香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

ICML International Conference On Machine Learning

Don't Reinvent the Wheel, Just Realign the Spokes: Resource-Efficient Federated Fine-Tuning via Rank-Wise Expert Assembly

Yebo Wu [1 *] , Jingguang Li [2*] , Zhijiang Guo [3 4 †] , Li Li [1 †]

University of Macau [1] KAIST [2] HKUST [3] HKUST (Guangzhou) [4]

香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Motivation

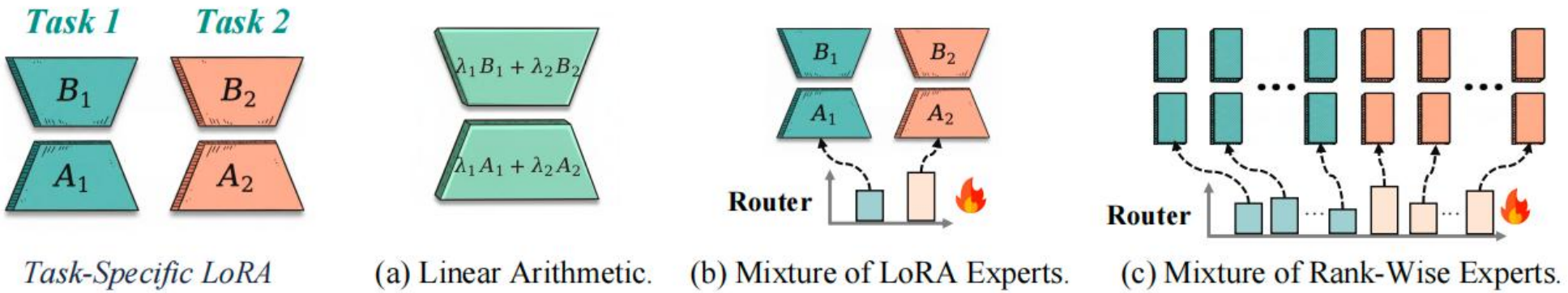
(a) Classic Federated Fine-Tuning Framework.

(b) Our SmartFed Framework.

- **Existing Work:** Train LoRA adapters from scratch on edge devices, leading to high computation, communication, and energy costs.
- Question: *Can we simply reuse existing LoRA modules to adapt LLMs to new tasks, thereby circumventing costly retraining?*
- **Our SmartFed:** Reuse existing LoRA modules, freeze adapter weights, and train only a lightweight rank-wise router for efficient federated adaptation.

Takeaway: From training new adapters to composing existing knowledge.

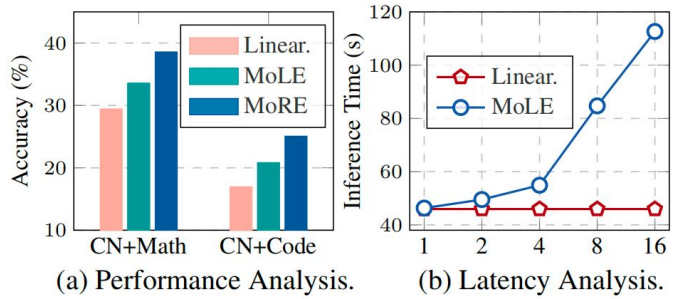
Limitations of Existing LoRA Reuse Methods



Task-Specific LoRA (a) Linear Arithmetic. (b) Mixture of LoRA Experts. (c) Mixture of Rank-Wise Experts.

- **Linear Arithmetic:** Parameter-level merging
  - Limitations: introduces destructive cross-task interference and requires LoRAs with identical dimensions.
- **Mixture of LoRA Experts:** Module-level routing
  - Limitations: activates entire LoRA modules, causing coarse adaptation and poor scalability.
- **Mixture of Rank-Wise Experts:** Rank-level routing
  - Advantages: decomposes LoRAs into fine-grained rank-wise experts and selectively activates task-relevant components.

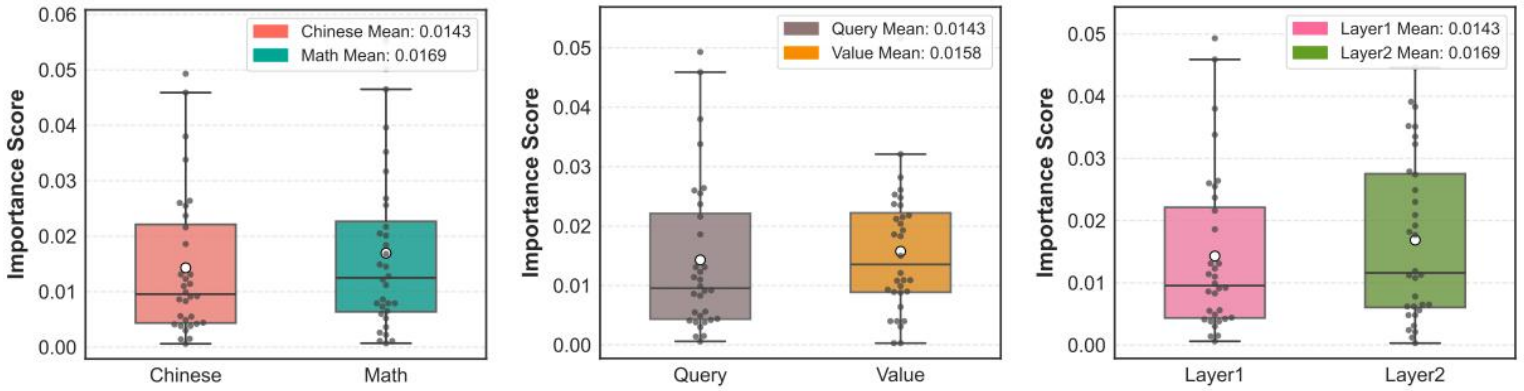
$$\begin{aligned}\mathbf{h}' &= \mathbf{W}_0\mathbf{x} + (\lambda_1\mathbf{B}_1 + \lambda_2\mathbf{B}_2)(\lambda_1\mathbf{A}_1 + \lambda_2\mathbf{A}_2)\mathbf{x} \\ &= \mathbf{W}_0\mathbf{x} + \underbrace{\lambda_1^2\mathbf{B}_1\mathbf{A}_1\mathbf{x} + \lambda_2^2\mathbf{B}_2\mathbf{A}_2\mathbf{x}}_{\text{Target Task Adaptation}} \\ &\quad + \underbrace{\lambda_1\lambda_2\mathbf{B}_1\mathbf{A}_2\mathbf{x} + \lambda_1\lambda_2\mathbf{B}_2\mathbf{A}_1\mathbf{x}}_{\text{Destructive Cross-Task Interference}}.\end{aligned} \tag{3}$$



(a) Performance Analysis. (b) Latency Analysis.

Figure 3. Quantitative analysis of knowledge reuse strategies. (a) Performance comparison across different tasks. (b) Inference latency comparison under varying numbers of LoRA modules.

Heterogeneous Importance of Rank-Wise Experts



(a) Expert importance across tasks. *(b)* Expert importance across matrix types. *(c)* Expert importance across layers.

Figure 4. Heterogeneity of rank-wise expert importance in LLaMA2-7B on the Chinese mathematical reasoning task. (a) Comparison of the first-layer `Query` matrix across Chinese and Math LoRA modules. (b) Comparison between `Query` and `Value` matrices in the first layer (Chinese LoRA). (c) Layer-wise comparison of the `Query` matrix (Layer 1 vs. Layer 2, Chinese LoRA).

Figure 5. Impact of expert quota allocation on model performance.

- **Observation:** Heterogeneous Importance
  - Cross-task variation: experts from different tasks show different importance distributions.
  - Matrix-type sensitivity: Query and Value matrices contribute differently.
  - Layer-wise difference: expert importance varies significantly across layers.
- **Experiments:** Uniform Quota Is Suboptimal
  - Different quota allocations lead to different performance gains, indicating that equal expert budgets are inefficient.

Our Method: SmartFed

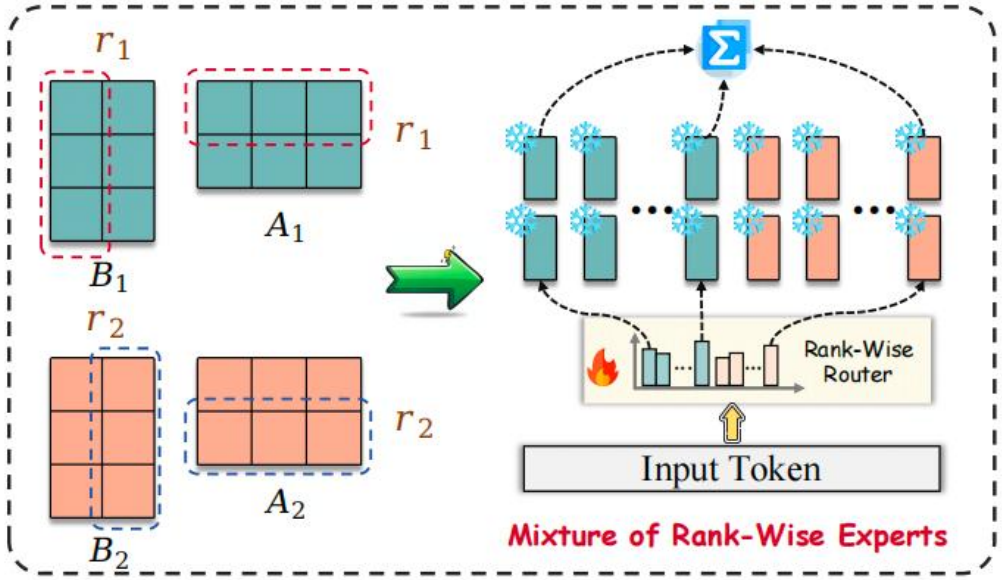


Figure 6. Overview of the Mixture of Rank-Wise Experts (MoRE).

- **Mixture of Rank-Wise Experts:** Fine-Grained LoRA Reuse
  - Decompose each LoRA module into rank-wise experts.
  - Use a lightweight router to select task-relevant experts.
  - Train only the router while keeping backbone and LoRAs frozen.

Target: Enable fine-grained, scalable, and interference-free knowledge reuse.

Our Method: SmartFed

Algorithm 1 The Elastic Expert Quota Allocation Strategy.

Require: Normalized importance scores $\{\alpha_j\}_{j=1}^{J}$, expert budget $B = K \cdot J$, and per-matrix upper bound M.
Ensure: Expert quota for each matrix: $\{\tilde{q}_j\}_{j=1}^{J}$.

Phase 1: Proportional Guarantee
for $j = 1$ **to** J **do**
  $\tilde{q}_j \leftarrow \min\left(\lfloor \alpha_j \cdot B \rfloor, M\right)$ {Allocate base share}
end for
$R \leftarrow B - \sum_{j=1}^{J} \tilde{q}_j$ { Calculate Residual Budget }
Phase 2: Greedy Residual Distribution
$\mathcal{S} \leftarrow \{j \mid \tilde{q}_j < M\}$ {Identify matrices with capacity}
Sort $\mathcal{S}$ by α_j in descending order
for $j \in \mathcal{S}$ **do**
  if $R = 0$ **then**
    break
  end if
  $\Delta \leftarrow \min(M - \tilde{q}_j,\ R)$
  $\tilde{q}_j \leftarrow \tilde{q}_j + \Delta; \quad R \leftarrow R - \Delta$
end for
Return $\{\tilde{q}_j\}_{j=1}^{J}$

- **Elastic Expert Quota Allocation:** Adaptive Resource Allocation
  - Estimate the importance of experts for each parameter matrix.
  - Dynamically assign more expert quota to more important matrices.
  - Avoid wasting computation on less useful experts.

Target: Allocate limited computation budget where it matters most.

Experiments

Evaluation Models.

LLaMA2-7B、LLaMA2-13B、Qwen2-7B

Evaluation Tasks.

Chinese mathematical reasoning、 Chinese code generation、hard math–word problems

Table 4. Overview of skill-composition tasks, corresponding datasets, and evaluation metrics.

Task	Skills Composed	Training Dataset	Testing Dataset	Evaluation Metric
Chinese Mathematical Reasoning	{Chinese, Math}	Math23K	MGSM	Accuracy
Chinese Code Generation	{Chinese, Code}	DoIT	DoIT	Pass@1
Hard Math-Word Problems	{Math, Code}	MathCodeInstruct	GSM-Hard	Execution Accuracy

Overall Performance

Table 1. Performance comparison of different methods on diverse skill-composition tasks across LLMs. *Activated Rank* denotes the average number of activated ranks per parameter matrix. The best results are highlighted in **bold**.

LLM	Method	MGSM	DoIT	GSM-Hard	Average	Activated Rank
LLAMA2-7B	FedIT	30.97	44.96	56.17	44.03 (↓ 6.72)	32
	FwdLLM	32.35	47.18	57.61	45.71 (↓ 5.04)	32
	DoFIT	31.28	46.09	56.89	44.75 (↓ 6.00)	32
	FedAdapter	32.47	48.70	58.05	46.41 (↓ 4.34)	32
	AdaLoRA	33.69	50.13	59.15	47.66 (↓ 3.09)	32
	Linear Arith.	29.43	43.81	54.90	42.71 (↓ 8.04)	32
	LoRAHub	33.41	49.25	58.79	47.15 (↓ 3.60)	32
	MoLE	33.56	49.54	58.69	47.26 (↓ 3.49)	64
	LoRA-Flow	33.75	49.95	59.06	47.59 (↓ 3.16)	64
	SMARTFED	**35.48**	**52.59**	**64.19**	**50.75**	32
LLAMA2-13B	FedIT	43.68	55.00	63.87	54.18 (↓ 8.77)	32
	FwdLLM	47.11	56.92	66.58	56.87 (↓ 6.08)	32
	DoFIT	45.85	56.37	64.91	55.71 (↓ 7.24)	32
	FedAdapter	48.29	58.57	66.79	57.88 (↓ 5.07)	32
	AdaLoRA	50.91	60.41	68.84	60.05 (↓ 2.90)	32
	Linear Arith.	42.24	52.79	63.19	52.74 (↓ 10.21)	32
	LoRAHub	50.43	60.06	68.42	59.64 (↓ 3.31)	32
	MoLE	50.20	60.17	68.36	59.58 (↓ 3.37)	64
	LoRA-Flow	51.12	60.32	68.70	60.05 (↓ 2.90)	64
	SMARTFED	**53.18**	**63.01**	**72.65**	**62.95**	32
QWEN2-7B	FedIT	30.59	44.17	55.04	43.27 (↓ 7.08)	32
	FwdLLM	31.82	46.41	56.52	44.92 (↓ 5.43)	32
	DoFIT	30.74	45.39	55.90	44.01 (↓ 6.34)	32
	FedAdapter	31.99	48.00	57.21	45.73 (↓ 4.62)	32
	AdaLoRA	33.05	49.45	58.01	46.84 (↓ 3.51)	32
	Linear Arith.	29.10	43.05	53.79	41.98 (↓ 8.37)	32
	LoRAHub	33.04	48.30	57.74	46.36 (↓ 3.99)	32
	MoLE	33.11	48.79	57.81	46.57 (↓ 3.78)	64
	LoRA-Flow	33.19	49.23	58.24	46.89 (↓ 3.46)	64
	SMARTFED	**35.31**	**52.11**	**63.62**	**50.35**	32

Main Results:

Finding 1: SmartFed achieves the best performance across all LLM backbones and tasks.

Finding 2: SmartFed consistently outperforms both knowledge-free fine-tuning and LoRA-reuse baselines.

Finding 3: SmartFed uses only 32 activated ranks, but surpasses MoLE and LoRA-Flow with 64 activated ranks.

Conclusion: Fine-grained rank-wise expert routing enables more effective and efficient federated adaptation.

Efficiency Evaluation

Resource Efficiency: Faster, Cheaper, Greener

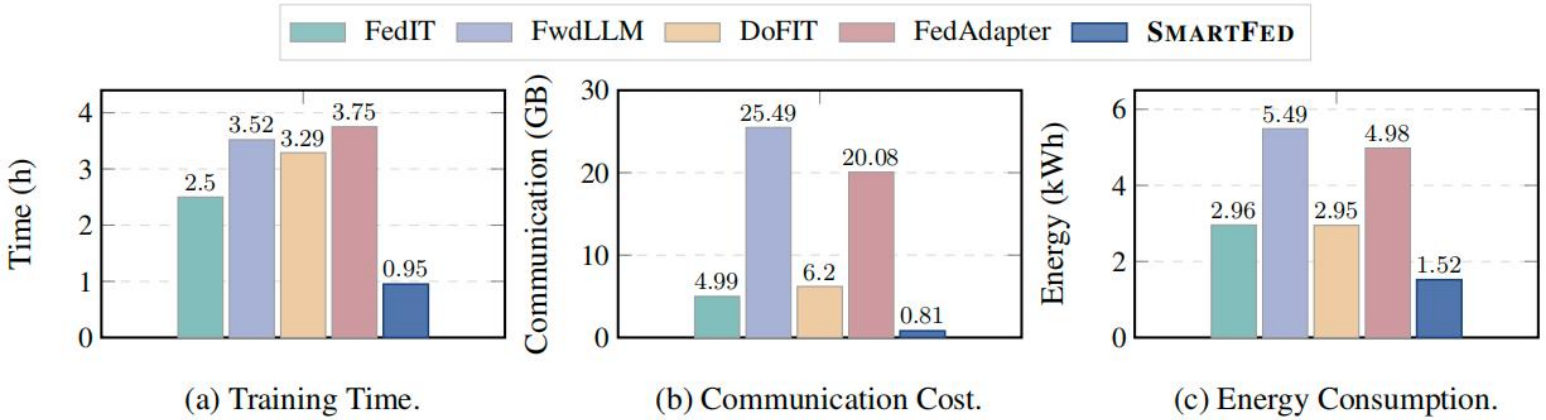


Figure 7. Overhead analysis of different federated fine-tuning methods on LLaMA2-7B for the Chinese+Math task.

Data Efficiency: Less Data, Better Adaptation

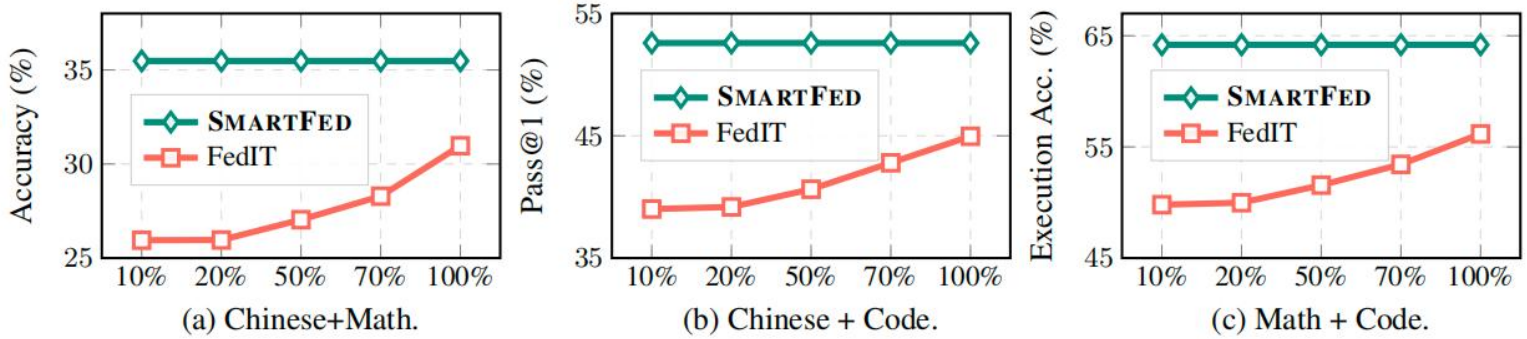


Figure 8. Performance comparison under varying fractions of training data on LLaMA2-7B across three downstream tasks.

More Analysis

Quota Allocation Analysis: Adaptive Expert Budgeting.

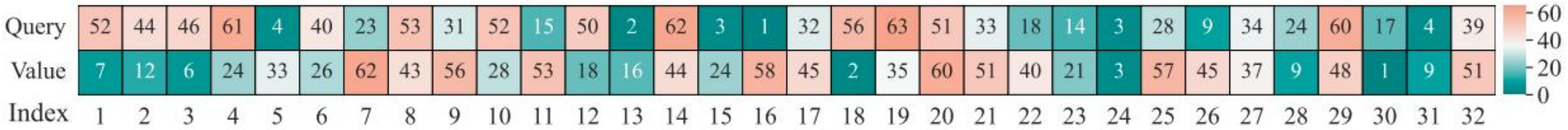


Figure 9. Distribution of expert quotas across different layers of LLaMA2-7B for the Chinese+Math task.

Fine-Grained Knowledge Fusion.

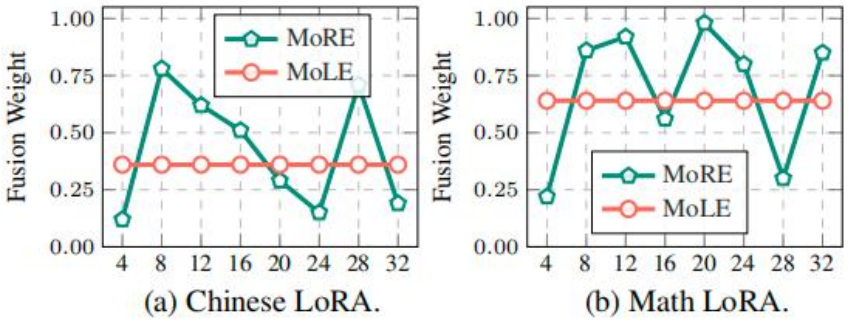


Figure 10. Average fusion weights for the `Query` matrix in the third layer of LLaMA2-7B on the Chinese+Math task.

Model Generalization & Ablation Analysis.

Table 2. Performance evaluation on recent state-of-the-art models.

Method	Evaluation Task			Average
	MGSM	DoIT	GSM-Hard	
LLaMA3-3B (Grattafiori et al., 2024)				
FedIT	35.42	41.48	48.20	41.70 (-11.86)
SmartFed	**46.89**	**53.57**	**60.21**	**53.56**
LLaMA3-8B (Grattafiori et al., 2024)				
FedIT	47.55	56.14	64.92	56.20 (-13.82)
SmartFed	**61.28**	**70.33**	**78.44**	**70.02**
Qwen3-8B (Yang et al., 2025)				
FedIT	50.25	58.54	66.49	58.43 (-13.82)
SmartFed	**64.21**	**72.99**	**79.56**	**72.25**

Table 3. Ablation study of SmartFed.

Configuration	Evaluation Task			Average
	MGSM	DoIT	GSM-Hard	
LLaMA2-7B (Touvron et al., 2023)				
SmartFed	**35.48**	**52.59**	**64.19**	**50.75**
w/o MoRE	32.52	48.81	58.22	46.52 (-4.23)
w/o EEQA	33.60	49.59	58.70	47.30 (-3.45)
LLaMA2-13B (Touvron et al., 2023)				
SmartFed	**53.18**	**63.01**	**72.65**	**62.95**
w/o MoRE	48.32	58.71	66.95	58.00 (-4.95)
w/o EEQA	50.25	60.20	68.46	59.64 (-3.31)

Conclusion

❑ We identify a new opportunity:

Existing federated LoRA fine-tuning still trains adapters from scratch, while public LoRA repositories already provide reusable task-specific knowledge.

❑ We propose SmartFed:

A resource-efficient federated fine-tuning framework that shifts from training adapters to composing rank-wise experts with a lightweight router.

❑ We demonstrate strong effectiveness:

SmartFed achieves superior performance across multiple LLMs and tasks, while significantly reducing training, communication, and energy costs.

Thank You!