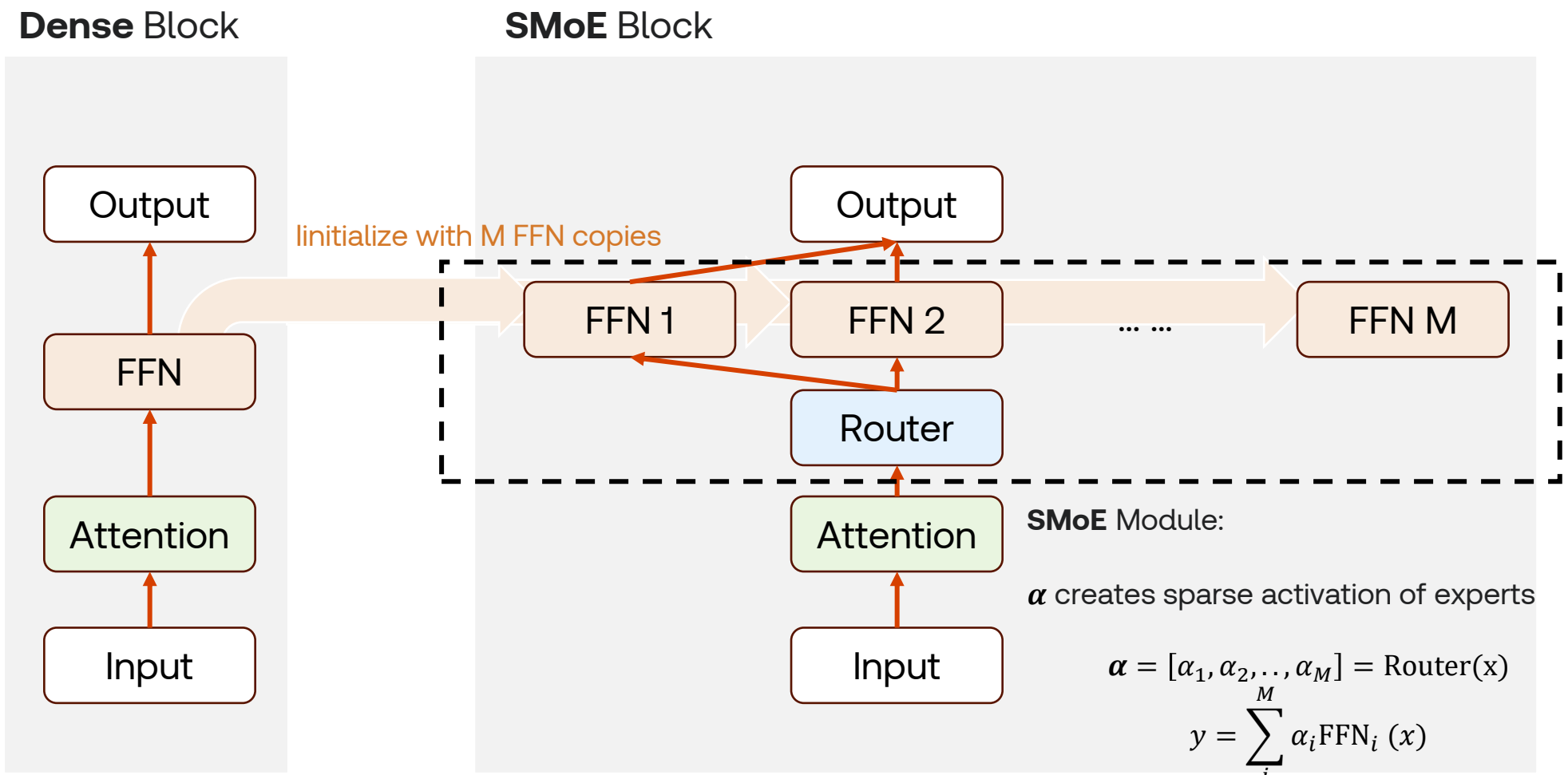
Automatic Expert Discovery in LLM Upcycling via Sparse Interpolated Mixture-of-Experts

Shengzhuang Chen[1], Ying Wei[2]*, Jonathan Richard Schwarz[1]*

[1]Thomson Reuters Foundational Research [2]Zhejiang University

* Joint senior authorship; Equal contribution

What is LLM Upcycling

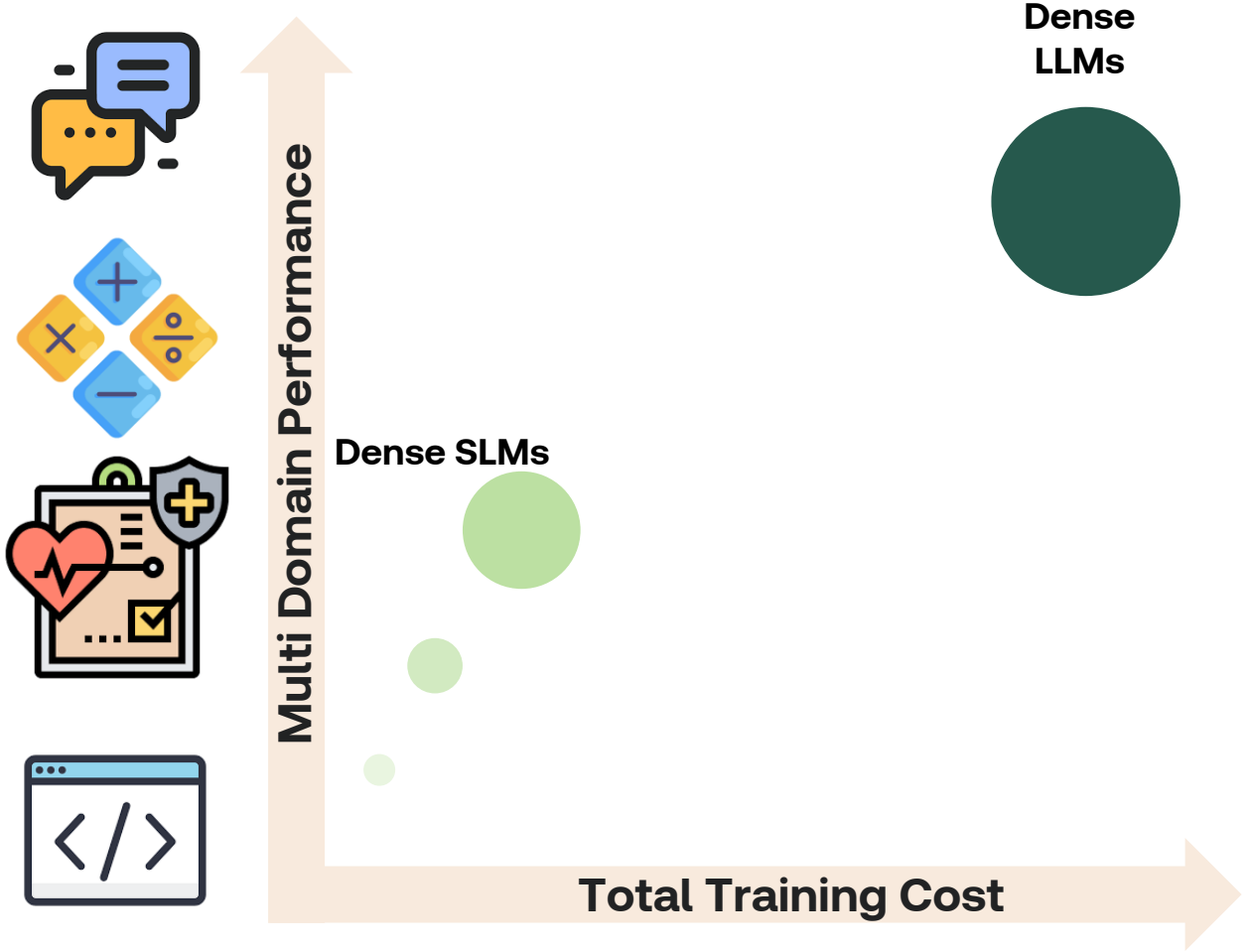


Thomson Reuters™

Why LLM Upcycling is Important

Growing Demand for Specialized LLMs

- Domain-specific applications require tailored model capabilities
- General-purpose models often underperform on domain-specific tasks



Thomson Reuters™

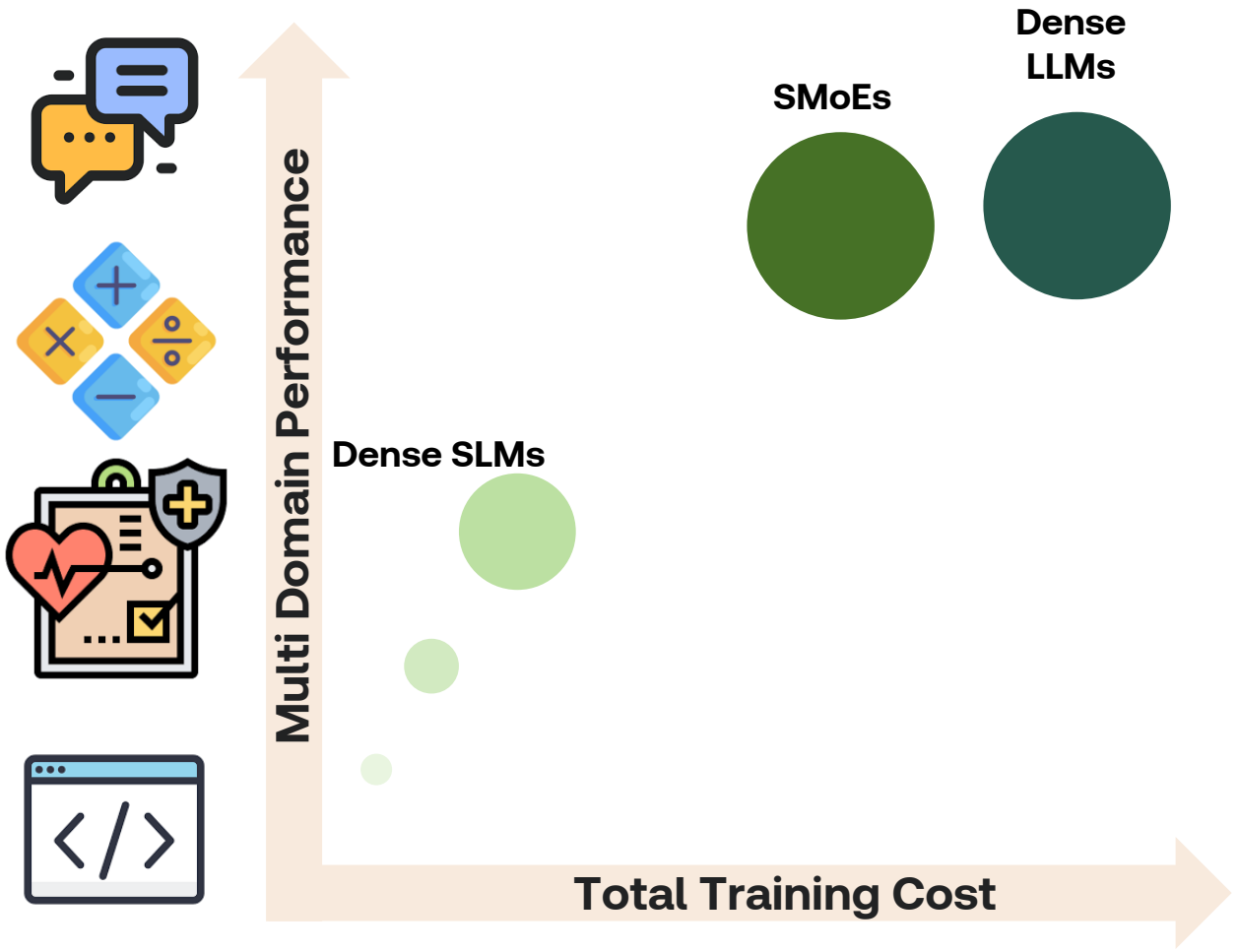
Why LLM Upcycling is Important

Growing Demand for Specialized LLMs

- Domain-specific applications require tailored model capabilities
- General-purpose models often underperform on domain-specific tasks

The Promise of SMoEs

- Flexible capacity scaling without proportional compute increase
- Specialized experts for different domains/tasks

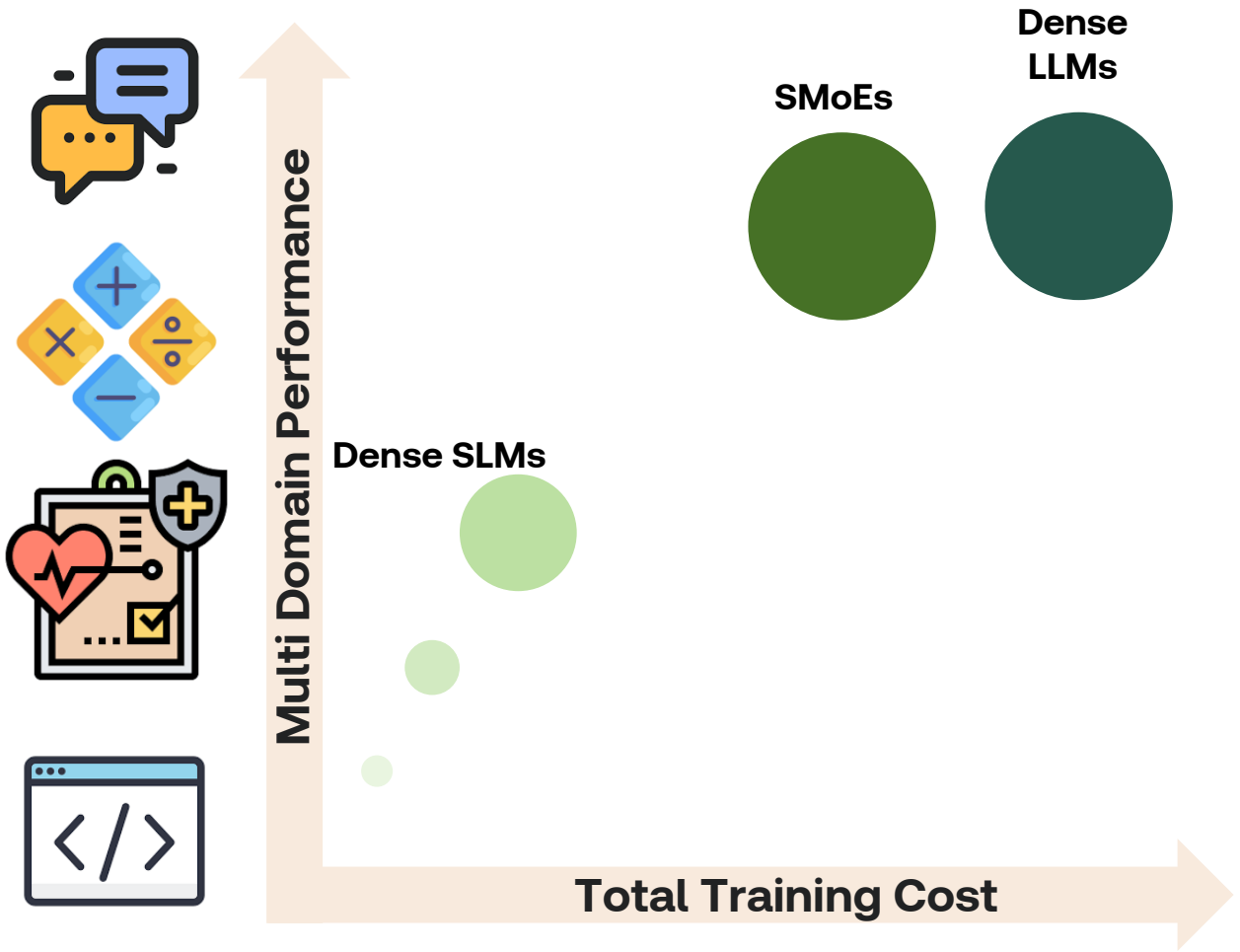


Thomson Reuters™

Why LLM Upcycling is Important

The Scaling Challenge in SMoEs

- Training from scratch: enormous computational cost per model

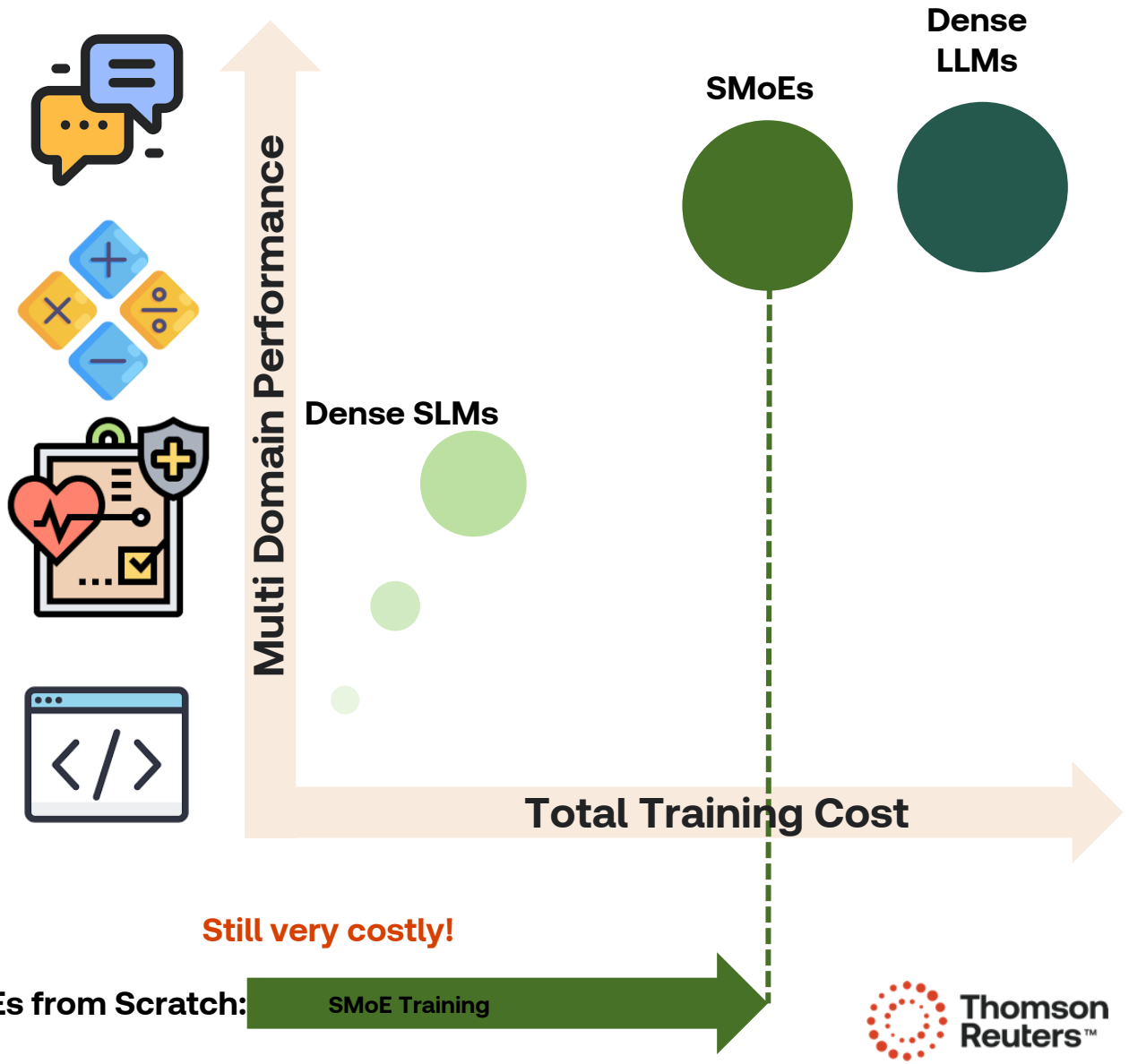


Thomson Reuters™

Why LLM Upcycling is Important

The Scaling Challenge in SMoEs

- Training from scratch: enormous computational cost per model



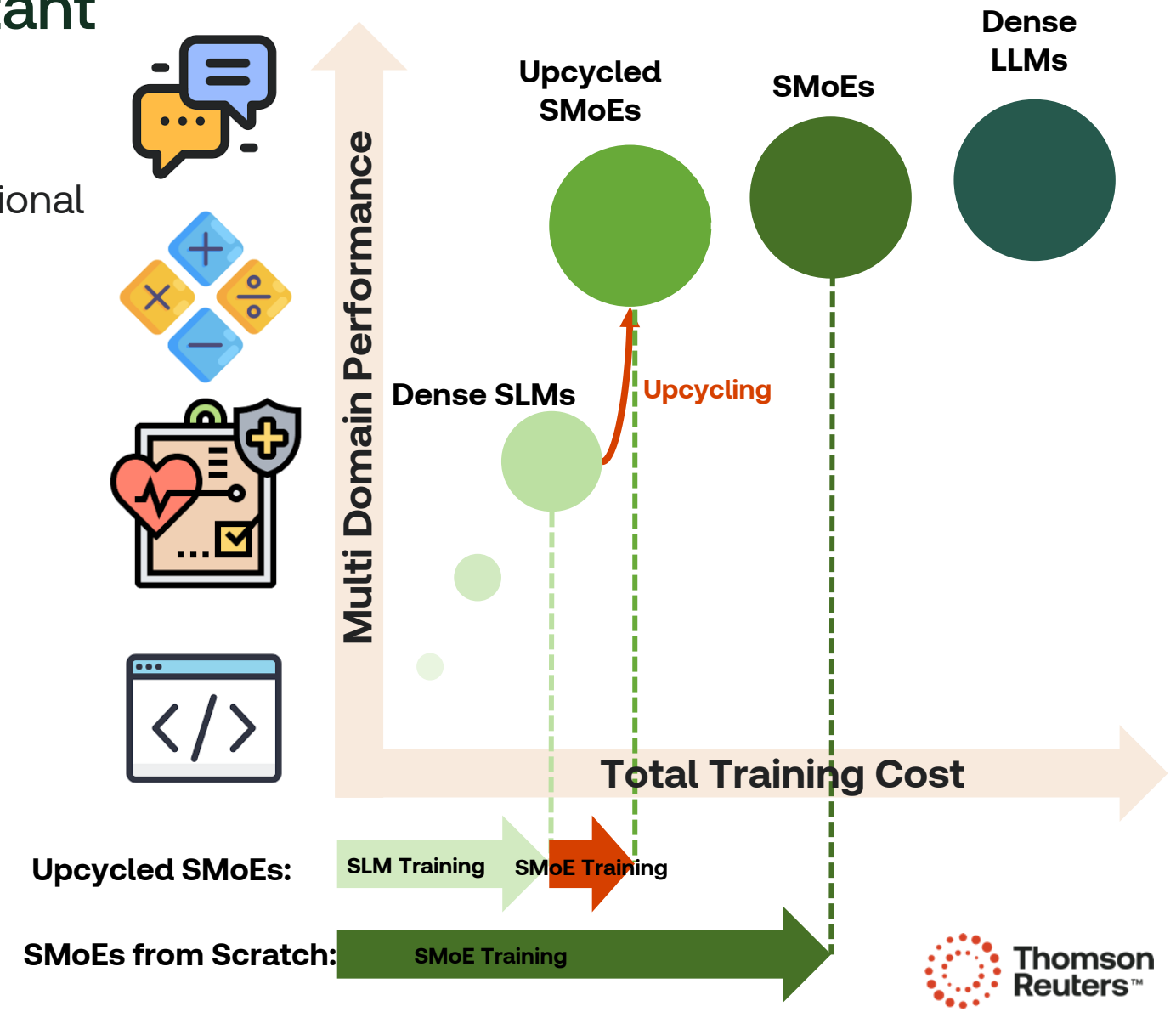
Why LLM Upcycling is Important

The Scaling Challenge in SMoEs

- Training from scratch: enormous computational cost per model

The Upcycling Opportunity

- Convert existing dense LLMs → SMoE architectures through post-training
- Leverage pre-trained knowledge + add specialized capacity



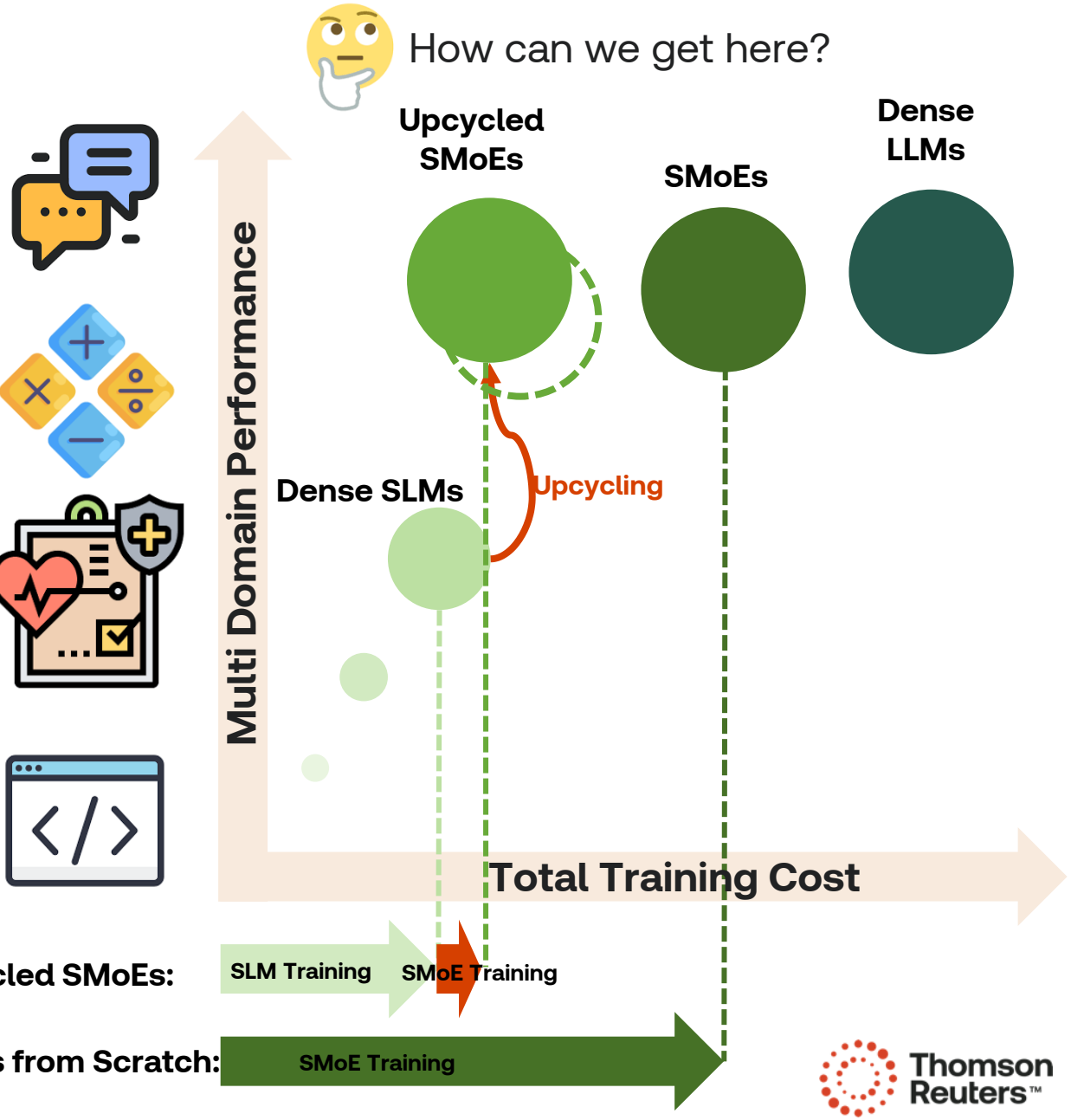
Why LLM Upcycling is Important

The Scaling Challenge in SMoEs

- Training from scratch: enormous computational cost per model

The Upcycling Opportunity

- **Challenge:** How can we keep pushing for a better compute-performance trade-off?



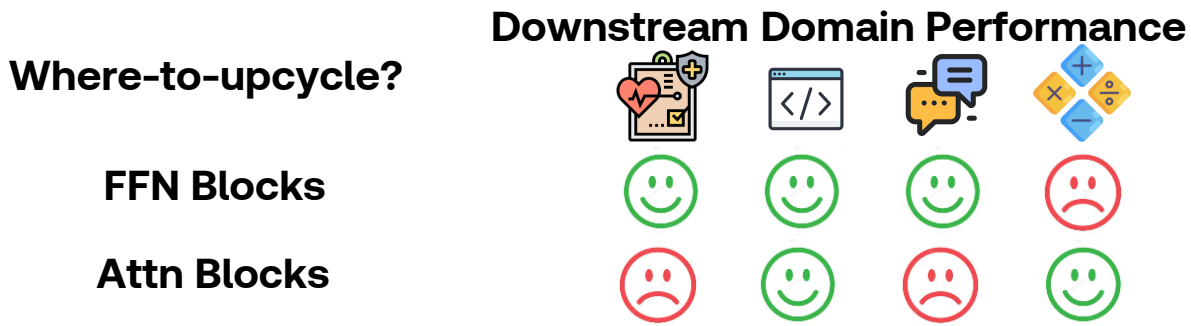
Limitations of Existing LLM Upcycling Methods

Limitation 1: Manual Upcycling Parameter Selection

Current methods commonly rely on heuristic rules to decide "where to upcycle“ in a LLM.

Typically, this involves manual selection of FFN or Attention modules for upcycling into SMoE modules.

- **Neglect Difference in Parameter Importance**
  - Different layers/parameters within the same LLM can have significantly different importance to the model's overall functionality.
- **Neglect Domain Adaptation Needs**
  - Different domain tasks can require specific optimal upcycling/fine-tuning locations.



Thomson Reuters™

Limitations of Existing LLM Upcycling Methods

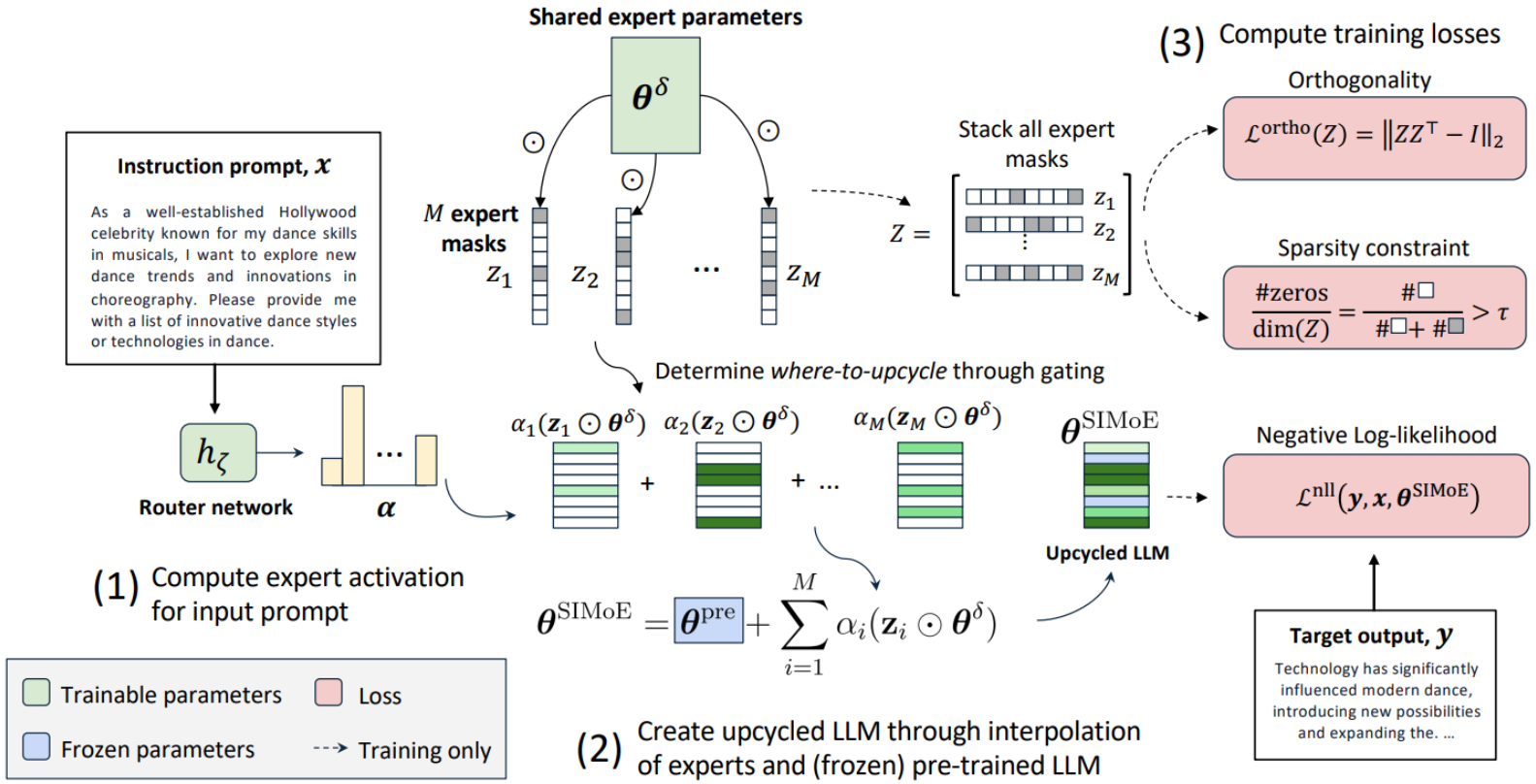
Limitation Lack of Expert Cooperation and Specialization

Current methods lack a systematic mechanism to balance expert specialization and collaborative cooperation.

- **Insufficient Specialization**
  - Traditional SMoE frameworks use fixed shared expert mechanisms to force collaboration.
  - Such designs suppress the specialization of domain experts.
  - In extreme cases, this results in model collapse.
- **Inefficient Collaboration**
  - Some upcycling methods (e.g., BTX[1]) use independent fine-tuning strategies to promote expert specialization.
  - Separate training of multiple domain experts on different datasets, followed by integration into a unified SMoE model.
  - Independent training hinders knowledge transfer, leading to redundant expert parameters.

[1] Sainbayar Sukhbaatar, et al. "Branch-train-mix: Mixing expert LLMs into a mixture-of-experts LLM". In First Conference on Language Modeling.

Sparse Interpolated Mixture-of-Experts (SIMoE)



Automatic Expert Discovery

Sparsity-constrained optimization enables automatic expert discovery through learnable structured sparsity patterns

Balanced Specialization and Cooperation

Parameter sharing + orthogonal penalty balances specialization-cooperation

Thomson Reuters™

Automatic Expert Discovery

Key idea: We propose to formulate "where-to-upcycle" as a **sparsity-constrained optimization** problem.

SIMoE optimizes for "where-to-upcycle" *globally*:

- Introducing learnable binary masks $z_i \in \{0,1\}$ **for each linear layer** in the LLM
- The resultant SIMoE Layer Formulation

$$y = f_{\theta^{\mathrm{SIMoE}}}(x),$$

$$\theta^{\mathrm{SIMoE}} = \underbrace{\theta^{\mathrm{pre}}}_{\text{冻结基座}} + \sum_{i=1}^{M} \alpha_i \cdot \underbrace{\mathbf{z}_i \odot \theta_i^{\delta}}_{\text{稀疏专家增量}}$$

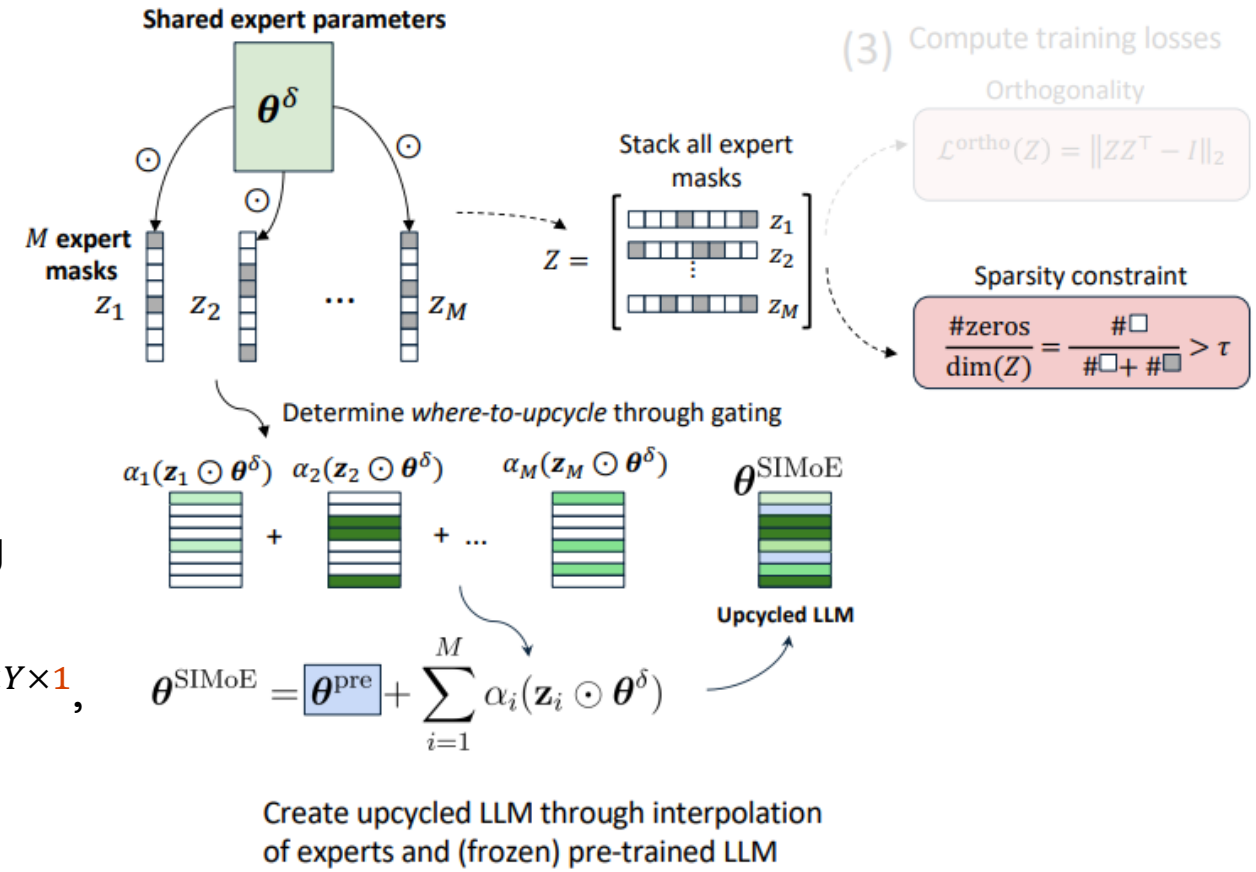
Automatic Expert Discovery

SIMoE Layer Formulation

$$\theta^{\text{SIMoE}} = \underbrace{\theta^{\text{pre}}}_{\text{冻结基座}} + \sum_{i=1}^{M} \alpha_i \cdot \underbrace{\mathbf{z}_i \odot \theta_i^{\delta}}_{\text{稀疏专家增量}}$$

Neuron-level Sparsity

- Traditional expert parameter mask $Z \in \{0,1\}^{\boldsymbol{M}\times Y\times X}$ leads to **M**-fold increase in training parameters.
- SIMoE enforces structued sparsity $Z \in \{0,1\}^{\boldsymbol{M}\times Y\times 1}$, masking input neurons.
- Ensures both **fine-grained control** and **computational feasibility** in learning "where-to-upcycle"



Thomson Reuters™

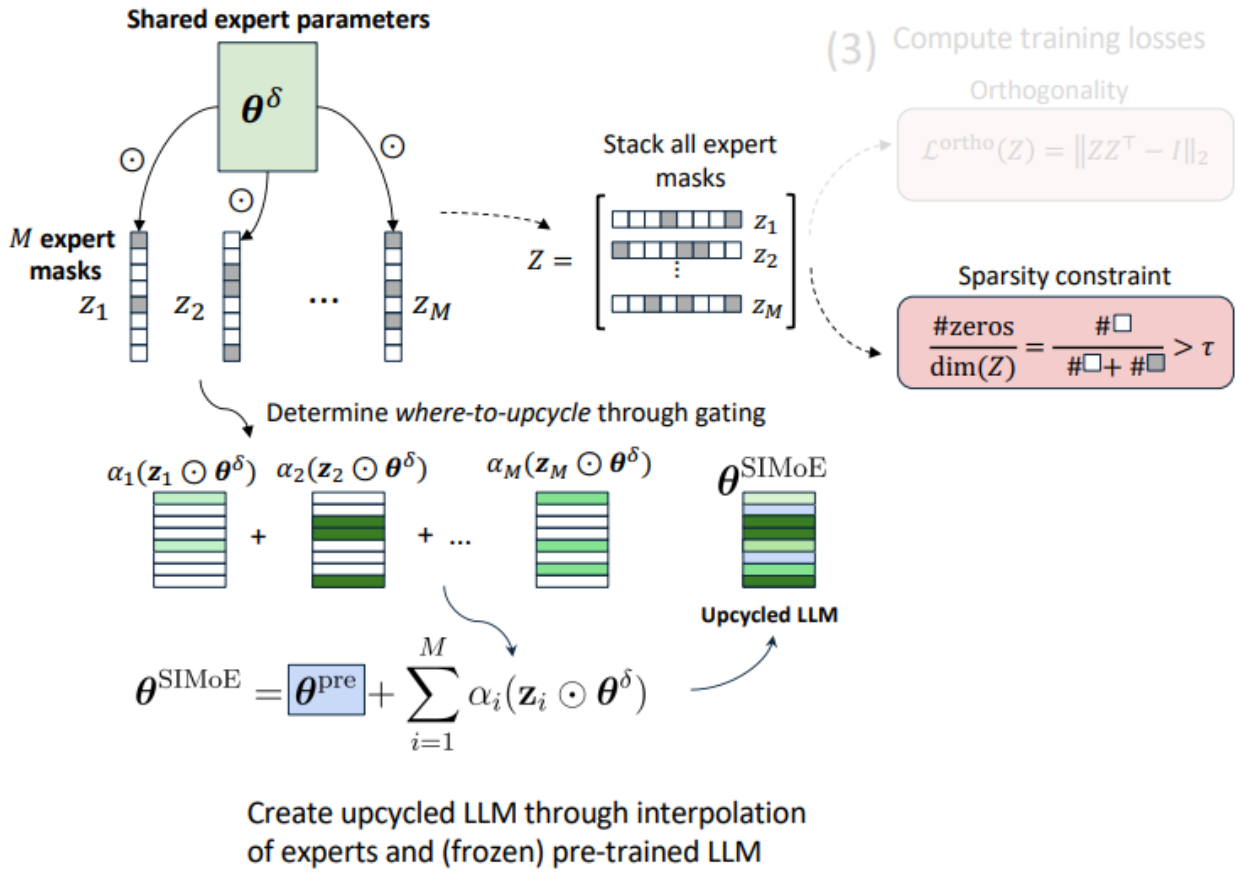
Automatic Expert Discovery

SIMoE Layer Formulation

$$\theta^{\text{SIMoE}} = \underbrace{\theta^{\text{pre}}}_{\text{冻结基座}} + \sum_{i=1}^{M} \alpha_i \cdot \underbrace{\mathbf{z}_i \odot \theta_i^{\delta}}_{\text{稀疏专家增量}}$$

Prevent Catastrophic Forgetting

- **Freeze** pre-trained model parameters θ^{pre}
- Ensures **parameter updates remain sparse additions to**– rather than replacements of–pre-trained functionality.
- Mitigating catastrophic forgetting while maintaining the same model expressiveness.



Create upcycled LLM through interpolation of experts and (frozen) pre-trained LLM

Thomson Reuters™

Balanced Specialization and Cooperation

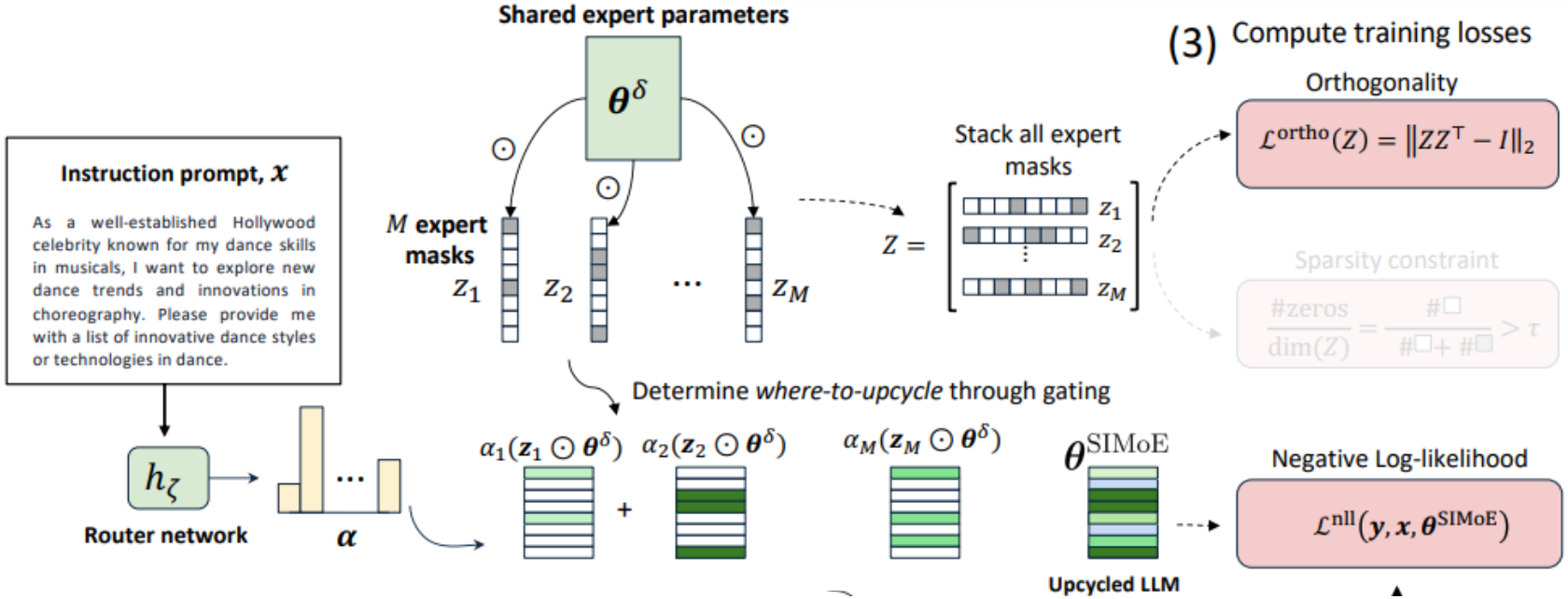
Promote Cooperation through Parameter Sharing

- **Shared parameters θ^{δ}** for all experts
- Weighted merge of experts in the parameter space via a learned router network h_{ζ}

Ensure Specialization through Orthogonal Masks

- **Distinct sparse masks,** $\{z_i\}_{i=1}^{M}$, for experts
- Orthogonal Penalty on Masks

$$\min_{\theta^{\delta},Z,\zeta} \max_{\lambda \geq 0} \mathcal{L}_{\text{nll}} + \underbrace{\beta \|\mathbf{Z}\mathbf{Z}^{\top} - \mathbf{I}\|_2}_{\text{正交约束}} + \underbrace{\lambda(\tau - (1 - \bar{L}_0(\mathbf{Z})))}_{\text{稀疏约束}}$$



Thomson Reuters™

Sparsity-Constrained Optimization

SIMoE Training Objective

- **Minimize:** $L^{\text{nll}}(y, x, \theta^{\text{SIMoE}}) + \beta L^{\text{ortho}}(Z)$
- **Subject to the sparsity constraint :** $1 - L_0(Z) \geq \tau$
  - where $\tau \in [0,1]$ is the target sparsity level

Practical Implementation for Solving the Problem

- We perform simultaneous gradient ascent and descent:

$$\min_{\theta^{\delta}, Z, \zeta} \max_{\lambda \geq 0} \mathcal{L}_{\text{nll}} + \underbrace{\beta \|\mathbf{Z}\mathbf{Z}^{\top} - \mathbf{I}\|_2}_{\text{正交约束}} + \underbrace{\lambda(\tau - (1 - \bar{L}_0(\mathbf{Z})))}_{\text{稀疏约束}}$$

- Pruning of non-essential expert parameters during training, as sparsity level increases.
- Optimized solution is guaranteed to have sparsity at at least τ.
- **Precise control of # activated parameters**.

Experimental Validation

SNI Dataset

- 1616 instructed NLP tasks
- Training on 64 distinct task categories
- Testing on 12 unseen categories
- Cross-task generalization evaluation

Tülu-v3 Dataset

- Large-scale instruction-tuning dataset
- 939,343 unique training instances
- Multi-domain evaluation

Model Configuration

- 8 experts maximum
- $\tau = 75\%$ sparsity constraint
- Llama3 backbone models

Superior Cross-Task Performance

Key Achievements

- **3B Model:** +2.5% improvement over Full FT
- **8B Model:** +1.6% improvement over Full FT
- **Consistency:** Wins in 7+ out of 12 task categories

Detailed Results

- Complete SNI benchmark results showing performance across all 12 unseen task categories

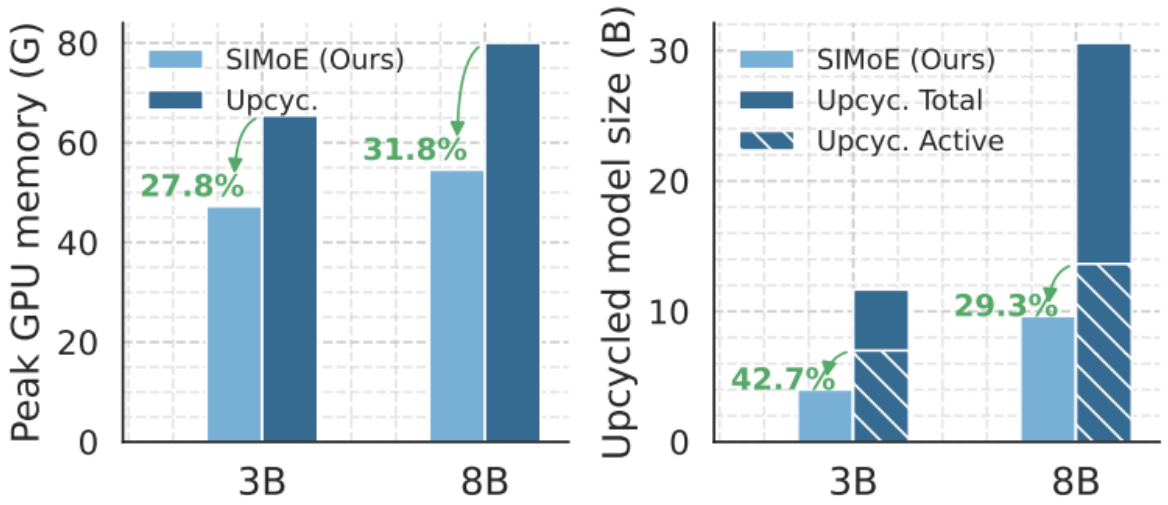
Seed LLM	Method	TitleGen.	Coref. Res.	Text. Entail.	Quest. Rewrit.	Cause Eff. Class.	Dialog Act Recog.	Ans. Class.	Keyword Tag.	Data to Text	Word Analogy	Overlap Extr.	Grammar Corr.	Avg. (↑)
Llama3.2 3B	Full FT	40.20	55.33	58.80	67.60	**70.52**	62.38	**68.13**	59.60	**52.08**	39.50	66.35	**88.68**	60.76
	Upcyc.	**41.25**	57.54	62.28	67.97	68.82	66.14	67.23	63.61	51.21	46.17	62.05	87.86	61.84
	SIMoE	41.14	**57.67**	**63.17**	**68.08**	69.54	**68.31**	67.59	**67.87**	51.40	**48.50**	**68.27**	87.64	**63.26**
Llama3 8B	Full FT	41.35	57.20	64.46	67.35	71.05	**73.17**	67.14	66.58	**53.04**	52.71	66.93	**87.82**	64.07
	Upcyc.	41.95	62.24	64.45	68.49	73.40	69.06	66.93	66.61	52.30	**55.55**	72.05	87.59	65.05
	SIMoE	**43.04**	**64.37**	**66.49**	**68.86**	**76.40**	70.70	**68.92**	**67.79**	51.73	52.79	**72.06**	85.38	**65.71**

Scalability and Efficiency

SOTA Performance on Tülu-v3

Method	MMLU	PopQA	Truthful QA	BBH	DROP	MATH	GSM8K	Human Eval	Human Eval+	IFEval	Alpaca Eval 2	Safety	Avg. (↑)
Tülu v2 8B SFT	61.8	23.3	49.4	57.1	**61.7**	14.0	60.4	66.9	63.1	42.3	8.9	70.7	48.3
RLHFlow v2 SFT	65.8	29.7	**56.0**	69.3	57.2	**35.7**	**81.6**	86.2	80.9	52.7	**13.6**	43.5	56.0
MAmmoTH2 8B	63.6	20.8	42.7	63.4	43.8	30.5	63.7	72.8	66.4	34.9	6.5	47.8	46.4
Tülu v3 8B SFT	65.9	29.3	46.8	67.9	61.3	31.5	76.2	86.2	**81.4**	72.8	12.4	93.1	60.4
BTX	64.5	**30.9**	48.9	69.0	58.9	33.0	80.9	85.2	80.9	73.1	11.7	93.4	60.9
SIMoE (Ours)	**66.5**	28.7	51.6	**69.5**	57.5	30.1	81.3	**86.5**	81.3	**74.1**	12.4	**94.8**	**61.1**

Parameter Efficiency and Memory Savings

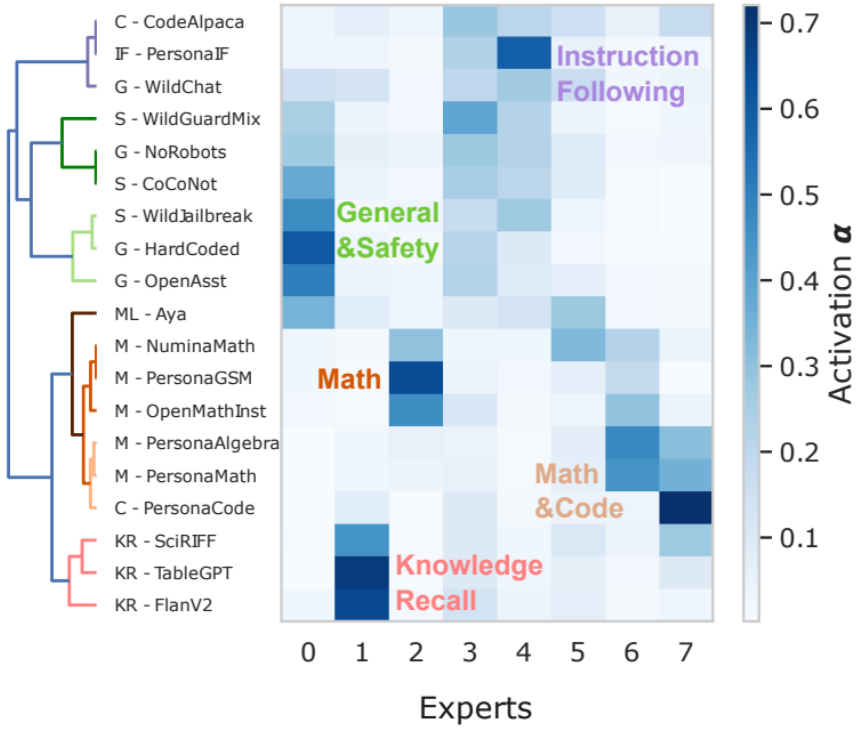


Thomson Reuters™

Interpretable Expert Discovery and Specialization

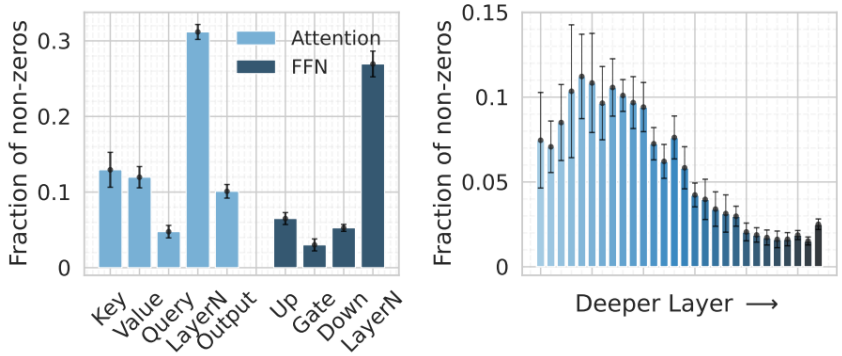
Domain Discovery

- Task similarity dendrogram showing automatic domain clustering

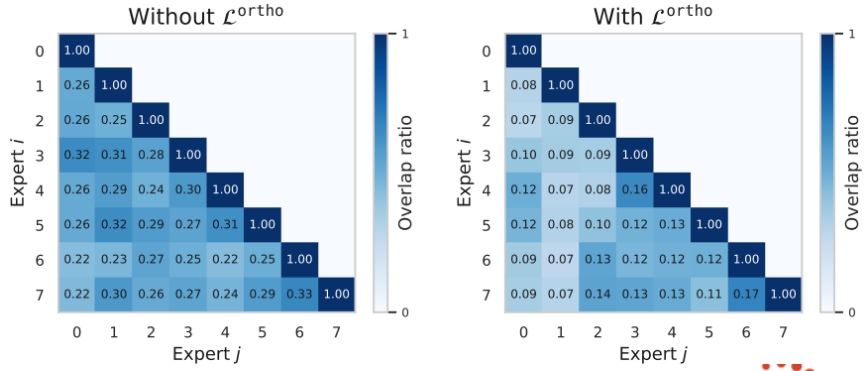


Sparsity Patterns

- Learned upcycling patterns by layer depth and type



- Expert overlap analysis showing balanced specialization



Thomson Reuters™

Importance of SIMoE Components

Ablated Variants

(a) We **upcycle only the FNN layers**.
(b) We **exclude the Orthogonal Penalty on expert masks.**
(c) We **do not impose Sparsity Constraints on the masks z**, allowing them to be freely optimized as standard trainable parameters with values on the full real axis.
(d) We replace Instance-level Routing with **token-level routing in the SIMoE module**.

Model	L.U.	O.P.	S.C.	I.R.	ROUGE-L(↑)	Params.(↓) (B)
(a)		✓	✓	✓	61.52	4.01
(b)	✓		✓	✓	62.67	4.01
(c)		✓		✓	62.54	6.08
(d)	✓	✓	✓		62.51	4.01
SIMoE	✓	✓	✓	✓	**63.26**	**4.01**

Hyperparameter Sensitivity Analysis

SIMoE **consistently outperforms the best baseline score of 65.05** across all evaluated hyperparameter settings, highlighting its robustness and reliability

Orthogonal Penalty

- Extreme values of β lead to suboptimal outcomes due to excessive or minimal overlap among experts, which either impedes specialization or limits combinatorial generalization capabilities.

Orthogonality β	0	$5e^{-6}$	$\mathbf{5e^{-5}}$	$5e^{-4}$
Avg. ROUGE-L (↑)	65.56	65.71	**65.77**	65.31
Approx. Expert Overlap %	25	11	7	2

Target Sparsity

- Low and high extremes of τ result in performance degradation, either through parameter redundancy or excessive sparsity, which constrains model capacity.

Sparsity τ	0	0.5	**0.75**	0.9
Avg. ROUGE-L (↑)	65.37	65.31	**65.71**	65.52

Summary

SIMoE Enables Efficient and Effective LLM Upcycling Without Manual Intervention

- First automatic framework for where-to-upcycle determination
- Novel parameter sharing + orthogonal penalty design
- Significant performance gains with double-digit compute savings

Limitation and Future Directions

- Multimodal applications
- Theoretical analysis

Background: BTX

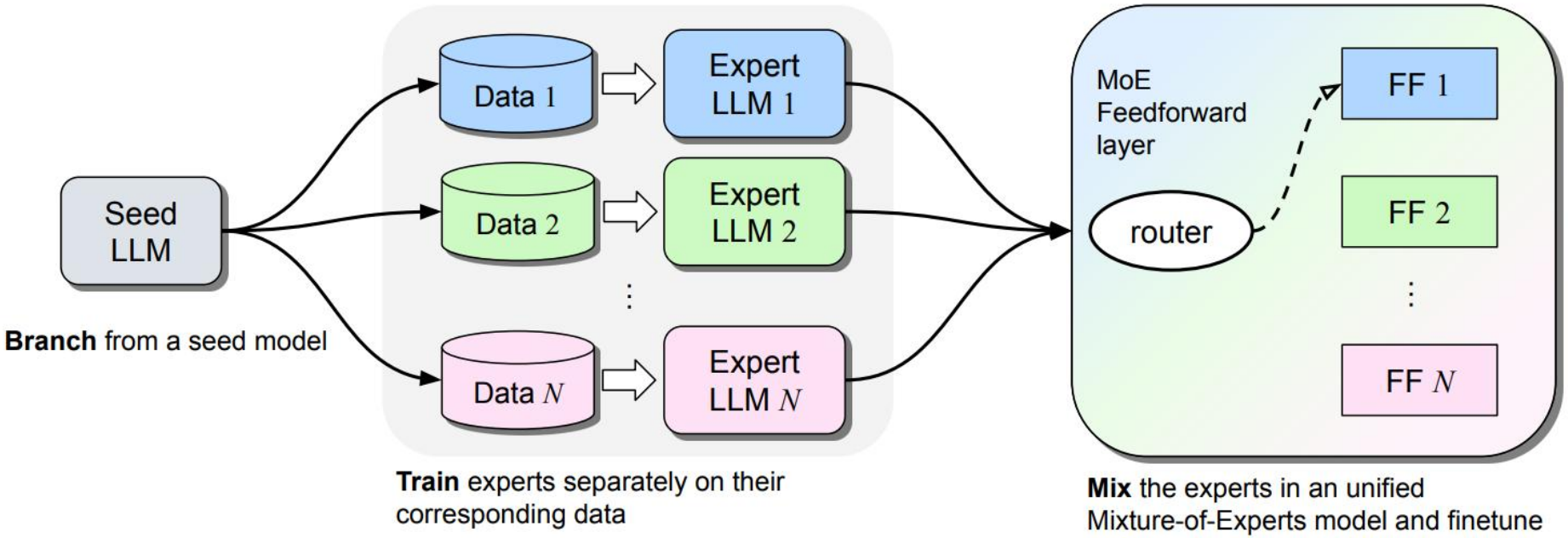


Figure 1 The Branch-Train-MiX (BTX) method has three steps: **1) branch** from a pretrained seed LLM by making multiple copies of it; **2) train** those copies separately on different subsets of data to obtain expert LLMs; **3) mix** those expert LLMs by combining them into a single LLM using mixture-of-experts feedforward (FF) layers, and finetuning the overall unified model.

Thomson Reuters™