Paper Review : 2024 Fall Lab Seminar

LLaMA PRO vs. SOLAR 10.7B

Wu et al. (Tencent) LLaMA PRO : Progressive LLaMA with Block Expansion
(ACL2024 Findings)

Kim et al. (Upstage AI) SOLAR 10.7B: Scaling Language Models with Simple yet Effective Depth Up-Scaling
(NAACL2024 Industry Track)

Yejin Yoon

Natural Language Processing Lab.,
Hanyang University.

HYU 한양대학교 HANYANG UNIVERSITY

Paper Review : 2024 Fall Lab Seminar

Block Expansion LLaMA PRO vs. SOLAR 10.7B Depth Up-Scaling

Wu et al. (Tencent) LLaMA PRO : Progressive LLaMA with Block Expansion
(ACL2024 Findings)

Kim et al. (Upstage AI) SOLAR 10.7B: Scaling Language Models with Simple yet Effective Depth Up-Scaling
(NAACL2024 Industry Track)

Yejin Yoon

HYU 한양대학교 HANYANG UNIVERSITY

Natural Language Processing Lab.,
Hanyang University.

PRE-REQUISITE

Task-specific Training vs. (General) LLM Training

Parameter-efficient Fine-tuning vs. Task-specialization

~~# Overoptimization Recap.~~

Task-specific Training vs. General LLM Training

- **Goals**
  - Task-specific Training: Focused on enhancing performance for specific tasks.
  - LLM Training: Designed for general performance improvement across a wide range of tasks.

[Task-specific Training]

Pretrining → **Fine-tuning**

Pretrining: Initial large-scale training of the model on vast amounts of generic data.

Fine-tuning: The model is fine-tuned on task-specific datasets to improve its performance.

PEFT

A specialized method that fine-tunes only a smaller subset of parameters.
→ reduce the computational cost & resource requirements

[General LLM Training]

Pretrining → **Instruction Tuning** → **Alignment Tuning**

Pretrining: Similar to task-specific training, pretraining serves as the foundation where the model learns broad patterns in data.

Instruction Tuning: The model is trained to follow human instructions better, improving its ability to understand and respond to user queries.

Alignment Tuning: Fine-tuning to align the model's outputs with human values or ethical guidelines. This helps in reducing biases and harmful outputs, making the model safer and more reliable.

sLLM

Make the model more efficient in terms of memory usage & computational cost
→ more practical for deployment in resource-constrained environments

Task-specific Training vs. General LLM Training

- **Goals**
  - Task-specific Training: Focused on enhancing performance for specific tasks.
  - LLM Training: Designed for general performance improvement across a wide range of tasks.

[Task-specific Training] *The model might lose some general capability!*

Pretrining → **Fine-tuning**

Initial large-scale training of the model on vast amounts of generic data.

Fine-tuning allows for performance improvement on specific tasks, such as better accuracy, relevance, or efficiency for your domain, while still respecting human-aligned behavior.

PEFT

A specialized method that fine-tunes only a smaller subset of parameters.
→ reduce the computational cost & resource requirements

[General LLM Training]

Pretrining → **Instruction Tuning** → **Alignment Tuning**

Similar to task-specific training, pretraining serves as the foundation where the model learns broad patterns in data.

The model is trained to follow human instructions better, improving its ability to understand and respond to user queries.

Fine-tuning to align the model's outputs with human values or ethical guidelines. This helps in reducing biases and harmful outputs, making the model safer and more reliable.

sLLM

Make the model more efficient in terms of memory usage & computational cost
→ more practical for deployment in resource-constrained environments

Task-specific Training vs. General LLM Training

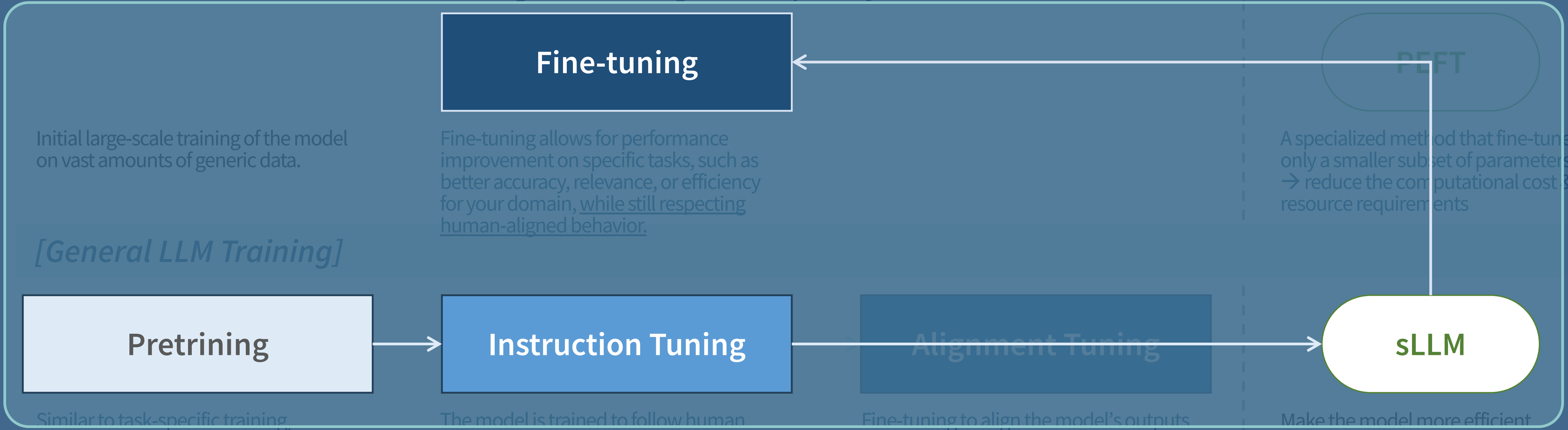
Goals

Problem States:

How to improve specific task(s) performance while maintaining general model performance

- LLM Training: Designed for general performance improvement across a wide range of tasks.

LLaMA PRO Block Expansion



Initial large-scale training of the model on vast amounts of generic data.

Fine-tuning allows for performance improvement on specific tasks, such as better accuracy, relevance, or efficiency for your domain, while still respecting human-aligned behavior.

A specialized method that fine-tunes only a smaller subset of parameters.
→ reduce the computational cost & resource requirements

[General LLM Training]

Similar to task-specific training, pretraining serves as the foundation where the model learns broad patterns in data.

The model is trained to follow human instructions better, improving its ability to understand and respond to user queries.

Fine-tuning to align the model's outputs with human values or ethical guidelines. This helps in reducing biases and harmful outputs, making the model safer and more reliable.

Make the model more efficient in terms of memory usage & computational cost
→ more practical for deployment in resource-constrained environments

Task-specific Training vs. General LLM Training

Problem States:

How to improve specific task(s) performance while maintaining general model performance

- Goals
  - LLM Training: Designed for general performance improvement across a wide range of tasks.

[Task-specific Training] *The model might lose some general capability!*

Pretrining → Fine-tuning

PEFT

Initial large-scale training of the model on vast amounts of generic data.

Fine-tuning allows for performance improvement on specific tasks, such as better accuracy, relevance, or efficiency for your domain, while still respecting

SOLAR 10.7B Depth Up-Scaling

[General LLM Training]

Pretrining → Instruction Tuning → Alignment Tuning → sLLM

Continual Pretraining

Similar to task-specific training, pretraining serves as the foundation wh... in data.

The model is trained to follow human instructions better, improving its ability to understand and respond to user queries.

Fine-tuning to align the model's outputs with human values or ethical guidelines. This helps in reducing biases and harmful outputs, making the model safer and more reliable.

Make the model more efficient in terms of memory usage & computational cost → more practical for deployment in resource-constrained environments

Parameter-efficient Fine-tuning (PEFT) vs. Task-specialization

- **Goals**
  - PEFT: Reducing the number of trainable parameters while still enhancing task-specific performance.
  - Task-specialization: Achieving specialization for tasks by modifying either the entire model or parts of it (like output heads), but typically not as parameter-efficient.

[Parameter-efficient Fine-tuning]

LoRA	Adds low-rank matrices to fine-tune without updating the full model parameters
Adaptor Layers	Inserts small, task-specific layers into the model that are trained while the rest of the model remains frozen.
BitFit	Fine-tunes only the bias terms in the model's layers, keeping most of the parameters unchanged.
Prompt Tuning	Only tunes the input prompts without modifying the model weights.

[Task-specialization]

Full Fine-tuning	Adjusts all parameters of the model for a specific task.
Task-specific Heads	Keeps the core layers of the model unchanged while adding separate heads for different tasks. Each task-specific head is fine-tuned independently.
Multi-task Learning	Trains a model across multiple tasks at the same time, often using shared layers but with different heads or output layers for each task.

[MoE] Uses a gating network to activate only certain experts (sub-networks) based on the input, meaning not all parts of the model are used for every task.

[sLLM] Smaller, more efficient versions of large models, typically trained with fewer parameters while aiming to maintain strong performance.

Parameter-efficient Fine-tuning (PEFT) vs. Task-specialization

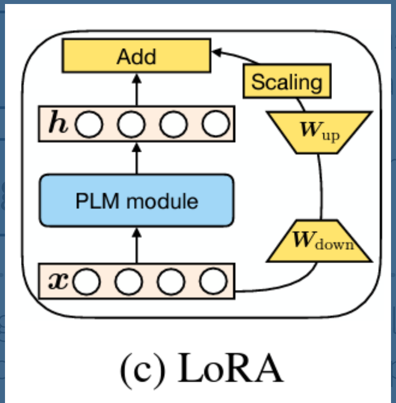
- He et al. (CMU, ICLR2022) “Towards a Unified View of Parameter-Efficient Transfer Learning”
- Houlsby et al. (Google, PMLR2019) “Parameter-Efficient Transfer Learning for NLP”

Goals
- PEFT: Reducing the number of trainable parameters while still enhancing task-specific performance.
- Task-specialization: Achieving specialization for tasks by modifying either the entire model or parts of it (like output heads), but typically not as parameter-efficien

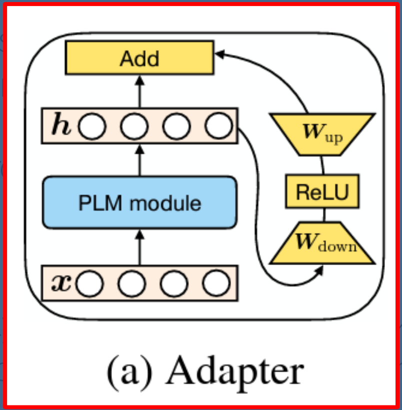
[Parameter-efficient Fine-tuning]

LoRA	Adds low-rank matrices to fine-tune without updating the full model parameters
Adaptor Layers	Inserts small, task-specific layers into the model that are trained while the rest of the model remains frozen.
BitFit	
Prompt Tunin	

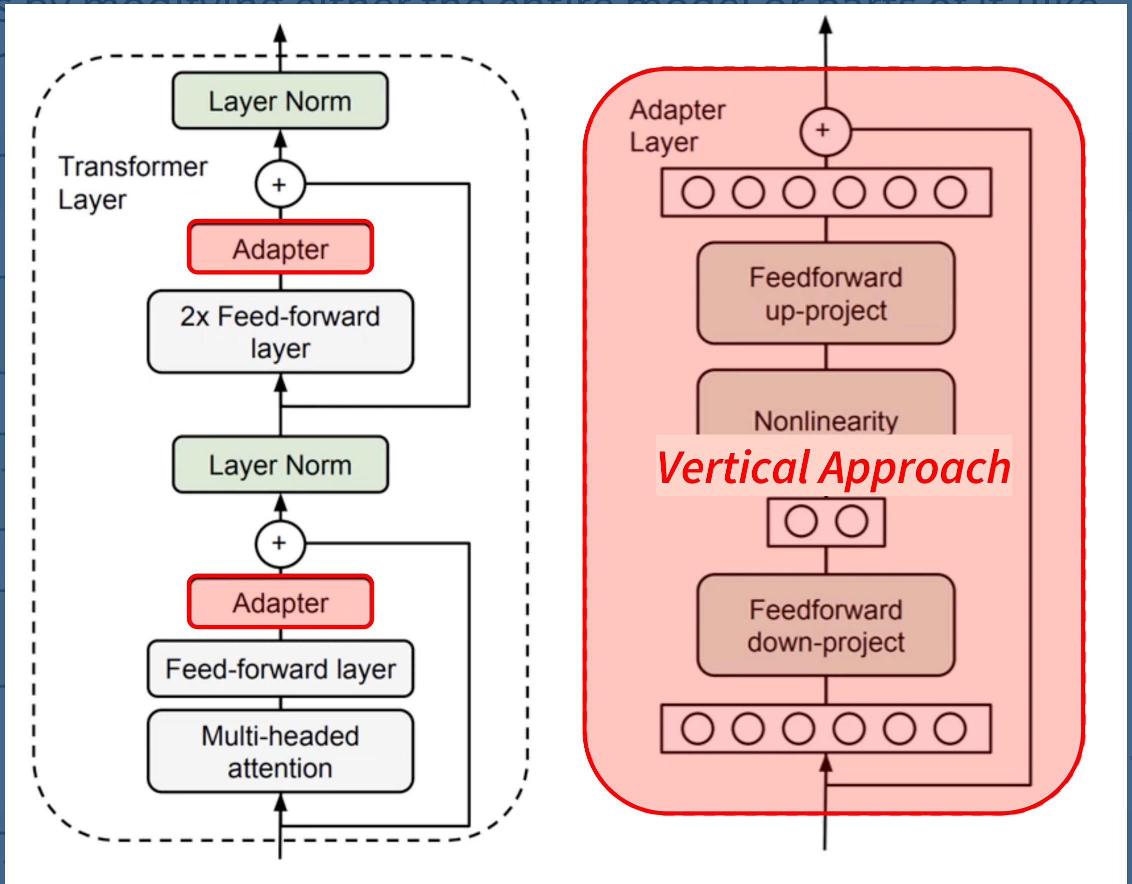
[MoE] Uses a g... based o... every task.



(c) LoRA



(a) Adapter



Shared Necessity :: LLaMA PRO & SOLAR 10.7B

A. ***Background***

Both approaches start with **the need for a more powerful sLLM**.

B. ***Goal***

(In fact) Both approaches aim
to maintain the general ability of the Initial Model
while additionally **improving specific tasks**.

C. ***Suggestion***

A vertical expansion approach **by increasing depth**.

LLaMA PRO vs. SOLAR 10.7B

Contents

1 Pre-requisite

- Task-specific Training vs. General LLM Training
- Parameter-efficient Fine-tuning (PEFT) vs. Task-specialization

2 LLaMA PRO

- Block Expansion

3 SOLAR 10.7B

- Pretraining: Depth Up-Scaling (DUS)
- Instruction Tuning: math synthetic datasets
- Alignment Learning: sDPO

4 Conclusion

Paper Review : 2024 Fall Lab Seminar

LLaMA PRO : Progressive LLaMA with Block Expansion

Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ying Shan, Ping Luo (Tencent PCG)

Accepted to ACL2024 Findings

Yejin Yoon

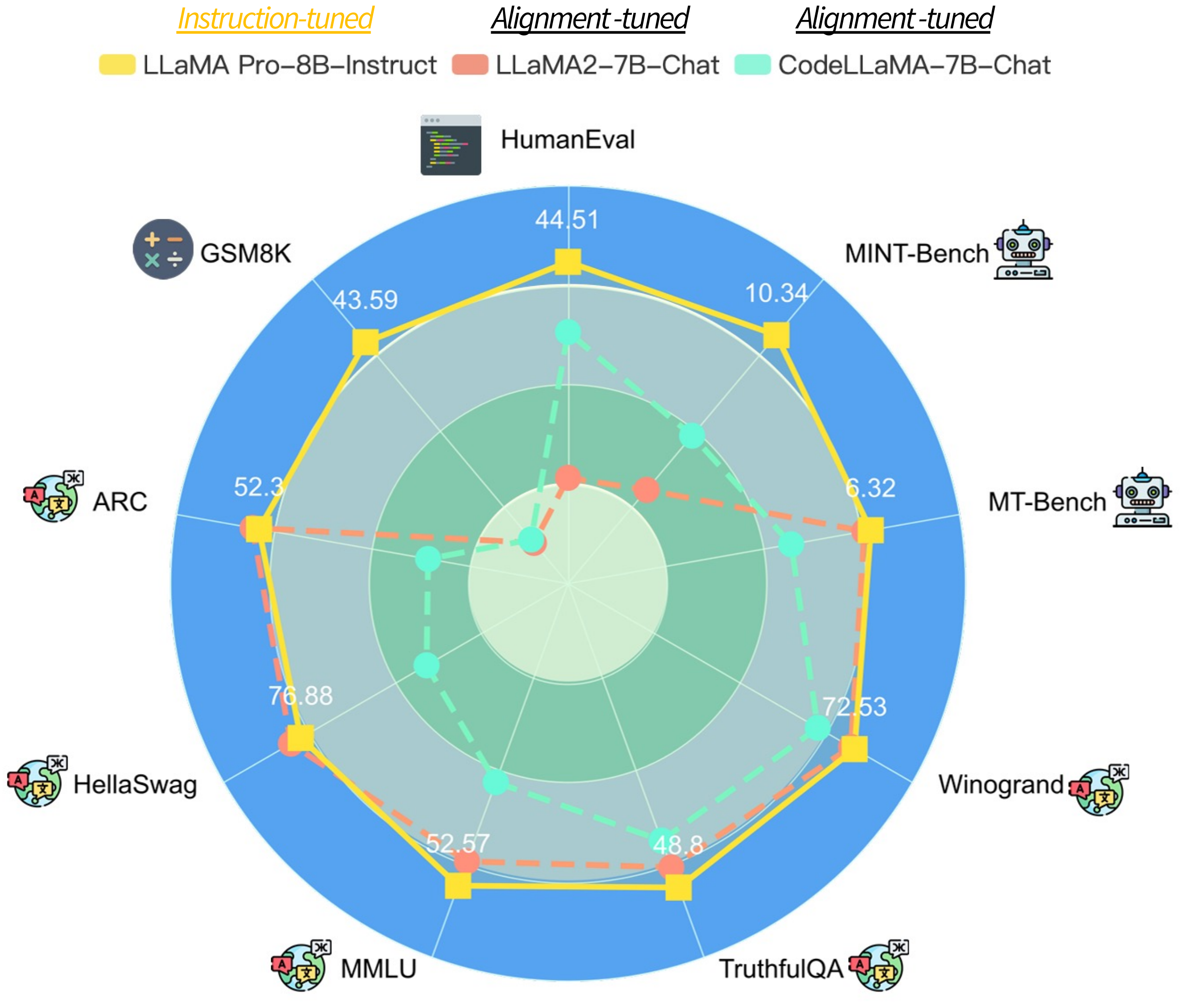
HYU 한양대학교 HANYANG UNIVERSITY

Natural Language Processing Lab.,
Hanyang University.

Problem States

- **Background**

  - Llama 7B is a large language model (LLM) that performs well across various tasks.
  - However, new learning can lead to **catastrophic forgetting** of old knowledge.
  - The challenge is enabling **continual learning** that retains previous knowledge while learning new information.



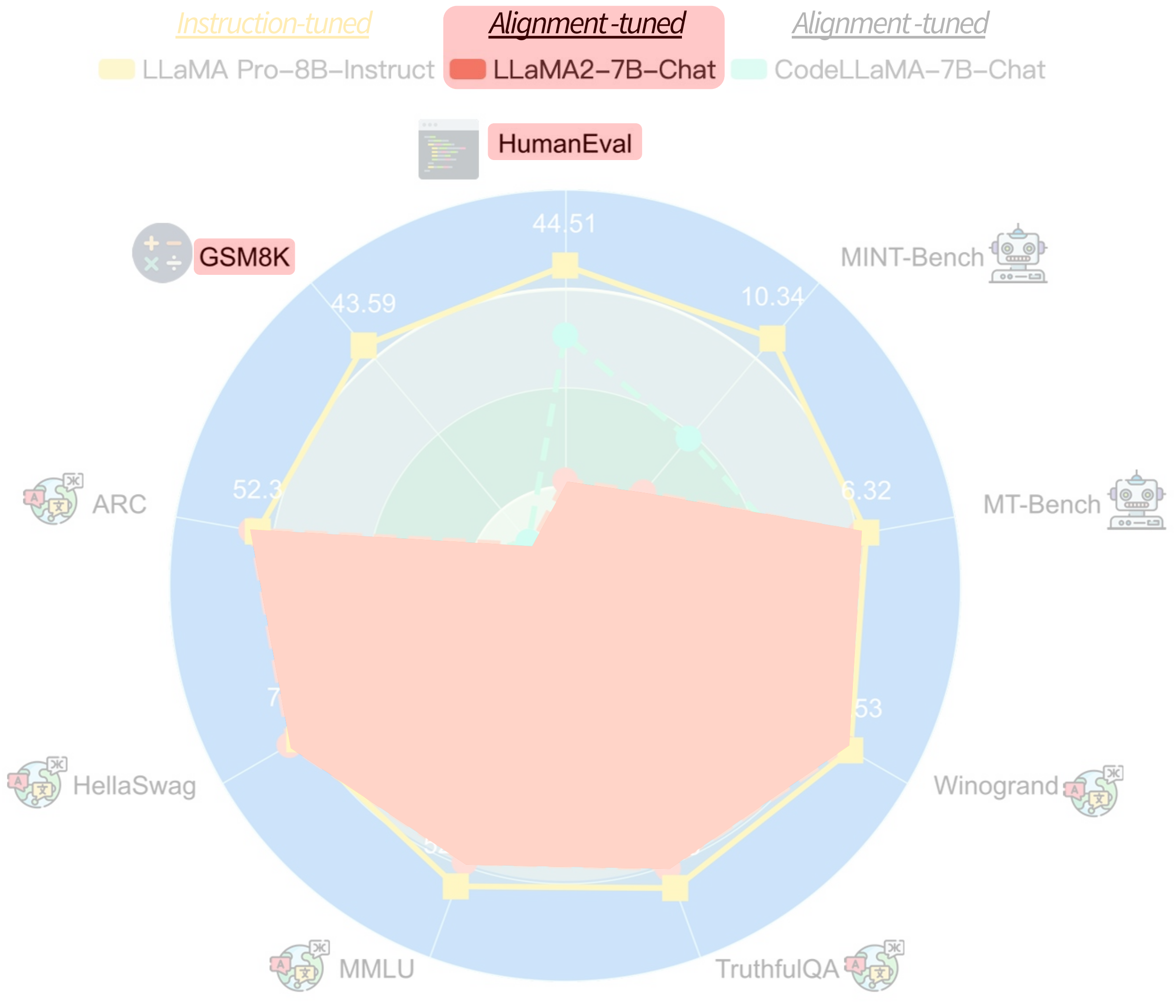
Problem States

- **Background**

  - Llama 7B is a large language model (LLM) that performs well across various tasks.

  - However, new learning can lead to **catastrophic forgetting** of old knowledge.

  - The challenge is enabling **continual learning** that retains previous knowledge while learning new information.



Preventing Catastrophic Forgetting in LLM

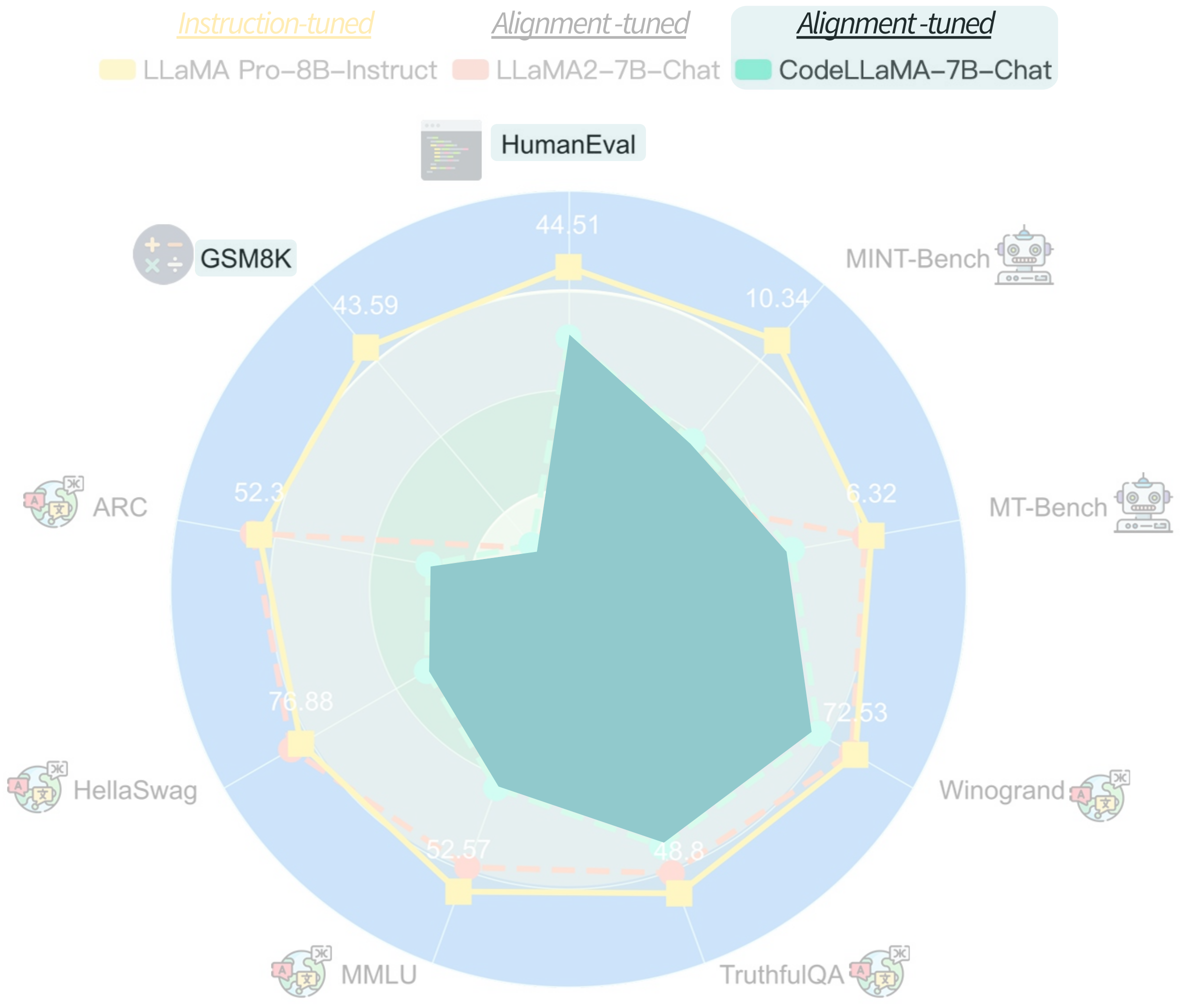
Problem States

- **Background**

  - Llama 7B is a large language model (LLM) that performs well across various tasks.

  - However, new learning can lead to **catastrophic forgetting** of old knowledge.

  - The challenge is enabling **continual learning** that retains previous knowledge while learning new information.

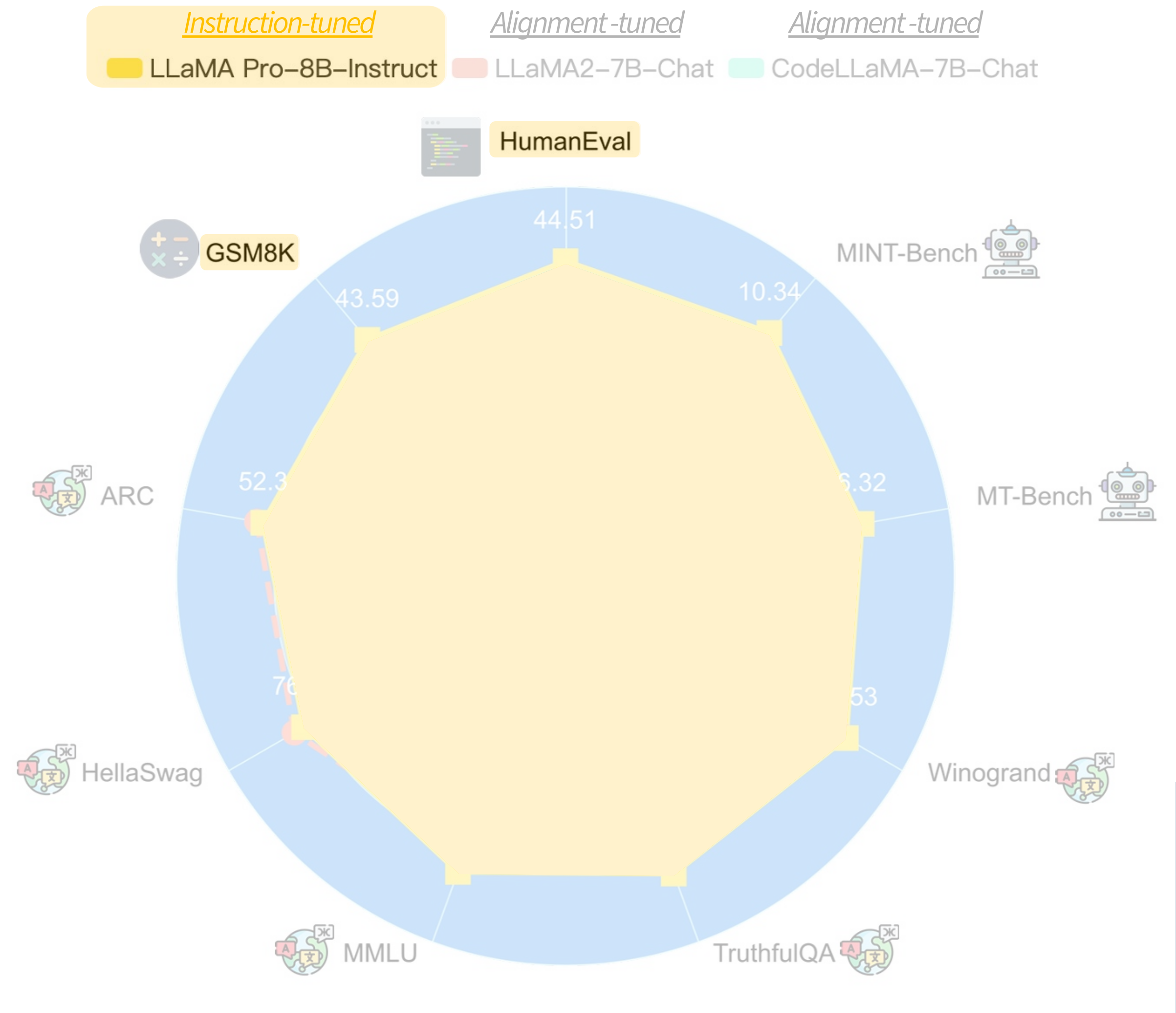


Preventing Catastrophic Forgetting in LLM

Problem States

- **Background**

  - Llama 7B is a large language model (LLM) that performs well across various tasks.
  - However, new learning can lead to **catastrophic forgetting** of old knowledge.
  - The challenge is enabling **continual learning** that retains previous knowledge while learning new information.

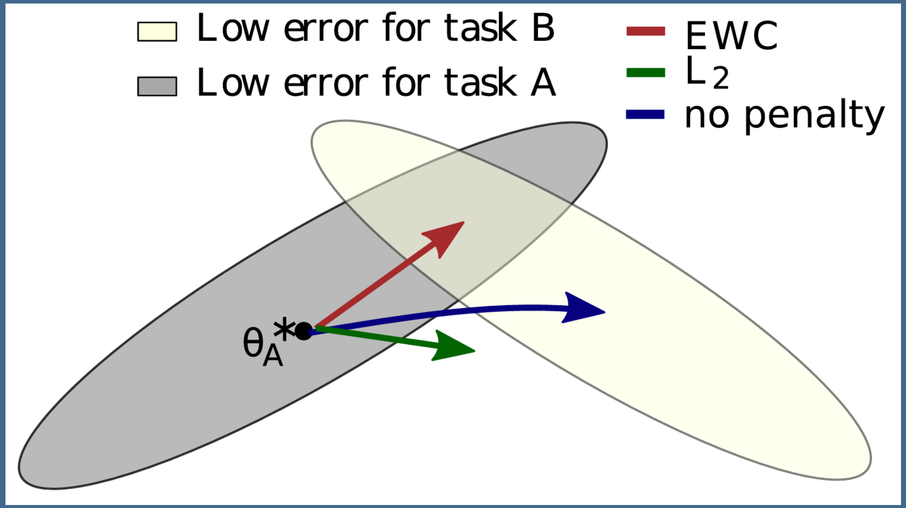


With the proposed Block Expansion, the model enables continual learning, retaining previous knowledge.

Background

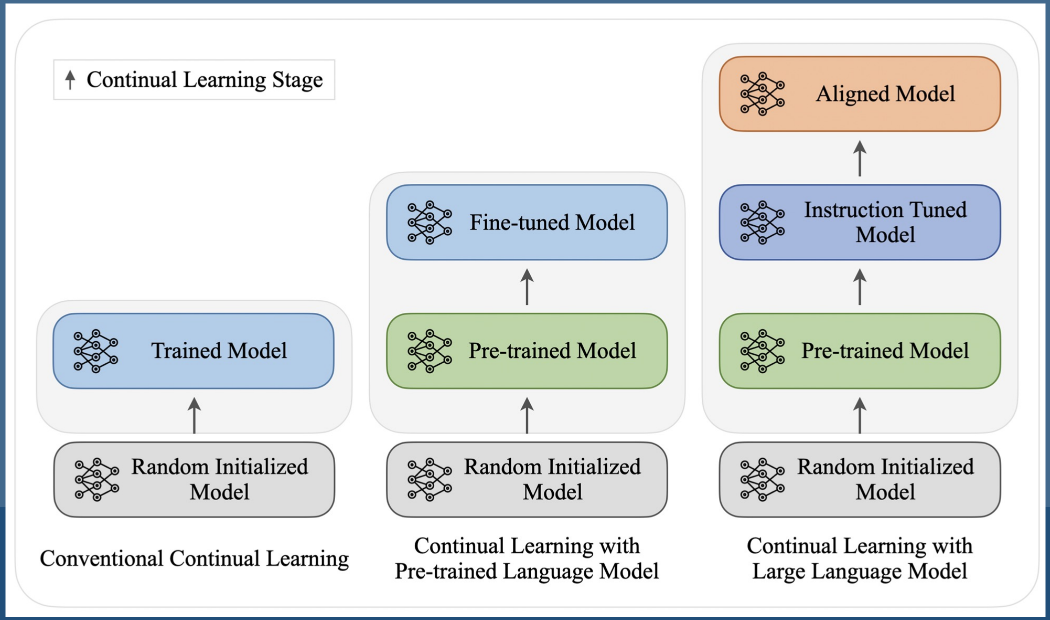
Long et al. (DeepMind, PNAS2016) “Overcoming catastrophic forgetting in neural networks”
Wu et al. (Monash Univ, 2024) “Continual Learning for Large Language Models: A Survey”

- **Catastrophic Forgetting**
  : When a model forgets old knowledge after learning new tasks, causing performance on previous tasks to degrade.



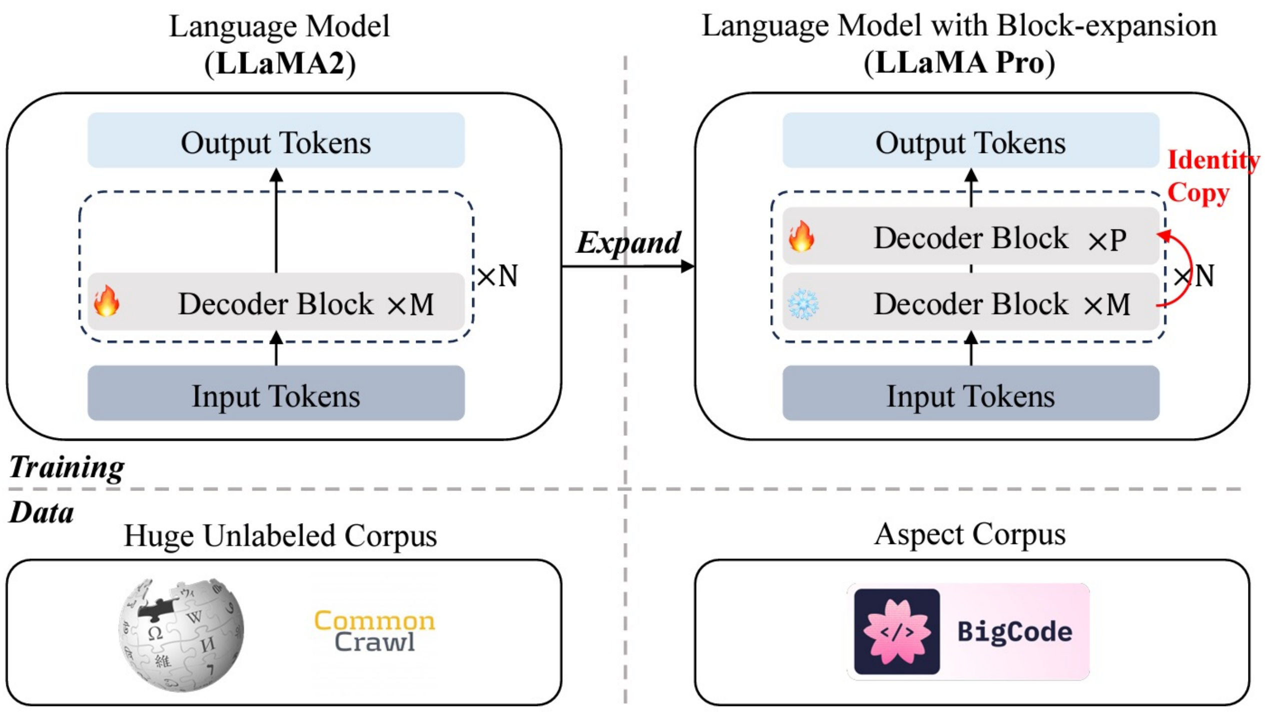
$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i(\theta_i - \theta_{A,i}^*)^2$$

- **Continual Learning**
  : A method where a model learns new tasks incrementally while retaining knowledge of previous tasks, avoiding catastrophic forgetting.

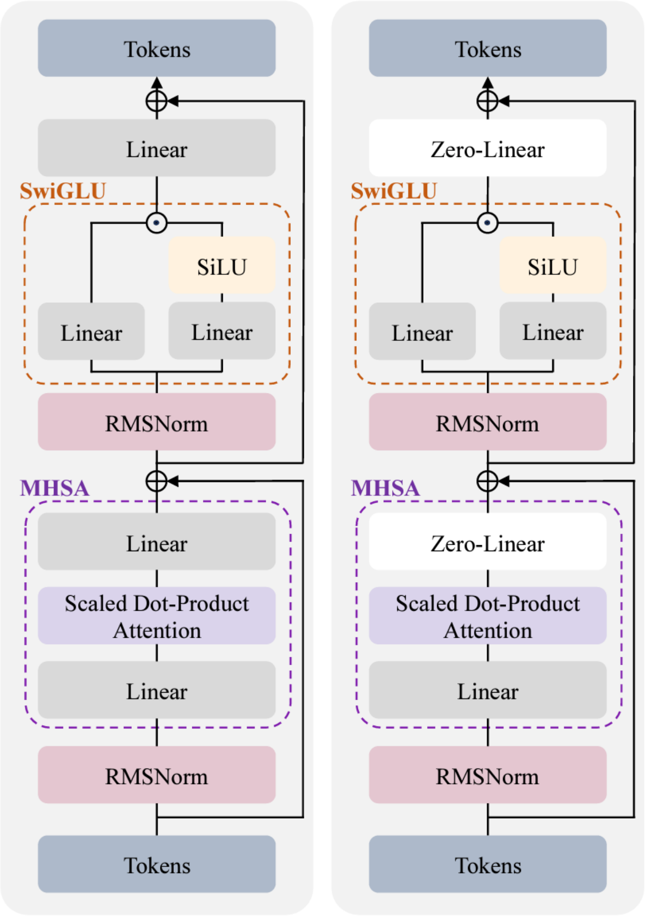


Suggestions

- **Block Expansion**



(a) Pre-Training (b) Block-expansion Post-pretraining

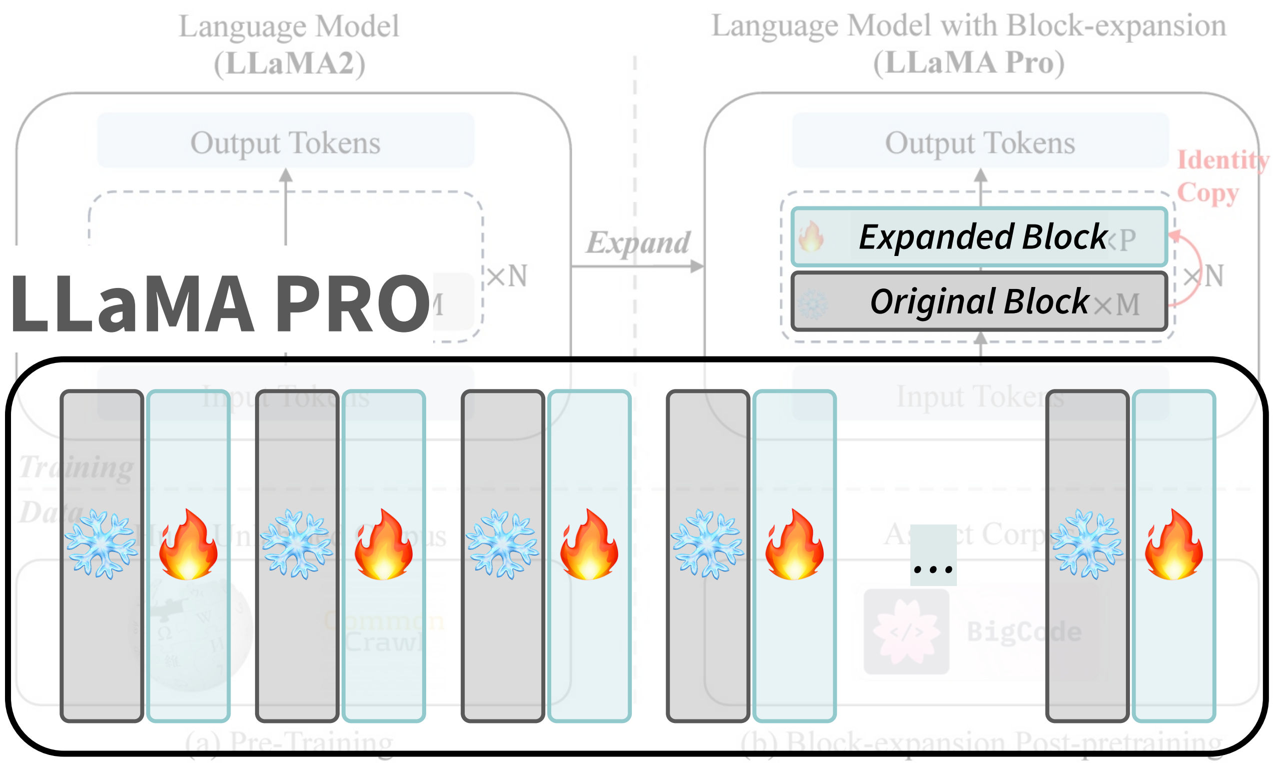


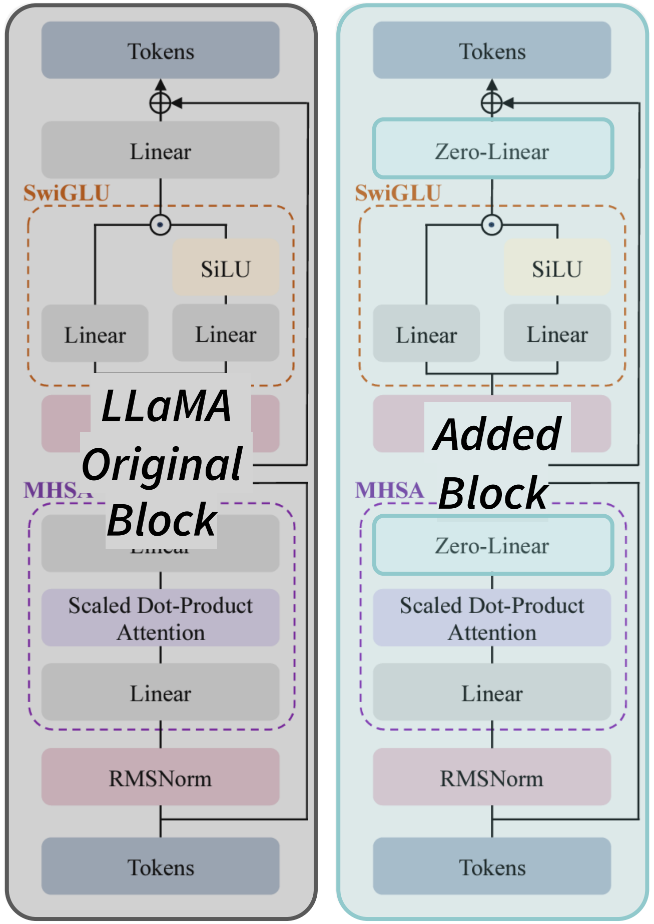
(a) LLaMA Block (b) LLaMA Block after Identity Copy

Every n-th layer is copied, and new blocks are added.

Suggestions

- **Block Expansion**



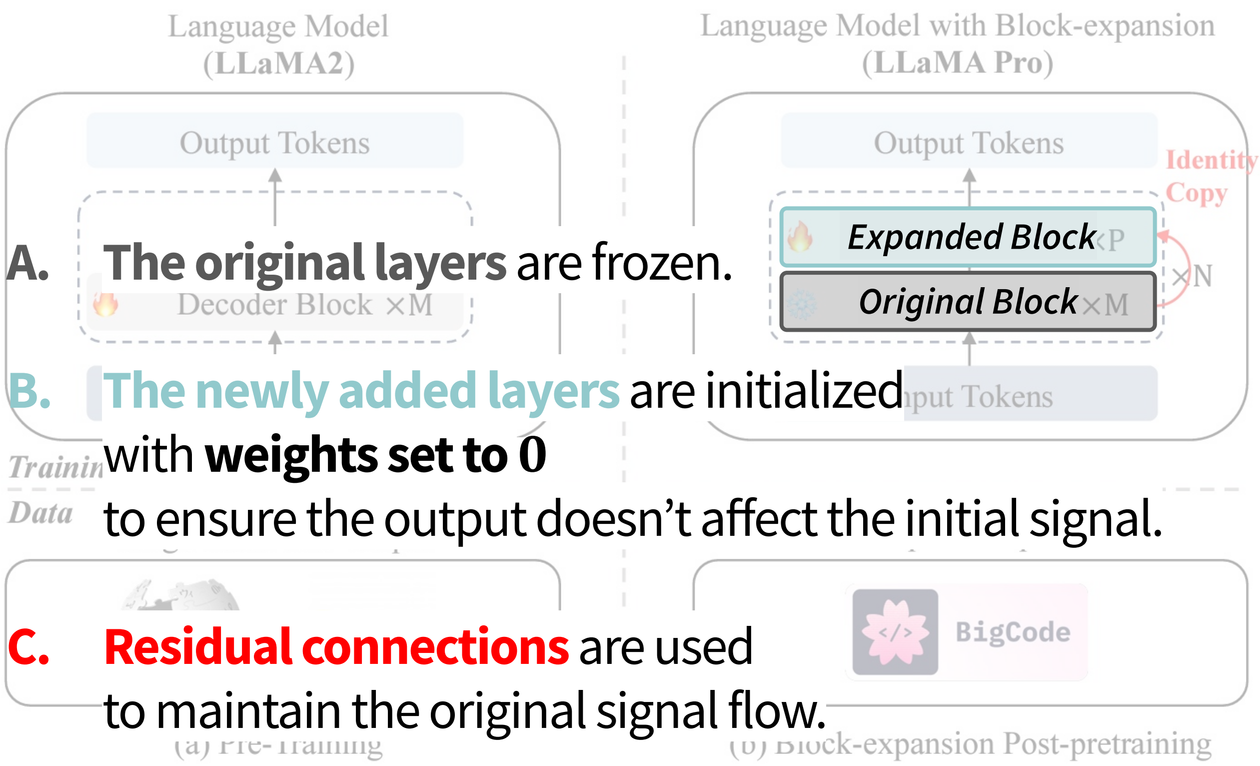


(a) LLaMA Block (b) LLaMA Block after Identity Copy

Every n-th layer is copied, and new blocks are added.

Suggestions

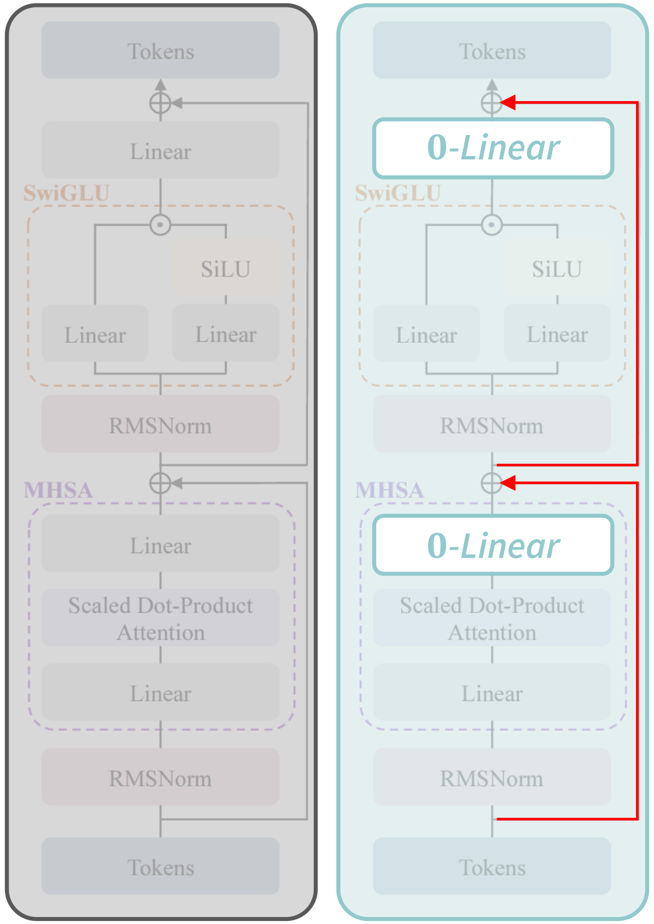
- **Principle of Block Expansion**



A. **The original layers** are frozen.

B. **The newly added layers** are initialized with **weights set to 0** to ensure the output doesn't affect the initial signal.

C. **Residual connections** are used to maintain the original signal flow.

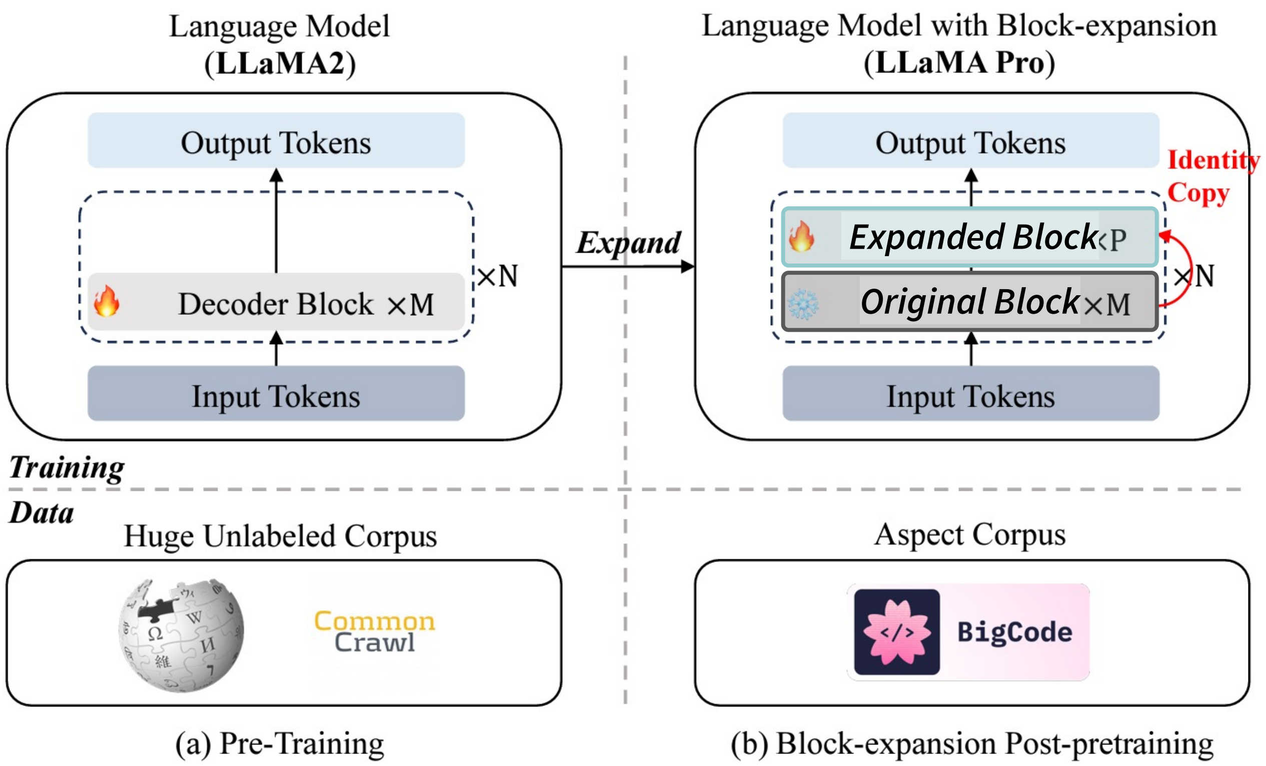


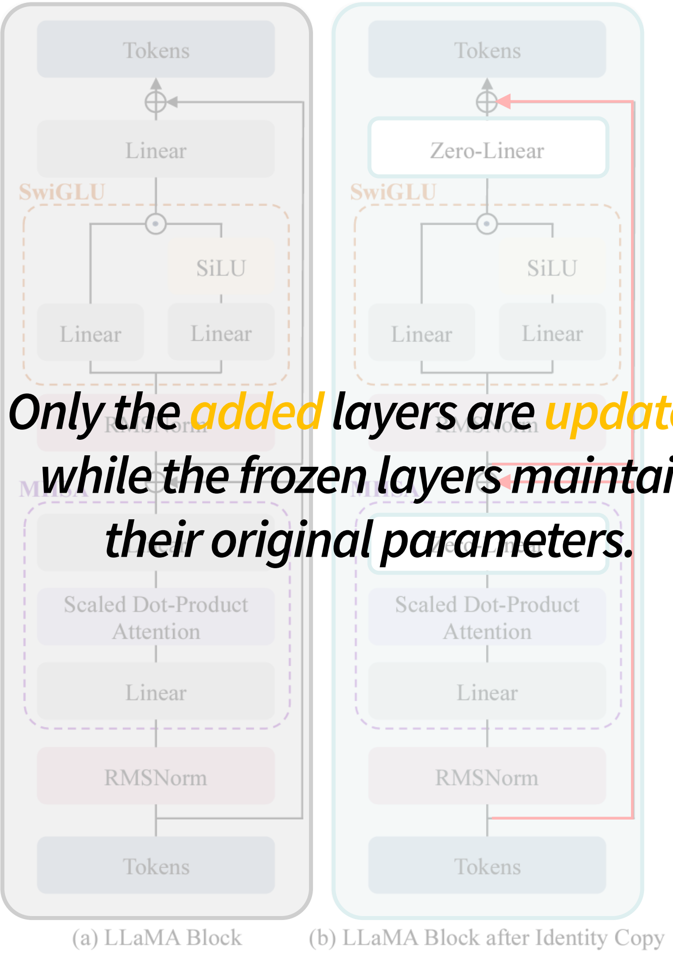
(a) LLaMA Block (b) LLaMA Block after Identity Copy

In LLaMA Pro, the original layers are frozen, and only the newly added layers are trained.

Suggestions

- **Training Strategy**





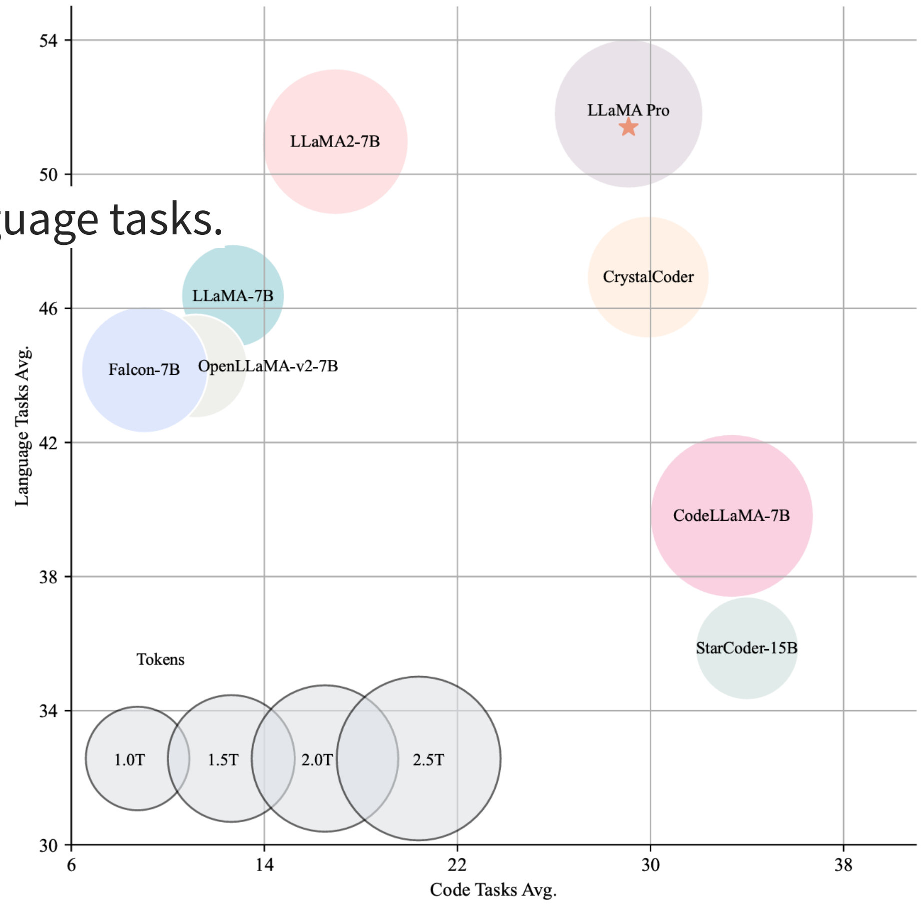
Only the added layers are updated, while the frozen layers maintain their original parameters.

The model is trained using only new data, without mixing in old data.

Results

- **Pretraining& SFT Results**
  - It not only preserves the general performance of its base model, LLaMA2-7B, but also surpasses it in the average performance of general language tasks.

Model	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K	GSM8K-PoT	HumanEval	MBPP	Avg.
	Language Tasks					Math Tasks		Code Tasks		
Pretrained comparison										
LLaMA Pro (8B)	54.10	77.94	47.88	39.04	73.95	17.89	25.42	28.66	33.20	**44.23**
CrystalCoder (7B)	47.01	71.97	48.78	35.91	67.17	10.77	24.96	28.38	36.38	41.26
LLaMA2-7B	53.07	78.59	46.87	38.76	74.03	14.48	17.68	13.05	20.09	39.62
CodeLLaMA-7B	39.93	60.80	31.12	37.82	64.01	5.16	25.20	33.50	41.40	37.66
StarCoder-15B	30.38	47.93	29.96	41.28	56.12	9.48	25.09	33.63	43.28	35.24
LLaMA-7B	50.94	77.81	35.69	34.33	71.43	8.04	10.46	10.61	17.04	35.15
OpenLLaMA-v2-7B	43.69	72.20	41.29	35.54	69.38	3.49	5.46	15.32	12.69	33.23
Falcon-7B	47.87	78.13	27.79	34.26	72.38	4.62	4.32	9.42	13.39	32.46
SFT comparison										
LLaMA Pro - Instruct	52.30	76.88	52.57	48.80	72.53	43.59	55.61	44.51	37.88	**53.85**
LLaMA2-7B-Chat	52.90	78.55	48.32	45.57	71.74	7.35	19.73	14.63	21.60	40.04
CodeLLaMA-7B-Instruct	36.52	55.44	34.54	41.25	64.56	7.96	34.67	34.80	44.4	39.35
WizardCoder-Python-7B	41.81	65.06	32.29	36.32	61.72	4.70	17.60	42.07	47.20	38.75
WizardMath-7B	54.10	79.55	45.97	43.65	72.69	2.73	25.57	12.20	18.00	39.38



By adding new layers to the existing Llama2-7B,
LLaMA PRO likely grew by about 1 billion parameters, reaching a total of 8B parameters.

Paper Review : 2024 Fall Lab Seminar

SOLAR10.7B: Scaling Large Language Models with Simple yet Effective Depth Up-Scaling

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song et al. (Upstage AI)

Accepted to NAACL2024 Industry Track

Yejin Yoon

HYU 한양대학교 HANYANG UNIVERSITY

Natural Language Processing Lab.,
Hanyang University.

Task-specific Training vs. General LLM Training

- Goals
  - LLM Training: Designed for general performance improvement across a wide range of tasks.

Problem States:

How to improve specific task(s) performance while maintaining general model performance

[Task-specific Training] *The model might lose some general capability!*

Pretrining → Fine-tuning ← PEFT

Initial large-scale training of the model on vast amounts of generic data.

SOLAR 10.7B Depth Up-Scaling

Pretrining → Instruction Tuning → Alignment Tuning → sLLM

Continual Pretraining

The model is trained to follow human instructions better, improving its ability to understand and respond to user queries.

Fine-tuning to align the model's outputs with human values or ethical guidelines. This helps in reducing biases and harmful outputs, making the model safer and more reliable.

Make the model more efficient in terms of memory usage & computational cost → more practical for deployment in resource-constrained environments

Problem States

Shazeer et al. (Google, ICLR2017) "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer"
Tan et al. (Google, ICML2019) "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks"

- **Background**

  - **Scaling Laws** in LLMs: Increasing model size & training data improves performance.

  - Goal: Efficiently scale models while maintaining performance and compatibility.
    - Related Work: **Mixture-of-Experts** (MoE)
      - Parallel connection of multiple small models/networks/… (experts)

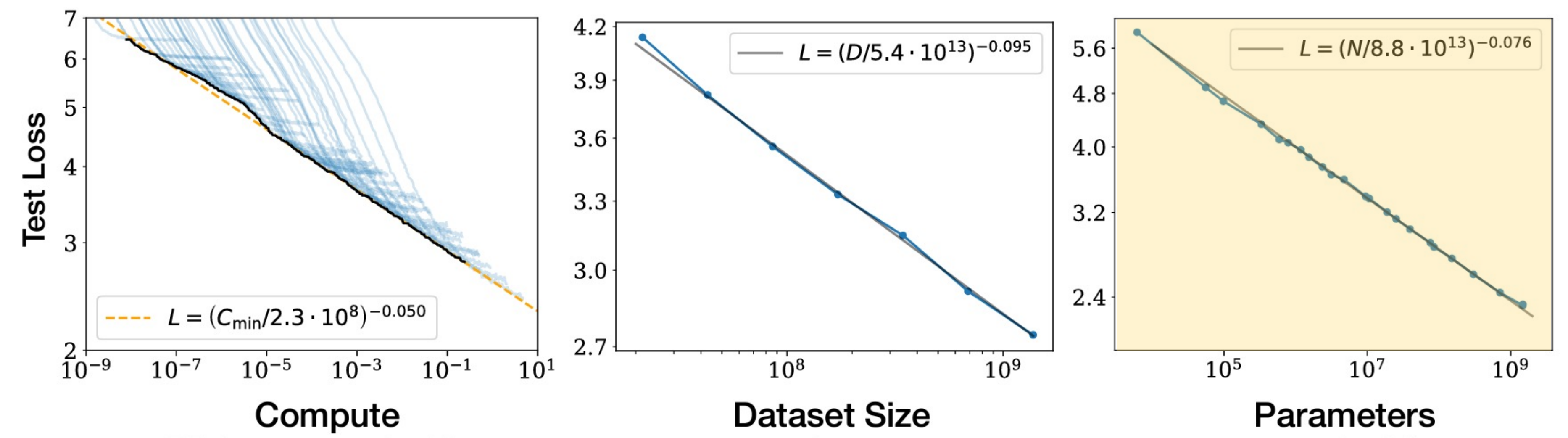
  - Inspired by **EfficientNet**

Applies depth-wise scaling from computer vision to LLMs.

Background ① Scaling Laws

Kaplan et al. (OpenAI, 2020) "Scaling Laws for Neural Language Models"

- **(Empirical Study) Scaling Laws for Language Model Performance**
  - Model size (parameters) N, Dataset size D, Computing Power C
  - Language modeling performance improves smoothly as we increase the **model size, dataset size**, and **amount of compute**[2] used for training.

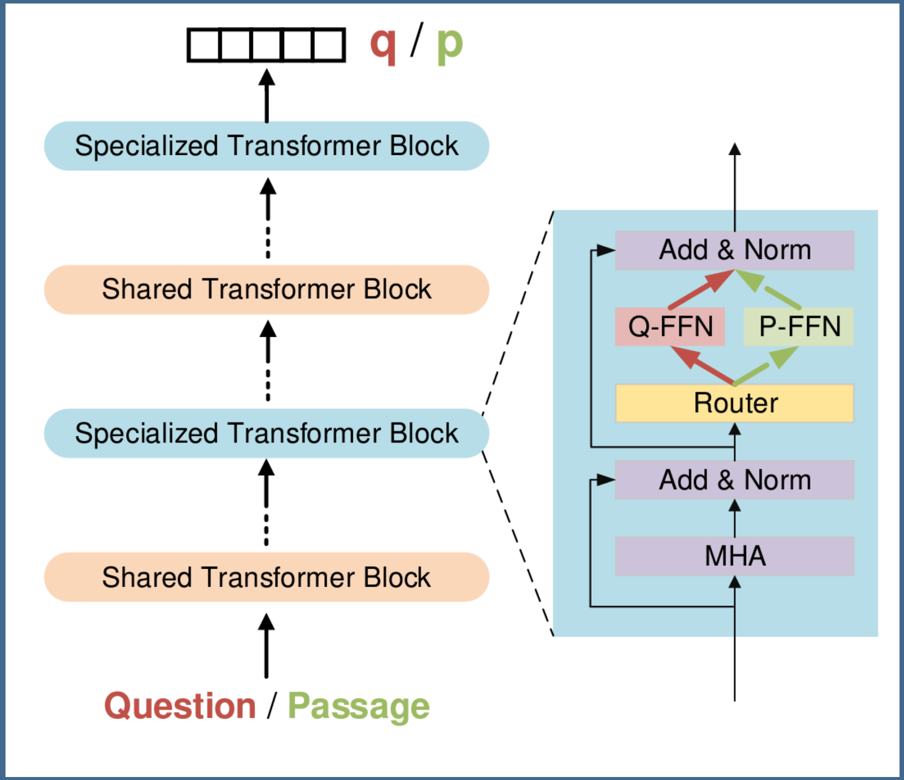


Larger language models will continue to perform better

Background ② Mixture-of-Experts

Cheng et al. (Microsoft, ACL2023) “Task-Aware Specialization for Efficient and Robust Dense Retrieval for Open-Domain Question Answering”
Fedus et al. (Google, JMLR2023) “Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity”

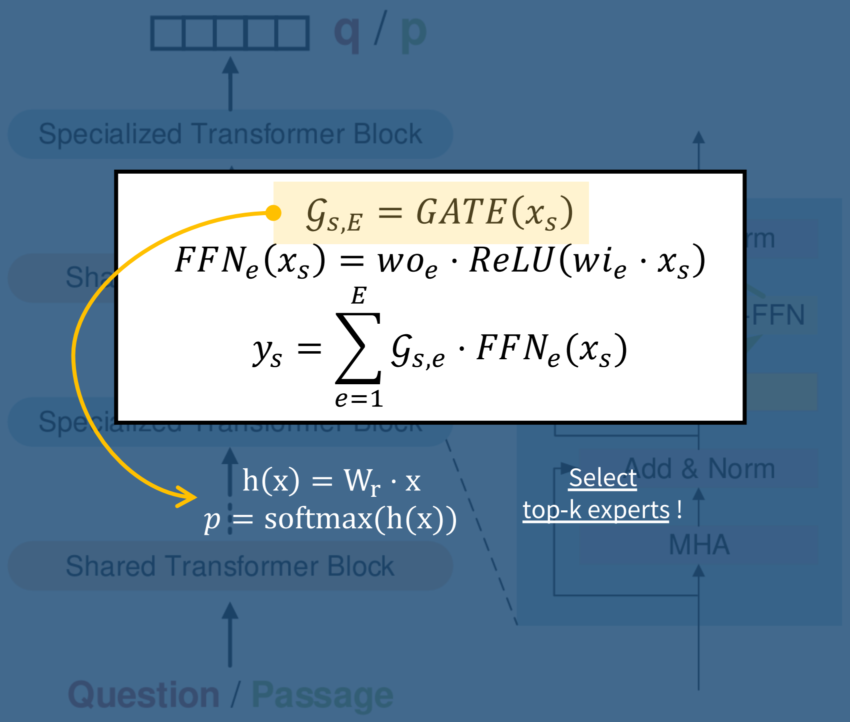
- **MoE with routing mechanism**



Background ② Mixture-of-Experts

Cheng et al. (Microsoft, ACL2023) "Task-Aware Specialization for Efficient and Robust Dense Retrieval for Open-Domain Question Answering"
Fedus et al. (Google, JMLR2023) "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity"

- **MoE with routing mechanism**



$$\mathcal{G}_{s,E} = GATE(x_s)$$

$$FFN_e(x_s) = wo_e \cdot ReLU(wi_e \cdot x_s)$$

$$y_s = \sum_{e=1}^{E} \mathcal{G}_{s,e} \cdot FFN_e(x_s)$$

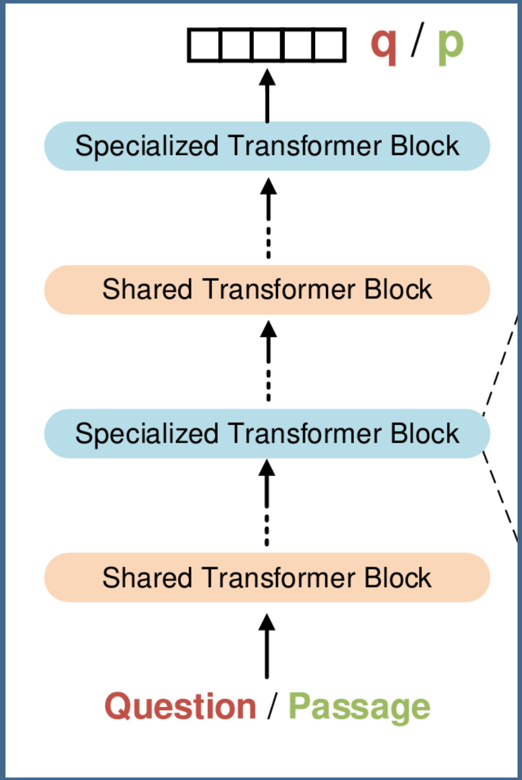
$$h(x) = W_r \cdot x$$

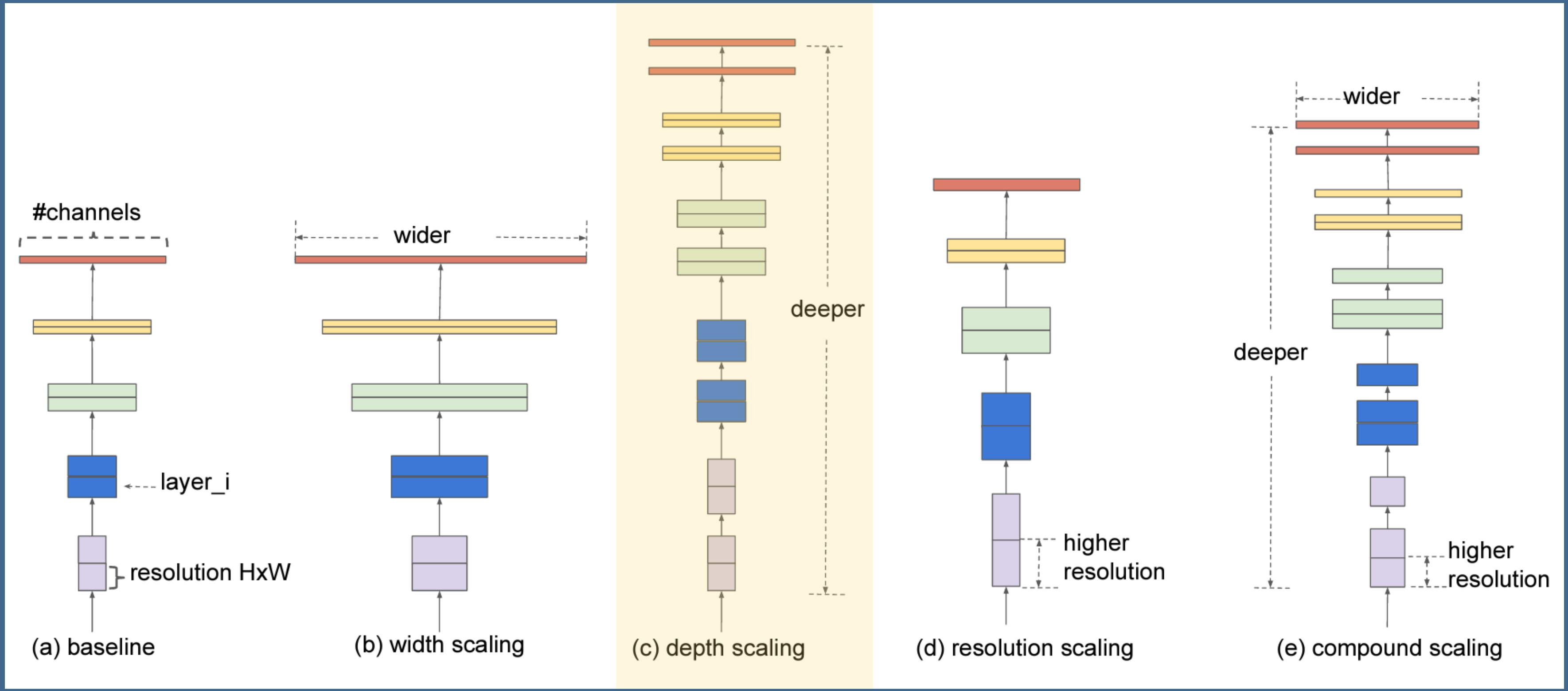
$$p = \mathrm{softmax}(h(x))$$

Background ③ EfficientNet

Cheng et al. (Microsoft, ACL2023) “Task-Aware Specialization for Efficient and Robust Dense Retrieval for Open-Domain Question Answering”
Tan et al. (Google, ICML2019) “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”

- **TASER || EfficientNet – Depth Scaling**



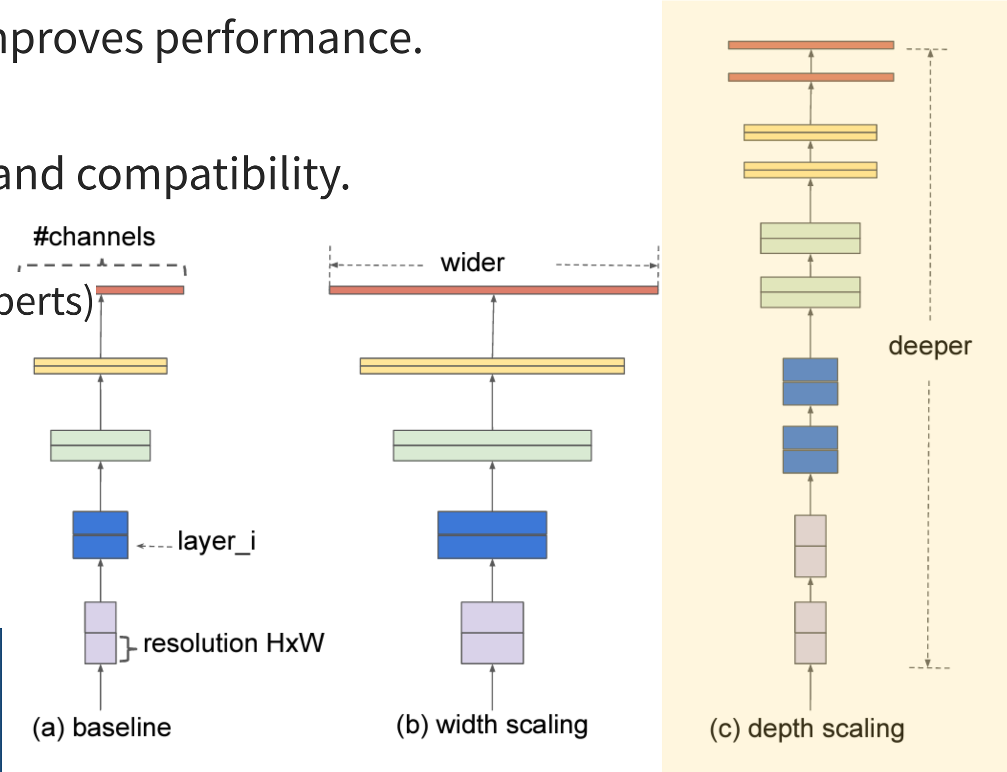


Problem States

Shazeer et al. (Google, ICLR2017) "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer"
Tan et al. (Google, ICML2019) "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks"

- **Background**

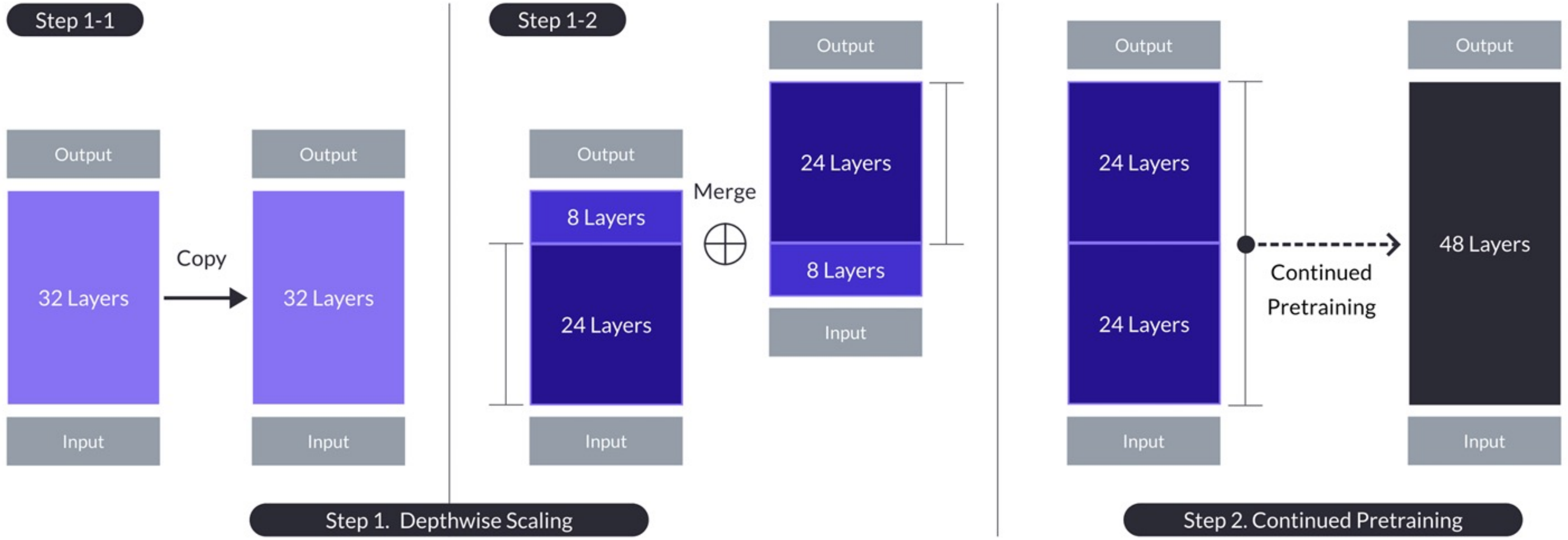
  - **Scaling Laws** in LLMs: Increasing model size & training data improves performance.
  - Goal: Efficiently scale models while maintaining performance and compatibility.
    - Related Work: **Mixture-of-Experts** (MoE)
      - Parallel connection of multiple small models/networks/… (experts)
  - Inspired by **EfficientNet**

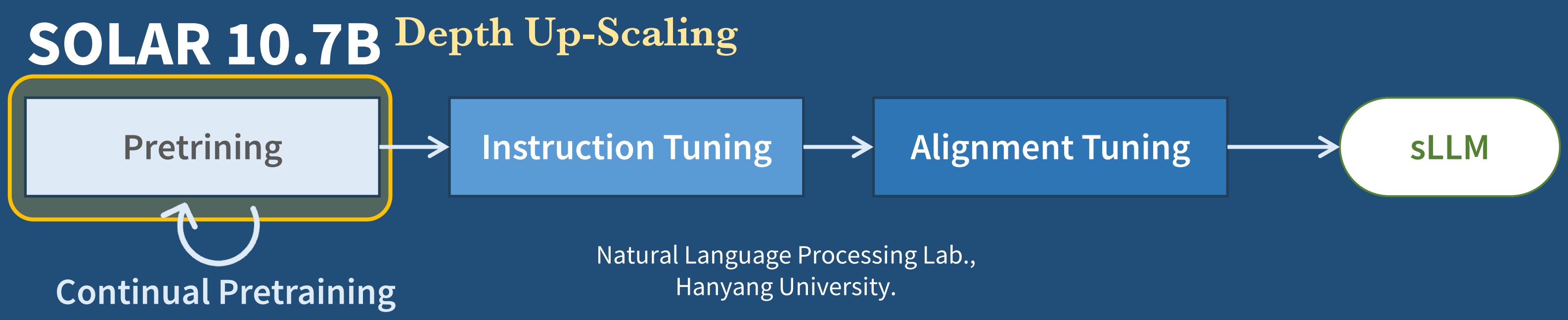


Applies depth-wise scaling from CV to LLMs.

Suggestions

- **Depth Up-Scaling (DUS)**

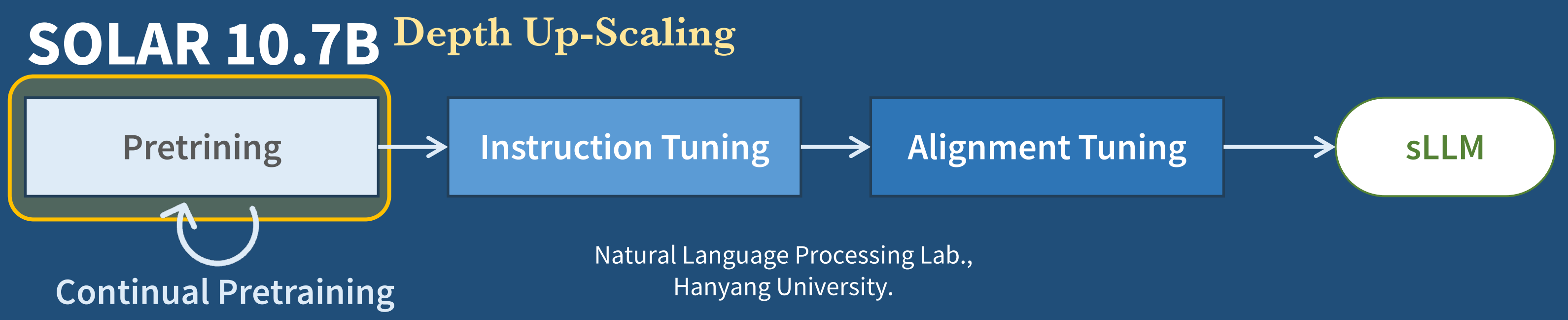




Results

- **Effect from DUS Continual Learning**
  - Mistral 7B → SOLAR 10.7B : Better performance

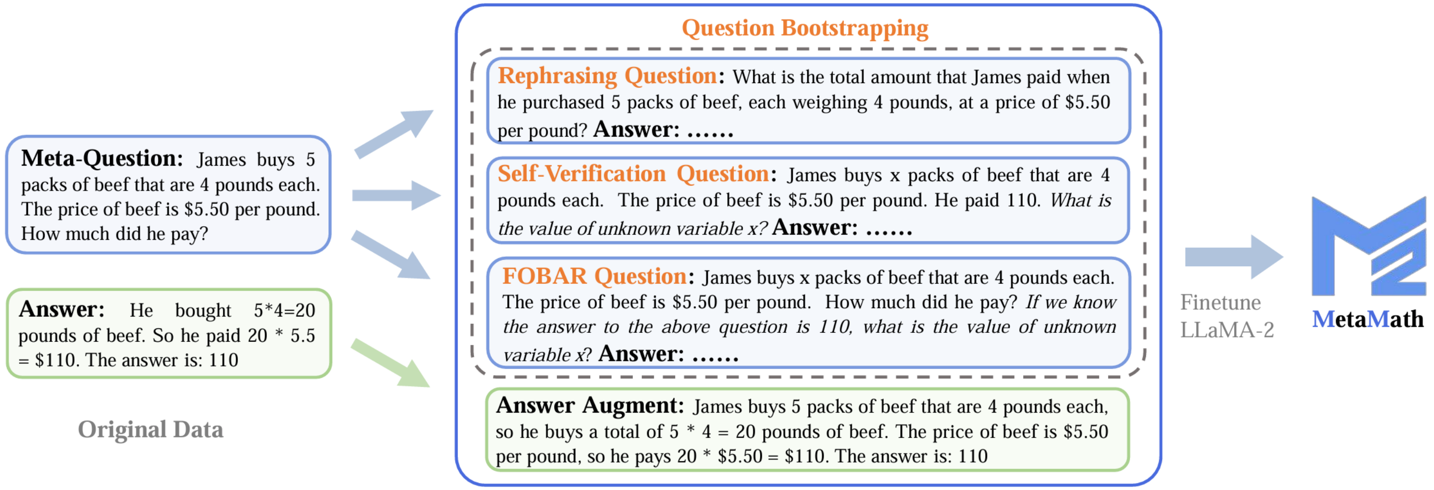
Model	Size	Type	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
SOLAR 10.7B-Instruct	~ 11B	Alignment-tuned	**74.20**	**71.08**	88.16	66.21	**71.43**	83.58	64.75
Qwen 72B	~ 72B	Pretrained	73.60	65.19	85.94	**77.37**	60.19	82.48	**70.43**
Mixtral 8x7B-Instruct-v0.1	~ 47B	Instruction-tuned	72.62	70.22	87.63	71.16	64.58	81.37	60.73
Yi 34B-200K	~ 34B	Pretrained	70.81	65.36	85.58	76.06	53.64	82.56	61.64
Yi 34B	~ 34B	Pretrained	69.42	64.59	85.69	76.35	56.23	83.03	50.64
Mixtral 8x7B-v0.1	~ 47B	Pretrained	68.42	66.04	86.49	71.82	46.78	81.93	57.47
Llama 2 70B	~ 70B	Pretrained	67.87	67.32	87.33	69.83	44.92	83.74	54.06
Falcon 180B	~ 180B	Pretrained	67.85	69.45	**88.86**	70.50	45.47	**86.90**	45.94
SOLAR 10.7B	~ 11B	Pretrained	66.04	61.95	84.60	65.48	45.04	83.66	55.50
Qwen 14B	~ 14B	Pretrained	65.86	58.28	83.99	67.70	49.43	76.80	58.98
Mistral 7B-Instruct-v0.2	~ 7B	Instruction-tuned	65.71	63.14	84.88	60.78	68.26	77.19	40.03
Yi 34B-Chat	~ 34B	Instruction-tuned	65.32	65.44	84.16	74.90	55.37	80.11	31.92
Mistral 7B	~ 7B	Pretrained	60.97	59.98	83.31	64.16	42.15	78.37	37.83



Suggestions

Yu et al. (Univ. of Cam., ICLR2024 Spotlight) “MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models”

- **Training Details : Synth. Math-Instruct**
  - While using open-sourced dataset, add synthetic math QA dataset. (Rephrase GSM8K question & Answer)



SOLAR 10.7B

Pretrining → **Instruction Tuning** → Alignment Tuning → sLLM

Results

- **Training Details : Synth. Math-Instruct**

Model	Alpaca-GPT4	OpenOrca	Synth. Math-Instruct	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
SFT v1	O	✗	✗	69.15	**67.66**	**86.03**	65.88	**60.12**	**82.95**	52.24
SFT v2	O	O	✗	69.21	65.36	85.39	65.93	58.47	82.79	57.32
SFT v3	O	O	O	70.03	65.87	85.55	65.31	57.93	81.37	64.14
SFT v4	O	✗	O	70.88	67.32	85.87	65.87	58.97	82.48	64.75
SFT v3 + v4	O	O	O	**71.11**	67.32	85.96	**65.95**	58.80	82.08	**66.57**

- Adding *Synth. Math-Instruct* was found to improve overall performance, especially in math-related tasks.
- However, new learning can lead to **catastrophic forgetting** of old knowledge.

SOLAR 10.7B

Pretrining → Instruction Tuning → Alignment Tuning → sLLM

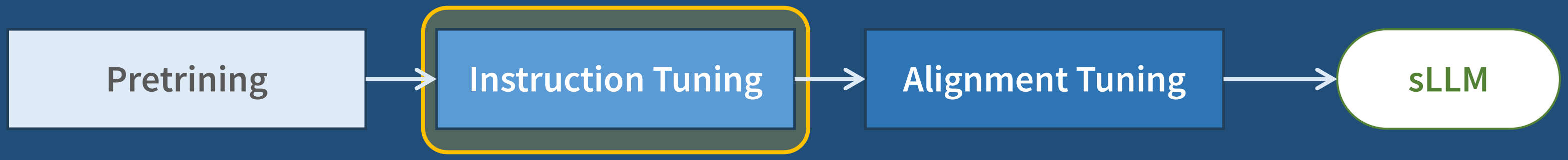
Results

- **Training Details : Synth. Math-Instruct**

Model	Alpaca-GPT4	OpenOrca	Synth. Math-Instruct	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
SFT v1	O	✗	✗	69.15	**67.66**	**86.03**	65.88	**60.12**	**82.95**	52.24
SFT v2	O	O	✗	69.21	65.36	85.39	65.93	58.47	82.79	57.32
SFT v3	O	O	O	70.03	65.87	85.55	65.31	57.93	81.37	64.14
SFT v4	O	✗	O	70.88	67.32	85.87	65.87	58.97	82.48	64.75
SFT v3 + v4	O	O	O	**71.11**	67.32	85.96	**65.95**	58.80	82.08	**66.57**

- Adding *Synth. Math-Instruct* was found to improve overall performance, especially in math-related tasks.
- However, new learning can lead to **catastrophic forgetting** of old knowledge.

SOLAR 10.7B



Results

- **Training Details ~~: Synth. Math-Instruct~~**

Model	Alpaca-GPT4	OpenOrca	Synth. Math-Instruct	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
SFT v1	O	✗	✗	69.15	**67.66**	**86.03**	65.88	**60.12**	**82.95**	52.24
SFT v2	O	O	✗	69.21	65.36	85.39	65.93	58.47	82.79	57.32
SFT v3	O	O	O	70.03	65.87	85.55	65.31	57.93	81.37	64.14
SFT v4	O	✗	O	70.88	67.32	85.87	65.87	58.97	82.48	64.75
SFT v3 + v4	O	O	O	**71.11**	67.32	85.96	**65.95**	58.80	82.08	**66.57**

Properties	Training Datasets					
	Instruction			Alignment		
	Alpaca-GPT4	OpenOrca	Synth. Math-Instruct	Orca DPO Pairs	Ultrafeedback Cleaned	Synth. Math-Alignment
Total # Samples	52K	2.91M	126K	12.9K	60.8K	126K
Maximum # Samples Used	52K	100K	52K	12.9K	60.8K	20.1K
Open Source	O	O	✗	O	O	✗

SOLAR 10.7B

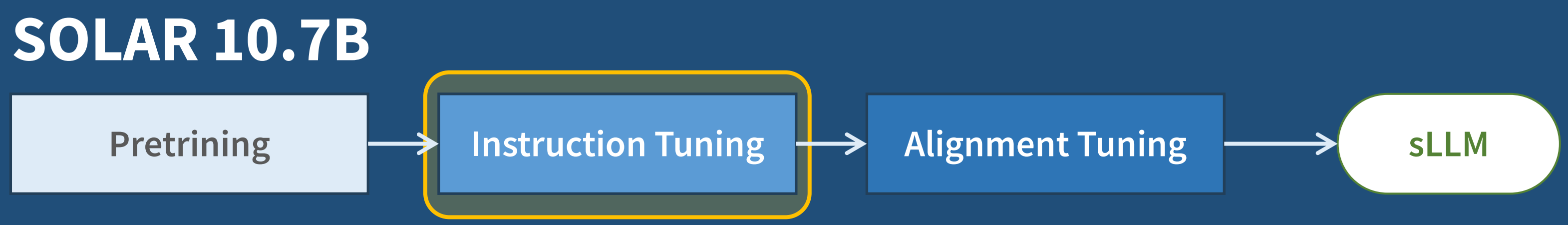
Pretrining → Instruction Tuning → Alignment Tuning → sLLM

Results

- **Training Details ~~: Synth. Math-Instruct~~**

Model	Alpaca-GPT4	OpenOrca	Synth. Math-Instruct	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
SFT v1	O	✗	✗	69.15	**67.66**	**86.03**	65.88	**60.12**	**82.95**	52.24
SFT v2	O	O	✗	69.21	65.36	85.39	65.93	58.47	82.79	57.32
SFT v3	O	O	O	70.03	65.87	85.55	65.31	57.93	81.37	64.14
SFT v4	O	✗	O	70.88	67.32	85.87	65.87	58.97	82.48	64.75
SFT v3 + v4	O	O	O	**71.11**	67.32	85.96	**65.95**	58.80	82.08	**66.57**

Properties	Training Datasets					
	Instruction			Alignment		
	Alpaca-GPT4	OpenOrca	Synth. Math-Instruct	Orca DPO Pairs	Ultrafeedback Cleaned	Synth. Math-Alignment
Total # Samples	52K	2.91M	126K	12.9K	60.8K	126K
Maximum # Samples Used	52K	100K	52K	12.9K	60.8K	20.1K
Open Source	O	O	✗	O	O	✗



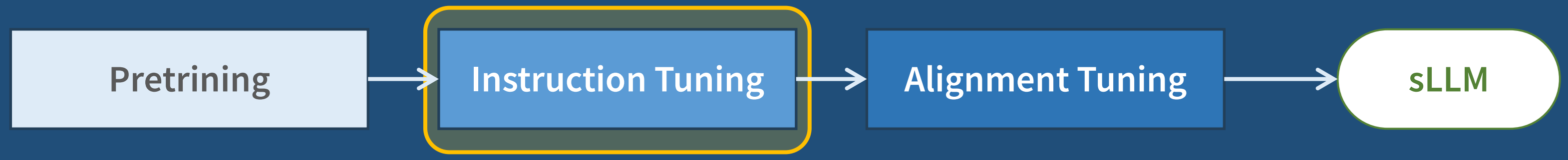
Results

- **Training Details : Synth. Math-Instruct**

Model	Alpaca-GPT4	OpenOrca	Synth. Math-Instruct	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
SFT v1	O	✗	✗	69.15	**67.66**	**86.03**	65.88	**60.12**	**82.95**	52.24
SFT v2	O	O	✗	69.21	65.36	85.39	65.93	58.47	82.79	57.32
SFT v3	O	O	O	70.03	65.87	85.55	65.31	57.93	81.37	64.14
SFT v4	O	✗	O	70.88	67.32	85.87	65.87	58.97	82.48	64.75
SFT v3 + v4	O	O	O	**71.11**	67.32	85.96	**65.95**	58.80	82.08	**66.57**

- Adding *Synth. Math-Instruct* was found to improve overall performance, especially in math-related tasks.
- However, new learning can lead to **catastrophic forgetting** of old knowledge.
- Combining models trained with and without *OpenOrca* improved performance across multiple tasks.
- ‘SFTv3+v4’ indicates that the model is merged from ‘SFTv3’&‘SFTv4’ by simply averaging the model weights.

SOLAR 10.7B



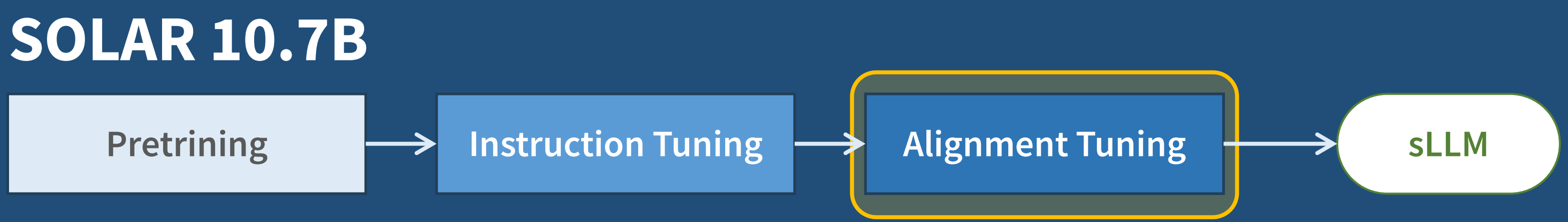
Suggestions

Yu et al. (Univ. of Cam., ICLR2024 Spotlight) “MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models”

- **Training Details : Alignment Learning (DPO)**
  - ‘SFTv3’ is used as the SFT base model for DPO.

Model	Ultrafeedback Clean	Synth. Math-Alignment	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
DPO v1	O	✗	73.06	71.42	**88.49**	**66.14**	72.04	81.45	58.83
DPO v2	O	O	**73.42**	**71.50**	88.28	65.97	71.71	**82.79**	**60.27**
DPO v1 + v2	O	O	73.21	71.33	88.36	65.92	**72.65**	**82.79**	58.23

Properties	Training Datasets		
	Alignment		
	Orca DPO Pairs	Ultrafeedback Cleaned	Synth. Math-Alignment
Total # Samples	12.9K	60.8K	126K
Maximum # Samples Used	12.9K	60.8K	20.1K
Open Source	O	O	✗



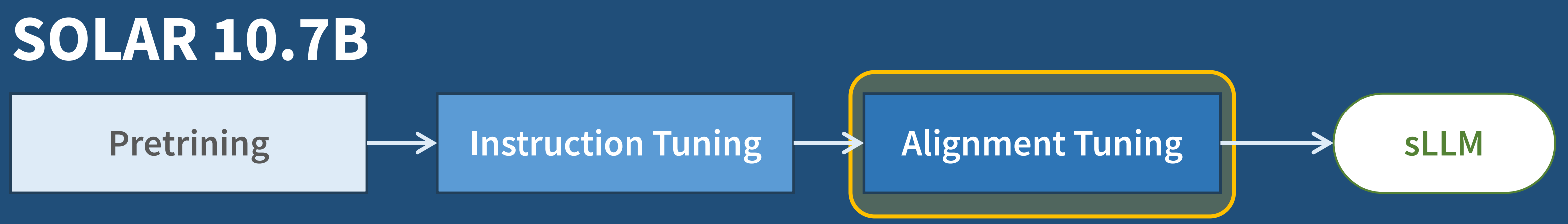
Suggestions

Yu et al. (Univ. of Cam., ICLR2024 Spotlight) "MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models"

- **Training Details : Alignment Learning (DPO)**
  - 'SFTv3' is used as the SFT base model for DPO.

Model	Alpaca-GPT4	OpenOrca	Synth. Math-Instruct	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
SFT v1	O	✗	✗	69.15	**67.66**	**86.03**	65.88	**60.12**	**82.95**	52.24
SFT v2	O	O	✗	69.21	65.36	85.39	65.93	58.47	82.79	57.32
SFT v3	O	O	O	70.03	65.87	85.55	65.31	57.93	81.37	64.14
SFT v4	O	✗	O	70.88	67.32	85.87	65.87	58.97	82.48	64.75
SFT v3 + v4	O	O	O	**71.11**	67.32	85.96	**65.95**	58.80	82.08	**66.57**

Model	Base SFT Model	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
DPO v2	SFT v3	73.42	**71.50**	**88.28**	**65.97**	71.71	**82.79**	60.27
DPO v3	SFT v3 + v4	**73.58**	71.33	88.08	65.39	**72.45**	81.93	**62.32**



Suggestions

Yu et al. (Univ. of Cam., ICLR2024 Spotlight) "MetaMath: Bootstrap Your Own Mathematical Questions for Large Language Models"

- **Training Details : Hyper-parameter Tuning (for merging technique)**

Model	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
Cand. 1	**73.73**	70.48	87.47	65.73	70.62	81.53	**66.57**
Cand. 2	73.28	**71.59**	**88.39**	**66.14**	**72.50**	**81.99**	59.14

Model	Merge Method	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
Merge v1	Average (0.5, 0.5)	74.00	**71.16**	88.01	66.14	71.71	**82.08**	64.90
Merge v2	Average (0.4, 0.6)	73.93	71.08	**88.08**	**66.27**	**71.89**	81.77	64.52
Merge v3	Average (0.6, 0.4)	**74.05**	71.08	87.88	66.13	71.61	**82.08**	**65.50**
Merge v4	SLERP	73.96	**71.16**	88.03	66.25	71.79	81.93	64.59

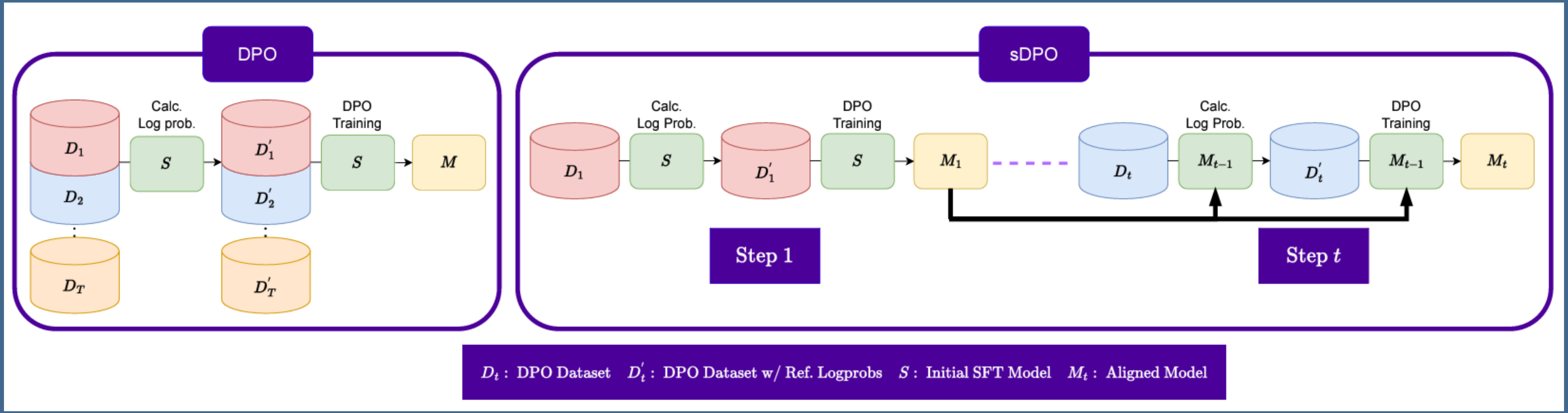
SOLAR 10.7B

Pretrining → Instruction Tuning → Alignment Tuning → sLLM

Background ④ sDPO

Kim et al. (Upstage, 2024) "sDPO: Don't Use Your Data All at Once"

- **Stepwise DPO**
  - Preference datasets are divided to be used in multiple steps instead of using them all at once.



Preventing Overoptimization

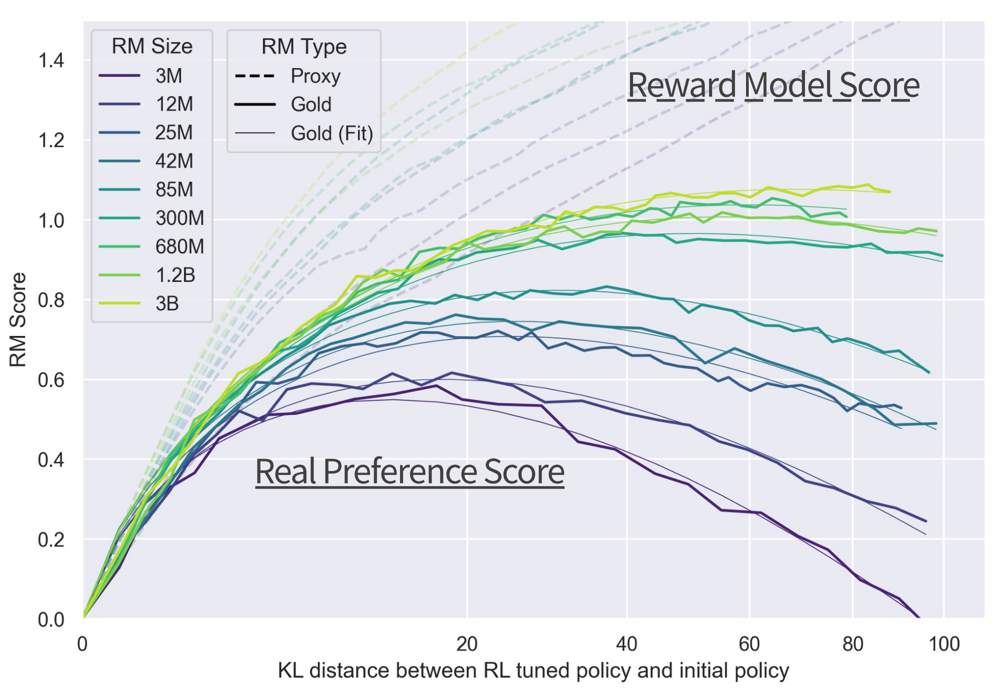
Recap.

- **Overoptimization**
  - Goodhart's Law

" When a measure becomes a target, it ceases to be a good measure. "

e.g. In machine learning, this effect arises with **proxy objectives** provided by static learned models, such as discriminators and reward models.

  - Solid Line: Gold RM Score
    = Real Preference Score
  - Dash Line: Proxy RM Score
  - Optimizing too much against such proxy model, eventually *hinders* the true objective

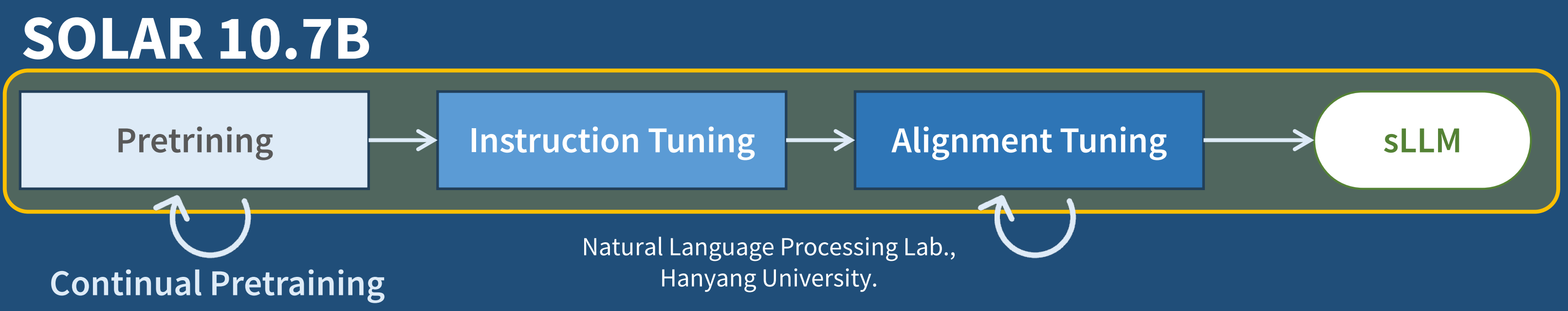


Reward models don't fully represent the real world.

Results

- **Effect from DUS + Synth. Math-Instruct + sDPO**
  - Evaluation results in OpenLLM Leaderboard

Model	Size	Type	H6 (Avg.)	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
SOLAR 10.7B-Instruct	∼ 11B	Alignment-tuned	**74.20**	**71.08**	88.16	66.21	**71.43**	83.58	64.75
Qwen 72B	∼ 72B	Pretrained	73.60	65.19	85.94	**77.37**	60.19	82.48	**70.43**
Mixtral 8x7B-Instruct-v0.1	∼ 47B	Instruction-tuned	72.62	70.22	87.63	71.16	64.58	81.37	60.73
Yi 34B-200K	∼ 34B	Pretrained	70.81	65.36	85.58	76.06	53.64	82.56	61.64
Yi 34B	∼ 34B	Pretrained	69.42	64.59	85.69	76.35	56.23	83.03	50.64
Mixtral 8x7B-v0.1	∼ 47B	Pretrained	68.42	66.04	86.49	71.82	46.78	81.93	57.47
Llama 2 70B	∼ 70B	Pretrained	67.87	67.32	87.33	69.83	44.92	83.74	54.06
Falcon 180B	∼ 180B	Pretrained	67.85	69.45	**88.86**	70.50	45.47	**86.90**	45.94
SOLAR 10.7B	∼ 11B	Pretrained	66.04	61.95	84.60	65.48	45.04	83.66	55.50
Qwen 14B	∼ 14B	Pretrained	65.86	58.28	83.99	67.70	49.43	76.80	58.98
Mistral 7B-Instruct-v0.2	∼ 7B	Instruction-tuned	65.71	63.14	84.88	60.78	68.26	77.19	40.03
Yi 34B-Chat	∼ 34B	Instruction-tuned	65.32	65.44	84.16	74.90	55.37	80.11	31.92
Mistral 7B	∼ 7B	Pretrained	60.97	59.98	83.31	64.16	42.15	78.37	37.83



Conclusion

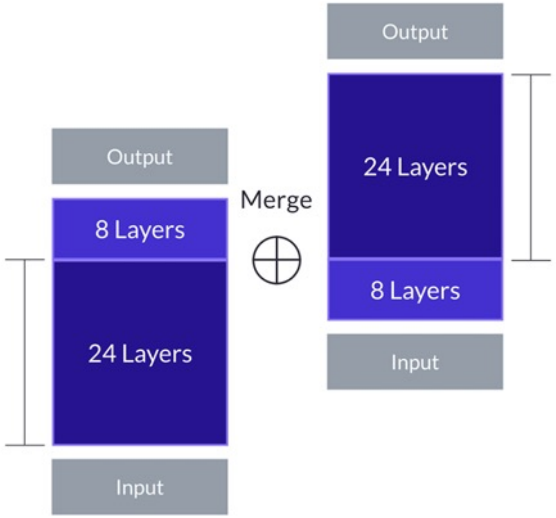
Main Findings

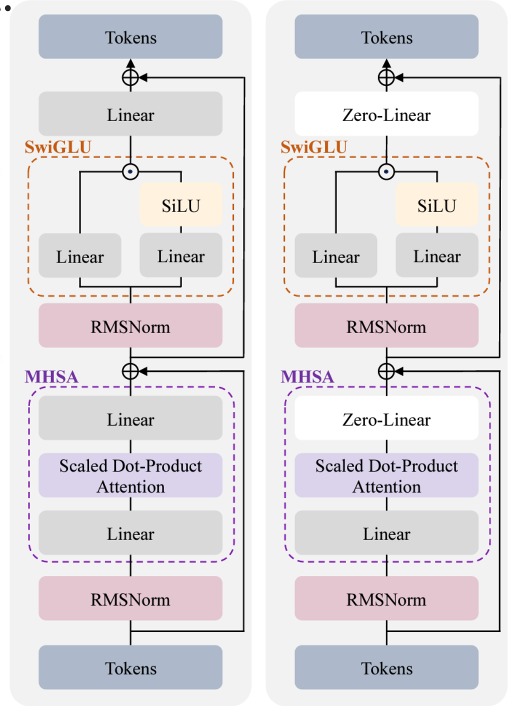
Summary

1. **(Background) Both approaches start with the need for a more powerful sLLM.**
   - Both make the model more efficient in terms of memory usage & computational cost.

2. **(Goal) Both approaches aim to maintain the general ability of the Initial Model while additionally improving specific tasks.**
   - Target task: Mathematical Reasoning

3. **(Suggestion) A vertical expansion approach by increasing depth.**
   - Block Expansion: Adding New Knowledge
   - Depth Up-Scaling: Improving overall performance with small scale increases



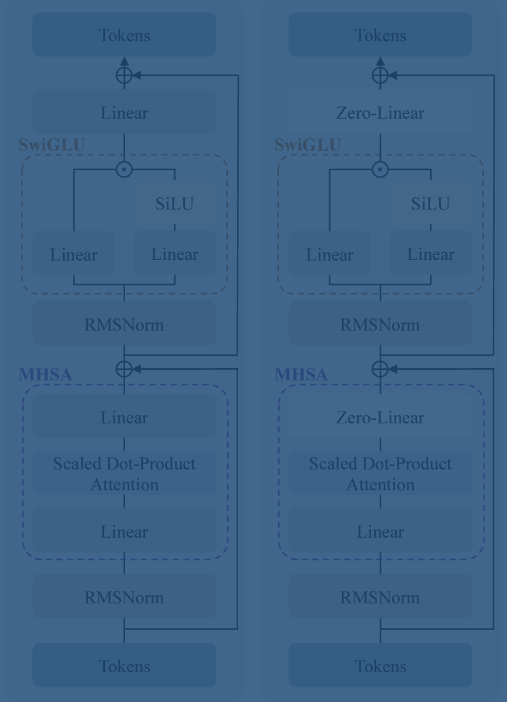


(a) LLaMA Block (b) LLaMA Block after Identity Copy

Hyper parameter optimization challenges / HW constraints

Summary

1. (Background) B … sLLM.
   - Both make the m … t.
2. (Goal) Both app … the general abi … while additiona …
   - Target task: Math
3. (Suggestion) A … by increasing d …
   - Block Expansion …



(a) LLaMA Block (b) LLaMA Block after Identity Copy

Stay tuned for SOLAR PRO 💫 and better generalization in sLLMs 👀

Thank You

Yejin Yoon

HYU NLP Lab.
Hanyang University, South Korea

stillwithyou@hanyang.ac.kr