COLM 2026 · Conference on Language Modeling

To Mix or To Merge?
Toward Multi-Domain Reinforcement Learning for Large Language Models

Haoqing Wang[1†] **Xiang Long**[1†] **Ziheng Li**[2,1†] **Yilong Xu**[1] **Tingguang Li**[1] **Yehui Tang**[1 ✉]

[1] Samsung Research, Beijing [2] Peking University

A systematic comparison of two multi-domain RLVR paradigms — efficiency, policy neighborhoods, and self-verification.

MOTIVATION

The Question: Mix or Merge?

RLVR gives **expert-level** ability in single domains (math, coding). To build a **general multi-domain expert**, how should domains collaborate during RL? SOTA models use two paradigms — but rarely compare them.

① MIX — Multi-Task RLVR

- One RL run over **mixed** heterogeneous rewards from all domains **simultaneously**.
- Domain-routed reward per batch; 1000 steps.
- *Used by DeepSeek-R1, Qwen3.*

② MERGE — Separate RL + Fuse

- Train **one expert per domain**, then combine.
- Weight merging (Avg·TA·Ties·SCE) or distillation.
- *Used by GLM-4.5, MiMo-V2-Flash.*

Our study: 5 domains (math · coding · science · instruction-following · agent) × 9 benchmarks, on Qwen3-4B-Base with GRPO. We compare both paradigms and dissect the internal mechanisms.

EXPERIMENTAL SETUP

Setup at a Glance

5
Domains

9
Benchmarks (Avg@K)

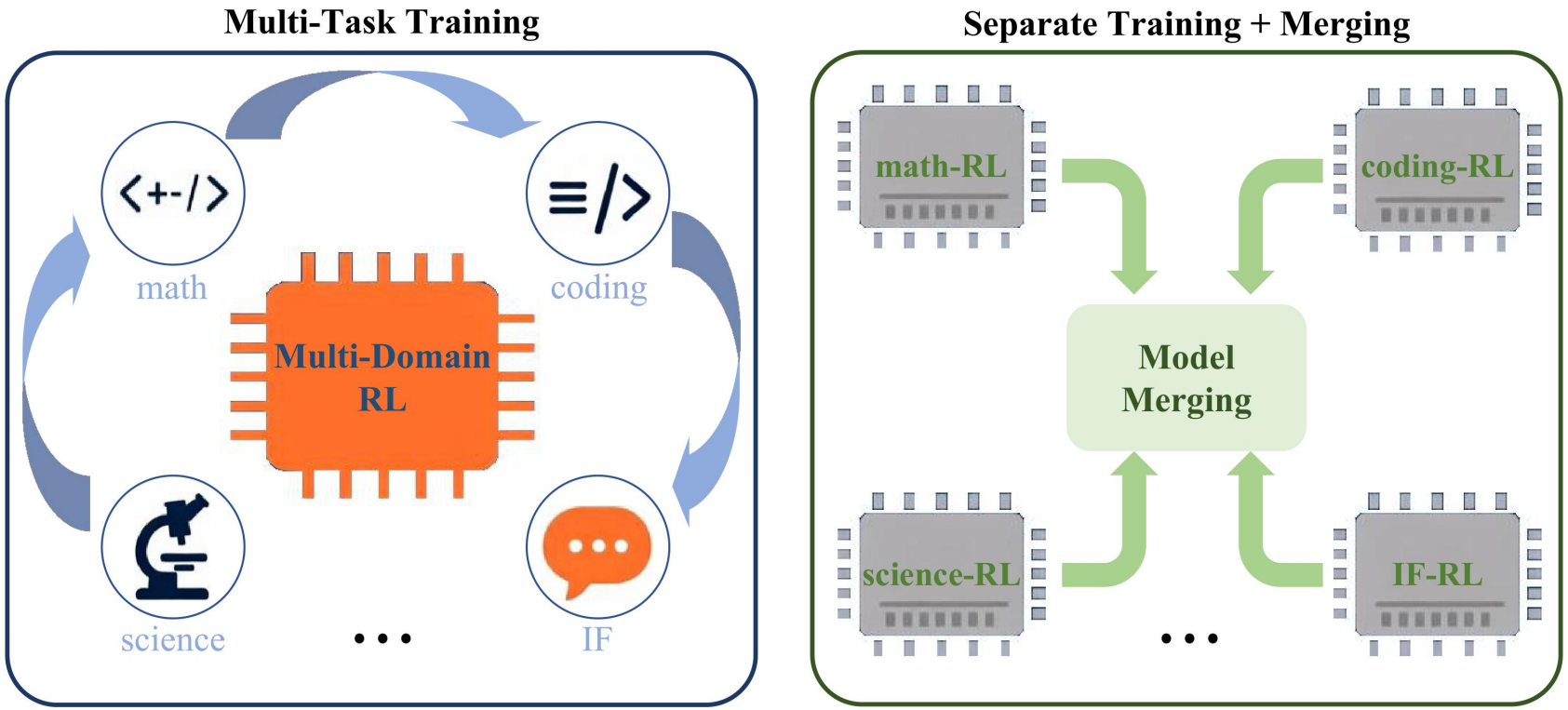
4B
Qwen3-4B-Base · GRPO

6
Merge methods

Pipeline & Data

- Simplified **SFT** → **RL** pipeline for controllable comparison; open-source **Nemotron-3 Nano** blend (14M SFT samples).
- Per-domain RLVR: **400 steps** · Multi-task RLVR: **1000 steps** · GRPO, group size 16, max length 32K.
- Merging: **weight merging** (Average · Task-Arithmetic · Ties · SCE, ± DARE) and **multi-teacher distillation** (on/off-policy).
- 9 benchmarks: **AIME'24/'25, LCB v5/v6, HLE, GPQA-D, IFEval, IFBench, BFCL v3**.

The Two Paradigms



Left — Multi-Task Training: all domains learned jointly in a single RL run.

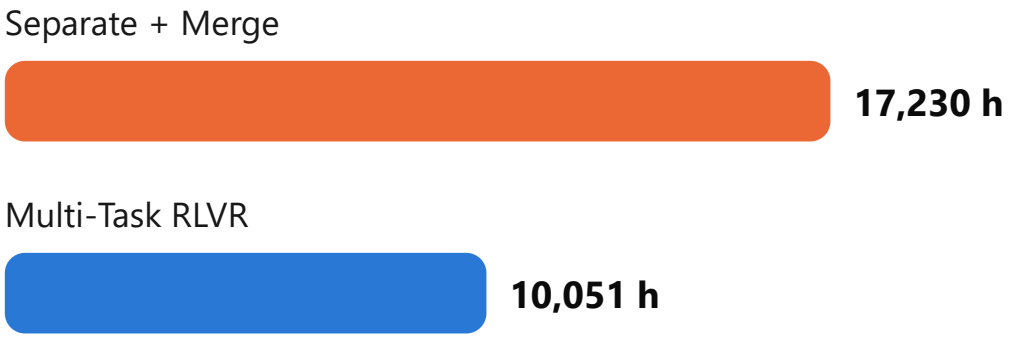
Right — Separate Training + Merging: independent per-domain experts fused into one model.

FINDING 1

Mix ≈ Merge, at 58% the Cost

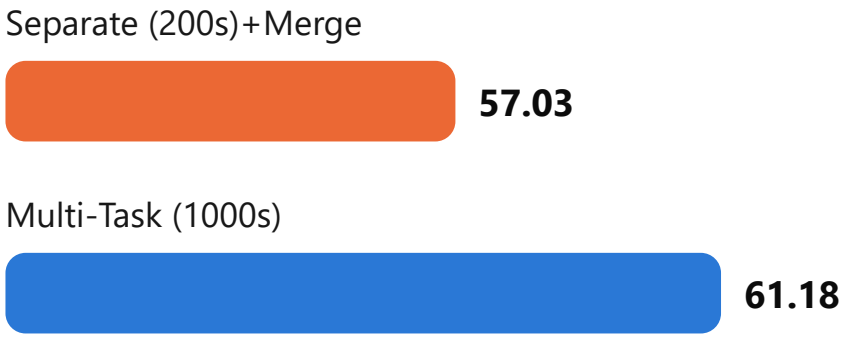
Multi-task RLVR **matches** 'separate RLVR + merging' using only **58.3% of GPU hours**. Under an equal budget it **wins by +4.15 pts** avg over 9 benchmarks.

Total GPU Hours



→ 58.3% of the compute

Equal-Budget Avg. Score



+4.15 points

- Cross-domain interference is **small**; reasoning domains (math · coding · science) are **mutually synergistic**.
- Agent gains nothing from others *in isolation*, but benefits once its own tool-use trajectories are present.

FINDING 1

Main Results — Avg@K over 9 Benchmarks

Benchmark	Base	SFT	Best Single	Merge	Multi-Task
Math					
AIME'24	9.7	56.0	77.7	**80.6**	**81.2**
AIME'25	5.7	51.3	70.4	**74.5**	**73.4**
Coding					
LCB v5	16.5	55.9	**65.0**	**63.3**	63.2
LCB v6	18.3	52.0	**58.9**	**58.9**	56.6
Science					
HLE	4.5	5.8	7.0	**7.3**	**7.2**
GPQA-D	20.1	42.7	**58.5**	**55.2**	53.6
Instruction Following					
IFEval	35.1	79.5	90.9	**93.9**	**93.5**
IFBench	11.9	38.4	**59.5**	59.2	**61.2**
Agent					
BFCL v3	29.7	50.1	59.1	**61.4**	**60.7**

Read-off

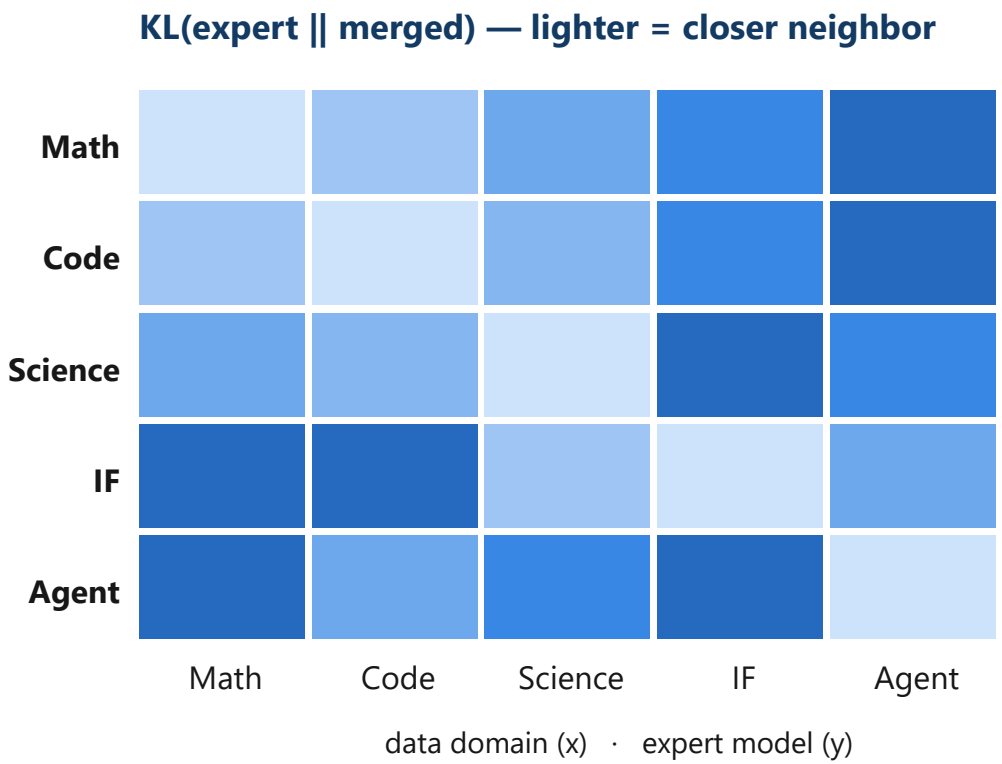
- best
- 2nd best
- Merge
- Multi-task

- Both multi-domain models often **surpass every single expert**.
- Gains show on AIME, HLE, IFEval, BFCL — cross-domain synergy is **real**.
- Matches official Qwen3-4B using open data only.

FINDING 2

Why Merging Works: Policy Neighborhoods

Using **KL divergence** between each expert and the merged policy, we find **policy neighborhoods**: a nearby expert's distribution transfers toward the optimum during weight merging — boosting the target domain.



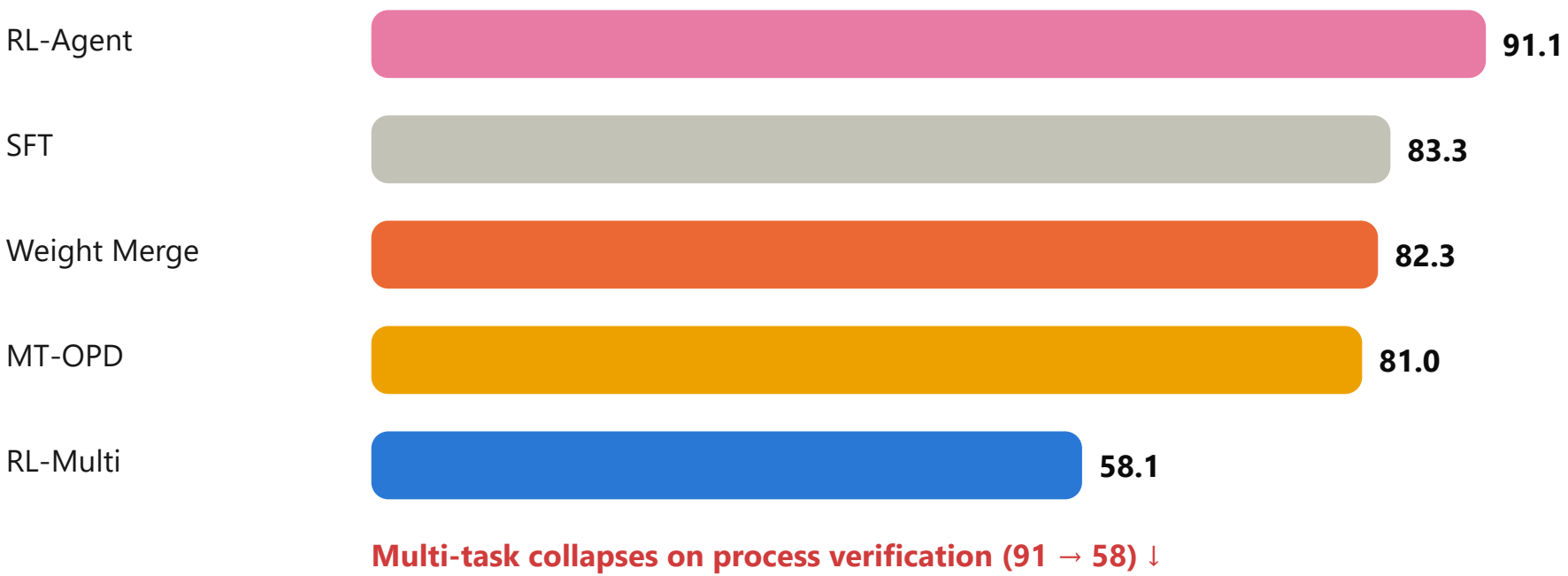
What it means

- Off-diagonal **light** cells = policy neighbors of the merged model.
- **Coding** is a neighbor of **Math**: merging them adds **+2.6 / +2.0** on AIME'24/'25.
- Neighborhood is **asymmetric** (KL is asymmetric): Math↔Coding gain does not reverse.
- Merging non-neighbors yields **no** extra gain.

FINDING 3

Self-Verification: A Robustness Trade-off

RLVR induces emergent **self-verification**. Outcome-judging **rises** with skill (r=+0.63); process-judging can **degrade** (r=−0.53). Multi-task RL is efficient but its process verification **collapses**.



Key signals

- **Agent** expert = best process verifier.
- **Merging** stays robust & balanced.
- **Task dictates modality:**
- logic→process, constraint→outcome.

TAKEAWAYS

Practical Guidance — When to Mix vs. Merge

Choose Multi-Task RL

- Compute is limited.
- You want emergent cross-domain generalist skills.
- Comparable quality at ~58% cost.

Choose Separate + Merge

- Reusing existing expert models.
- Maximizing single-domain peaks.
- Self-verification reliability matters.

Design Rewards by Error Locus

- Logic tasks (math/code) → process verification.
- Constraint tasks (IF) → outcome verification.
- Agent training cultivates strong process judges.

Bonus: mutual gain between reasoning domains means you can add coding/STEM/IF data to build a stronger single-domain expert.

CONCLUSION

Mix when efficient, Merge when robust.

- Multi-task RLVR shows **small inter-task interference** and **synergy** among reasoning domains — comparable quality at 58.3% cost.
- Weight merging works via **neighborhood policy transfer**; it inherits single-task skills predictably.
- Multi-task training develops **emergent** strategies beyond skill aggregation.
- Critical **robustness–efficiency trade-off**: multi-task is cheap but degrades process-level self-verification; merging preserves it.