MergeVLA: Cross-Skill Model Merging Toward a Generalist Vision-Language-Action Agent

Supplementary Material

1. Experimental Details

Our vision-language backbone is Qwen2.5-0.5B [7]. By default, we set $l = L$, $k_r = 8$, mask ratio $\lambda = 0.6$, merging scaling factor $\alpha = 1$. All finetuning is conducted on a single NVIDIA A6000 Ada GPU (48 GB). We summarize all fine-tuning hyperparameters used across LIBERO, RoboTwin, and real-world SO-101 experiments in Table 1. For the LIBERO benchmark, the Long task suite is trained for 50k steps, while all other LIBERO task suites and all RoboTwin tasks are trained for 30k steps. All experiments use the same VLM backbone and training configuration unless otherwise specified.

Table 1. **Fine-tuning hyperparameters used in all experiments.**

Hyperparameter	Value
Backbone	Qwen2.5-0.5B
Batch size	8
Learning rate	5×10^{-4}
LoRA rank	32
Use proprioception	True
Num images	2 (3 for RoboTwin)
Gradient step	30k (50k for LIBERO-Long)

2. Algorithm Details

In this section, we give a detailed algorithm description in Algorithm. 1 of how MergeVLA performs inference using our test-time task router when the task identity is unknown.

Algorithm 1 Test-time Task Routing and Inference in MergeVLA

Inputs: Task masks $\{\mathbf{S}_m\}_{m=1}^{M}$; Expert heads $\{\mathbf{H}_m^{l\to L}\}_{m=1}^{M}$; Pretrained VLM weights Θ_0; Merged task vector τ_{merge}; Value projections of the action expert at block l: $\mathbf{V}_{\text{T},m}^{l}$, $\mathbf{V}_{\text{A},m}^{l}$; Initial observation $(\mathbf{I}_0^v, \mathbf{I}_0^w, L)$
Routing phase (at $t = 0$):
for $m = 1$ to M **do**
  $\Theta_{\text{VLM}}^{(m)} = \Theta_0 + \mathbf{S}_m \odot \tau_{\text{merge}}$ ▷ Construct masked VLM
  $\mathbf{h}_{\text{T},m}^{l}, \mathbf{h}_{\text{A},m}^{l} = \Theta_{\text{VLM}}^{(m)}(\mathbf{I}_0^v, \mathbf{I}_0^w, L)$ ▷ Extract l-th block hidden states
  $\mathbf{V}_{\text{T},m}^{l} = \mathbf{L}_{\text{T},m}^{l}\,\mathbf{\Sigma}_{\text{T},m}^{l}\,(\mathbf{R}_{\text{T},m}^{l})^{\top}$
  $\mathbf{V}_{\text{A},m}^{l} = \mathbf{L}_{\text{A},m}^{l}\,\mathbf{\Sigma}_{\text{A},m}^{l}\,(\mathbf{R}_{\text{A},m}^{l})^{\top}$
  $\mathbf{r}_{\text{T},m} = \left\|\mathbf{P}_{\text{T},m}^{l}\mathbf{h}_{\text{A},m}^{l}\right\|_2$ ▷ Choose top-r_k singular vectors from $\mathbf{R}$ to get $\mathbf{P}$
  $\mathbf{r}_{\text{A},m} = \left\|\mathbf{P}_{\text{A},m}^{l}\mathbf{h}_{\text{T},m}^{l}\right\|_2$
  $\mathbf{r}_m = \frac{1}{2}\left(\mathbf{r}_{\text{T},m} + \mathbf{r}_{\text{A},m}\right)$
end for
$m^* = \arg\max_m \text{softmax}(\mathbf{r})$. ▷ Normalize scores with softmax and select task index
return m^*
Inference phase: Use $\mathbf{S}_{m^*}$, and expert head $\mathbf{H}_{m^*}^{l\to L}$ for all $t \geq 0$.

3. Preliminary Investigation on OpenVLA

In the early stage of this work, we explored the feasibility of directly applying existing model merging methods to OpenVLA [4], a popular VLA model. OpenVLA consists of three main components: a vision backbone, a projector, and a language model. The language model itself contains 32 transformer blocks followed by a single-layer MLP head (lm_head). We first attempted to merge all components of OpenVLA using Weighted Average and Task Arithmetic methods. However, the merged checkpoint completely failed on all tasks. This was surprising, as OpenVLA is essentially a VLM, and previous studies [1, 2, 6] have shown that VLMs can usually be merged successfully. This prompted us to investigate which part of OpenVLA prevents successful merging.

3.1. Non-mergeable Components in OpenVLA

To locate the source of failure, we decomposed the model into four submodules: A. the vision backbone; B. the projector; C. the language model body (excluding lm_head); and D. the lm_head. We then merged each submodule separately across the four official LIBERO task checkpoints using the existing merging method Iso-CTS [5]. Each merged checkpoint was evaluated on 50 trials per subtask. The results are summarized in Table 2, where the gray-highlighted row denotes the single-task fine-tuning performance. From the table, we observe that merging modules A, B, or D only slightly decreases success rates, whereas merging the language model body (C) causes complete failure on all tasks. This clearly indicates that the language model is the primary source of merging failure.

We hypothesize that this phenomenon arises because VLA tasks impose much stricter precision requirements on the model outputs than typical LLM or VLM tasks. In LLMs or VLMs, outputs are often discrete token sequences

or probability distributions, where small deviations are tolerable. In contrast, robotic control requires continuous numeric outputs, where even minor errors can cause irreversible physical or simulated state changes. Once the environment diverges from the model's training distribution, subsequent actions fail catastrophically. As component C is directly responsible for decoding actions, it likely accumulates task-specific differences that make naive merging infeasible. This also explains why, as shown in the main paper, applying task masks to preserve localized task information can effectively mitigate such conflicts and enable multi-task unification.

Additional patterns can be observed from Table 2. Interestingly, merging only the vision backbone (A) consistently yields higher success rates than merging both A and B together. This counterintuitive result suggests that, in robotic domains, all modules may exhibit nontrivial task interference, and increasing the number of merged modules amplifies this conflict. In contrast, merging the lm_head (D) has little impact on performance. It is only about 3 points below the fine-tuned baseline. Moreover, combinations such as A + D and A + B + D show negligible difference from A and A + B respectively. To further confirm this, we swapped the lm_head (D) between the LIBERO-Object and LIBERO-Spatial tasks and tested the model on LIBERO-Spatial. Remarkably, it still achieved an 82% success rate over 20 trials, indicating that heads of OpenVLA are largely interchangeable across tasks.

Table 2. Success rates on the four LIBERO task suites when merging different components of OpenVLA [4] using the Iso-CTS [5] merging method. Each merged checkpoint combines four task-specific models, while unmerged components retain their original weights. During evaluation, each subtask is tested with 50 trials. Gray-highlighted row indicates the success rates of individually fine-tuned models.

Method	Spatial	Object	Goal	Long	Avg.
Finetuned	84.7	88.4	79.2	53.7	76.5
A	61.4	60.8	59.0	8.4	47.4
A + B	56.6	58.0	55.6	6.6	44.2
C	0.0	0.0	0.0	0.0	0.0
D	83.4	88.8	72.6	49.6	73.6
A + D	61.0	61.0	62.6	8.4	48.3
A + B + D	58.0	57.4	53.8	7.4	44.2

3.2. Progressive Block-wise Merging of the Language Model

To further analyze why the language model component cannot be merged, we conducted a block-wise study by progressively merging the first k transformer blocks (from 1 to 32) while keeping all other parts fixed to the LIBERO-Spatial task weights. Each merged model was evaluated on 10 trials per subtask, and results are shown in Figure 1. When merging only a few shallow blocks (*e.g.,* up to 8), the model still achieved roughly 80% success rate. However, as the number of merged blocks increased, performance degraded sharply, and beyond 21 merged blocks the model completely failed. This again validates our hypothesis: Task conflicts grow with layer depth, and deeper layers show stronger task-specific divergence that hinders effective merging.

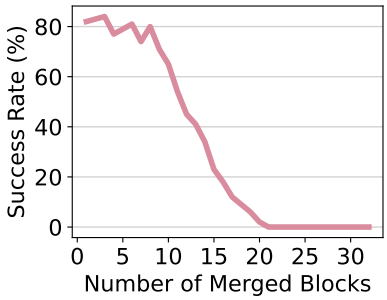

Figure 1. Success rate on the LIBERO-Spatial task when progressively merging the first k language model blocks of OpenVLA [4] using the Iso-CTS [5] merging algorithm. Each configuration merges four task-specific checkpoints and is evaluated over 10 trials per subtask.

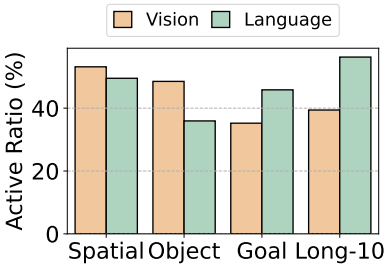

Figure 2. Mask active ratios for each LIBERO task suite, computed for both the vision backbone and the language model components following the same definition as in the main paper. Masks are obtained using the Task Arithmetic [3] merging method with $\lambda = 0.6$.

4. Visualization of Mask Ratios in the Vision Backbone and Language Model

To examine how task masks behave across different components of the VLM, we visualize the mask active ratio for each LIBERO task suite. The mask active ratio measures the proportion of positions where the task mask is active (*i.e.,* set to True), indicating that the model uses the pretrained weight + task vector at that location. In contrast, inactive positions fall back to the pretrained weights only. Because the VLM consists of a vision backbone and

a language model, we compute the active ratio separately for these two parts to analyze their task-specific behavior. A higher active ratio suggests stronger task-specific contributions, while a lower ratio indicates greater reliance on pretrained weights. Figure 2 shows that the patterns across tasks differ markedly between the vision backbone and the language model. For example, LIBERO-Long exhibits very low activation in the vision backbone but the highest activation in the language model, whereas LIBERO-Object shows the opposite trend—high activation in the vision backbone but minimal activation in the language model. LIBERO-Spatial, in contrast, maintains relatively high and balanced activation across both components. These observations suggest that visual and linguistic pathways contribute task-specific information in distinct ways, offering useful insights for future work on understanding and leveraging task specialization in VLA models.

5. Real-World Experiments

5.1. Experimental Setup

Tasks. As shown in Figure 3, we evaluate MergeVLA on three cube-based manipulation tasks using a real SO-101 robotic arm:

(i) PICK & PLACE: the robot must grasp a cube and place it into a black box; success is recorded when the cube is stably placed inside the container.

(ii) PUSH CUBE: the robot must push the cube into a designated white goal zone; success requires the cube to fully enter the region.

(iii) STACK CUBE: the robot must pick up the red cube and place it on top of a blue cube; success is defined by a stable, non-slipping stacked configuration.

Data Collection. We collect demonstrations using the SO-101 arm under a leader–follower teleoperation setup, with two RGB camera views: a fixed top-down camera and a wrist-mounted camera. For each task, we collect 50 demonstrations at a frequency of 20 Hz, with randomized cube starting positions. For the PICK & PLACE and PUSH CUBE tasks, only the red cube is used during data collection. Each demonstration includes synchronized RGB observations, 6 DoF joint actions, and task instructions. We train MergeVLA for 30k steps per task.

Evaluation Protocol. For each model to be evaluated, we perform 20 rollouts per task, with randomized cube initial positions in every rollout. For PICK & PLACE and PUSH CUBE tasks, we use cubes with randomly different colors that are unseen in the training data, providing a visual shift evaluation. Success is determined according to the task-specific criteria defined above. We report the success rate as the percentage of successful trials out of the 20 rollouts.

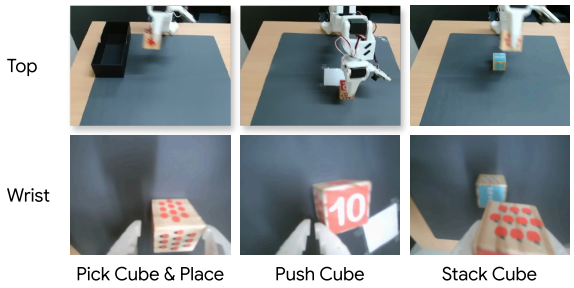

Figure 3. Setup of the real-world SO-101 arm experiments with three cube manipulation tasks.

Table 3. Real-world SO-101 robot performance, reported as success rates (%) over 20 rollouts per task.

Method	Pick & Place	Push Cube	Stack Cube	Avg.
Single-task finetune	90.0	85.0	95.0	90.0
MergeVLA$_{\text{TA}}$	70.0	70.0	60.0	66.7
MergeVLA$_{\text{TIES}}$	**90.0**	**90.0**	**90.0**	**90.0**

5.2. Experimental Results

In Table 3, we present both fine-tuning and model-merging results of MergeVLA on the real SO-101 robotic arm. For fine-tuning, MergeVLA achieves high success rates across all three cube-manipulation tasks. Notably, in PICK & PLACE and PUSH CUBE, the robot is required to operate on cubes whose colors differ from those seen during training. MergeVLA remains robust under this distribution shift, reliably detecting the target object and executing the required manipulation, which highlights its strong visual out-of-distribution generalization in real-world settings.

For model merging, we evaluate MergeVLA using TA and TIES as the merging strategies. TIES-based merging delivers the best overall performance, often matching the results of the corresponding single-task models. This demonstrates that MergeVLA preserves cross-task merging ability even when deployed on physical hardware, and is able to reuse skill components without degradation, an encouraging indication of its practicality for multi-skill real robot systems.

References

[1] Shiqi Chen, Jinghan Zhang, Tongyao Zhu, Wei Liu, Siyang Gao, Miao Xiong, Manling Li, and Junxian He. Bring reason to vision: Understanding perception and reasoning through model merging. *CoRR*, abs/2505.05464, 2025. 1

[2] Yiyang Du, Xiaochen Wang, Chi Chen, Jiabo Ye, Yiru Wang, Peng Li, Ming Yan, Ji Zhang, Fei Huang, Zhifang Sui, Maosong Sun, and Yang Liu. Adamms: Model merging for heterogeneous multimodal large language models with unsupervised coefficient optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025,*

Nashville, TN, USA, June 11-15, 2025, pages 9413–9422. Computer Vision Foundation / IEEE, 2025. 1

[3] Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. 2

[4] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, pages 2679–2713. PMLR, 2024. 1, 2

[5] Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartlomiej Twardowski, Andrew D. Bagdanov, and Joost van de Weijer. No task left behind: Isotropic model merging with common and task-specific subspaces. *CoRR*, abs/2502.04959, 2025. 1, 2

[6] Huaizhi Qu, Xinyu Zhao, Jie Peng, Kwonjoon Lee, Behzad Dariush, and Tianlong Chen. Uq-merge: Uncertainty guided multimodal large language model merging. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 1401–1417. Association for Computational Linguistics, 2025. 1

[7] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. 1