Self-MoE:
Towards Compositional Large Language Models with Self-Specialized Experts

Junmo Kang[1] Leonid Karlinsky[2] Hongyin Luo[3] Zhen Wang[4]
Jacob Hansen[3] James Glass[3] David Cox[2] Rameswar Panda[2]
Rogerio Feris[2] Alan Ritter[1]

1 Georgia Institute of Technology
2 MIT-IBM Watson AI Lab
3 MIT CSAIL
4 UC San Diego

Towards Compositional Large Language Models

Rapid advancement of LLMs

- as a generalist
- relying on substantial resources - data, compute, and parameters

Towards Compositional Large Language Models

Rapid advancement of LLMs

- as a generalist
- relying on substantial resources - data, compute, and parameters
- a monolithic structure

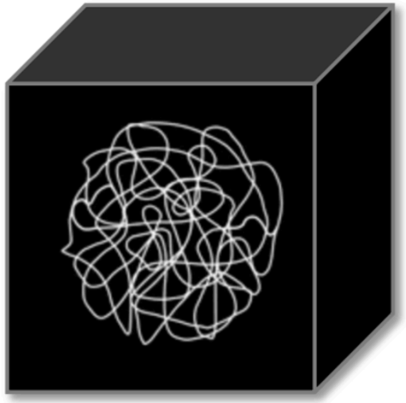
Monolithic

Generalist

Inefficient

Forgetting

Black-box



Towards Compositional Large Language Models

Rapid advancement of LLMs

- as a generalist
- relying on substantial resources - data, compute, and parameters
- a monolithic structure

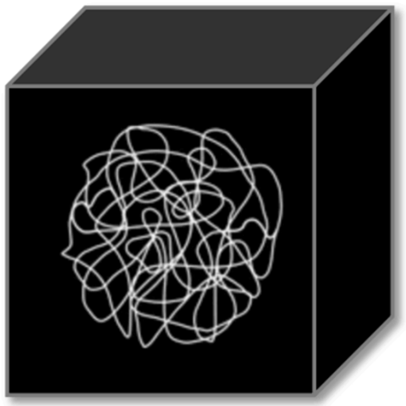
Monolithic

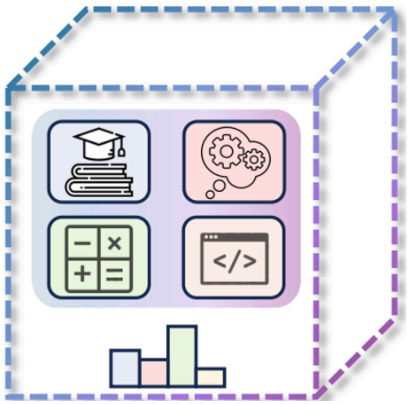
Generalist

Inefficient

Forgetting

Black-box





Compositional

Specialists

Efficient

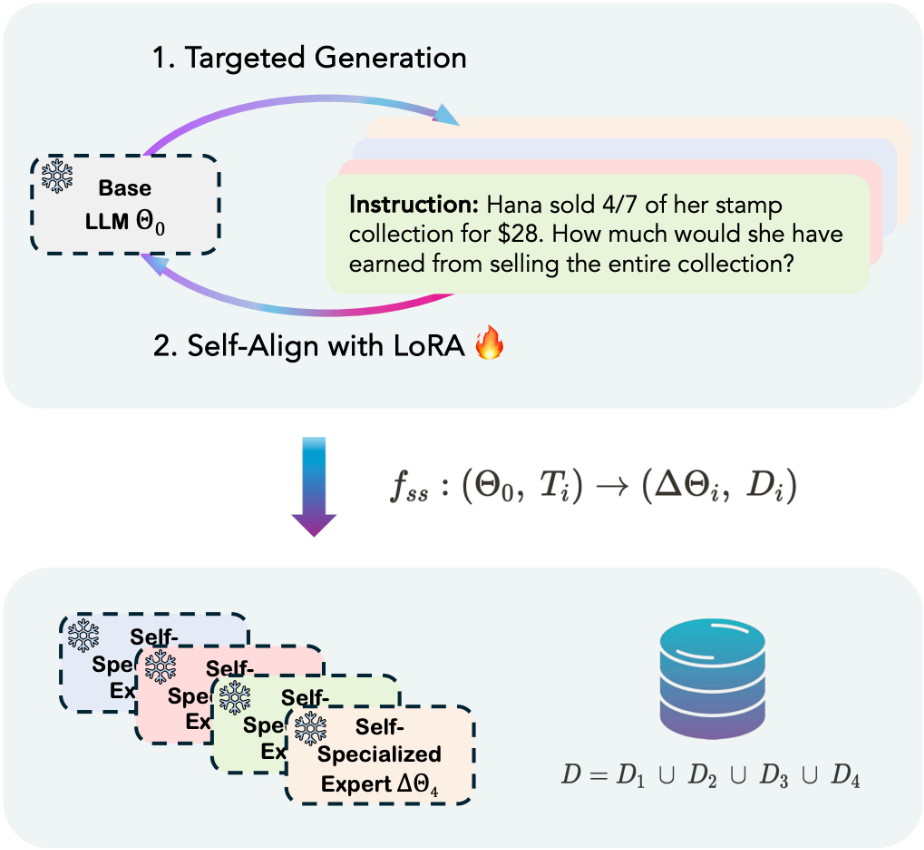
Adaptable

Interpretable

Approach

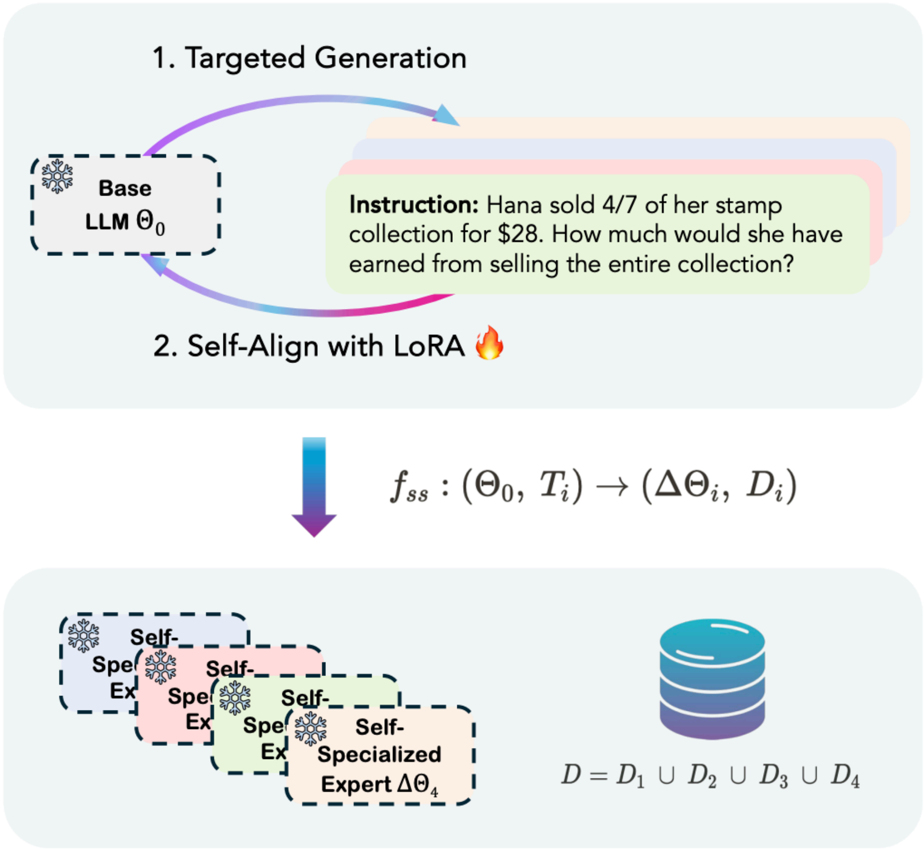
Overview of Self-MoE

Self-Specialization

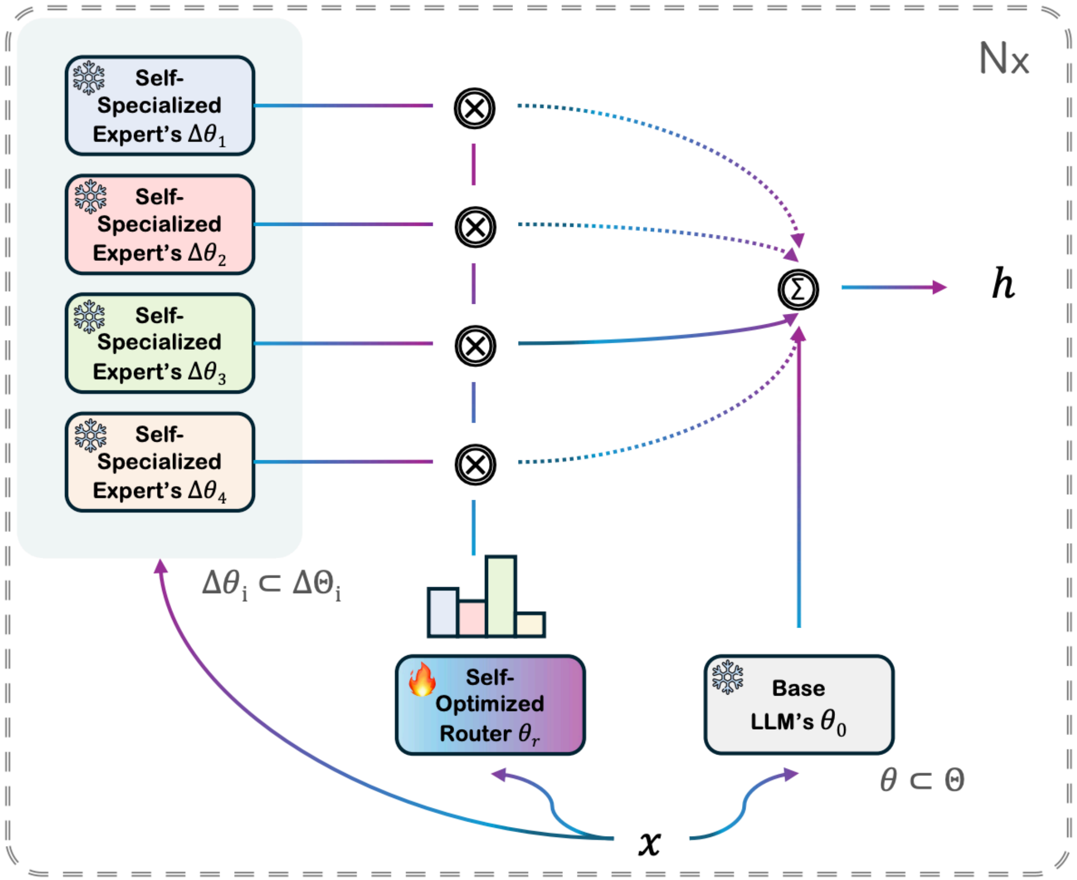


Overview of Self-MoE

Self-Specialization



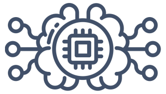
MiXSE (MiXture of Self-Specialized Experts)



Approach

Self-Specialization for Uncovering Domain Expertise

Domain-Specific Seed Instructions (e.g., 80)



Base LLM M_{base}

1. Domain-Specific Instruction Generation

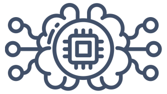
Instruction: Generate a list of drugs which can be used for the treatment of the given symptom.

Instruction: Given medication records, predict possible drug-drug interactions.

Instruction: You are given data of genetic variations and mutations, generate a comprehensive report.

⋮

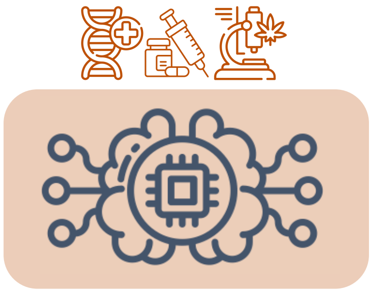
Instruction: Provide an answer to the following question about the patient's medical history.



Base LLM M_{base}

2. Domain-Specific Response Generation

(Instruction, Input, Output)



3. Triggering Specialization

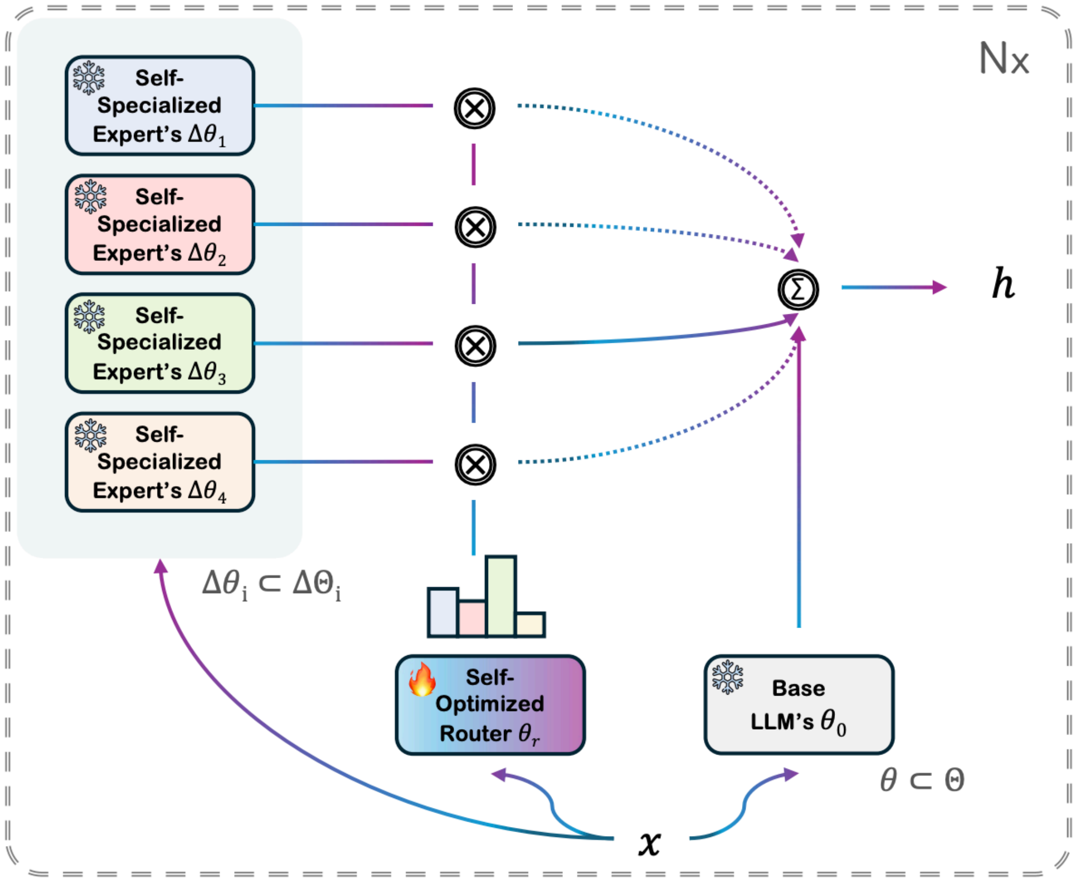
Self-Specialization: Uncovering Latent Expertise within Large Language Models. ACL 2024 Findings

Overview of Self-MoE

$\theta' = \theta + \Delta\theta = \theta + \theta_B\theta_A$

$h = \theta x + \theta_B\theta_A x$ for a single expert LoRA

MiXSE (**MiX**ture of **S**elf-Specialized **E**xperts)



Overview of Self-MoE

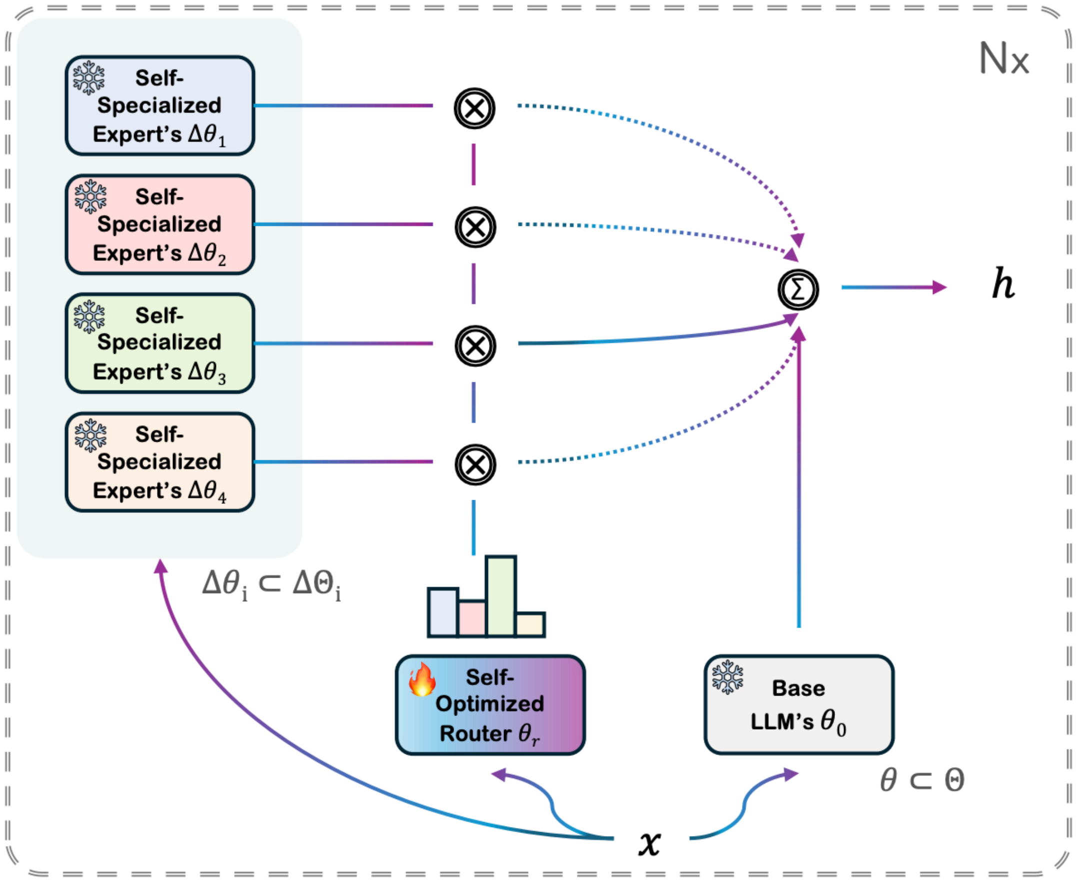
$$\theta' = \theta + \Delta\theta = \theta + \theta_B\theta_A$$

$$h = \theta x + \theta_B\theta_A x$$ for a single expert LoRA

Each Expert 1, 2, 3, 4

$$h = \theta x + \sum \alpha_i \Delta\theta_i x$$
$$= \theta x + \sum \alpha_i \theta_{Bi}\theta_{Ai} x$$

MiXSE (MiXture of Self-Specialized Experts)



Overview of Self-MoE

$$\theta' = \theta + \Delta\theta = \theta + \theta_B\theta_A$$

$h = \theta x + \theta_B\theta_A x$ for a single expert LoRA

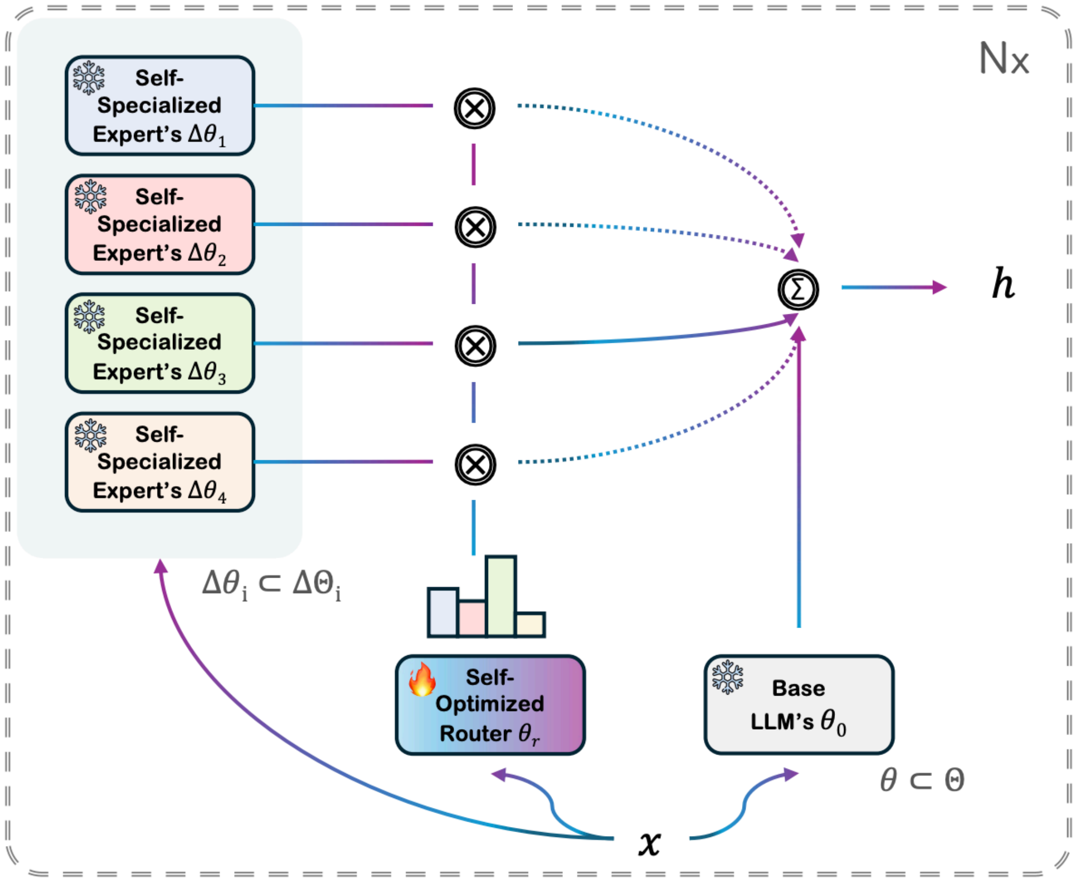
Each Expert 1, 2, 3, 4

$$h = \theta x + \sum \alpha_i \Delta\theta_i x$$
$$= \theta x + \sum \alpha_i \theta_{Bi}\theta_{Ai} x$$

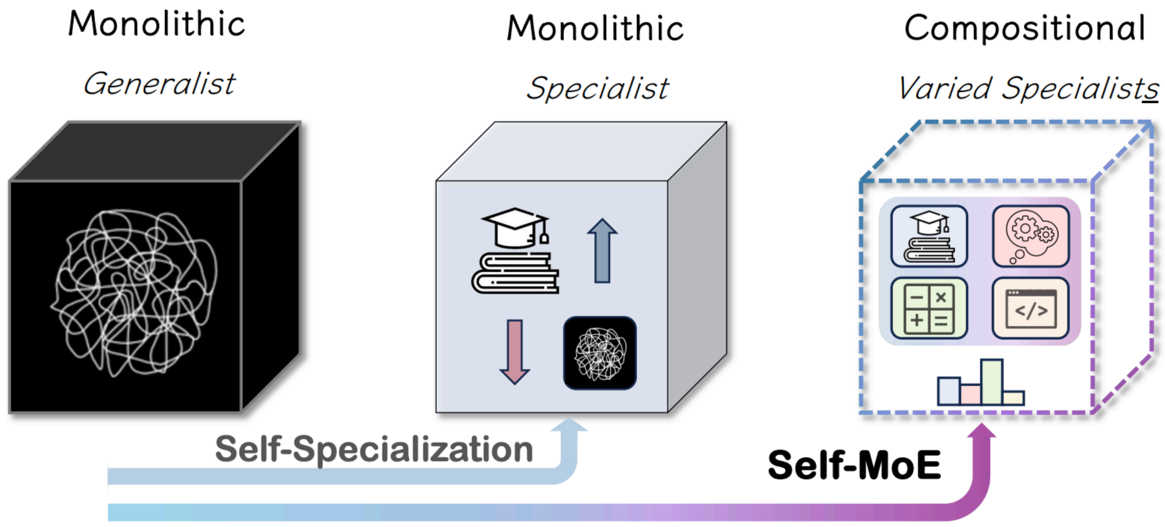
$$\alpha = softmax(\theta_r x)\ (topk)$$

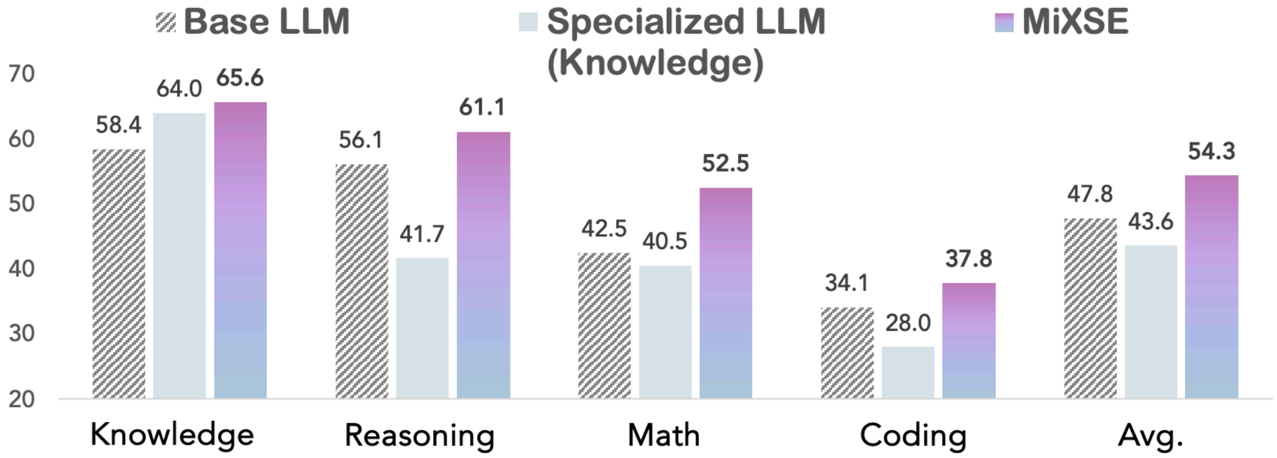
Routing Weights

MiXSE (MiXture of Self-Specialized Experts)

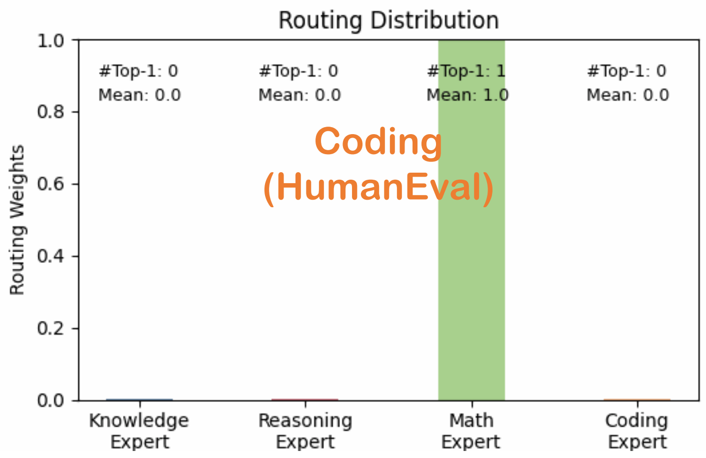


Improved Capabilities across All Domains

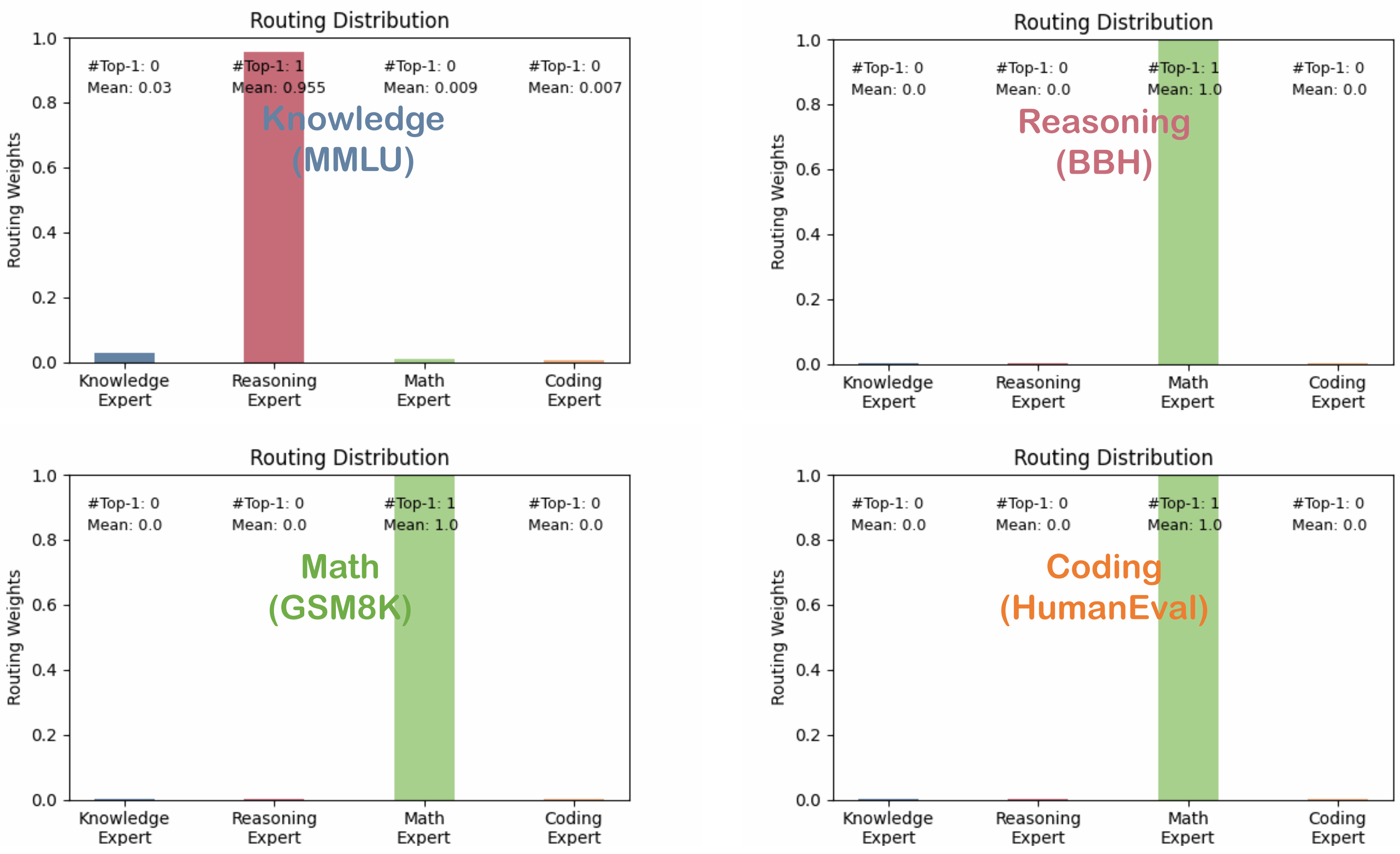




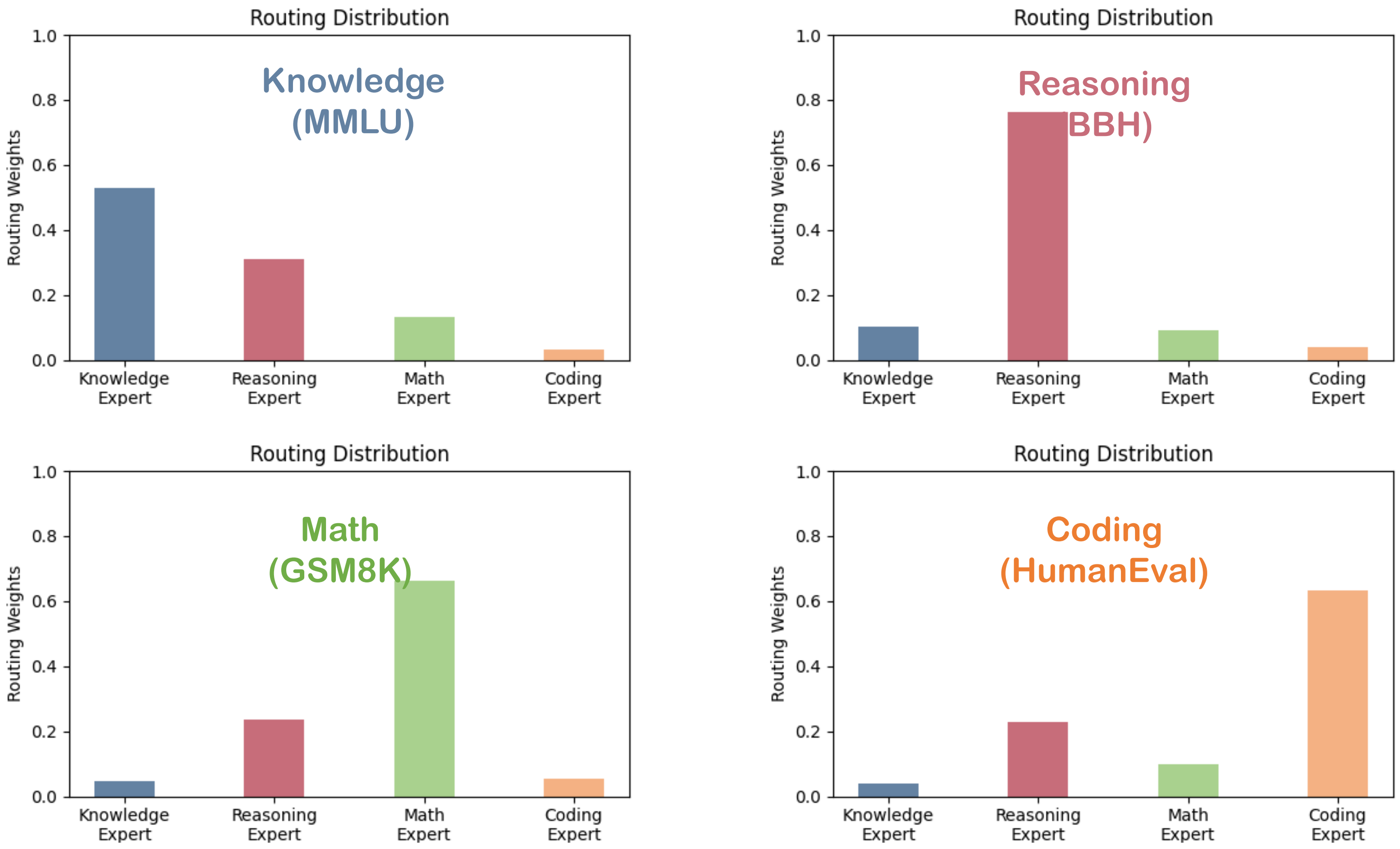
Routing Analysis (Token-by-Token)



Routing Analysis (Token-by-Token)



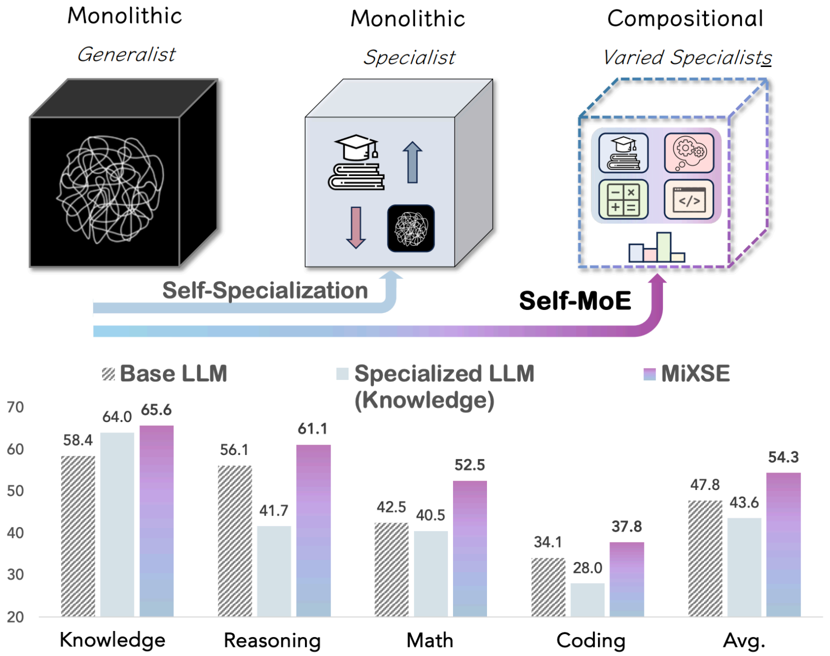
Routing Analysis (Mean over Tokens)



Conclusion

Key Takeaways

Q. Can we build compositional LLMs that achieve uncompromising multiple expertise with minimal resources?



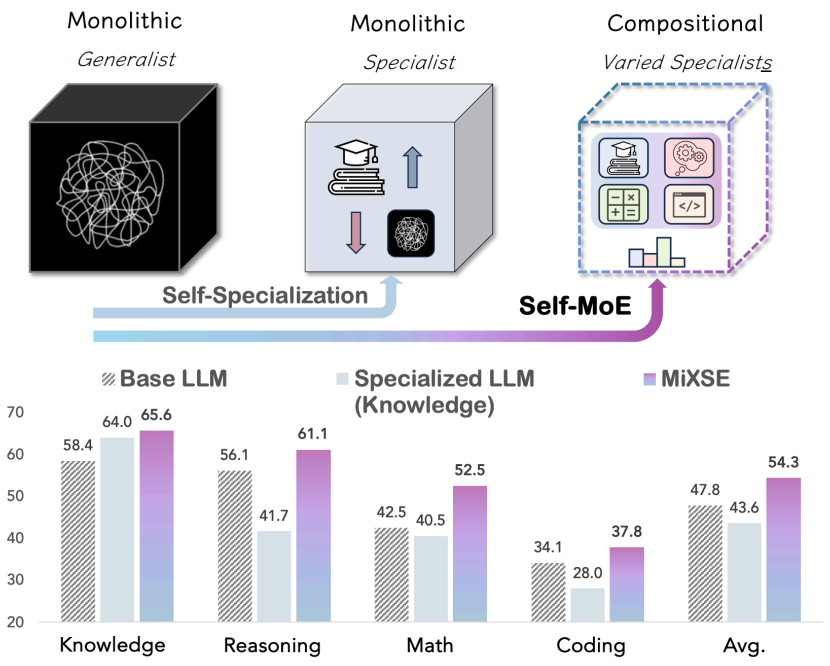
1. Highlighting Limitations of Monolithic Models

Focusing on a specific capability often comes at the cost of degrading performance in other domains

Conclusion

Key Takeaways

Q. Can we build compositional LLMs that achieve uncompromising multiple expertise with minimal resources?



1. Highlighting Limitations of Monolithic Models

Focusing on a specific capability often comes at the cost of degrading performance in other domains

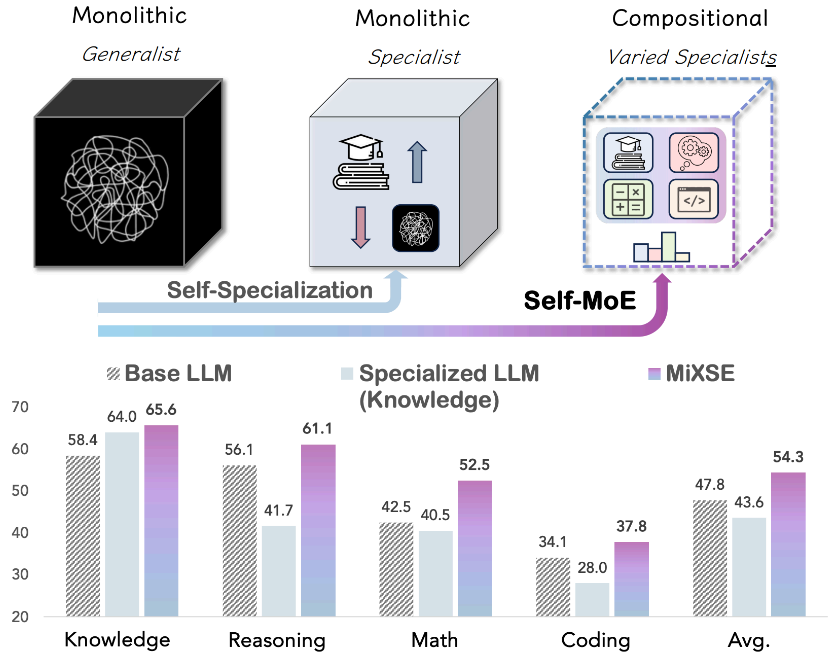
2. Introducing Self-MoE for Modular Specialization

Self-MoE transforms a monolithic LLM into a lightweight, modular system of self-specialized experts, without requiring extensive human supervision, compute resources, or added overhead in active parameters.

Conclusion

Key Takeaways

Q. Can we build compositional LLMs that achieve uncompromising multiple expertise with minimal resources?



1. Highlighting Limitations of Monolithic Models

Focusing on a specific capability often comes at the cost of degrading performance in other domains

2. Introducing Self-MoE for Modular Specialization

Self-MoE transforms a monolithic LLM into a lightweight, modular system of self-specialized experts, without requiring extensive human supervision, compute resources, or added overhead in active parameters.

3. Findings

- Consistent improvement over a base LLM, outperforming various baselines
- Ablation studies validate the impact of modularity, routing strategies, and self-generated synthetic data
- Analyses explore routing distributions, forgetting issues, and the applicability to various base LLMs

Thank you