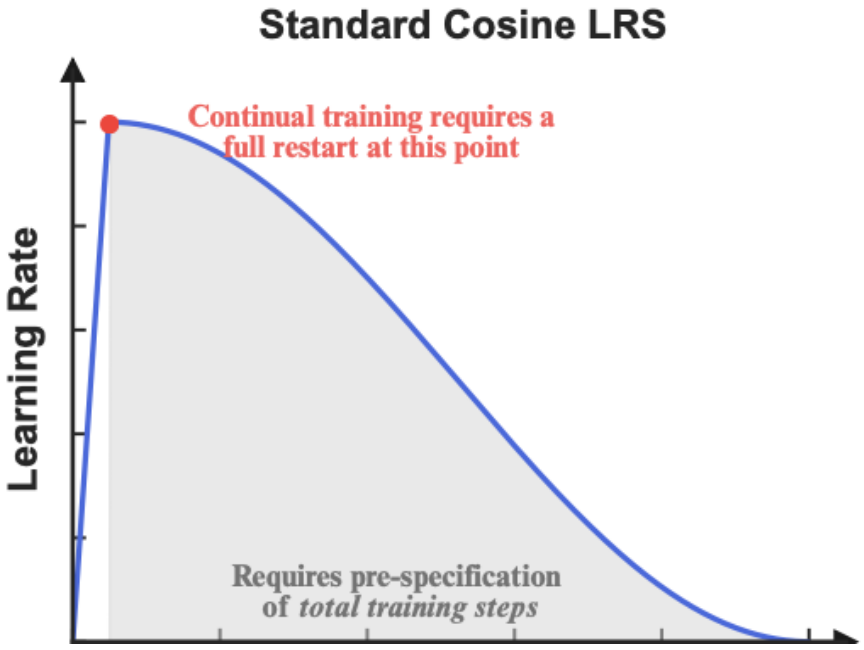
WSM: Decay-free Learning Rate Schedule via Checkpoint Merging for LLM Pre-training

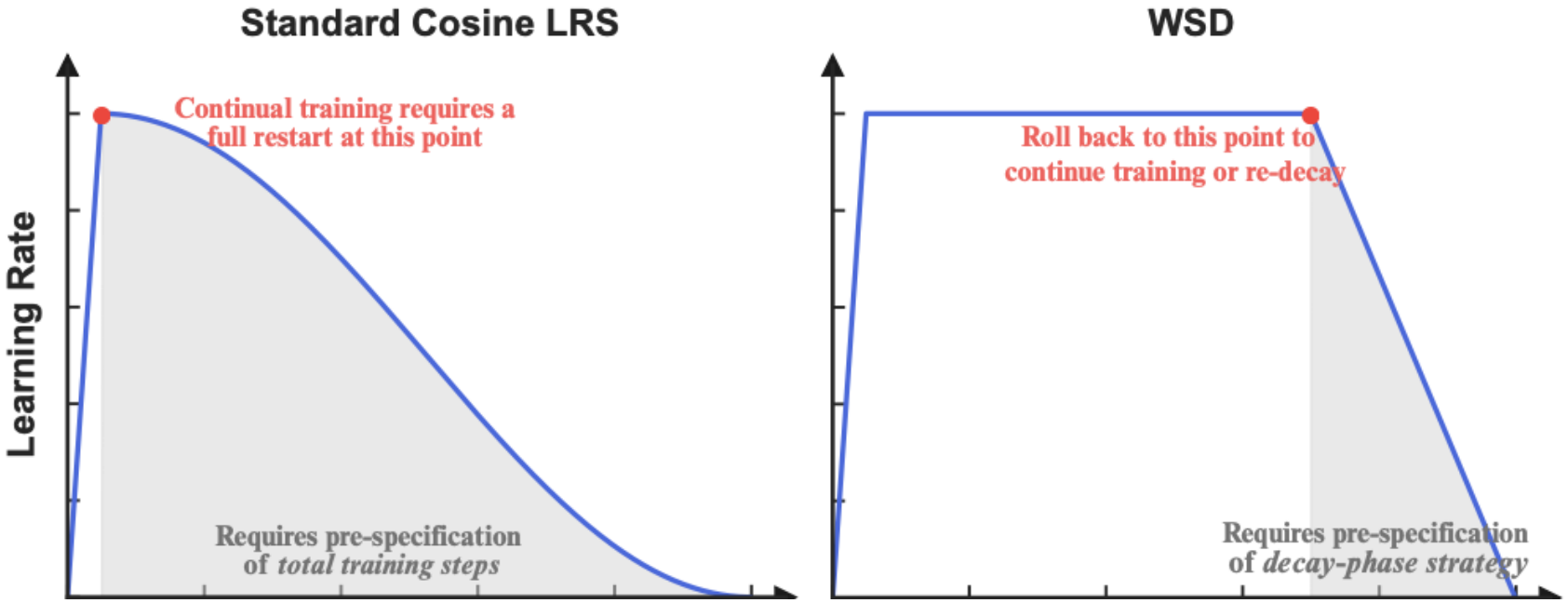
Changxin Tian[1]*, Jiapeng Wang[1,2]*, Qian Zhao[1], Kunlong Chen[1],
Jia Liu[1], Ziqi Liu[1], Jiaxin Mao[2],
Wayne Xin Zhao[2]†, Zhiqiang Zhang[1]† and Jun Zhou[1]

[1]Ant Group, [2]Renmin University of China
tianchangxin.tcx@antgroup.com, wangjp1010@ruc.edu.cn

Limitations of Cosine LRS



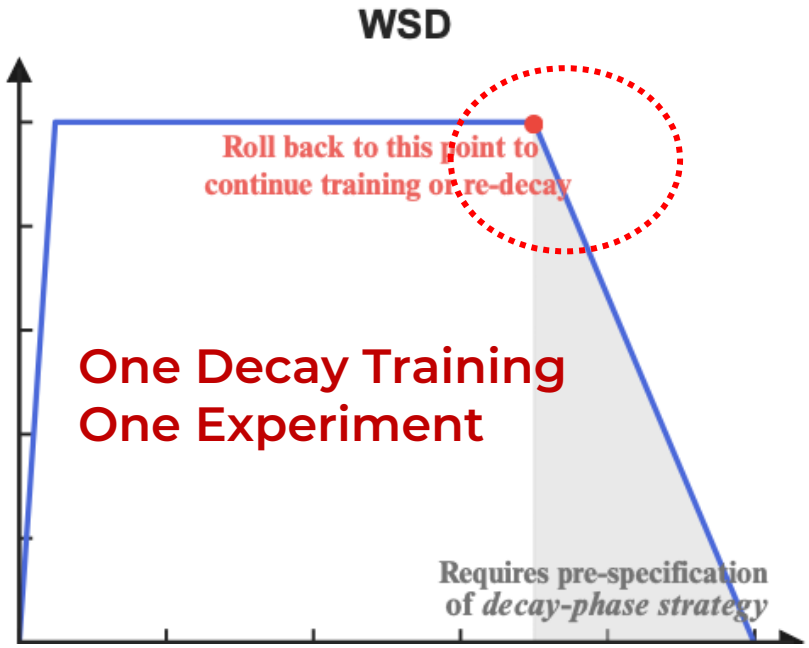
From Cosine Scheduler to WSD



Pretraining-friendly

Limitations of WSD

- When to start?
- How long to decay?
- Which Curve?
- What if modifications?

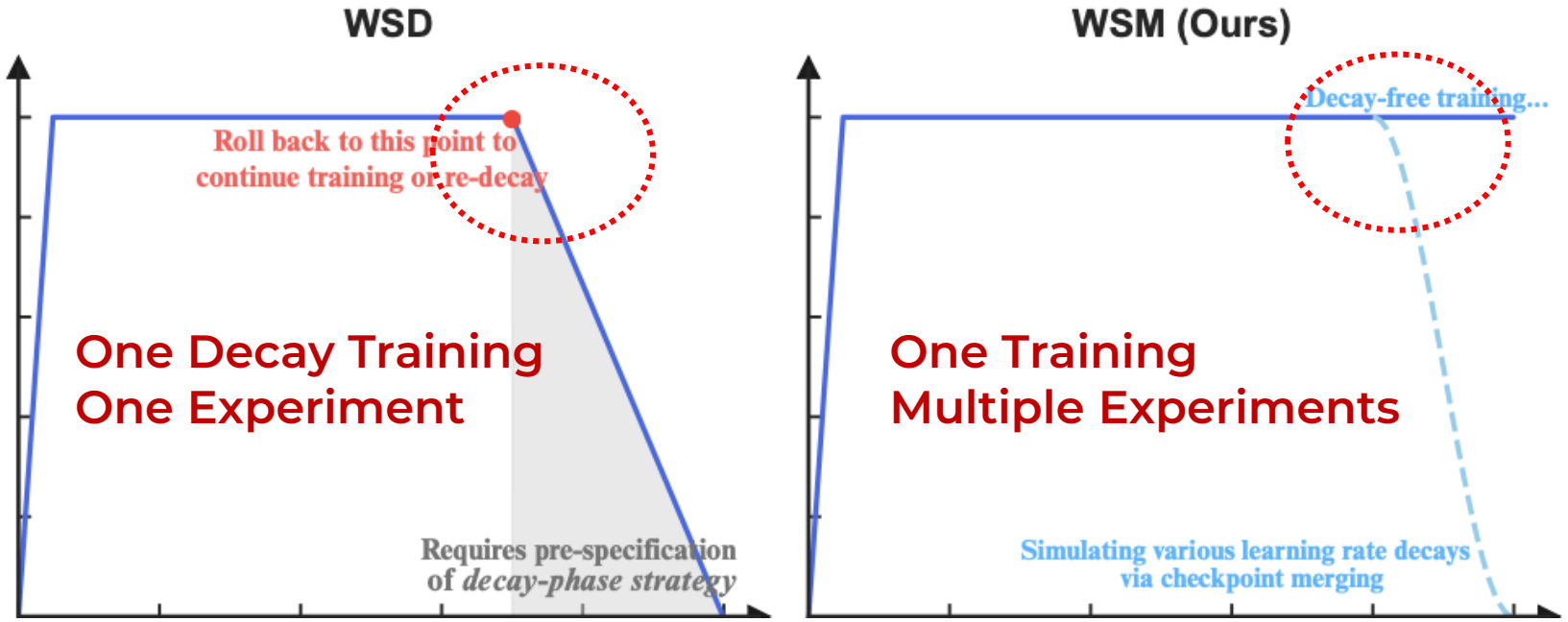


Requires pre-specifying phase/strategies. **Rolls back to the end of the stable phase.**

Limitations of WSD

- When to start?
- How long to decay?
- Which Curve?
- What if modifications?

WSD vs. WSM at a glance



Requires pre-specifying phase/strategies. **Rolls back to the end of the stable phase.**

No annealing required during training. **Simulates various decay methods** via checkpoint merging.

Merging as Synthetic Decay

Checkpoint Merging

$$\hat{\theta}_{n+k} = \sum_{j=0}^{k} c_j \theta_{n+j}$$

(c_j ~ weight of merged ckpts)

Gradient updating (with LR Decay)

$$\hat{\theta}_{n+k} = \theta_n - \sum_{i=1}^{k} w_i \cdot g_{n+i-1}$$

(w_j ~ weight of accumulated gradient)

Merging as Synthetic Decay

Checkpoint Merging

$$\hat{\theta}_{n+k} = \sum_{j=0}^{k} c_j \theta_{n+j}$$

Gradient updating (with LR Decay)

$$\hat{\theta}_{n+k} = \theta_n - \sum_{i=1}^{k} w_i \cdot g_{n+i-1}$$

Link between Checkpoint Merging and Gradient updating

$$\hat{\theta}_{n+k} = \sum_{j=0}^{k} c_j \left(\theta_n - \sum_{l=1}^{j} g_{n+l-1} \right) = \theta_n - \sum_{i=1}^{k} \left(\sum_{j=i}^{k} c_j \right) g_{n+i-1}$$

$$\hat{\theta}_{n+k} = \theta_n - \sum_{i=1}^{k} w_i \cdot g_{n+i-1}$$

Merging as Synthetic Decay

Checkpoint Merging

$$\hat{\theta}_{n+k} = \sum_{j=0}^{k} c_j \theta_{n+j}$$

Gradient updating (with LR Decay)

$$\hat{\theta}_{n+k} = \theta_n - \sum_{i=1}^{k} w_i \cdot g_{n+i-1}$$

Link between Checkpoint Merging and Gradient updating

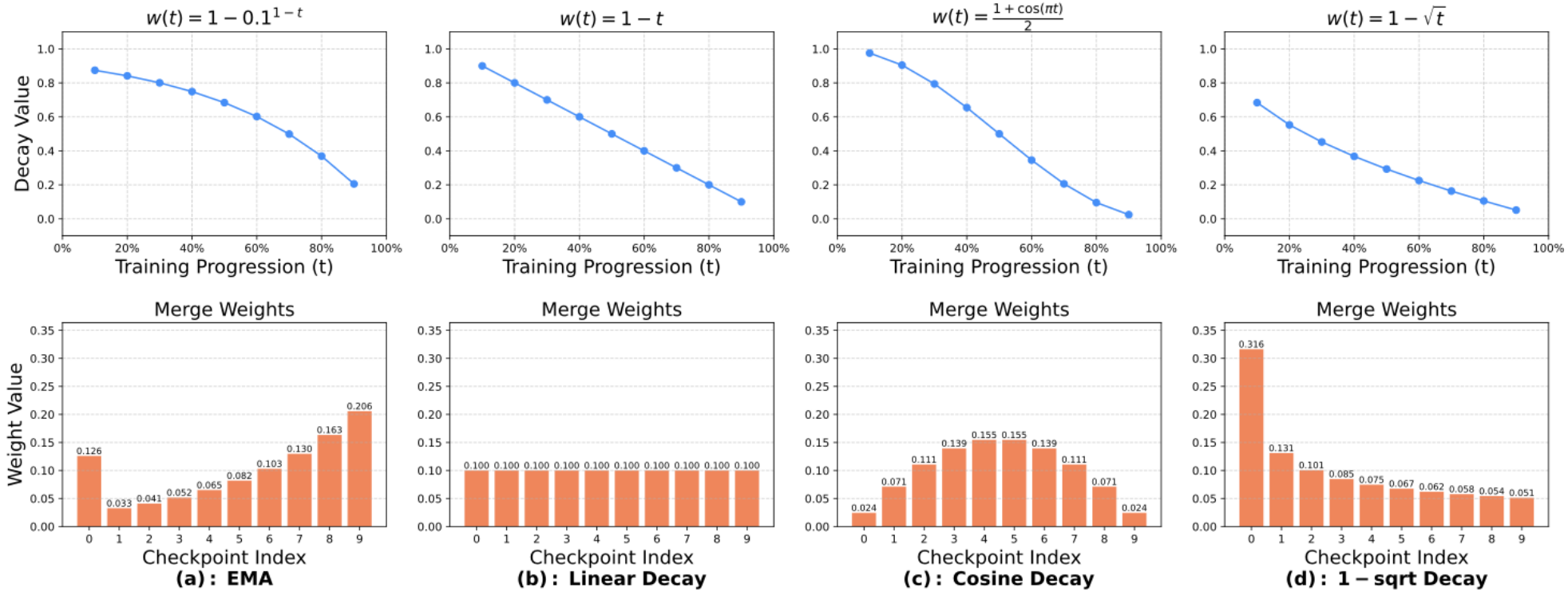
$$\hat{\theta}_{n+k} = \sum_{j=0}^{k} c_j \left(\theta_n - \sum_{l=1}^{j} g_{n+l-1} \right) = \theta_n - \sum_{i=1}^{k} \left(\sum_{j=i}^{k} c_j \right) g_{n+i-1}$$

$$\hat{\theta}_{n+k} = \theta_n - \sum_{i=1}^{k} w_i \cdot g_{n+i-1}$$

$$\begin{cases} c_k = w_k \\ c_j = w_j - w_{j+1}, \\ c_0 = 1 - \sum_{j=1}^{k} c_j = 1 - w_1 \end{cases}$$

Merging as Synthetic Decay

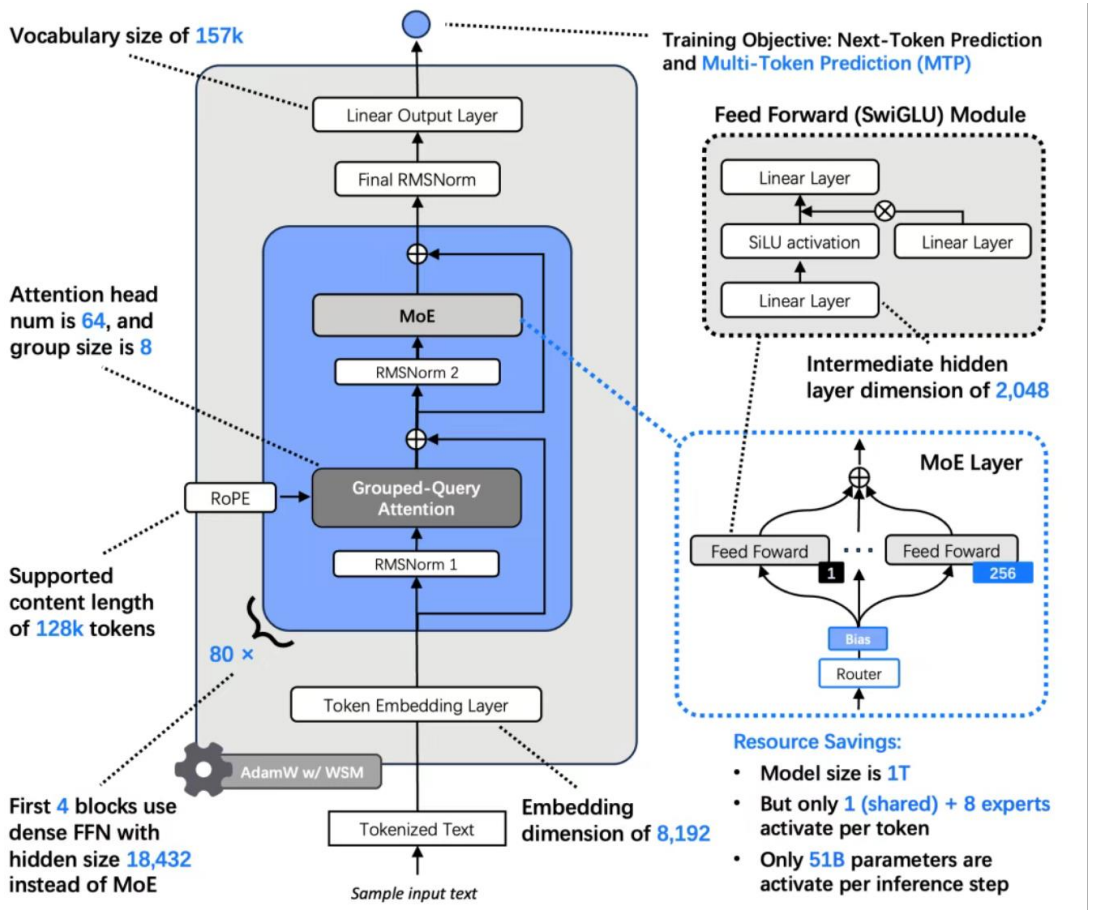
Any decay schedule can be translated into a principled model merging weight.



The empirical ranking is consistent across both worlds: **1-sqrt > Mean > EMA.**

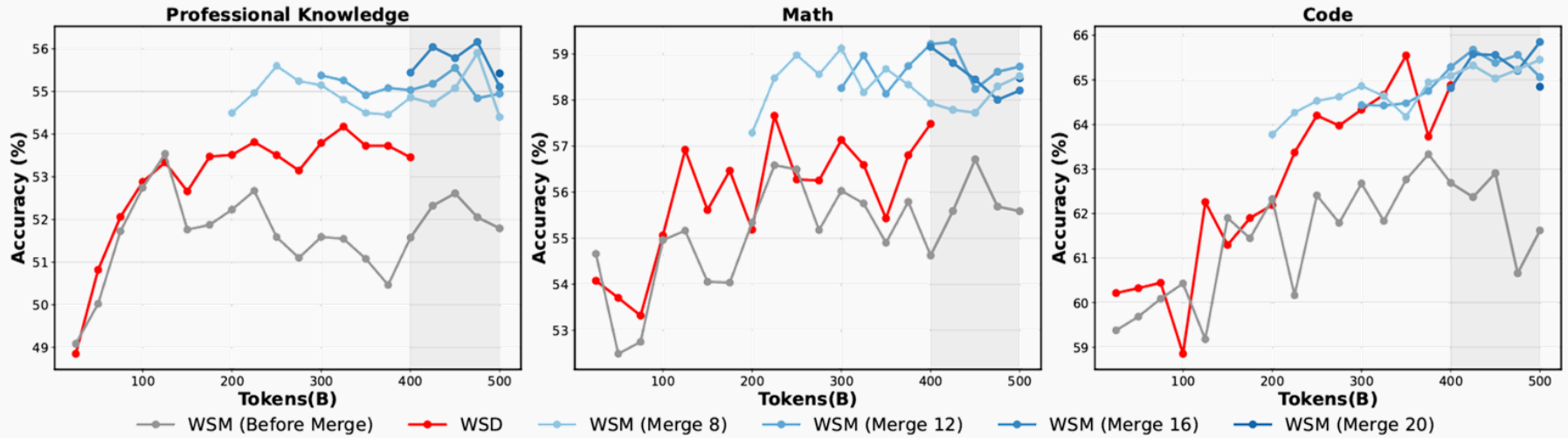
Experiments

- Experiment Setup
  - Ling-mini, a standard MoE model with 16.3B ~ A1.4B parameters
  - Begin with a checkpoint pretrained on 10.2T tokens



WSM vs. WSD: Main Results

- Red: LR decay in WSD
- Gray: Stable LR in WSM
- Blue: Post-merge Model in WSM

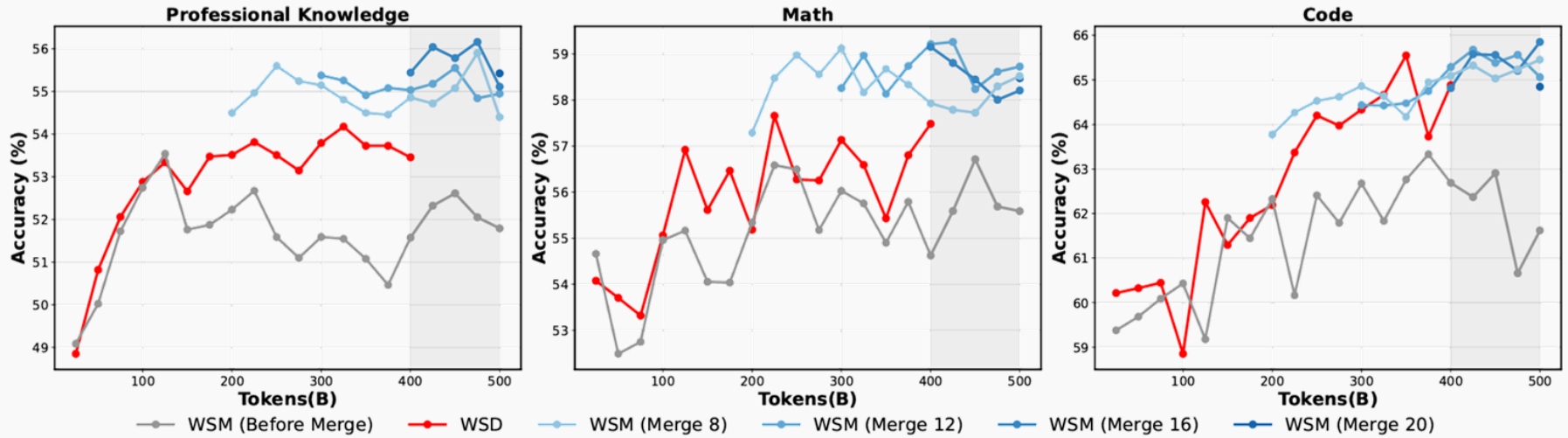


		General Knowledge	Language Modeling	Math	Code	Professional Knowledge	Overall Average
Base Model	WSD	69.06	67.78	57.49	64.88	53.46	62.67
	WSM	**70.22**	**68.67**	**58.81**	**65.58**	**56.04**	**63.95**
	Improv.	+1.68%	+1.31%	+2.30%	+1.08%	+4.83%	+2.04%

WSM outperforms WSD by over 2%.

WSM vs. WSD: Main Results

- Red: LR decay in WSD
- Gray: Stable LR in WSM
- Blue: Post-merge Model in WSM

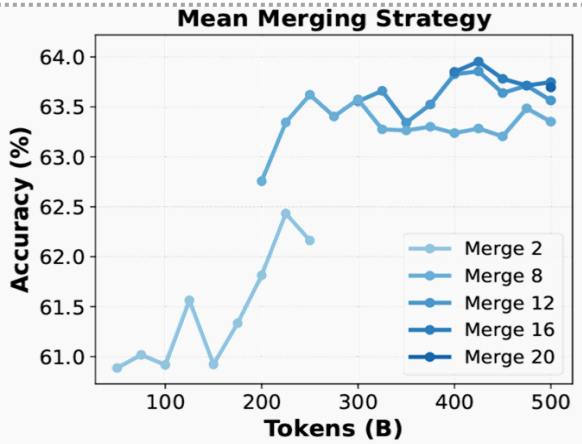


		Language	Knowledge	Math	Code	Reason	Agent	Overall Average
Inst Model	WSD	81.12	60.00	61.43	**58.23**	63.21	68.16	62.90
	WSM	**84.78**	**61.73**	**62.28**	57.95	**64.94**	**69.33**	**64.07**
	Improv.	+4.51%	+2.88%	+1.38%	-0.48%	+2.74%	+1.72%	+1.86%

This advantage holds up after SFT.

Key Drivers of WSM

Merging Duration



- ✓ Larger merging duration
- ✓ Good merging algorithm
- ✓ Finer merging granularity
- ✓ Introduction of high-quality annealing data

Merging Algorithm

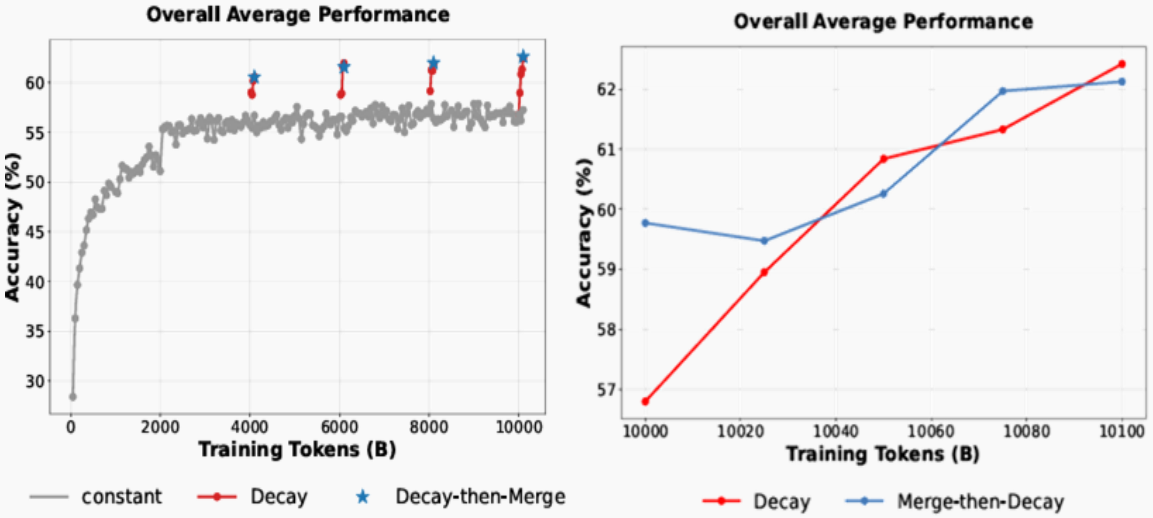
		General Knowledge	Language Modeling	Math	Code	Professional Knowledge	Overall Average
Decay	1-*sqrt*	69.06	67.78	57.49	64.88	53.46	62.67
Merging	EMA	69.05	67.64	58.81	64.19	54.44	63.01
	Mean	70.22	**68.67**	58.81	65.58	**56.04**	63.95
	1-*sqrt*	**70.27**	68.26	**59.65**	**65.70**	55.42	**64.06**

Merging Granularity

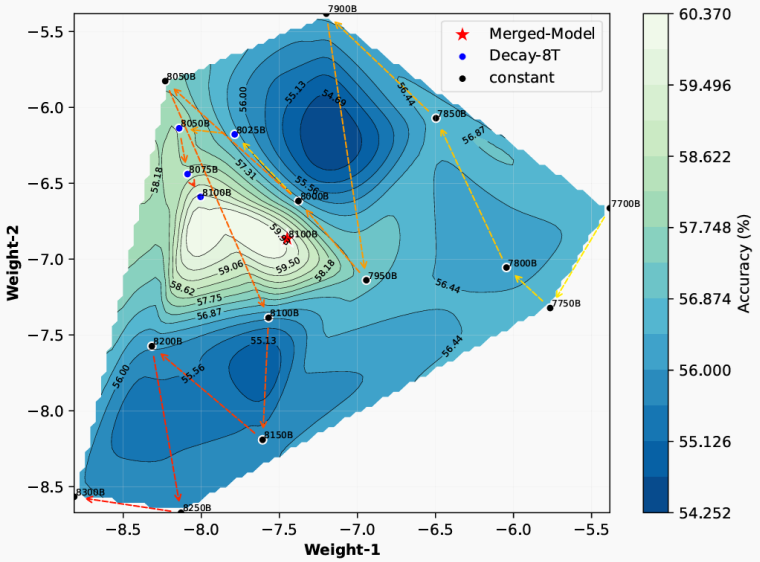
Merging Granularity	General Knowledge	Language Modeling	Math	Code	Professional Knowledge	Overall Average
(5B,16)	**69.46**	79.95	57.07	**64.68**	53.82	63.63
(10B,8)	69.23	80.00	**57.94**	64.39	53.78	**63.78**
(20B,4)	69.29	**80.29**	56.79	63.87	**54.19**	63.36
(40B,2)	68.47	79.83	56.57	63.59	52.20	62.77
(80B,1)	67.47	64.98	55.07	61.69	51.61	60.33

Additional Observations

✓ Decay & Merging: Not additive



✓ Improved Generalization



✓ Better Load Balancing

	language modeling loss	mean_global_max_violation	mean_global_min_violation
WSD	0.675	0.601	0.322
WSM	0.697	**0.545**	**0.201**

Conclusion

- We present Warmup-Stable and Merge (WSM), a simple yet general framework for LLM pretraining LR scheduling.
- Through extensive experiments, we systematically investigate key factors in instantiating this framework, including merge methods, merge frequency, and the compatibility of merging and decay strategies.
- WSM achieves substantial improvements over WSD, and offers significant flexibility in training.

Thanks for your attention