[117]	L_{MI}	$w_{MI} = 0.001$

Importance loss: encourages all experts to have equal importance

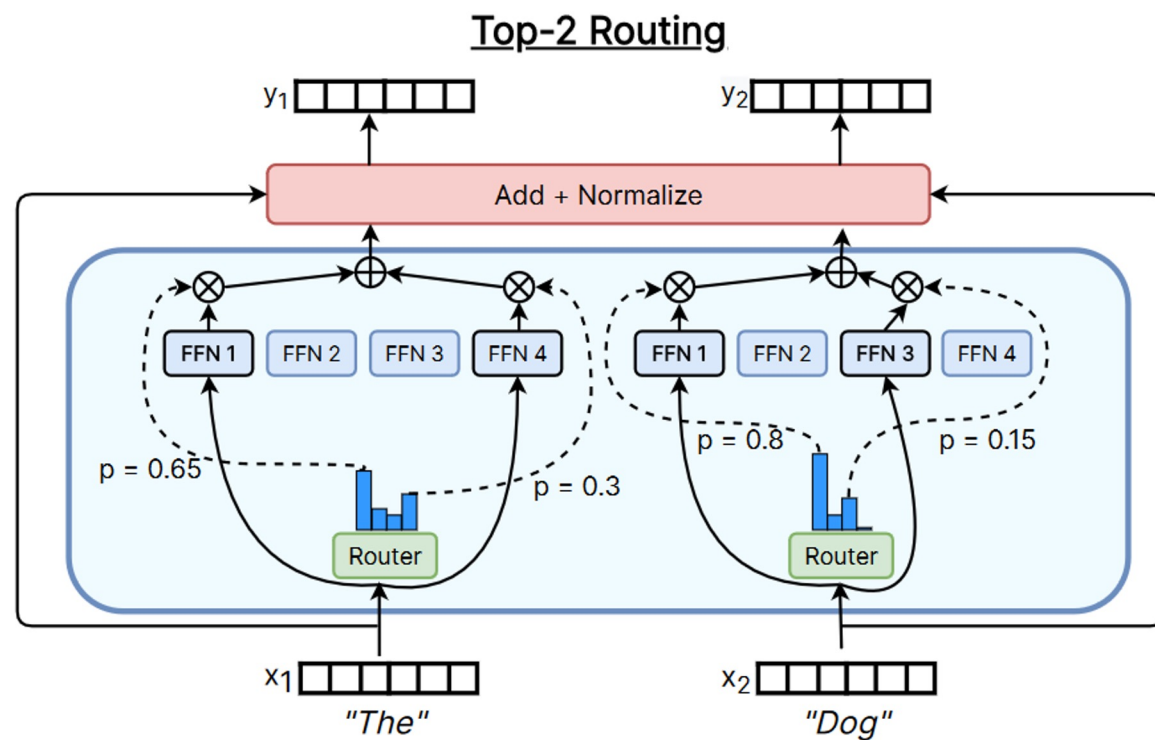
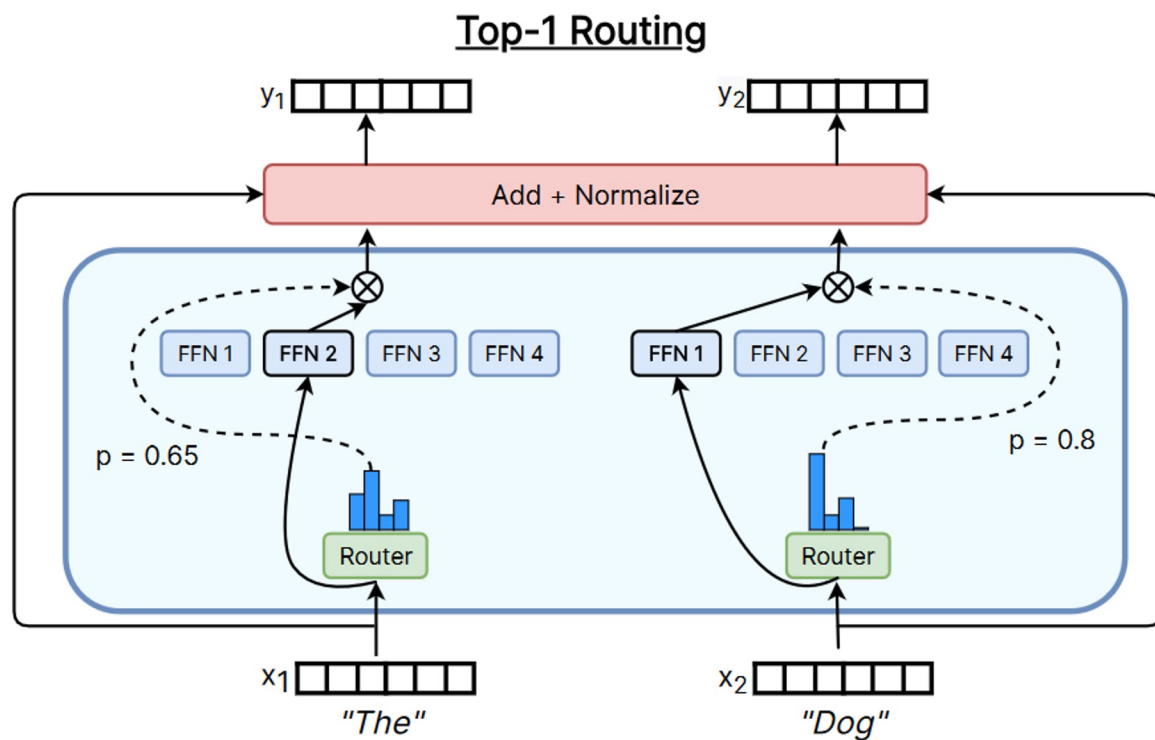
Load loss: ensure balanced loads

Auxiliary loss: mitigating load balance losses

Z-loss: improving training stability by penalizing large logits

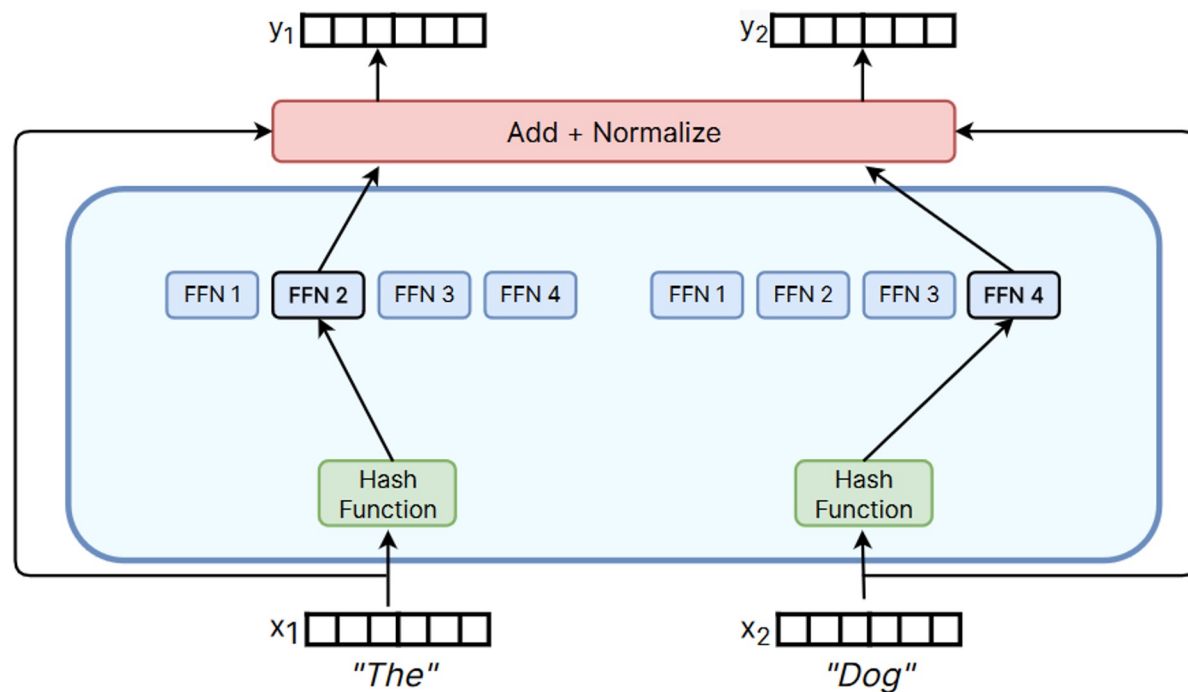
MI-loss: mutual information (MI) between experts and tasks to build task-expert alignment

Routing Algorithms

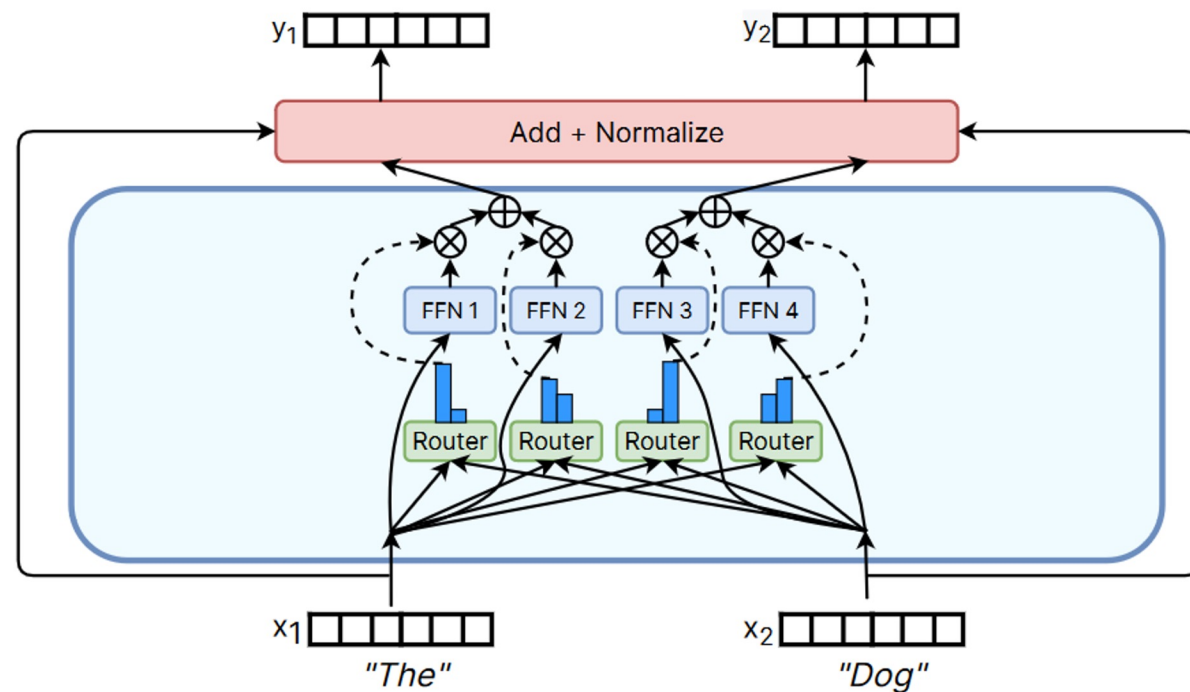


Routing Algorithms

Hash Routing

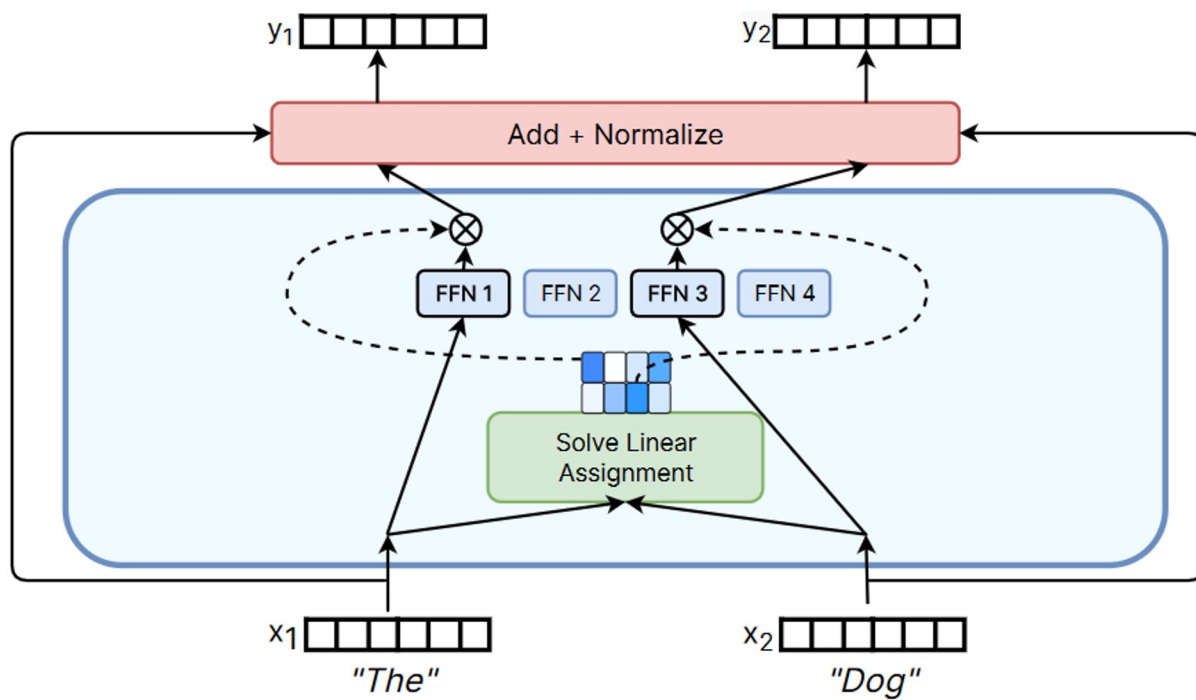


Expert Chooses Tokens

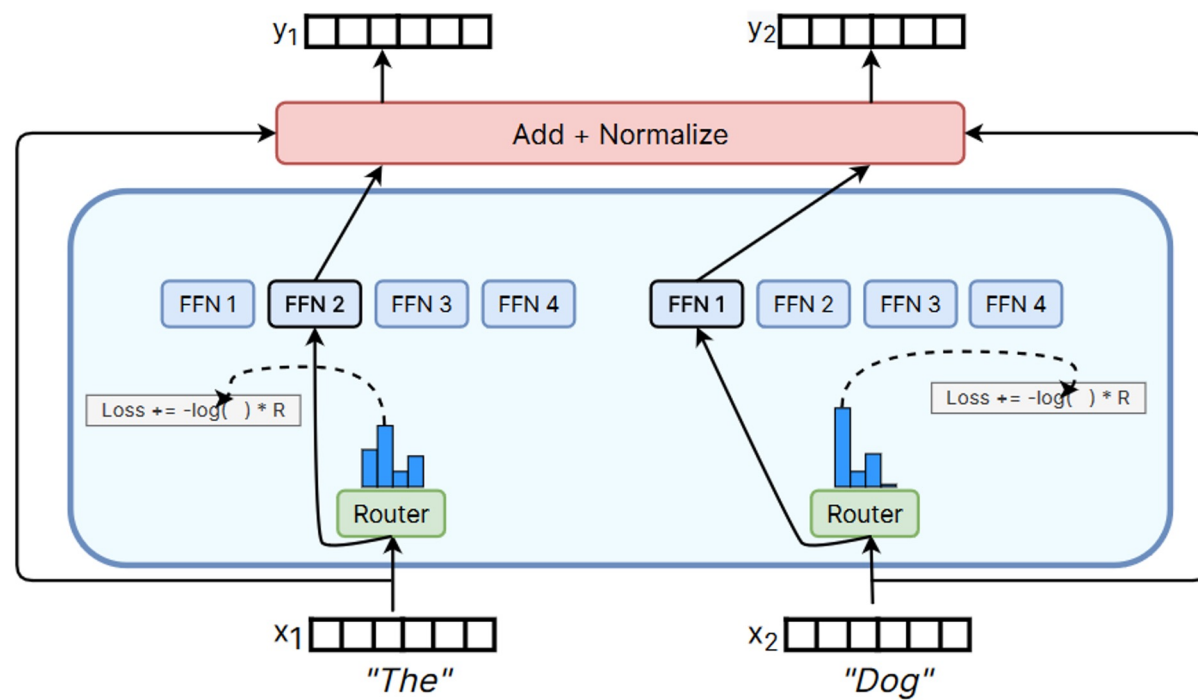


Routing Algorithms

BASE Routing



Reinforcement Learning

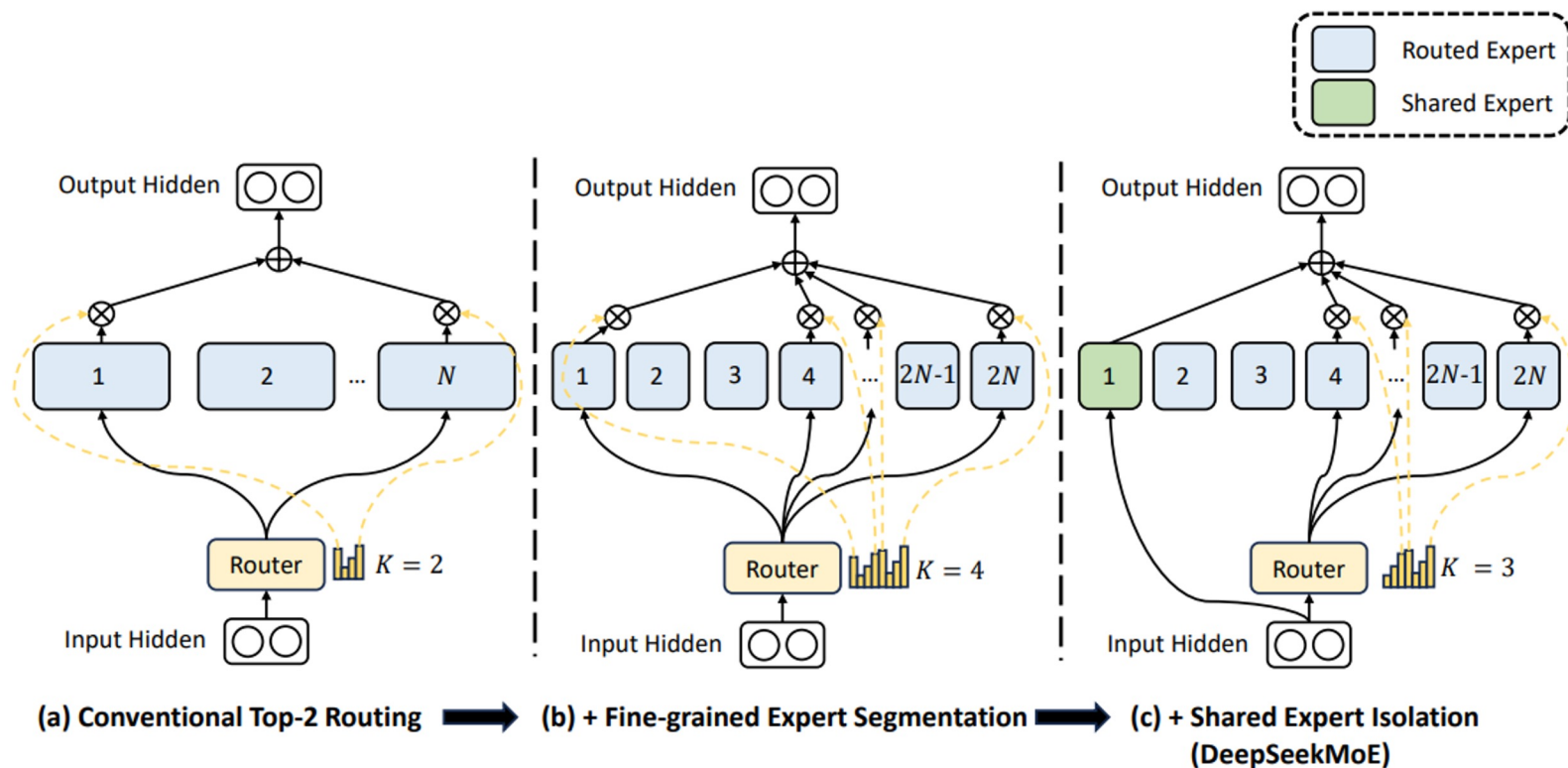


Training MoE - Deepseek

Example 1: Deepseek-MoE

Deepseek-MoE 16B, total 16.4B parameters, 2.8B activate parameters.

Each MoE layer consists of 2 shared experts and 64 routed experts (select 6 experts).

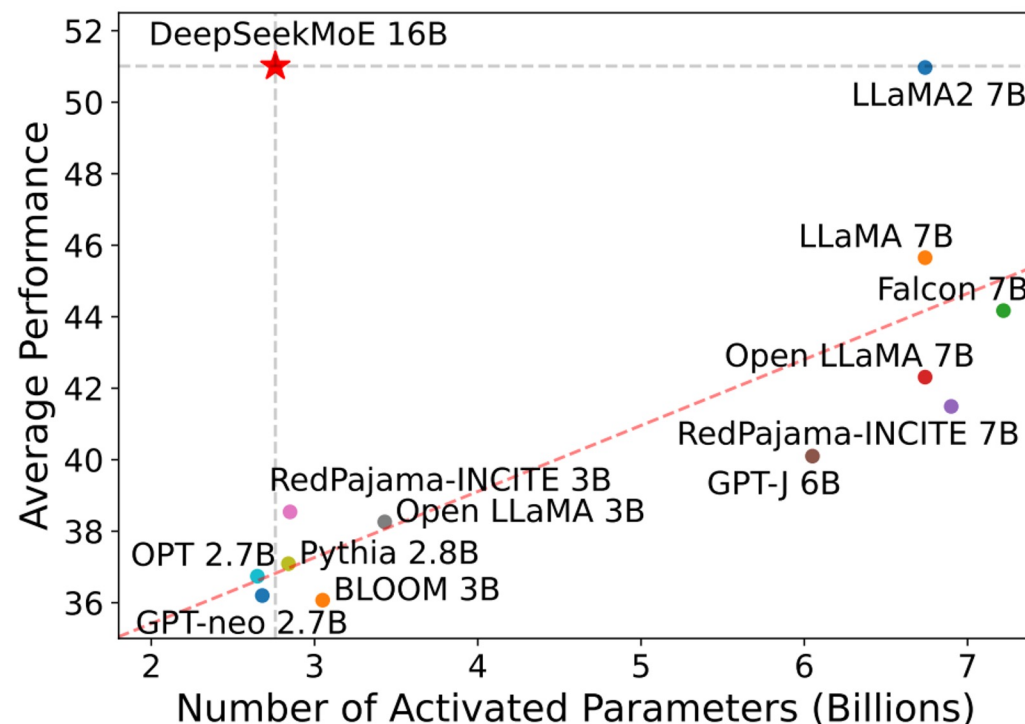


Key points:

- Fine-grained experts
- Shared experts

Training MoE - Deepseek

Example 1: Deepseek-MoE

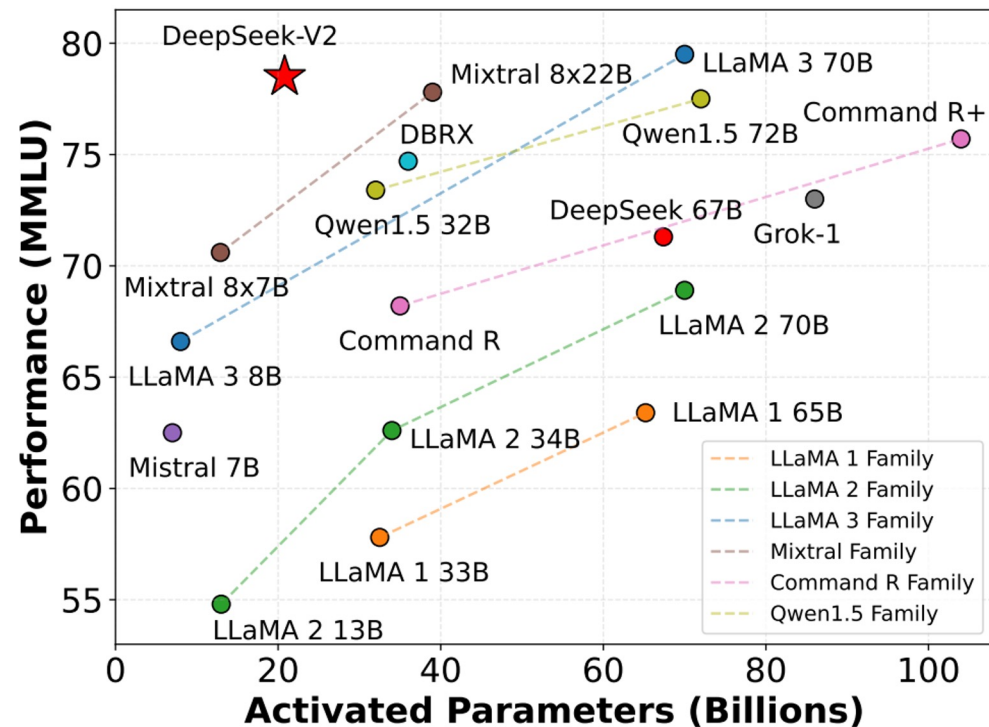


Metric	# Shot	DeepSeek 7B (Dense)	DeepSeekMoE 16B
# Total Params	N/A	6.9B	16.4B
# Activated Params	N/A	6.9B	2.8B
FLOPs per 4K Tokens	N/A	183.5T	74.4T
# Training Tokens	N/A	2T	2T
Pile (BPB)	N/A	0.75	0.74
HellaSwag (Acc.)	0-shot	75.4	77.1
PIQA (Acc.)	0-shot	79.2	80.2
ARC-easy (Acc.)	0-shot	67.9	68.1
ARC-challenge (Acc.)	0-shot	48.1	49.8
RACE-middle (Acc.)	5-shot	63.2	61.9
RACE-high (Acc.)	5-shot	46.5	46.4
DROP (EM)	1-shot	34.9	32.9
GSM8K (EM)	8-shot	17.4	18.8
MATH (EM)	4-shot	3.3	4.3
HumanEval (Pass@1)	0-shot	26.2	26.8
MBPP (Pass@1)	3-shot	39.0	39.2
TriviaQA (EM)	5-shot	59.7	64.8
NaturalQuestions (EM)	5-shot	22.2	25.5
MMLU (Acc.)	5-shot	48.2	45.0
WinoGrande (Acc.)	0-shot	70.5	70.2
CLUEWSC (EM)	5-shot	73.1	72.1
CEval (Acc.)	5-shot	45.0	40.6
CMMLU (Acc.)	5-shot	47.2	42.5
CHID (Acc.)	0-shot	89.3	89.4

Training MoE - Deepseek

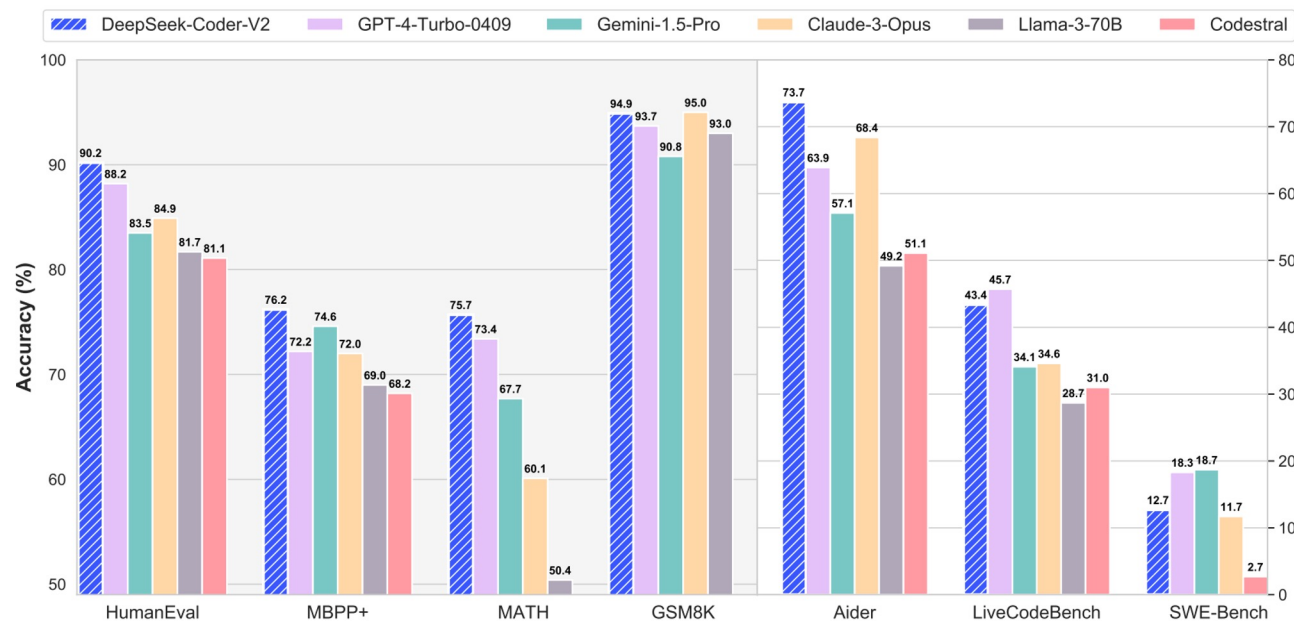
Deepseek-V2

236B total parameters, 21B are activated.
2 shared experts and 160 routed experts (6 select).



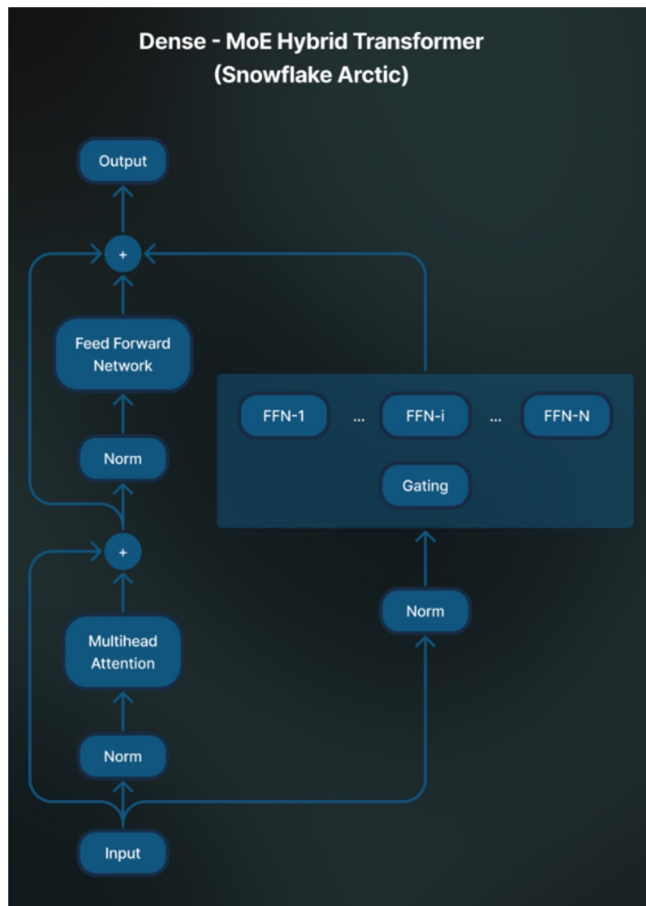
Deepseek-Coder-V2

Continue pretraining from an intermediate checkpoint of Deepseek-V2 (4.2T) and further train 6T. Total 10.2T tokens.



Training MoE - Arctic

Example 2: Arctic (Dense and Sparse)



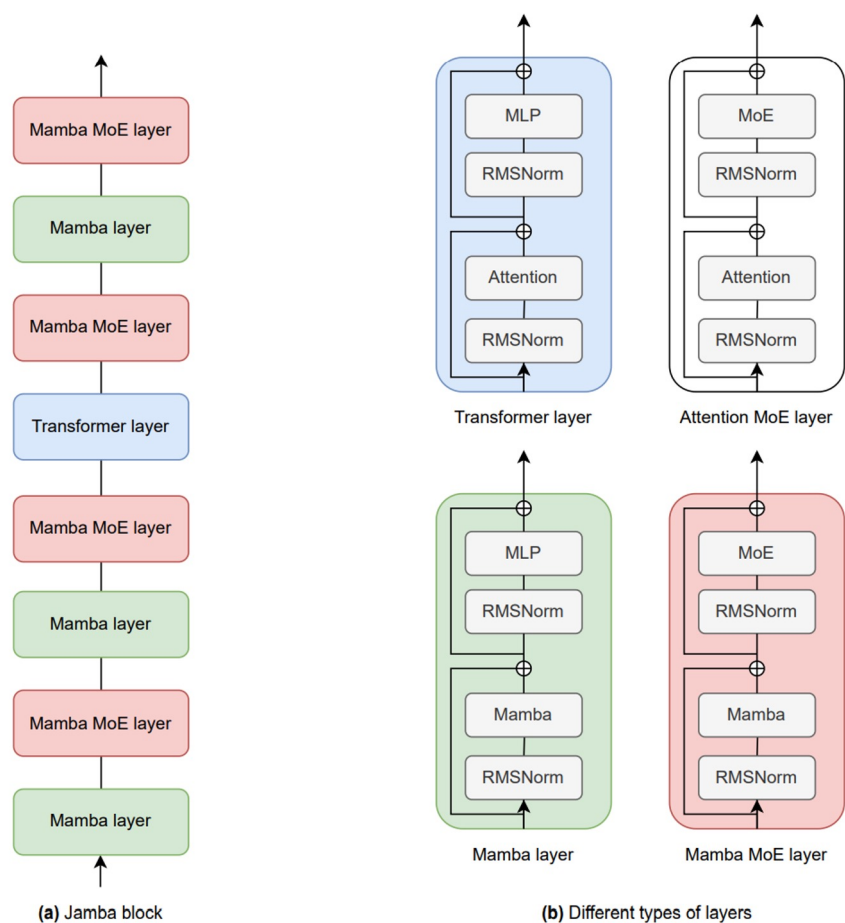
Arctic uses a unique Dense-MoE Hybrid transformer architecture.

- It combines a 10B dense transformer model with a residual 128×3.66B MoE MLP.
- 480B total and 17B active parameters chosen using a top-2 gating.

	Snowflake Arctic	DBRX	Llama 3 8B	Llama 2 70B	Llama 3 70B	Mixtral 8x7B	Mixtral 8x22B
Active Parameters	17B	36B	8B	70B	70B	13B	44B
ENTERPRISE							
SQL Generation (Spider)	79.0	76.3	69.9	62.8	80.2	71.3	79.2
Coding (HumanEval+, MBPP+)	64.3	61.0	59.2	33.7	71.9	48.1	69.9
Instruction Following (IFEval)	57.4	54.8	42.7	-	43.6	52.2	61.5
ACADEMIC							
Math (GSM8K)	74.2	73.5	75.4	52.6	91.4	63.2	84.2
Common Sense (Avg of 11 metrics)	73.1	74.8	68.5	72.1	72.6	74.1	75.6
World Knowledge (MMLU)	67.3	73.3	65.7	68.6	79.8	70.4	77.5

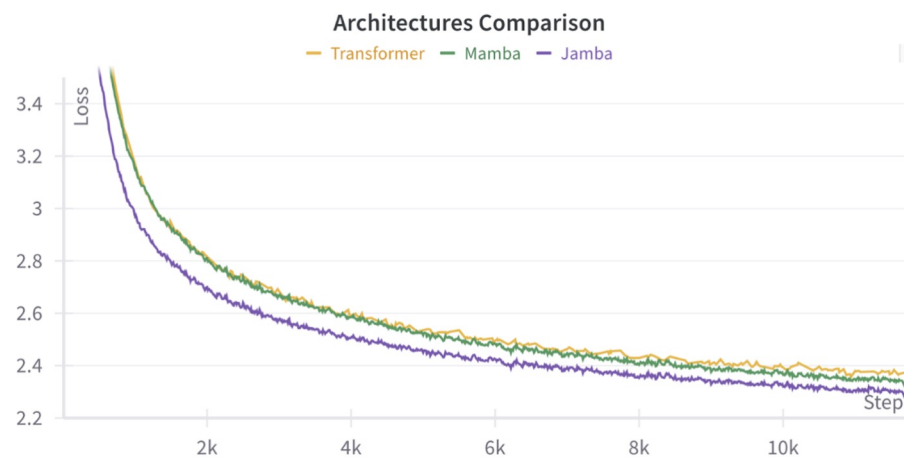
Training MoE - Jamba

Example 3: Jamba (Hybrid architecture)



Jamba is a hybrid decoder architecture that mixes Transformer layers with Mamba layers, in addition to a mixture-of-experts (MoE) module.

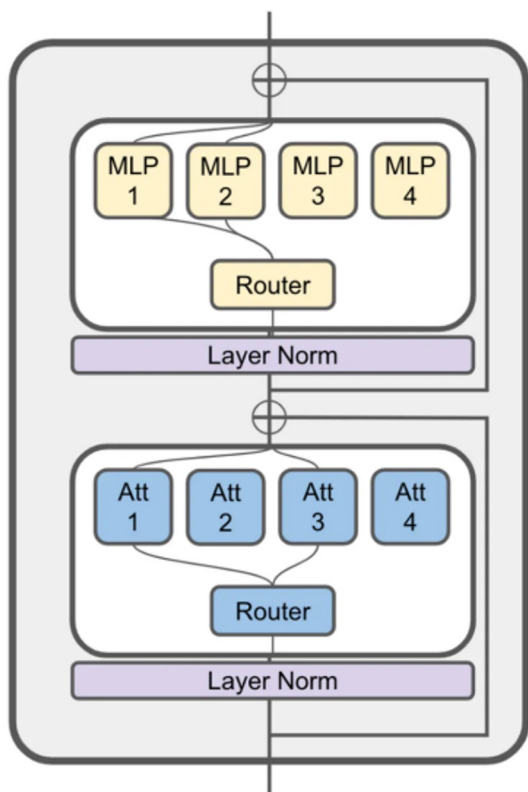
	Available params	Active params
LLAMA-2	6.7B	6.7B
Mistral	7.2B	7.2B
Mixtral	46.7B	12.9B
Jamba	52B	12B



Jamba: A Hybrid Transformer-Mamba Language Model

Training MoE - JetMoE

Example 4: JetMoE



Both attention and feedforward layers are sparsely activated, allowing JetMoE-8B to have 8B parameters while only activating 2B for each input token.

	LLaMA2	DeepseekMoE	Gemma	JetMoE
# Total Params	7B	16B	2B	8B
# Activate Params	7B	2.8B	2B	2.2B
# Training tokens	2T	2T	2T	1.25T
ARC-challenge	53.1	53.2	48.4	48.7
Hellaswag	78.6	79.8	71.8	80.5
MMLU	46.9	46.3	41.8	49.2
TruthfulQA	38.8	36.1	33.1	41.7
WinoGrande	74.0	73.7	66.3	70.2
GSM8k	14.5	17.3	16.9	27.8
OpenLLM Leaderboard Avg.	51.0	51.1	46.4	53.0
MBPP (Pass@1)	20.8	34.0	28.0	34.2
HumanEval (Pass@1)	12.8	25.0	24.4	14.6
All Avg.	45.5	47.3	43.2	47.6

Outline

1. MoE Design

- Architecture, Auxiliary Loss, Routing

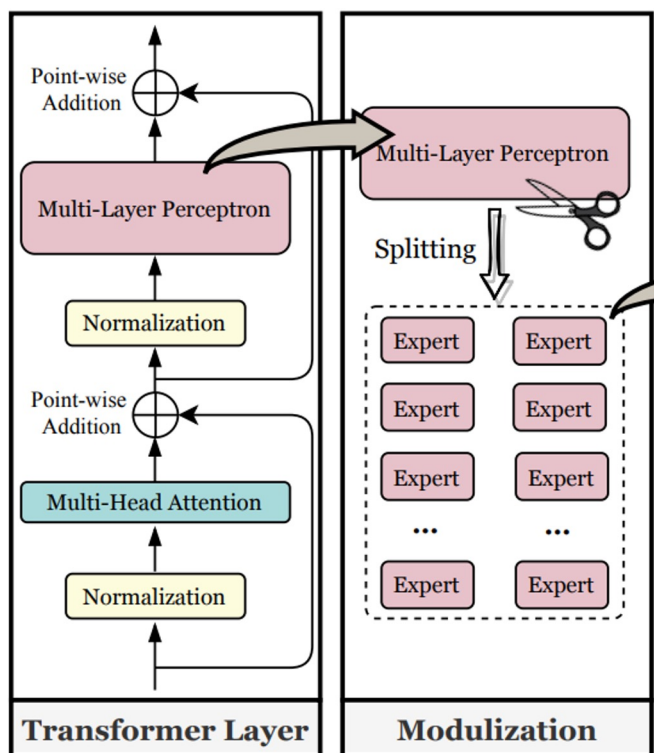
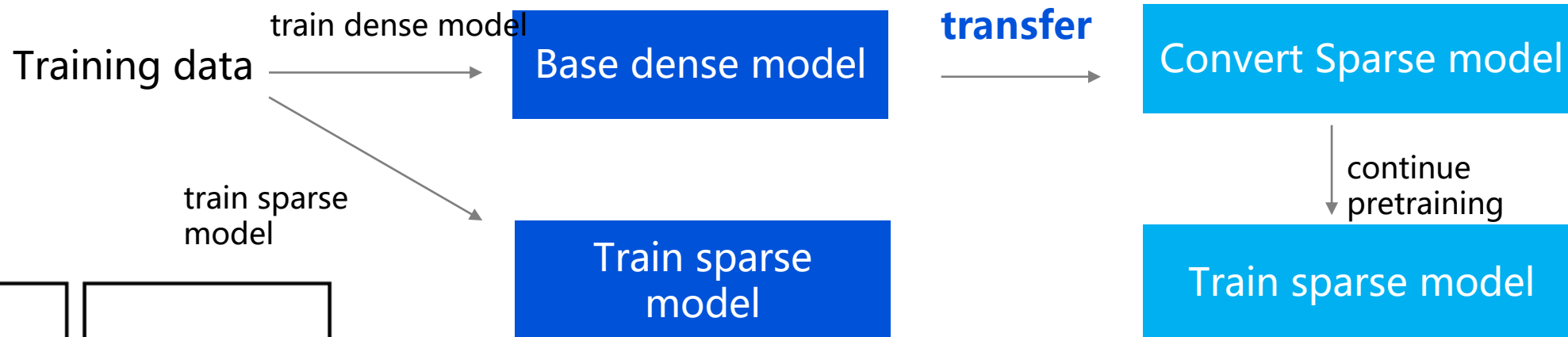
1. Building MoE from Dense LLMs

- Upcycling
- Sparse Splitting

2. MoE Beyond Efficiency

- Scaling Law, Fine-tuning MoE
- Other derivatives in this era

Building MoE LLMs



Sparse transformation

copy FFNs to form experts

parameters increased

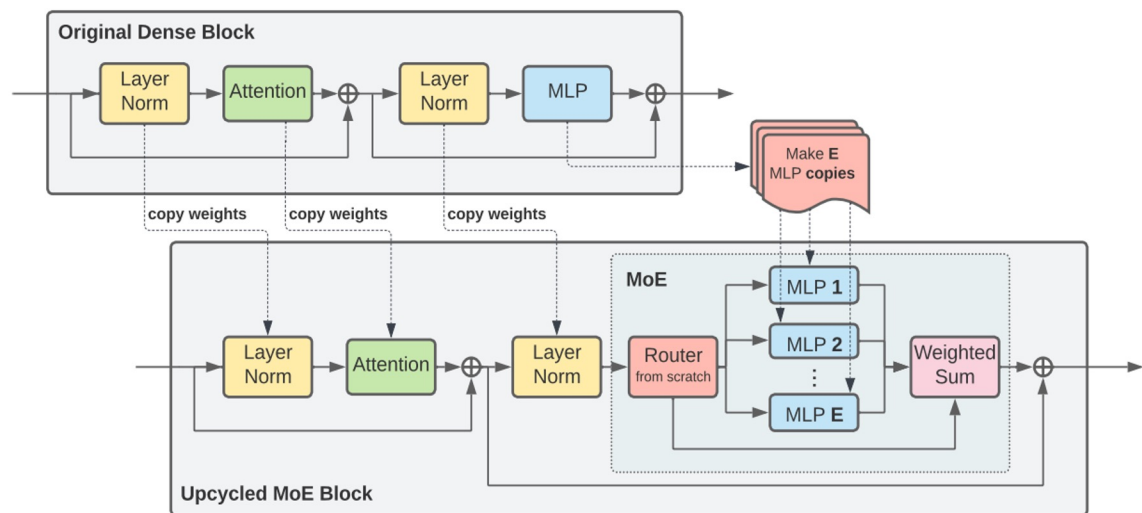
split FFNs to form experts

parameters unchanged

Building MoE from Dense LLMs

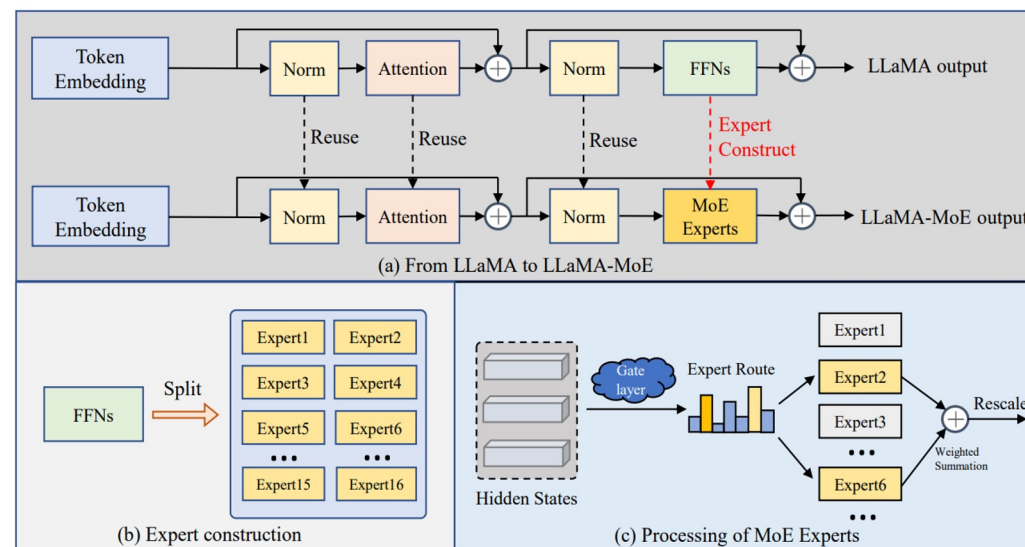
Two routes to build MoE from Dense models:

- **Sparse Upcycling**



Copying the FFNs to form experts

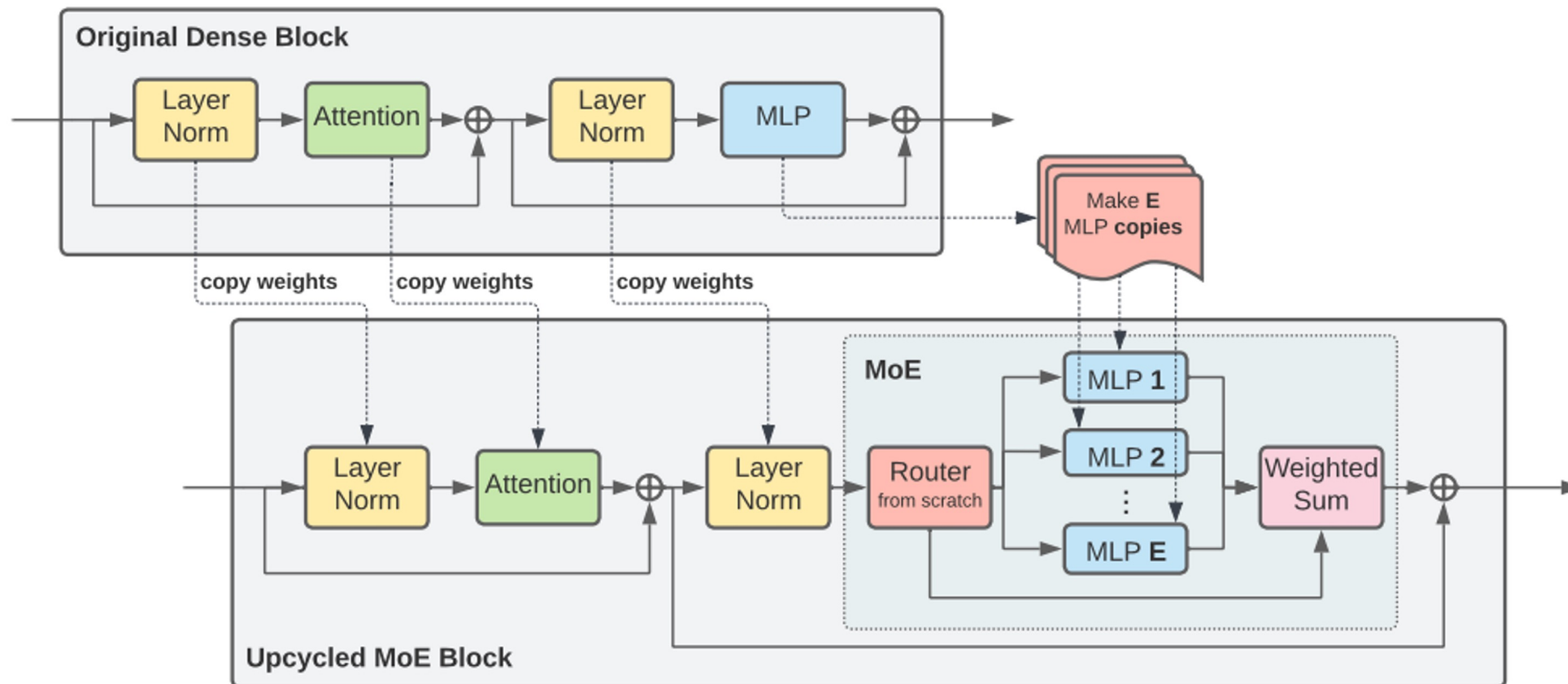
- **Sparse Splitting (MoEfication)**



Splitting the FFNs to form experts

Sparse Upcycling

Sparse upcycling solution



copying the MLP layers:

Upcycled T5-Large and T5-Base models outperform their dense counterparts by 1.5-2 absolute points on SuperGLUE using 46% and 55% extra training, respectively.

Sparse Upcycling - Mixtral MoE

Upcycling from Dense to MoE?

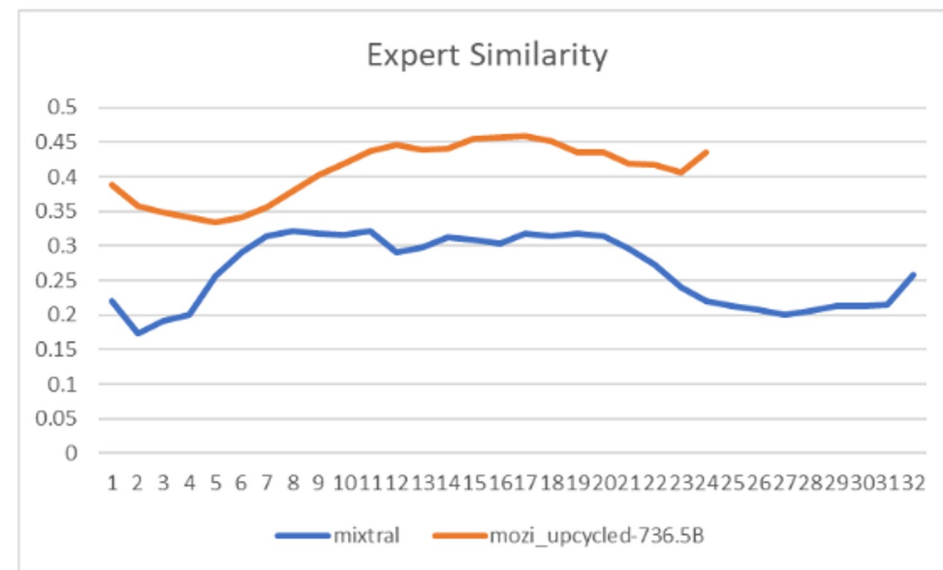
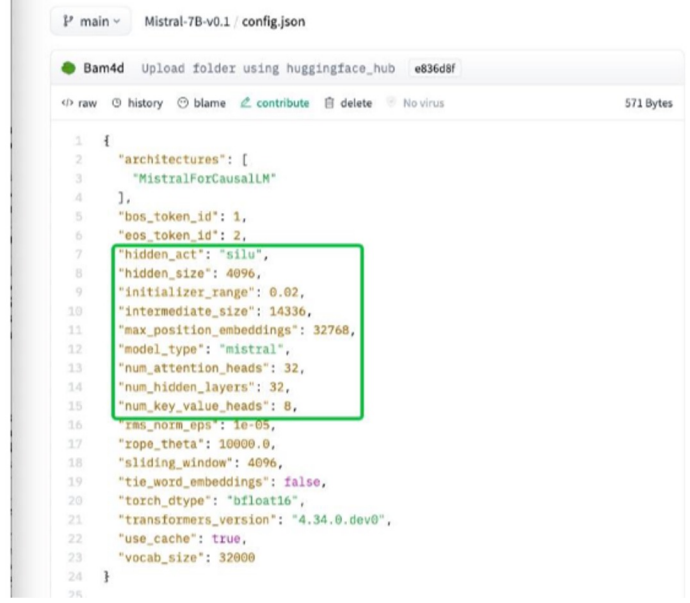
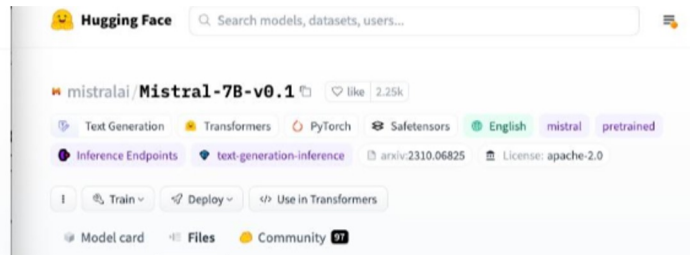
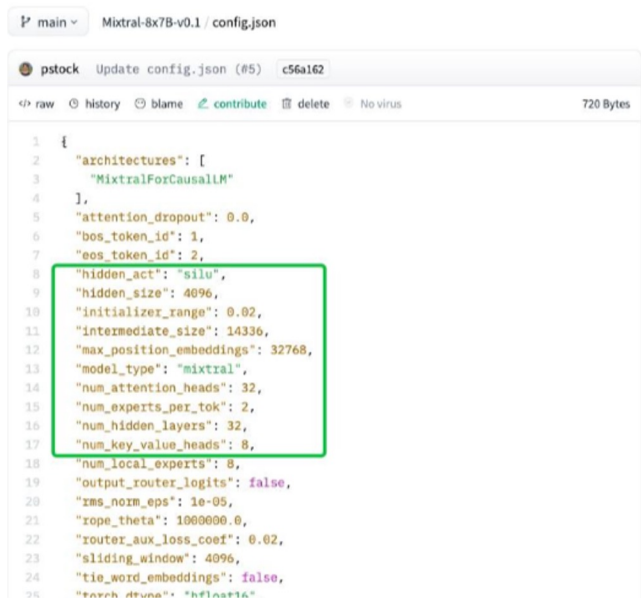
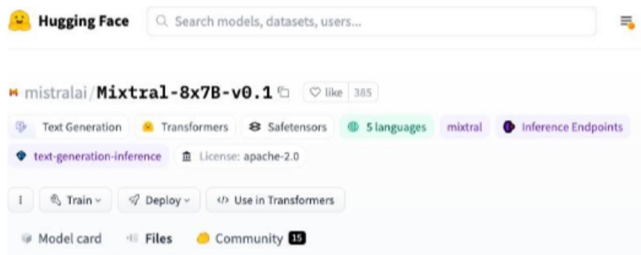
Example 1: Mixtral 8×22B(7B) (April, 2024)

Total 141B parameters, 39B activate parameters, (8 experts and 2 experts are selected)

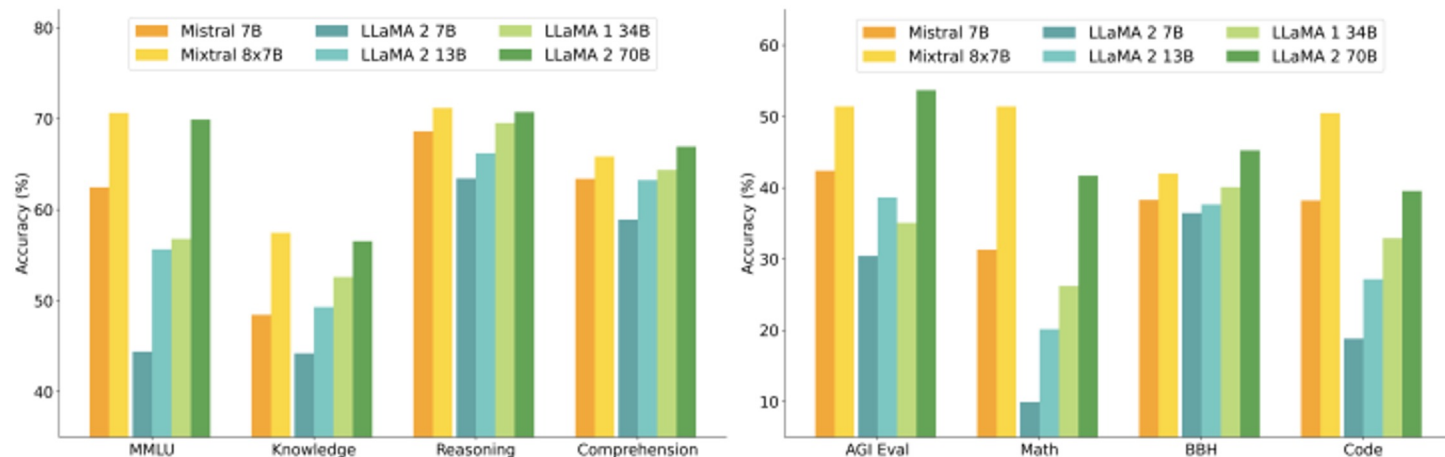
Model	Active parameters	Common sense and reasoning					Knowledge	
		MMLU	HellaS	WinoG	Arc C (5)	Arc C (25)	TriQA	NaturalQS
LLaMA 2 70B	70B	69.9%	87.1%	83.2%	86.0%	85.1%	77.57%	35.5%
CC-BY-NC license	Command R	35B	68.2%	87.0%	81.5%	-	66.5%	-
	Command R+	104B	75.7%	88.6%	85.4%	-	71.0%	-
Mistral 7B	7B	62.47%	83.1%	78.0%	77.2%	78.1%	68.8%	28.1%
Mixtral 8x7B	12.9B	70.63%	86.6%	81.2%	85.8%	85.9%	78.4%	36.5%
Mixtral 8x22B	39B	77.75%	88.5%	84.7%	91.3%	91.3%	82.2%	40.1%

Sparse Upcycling - Mixtral MoE

Upcycling from Dense to MoE?



Mixtral 8x7B



Model	Active Params	MMLU	HellaS	WinoG	PIQA	Arc-e	Arc-c	NQ	TriQA	HumanE	MBPP	Math	GSM8K
LLaMA 2 7B	7B	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	17.5%	56.6%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	13B	55.6%	80.7%	72.9%	80.8%	75.2%	48.8%	16.7%	64.0%	18.9%	35.4%	6.0%	34.3%
LLaMA 1 33B	33B	56.8%	83.7%	76.2%	82.2%	79.6%	54.4%	24.1%	68.5%	25.0%	40.9%	8.4%	44.1%
LLaMA 2 70B	70B	69.9%	85.4%	80.4%	82.6%	79.9%	56.5%	25.4%	73.0%	29.3%	49.8%	13.8%	69.6%
Mistral 7B	7B	62.5%	81.0%	74.2%	82.2%	80.5%	54.9%	23.2%	62.5%	26.2%	50.2%	12.7%	50.0%
Mixtral 8x7B	13B	70.6%	84.4%	77.2%	83.6%	83.1%	59.7%	30.6%	71.5%	40.2%	60.7%	28.4%	74.4%

Table 2: Comparison of Mixtral with Llama. Mixtral outperforms or matches Llama 2 70B performance on almost all popular benchmarks while using 5x fewer active parameters during inference.

Mixtral 8x7B

Model ▲	★ Arena Elo rating ▲	📊 MT-bench (score) ▲	License ▲
GPT-4-Turbo	1243	9.32	Proprietary
GPT-4-0314	1192	8.96	Proprietary
GPT-4-0613	1158	9.18	Proprietary
Claude-1	1149	7.9	Proprietary
Claude-2.0	1131	8.06	Proprietary
Mixtral-8x7b-Instruct-v0.1	1121	8.3	Apache 2.0
Claude-2.1	1117	8.18	Proprietary
GPT-3.5-Turbo-0613	1117	8.39	Proprietary
Gemini Pro	1111		Proprietary
Claude-Instant-1	1110	7.85	Proprietary
Tulu-2-DPO-70B	1110	7.89	AI2 ImpACT Low-risk
Yi-34B-Chat	1110		Yi License
GPT-3.5-Turbo-0314	1105	7.94	Proprietary
Llama-2-70b-chat	1077	6.86	Llama 2 Community

Figure 6: LMSys Leaderboard. (Screenshot from Dec 22, 2023) Mixtral 8x7B Instruct v0.1 achieves an Arena Elo rating of 1121 outperforming Claude-2.1 (1117), all versions of GPT-3.5-Turbo (1117 best), Gemini Pro (1111), and Llama-2-70b-chat (1077). Mixtral is currently the best open-weights model by a large margin.

Sparse Upcycling - Skywork-MoE

Example 2: Skywork-MoE (June, 2024)

Total 146B parameters, 22B activate parameters, (16 experts and 2 experts are selected)

Initialize from Skywork-13B

	#AP	#TP	CEVAL	CMMLU	MMLU	GSM8K	MATH	HumanEval
Deepseek-67B	67	67	66.1	70.8	71.3	63.4	18.7	42.7
Qwen1.5-72B	72	72	84.1	83.5	77.5	79.5	34.1	41.5
Llama2-70B	70	70	-	-	68.9	56.8	13.6	29.9
Llama3-70B	70	70	-	-	78.8	82.7	36.7	39.0
Mixtral 8*7B	13	47	-	-	70.6	58.4	28.4	40.2
Mixtral 8*22B	39	141	-	-	77.8	78.6	41.8	45.1
Grok-1	86	314	-	-	73.0	62.9	23.9	63.2
DBRX-Instruct	36	132	-	-	73.7	66.9	-	70.1
Deepseek-V2	21	236	81.7	84.0	78.5	79.2	43.6	48.8
Skywork-13B	13	13	62.1	62.4	62.7	60.2	8.4	18.9
Skywork-MoE	22	146	82.2	79.5	77.4	76.1	31.9	43.9

Sparse Upcycling - Qwen-MoE

Example 3: Qwen1.5-MoE-A2.7B (Mar, 2024)

Upcycled from Qwen-1.8B, 14.3B parameters in total and 2.7B activated parameters.

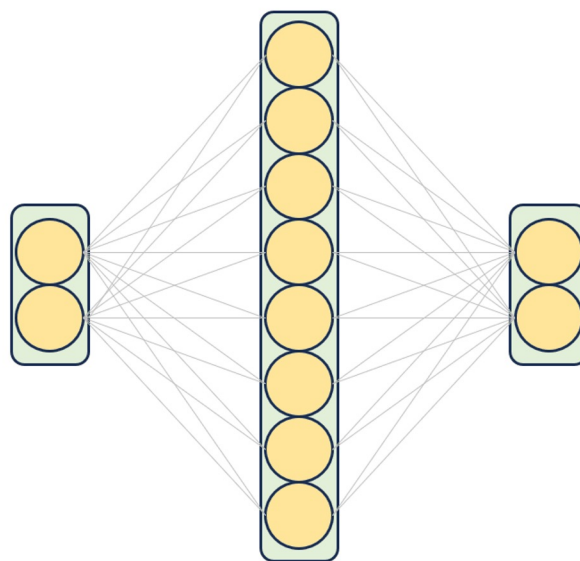
- Fine-grained experts (total 64 experts)
- use shared (4 experts) and routing experts (60 experts, choose 4)

Model	MMLU	GSM8K	HumanEval	Multilingual	MT-Bench
Mistral-7B	64.1	47.5	27.4	40.0	7.60
Gemma-7B	64.6	50.9	32.3	-	-
Qwen1.5-7B	61.0	62.5	36.0	45.2	7.60
DeepSeekMoE 16B	45.0	18.8	26.8	-	6.93
Qwen1.5-MoE-A2.7B	62.5	61.5	34.2	40.8	7.17

A remarkable reduction of 75%
in training

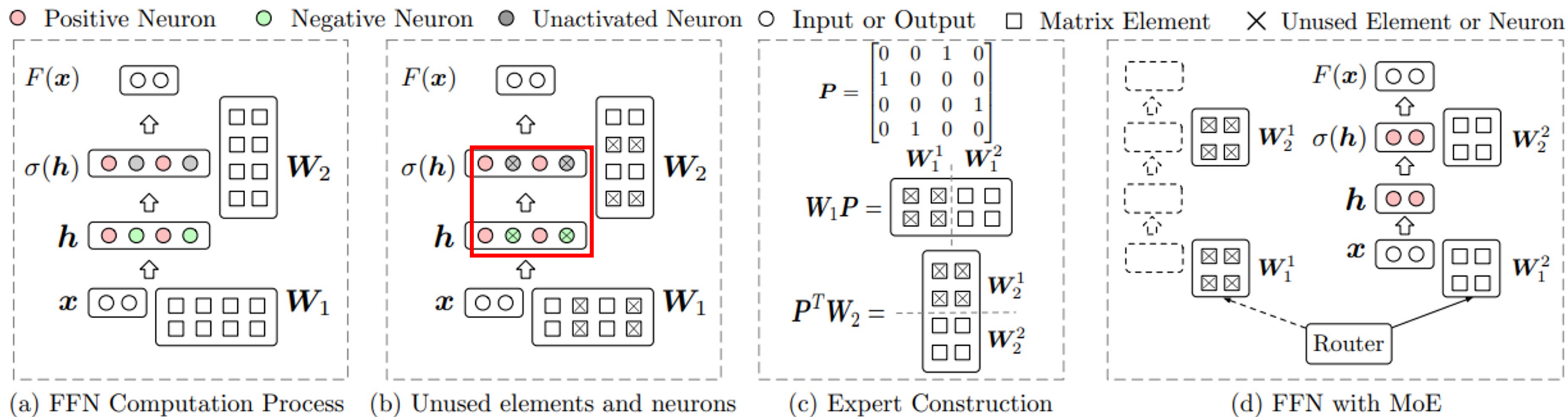
Sparse Splitting

Original FFN



Sparse Splitting - MoEfication

One solution for sparse splitting - MoEfication

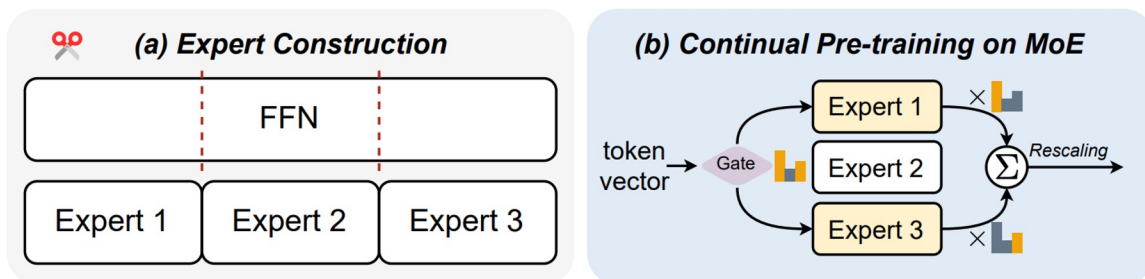


Splitting the FFN layers based on the activation diversity of different neurons.

Sparse Splitting - LLaMA-MoE

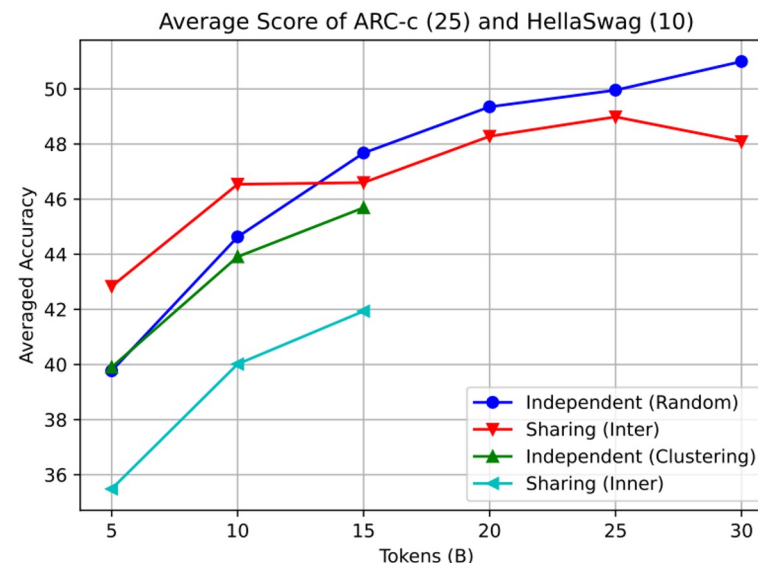
Sparsifying from Dense to MoE

Example 1: LLaMA-MoE (Dec, 2023)



Explore different FFN splitting strategies:

- Neuro-Independent
 1. Random splitting the FFNs
 2. Clustering with n centroids
- Neuro-Sharing
 1. Obtain n importance vectors
 2. Set aside the neuros shared by most experts and then obtain n importance vectors



Random splitting obtains the best.

Sparse Splitting - LLama-MoE

Sparsifying from Dense to MoE

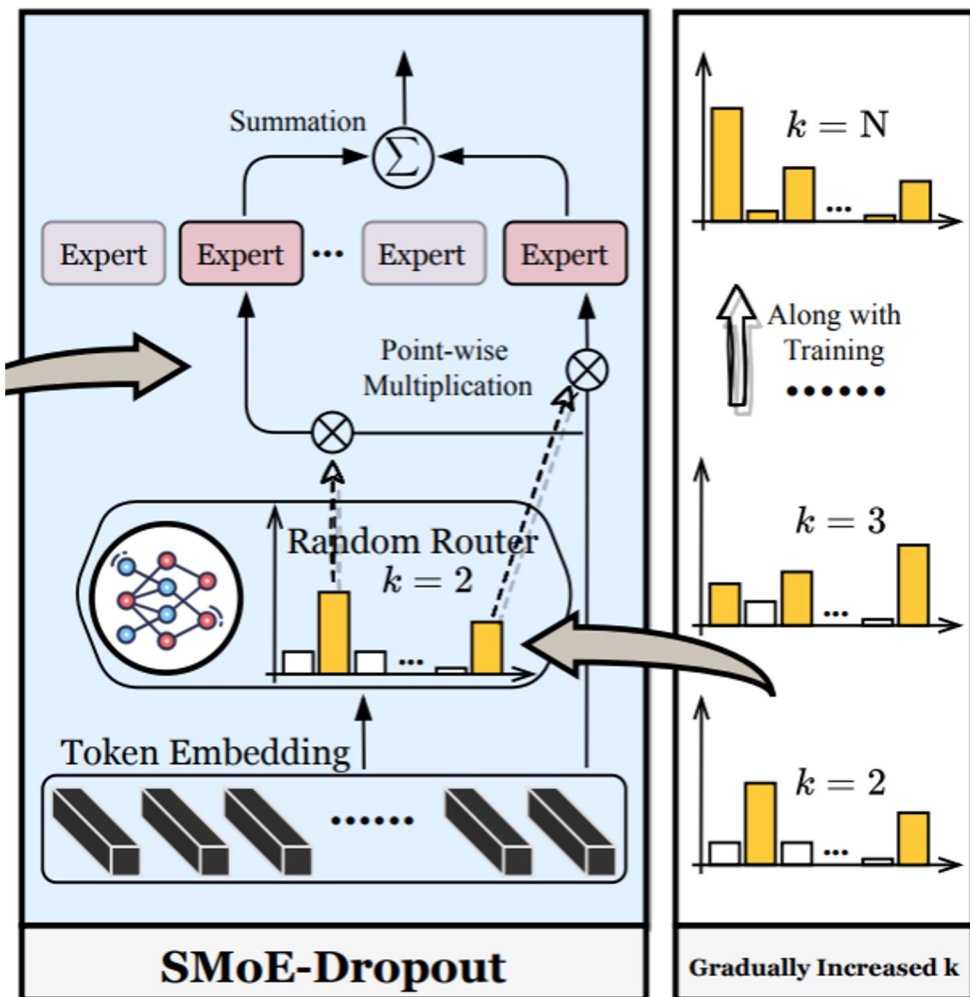
Model	Commonsense & Reading Comprehension					
	SciQ	PIQA	WinoGrande	ARC-E	ARC-C (25)	HellaSwag (10)
OPT-2.7B	78.9	74.8	60.8	54.4	34.0	61.4
Pythia-2.8B	83.2	73.6	59.6	58.8	36.7	60.7
INCITE-Base-3B	85.6	73.9	63.5	61.7	40.3	64.7
Open-LLaMA-3B-v2	88.0	77.9	63.1	63.3	40.1	71.4
Sheared-LLaMA-2.7B	87.5	76.9	65.0	63.3	41.6	71.0
LLaMA-MoE-3.0B	84.2	77.5	63.6	60.2	40.9	70.8
LLaMA-MoE-3.5B (4/16)	87.6	77.9	65.5	65.6	44.2	73.3
LLaMA-MoE-3.5B (2/8)	88.4	77.6	66.7	65.3	43.1	73.3

Model	Continued		LM	World Knowledge		Average
	LogiQA	BoolQ (32)	LAMBADA	NQ (32)	MMLU (5)	
OPT-2.7B	25.8	63.3	63.6	10.7	25.8	50.3
Pythia-2.8B	28.1	65.9	64.6	8.7	26.8	51.5
INCITE-Base-3B	27.5	65.8	65.4	15.2	27.2	53.7
Open-LLaMA-3B-v2	28.1	69.2	67.4	16.0	26.8	55.6
Sheared-LLaMA-2.7B	28.3	73.6	68.3	17.6	27.3	56.4
LLaMA-MoE-3.0B	30.6	71.9	66.6	17.0	26.8	55.5
LLaMA-MoE-3.5B (4/16)	29.7	75.0	69.5	20.3	26.8	57.7
LLaMA-MoE-3.5B (2/8)	29.6	73.9	69.4	19.8	27.0	57.6

With 200B tokens continual pretraining,

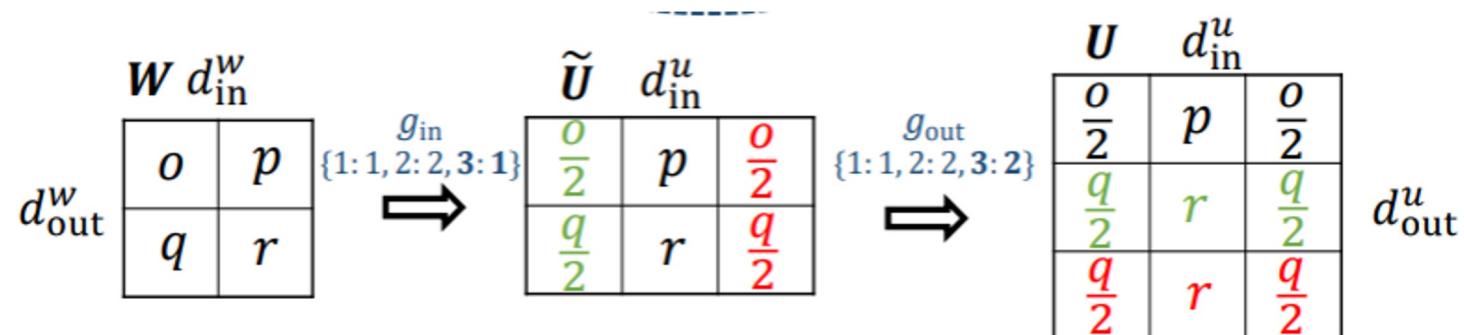
LLaMA-MoE surpasses dense models with similar activation parameters.

Sparse Dropout

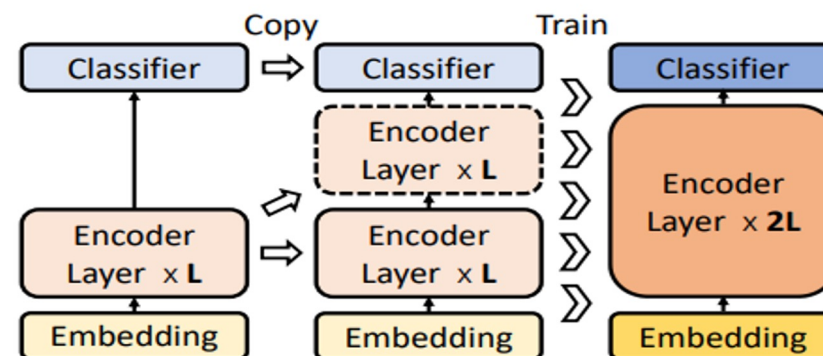


1. Gradually increasing the number of experts when training the model;

1. Gradually increasing FFN dimension, following bert2BERT



1. Gradually increasing the layer, following stackBERT



Outline

1. MoE Design

- Architecture, Auxiliary Loss, Routing

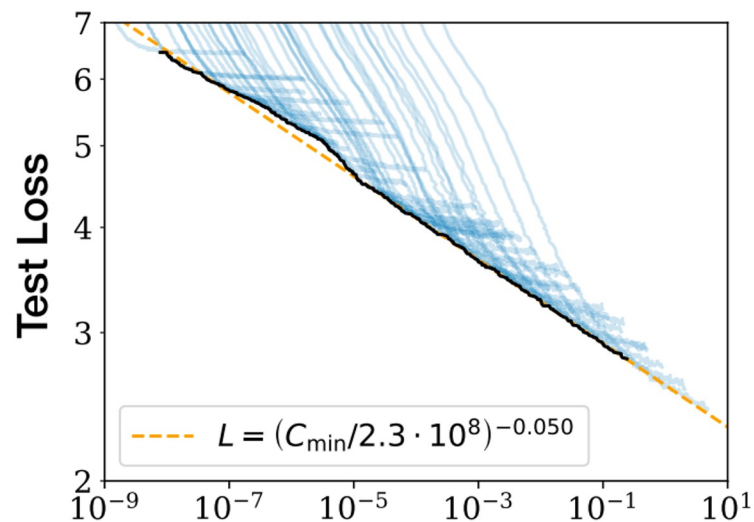
1. Building MoE from Dense LLMs

- Upcycling
- Sparse Splitting

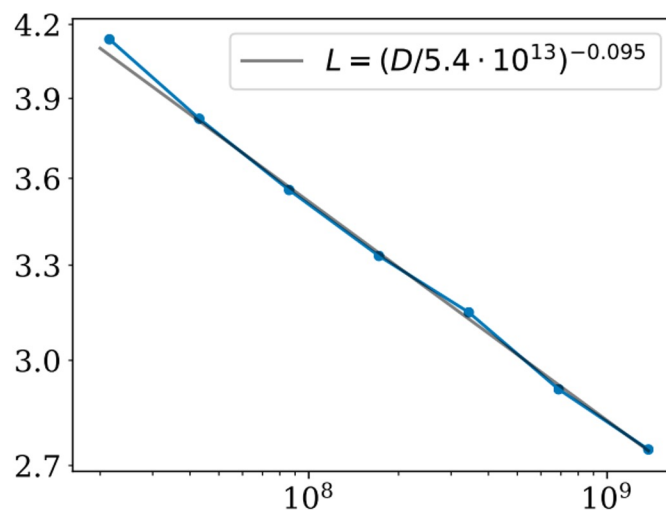
2. MoE Beyond Efficiency

- Scaling Law, Fine-tuning MoE
- Other derivatives in this era

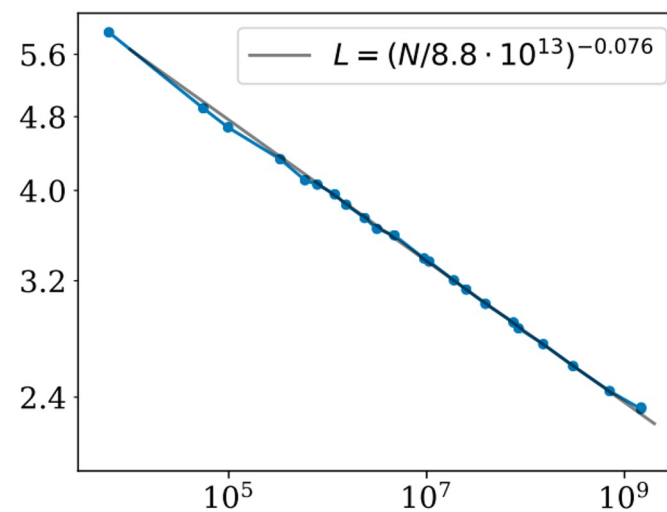
Scaling Law



Compute
PF-days, non-embedding



Dataset Size
tokens



Parameters
non-embedding

Upstream Scaling (Pre-training)

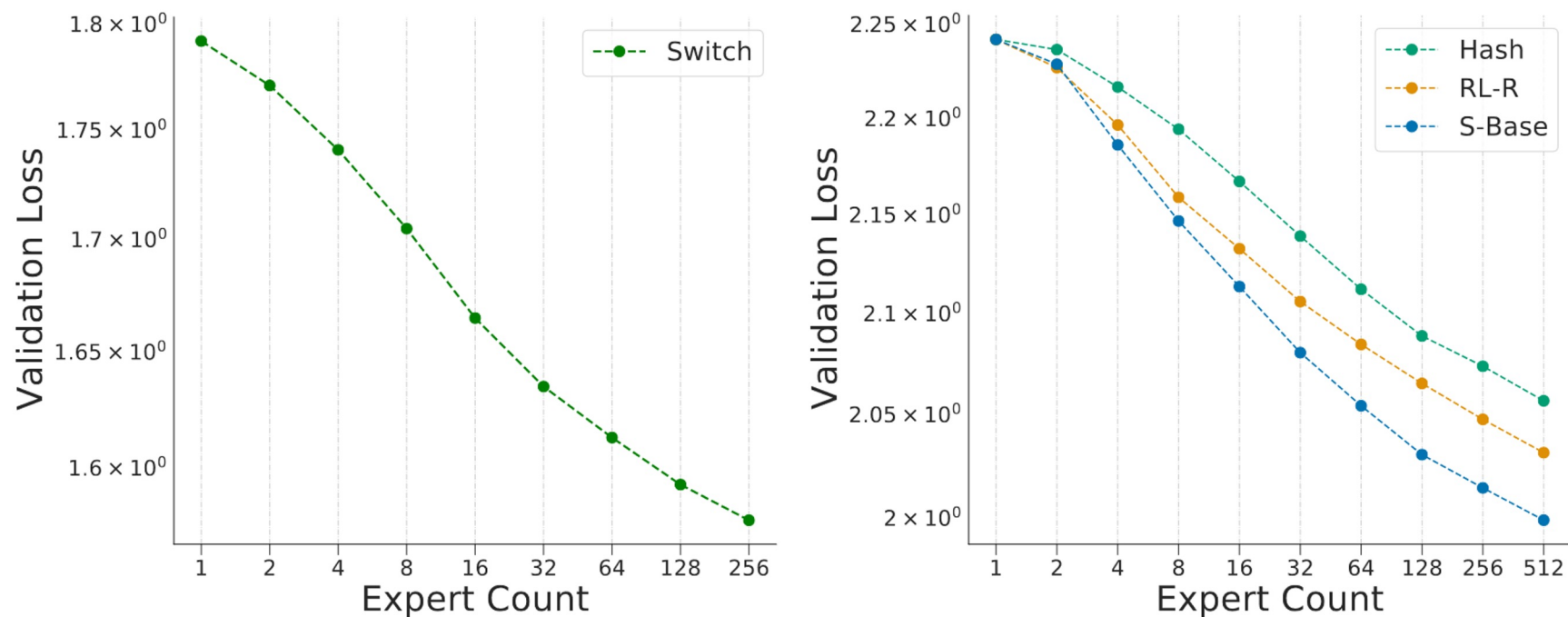


Figure 3: **Sparse scaling plots with expert count.** The cross-entropy scaling plots as a function of the number of experts are shown from [Fedus et al. \(2021\)](#) (**left**) and the three sparse variants from [Clark et al. \(2022\)](#), S-Base, RL-R, Hash (**right**). The top left-most point in both plots is an approximately compute-matched dense model. As the expert count increases, the models become increasingly sparse and yield lower validation losses.

Downstream Scaling (Fine-tuning)

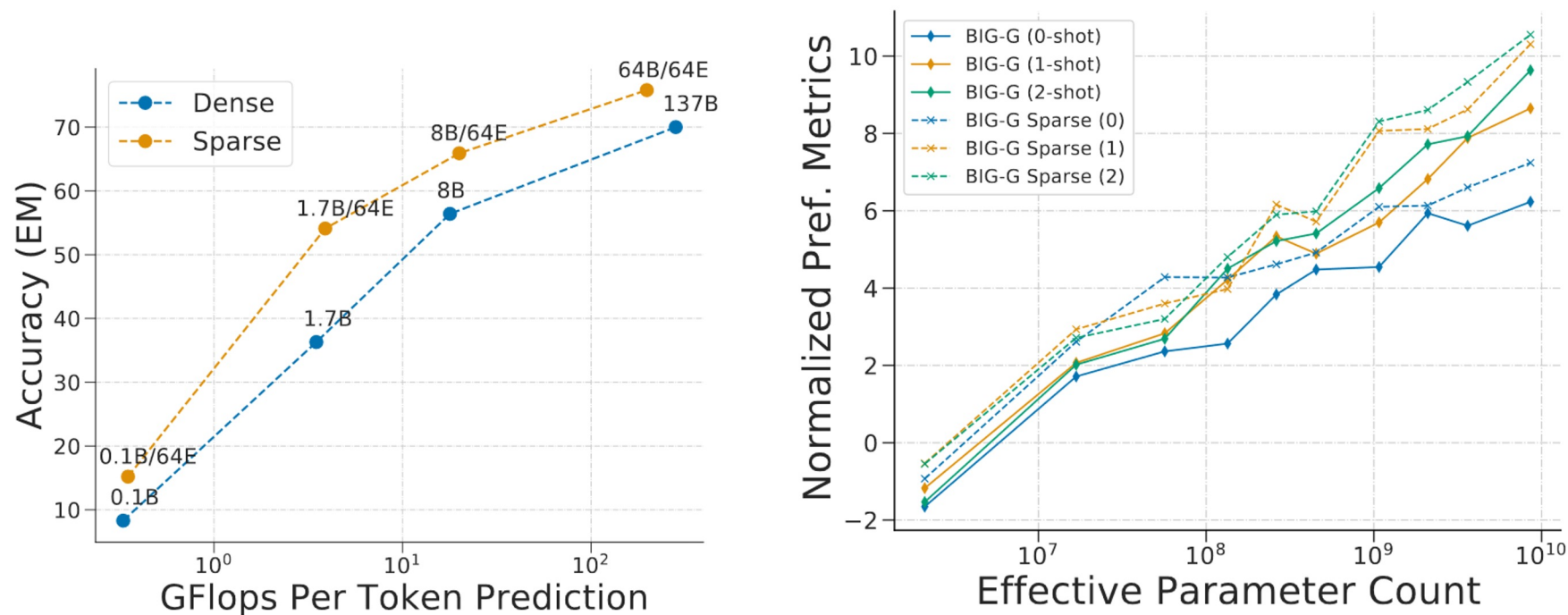


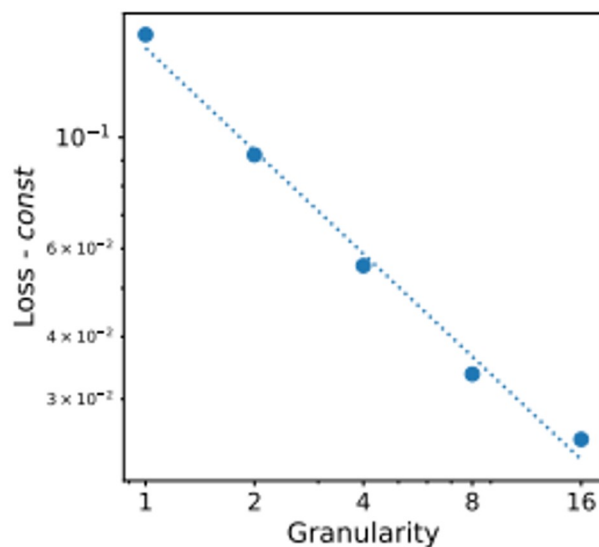
Figure 4: **Sparse scaling for few-shot inference.** **Left:** Du et al. (2021) measures the few-shot inference performance on TriviaQA, demonstrating consistent gains of sparse MoE models over dense models up to 137B parameters. Each label, such as 8B/64E, says how many parameters per input are used (8B) and how many experts (64E). **Right:** BigBench (Srivastava et al., 2022) studied the few-shot scaling properties on a larger set of 161 contributed JSON tasks to confirm improvements of sparse expert models over their FLOP-matched dense counterparts.

$$\bar{N} \triangleq (N)^{\alpha(\hat{E})/\alpha(E_{\text{start}})} \left(\hat{E}/E_{\text{start}} \right)^{b/\alpha(E_{\text{start}})}$$
$$\alpha(E) = a + c \log E$$

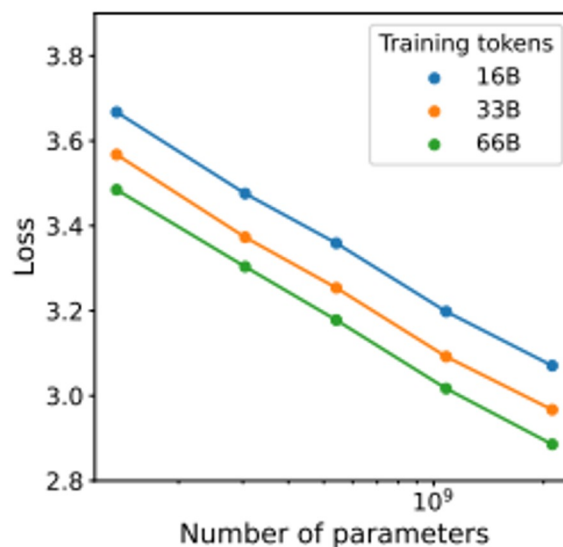
“The size \bar{N} of a dense model giving the same performance as a Routing Network.”

Scaling (Fine-grained Experts)

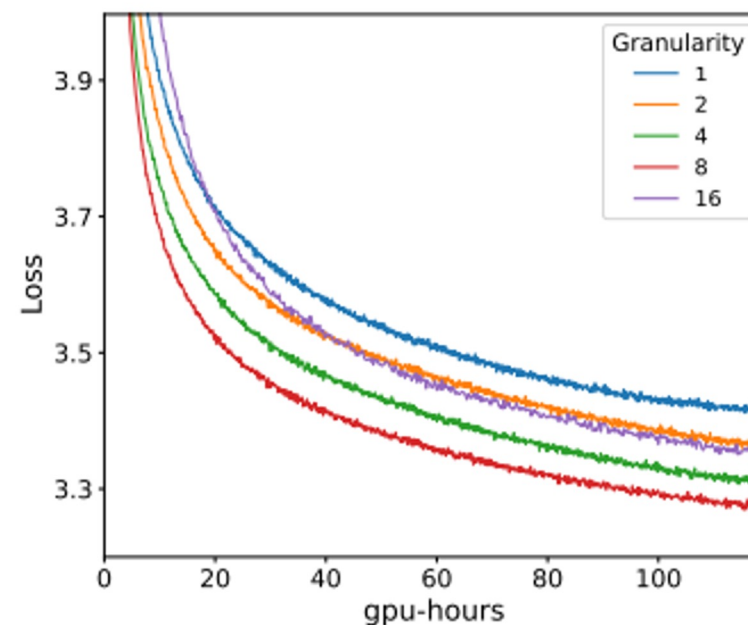
- Granularity denotes the size as the feed-forward layer divides the inner dimension of each expert network;
- Increasing granularity results in a lower loss, in different numbers of training tokens;
- Considering GPU-hours, the conclusion is slightly different.



(a)



(b)



- Instruction-tuning is better than fine-tuning: training with mixed prompt settings (zero-shot, few-shot, and chain-of-thought;
- Sparse models have performed remarkably well in the regime of large datasets, but have sometimes performed poorly when fine tuning data is limited;
- In general, MoE model performance scales better with respect to the number of tasks rather than the number of experts.

Flan-MoE

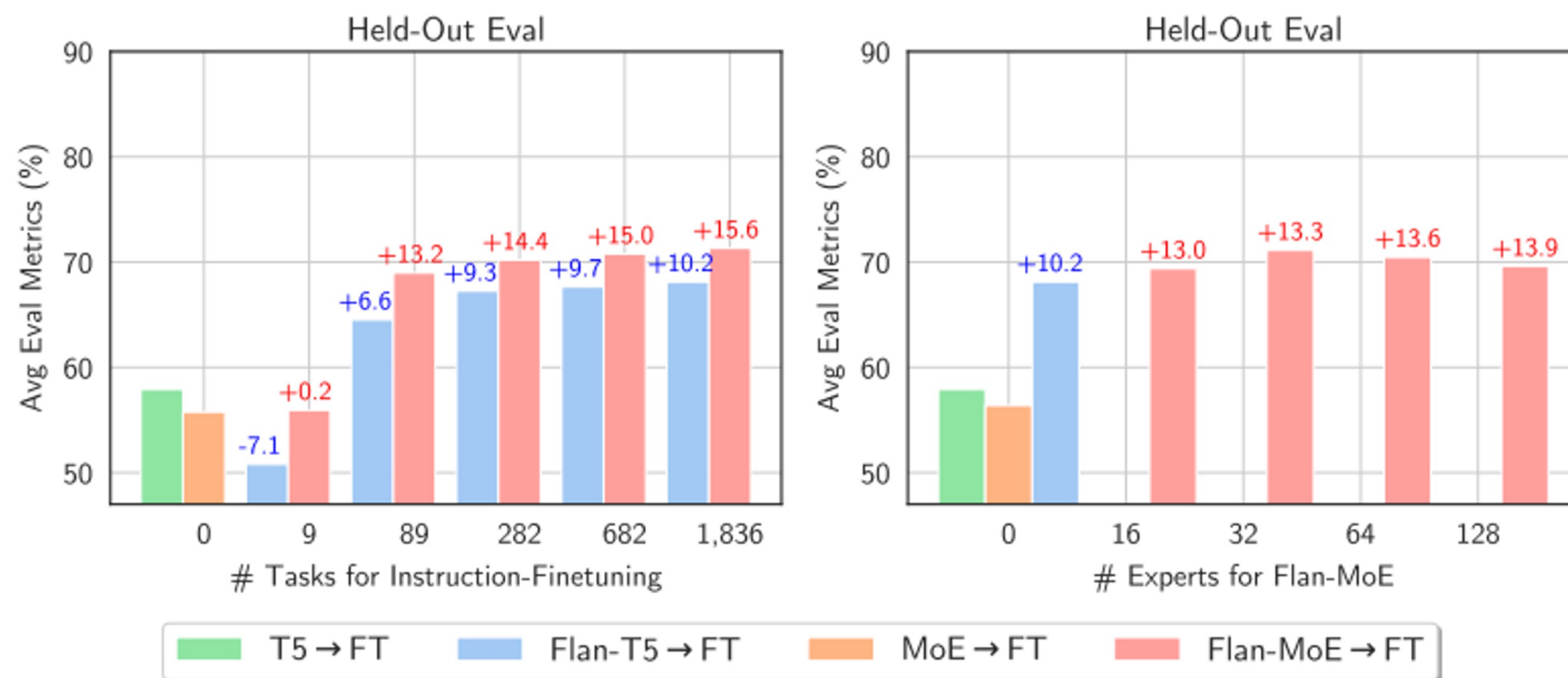


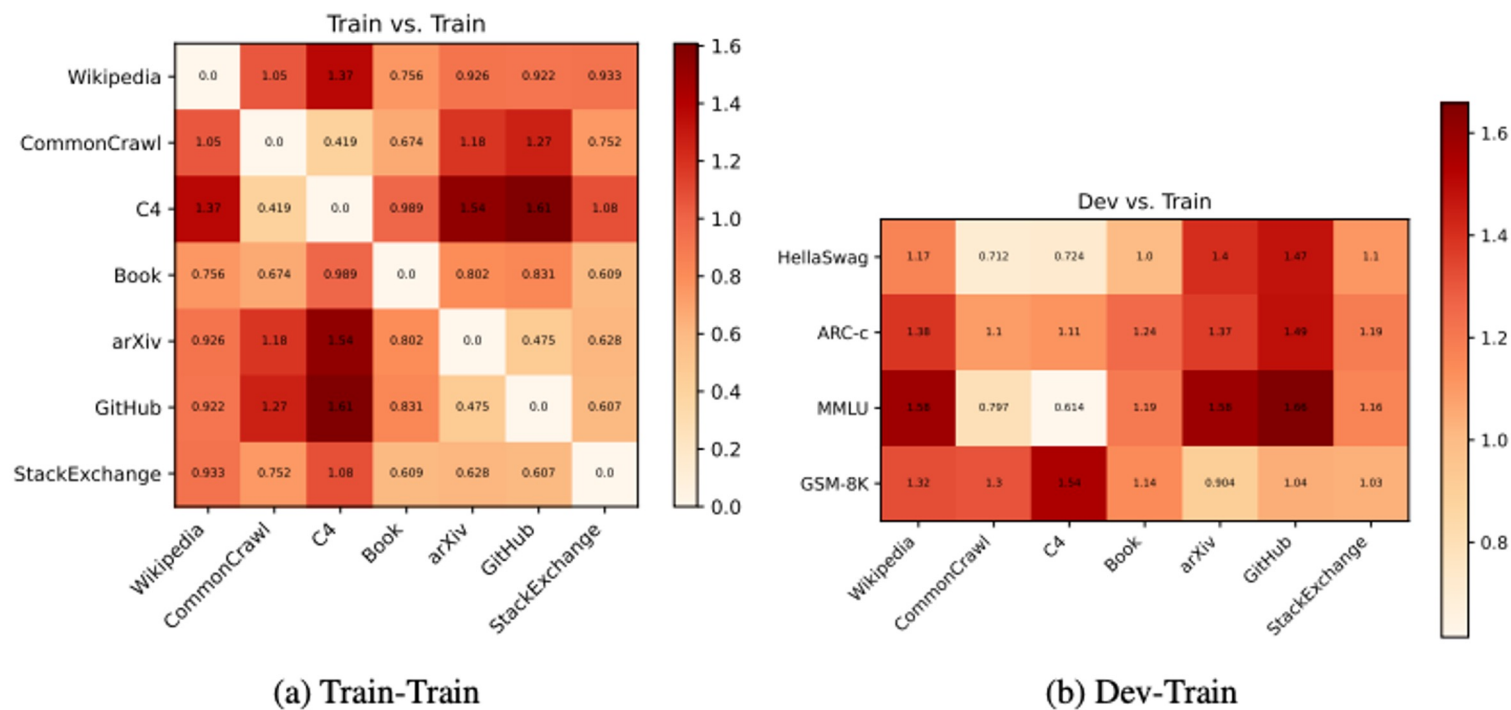
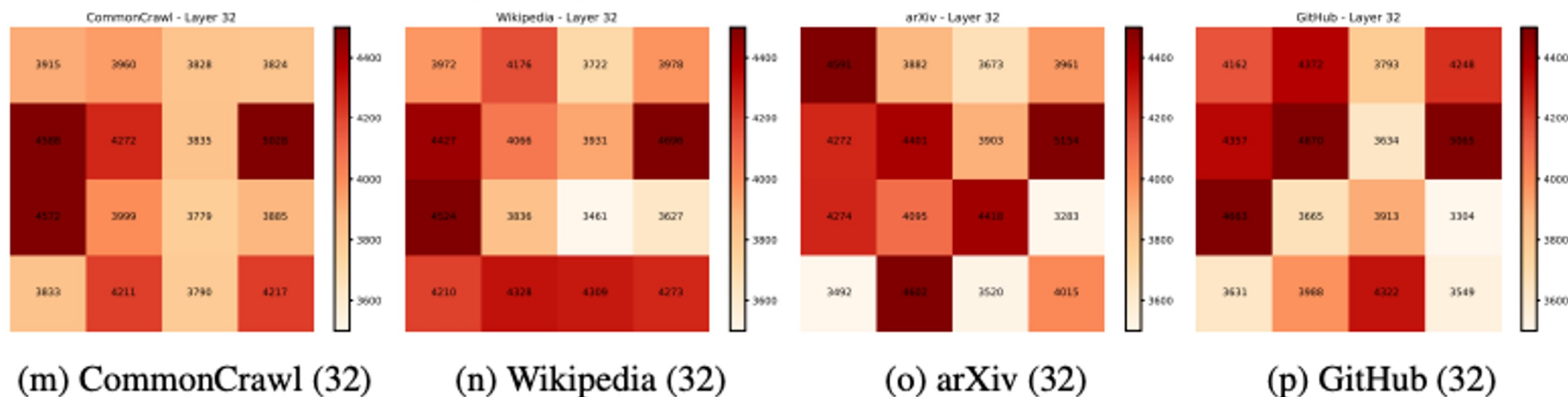
Figure 1: The effect of instruction tuning on MoE models versus dense counterparts for base-size models (same flops across all models in this figure). We perform single-task finetuning for each model on held-out benchmarks. **Compared to dense models, MoE models benefit more from instruction-tuning, and are more sensitive to the number of instruction-tuning tasks.** Overall, the performance of MoE models scales better with respect to the number of tasks, than the number of experts.

Expert Specialization

Expert	Top-5 preceding tokens
5	year, years, billion, millions, tonnes
9	electronic, local, public, national, outdoor
34	to, will, should it, may
42	two, 50, 1, 80, 000
62	work, started, involved, working, launched
72	is, was, be, been, were
74	going, go, come, back, return
101	B, T, W, H, k

Table 2: **Expert specialization based on preceding context in BASE Layers.** We reproduce a portion of table of [Lewis et al. \(2021\)](#), presenting the most frequent preceding top-five tokens for the selected experts. This example shows experts specializing in punctuation, conjunctions & articles, verbs, visual descriptions, proper names, counting & numbers.

Expert Specialization

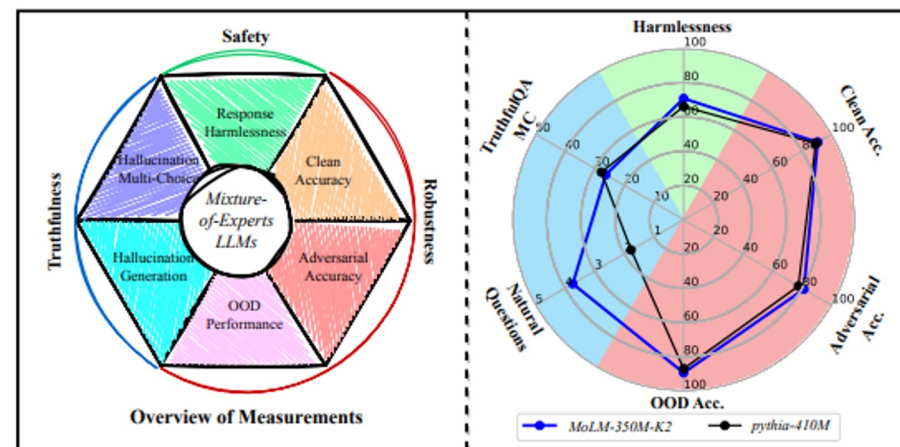


MoE-RBench

Model	Std. RA	Std. SA	Adv. RA				Adv. SA			
			R1	R2	R3	Avg.	R1	R2	R3	Avg.
<i>t5-base</i>	80.20	90.95	50.60	46.50	47.67	48.26	89.62	89.60	90.99	90.07
<i>switch-base</i>	82.40	92.01	52.40	48.6	50.08	50.36	90.14	91.39	91.70	91.08
<i>pythia-410M</i>	77.44	89.17	47.40	43.70	45.33	45.48	87.62	88.03	87.79	87.81
<i>pythia-1.4B</i>	78.28	90.11	49.00	45.70	47.42	47.37	88.58	88.92	90.69	89.40
<i>MoLM-350M-K2</i>	81.15	90.43	49.30	47.00	48.00	48.10	87.91	89.05	90.24	89.07
<i>MoLM-700M-K4</i>	81.27	91.58	54.20	47.90	49.17	50.42	89.29	90.20	90.66	90.05
<i>OpenLlama-3B</i>	83.33	93.14	60.70	50.90	54.17	55.26	91.69	91.95	92.84	92.16
<i>LlamaMoE-3B-K2</i>	83.73	92.44	62.10	53.20	56.33	57.21	91.93	92.38	92.73	92.35
<i>LlamaMoE-3.5B-K4</i>	84.68	93.26	67.90	55.70	56.83	60.14	92.33	92.47	92.94	92.58
<i>LlamaMoE-3.5B-K2</i>	84.74	93.30	67.90	54.50	59.58	60.66	92.22	92.88	93.15	92.75

- MoE models not only respond with a comparable degree of safety and correctness, but also exhibit markedly enhanced robustness compared to the dense counterparts.

Model	ID	Word OOD		Sentence OOD							
		Aug.	Shake	p=0				p=0.6			
				Tweet	Shake	Bible	Poetry	Tweet	Shake	Bible	Poetry
<i>t5-base</i>	93.8	91.8	89.1	91.2	90.4	88.4	86.9	90.5	86.1	84.9	88.4
<i>switch-base</i>	94.5	94.0	91.1	92.5	91.9	89.4	88.0	92.4	89.1	85.8	88.0
<i>pythia-410m</i>	92.4	89.3	87.6	88.8	89.0	86.0	86.2	89.6	85.2	81.9	86.5
<i>pythia-1.4b</i>	95.1	89.9	90.0	91.1	90.9	87.7	87.8	91.6	87.2	86.2	88.0
<i>MoLM-350M-K2</i>	94.4	92.2	90.0	90.3	91.6	88.8	88.1	91.7	86.5	86.6	88.1
<i>MoLM-700M-K4</i>	95.5	92.3	90.1	91.5	90.6	89.1	88.2	92.2	87.7	86.6	88.4
<i>OpenLlama-3b</i>	96.8	95.8	93.7	92.8	91.9	89.5	88.0	92.1	89.3	86.7	88.5
<i>LlamaMoE-3.5B-K4</i>	96.9	95.3	91.8	94.5	93.0	90.4	90.1	94.3	89.6	88.6	89.3
<i>LlamaMoE-3.5B-K2</i>	96.9	96.1	92.2	93.8	93.1	90.6	89.3	93.8	90.6	86.8	91.4
<i>LlamaMoE-3B-K2</i>	96.6	95.2	93.7	93.0	92.2	89.8	88.1	92.7	89.9	87.5	88.7



Thanks

Q&A

Yu Cheng

chengyu@cse.cuhk.edu.hk



Tutorial: Mixture-of-Experts in the Era of LLMs: A New Odyssey

Mixture-of-Experts at Speed and Scale: A System Perspective

Minjia Zhang

University of Illinois at Urbana-Champaign

minjiaz@illinois.edu



ICML
International Conference
On Machine Learning



Outline

- Motivation and Challenges
- Training Large-Scale MoEs
 - Expert Parallelism and its Combination with 3D Parallelism
- Highly-Scalable MoE Training System
 - DeepSpeed-MoE
 - DeepSpeed-TED
 - Tutle

AI Scale is Limited By Compute

- Compute is the primary challenge of training massive models
- Ambitious model at scale and time to train

Model	Model Size	Hardware	Days to Train
BLOOM	176B	384 A100 GPUs	115 days
OPT	175B	992 A100 GPU	56 days
MT-NLG	530B	2200 A100 GPU	60 days
PaLM	540B	6144 TPU v4	57 days

Next jump in scale:

- Next-generation hardware
- Significant investment in GPUs

Next AI Scale?

- Can we achieve next generation model quality on current generation of hardware?
- From a computation perspective sparse Mixture-of-Experts provides a promising path
 - Scale at sub-linear cost

MoE Models are Sparse and Need Less Compute

Dense Models:

- All parameters are used in forward and backward paths
- Increasing model capacity needs more computation
- **Larger model size → Higher compute requirements (FLOPs)**

Sparse MoE models

- Sparse utilization of subset of parameters based on input
- Same computation is needed regardless of the model size
- **Larger model size → Similar/Same Compute requirements**

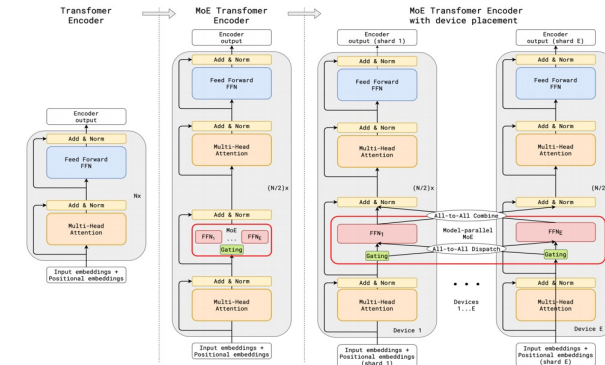


Figure 3: Illustration of scaling of Transformer Encoder with MoE Layers. The MoE layer replaces the every other Transformer feed-forward layer. Decoder modification is similar. (a) The encoder of a standard Transformer model is a stack of self-attention and feed forward layers interleaved with residual connections and layer normalization. (b) By replacing every other feed forward layer with

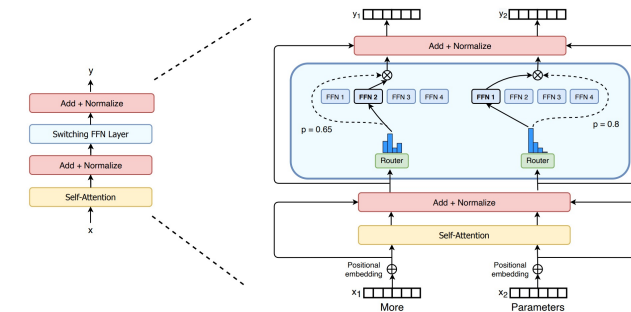
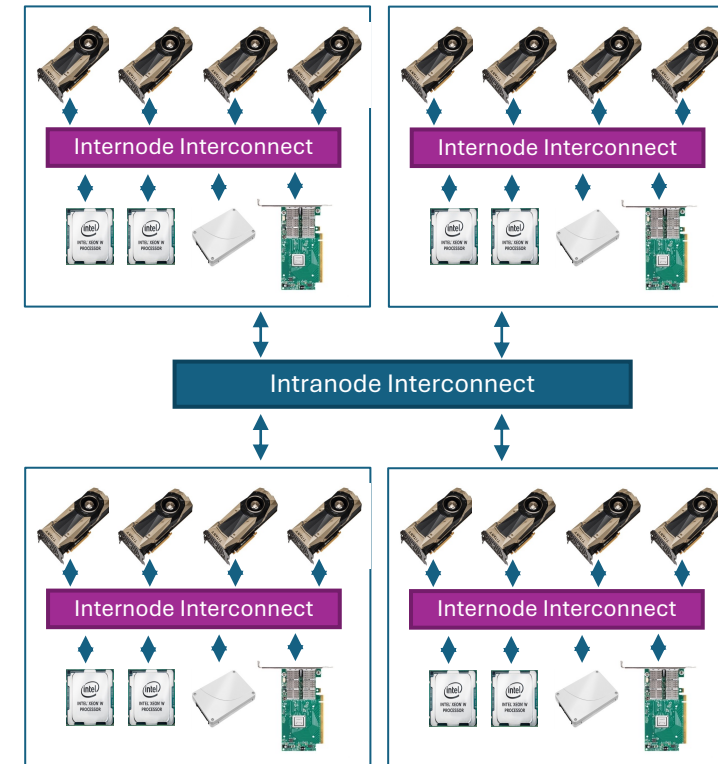
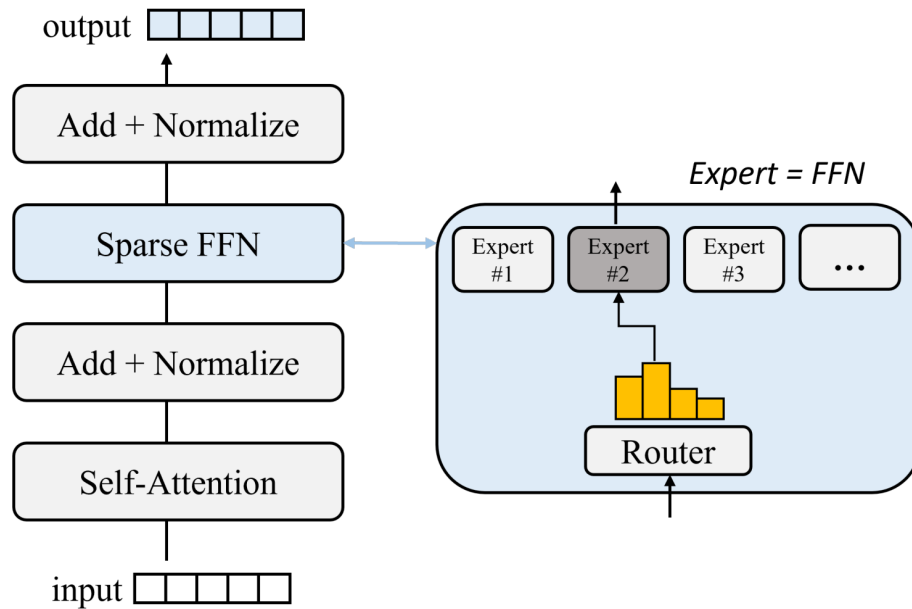


Figure 2: **Illustration of a Switch Transformer encoder block.** We replace the dense feed forward network (FFN) layer present in the Transformer with a sparse Switch FFN layer (light blue). The layer operates independently on the tokens in the sequence. We diagram two tokens ($x_1 = \text{"More"}$ and $x_2 = \text{"Parameters"}$ below) being routed (solid lines) across four FFN experts, where the router independently routes each token. The switch FFN layer returns the output of the selected FFN multiplied by the router gate value (dotted-line).

Mixture of Experts (MoE): Overview

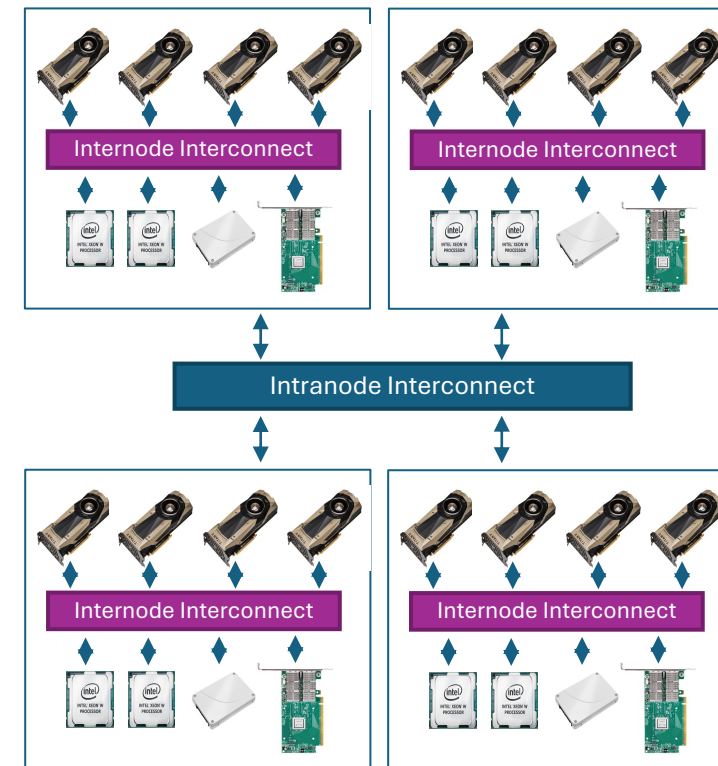
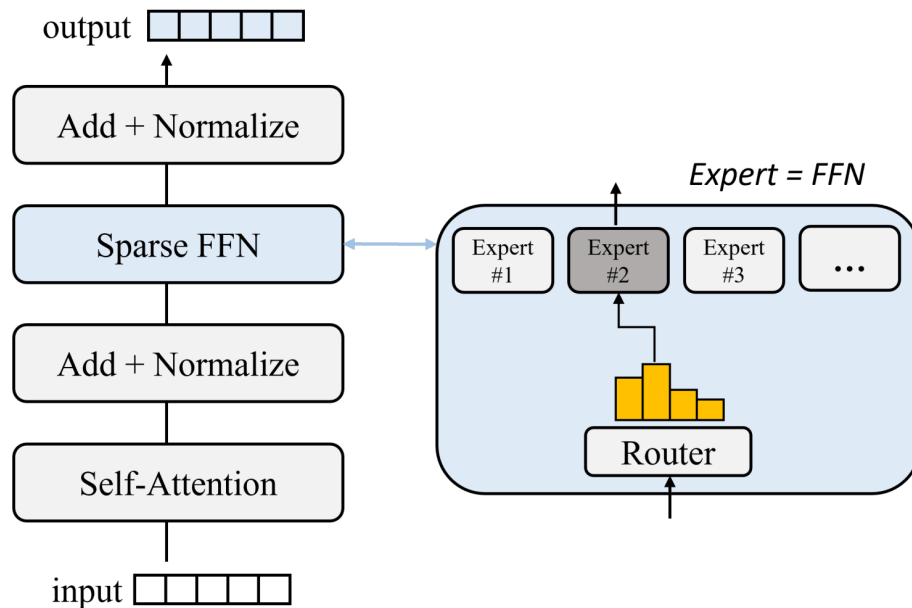
- MoE models have been around for a while..
- [Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer](#)
 - Harder to scale, instability during training, and inefficient training
- [GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding](#)
 - 600B models beating 96-layer dense models, 10x training speedup, generic sharding framework (Tensorflow XLA)
 - Less stability with larger models, full precision training
- [Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity](#)
 - More efficient training
 - Top-1 gating instead of top-2/top-k, Better initialization conditions, Mixed precision training: FP32 gating (instead of FP16), Stable training with larger models
 - SOTA results on language understanding task

MoE Training Challenges on Modern Hardware with Massive Parallelism



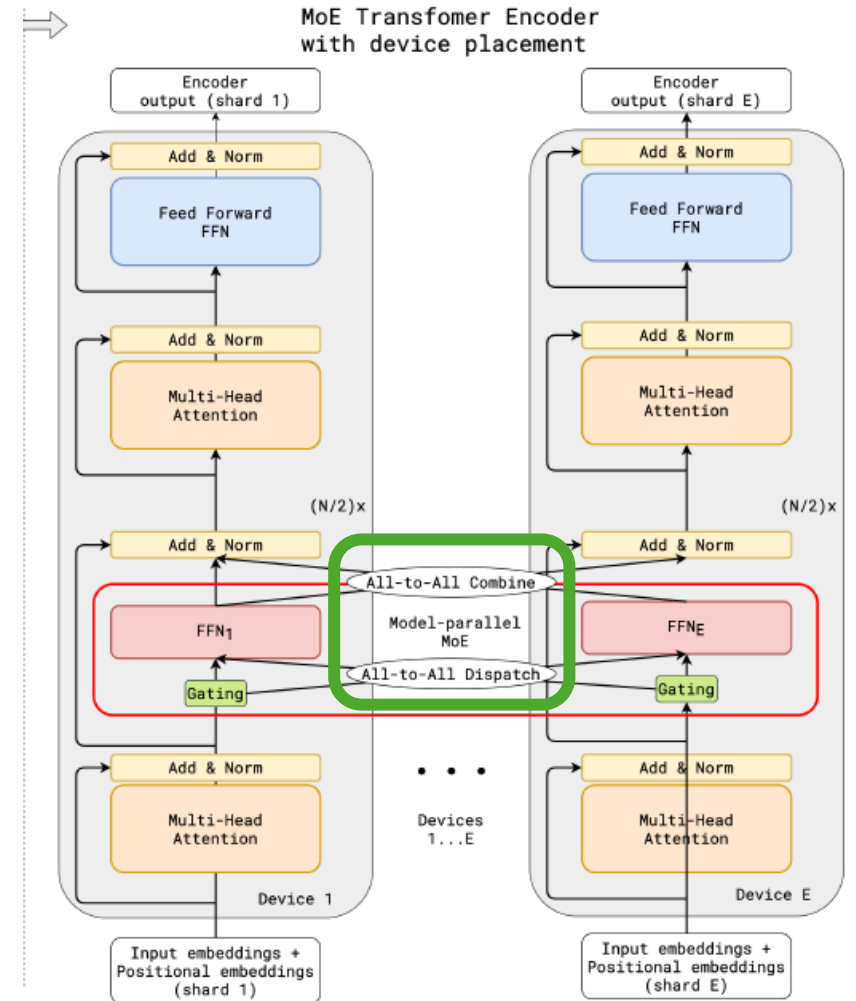
MoE Training Challenges on Modern Hardware with Massive Parallelism

- How to break the memory wall to enable massive MoEs?
- How to efficiently route tokens to different experts across GPUs?
- How to minimize communication overhead while achieving high per-GPU compute throughput?



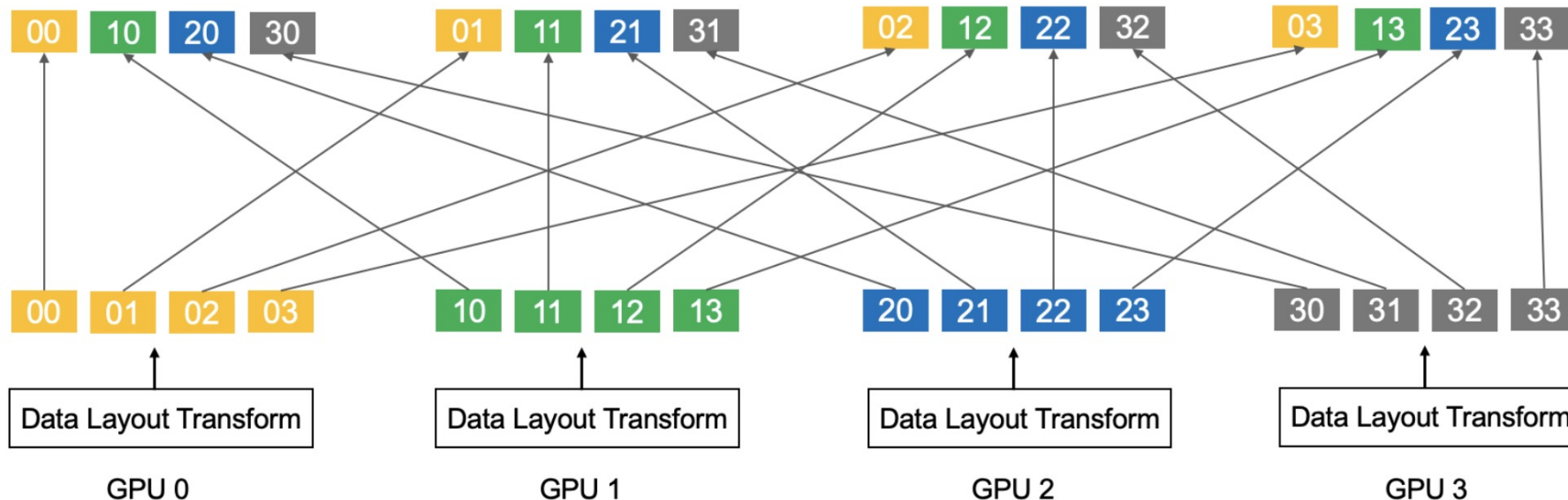
Expert Parallelism

- Expert parameters – partitioned (sharded)
 - Like model parallelism (MP)
 - Each expert process a subset of tokens
- **Two All-to-All(s) in Forward and Backward**



Expert Parallelism

1. Gating function: decide target experts for each token
2. **Dispatch phase:**
 - a. 1st layout transformation: tokens to the same target experts are grouped in a continuous memory buffer
 - b. 1st All2All: dispatch tokens to their corresponding experts
3. Expert compute: each expert process its tokens
4. **Combine phase:**
 - a. 2nd All2All: combine processed tokens back to their GPUs
 - b. 2nd layout transform: restore tokens to their original positions



How to Design Highly-Scalable Training Systems for Trillion-Parameter MoEs?

- DeepSpeed-MoE [1]
 - Multi-dimensional parallelism for scaling both the base model and expert layers
- DeepSpeed-TED [2]
 - Further push the limit of MoE scalability by eliminating unnecessary communication in hybrid parallelism
- Tutle [3]
 - System and algorithm co-design achieving excellent scalability at 2048 A100 GPUs

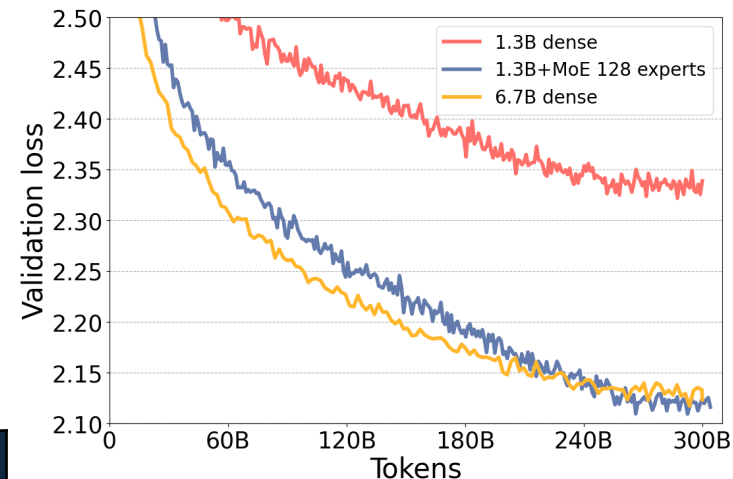
DeepSpeed-MoE: Multidimensional Parallelism

Short Name	Flexible Parallelism Combinations	Benefit
E	Expert	Scales the model size by increasing the number of experts
E+D	Expert + Data	Accelerates training throughput by scaling to multiple data parallel groups
E+Z	Expert + ZeRO	Partitions the nonexpert parameters to support larger base models
E+D+M	Expert + Data + Model	Supports massive hidden sizes and even larger base models than E+Z
E+D+Z	Expert + Data + ZeRO	
E+Z-Off+M	Expert + ZeRO-Offload + Model	Leverages both GPU and CPU memory for large MoE models on limited GPU resources

Optimal parallelism strategy depends on model and hardware specifics

DeepSpeed-MoE: Cheaper GPT Model Training with MoE

- 1.3B+MoE with 128 experts, compared to 1.3B and 6.7B dense (GPT-3 like)
- **8x** more parameters to same accuracy using MoE
- **5x** lower training cost to same accuracy using MoE



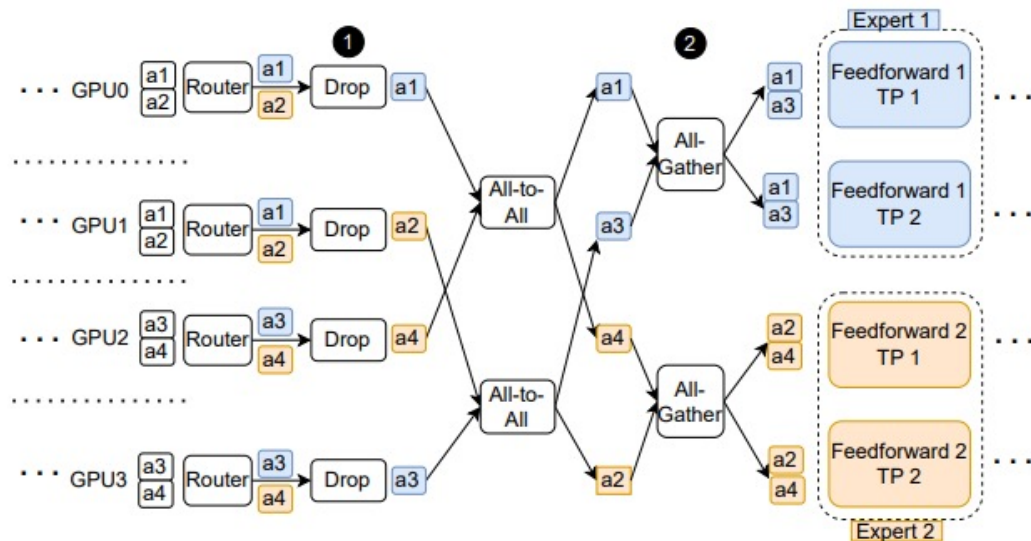
Case	Model size	LAMBADA: completion prediction	PIQA: commonsense reasoning	BoolQ: reading comprehension	RACE-h: reading comprehension	TriviaQA: question answering	WebQs: question answering
Dense GPT:							
(1) 350M	350M	52.03	69.31	53.64	31.77	3.21	1.57
(2) 1.3B	1.3B	63.65	73.39	63.39	35.60	10.05	3.25
(3) 6.7B	6.7B	71.94	76.71	67.03	37.42	23.47	5.12
Standard MoE GPT:							
(4) 350M+MoE-128	13B	62.70	74.59	60.46	35.60	16.58	5.17
(5) 1.3B+MoE-128	52B	69.84	76.71	64.92	38.09	31.29	7.19

	Training samples per sec	Throughput gain/ Cost Reduction
6.7B dense	70	1x
1.3B+MoE-128	372	5x

DeepSpeed-TED

- Further push the limit of MoE scalability by eliminating unnecessary communication

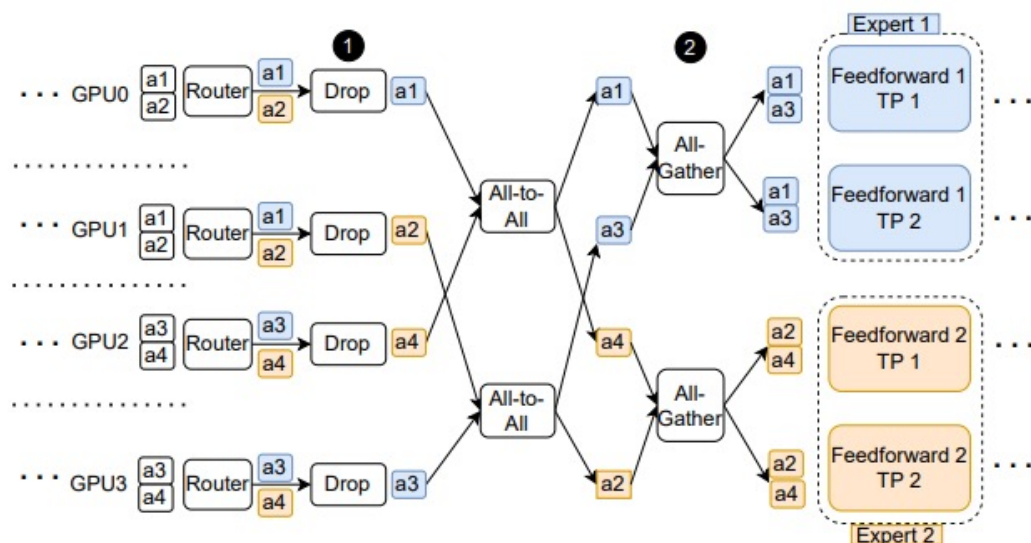
Duplicate token dropping (DTD): Eliminating unnecessary tokens, e.g., in all2all and all-gather from EP + TP.



DeepSpeed-TED

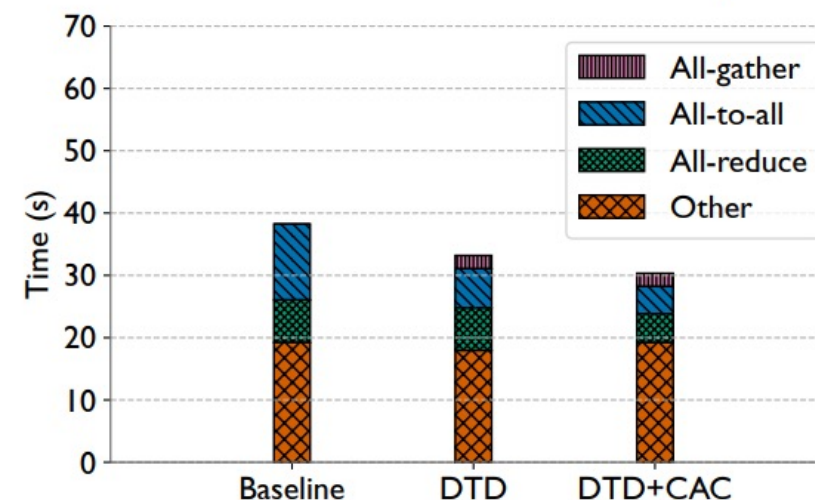
- Further push the limit of MoE scalability by eliminating unnecessary communication

Duplicate token dropping (DTD): Eliminating unnecessary tokens, e.g., in all2all and all-gather from EP + TP.



Communication-aware Activation Checkpointing (CAC): selective activation checkpointing by avoiding all2all during recomputation

Performance Profile of a 6.7B Base Model with 16 Experts on Summit



128 GPUs

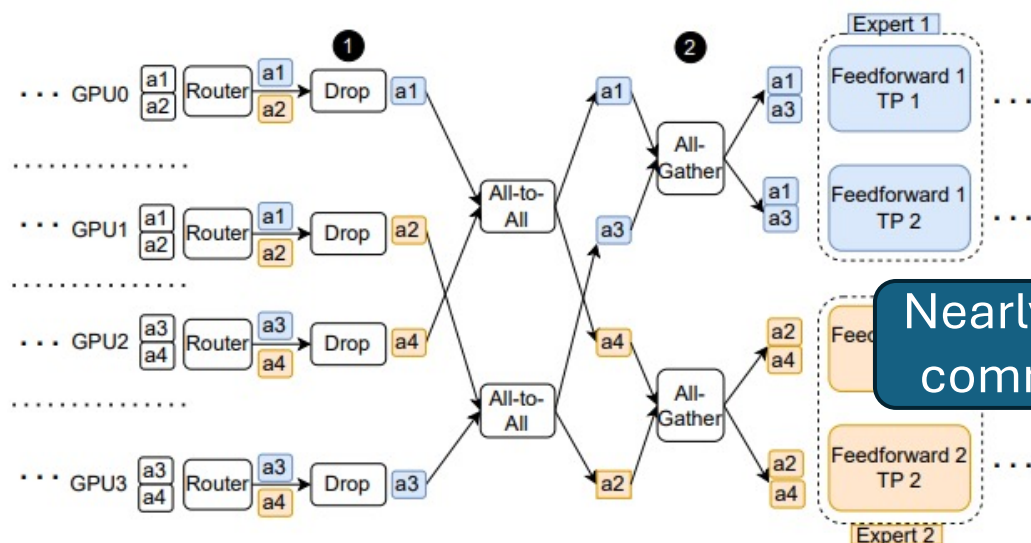
A Hybrid Tensor-Expert-Data Parallelism Approach to Optimize Mixture-of-Experts Training

DeepSpeed-TED

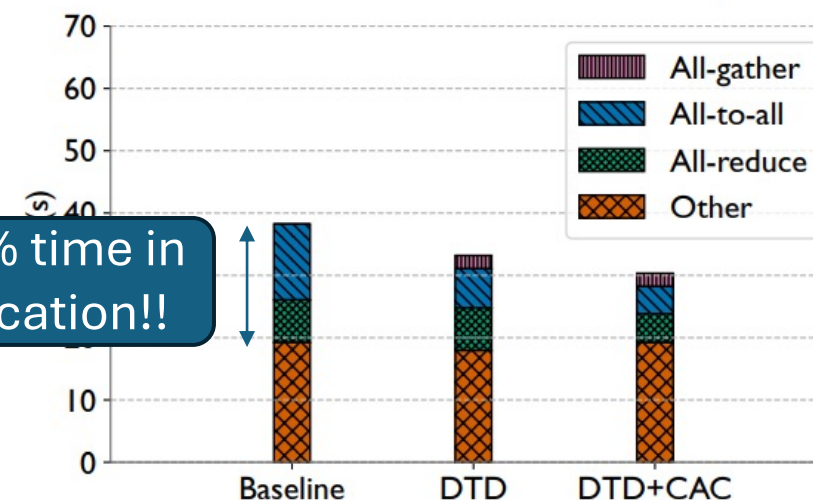
- Further push the limit of MoE scalability by eliminating unnecessary communication

Duplicate token dropping (DTD): Eliminating unnecessary tokens, e.g., in all2all and all-gather from EP + TP.

Communication-aware Activation Checkpointing (CAC): selective activation checkpointing by avoiding all2all during recomputation



Performance Profile of a 6.7B Base Model with 16 Experts on Summit



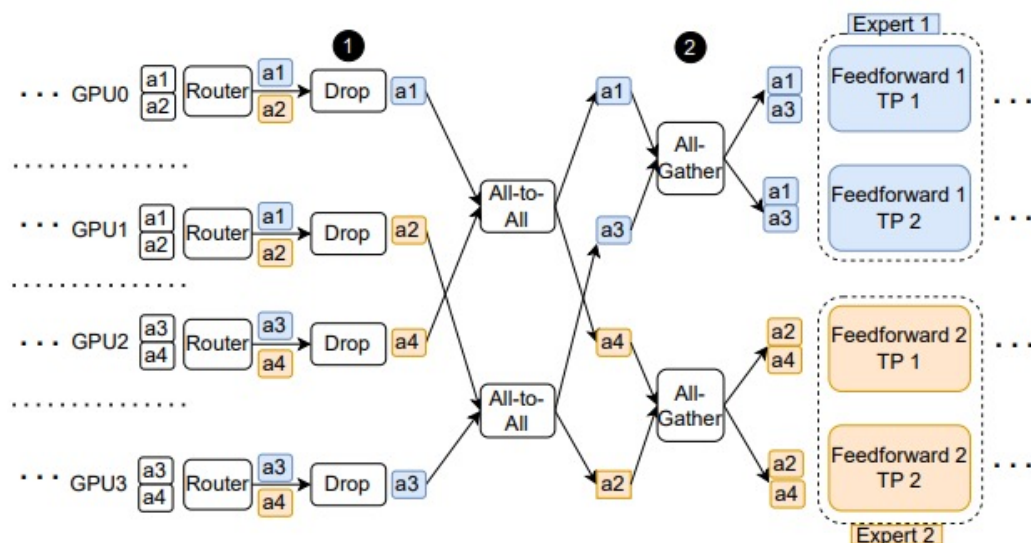
128 GPUs

A Hybrid Tensor-Expert-Data Parallelism Approach to Optimize Mixture-of-Experts Training

DeepSpeed-TED

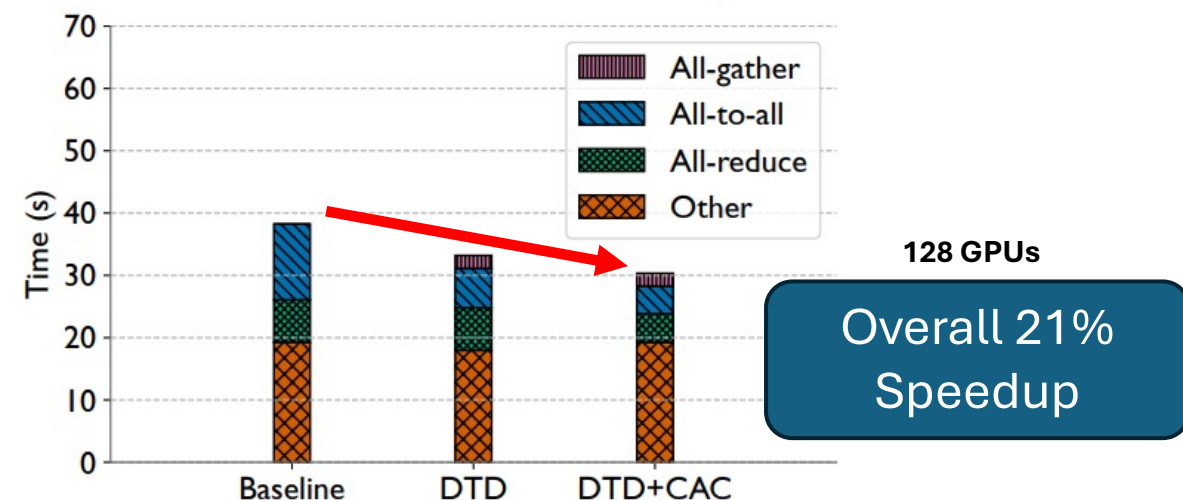
- Further push the limit of MoE scalability by eliminating unnecessary communication

Duplicate token dropping (DTD): Eliminating unnecessary tokens, e.g., in all2all and all-gather from EP + TP.



Communication-aware Activation Checkpointing (CAC): selective activation checkpointing by avoiding all2all during recomputation

Performance Profile of a 6.7B Base Model with 16 Experts on Summit



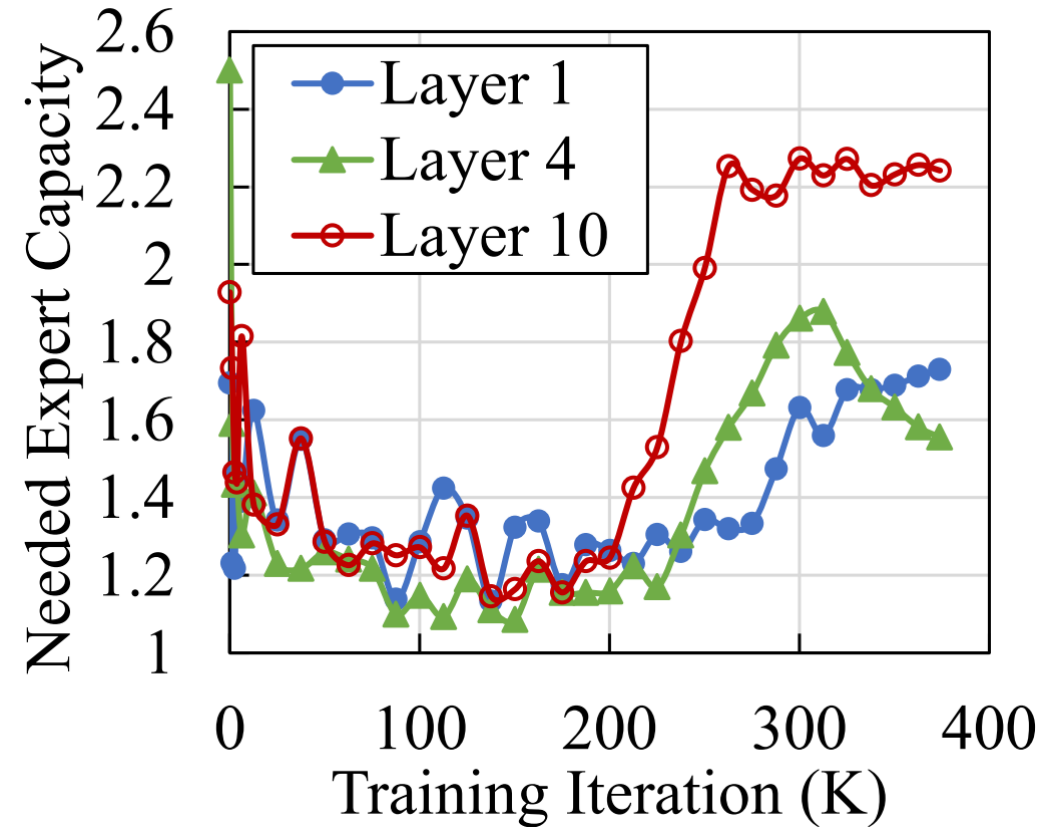
A Hybrid Tensor-Expert-Data Parallelism Approach to Optimize Mixture-of-Experts Training

Title: Adaptive MoE at Scale

- Key idea: system-algorithm co-design
- Dynamically adapt parallelism
- 2D hierarchical all2all
- Adaptive pipeline

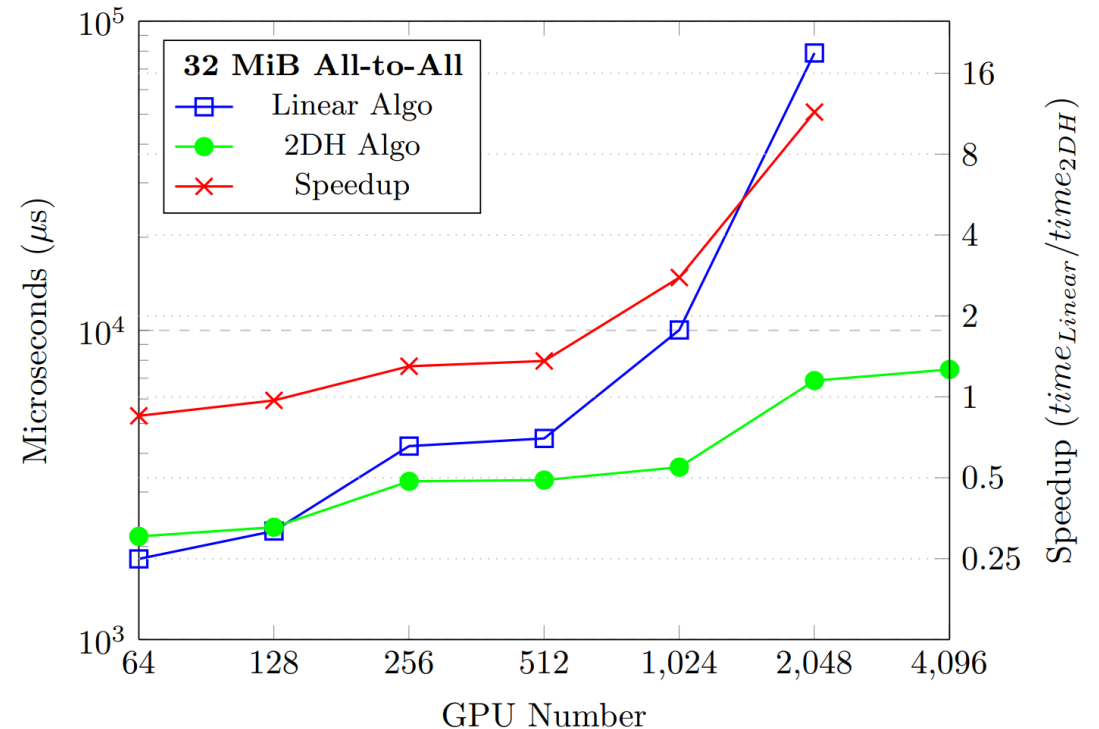
Title: Adaptive MoE at Scale

- Observation: Workload per expert changes during training
- Solution: Dynamically adapt parallelism



Title: Adaptive MoE at Scale

- Observation: All2all is expensive across nodes and with many small messages
- Solution 1: Take into account of network hierarchy with 2D hierarchical all2all: Intra-node all2all + Inter-node all2all
- Solution 2: Leverage highly-optimized communication collectives from MSCCL

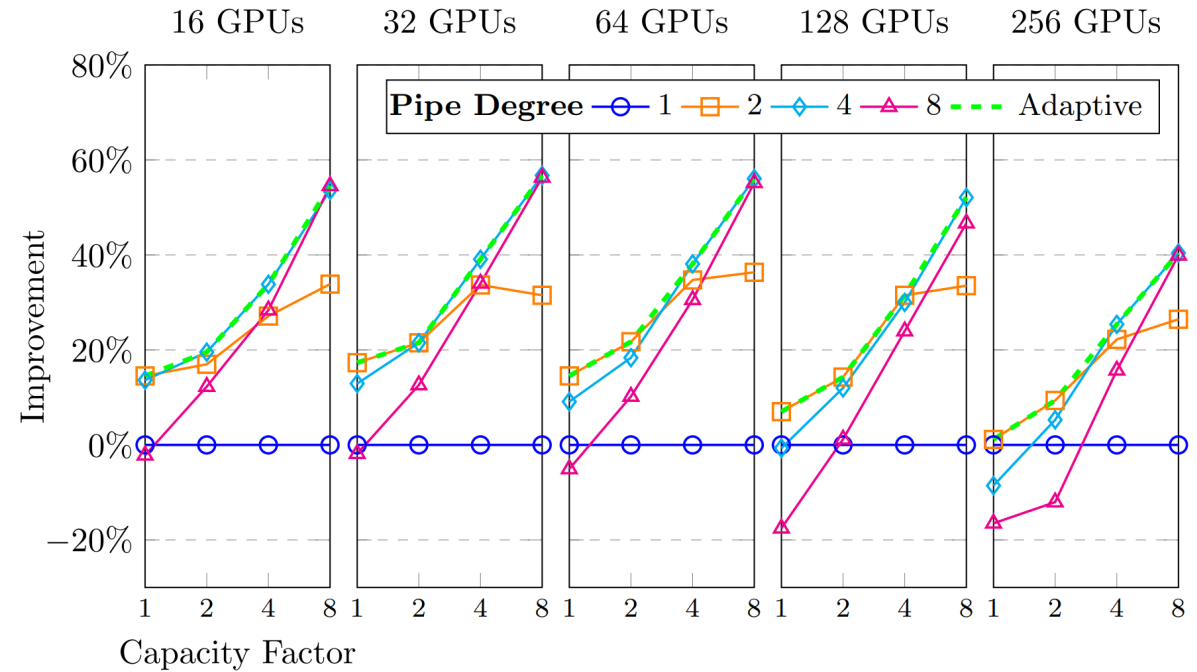


Up to 10x all2all speedup

Tutel: Adaptive mixture-of-experts at scale

Title: Adaptive MoE at Scale

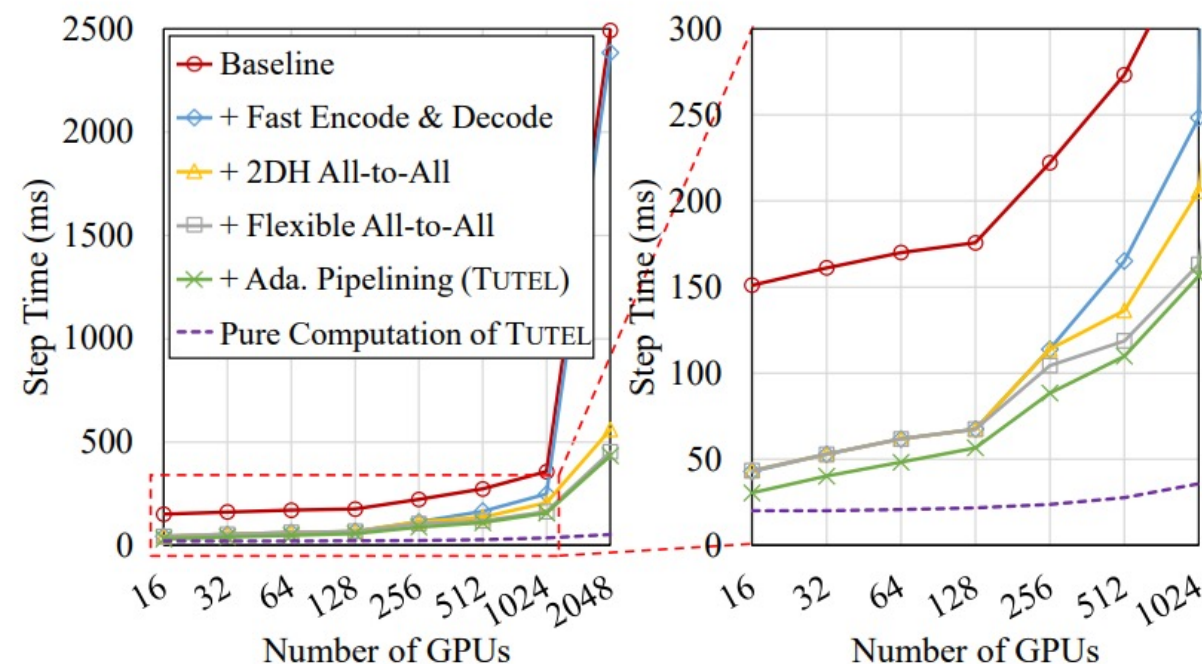
- Observation: Token partitioning + concurrent CUDA kernels => pipeline parallelism that overlap all2all with FFN layer compute
- Solution: Adaptive pipeline degree based on workloads



Up to 57% improvement in comparison to pipeline degree 1

Title: Adaptive MoE at Scale

- Dynamically adaptive parallelism
- Dynamic pipelining
- 2D hierarchical all2all



5.7× end-to-end speed at 2048 A100 GPUs!

Tutel: Adaptive mixture-of-experts at scale

Thank you!
Q&A

Minjia Zhang
minjiza@illinois.edu

Moving Forward

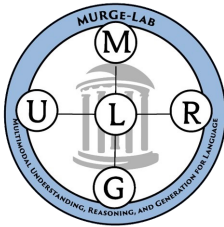
- Expect more optimizations against the training efficiency of MoE models, e.g., parameter-efficient MoE, multi-modal MoE
- System optimizations that leverage heterogeneous hardware resource to lower the cost of training and fine-tuning MoE
- Efficient MoE inference systems to achieve low latency and high-throughput

Key Extension: Multi-Modal MoE, Multi-Agent Communications

MOHIT BANSAL, TIANLONG CHEN

Computer Science

University of North Carolina at Chapel Hill



Multi-Modal Multi-Task Capability – Challenges?

(1) Modality/Task Forgetting Issues

Diverse modalities and tasks may prefer conflicting optimization directions, resulting in ineffective learning or knowledge forgetting.

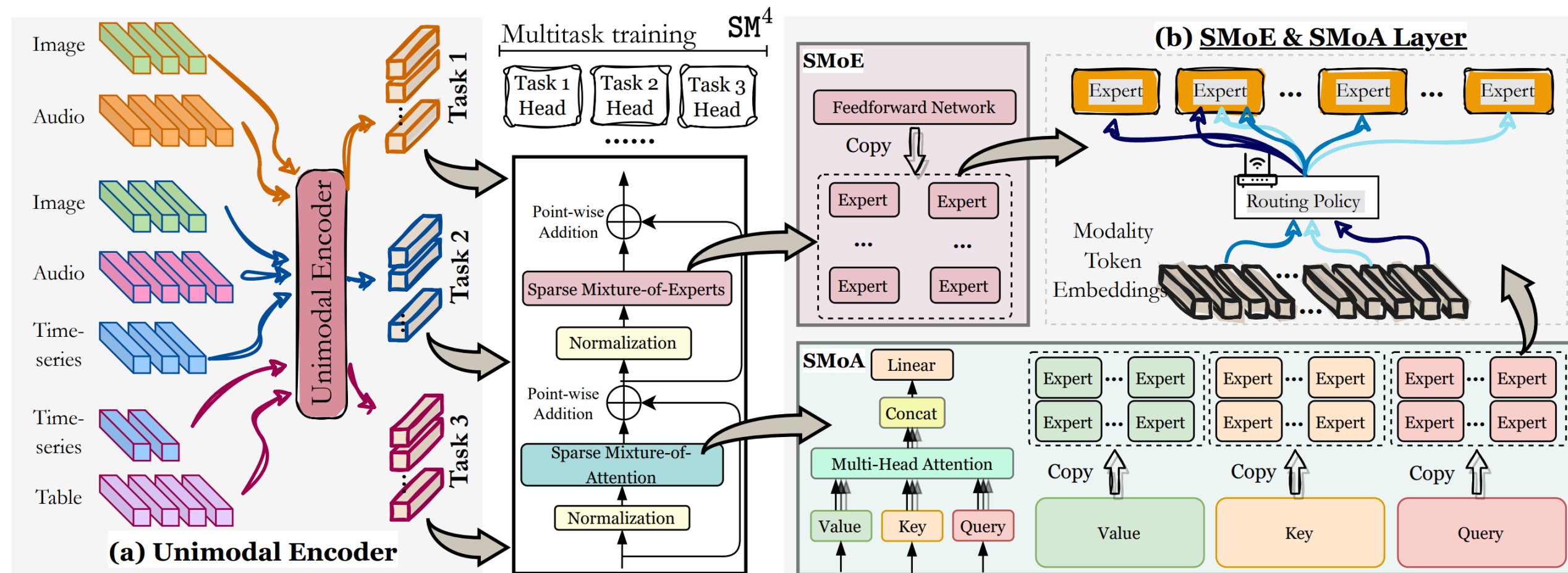
(2) Modality/Task Fitting Issues

Current LLMs or SMoE-based LLMs use a fixed amount of parameter counts for all modalities or tasks, which can end up over-fitting to simpler modalities or tasks or under-fitting complex ones.

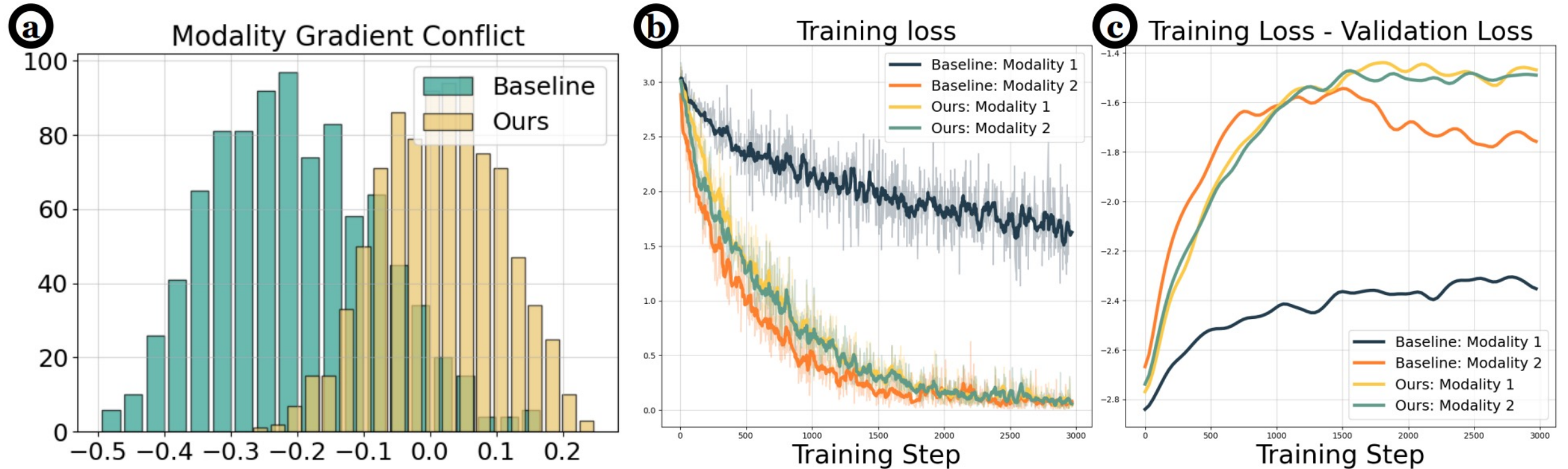
(3) Heterogeneous Learning Pace

The varied modality attributes, task resources (*i.e.*, the number of input samples), and task objectives usually lead to distinct optimization difficulties and convergence.

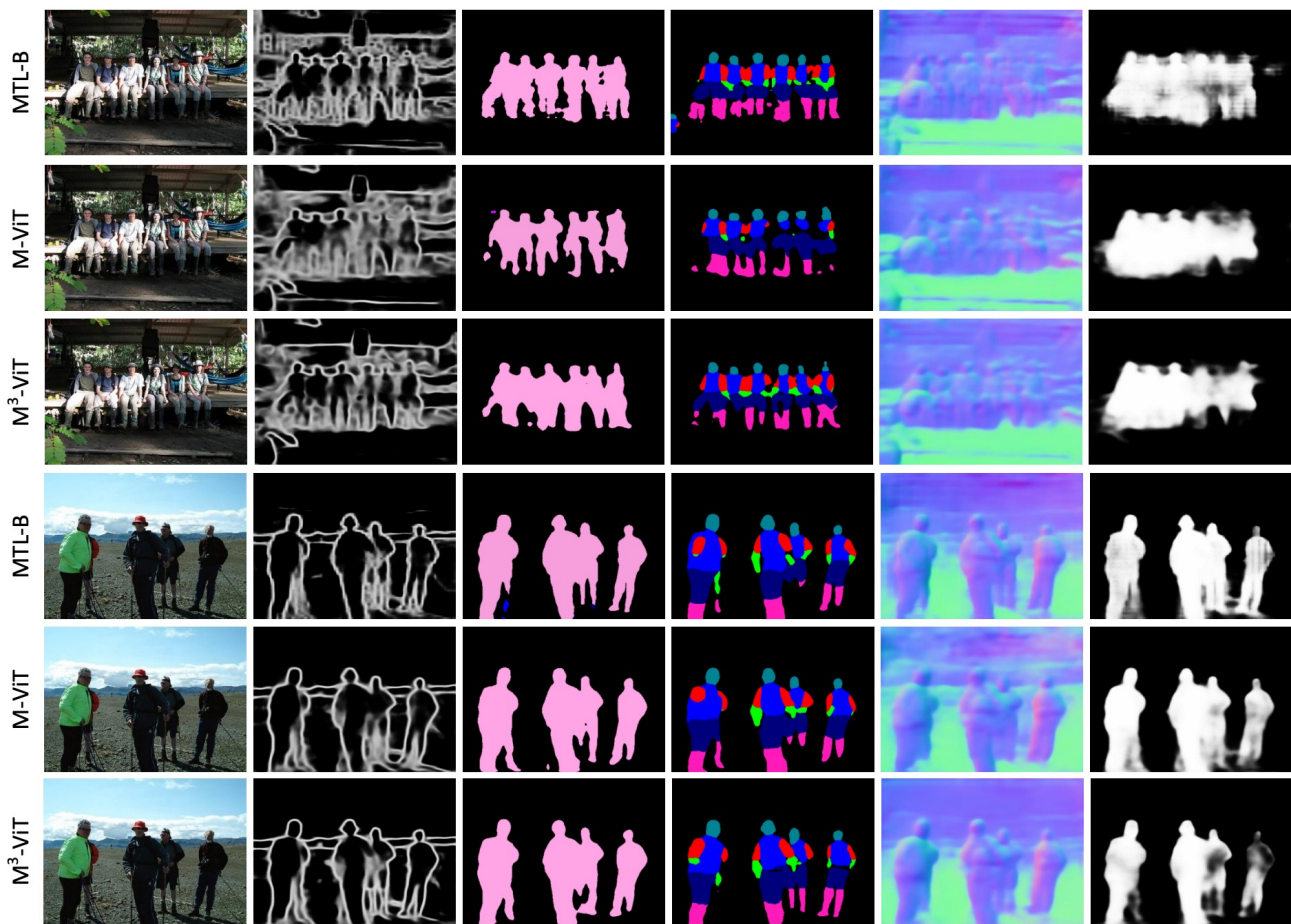
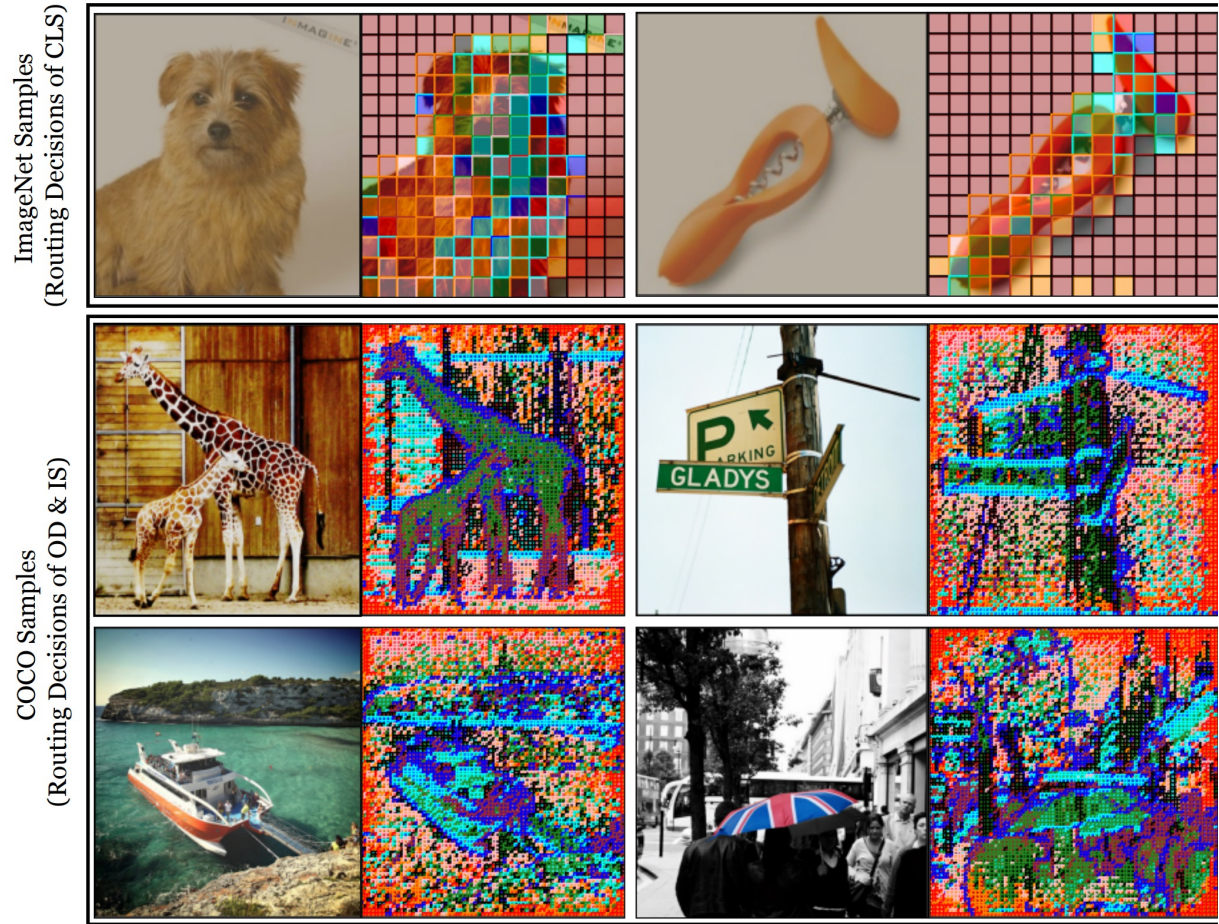
Adaptive Multi-Modal Multi-Task Sparse Mixture-of-Experts



Multi-Modal Multi-Task MoE

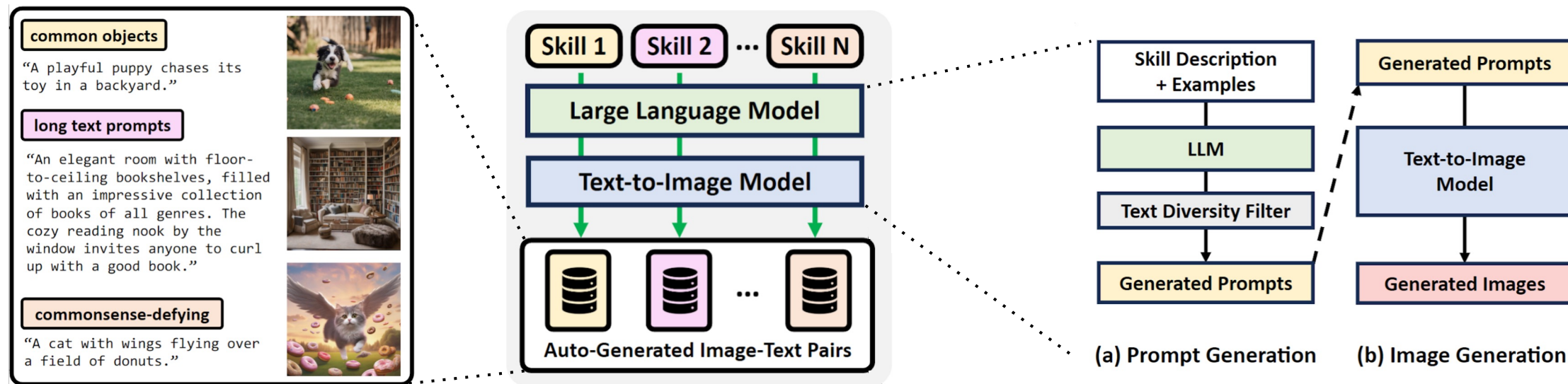


Multi-Modal Multi-Task Capability – More



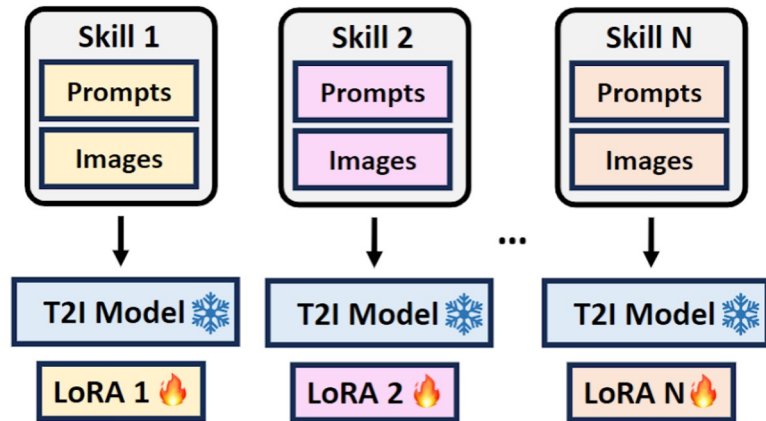
SELMA: Skill-Specific T2I Experts with Auto-Generated Data

- A novel paradigm to improve the faithfulness of T2I models by fine-tuning models on **automatically generated, "multi-skill" image-text datasets**, with **skill-specific expert learning and merging**.
- We first generate prompts to teach the skill with an LLM, while maintaining prompt diversity via text-similarity based filtering. We generate training images with a T2I model.

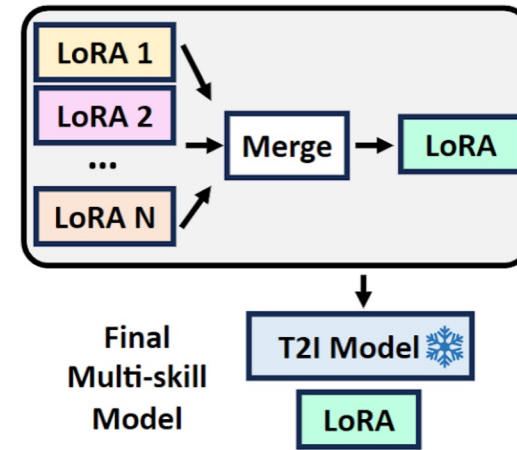


Skill-specific Expert Learning and Merging

- We learn **skill-specific expert** T2I models based on LoRA fine-tuning, and finally **merge the experts**.



(c) Skill-Specific Expert Learning



(d) Merging Expert Models

- Model merging can help mitigate the **knowledge conflicts** between datasets, and we only need to adjust the merging ratios without re-training the task-specific models.

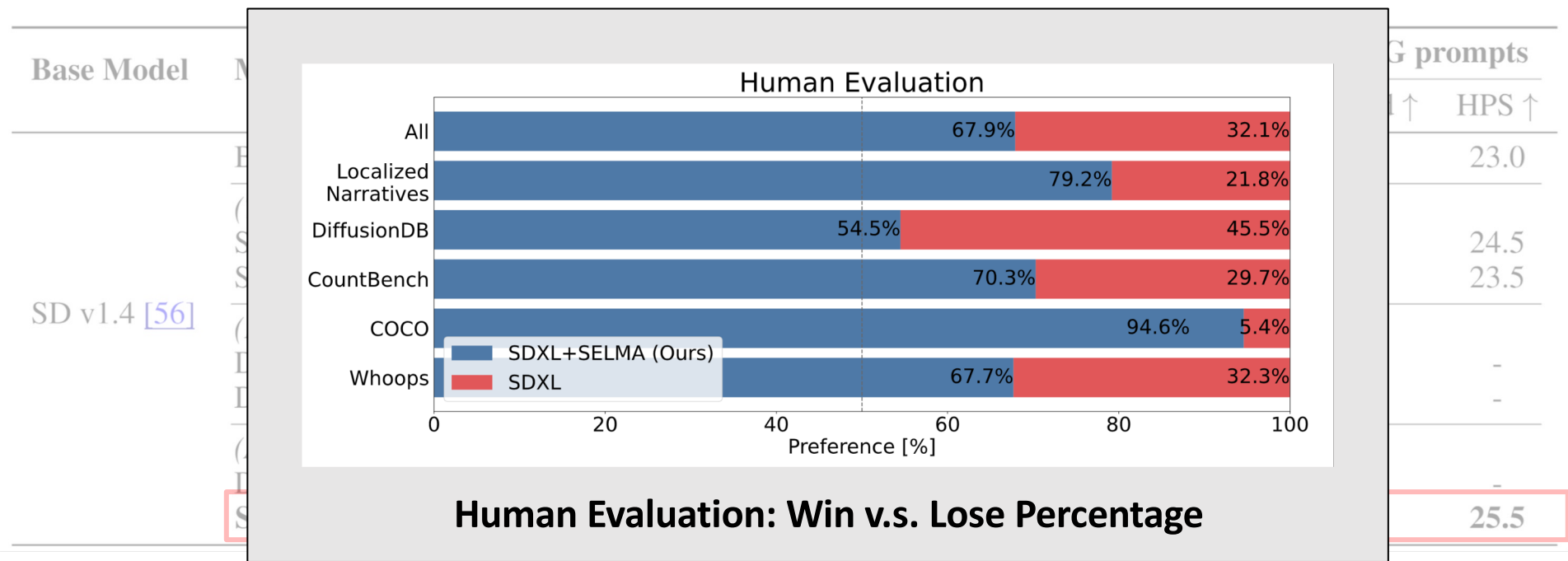
Quantitative Analysis

- We evaluate models on two evaluation benchmarks that measure the alignment between text prompts and generated images: DSG and TIFA. We measure text faithfulness with DSG and TIFA score, and human preference with PickScore, ImageReward and HPS.

Base Model	Methods	Text Faithfulness		Human Preference on DSG prompts		
		DSG ^{mPLUG} ↑	TIFA ^{BLIP2} ↑	PickScore ↑	ImageReward ↑	HPS ↑
SD v1.4 [56]	Base model	67.3	76.6	20.3	-0.22	23.0
	(Training-free)					
	SynGen [55]	66.2	76.8	20.4	-0.24	24.5
	StructureDiffusion [20]	67.1	76.5	20.3	-0.14	23.5
	(RL)					
	DPOK [19]	-	76.4	-	-0.26	-
	DDPO [5]	-	76.7	-	-0.08	-
	(Automatic data generation)					
	DreamSync [66]	-	77.6	-	-0.05	-
	SELMA (Ours)	71.3	79.5	20.5	0.36	25.5

Quantitative Analysis

- We evaluate models on two evaluation benchmarks that measure the alignment between text prompts and generated images: DSG and TIFA. We measure text faithfulness with DSG and TIFA score, and human preference with PickScore, ImageReward and HPS.

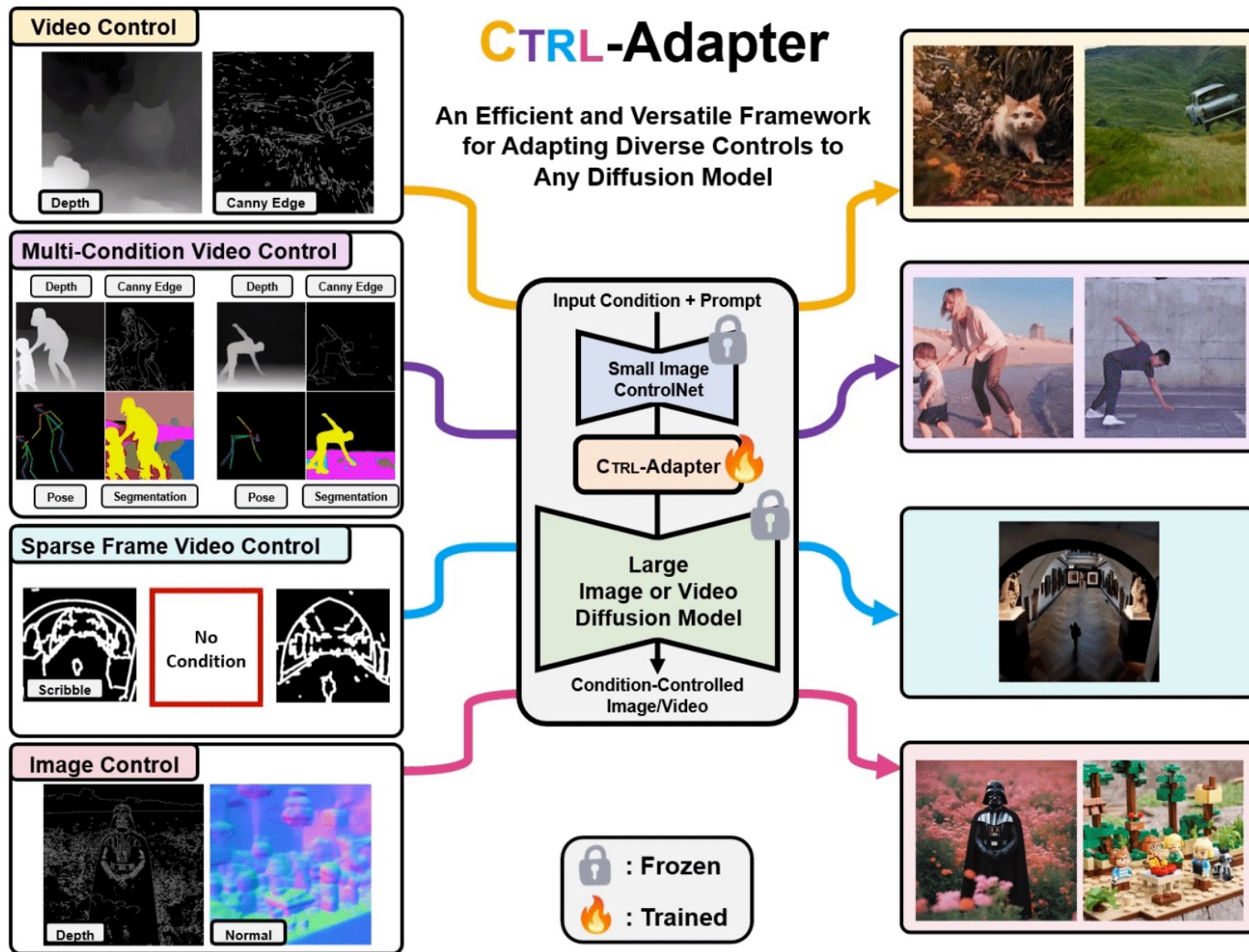


Qualitative Analysis

- We find that T2I model struggles with accommodating distinct skills and writing styles from different datasets, and **merging LoRA experts can help mitigate the knowledge conflict** between multiple skills.
- We find that a strong T2I model benefits from learning from images generated with a weaker T2I model, suggesting potential **weak-to-strong generalization**.

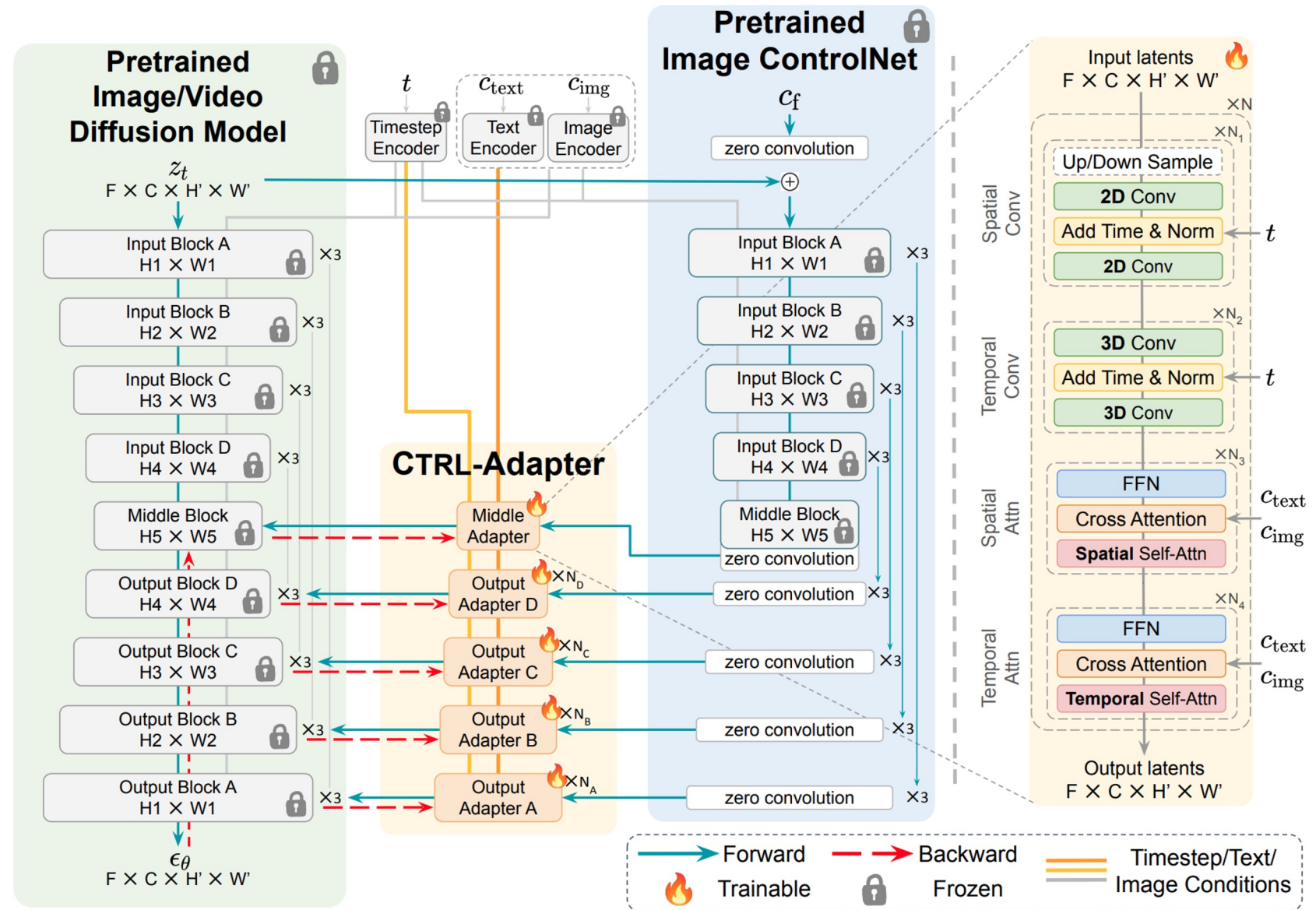


CTRL-Adapter: Efficient+Versatile Adaptation of Any Control to Any Diffusion



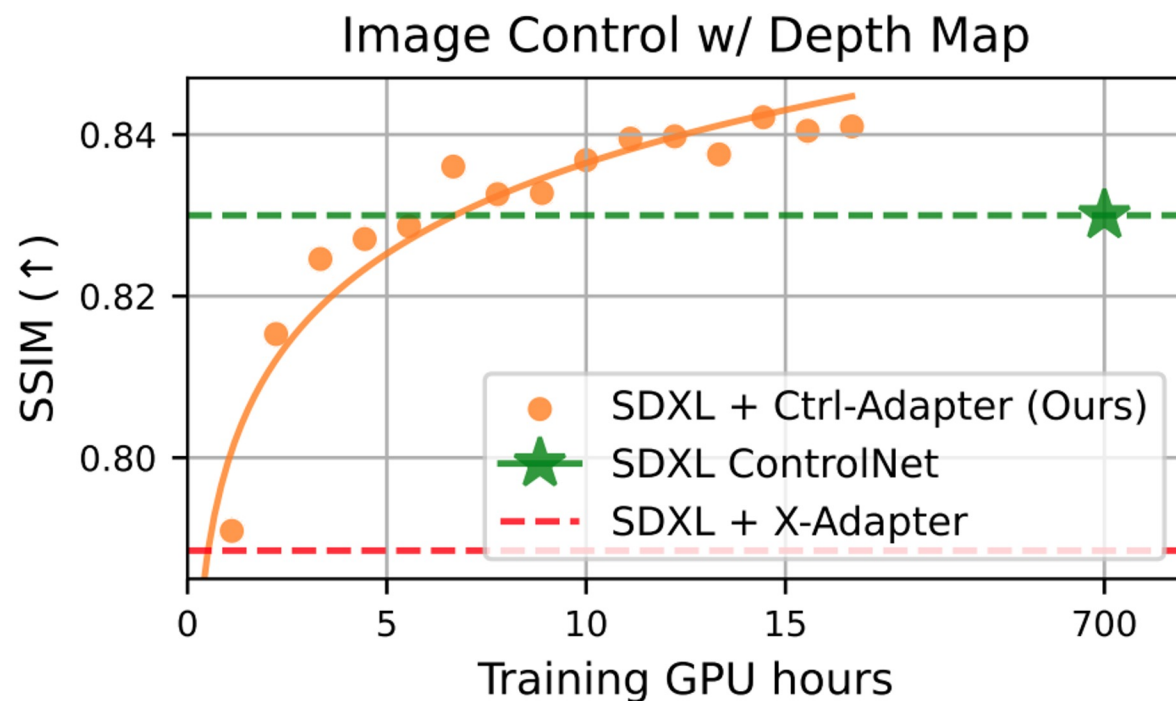
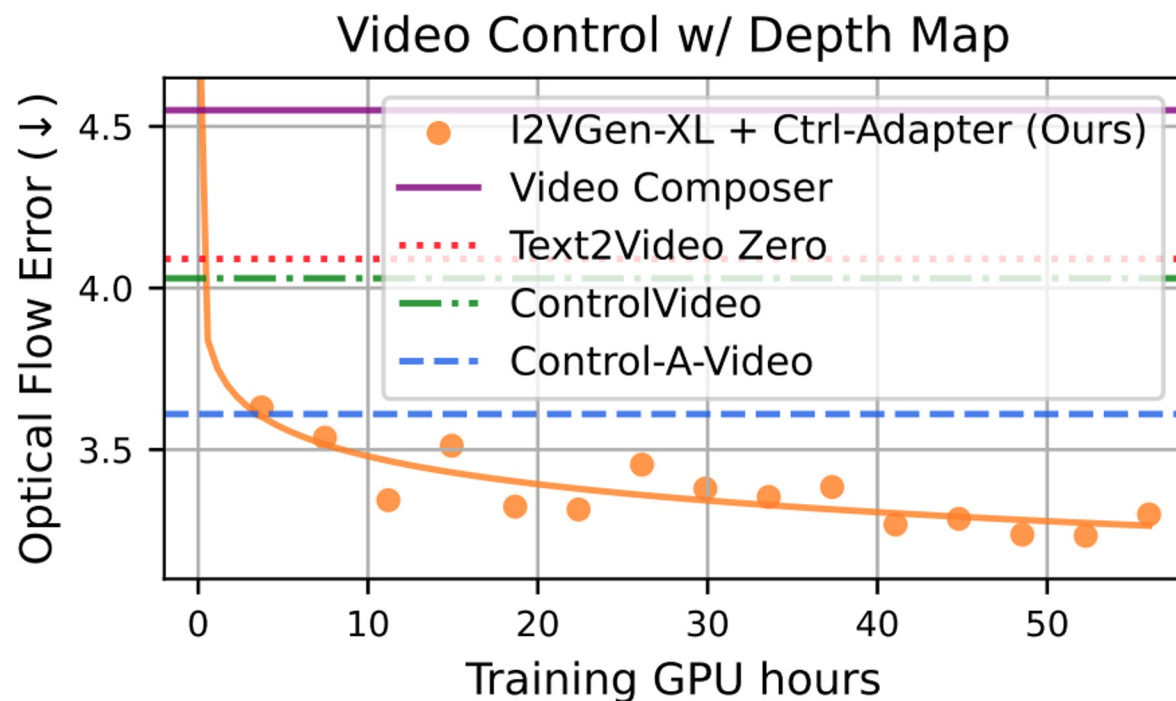
CTRL-Adapter: Efficient+Versatile Adaptation of Any Control to Any Diffusion

- Ctrl-Adapter (colored orange) enables to reuse pretrained image ControlNets (colored blue) for new image/video diffusion models (colored green)
- The temporal convolution and attention modules effectively fuse the ControlNet features to the video backbone models for better temporal consistency



Ctrl-Adapter: Matching SoTA Video/Image Control Methods in < 10 GPU hours

- Ctrl-Adapter matches the performance of pretrained ControlNets on COCO and achieves the state-of-the-art on DAVIS 2017 with significantly low computation (< 10 GPU hours)



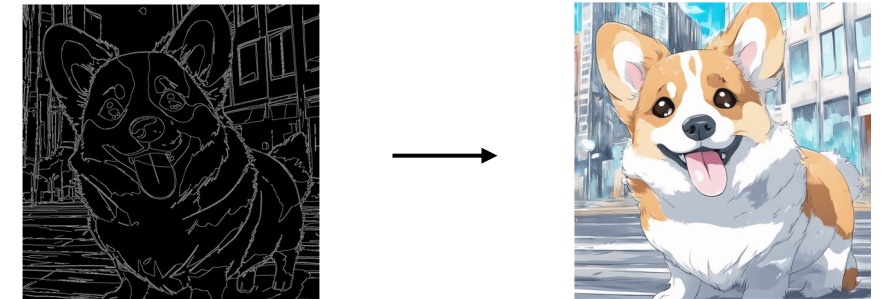
Ctrl-Adapter: Video / Image Control Examples

- Video and image control examples of Ctrl-Adapter with different types of conditions, such as depth, canny edge, and user scribbles

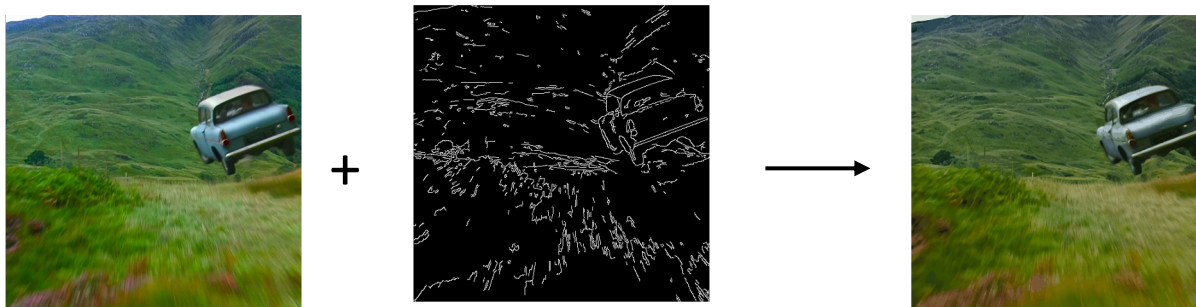
"A fish swimming"



"Cute fluffy corgi dog in the city in anime style"



"A car flies over a hill"



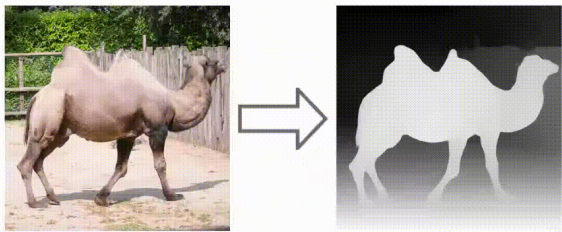
"Darth Vader in a beautiful field of flowers, colorful flowers everywhere, perfect lighting"



Ctrl-Adapter: Diverse control capabilities – Video Editing

- Video editing can be achieved by combining image/video Ctrl-Adapters with **user edited prompts**

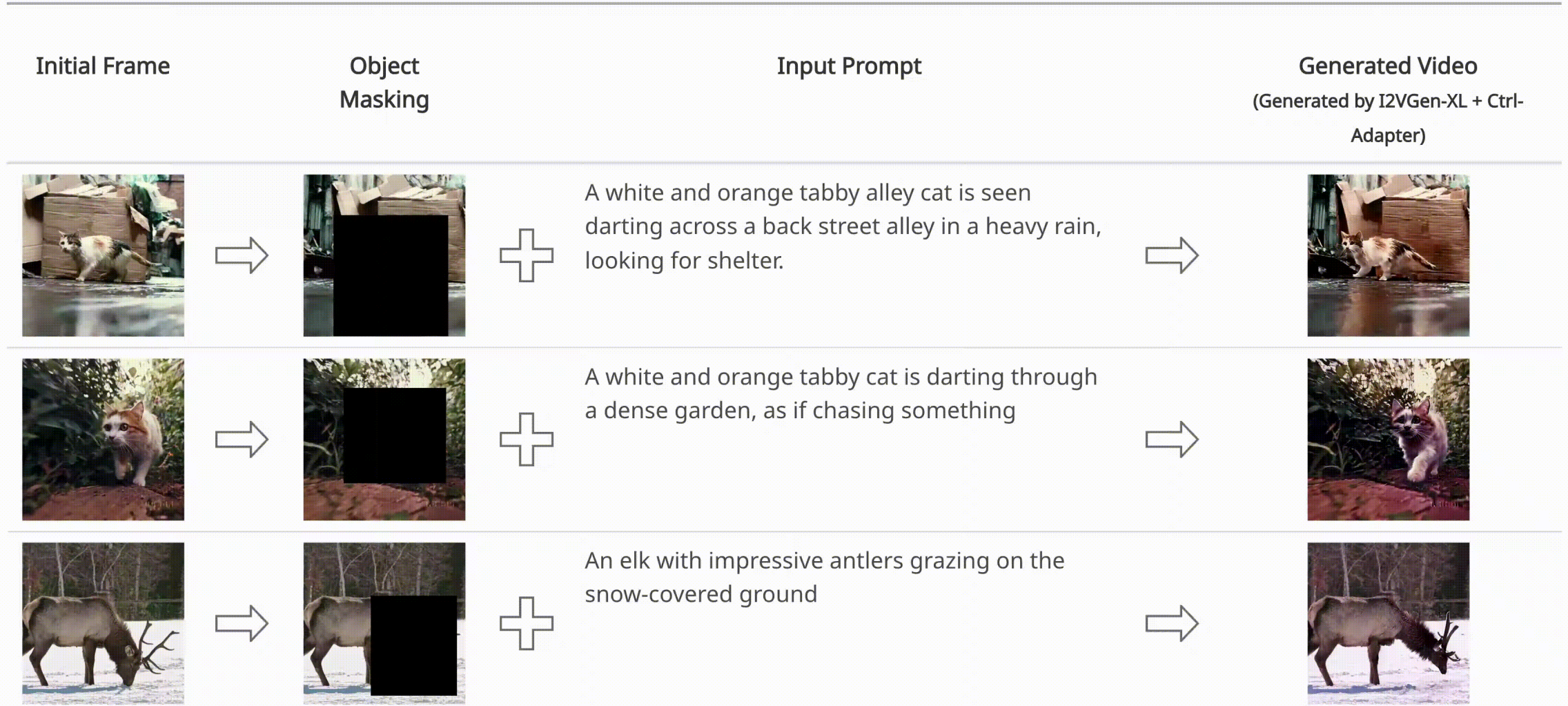
(1) Control Condition Extraction



Input Prompt		(2) Generated Frame (Generated by SDXL + Ctrl-Adapter)	(3) Generated Video (Generated by I2VGen-XL + Ctrl-Adapter)
+	A camel with rainbow fur walking.		
	A zebra striped camel walking.		
	A camel walking, ink sketch style.		
	A camel walking, van gogh-style.		


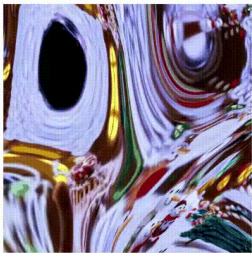


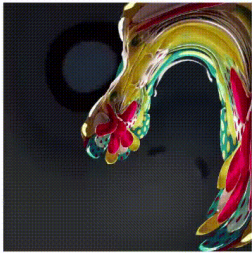
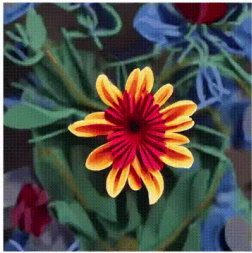

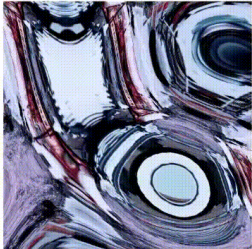

Ctrl-Adapter: Diverse control capabilities – Text-Guided Motion Control

- Video style transfer can be achieved by combining Ctrl-Adapters with **inpainting ControlNet**



Ctrl-Adapter: Diverse control capabilities – Video Style Transfer

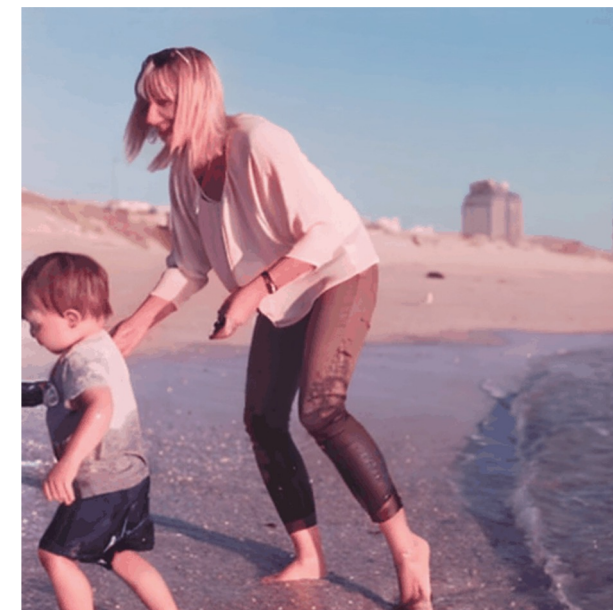
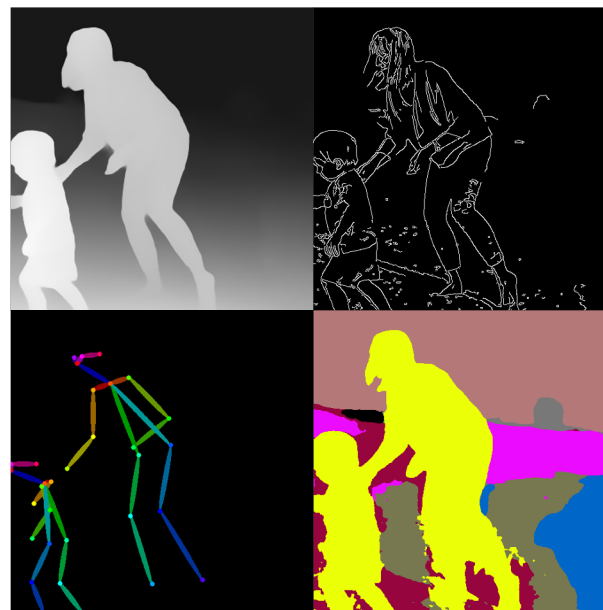
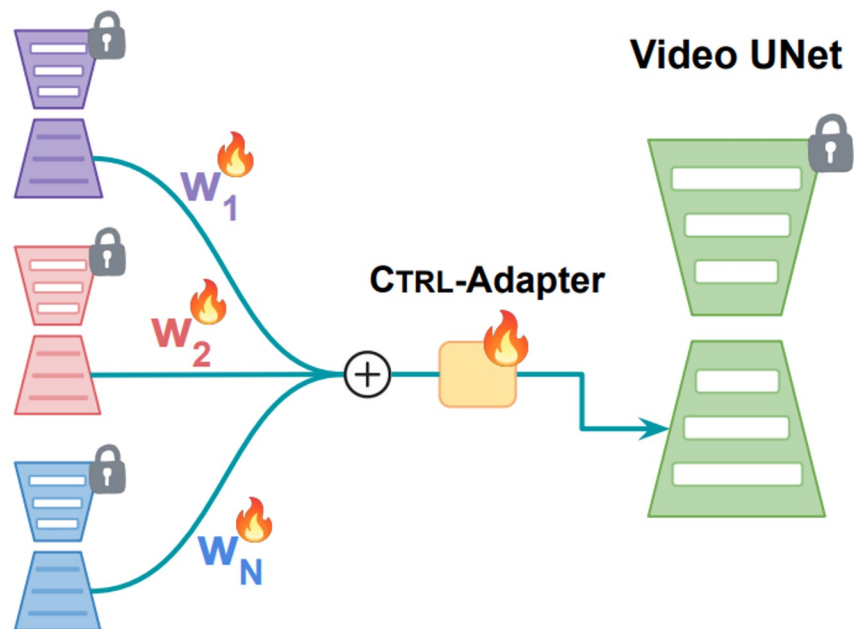
- Video editing can be achieved by combining Ctrl-Adapters with **shuffle ControlNet**

Initial Frame	Shuffled	Input Prompt	Generated Video (Generated by I2VGen-XL + Ctrl-Adapter)
		+ A miniature Christmas village with snow-covered houses, glowing windows, decorated trees, festive snowmen, and tiny figurines in a quaint, holiday-themed diorama evoking a cozy, celebratory winter atmosphere	
		+ Stop motion of a colorful paper flower blooming	
		+ Beautiful, snowy Tokyo city is bustling	

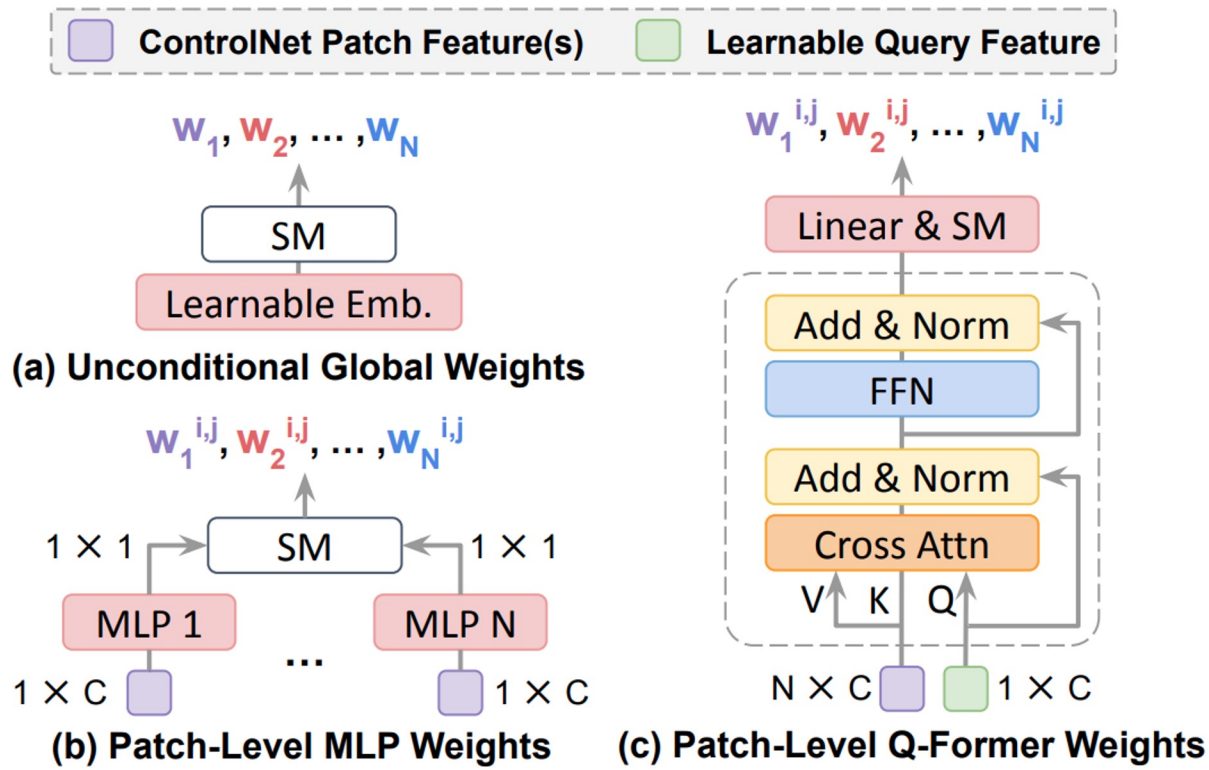
Ctrl-Adapter: Combining Multiple ControlNets with MoE Router

- To achieve more accurate spatial control, we can easily combine the control features of multiple ControlNets via Ctrl-Adapter
- We learn **MoE router** to learn weights to combine multiple ControlNet outputs

ControlNets



Ctrl-Adapter: Combining Multiple ControlNets with MoE Router



- We propose a light-weight **Patch-level MoE router** to learn the weights to combine the output features from multiple ControlNets
- **Patch-level MoE router is better** than using equal weights / learning unconditional weights

Patch-level MoE routers

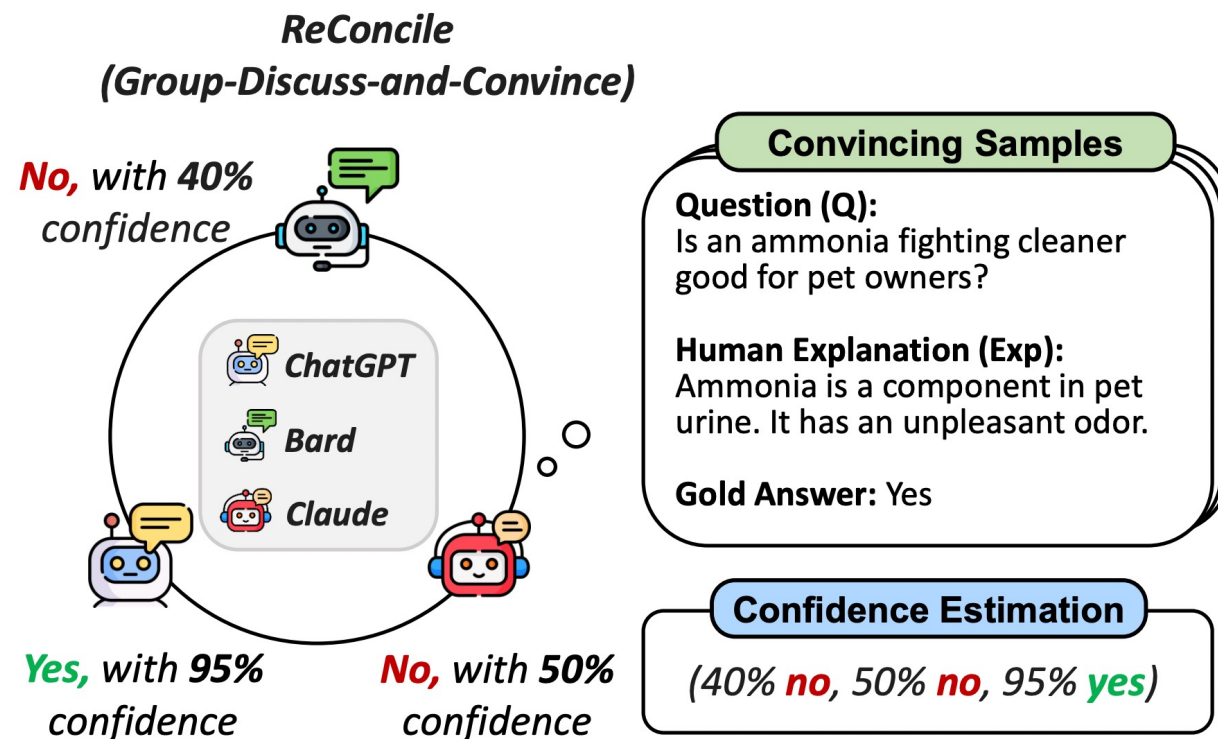
	D+C		D+P		D+C+N+S		D+C+N+S+Seg+L+P	
	FID (↓)	Flow Error (↓)	FID (↓)	Flow Error (↓)	FID (↓)	Flow Error (↓)	FID (↓)	Flow Error (↓)
Baseline: Equal Weights	8.50	2.84	11.32	3.48	8.75	2.40	9.48	2.93
(a) Unconditional Global Weights	9.14	2.89	10.98	3.32	8.39	2.36	8.18	2.48
(b) Patch-Level MLP Weights	8.40	2.34	9.37	3.17	7.87	2.11	8.26	2.00
(c) Patch-Level Q-Former Weights	7.54	2.39	9.22	3.22	7.72	2.31	8.00	2.08

Discussion and Collaboration based Mixture of Agents

- LLMs struggle with complex reasoning!
- Mixing multiple expert LLMs ‘interactively’ → improve reasoning of each!
- ReConcile: A discussion-based multi-agent mixture framework

- Key components:

- Multi-LLM discussion via explanations
- Multiple discussion rounds
- Correctively-convincing other agents
- Confidence-weighted voting



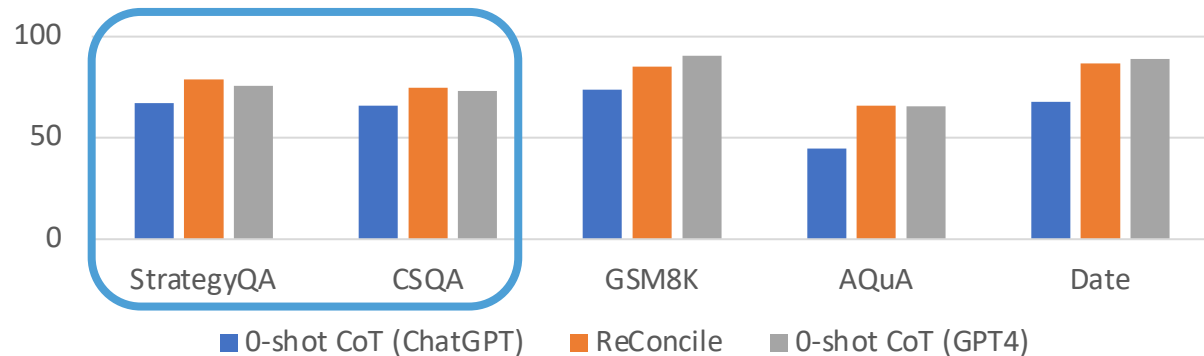
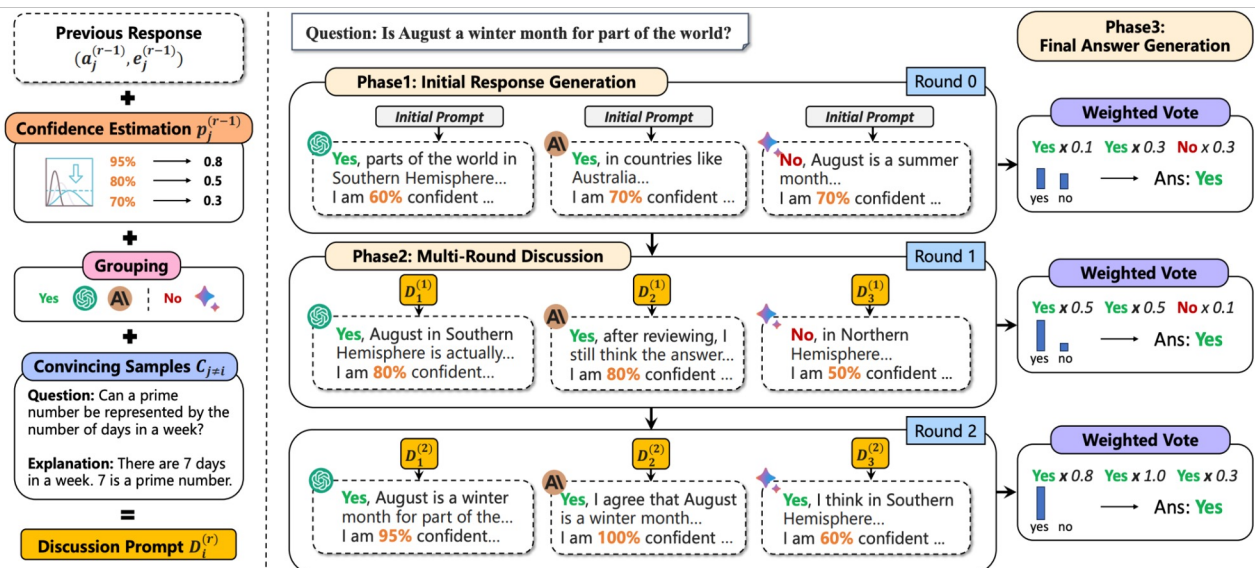
ReConcile

- Discussion across 3 phases:
 1. Initial response/explanation
 2. Multi-round discussion
 3. Final answer generation
- ReConcile w/o GPT4 outperforms it!
- ReConcile w/ GPT4 outperforms it by 10%!

Method Category	Method	Agent	StrategyQA	CSQA	GSM8K	AQuA	Date
Vanilla Single-agent	Zero-shot CoT	GPT-4	75.6 \pm 4.7	73.3 \pm 0.4	90.7 \pm 1.7	65.7 \pm 4.6	89.0 \pm 2.2
	Zero-shot CoT	ChatGPT	67.3 \pm 3.6	66.0 \pm 1.8	73.7 \pm 3.1	44.7 \pm 0.5	67.7 \pm 1.2
	Zero-shot CoT	Bard	69.3 \pm 4.4	56.8 \pm 2.7	58.7 \pm 2.6	33.7 \pm 1.2	50.2 \pm 2.2
	Zero-shot CoT	Claude2	73.7 \pm 3.1	66.7 \pm 2.1	79.3 \pm 3.6	60.3 \pm 1.2	78.7 \pm 2.1
Advanced Single-agent	Self-Refine (SR)	ChatGPT	66.7 \pm 2.7	68.1 \pm 1.8	74.3 \pm 2.5	45.3 \pm 2.2	66.3 \pm 2.1
	Self-Consistency (SC)	ChatGPT	73.3 \pm 2.1	70.9 \pm 1.3	80.7 \pm 1.2	54.0 \pm 2.9	69.0 \pm 0.8
	SR + SC	ChatGPT	72.2 \pm 1.9	71.9 \pm 2.1	81.3 \pm 1.7	58.3 \pm 3.7	68.7 \pm 1.2
Single-model Multi-agent	Debate	$\times 3$	66.7 \pm 3.1	62.7 \pm 1.2	83.0 \pm 2.2	65.3 \pm 3.1	68.0 \pm 1.6
	Debate	$\times 3$	65.3 \pm 2.5	66.3 \pm 2.1	56.3 \pm 1.2	29.3 \pm 4.2	46.0 \pm 2.2
	Debate	$\times 3$	71.3 \pm 2.2	68.3 \pm 1.7	70.7 \pm 4.8	62.7 \pm 2.6	75.3 \pm 3.3
	Debate+Judge	$\times 3$	69.7 \pm 2.1	63.7 \pm 2.5	74.3 \pm 2.9	57.3 \pm 2.1	67.7 \pm 0.5
Multi-model Multi-agent	RECONCILE		79.0 \pm 1.6	74.7 \pm 0.4	85.3 \pm 2.2	66.0 \pm 0.8	86.7 \pm 1.2

Most powerful / expensive model considered

ReConcile w/o GPT4 outperforms it!



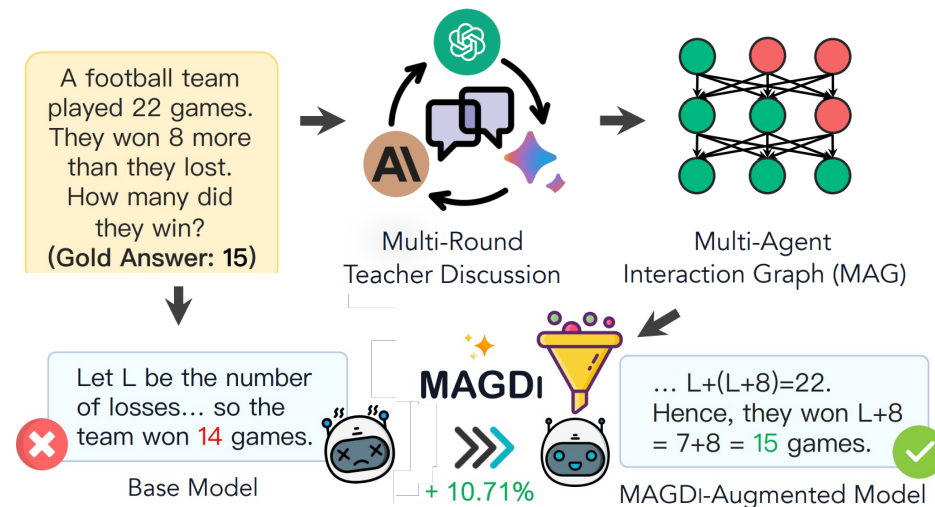
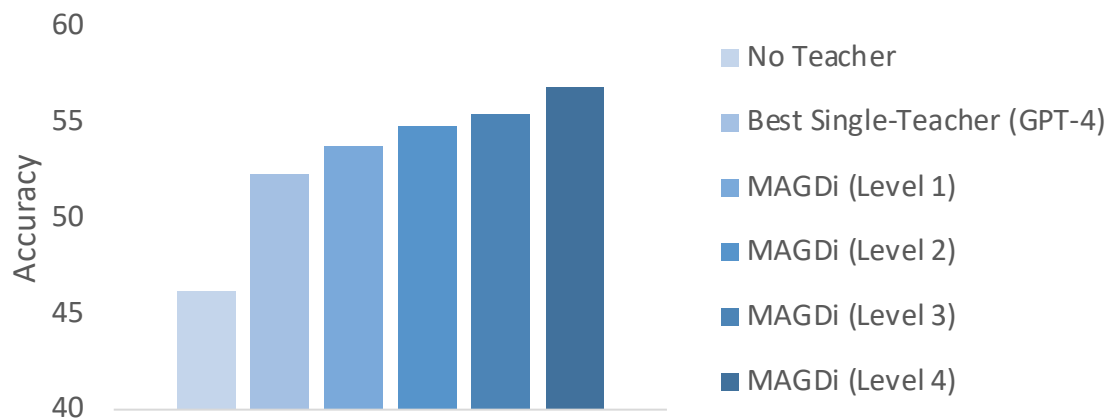
Distill Multi-Agent Mixture+Interaction into Single Model

Strong performance boost but multiple LLMs across multiple rounds is expensive!

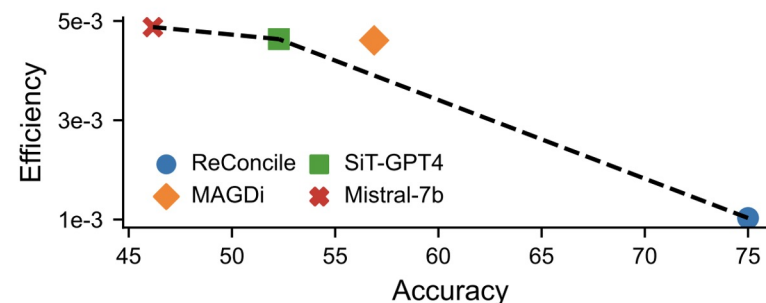
Structured distillation from graph

4 levels

Improvements across StrategyQA, CSQA, ARC, GSM8K, MATH



Best tradeoff between performance and efficiency!



MAGDi: Structured Distillation of Multi-Agent Interaction Graphs Improves Reasoning in Smaller Language Models, Justin Chih-Yao Chen*, Swarnadeep Saha*, Elias Stengel-Eskin Mohit Bansal (ICML 2024)