MMCR: Sequential Layer-wise Model Merging via Reinforcement Learning

Chou Fang, Chang [1] Zi-Yong, Lai [1] Chih-Pao, Lin [1]

[1]National Yang Ming Chiao Tung University RL **Group 8**

Project Page

Abstract

Model merging aims to combine multiple task-specific models into a single unified model without full retraining. However, existing methods such as Task Arithmetic, TIES, and AdaMerging usually rely on fixed rules or one-shot coefficient optimization. These methods may not fully capture layer-wise conflicts or sequential dependencies among merging decisions. We propose MMCR, a reinforcement-learning-based framework that learns adaptive layer-wise task-vector coefficients. Experiments on eight vision classification benchmarks show that MMCR achieves strong average accuracy and benefits from trajectory-level parallelization.

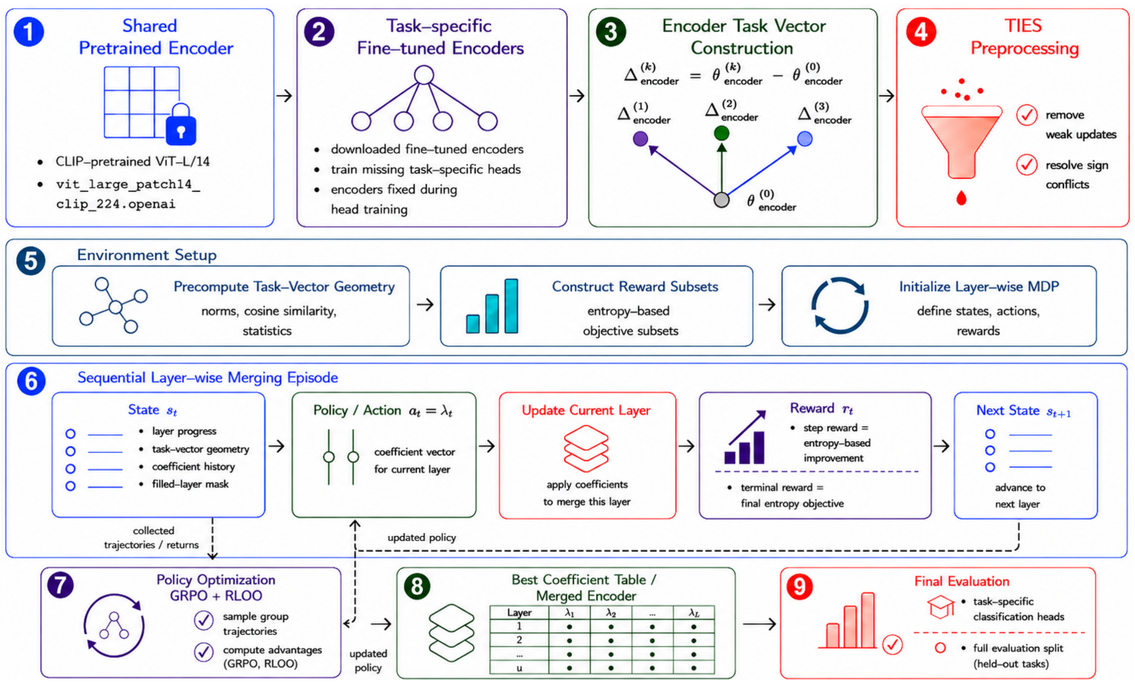
Research Objectives

The present study investigates the following objectives:

- **Objective 1:** Formulate task-vector model merging as a sequential layer-wise decision-making problem.
- **Objective 2:** Learn adaptive coefficients for each layer using an RL policy instead of fixed or one-shot merging coefficients.
- **Objective 3:** Evaluate whether sequential coefficient learning improves merged-model accuracy and runtime scalability.

Method Overview

MMCR learns how to merge downloaded task-specific encoders layer by layer. At each layer, an RL policy selects coefficients for combining TIES-preprocessed encoder task vectors.



MMCR Formulation and Optimization

1. **Encoder task vector.** MMCR merges only encoder parameters. For task k:

$$\Delta_{\text{enc}}^{(k)} = \theta_{\text{enc}}^{(k)} - \theta_{\text{enc}}^{(0)},$$

where $\theta_{\text{enc}}^{(0)}$ is the pretrained encoder and $\theta_{\text{enc}}^{(k)}$ is the downloaded task-specific encoder.

2. **Layer-wise MDP.** At each step, the agent decides how to merge one encoder layer.

State s_t	Layer progress, task-vector geometry, and coefficient history.
Action $a_t = \lambda_t$	Layer-wise coefficient vector for the current encoder layer.
Reward r_t	Improvement of an entropy-based objective, plus terminal reward.

Given action $\lambda_{l,k}$, layer l is merged by:

$$\theta_{\text{enc},l}^{(m)} = \theta_{\text{enc},l}^{(0)} + \sum_{k=1}^{K} \lambda_{l,k} \tilde{\Delta}_{\text{enc},l}^{(k)}.$$

3. **Entropy-based reward.** The merged encoder is evaluated only at selected reward steps. Let e_t indicate whether step t is evaluated. The multi-task objective is:

$$J_t = \frac{1}{K} \sum_{k=1}^{K} q_{t,k} - \alpha \cdot \text{Std}(q_{t,1}, \ldots, q_{t,K}),$$

where $q_{t,k}$ is the negative entropy score for task k. The reward is:

$$r_t = e_t \eta_{\text{step}} (J_t - J_{\text{prev}}) + \mathbb{I}[t = L] \eta_{\text{term}} J_t.$$

If $e_t = 0$, no evaluation is performed, and the step reward is zero.

4. **GRPO + RLOO update.** For trajectory τ_i, the return is:

$$R_i = \sum_{t=1}^{L} r_{i,t}.$$

GRPO compares each trajectory with the others in the same group and normalizes the relative return:

$$A_i = \frac{R_i - \frac{1}{G-1} \sum_{j \neq i} R_j}{\text{Std}(R_1, \ldots, R_G) + \epsilon}.$$

This provides a group-relative learning signal without training a separate critic.

References

[1] Enneng Yang et al. AdaMerging: Adaptive model merging for multi-task learning. arXiv preprint arXiv:2310.02575, 2023.

[2] Gabriel Ilharco et al. Editing models with task arithmetic. arXiv preprint arXiv:2212.04089, 2022.

[3] Prateek Yadav et al. TIES-Merging: Resolving interference when merging models. NeurIPS, 2023.

[4] Chongjie Si et al. NAN: A training-free solution to coefficient estimation in model merging. arXiv preprint arXiv:2505.16148, 2025.

[5] Alexey Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

Experimental Setup

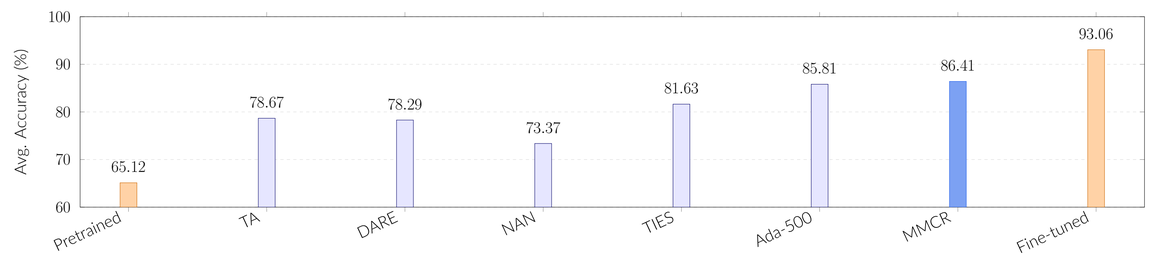
- **Backbone:** `vit_large_patch14_clip_224.openai`, a CLIP-pretrained ViT-L/14 image encoder.
- **Encoders:** Downloaded task-specific fine-tuned encoders for eight downstream tasks.
- **Heads:** Missing task-specific classification heads are trained by us while keeping encoders frozen.
- **Merged component:** MMCR merges **only encoder parameters**.
- **Tasks:** SUN397, Stanford Cars, RESISC45, EuroSAT, SVHN, GTSRB, MNIST, and DTD.
- **Evaluation:** The merged encoder is evaluated with the corresponding task-specific head.

Main Results

Result Highlights

86.41%	+4.78 pp	1.57×
Avg. Acc.	over TIES	GPU speedup

Average Accuracy Comparison



TA = Task Arithmetic; **DARE** = DARE-TA; **NAN** = NAN-TA; **Ada-500** = AdaMerging-500.

Method	Avg. Acc.	Role
TIES	81.63	Data-free conflict reduction
AdaMerging-300 Epochs	77.86	Budget-matched baseline
AdaMerging-500 Epochs	85.81	Long-run baseline
AdaMerging-600 Epochs	86.36	Stronger long-run baseline
MMCR (ours)	**86.41**	**Best merged model**

Key finding: MMCR improves over TIES by **+4.78 percentage points** and slightly surpasses AdaMerging-600 Epochs, showing that sequential layer-wise coefficient learning can improve the multi-task balance.

Runtime Comparison

Method	GPUs	Time	Avg. Acc.
TIES	CPU	3.23 min	81.63
AdaMerging-300 Epochs	1	185.00 min	77.86
AdaMerging-500 Epochs	1	356.87 min	85.81
AdaMerging-600 Epochs	1	418.90 min	86.36
MMCR	1	245.17 min	86.41
MMCR	**2**	**155.97 min**	**86.41**

Practical advantage: AdaMerging improves with longer optimization, but runtime increases. MMCR achieves higher accuracy than AdaMerging-600 with shorter single-GPU runtime, and further reduces wall-clock time through two-GPU trajectory parallelization.

Policy Transferability Findings

Can the learned policy transfer to new task groups?

Related 2-task	Related 3-task	Unrelated 2-task
TIES: 95.20 Transferred: **97.33** Target: 97.54	TIES: 93.20 Transferred: 91.05 Target: **96.26**	TIES: 96.97 Transferred: 94.02 Target: **97.06**
Transfers well	Harder with more tasks	Domain gap hurts

Takeaway. MMCR learns reusable layer-wise merging patterns for related task pairs, but transfer becomes less reliable as task complexity or domain mismatch increases.

Ablation and Advantage Summary

- **TIES preprocessing:** Cleaning weak and sign-conflicting updates improves Avg. Acc. from 83.69% to **86.41%**.
- **RL optimization algorithm:** GRPO + RLOO achieves the best average accuracy among the tested policy optimization variants.
- **Reward evaluation frequency:** Denser intermediate rewards can slightly improve accuracy, but they increase reward-evaluation cost.
- **Policy transferability:** The learned policy transfers well for related two-task groups, but transfer becomes harder with more tasks or larger domain gaps.

Conclusions

- MMCR formulates encoder merging as a **sequential layer-wise decision problem**.
- The RL policy learns adaptive coefficients for combining TIES-preprocessed task vectors.
- MMCR achieves **86.41%** average accuracy across eight vision benchmarks.
- MMCR benefits from trajectory-level parallelization, reducing wall-clock time by about **1.57×**.
- Policy transfer is promising for related task pairs, but remains challenging for larger or less related task groups.