CPCM: CONFORMAL PREDICTION-CALIBRATED CURRICULUM MODEL MERGING FOR MLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Fine-tuning MLLMs on task-specific multimodal data has produced many experts with complementary capabilities. Model merging offers a practical way to integrate these experts into a unified MLLM without joint training or multi-model serving. However, existing merging methods mainly preserve task-specific knowledge through parameter-space heuristics, such as task-vector alignment, pruning, or reweighting, estimating expert capability contribution from weight-space operations rather than statistically calibrated prediction behavior. Moreover, reducing this calibrated signal to a single scalar weight for each expert task vector cannot distinguish the disentangled contributions of different modules. To address these issues, we introduce CPCM, a Conformal Prediction-calibrated Curriculum model Merging framework designed for MLLM expert merging. Specifically, at the expert level, we quantify expert prediction reliability via conformal prediction sets with statistical validity guarantees, deriving curriculum weights and ordering accordingly. At the module level, we further employ a benchmark-module affinity mechanism that projects benchmark-level reliability improvements into module-wise merging strengths for the visual encoder, multimodal projector, and language decoder. Experiments on multiple multimodal benchmarks show that our method surpasses strong task-vector merging baselines, while module-wise weight allocation further enhances the preservation of complementary expert capabilities.

1 INTRODUCTION

Multimodal large language models (MLLMs) are increasingly adapted to specialized multimodal tasks, producing expert checkpoints with distinct capability profiles. Experts fine-tuned for OCR, document understanding, spatial perception, visual question answering, or multimodal reasoning encode complementary task-specific knowledge (Wei et al., 2026; Zhang et al., 2026). Serving these experts independently introduces substantial storage, memory, and deployment overhead, while joint retraining requires access to heterogeneous training data and large-scale computational resources. Model merging addresses this deployment barrier by consolidating complementary expert capabilities into a single parameterized MLLM (Yang et al., 2026).

Despite its popularity, current model merging methods still assess expert contribution mainly through parameter-space operations. Classical approaches such as Task Arithmetic (Ilharco et al., 2022) construct a merged model by adding expert task vectors to the base model with fixed coefficients, while later methods refine task vector composition through alignment, pruning, or reweighting to reduce redundancy and interference (Yadav et al., 2023; Yu et al., 2024; Yang et al., 2024; Cheng et al., 2025). These methods improve parameter-level compatibility, but lack a statistical reliability assessment of expert behavior. This limitation is amplified in MLLM expert merging, where experts are fine-tuned toward distinct capability regions (Wei et al., 2026; Zhang et al., 2026). Although a task vector captures the parameter displacement induced by fine-tuning, it provides limited evidence about whether the resulting expert produces reliable predictions on samples requiring OCR, spatial perception, visual reasoning, or other multimodal capabilities. The expert contribution therefore requires prediction-level calibration in addition to parameter-space compatibility.

Furthermore, a single coefficient summarizes an expert's reliable contribution at the expert-level solely, ignoring that an MLLM task vector spans functionally different modules (Chen et al., 2024a). The visual encoder is more closely related to visual perception and feature extraction whereas

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the language decoder contributes more directly to logical reasoning. Although coefficient-learning methods introduce task-wise or layer-wise coefficients (Yang et al., 2024), such granularity does not explicitly reflect this modular structure, weakening the preservation of module-level complementary knowledge.

Inspired by the stated shortcomings, we introduce **CPCM**, a Conformal Prediction-calibrated Curriculum model Merging framework for MLLM experts. CPCM replaces parameter-space expert assessment with statistically calibrated prediction behavior. At the expert level, CPCM constructs conformal prediction sets on calibration samples and quantifies expert prediction reliability through their set sizes. Since conformal prediction provides validity guarantees under standard exchangeability assumptions, the resulting reliability signal is grounded in calibrated uncertainty rather than heuristic parameter criteria. CPCM converts this expert-level reliability into curriculum weights and ordering, enabling the merging process to inject experts according to their calibrated task-specific contributions. At the module level, the framework further employs benchmark-module affinity to project benchmark-level reliability improvements into module-wise merging strengths for the visual encoder, multimodal projector, and language decoder. This module-aware extension of CP-calibrated reliability preserves the expert-level curriculum while allowing different MLLM components to receive different task-vector injection strengths.

We evaluate CPCM on multimodal benchmarks spanning general visual question answering, document understanding, and spatial understanding. CPCM achieves the best overall performance among task-vector merging baselines, outperforming the strongest baseline, Curriculum Model Merging, by 2.44 points, obtains the strongest results on all calibration-aligned general VQA benchmarks, and maintains competitive document and spatial understanding within a single merged expert.

Our contributions are summarized as follows: (1) CPCM, a conformal prediction-calibrated curriculum merging framework for MLLM experts that estimates expert contribution from calibrated prediction behavior instead of parameter-space operations; (2) a reliability-aware merging design that uses the calibrated signal to determine expert-level curriculum order and derives module-wise merging strengths through benchmark-module affinity for the visual encoder, multimodal projector, and language decoder; and (3) results on multimodal benchmarks covering general visual question answering, document understanding, and spatial understanding, showing CPCM surpasses task-vector merging baseline while limiting degradation from capability-specific expert weaknesses within a single merged expert.

2 Related Work

2.1 Task-vector-based Model Merging

Task-vector-based model merging combines task-specific models by operating on their parameter differences from a shared base model. Task Arithmetic represents each fine-tuned model as a task vector and merges experts by adding scaled task vectors to the base model (Ilharco et al., 2022). TIES-Merging (Yadav et al., 2023) and DARE (Yu et al., 2024) reduce interference through sign-conflict resolution or sparsification, while WUDI (Cheng et al., 2025), AdaMerging (Yang et al., 2024), and RobustMerge (Zeng et al., 2026) further improve merging through layer-wise objectives, coefficient optimization, low-rank structure, or robustness-oriented parameter adjustment; and Curriculum Model Merging introduces curriculum ordering into task-vector merging (He et al., 2026). Despite these differences, existing task-vector merging methods largely assess expert contribution through parameter compatibility, coefficient objectives, or task-vector quality, rather than through calibrated prediction behavior. Even when prediction-level signals such as entropy or confidence are used, they measure sharpness rather than statistically calibrated reliability on capability-specific samples.

2.2 MLLM Expert Merging

MLLM expert merging aims to combine multiple multimodal experts into a single MLLM while preserving their complementary capabilities. In the homogeneous expert setting, these experts share the same base architecture but specialize in different abilities, such as OCR, document understanding, spatial perception, visual question answering, or multimodal reasoning. Recent MLLM-oriented merging methods study this setting from different perspectives. AdaMMS improves hyperparameter

selection for MLLM merging (Du et al., 2025), UQ-Merge uses uncertainty estimates to determine the merging order (Qu et al., 2025), OptMerge analyzes capability-level MLLM merging and task-vector interactions (Wei et al., 2026), and RobustMerge improves robustness by addressing task-vector quality and parameter interference (Zeng et al., 2026). These methods make model merging a practical route for integrating specialized MLLM experts without joint retraining or multi-model serving, while treating expert contribution primarily at the expert-level. In the homogeneous expert setting, capability gains are distributed across functional modules, leaving limited support for allocating reliable contributions across the visual encoder, multimodal projector, and language decoder.

2.3 Reliability Estimation and Conformal Prediction

Expert merging requires a signal for deciding which expert contributions should be preserved and how strongly they should be injected. Existing methods use manually chosen coefficients (Ilharco et al., 2022), validation accuracy or average benchmark performance (He et al., 2026; Zhang et al., 2026), entropy-based objectives (Yang et al., 2024), confidence or uncertainty estimates (Qu et al., 2025), and parameter-space compatibility (Yadav et al., 2023; Yu et al., 2024; Zeng et al., 2026) for this purpose. These signals can guide merging, but they are not equivalent to statistically calibrated reliability. Parameter-space signals describe how task vectors interact, while entropy and confidence measure prediction sharpness without finite-sample coverage guarantees. Average benchmark performance also compresses heterogeneous sample-level behavior into a single score, obscuring whether an expert is reliable on the samples associated with a specific capability.

Conformal prediction provides a distribution-free framework for constructing prediction sets with finite-sample coverage guarantees under standard exchangeability assumptions (Vovk et al., 2005). It has been studied for uncertainty quantification and reliable prediction in deep learning, with recent applications to LLMs (Quach et al., 2024; Noh et al., 2026; Wang et al., 2025a). However, its use in MLLM expert merging remains limited. Existing CP-based work primarily treats conformal prediction as an uncertainty quantification or reliable prediction tool, rather than as a calibrated merging signal for expert weighting, curriculum ordering, and module-wise allocation.

3 Method

We propose **Conformal Prediction-calibrated Curriculum Model Merging (CPCM)**, a training-free framework for merging multiple MLLM experts into a single model. As shown in Figure 1, CPCM uses conformal prediction to estimate calibrated reliability signals on multiple calibration benchmarks, and transforms these signals into expert-level and module-level merging decisions. We first introduce the task-vector merging formulation and the conformal prediction statistics used in our method.

3.1 Preliminaries

Task-vector merging. We consider a set of expert MLLMs $\mathcal{E} = \{1, \dots, K\}$ that share the same architecture and are initialized from a common base model. Let θ_0 denote the parameters of the base model and θ_e denote the parameters of expert e. Following task-vector based model merging, each expert is represented by its parameter displacement from the base model:

$$\tau_e = \theta_e - \theta_0.$$

The base model serves as the reference point for task-vector computation and reliability comparison, but is not treated as an expert to be merged. Given a curriculum order $\pi = (\pi_1, \dots, \pi_K)$ and expert-level merging weights $\{\beta_e\}_{e=1}^{K}$, sequential task-vector merging can be written as

$$\theta^{(0)} = \theta_0, \qquad \theta^{(t)} = \theta^{(t-1)} + \beta_{\pi_t}\tau_{\pi_t}, \quad t = 1, \dots, K.$$

The goal is to determine both the merging weight β_e and the curriculum order π according to the reliable contribution of each expert, rather than relying on fixed coefficients or average benchmark performance.

Conformal prediction. Conformal prediction provides a distribution-free framework for constructing prediction sets with finite-sample coverage guarantees under the exchangeability assumption.

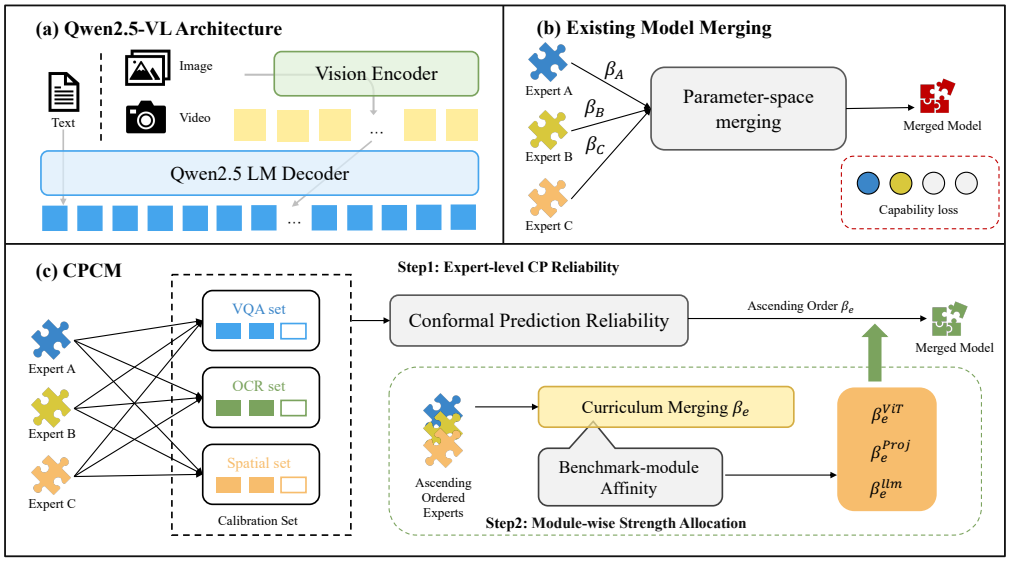

Figure 1: The overview of CPCM. (a) illustrates the Qwen2.5-VL architecture as a representative MLLM framework. (b) shows existing parameter-space merging with fixed expert weights, leading to possible capability loss. (c) presents CPCM's two-stage merging pipeline: (1) expert-level conformal prediction calibrates statistically valid prediction reliability to derive curriculum weights and expert ordering, while (2) module-level benchmark-module affinity projects benchmark-level reliability improvements into module-wise merging strengths for the vision encoder, multimodal projector, and language decoder.

In this work, we use conformal prediction to measure the prediction concentration of MLLMs on multiple-choice calibration samples.

Let $\mathcal{J}$ denote the set of calibration benchmarks, and let $\mathcal{D}_j^{\text{cal}} = \{(x_i, y_i)\}_{i=1}^{n_j}$ be the calibration set for benchmark $j \in \mathcal{J}$, where x_i is the multimodal input and $y_i \in \mathcal{Y}$ is the ground-truth option. For a model with parameters θ, we define the nonconformity score as

$$s_\theta(x_i, y_i) = 1 - p_\theta(y_i \mid x_i),$$

where $p_\theta(y_i \mid x_i)$ is the probability assigned by the model to the ground-truth option. A larger score indicates lower confidence in the correct answer.

For each calibration benchmark j, we compute a benchmark-specific conformal threshold from the calibration scores. Given a miscoverage level α, the empirical threshold is

$$\hat{q}_{\theta,j} = \text{Quantile}_{\lceil (n_j+1)(1-\alpha) \rceil / n_j} \left(\{s_\theta(x_i, y_i)\}_{i=1}^{n_j} \right).$$

For a new sample x, the corresponding prediction set is

$$\Gamma_{\theta,j}(x) = \{y \in \mathcal{Y} : 1 - p_\theta(y \mid x) \le \hat{q}_{\theta,j}\}.$$

Under exchangeability, this prediction set satisfies the marginal coverage guarantee

$$\mathbb{P}\left(y \in \Gamma_{\theta,j}(x)\right) \ge 1 - \alpha.$$

To compare experts under a shared benchmark-specific reliability criterion, CPCM uses the base model to compute the reference threshold $\hat{q}_{0,j}$ for each calibration benchmark j, and reuses this threshold when evaluating the prediction-set size of each expert. Specifically, for expert e, we define

$$\Gamma_{e,j}(x) = \{y \in \mathcal{Y} : 1 - p_e(y \mid x) \le \hat{q}_{0,j}\}.$$

We then measure the average prediction set size (APSS) of expert e on benchmark j as

$$\text{APSS}_{e,j} = \frac{1}{|\mathcal{D}_j^{\text{cal}}|} \sum_{x_i \in \mathcal{D}_j^{\text{cal}}} |\Gamma_{e,j}(x_i)|.$$

A smaller APSS indicates that the model produces more concentrated prediction sets under the same calibration threshold, and therefore reflects stronger calibrated prediction reliability on the corresponding benchmark. CPCM uses this APSS-based reliability statistic to estimate expert-level capability contribution and to guide the subsequent curriculum merging process.

3.2 CP-CALIBRATED EXPERT WEIGHTING

CPCM uses the conformal prediction statistic defined above to estimate how much each expert improves prediction reliability over the base model on each calibration benchmark. For the base model, we compute its prediction set under the benchmark-specific reference threshold $\hat{q}_{0,j}$ and define

$$\mathrm{APSS}_{0,j} = \frac{1}{|\mathcal{D}_j^{\mathrm{cal}}|} \sum_{x_i \in \mathcal{D}_j^{\mathrm{cal}}} |\Gamma_{0,j}(x_i)|.$$

For expert e on benchmark j, the reliability improvement over the base model is measured by

$$\Delta_{e,j} = \mathrm{APSS}_{0,j} - \mathrm{APSS}_{e,j}.$$

A positive $\Delta_{e,j}$ indicates that expert e produces a smaller prediction set than the base model under the same benchmark-specific conformal threshold, and therefore gives more concentrated predictions on benchmark j.

To obtain an expert-level reliability score, CPCM aggregates the benchmark-wise improvements across the calibration benchmark set $\mathcal{J}$:

$$\Delta_e = \frac{1}{|\mathcal{J}|} \sum_{j \in \mathcal{J}} \Delta_{e,j}.$$

This score estimates the overall reliable contribution of expert e across the calibrated capability space. Since the prediction-set size is bounded by the number of answer options, we normalize the reliability improvement as

$$\tilde{\Delta}_e = \mathrm{clip}\left(\frac{\Delta_e}{|\mathcal{Y}| - 1}, 0, 1\right),$$

where $|\mathcal{Y}|$ is the number of candidate answers. The normalized score is then mapped to an expert-level merging weight:

$$\beta_e = \beta_{\min} + (\beta_{\max} - \beta_{\min})\tilde{\Delta}_e.$$

Here $\beta_{\min}$ and $\beta_{\max}$ define the lower and upper bounds of expert participation. This mapping assigns larger weights to experts that provide stronger calibrated reliability improvements, while preventing unreliable or negatively contributing experts from dominating the merge.

The curriculum order is determined by the expert-level weights. Specifically, CPCM sorts experts in ascending order of β_e:

$$\beta_{\pi_1} \le \beta_{\pi_2} \le \cdots \le \beta_{\pi_K}.$$

The merged model is then constructed sequentially:

$$\theta^{(0)} = \theta_0, \qquad \theta^{(t)} = \theta^{(t-1)} + \beta_{\pi_t}\tau_{\pi_t}, \quad t = 1, \ldots, K.$$

This curriculum follows the CP-derived reliability order by injecting lower-weight experts first and higher-weight experts later. In this way, both the weight and the order of each expert are determined by conformal prediction-calibrated behavior on capability-specific calibration data, rather than by fixed coefficients or manually specified orders.

3.3 MODULE-WISE WEIGHT ALLOCATION

Expert-level weighting determines how much an expert contributes to the overall curriculum, but a single scalar weight cannot distinguish where an expert's contribution should enter an MLLM. An MLLM contains functionally different modules, and different expert capabilities depend on these modules in different ways. We therefore extend the CP-derived reliability signal from expert-level weighting to module-wise weight allocation.

We partition the MLLM parameters into three module groups:

$$\mathcal{M} = \{\mathrm{vit}, \mathrm{proj}, \mathrm{llm}\},$$

corresponding to the visual encoder, multimodal projector, and language decoder. Let θ_m and $\tau_{e,m}$ denote the parameters and task vector restricted to module $m \in \mathcal{M}$. The visual encoder is more closely related to visual perception and feature extraction, the projector aligns visual features with the language space, and the language decoder contributes more directly to instruction following, textual generation, and reasoning. CPCM uses this modular structure to assign different merging strengths to different parts of each expert update.

To project benchmark-level reliability improvements to module-level scores, CPCM introduces a benchmark-module affinity matrix

$$A \in \mathbb{R}^{|\mathcal{J}|\times|\mathcal{M}|},$$

where $a_{j,m}$ indicates how strongly calibration benchmark j is associated with module m. The module-level reliability improvement of expert e on module m is computed as

$$\Delta_{e,m} = \frac{\sum_{j\in\mathcal{J}} a_{j,m}\Delta_{e,j}}{\sum_{j\in\mathcal{J}} a_{j,m}}.$$

This projection converts benchmark-level calibrated reliability into module-specific reliability estimates. For example, calibration benchmarks that emphasize visual perception contribute more strongly to the visual encoder score, while benchmarks that emphasize instruction following or reasoning contribute more strongly to the language decoder score.

The module-level score is normalized and mapped to a module-wise merging weight:

$$\tilde{\Delta}_{e,m} = \mathrm{clip}\left(\frac{\Delta_{e,m}}{|\mathcal{Y}|-1}, 0, 1\right),$$

$$\beta_{e,m} = \beta_{\min} + (\beta_{\max} - \beta_{\min})\tilde{\Delta}_{e,m}.$$

The expert-level weight β_e determines the curriculum order, while the module-wise weights $\{\beta_{e,m}\}_{m\in\mathcal{M}}$ determine how strongly each module of the expert task vector is injected.

With module-wise allocation, the sequential merging update becomes

$$\theta_m^{(0)} = \theta_{0,m}, \qquad \theta_m^{(t)} = \theta_m^{(t-1)} + \beta_{\pi_t,m}\tau_{\pi_t,m}, \quad m \in \mathcal{M},\ t = 1, \ldots, K.$$

Equivalently, CPCM applies different CP-calibrated coefficients to different tensor groups during task-vector merging. This design preserves the expert-level curriculum while avoiding uniform injection of the entire expert update. This design allows CP-derived reliability to affect not only the expert order, but also the module-specific strength of each expert update.

4 Experiment

4.1 Set Up

Backbones. We evaluate our method in a 7B-scale MLLM expert merging setting. The base model is Qwen2.5-VL-7B-Instruct (Bai et al., 2025), and all expert models are finetuned from the same base model so that their parameter differences can be represented as task vectors in a shared parameter space. We use four expert models with different capability profiles: Typhoon-OCR-7B (Nonesung et al., 2026) is mainly specialized in optical character recognition (OCR); SpatialThinker-7B (Batra et al., 2026) and ViGoRL-7B-Spatial (Sarch et al., 2026) are stronger in spatial understanding (Spatial); and ThinkLite-VL-7B (Wang et al., 2025b) provides stronger visual question answering (VQA).

Calibration data and evaluation benchmarks. To obtain calibrated sample-level reliability estimates, we use subsets of MMStar (Chen et al., 2024b), BLINK (Fu et al., 2024), and MMBench (Liu et al., 2024) as calibration data, using the model probability assigned to the ground-truth option to define the nonconformity score. For evaluation, we report results on five multimodal benchmarks covering three capability categories. Specifically, we evaluate VQA on MMBench, MMStar, and BLINK, OCR on ChartQA (Masry et al., 2022), and spatial on MMVP (Tong et al., 2024). We run

the evaluations using the VLMEvalKit under a zero-shot setting while GPT-4o-mini (Hurst et al., 2024) is used when answer extraction or response evaluation is required.

Baselines. We compare CPCM with representative task-vector merging baselines, including Task Arithmetic (Ilharco et al., 2022), TIES-Merging (Yadav et al., 2023), DARE-linear (Yu et al., 2024), DARE-ties (Yu et al., 2024), and Curriculum Model Merging (CMM) (He et al., 2026). These baselines cover fixed task-vector addition, conflict-aware merging, sparsification-based merging, and curriculum-based task-vector merging. We also report individual expert results as references for capability specialization and preservation.

Implementation details. For CP calibration, we randomly sample 300 four-choice multiple-choice examples from each calibration benchmark, including MMStar, BLINK, and MMBench. We use a 90% coverage level to compute benchmark-specific CP thresholds. Significantly, the base model is used only as a reference for reliability estimation and task-vector computation, and is not included as an expert during merging. We first perform CP-guided Expert Weighting and further apply module-wise weights through tensor-level filtering in mergekit. All experiments are conducted on 4 NVIDIA A100 GPUs.

4.2 Main Results

Across five multimodal benchmarks, our method delivers the strongest overall performance among all merged models while using only calibration data to guide expert weighting and ordering. As shown in Table 1, CPCM achieves an average score of 69.57, outperforming the strongest baseline(+2.44 over CMM) and even exceeding the best classic task-vector baseline DARE-ties (+3.42). This indicates that calibrated sample-level reliability provides a more effective merging signal than fixed task-vector addition, conflict-aware merging, sparsification-based merging, or average-performance-based curriculum ordering.

The gains are particularly consistent on general visual question answering benchmarks. CPCM achieves 83.20 on MMBench, 64.27 on MMStar, and 56.86 on BLINK, surpassing the calibration-aligned general visual question answering benchmarks. This pattern is aligned with our calibration design, where MMBench, MMStar, and BLINK are used to estimate expert reliability. The results suggest that CP-calibrated reliability estimation better retains expert contributions when the corresponding capability is represented in the calibration data.

On document and spatial understanding, CPCM reaches 85.52 on ChartQA, slightly below CMM at 86.52, and obtains 58.00 on MMVP, trailing DARE-ties at 60.00. While these benchmarks favor specific baselines, their single-task advantages do not translate into stronger overall performance. CPCM maintains competitive document and spatial scores and achieves the highest average across all benchmarks, indicating stronger cross-capability integration within a single merged expert.

Together, the main results support our key claim: transforming calibrated sample-level reliability into expert weighting and curriculum ordering improves the preservation of complementary expert capabilities in MLLM merging. Compared with task-vector merging baselines, CPCM better preserves these capabilities across general VQA, document understanding, and spatial understanding, leading to stronger cross-capability balance.

4.3 Reliability and Module-wise Analysis

Reliability coverage. We analyze how CP-derived reliability signals affect both curriculum ordering and module-wise merging in Table 2. The variants are designed to examine two sources of non-uniformity in MLLM expert merging: reliable capability coverage varies across calibration benchmarks, and expert contributions are unevenly distributed across model modules. This analysis therefore focuses on whether calibrated reliability estimates can provide more than a global expert ranking, and whether they can be further used to guide module-specific parameter merging.

Calibration scope. Using a single calibration benchmark already provides a useful reliability signal for expert weighting, as CP-Single-Calib achieves an average score of 69.10. Expanding the calibration set from MMStar to MMStar, BLINK, and MMBench improves the average score to 69.48, with stronger performance on MMBench and MMStar. This suggests that multi-benchmark calibration

Table 1: Main results on MLLM expert merging across multimodal benchmarks.

	General Visual Question Answering			Document Understanding	Spatial Understanding	
Benchmarks	**MMBench**	**MMStar**	**BLINK**	**ChartQA**	**MMVP**	**Avg.**
Expert model						
SpatialThinker-7B	81.97	60.00	54.92	87.20	57.33	68.28
ViGoRL-7B-Spatial	72.68	56.73	44.56	62.92	58.67	59.11
ThinkLite-VL-7B	82.12	61.90	56.08	86.80	58.00	68.98
Typhoon-OCR-7B	49.54	22.06	6.15	0.00	56.00	26.75
Merged model						
TIES	58.51	40.80	25.14	0.00	16.00	28.09
DARE-linear	81.97	61.00	56.65	69.76	57.30	65.34
DARE-ties	81.81	61.87	56.02	71.04	**60.00**	66.15
Task Arithmetic	73.30	53.27	55.55	69.52	58.67	62.06
CMM	80.42	60.20	54.50	**86.52**	54.00	67.13
CPCM(Ours)	**83.20**	**64.27**	**56.86**	85.52	58.00	**69.57**

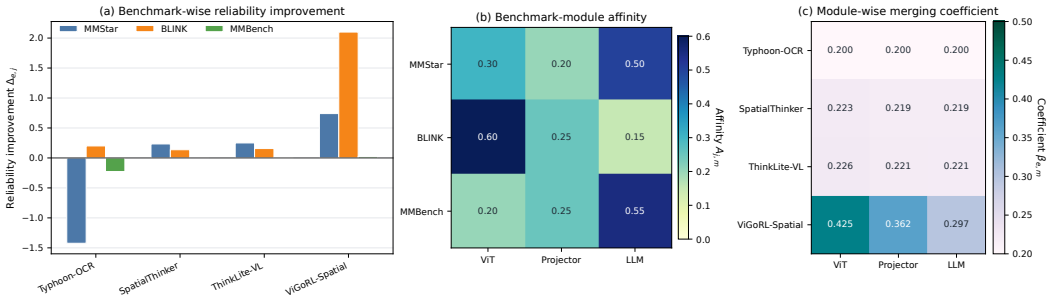

Figure 2: CP-derived reliability and module-wise allocation in CPCM. Panel (a) shows benchmark-wise reliability improvements across experts. Panel (b) shows the benchmark-module affinity matrix used to project benchmark-level reliability to MLLM modules. Panel (c) shows the resulting module-wise merging coefficients.

gives a broader estimate of reliable capability coverage, allowing the curriculum to better identify trustworthy expert contributions across related visual question answering settings.

Profile-based scaling. CP-Rule-Module introduces module-wise weights through manually designed expert profiles, assigning different scaling factors to the ViT, projector, and LLM components according to whether an expert is OCR-oriented, spatial-oriented, or reasoning-oriented. Although this design reflects the heterogeneous structure of MLLMs, its average score drops to 69.11. This indicates that coarse expert profiles are insufficient to determine module-level contributions: an expert's useful parameters may not be confined to the module most intuitively associated with its dominant capability.

Benchmark-module affinity. CPCM replaces fixed profile-based rules with CP-derived module-wise weights. Specifically, benchmark-level reliability improvements are projected to the ViT, projector, and LLM modules through a benchmark-module affinity matrix. Figure 2 visualizes this projection from benchmark-level reliability improvements to module-wise merging coefficients. BLINK assigns higher affinity to the ViT due to its fine-grained visual grounding and spatial perception requirements, while MMStar and MMBench assign larger affinity to the LLM because they involve broader multimodal reasoning and instruction-following behavior. The global CP-derived weight still determines the curriculum order, whereas the module-wise weights control the injection strength of different tensor groups during merging.

Module-wise preservation. With benchmark-module affinity, CPCM achieves the best average score among the variants, reaching 69.57. The improvement is especially reflected on BLINK and MMVP, where CPCM obtains 56.86 and 58.00, respectively. These results suggest that CP-derived module allocation better preserves visual-spatial capabilities while maintaining strong general per-

Table 2: Analysis of CP-guided curriculum and module-wise merging.

	General Visual Question Answering			Document Understanding	Spatial Understanding	
Variant	**MMBench**	**MMStar**	**BLINK**	**ChartQA**	**MMVP**	**Avg.**
CP-Single-Calib	82.97	63.60	56.13	**86.80**	56.00	69.10
CP-Multi-Calib	**83.28**	**64.67**	56.50	85.64	57.33	<u>69.48</u>
CP-Rule-Module	83.05	<u>64.53</u>	<u>56.60</u>	85.36	56.00	69.11
CPCM	<u>83.20</u>	64.27	**56.86**	<u>85.52</u>	**58.00**	**69.57**

formance. Compared with rule-based module scaling, CPCM provides a more adaptive way to account for MLLM module heterogeneity.

Summary. The results show that CPCM benefits from both reliability-guided curriculum ordering and CP-derived module-wise allocation. Multi-benchmark calibration improves expert-level reliability estimation, while benchmark-module affinity extends this signal to module-specific merging. This supports our main claim that module-wise weight allocation further enhances the preservation of complementary expert capabilities within a single merged MLLM.

5 Conclusion and Future Work

We introduce CPCM, a conformal prediction-calibrated curriculum model merging framework for multimodal large language models. CPCM transforms calibrated sample-level reliability estimates into expert-level merging weights and curriculum ordering decisions, and further extends these signals to module-wise weight allocation through benchmark-module affinity. Across multiple multimodal benchmarks, CPCM achieves the best overall performance among task-vector merging baselines and improves cross-capability balance across general VQA, document understanding, and spatial understanding. In future work, we plan to explore dynamic calibration during iterative merging.

AI use statement

In this work, we did not use generative AI tools for tasks with required disclosure. Generative AI tools were used for retrieving and discovering relevant literature and for language editing to improve the clarity and readability of the manuscript. All AI-assisted outputs were reviewed and verified by the authors. In particular, literature identified with the assistance of generative AI was checked against the original sources, and AI-assisted language edits were reviewed to ensure that they did not alter the intended technical meaning or scientific claims. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Reproducibility Statement

We make the experimental and implementation choices needed for reproducing CPCM explicit throughout the paper. Section 3 gives the full CPCM formulation, including CP-based reliability estimation, expert-level curriculum weighting, and module-wise allocation. Section 4 describes the model backbones, calibration data, evaluation benchmarks, baselines, implementation settings, and analysis protocol. Appendix A provides additional experimental configuration details, including benchmark-module affinity values, tensor-group definitions, checkpoint paths, calibration data processing, evaluation protocol, and baseline merge configurations. Appendix B defines the ablation variants used in the reliability and module-wise analysis. Together, these sections document the calibration, merging, evaluation, and ablation procedures used to produce the reported results.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Hunar Batra, Haoqin Tu, Hardy Chen, Yuanze Lin, Cihang Xie, and Ronald Clark. Spatialthinker: Reinforcing scene graph-grounded spatial reasoning via dense rewards, 2026. URL https://arxiv.org/abs/2511.07403.

Chi Chen, Yiyang Du, Zheng Fang, Ziyue Wang, Fuwen Luo, Peng Li, Ming Yan, Ji Zhang, Fei Huang, Maosong Sun, et al. Model composition for multimodal large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11246–11262, 2024a.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models?, 2024b. URL https://arxiv.org/abs/2403.20330.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. *arXiv preprint arXiv:2503.08099*, 2025.

Yiyang Du, Xiaochen Wang, Chi Chen, Jiabo Ye, Yiru Wang, Peng Li, Ming Yan, Ji Zhang, Fei Huang, Zhifang Sui, et al. Adamms: Model merging for heterogeneous multimodal large language models with unsupervised coefficient optimization. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9413–9422. IEEE, 2025.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive, 2024. URL https://arxiv.org/abs/2404.12390.

Baoyi He, Luotian Yuan, Ying Wei, and Fei Wu. Curriculum model merging: Harmonizing chemical llms for enhanced cross-task generalization. *Advances in Neural Information Processing Systems*, 38:39201–39226, 2026.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024.

Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL https://aclanthology.org/2022.findings-acl.177.

Kangjun Noh, Seongchan Lee, Ilmun Kim, and Kyungwoo Song. Multi-llm adaptive conformal inference for reliable llm responses. *arXiv preprint arXiv:2602.01285*, 2026.

Surapon Nonesung, Natapong Nitarach, Teetouch Jaknamon, Pittawat Taveekitworachai, and Kunat Pipatanakul. Typhoon ocr: Open vision-language model for thai document extraction, 2026. URL https://arxiv.org/abs/2601.14722.

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*, pp. 1401–1417, 2025.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi Jaakkola, and Regina Barzilay. Conformal language modeling. In *International Conference on Learning Representations*, volume 2024, pp. 11654–11681, 2024.

Gabriel Sarch, Snigdha Saha, Naitik Khandelwal, Ayush Jain, Michael J. Tarr, Aviral Kumar, and Katerina Fragkiadaki. Grounded reinforcement learning for visual reasoning, 2026. URL https://arxiv.org/abs/2505.23678.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms, 2024. URL https://arxiv.org/abs/2401.06209.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005. Springer, New York.

Qingni Wang, Tiantian Geng, Zhiyuan Wang, Teng Wang, Bo Fu, and Feng Zheng. Sample then identify: A general framework for risk control and assessment in multimodal large language models. In *International Conference on Learning Representations*, volume 2025, pp. 64280–64297, 2025a.

Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement, 2025b. URL https://arxiv.org/abs/2504.07934.

Yongxian Wei, Runxi Cheng, Weike Jin, Enneng Yang, Li Shen, Lu Hou, Sinan Du, Chun Yuan, Xiaochun Cao, and Dacheng Tao. Optmerge: Unifying multimodal llm capabilities and modalities via model merging. In *International Conference on Learning Representations*, volume 2026, pp. 61571–61595, 2026.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, volume 2024, pp. 22743–22763, 2024.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities. *ACM Computing Surveys*, 58(8):1–41, 2026.

Linyan Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Icml*, volume 2, pp. 21, 2024.

Fanhu Zeng, Haiyang Guo, Fei Zhu, Li Shen, and Hao Tang. Robustmerge: Parameter-efficient model merging for mllms with direction robustness. *Advances in Neural Information Processing Systems*, 38:71071–71095, 2026.

Dengming Zhang, Xiaowen Ma, Zhenliang Ni, Zhenkai Wu, Han Shu, Xin Jiang, and Xinghao Chen. Expert merging: Model merging with unsupervised expert alignment and importance-guided layer chunking. In *International Conference on Learning Representations*, volume 2026, pp. 21780–21803, 2026.

Table 3: Benchmark-module affinity matrix used by CPCM.

Benchmark	ViT	Projector	LLM
MMStar	0.30	0.20	0.50
BLINK	0.60	0.25	0.15
MMBench	0.20	0.25	0.55

Table 4: Base and expert checkpoints used in the experiments.

Model	Role	Checkpoint path	Capability profile
Qwen2.5-VL-7B-Instruct	Base	`Qwen/Qwen2.5-VL-7B-Instruct`	Generalist reference model
Typhoon-OCR-7B	Expert	`typhoon-ai/typhoon-ocr-7b`	OCR / document understanding
SpatialThinker-7B	Expert	`OX-PIXL/SpatialThinker-7B`	Spatial reasoning
ViGoRL-7B-Spatial	Expert	`gsarch/ViGoRL-7b-Spatial`	Spatial / visual reasoning
ThinkLite-VL-7B	Expert	`russwang/ThinkLite-VL-7B`	Visual reasoning

A Experimental Configuration

A.1 Module-wise Allocation Values

Table 3 reports the benchmark-module affinity matrix used for module-wise allocation. The same coefficient range, $\beta_{\min} = 0.2$ and $\beta_{\max} = 0.7$, is used for both global and module-wise weights.

A.2 Tensor Groups for Module-wise Merging

CPCM applies module-wise coefficients through tensor-level filtering in mergekit, using the following tensor name patterns:

- **ViT**: visual encoder tensors, including `visual.patch_embed.` and `visual.blocks.`.
- **Projector**: multimodal projector / visual-language merger tensors, including `visual.merger.`.
- **LLM**: language decoder tensors, including `model.embed_tokens`, `model.layers.`, `model.norm`, and `lm_head`.

A.3 Models and Calibration Data

Table 4 lists the base model and expert checkpoints used in our experiments. The base model is used as the reference model for task-vector computation and CP-based reliability estimation, while only the four expert checkpoints are merged.

For CP calibration, we use local TSV calibration pools from MMStar, BLINK, and MMBench. For MMBench, the calibration pool is based on MMBench_DEV_EN_V11. From each benchmark, we randomly sample 300 examples with seed 42 after filtering out rows with missing question, answer, image, or A/B/C/D options, and keeping only examples whose ground-truth answer is one of A/B/C/D. The calibration prompt contains the question and four candidate options, followed by the instruction "Answer with the option letter only." To compute the answer probability, we take the last-token logits corresponding to the Qwen option-token IDs for A/B/C/D, namely `[32, 33, 34, 35]`, and apply a softmax over these four logits. The resulting normalized probability assigned to the ground-truth option is used to compute the nonconformity score.

A.4 Evaluation Protocol

All evaluations are conducted with VLMEvalKit using its default model and dataset prompts. Table 5 summarizes the evaluation benchmarks, splits and sample counts.

Table 5: Evaluation protocol used for the reported benchmark results.

Benchmark	Capability category	Split	#Samples
MMBench	General VQA	MMBench_DEV_EN_V11 / dev	8105
MMStar	General VQA	Default	2106
BLINK	General VQA	Default	4980
ChartQA	Document understanding	ChartQA_TEST	2500
MMVP	Spatial understanding	MMVP	300

Table 6: Definitions of ablation variants used in the reliability and module-wise analysis.

Variant	Calibration benchmarks	Module weights	Purpose
CP-Single-Calib	MMStar	Global CP-derived weight only	Single-benchmark calibration
CP-Multi-Calib	MMStar, BLINK, MMBench	Global CP-derived weight only	Multi-benchmark calibration
CP-Rule-Module	MMStar, BLINK, MMBench	Rule-based profile scaling	Manual module-prior allocation
CPCM	MMStar, BLINK, MMBench	Benchmark-module affinity	Full method

Our evaluation script does not override VLMEvalKit decoding parameters such as temperature, top-p, or maximum generation length. GPT-4o-mini is used by VLMEvalKit as the default judge for MCQ-like datasets unless dataset-specific exact matching applies.

A.5 Baseline Merge Configurations

All baselines use Qwen2.5-VL-7B-Instruct as the base model and merge the same four expert checkpoints. For one-shot baselines, the expert task vectors are merged in parallel. Task Arithmetic uses a fixed coefficient of 0.3 for each expert and disables normalization. TIES-Merging also uses a coefficient of 0.3 for each expert, with density 0.5, int8 masking, and normalization enabled. DARE-linear uses density 0.5 with rescaling enabled and normalization disabled, while DARE-ties uses density 0.5 with int8 masking and rescaling enabled, also with normalization disabled. None of these one-shot baselines uses calibration data.

CMM is implemented as iterative task arithmetic. It uses the manually specified curriculum order Typhoon-OCR-7B $\rightarrow$ ThinkLite-VL-7B $\rightarrow$ SpatialThinker-7B $\rightarrow$ ViGoRL-7B-Spatial, with step-wise coefficients 0.3, 0.4, 0.5, and 0.5, respectively. The output of each merging step is used as the base model for the next step, and normalization is disabled.

CPCM follows the same iterative merging protocol as CMM, but replaces the manually specified order and fixed step-wise coefficients with CP-derived curriculum weights and module-wise coefficients. The CPCM curriculum order is Typhoon-OCR-7B $\rightarrow$ SpatialThinker-7B $\rightarrow$ ThinkLite-VL-7B $\rightarrow$ ViGoRL-7B-Spatial. Module-wise coefficients are applied through tensor-level filtering, as described in Appendix A.2. CPCM is the only merging method in our experiments that uses calibration data.

B Additional Analysis

B.1 Ablation Variant Definitions

Table 6 summarizes the variants used in the reliability and module-wise analysis. All variants use the same base model and expert checkpoints; they differ only in the calibration scope, curriculum signal, and module-wise allocation strategy.

For CP-Single-Calib, expert reliability and curriculum order are computed using only MMStar calibration samples. CP-Multi-Calib extends the calibration scope to MMStar, BLINK, and MMBench, but still applies a single global coefficient to each expert task vector. CP-Rule-Module keeps the multi-benchmark CP-derived curriculum but applies manually defined module scaling factors according to expert profiles. CPCM replaces these manually specified profile rules with benchmark-module affinity, which maps benchmark-level reliability improvements to module-wise coefficients.

The rule-based module scaling factors used by CP-Rule-Module are listed below:

- **OCR profile**: ViT $= 0.55$, Projector $= 1.15$, LLM $= 0.95$.
- **Spatial profile**: ViT $= 1.10$, Projector $= 0.90$, LLM $= 0.75$.
- **Reasoning profile**: ViT $= 0.65$, Projector $= 0.90$, LLM $= 1.10$.
- **Balanced profile**: ViT $= 0.85$, Projector $= 1.00$, LLM $= 0.95$.