Beyond Disjoint Tasks: Towards More Natural Continual Learning for Vision-Language Models

Xiang Xu[1], Yiyang Su[1,2⋆], Tianchen Zhao[1], Zheng Zhang[1], Zhuowen Tu[3,4], Anil Jain[2,3], and Jon Wu[3]

[1]Amazon AGI, [2]Michigan State University, [3]AWS AI Labs, [4]University of California, San Diego

Abstract. Continual learning methods for vision-language models are developed on benchmarks where each new task introduces entirely new domain knowledge. Real-world task sequences are more natural: they routinely share visual concepts, language patterns, and even training samples across stages. However, existing mixture-of-expert methods that assign one expert per task with fixed routing can split similar inputs across different experts and degrade performance. We introduce Semantic Overlap-aware Continual Learning (SOCiaL), a simple framework designed to identify and leverage shared structure across tasks. SOCiaL employs a Gaussian mixture model to estimate contextual similarity, generate synthetic replay samples, and guide expert routing. When task contexts are highly similar, their adapters are consolidated and the router is updated accordingly. To study this more realistic continual learning scenario, we also present UCIT-O, a new benchmark with three protocols that progressively increase semantic similarity across tasks. Across both disjoint and overlapping benchmarks, SOCiaL consistently outperforms existing methods, achieving 7.35, 8.04, and 3.07–9.77 points above the strongest baseline on CoIN, UCIT, and UCIT-O while reducing deployed adapters by up to threefold.

Keywords: Continual Learning · Vision-Language Models · Task Overlap · Mixture-of-Experts

1 Introduction

Vision-language models (VLMs) [1,5,15,16,27,28,47] must learn new capabilities over time without forgetting existing ones, a core requirement for the next generation of foundation models [2,26] that support an expanding range of applications [14,25,31–33,42,45,46]. Continual learning methods, from regularization-based approaches [12,13,20,23,24,36], LoRA based approaches [8,17,18,44], to Mixture of Expert (MoE) based methods for VLM [4,6,30,41], all assume each task introduces entirely new content (Fig. 1). In practice, new tasks routinely reuse visual concepts, language patterns, or training samples from earlier ones. We call this more realistic setting *natural continual learning*: tasks arrive

⋆Work done during an internship at Amazon.

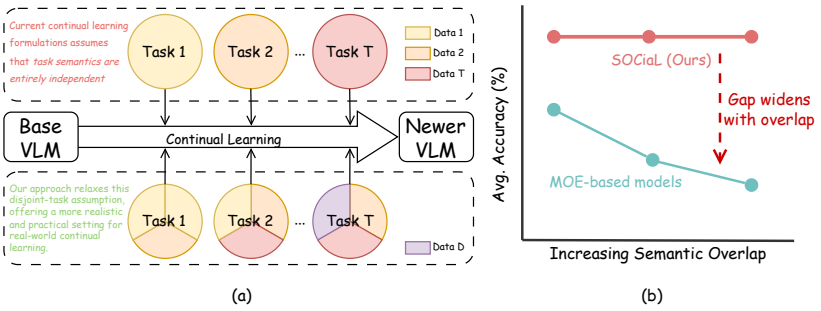

Fig. 1: (a) Existing continual learning benchmarks assume that sequential tasks occupy separate domains (top). In practice, tasks often share images, concepts, or language patterns (bottom). **(b)** As inter-task overlap increases, HiDe [6] degrades sharply: its standard adapter merging mixes the internal factors of different tasks, producing conflicted weight updates that corrupt the merged expert. SOCiaL avoids this by using a generative model to detect overlap and merge adapters in full weight space, eliminating the cross-term artifacts.

sequentially but are not artificially separated. This is especially true for foundation models, where the large scale of training data makes it impractical to ensure that sequential tasks remain disjoint. *Can a continual learner exploit shared content across tasks rather than treating it as a source of interference?* No existing method does so.

We address this gap with Semantic Overlap-aware Continual Learning (SOCiaL), built around one central idea: *task context matters* in natural continual learning. Prior MoE methods [4,6] treat each task as independent: they train separate experts, route with fixed classifiers, and never ask whether two tasks share content. SOCiaL instead conditions all three decisions, when to merge, how to continual adapt, and where to route, on a measured notion of inter-task context similarity or semantic overlap. It introduces three modules beyond the standard MoE-based VLM setup: (1) a generative model that captures each task's context and guides the downstream LoRA merging and router optimization; (2) continual LoRA merging and fine-tuning when tasks share the same semantic context; and (3) an elastic router that adapts as the context landscape changes.

SOCiaL fits a lightweight Gaussian mixture model (GMM) to each task's distribution. GMMs naturally model multimodal distributions with minimal fitting cost and offer three benefits for continual learning. First, they provide asymmetric density-based coverage scores that measure directional semantic overlap between tasks, guiding which expert LoRAs should be merged. Second, they generate synthetic embeddings for router training without storing real samples, preventing catastrophic forgetting in the router. Third, they are trivial to merge when two or more tasks share similar context, keeping the density model consistent as experts are consolidated.

With the GMM's guidance, LoRA adapters within the same context are merged in full weight space rather than mixing the low-rank factors [8, 17, 18]. Standard LoRA merging averages the factors directly, which creates cross-term

artifacts that grow quadratically with the number of merged experts. Merging in full weight space eliminates these artifacts entirely. A lightweight fine-tuning step then adapts the merged adapter on the current task's data, regularized by the original experts so that new knowledge is further learned without forgetting old capabilities. When tasks share the same semantic context, we observe that merging can even improve performance on both old and new tasks. Finally, the router is formulated as a contextual bandit with a binary reward, learning to select the right expert even for ambiguous inputs. GMM replay exposes the router to samples from all seen tasks, including the overlap regions where mistakes are most likely. Unlike standard classifiers that require fixed or incremental task labels, the bandit formulation naturally handles changing expert dynamics: it adapts when new experts are added and when merges reduce the action space.

To measure how semantic overlap affects continual learning and to simulate a more natural scenario, we introduce the UCIT-O benchmark. Built on the UCIT suite, UCIT-O arranges six vision-language datasets into ten sequential tasks under three protocols of increasing overlap: *Selective Overlap* (Protocol I), where only some tasks share content; *Universal Overlap* (Protocol II), where every task shares data partitions with others; and *Sample-Level Overlap* (Protocol III), where identical samples appear across tasks, the most realistic scenario. On this benchmark, we confirm that existing MoE-based VLM continual learning methods degrade significantly under semantic overlap. SOCiaL, in contrast, consistently outperforms all baselines in both the disjoint-domain setting and this more natural one. Therefore, our contributions can be summarized as:

1. Natural continual learning and SOCiaL framework: We formalize a more realistic continual learning setting where sequential tasks share content, and propose SOCiaL, a framework that detects inter-task overlap, merges redundant experts, and routes inputs, all guided by a single GMM per task.
2. GMM-guided LoRA merging and elastic routing: In the similar context measured by GMM, adapters are merged in full weight space, eliminating the cross-term artifacts of standard low-rank merging and reducing deployed adapters to K, the number of distinct task types. The router, formulated as a contextual bandit, adapts naturally as experts are added or merged.
3. UCIT-O benchmark and state-of-the-art results: We introduce UCIT-O, the first continual VLM benchmark with controlled inter-task overlap across three protocols. SOCiaL achieves 70.17% on CoIN (+7.35 over the strongest baseline), 72.23% on UCIT (+8.04), and up to 70.18% on UCIT-O (+3.07 to +9.77 across protocols), with the gap widening as overlap increases.

2 Related Work

Continual Learning. Continual learning (CL) seeks to enable models to assimilate new knowledge without catastrophic forgetting of prior tasks [27]. Classical CL in computer vision can be grouped into three families: replay-based, regularization-based, and architecture-based approaches. Replay methods [20,23] mitigate forgetting by storing exemplars or generating synthetic memories. Their

memory cost grows linearly with task number and often conflicts with privacy constraints. Regularization-based strategies [12–14, 30] introduce parameter importance metrics or gradient constraints to preserve stability. These scale poorly to billion-parameter VLM backbones due to expensive Fisher or Hessian computations. Architecture-based methods [24, 36] dynamically allocate submodules per task. They achieve strong performance with bounded forgetting but at the cost of increased model size. Continual learning is also closely related to domain adaptation, which transfers knowledge across shifting input distributions [34]; unlike that setting, we assume no simultaneous access to data from past tasks. Recent LoRA-based continual learning methods [8, 17, 18, 44] focus on preventing interference between task-specific adapters but target class-incremental tasks rather than the multi-task instruction tuning setting we address. These methods all assume disjoint tasks and focus on preventing forgetting through subspace isolation.

LoRA Merging and Model Composition. When multiple LoRA adapters [9] need to be combined, several strategies exist. Task Arithmetic [11] averages full-rank weight changes, TIES-Merging [35] resolves sign conflicts by trimming small values, and DARE [37] randomly drops entries before merging. In the continual learning setting, Merge-before-Forget [22] merges sequential LoRAs. Zhou *et al*. [43] align updates in shared subspaces. CONEC-LoRA [21] splits adapters into shared and task-specific parts. I-LoRA [39] iteratively merges routing-tuned adapters. None of these address the cross-term problem that arises when averaging LoRA's low-rank factors directly. Nor do they decide *when* merging is appropriate based on how similar the tasks actually are.

Continual Learning for VLM/MLLM. Extending CL to multimodal large language models (MLLMs) has recently attracted increasing attention [7, 19]. Benchmarks such as CoIN [4] and UCIT [6] evaluate sequential instruction tuning, but both focus on disjoint-task scenarios. MoE-style approaches assign LoRA modules as experts with learned routing [4], combine instance- and task-level routing with momentum updates [10], or dynamically expand the expert pool based on activation patterns [40]. Other methods decompose layers into task-specific and shared components guided by prototype matching [6], construct dual embeddings that combine instruction-aware and cross-task features [3], or progressively allocate LoRA adapters from a shared pool [38]. MLLM-CL [41] further introduces domain and ability continual learning settings with multimodal routing. When tasks do overlap, their fixed routing and independent expert pools lead to misrouting and redundant adapters.

Our approach differs from both lines of work in three ways: (i) we use a fitted GMM per task to explicitly measure overlap between tasks, so the system knows *when* to merge and *when* to keep experts separate; (ii) we merge adapters by averaging their full weight changes (B_iA_i) rather than mixing the low-rank factors, which eliminates the harmful cross-terms that all prior merging methods inherit; (iii) our router is trained on GMM-generated replay from all tasks,

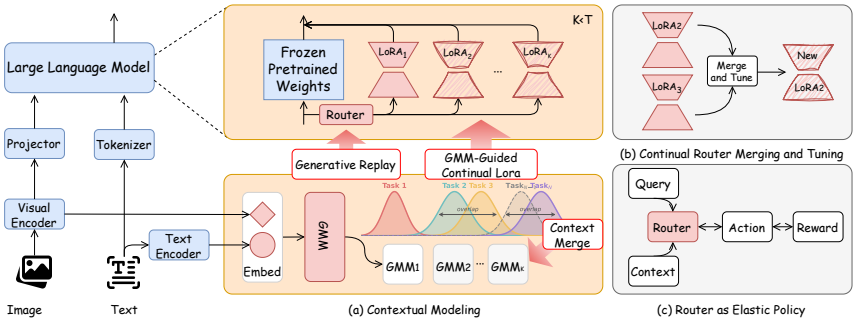

Fig. 2: SOCiaL framework. SOCiaL leverages a per-task GMM (a) to serve triple duty: it models the semantic context of each task, guides merging of LoRAs within the same semantic context, and generates synthetic embeddings for the router's continual updates. When semantic overlap is detected, the corresponding LoRA adapters are merged in full weight space and briefly fine-tuned (b). The router (c) is trained on real and GMM-generated embeddings with elastic output that adapts during continual learning and LoRA merging.

including the ambiguous overlap regions, so it learns to handle inputs that sit between task boundaries rather than treating every boundary as a hard wall.

3 Methodology

Problem Formulation. We consider a sequence of T tasks $\{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$, where each task $\mathcal{D}_t = \{(v_i, q_i, y_i)\}_{i=1}^{N_t}$ consists of images v_i, text queries q_i, and target responses y_i. The goal is to adapt a pretrained VLM/MLLM f_{θ_0} to each new task sequentially while retaining performance on all previous ones, *without storing raw data from past tasks*. Unlike prior MoE-based continual learning methods [4, 6], we make no assumption about inter-task relationships: consecutive tasks may be entirely disjoint, partially overlapping, or nearly identical. We call this more realistic setting *natural continual learning*.

SOCiaL Framework. Effective natural continual learning requires understanding each task's semantic context and the relationships across tasks. Following a MoE-based VLM setup [4, 6], each task receives a dedicated expert LoRA adapter trained sequentially on its own data. The framework must then answer three questions: *when* to merge experts whose tasks share overlapping content, *how* to perform the merge so that knowledge is consolidated without interference, and *where* to route inputs when the expert pool grows or shrinks over time. Figure 2 illustrates our framework for addressing natural continual learning, organized around the three questions above.

For the *when*, we need a model that can cheaply represent each task's input distribution. When a new task t arrives, we extract embeddings from the base model (*e.g.*, CLIP embeddings for LLaVA) and fit a GMM to capture the task's semantic context (Sec. 3.1), which plays the central role in our framework.

GMMs naturally model multimodal distributions with minimal fitting cost, provide asymmetric coverage scores that quantify directional semantic overlap between task distributions, and are trivial to merge when their corresponding tasks are highly overlapped.

For the *how*, when the GMM coverage scores indicate sufficient overlap, we merge the corresponding expert LoRAs in full weight space using interference-free delta merging, avoiding the cross-term artifacts of standard low-rank factor averaging (Appendix C.3). The merged adapter provides a strong initialization that is briefly fine-tuned with regularization toward the original experts (Sec. 3.2), adapting to the new task without forgetting the old ones. The corresponding GMMs are also merged to reflect the consolidated distribution.

For the *where*, because the expert pool grows as tasks arrive and shrinks as overlapping experts are merged, the router must update continuously in an elastic manner. We formulate it as a contextual bandit trained with reinforcement learning on current-task embeddings and synthetic replay generated from all previous GMMs, enabling it to adapt as the expert pool evolves (Sec. 3.3).

Our framework can be interpreted through conditional risk decomposition (Appendix C.1): the total risk decomposes into per-task conditional risks weighted by task priors. LoRA training minimizes each expert's conditional risk, GMM-based semantic overlap measurement can identify when two or more terms can be safely collapsed, interference-free merging consolidates the corresponding experts without increasing risk, and the bandit router minimizes routing error across the reduced expert pool.

3.1 GMM-Based Task Context Modeling

Embedding extraction and GMM Fitting. At task t, for each sample (v_i, q_i), we extract a vision embedding $z_i^v = \mathrm{Enc}_v(v_i) \in \mathbb{R}^{d_v}$ and a text embedding $z_i^t = \mathrm{Enc}_t(q_i) \in \mathbb{R}^{d_t}$ using frozen encoders from the base model without introducing additional models in the pipeline. Then, we concatenate them into $z_i = [z_i^v;\, z_i^t] \in \mathbb{R}^d$ with $d = d_v + d_t$. With that, we can fit a Gaussian Mixture Model with M components and diagonal covariance:

$$p(z \mid \mathcal{D}_t) = \textstyle\sum_{m=1}^{M} \pi_m \mathcal{N}(z;\, \mu_m, \Sigma_m)\,. \tag{1}$$

Diagonal covariance keeps computation tractable in high-dimensional space and avoids overfitting with limited per-task data. The component means $\{\mu_m\}$ serve as prototypes of the task's distribution. The mixture naturally captures multimodal structure when a task spans multiple visual domains.

Semantic overlap measurement. The fitted GMMs also provide a natural measure of inter-task overlap. Let $\mathcal{G}_k$ denote the GMM fitted on task k (Eq. 1). For each task k, we establish a density floor τ_k as the q-th percentile (q=5 by default) of the log-likelihoods that $\mathcal{G}_k$ assigns to its own training embeddings $\mathcal{Z}_k = \{z_i\}_{i=1}^{N_k}$:

$$\tau_k = \mathrm{Percentile}_q\big(\{\log p(z \mid \mathcal{G}_k) : z \in \mathcal{Z}_k\}\big)\,. \tag{2}$$

The directional coverage of task t's data under task k's GMM is then the fraction of t's embeddings that exceed this floor:

$$c_{t\leftarrow k} = \frac{1}{N_t} \sum_{i=1}^{N_t} \mathbf{1}\big[\log p(z_i \mid \mathcal{G}_k) \geq \tau_k\big] , \tag{3}$$

where $z_i \in \mathcal{Z}_t$ are the embeddings of task t and $c_{t\leftarrow k} \in [0, 1]$. Here $\mathcal{Z}_t$ are task t's real current training embeddings and $\log p(z_i \mid \mathcal{G}_k)$ is evaluated analytically from the stored GMM parameters; the generative samples of Sec. 3.1 are used only for router replay, never for overlap detection. Given an overlap threshold ρ, when both $c_{t\leftarrow k} \geq \rho$ and $c_{k\leftarrow t} \geq \rho$ (mutual overlap), the tasks share sufficient distributional support to justify merging their experts; when only one direction exceeds ρ, one task is a subset of the other. Full details are in Appendix C.2.

GMM merging. When two distributions are merged because the high contextual similarity of their tasks, their GMMs are combined by concatenating the component sets and renormalizing the mixture weights:

$$p(z \mid \mathcal{G}_{k\cup t}) = \sum_{m\in\mathcal{G}_k\cup\mathcal{G}_t} \frac{\pi_m}{\sum_{m'} \pi_{m'}} \mathcal{N}(z;\, \mu_m, \Sigma_m) . \tag{4}$$

This ensures that subsequent overlap detection and router replay reflect the merged distribution. The operation is closed-form and requires no refitting, which is a key reason we chose GMMs as the task context model.

Generative replay. Because GMMs are generative models, they can produce unlimited synthetic embeddings that faithfully represent each task's distribution. This property makes them a natural fit for continual learning: when training the router at task t, we sample S synthetic embeddings from each previous GMM, $\tilde{z}_{t'} \sim p(z \mid \mathcal{G}_{t'})$ for $t' < t$, and combine them with real embeddings from $\mathcal{D}_t$. The result is a balanced training set covering all tasks seen so far, without storing any real samples from past tasks. As experts are merged, the corresponding GMMs are also merged, so the replay distribution automatically reflects the current expert pool. The total memory cost is $\mathcal{O}(T \cdot M \cdot d)$ for storing all GMM parameters, orders of magnitude smaller than raw sample replay.

3.2 In-context Continual LoRA Adaptation

Each task t receives a dedicated LoRA adapter [9] that modifies the frozen VLM. For a pretrained weight matrix $W_0 \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$, the adapted weight is $W_t = W_0 + \alpha B_t A_t$, where $A_t \in \mathbb{R}^{r\times d_{\text{in}}}$, $B_t \in \mathbb{R}^{d_{\text{out}}\times r}$, and $r \ll \min(d_{\text{out}}, d_{\text{in}})$. Training each adapter independently on its own data prevents forgetting at the parameter level, but it requires a perfect router to direct each input to the correct expert, which is hardly hold. Moreover, when two tasks share similar semantic context, their adapters encode redundant knowledge in overlapping parameter subspaces, increasing both the number of hosted experts and the

difficulty of routing. In this natural continual learning setting, consolidating semantically similar experts reduces routing ambiguity and deployment cost. The question then becomes how to merge these adapters in the similar context without degrading performance on either task.

In-context LoRA Merging. The obvious approach to combining LoRA adapters, averaging the low-rank factors directly, introduces cross-term artifacts that grow quadratically with the number of merged adapters (Appendix C.3). We avoid this by merging in full weight space. We first compute each adapter's actual weight change per layer, $\Delta W_i^{(\ell)} = B_i^{(\ell)} A_i^{(\ell)} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ (the superscript ℓ indexes the layer), and then form the weighted average in delta space, $\Delta\bar{W}^{(\ell)} = \sum_{i=1}^{n} \omega_i \, \Delta W_i^{(\ell)}$. Because each $\Delta W_i^{(\ell)}$ is computed from a single adapter's factors, the summation is free of cross-terms by construction. Optionally, DARE-style stochastic sparsification [37] can be applied to each $\Delta W_i^{(\ell)}$ before summation to further suppress low-magnitude noise. The merged delta is then compressed back into LoRA format via a rank-r truncated SVD:

$$\Delta\bar{W}^{(\ell)} \approx \bar{B}^{(\ell)} \bar{A}^{(\ell)}, \quad \bar{B}^{(\ell)} = U_r \sqrt{\Sigma_r}, \quad \bar{A}^{(\ell)} = \sqrt{\Sigma_r} \, V_r^{\top}, \tag{5}$$

where $U_r \Sigma_r V_r^{\top}$ is the rank-r truncated SVD of $\Delta\bar{W}^{(\ell)}$. This in-context LoRA merging is only triggered when tasks share similar semantic context, so the adapters are functionally aligned (they map shared-region inputs to similar outputs) and the merged delta's energy stays concentrated in its top-r singular directions, making the SVD compression near-lossless in practice. If either task already belongs to a merge group, all groups are unified and re-merged from scratch using the original adapters, preventing error accumulation from cascaded in-context merges.

Continual LoRA Adaptation. The merged adapter encodes unified tasks' knowledge but has not been trained on their combined data. We refine it with a small adaptation on the current task, anchored to the merged initialization so it does not drift too far:

$$\mathcal{L}_{\text{ft}} = \mathcal{L}_{\text{CE}}(\bar{w}; \mathcal{D}_t) + \lambda \sum_{\ell} \left\| \bar{w}^{(\ell)} - \bar{w}_{\text{init}}^{(\ell)} \right\|_2^2, \tag{6}$$

where $\bar{w}_{\text{init}}$ denotes the merged weights from Eq. (5) (a deterministic SVD-compressed initialization, not learned) and λ controls how strongly the fine-tuned weights stay near the merge. The first term adapts to the current task; the second prevents the merged knowledge from being overwritten. This two-step approach separates the merge (exact, no data needed) from the adaptation (needs data). The result is a merged expert that works well on both tasks. Merging also reduces the number of hosted adapters from T to K, the number of distinct task types. This provides a direct deployment benefit in memory and serving cost.

Algorithm 1 SOCiaL: Semantic Overlap-aware Continual Learning

Require: Tasks $\{\mathcal{D}_t\}_{t=1}^{T}$, VLM f_{θ_0}, overlap threshold ρ

$\mathcal{E} \leftarrow \emptyset$ ▷ expert pool
for $t = 1, \ldots, T$ **do**
  Fit GMM $\mathcal{G}_t$ on embeddings of $\mathcal{D}_t$ ▷ Sec. 3.1
  Train LoRA (A_t, B_t) on $\mathcal{D}_t$; register in $\mathcal{E}$
  for each $e \in \mathcal{E} \setminus \{t\}$ **do**
    if $c_{t \leftarrow e} \geq \rho$ and $c_{e \leftarrow t} \geq \rho$ **then** ▷ Eq. 3
      Merge: $\Delta\bar{W}^{(\ell)} \leftarrow \sum_i \omega_i B_i^{(\ell)} A_i^{(\ell)}$; compress via SVD ▷ Eq. 5
      Fine-tune merged adapter; unify GMMs ▷ Eqs. 6,4
    end if
  end for
  Train router π_ϕ on $\mathcal{D}_t$ + GMM replay ▷ Eq. 7
end for

3.3 Continuous Router as Elastic Policy

We formulate this continuous router into a contextual bandit $(\mathcal{C}, \mathcal{A}, r)$ problem, where the context is the embedding $z \in \mathbb{R}^d$, the action is the choice of expert $a \in \{1, \ldots, E_t\}$, and $r(z, a)$ is a reward signal: the action space naturally grows and shrinks with the expert pool, and the policy is trained with a scalar reward rather than fixed task labels, which may be ambiguous when tasks overlap.

Policy network. The router is a two-layer MLP π_ϕ that maps the embedding $z \in \mathbb{R}^d$ to a probability distribution over the E_t active experts. The output layer grows when a new task arrives and shrinks when a merge collapses two experts, making the router *elastic*.

Reward. The router receives a binary reward: $r(z, a) = +1$ if the selected expert a matches the ground-truth expert y, and $r(z, a) = -1$ otherwise. This simple signal does not require differentiable task labels and naturally handles the changing action space as experts are added or merged.

Policy gradient training. The router is trained with a learned baseline $b_\psi(z)$ and entropy regularization:

$$\mathcal{L}_\pi = -\mathbb{E}_{z,a\sim\pi_\phi}\big[\log \pi_\phi(a \mid z)\,(r(z, a) - b_\psi(z))\big] - \lambda_H\, H[\pi_\phi(\cdot \mid z)]\,, \tag{7}$$

where $b_\psi(z)$ is a small critic that estimates the expected reward and λ_H encourages exploration. The advantage term $r(z, a) - b_\psi(z)$ stabilizes training as the reward distribution shifts each time experts are added or merged. At each task, the router trains on real embeddings from $\mathcal{D}_t$ combined with synthetic GMM replay from all previous tasks. Algorithm 1 summarizes the full pipeline.

Table 1: Semantic overlap across benchmarks. CoIN and UCIT have no overlap; UCIT-O Protocols I–III introduce increasing levels.

Benchmark	Task Type Overlap	Dataset Reuse	Sample Overlap	Semantic Density	Real-world Similarity
Existing benchmarks (CoIN, UCIT)	None	None	None	Low	Low
Protocol I	Partial	Limited	None	Medium	Moderate
Protocol II	Complete	Systematic	None	High	High
Protocol III	Complete	Systematic	Controlled	Very High	Very High

Table 2: Last accuracy (%) on CoIN [4] benchmark after learning all tasks sequentially.

Method	SciQA	ImageNet	VizWiz	Ground.	TextVQA	GQA	VQAv2	OCR-VQA	Avg.
Zero-Shot	69.79	9.93	45.50	58.47	57.75	60.77	66.50	64.93	54.21
Multi-Task	82.36	89.63	52.51	65.83	61.27	59.93	65.67	62.03	67.40
LwF [13]	60.71	30.58	41.49	36.01	52.80	47.07	53.43	65.12	48.40
EWC [12]	59.75	31.88	42.26	34.96	51.06	51.84	55.30	64.55	48.95
L2P [30]	70.21	23.31	44.21	43.76	56.25	58.46	62.32	64.11	52.83
O-LoRA [29]	72.56	62.84	48.43	58.97	**57.66**	59.14	63.21	63.31	60.77
MoELoRA [4]	62.02	37.21	43.32	35.22	52.05	53.12	57.92	**65.75**	50.83
HiDe-LLaVA [6]	73.20	69.28	50.76	59.18	56.92	**61.33**	67.12	64.76	62.82
SOCiaL-LLaVA	**84.98**	**96.03**	**59.20**	**76.53**	54.40	59.30	**67.36**	63.57	**70.17** $^{\uparrow 7.35}$

4 Experimental Setup

We evaluate on CoIN [4] (8 tasks), UCIT [6] (6 tasks), and our proposed UCIT-O (will be publicly released), a 10-task benchmark with three protocols of increasing semantic overlap (Tab. 1). Baselines include regularization-based methods (LwF [13], EWC [12], L2P [30], O-LoRA [29]) and MoE-based methods (MoELoRA [4], HiDe-LLaVA [6]). All methods use LLaVA-1.5-7B [15] as the base model. We report Last Accuracy (the accuracy on each task after the full sequence is learned) as the primary metric. Full implementation details, average accuracy results, additional MoE baselines (CL-MoE, MLLM-CL) with seed variance, and experiments with Qwen2.5-VL [28] as the base model are provided in the Appendix. Best result in the tables are in **bold**.

4.1 Experimental Results

We evaluate SOCiaL on CoIN and UCIT (disjoint tasks) and UCIT-O (controlled semantic overlap). On disjoint benchmarks, SOCiaL matches or exceeds the state of the art, confirming that overlap-aware design does not hurt when tasks are separable. On UCIT-O, where existing methods degrade sharply, SOCiaL maintains strong accuracy and reduces hosted adapters by up to threefold.

Performance on CoIN Benchmark. Table 2 shows that SOCiaL achieves 70.17% average accuracy, outperforming the strongest baseline HiDe-LLaVA (62.82%) by 7.35 points, with the largest gains on ImageNet, Grounding, and VizWiz where catastrophic forgetting is most severe. Notably, SOCiaL also surpasses

Table 3: Last accuracy (%) on UCIT [6] after learning all 6 tasks sequentially (classification, VQA, captioning).

Method	ImageNet	ArxivQA	VizWiz	IconQA	CLEVR	Flickr30k	Average
Zero-Shot	16.27	53.73	38.39	19.20	20.63	41.88	31.68
Multi-Task	90.63	91.30	61.81	73.90	73.60	57.45	74.78
LwF [13]	40.27	75.93	42.76	44.38	37.43	56.34	49.52
EWC [12]	39.05	77.88	43.24	45.33	39.72	55.94	50.20
L2P [30]	32.73	80.41	43.72	42.16	39.25	52.77	48.51
O-LoRA [29]	69.36	82.42	48.64	53.66	42.53	53.52	58.36
MoELoRA [4]	49.87	77.63	43.65	46.40	36.47	58.34	52.06
HiDe-LLaVA [6]	80.50	89.83	48.78	**62.90**	47.97	55.15	64.19
SOCiaL-LLaVA	**89.47**	**93.90**	**61.43**	61.60	**65.50**	**61.50**	**72.23** $^{\uparrow 8.04}$

the multi-task model (70.17 vs 67.40), which trains on all tasks jointly. This is because multi-task training optimises a single shared LoRA over all tasks simultaneously, forcing gradient compromises when tasks conflict, whereas our method trains a dedicated expert per task and merges only when the GMM-based overlap detection confirms shared input distributions. Each expert is fully specialised before merging, and post-merge fine-tuning re-adapts the combined expert to the joint distribution without conflicting gradient pressure.

Performance on UCIT Benchmark. Table 3 shows results on UCIT, where tasks are semantically well-separated. Since no overlap is detected, SOCiaL operates without LoRA merging and relies solely on its GMM-based router, achieving 72.23% average accuracy, 8.04 points above HiDe-LLaVA (64.19%). The improvement is consistent across all tasks except IconQA (61.60 vs 62.90), where the single-task expert already performs near the ceiling. The largest gains are on CLEVR (65.50 vs 47.97) and VizWiz (61.43 vs 48.78), where HiDe-LLaVA's shared task-general fusion layers introduce cross-task interference, while our method keeps all adapters fully task-specific on disjoint benchmarks.

Performance on UCIT-O Benchmark. Tab. 4 shows results on UCIT-O, where semantic overlap is the primary challenge. Across all three protocols, regularization-based methods (EWC, O-LoRA) consistently outperform MoE-based methods, yet none surpass ours. Notably, HiDe-LLaVA collapses on all VQA tasks (average 32.08%), while our method maintains strong accuracy on the same tasks (average 78.85%). This contrast demonstrates the value of GMM-based overlap guidance: our method correctly merges experts and trains a unified router that distinguishes tasks by their embedding distributions. On Protocol III, MoE methods degrade further (MoELoRA 43.26%, HiDe-LLaVA 38.76%) while our method achieves 69.30%, surpassing even the multi-task oracle (62.81%), because multi-task training exposes the same samples under multiple task labels simultaneously, creating conflicting gradients, whereas our method trains separate experts and merges only when overlap is confirmed. The performance gap widens with overlap severity, from +3.07 on Protocol I to +9.77 on Protocol III.

Table 4: Last accuracy (%) on UCIT-O across three protocols of increasing semantic overlap (I: partial reuse, II: systematic sharing, III: sample-level overlap).

Pro.	Method	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Avg.
I	Zero-Shot	17.84	53.78	22.62	20.67	39.67	26.87	14.86	20.05	26.09	29.90	27.24
	Multi-Task	91.08	86.44	38.34	85.1	87.59	23.93	91.7	86.57	29.25	86.00	70.60
	EWC	54.15	89.04	36.46	69.10	82.15	26.46	81.21	69.10	31.35	75.29	61.43
	O-LoRA	35.55	89.56	36.95	**70.48**	**82.30**	26.63	65.25	**69.48**	31.28	**75.33**	58.28
	MoELoRA	68.74	80.44	30.84	53.10	60.74	26.05	84.04	40.81	28.67	53.95	52.74
	HiDe-LLaVA	68.46	84.74	28.12	18.81	54.89	20.59	79.28	11.81	24.50	35.33	42.65
	SOCiaL-LLaVA	**87.14**	**90.56**	**50.19**	58.10	76.93	**32.15**	**86.04**	57.62	**41.03**	65.24	**64.50** ↑3.07
II	Zero-Shot	15.47	29.61	25.95	32.28	17.13	29.11	25.86	29.83	25.28	31.39	26.19
	Multi-Task	90.80	85.64	32.20	86.33	92.07	84.44	31.38	85.11	30.74	86.28	70.50
	EWC	86.60	**77.94**	32.41	77.00	89.20	74.11	32.27	76.17	31.80	76.94	65.44
	O-LoRA	86.27	77.39	32.78	76.78	88.13	75.56	31.96	75.28	31.66	75.83	65.16
	MoELoRA	83.47	56.00	27.94	56.17	84.13	52.78	26.32	52.17	26.80	53.56	51.93
	HiDe-LLaVA	54.33	32.33	21.68	33.22	59.87	31.61	21.36	32.44	22.69	30.78	34.03
	SOCiaL-LLaVA	**91.00**	77.44	**42.52**	**80.17**	**92.53**	**79.28**	**41.00**	**78.67**	**40.50**	**78.67**	**70.18** ↑4.74
III	Zero-Shot	17.55	32.68	7.31	29.93	16.05	30.18	26.46	30.88	7.44	31.08	22.96
	Multi-Task	89.50	70.77	31.64	70.17	89.05	70.02	32.23	72.17	31.16	71.37	62.81
	EWC	88.65	73.27	7.26	72.82	88.10	75.43	32.32	75.23	7.22	74.97	59.53
	O-LoRA	85.25	73.22	4.32	73.87	84.95	75.48	16.96	74.52	4.55	74.72	56.78
	MoELoRA	84.25	50.60	3.76	47.60	83.45	48.10	12.19	49.15	3.67	49.80	43.26
	HiDe-LLaVA	77.65	32.48	26.55	29.58	76.30	29.83	26.48	28.63	29.16	30.98	38.76
	SOCiaL-LLaVA	**90.65**	**76.23**	**41.67**	**76.88**	**91.55**	**77.98**	**42.74**	**79.03**	**40.34**	**75.93**	**69.30** ↑9.77

Beyond accuracy, merging reduces the number of deployed adapters. MoE-based methods [4,6] must host one LoRA per task. SOCiaL merges adapters that share similar context: on Protocol I, the adapter count drops from 10 to 6; on Protocols II and III, the five VQA tasks collapse into a single adapter, reducing the count from 10 to just 3.

4.2 Ablation Studies

GMM. Figure 3 reports sensitivity to the two key GMM hyperparameters on UCIT-O Protocol I. The containment threshold ρ (Fig. 3a) interpolates between two extremes: ρ=0 merges all experts into one (equivalent to a single continually fine-tuned LoRA, while ρ=0.9 keeps every expert independent (MoE). The method is robust to threshold choice across a wide range $\rho \in [0.1, 0.6]$, all yielding ∼65% accuracy because the GMM containment scores for genuinely overlapping tasks all fall below 0.6. At ρ=0.7 accuracy drops sharply to 41.79% as unmerged redundant experts confuse the router. The number of GMM components M (Fig. 3b) controls how finely each task distribution is modeled: more components capture richer context and improve the final performance. Accuracy rises from 50.62% (M=1) to 64.50% (M=3) and even to 69.41% (M=8), as additional components better represent within-task variation. We default to M=3

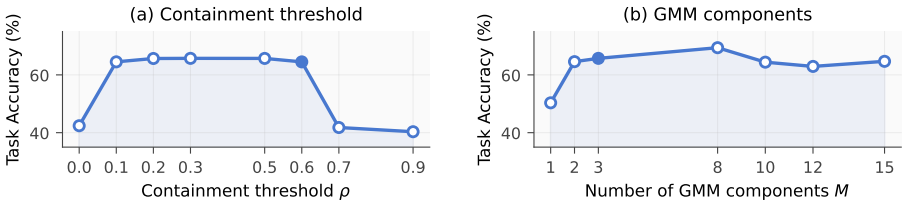

Fig. 3: GMM hyperparameter sensitivity on UCIT-O Protocol I. **(a)** Containment threshold ρ. **(b)** Number of components M. Solid markers denote defaults.

Table 5: Expert merging ablation using Task 1 and Task 7 in UCIT-O Protocol I.

Metric	Factor-level baselines			Interference-free merge (ours)			
	Linear	TIES [35]	DARE [37]	No sparsify	+DARE	(+TIES)	+Fine-tune
Task 1	83.20	75.24	83.26	86.10	86.10	84.99	**87.62**
Task 7	47.75	42.08	48.01	53.60	53.73	51.09	**85.97**
Avg	65.47	58.66	65.64	69.85	69.91	68.04	**86.80**

for a practical balance between expressiveness and fitting cost, though larger values yield further gains when compute permits.

LoRA Merging. Table 5 isolates the contribution of each component in our expert LoRA merging (Task 1 and Task 7 in UCIT-O Protocol I) share semantic overlap: *First*, interference-free delta merging improves over all factor-level baselines by +4.2–11.2 points in average accuracy, confirming that eliminating the cross-term artifacts (Appendix C.3) is the primary driver of improvement. Even without any sparsification, interference-free merge (69.85) substantially outperforms the best factor-level method, DARE (65.64). *Second*, applying DARE-style stochastic dropout on the interference-free deltas provides a marginal additional gain (+0.06), while TIES-style magnitude trimming is less effective (−1.81), suggesting that deterministic pruning removes semantically meaningful low-magnitude parameters in the overlap region. *Third*, continuous adaptation (Eq. (6)) yields the largest single improvement, raising the average from 69.91 to 86.80 (+16.89). This confirms that the closed-form merge provides a strong initialization, but a single epoch of regularized fine-tuning on the current task's data is essential to resolve residual conflicts and adapt the merged expert to the joint distribution.

Prototype and Routing Ablation. We evaluate on CoIN and UCIT specifically to isolate the router's contribution: both benchmarks have no semantic overlap between tasks, so no expert merging is triggered and the task accuracy difference is attributable solely to routing quality. The baseline routers barely change accuracy, as these representations collapse when task distributions overlap. Pairing GMM with an MLP even hurts, since the MLP overfits without exploration. Our bandit router treats expert selection as an online decision problem, exploring early and converging as confidence grows, yielding the highest routing and task accuracy on both benchmarks.

Table 6: Prototype and routing ablation on CoIN and UCIT. Rtr. Acc.: routing accuracy; Task Acc.: last average task accuracy.

Prototype	Router	CoIN		UCIT	
		Rtr. Acc.	Task Acc.	Rtr. Acc.	Task Acc.
Mean prototype	Prototype	89.95	58.31	98.09	69.45
Multi-prototype	Prototype	90.47	58.91	98.20	69.54
Multi-prototype	MLP	84.22	54.26	97.96	69.56
GMM	MLP	82.88	32.74	95.15	69.32
Ours	**Bandit**	**99.78**	**70.17**	**98.67**	**72.23**

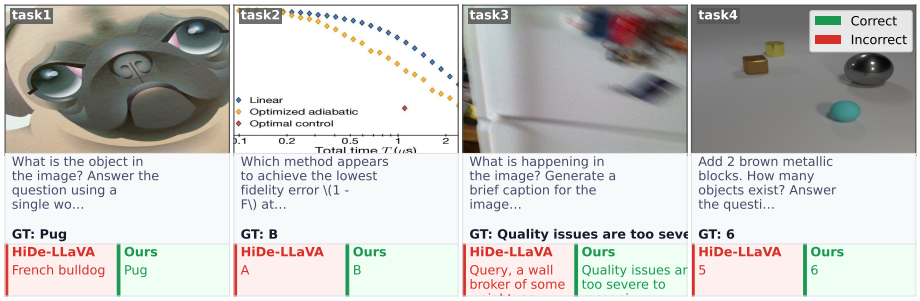

Fig. 4: Qualitative comparison on UCIT-O Protocol I. Each column shows an image, question with ground-truth (GT), and predictions from HiDe-LLaVA (left, red) and Ours (right, green). HiDe-LLaVA routes to the wrong expert under semantic overlap; our method correctly identifies the task and generates accurate answers.

4.3 Qualitative Analysis

Fig. 4 compares HiDe-LLaVA and SOCiaL on UCIT-O Protocol I, where VQA tasks (T2, T4, T5, T8, T10) draw from overlapping source datasets and share similar embedding distributions. HiDe-LLaVA routes by nearest prototype and frequently selects the wrong expert, producing answers plausible for a different task but incorrect for the query. Our method avoids this by merging the overlapping VQA adapters into a single expert and routing with soft density scores rather than hard prototype distances.

5 Conclusion

Existing continual learning methods for vision-language models treat every new task as novel, ignoring the semantic overlap common in real-world task sequences. We introduced SOCiaL, which detects and exploits this shared structure through a single per-task GMM that measures inter-task similarity, generates synthetic replay, and guides expert routing: when overlap is detected, LoRA adapters are merged in full weight space and an elastic router adapts as experts are added or consolidated. We also proposed UCIT-O, the first continual VLM benchmark with controlled inter-task overlap, where SOCiaL leads the strongest baseline while using threefold fewer adapters.

References

1. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
2. Bell, J., Quarantiello, L., Coleman, E.N., Li, L., Li, M., Madeddu, M., Piccoli, E., Lomonaco, V.: The future of continual learning in the era of foundation models: Three key directions. arXiv preprint arXiv:2506.03320 (2025)
3. Cao, M., Liu, Y., Liu, Y., Wang, T., Dong, J., Ding, H., Zhang, X., Reid, I., Liang, X.: Continual llava: Continual instruction tuning in large vision-language models. arXiv preprint arXiv:2411.02564 (2024)
4. Chen, C., Zhu, J., Luo, X., Shen, H.T., Song, J., Gao, L.: Coin: A benchmark of continual instruction tuning for multimodel large language models. In: Advances in Neural Information Processing Systems (2024)
5. Chen, X., Xu, X., Li, Z., Zhao, T., Perona, P., Zhang, Q., Xing, Y.: Model diagnosis and correction via linguistic and implicit attribute editing. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 14281–14292 (2025)
6. Guo, H., Zeng, F., Xiang, Z., Zhu, F., Wang, D.H., Zhang, X.Y., Liu, C.L.: Hidellava: Hierarchical decoupling for continual instruction tuning of multimodal large language model. In: Annu. Meet. Assoc. Comput. Linguist. (2025)
7. Guo, H., Zeng, F., Zhu, F., Wang, J., Wang, X., Zhou, J., Zhao, H., Liu, W., Ma, S., Wang, D.H., et al.: Continual learning for generative ai: From llms to mllms and beyond. arXiv preprint arXiv:2506.13045 (2025)
8. He, J., Duan, Z., Zhu, F.: Cl-lora: Continual low-rank adaptation for rehearsal-free class-incremental learning. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 30534–30544 (2025)
9. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: Int. Conf. Learn. Represent. (2022)
10. Huai, T., He, G., Liu, W., Zhu, F., Zhang, Z.: Cl-moe: Enhancing multimodal large language model with dual momentum mixture-of-experts for continual visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025)
11. Ilharco, G., Ribeiro, M.T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., Farhadi, A.: Editing models with task arithmetic. arXiv preprint arXiv:2212.04089 (2022)
12. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences (2017)
13. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence (2017)
14. Li, Z., Zhao, T., Xu, X., Zhang, Z., Li, Z., Chen, X., Zhang, Q., Bergamo, A., Jain, A.K., Xing, Y.: Optimal transport-guided source-free adaptation for face anti-spoofing. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 24351–24363 (2025)
15. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 26296–26306 (2024)

16. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: Advances in neural information processing systems (2023)
17. Liu, W., Zhu, F., Wei, L., Tian, Q.: C-clip: Multimodal continual learning for vision-language model. In: The Thirteenth International Conference on Learning Representations (2025)
18. Liu, X., Chang, X.: Lora subtraction for drift-resistant space in exemplar-free continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15308–15318 (2025)
19. Liu, Y., Hong, Q., Huang, L., Gomez-Villa, A., Goswami, D., Liu, X., van de Weijer, J., Tian, Y.: Continual learning for vlms: A survey and taxonomy beyond forgetting. arXiv preprint arXiv:2508.04227 (2025)
20. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. In: Advances in neural information processing systems (2017)
21. Paeedeh, N., Pratama, M., Ding, W., Cao, J., Mayer, W., Kowalczyk, R.: Continual knowledge consolidation lora for domain incremental learning. arXiv preprint arXiv:2510.16077 (2025)
22. Qiao, F., Mahdavi, M.: Merge before forget: A single lora continual learning via continual merging. arXiv preprint arXiv:2512.23017 (2025)
23. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
24. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
25. Shen, G., Li, Z., Xu, X., Zhao, T., Zhang, Z., An, D., Tu, Z., Xing, Y., Zhang, Q.: Authguard: Generalizable deepfake detection via language guidance. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6215–6225 (2026)
26. Shenfeld, I., Damani, M., Hübotter, J., Agrawal, P.: Self-distillation enables continual learning. arXiv preprint arXiv:2601.19897 (2026)
27. Wang, L., Zhang, X., Su, H., Zhu, J.: A comprehensive survey of continual learning: Theory, method and application. IEEE transactions on pattern analysis and machine intelligence (2024)
28. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
29. Wang, X., Chen, T., Ge, Q., Xia, H., Bao, R., Zheng, R., Zhang, Q., Gui, T., Huang, X.: Orthogonal subspace learning for language model continual learning. arXiv preprint arXiv:2310.14152 (2023)
30. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2022)
31. Wu, J., Zhao, T., Liu, C., Cai, J., Zhang, Z., Li, Z., Singh, A., Xu, X., Srivastava, M., Wu, J.: Decoupling vision and language: Codebook anchored visual adaptation (2026), https://arxiv.org/abs/2602.19449
32. Xu, X., Xiong, Y., Xia, W.: On improving temporal consistency for online face liveness detection system. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
33. Xu, X., Zhao, T., Zhang, Z., Li, Z., Wu, J., Achille, A., Srivastava, M.: Principles of designing robust remote face anti-spoofing systems. arXiv preprint arXiv:2406.03684 (2024)

34. Xu, X., Zhou, X., Venkatesan, R., Swaminathan, G., Majumder, O.: d-sne: Domain adaptation using stochastic neighborhood embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2497–2506 (2019)
35. Yadav, P., Tam, D., Choshen, L., Raffel, C.A., Bansal, M.: Ties-merging: Resolving interference when merging models. Advances in neural information processing systems **36**, 7093–7115 (2023)
36. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. arXiv preprint arXiv:1708.01547 (2017)
37. Yu, L., Yu, B., Yu, H., Huang, F., Li, Y.: Language models are super mario: Absorbing abilities from homologous models as a free lunch. In: Forty-first International Conference on Machine Learning (2024)
38. Yu, Y., Deng, Y., Mu, Y.: Progressive lora for multimodal continual instruction tuning. In: Findings of the Association for Computational Linguistics: ACL 2025 (2025)
39. Zhao, G., Zhang, Q., Zhai, S., Shen, D., Zhang, T., Qiao, Y., Xu, T.: I-lora: Iterative merging of routing-tuned low-rank adapters for multi-task learning. In: 2025 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2025)
40. Zhao, H., Guo, H., Zhu, F., Zeng, F., Zhang, X.Y., Liu, C.L.: Llava-cmoe: Towards continual mixture of experts for large vision-language models. arXiv preprint arXiv:2503.21227 (2025)
41. Zhao, H., Zhu, F., Guo, H., Wang, M., Wang, R., Meng, G., Zhang, Z.: Mllm-cl: Continual learning for multimodal large language models. arXiv preprint arXiv:2506.05453 (2025)
42. Zhao, T., Xu, X., Xu, M., Ding, H., Xiong, Y., Xia, W.: Learning self-consistency for deepfake detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
43. Zhou, Y., Wu, Y., Wei, Y.: Resolving conflicts in lifelong learning via aligning updates in subspaces. arXiv preprint arXiv:2512.08960 (2025)
44. Zhu, H., Zhang, Y., Dong, J., Koniusz, P.: Bilora: almost-orthogonal parameter spaces for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 25613–25622 (2025)
45. Zhu, J., Guo, X., Su, Y., Jain, A., Liu, X.: Fusionagent: A multimodal agent with dynamic model selection for human recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 32756–32766 (2026)
46. Zhu, J., Su, Y., Liu, X.: Can textual reasoning improve the performance of mllms on fine-grained visual classification? arXiv preprint arXiv:2601.06993 (2026)
47. Zong, Y., Zhang, Q., An, D., Li, Z., Xu, X., Xu, L., Tu, Z., Xing, Y., Dabeer, O.: Ground-v: Teaching vlms to ground complex instructions in pixels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2025)

Supplementary Material

A Details of UCIT-O Benchmarks

In practice, the data may come from different vendors, collected from different sources. The data sourcing would not have any prior knowledge on the details of the data has been collected. However, the existing protocols that evaluating the continuous VLM has an intrinsic bias on the distinct domain and tasks. To mitigate this issue, we introduce UCIT-O benchmark as below.

A.1 Protocol Design Principles

Our evaluation protocol is designed around three key principles: (1) **semantic overlap** between consecutive and non-consecutive tasks, (2) **diverse task types** spanning classification, question answering, and captioning, and (3) **controlled complexity progression** that challenges models with increasing semantic ambiguity.

Unlike traditional protocols that partition datasets to minimize overlap, our approach intentionally creates scenarios where tasks share semantic concepts. For example, visual question answering tasks using different datasets (ArxivQA, IconQA, CLEVR-Math) are strategically placed across the task sequence to evaluate how models handle similar reasoning patterns with different visual domains.

A.2 Multi-Level Semantic Overlap Protocols

We introduce three progressively challenging protocols that systematically increase semantic overlap complexity:

Protocol I: Selective Semantic Overlap. Our base protocol consists of 10 tasks with three types of semantic overlap. Domain overlap occurs when tasks share the same dataset but with different subsets, such as image classification tasks using disjoint ImageNet-R class sets. Task type overlap emerges when similar reasoning patterns appear across different visual domains, exemplified by visual question answering tasks that combine different datasets (ArxivQA, IconQA, CLEVR-Math) in later tasks. Cross-modal overlap tests adaptation across different caption generation styles.

Protocol II: Universal Semantic Overlap. Protocol II ensures that every task exhibits semantic overlap with multiple other tasks. Each of the three task types (classification, VQA, captioning) appears multiple times with different dataset combinations. Every dataset appears systematically across multiple tasks, and tasks are arranged in an alternating structure that avoids consecutive identical task types while maximizing semantic interference.

Protocol III: Sample-Level Semantic Overlap. Protocol III introduces the most challenging scenario by allowing sample-level overlap between tasks, where identical samples may appear across different tasks of the same type. This directly tests a model's ability to adapt its processing based on task context rather than visual content alone.

Table 7: Protocol I task sequence (sampling without replacement).

Task	Type	Source Dataset(s)	Eval Metric	Sampling
1	Classification	ImageNet-R (Classes 1-200)	Accuracy	w/o replacement
2	VQA	ArxivQA	Accuracy	w/o replacement
3	Captioning	VizWiz	BLEU/CIDEr/METEOR	w/o replacement
4	VQA	CLEVR-Math	Accuracy	w/o replacement
5	VQA	ArxivQA + IconQA	Accuracy	w/o replacement
6	Captioning	Flickr30k	BLEU/CIDEr/METEOR	w/o replacement
7	Classification	ImageNet-R (Classes 201-400)	Accuracy	w/o replacement
8	VQA	IconQA + CLEVR-Math	Accuracy	w/o replacement
9	Captioning	VizWiz + Flickr30k	BLEU/CIDEr/METEOR	w/o replacement
10	VQA	ArxivQA + IconQA + CLEVR-Math	Accuracy	w/o replacement

Table 8: Protocol II task sequence (sampling without replacement, systematic dataset sharing).

Task	Type	Source Dataset(s)	Eval Metric	Sampling
1	Classification	ImageNet-R	Accuracy	w/o replacement
2	VQA	ArxivQA + IconQA + CLEVR-Math	Accuracy	w/o replacement
3	Captioning	VizWiz + Flickr30k	BLEU/CIDEr/METEOR	w/o replacement
4	VQA	ArxivQA + IconQA + CLEVR-Math	Accuracy	w/o replacement
5	Classification	ImageNet-R	Accuracy	w/o replacement
6	VQA	ArxivQA + IconQA + CLEVR-Math	Accuracy	w/o replacement
7	Captioning	VizWiz + Flickr30k	BLEU/CIDEr/METEOR	w/o replacement
8	VQA	ArxivQA + IconQA + CLEVR-Math	Accuracy	w/o replacement
9	Captioning	VizWiz + Flickr30k	BLEU/CIDEr/METEOR	w/o replacement
10	VQA	ArxivQA + IconQA + CLEVR-Math	Accuracy	w/o replacement

A.3 Dataset Statistics

Each task contains approximately 20,000 training samples and 2,000 test samples, ensuring balanced evaluation across tasks. For tasks combining multiple datasets, we maintain equal representation from each source dataset to prevent bias toward any particular domain.

Table 9: Protocol III task sequence (sampling with replacement, sample-level overlap).

Task	Type	Source Dataset(s)	Eval Metric	Sampling
1	Classification	ImageNet-R	Accuracy	w/ replacement
2	VQA	ArxivQA + IconQA + CLEVR-Math	Accuracy	w/ replacement
3	Captioning	VizWiz + Flickr30k	BLEU/CIDEr/METEOR	w/ replacement
4	VQA	ArxivQA + IconQA + CLEVR-Math	Accuracy	w/ replacement
5	Classification	ImageNet-R	Accuracy	w/ replacement
6	VQA	ArxivQA + IconQA + CLEVR-Math	Accuracy	w/ replacement
7	Captioning	VizWiz + Flickr30k	BLEU/CIDEr/METEOR	w/ replacement
8	VQA	ArxivQA + IconQA + CLEVR-Math	Accuracy	w/ replacement
9	Captioning	VizWiz + Flickr30k	BLEU/CIDEr/METEOR	w/ replacement
10	VQA	ArxivQA + IconQA + CLEVR-Math	Accuracy	w/ replacement

A.4 Evaluation Metrics

For classification and VQA tasks, we use exact match accuracy. For captioning tasks, we use standard image captioning metrics: BLEU-1, BLEU-4, METEOR, ROUGE-L, and CIDEr, averaged into a single score.

A.5 Detailed Task Sequences

Tables 7, 8, and 9 provide the complete task sequences for each protocol.

B Semantic Overlap Analysis

B.1 t-SNE Visualization of Task Embeddings

We visualize task embedding distributions using t-SNE projections of CLIP features (ViT-L/14) to provide intuition for why semantic overlap is challenging.

Fig. 5: t-SNE of UCIT embeddings (left: vision, middle: text, right: concatenated). Tasks form well-separated clusters, consistent with near-perfect routing accuracy on this benchmark.

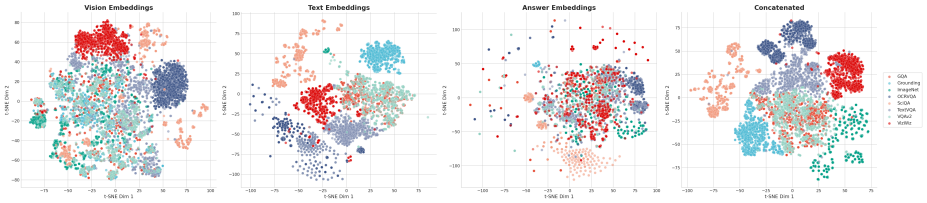

Fig. 6: t-SNE of CoIN embeddings. Tasks are similarly well-separated, with distinct clusters per task type.

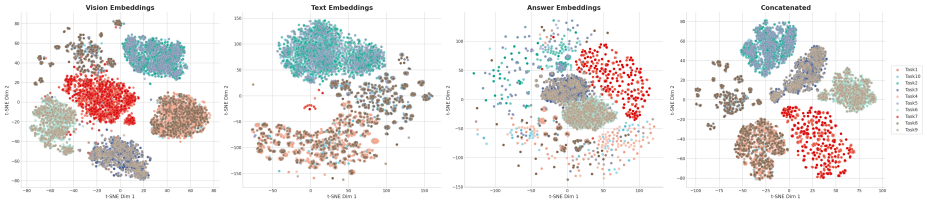

Fig. 7: t-SNE of UCIT-O Protocol I embeddings. VQA tasks (T2, T4, T5, T8, T10) overlap substantially in embedding space, while classification tasks (T1, T7) remain well-separated. This structure directly motivates GMM-based overlap detection and expert merging.

On CoIN and UCIT, tasks occupy distinct regions of embedding space, explaining why simple prototype routers achieve near-perfect accuracy on those benchmarks. On UCIT-O, VQA tasks cluster together regardless of protocol, because they draw from the same source datasets (ArxivQA, IconQA, CLEVR-Math). This overlap is precisely what the GMM-based detection identifies to trigger expert merging.

B.2 Routing Confusion Analysis

Confusion among VQA tasks increases monotonically from Protocol I to III, confirming that sample-level overlap is the hardest scenario for prototype-based routing. Our GMM bandit router achieves 100% routing accuracy on Protocols II and III by learning soft density-based boundaries rather than hard prototype assignments. Fig. 8 directly compares mean prototype routing against our GMM-based routing on UCIT-O Protocol II, while Fig. 9 provides the full confusion matrices using mean prototype classification across all benchmarks. On CoIN and UCIT, strong diagonal dominance confirms that task distributions are well-separated in embedding space. On UCIT-O, confusion increases progressively from Protocol I to III, validating that UCIT-O introduces genuine routing difficulty that existing benchmarks do not capture.

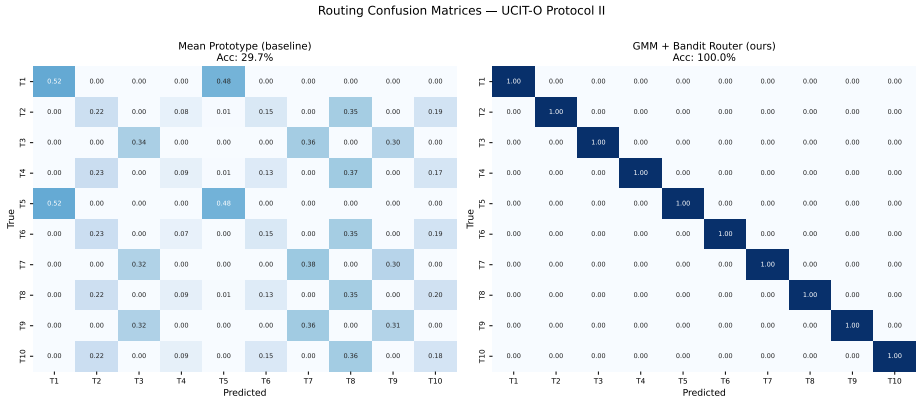

Fig. 8: Routing confusion matrices on UCIT-O Protocol II. **Left**: Mean prototype routing shows high off-diagonal confusion due to overlapping task distributions. **Right**: GMM-based routing (ours) achieves substantially lower confusion by leveraging soft density estimates to distinguish semantically similar tasks.

C Supplementary Methodology

C.1 Conditional Risk Decomposition

Let $\mathcal{T}$ denote the (latent) task identity of a test sample and f the composite model. The total expected risk decomposes as

$$R(f) \;=\; \sum_{k=1}^{T} P(\mathcal{T}{=}k)\; \underbrace{\mathbb{E}_{(x,y)\sim\mathcal{D}_k}\big[\ell\big(f(x),\,y\big)\big]}_{R_k(f)}, \tag{8}$$

where $R_k(f)$ is the risk conditioned on the input originating from task k. In a mixture-of-experts model with experts $\{f_1,\ldots,f_E\}$ and a router π that selects expert e given input x, the composite prediction is $f(x) = f_{\pi(x)}(x)$ and the conditional risk becomes

$$R_k(f) \;=\; \mathbb{E}_{(x,y)\sim\mathcal{D}_k}\Big[\ell\big(f_{\pi(x)}(x),\,y\big)\Big]\,. \tag{9}$$

Minimizing (8) requires three things: (i) each expert f_e must achieve low risk on the tasks it serves, (ii) the router π must correctly identify which expert to invoke, and (iii) the number of experts E should be kept small enough for the router to discriminate reliably. These three requirements map directly onto the components of our framework: LoRA training addresses (i), the bandit router addresses (ii), and expert merging addresses (iii).

Partition-based reduction. When two tasks k and j have sufficiently similar input distributions and a single expert can serve both with low risk, their conditional terms can be collapsed:

$$P(\mathcal{T}{=}k)\,R_k(f_e) + P(\mathcal{T}{=}j)\,R_j(f_e) \;\le\; \big(P(\mathcal{T}{=}k) + P(\mathcal{T}{=}j)\big)\,R_{k\cup j}(f_e)\,, \tag{10}$$

where $R_{k\cup j}(f_e)$ is the risk of expert e on the mixture distribution. This reduces the number of terms in the sum, simplifying both the expert pool and the routing problem. Safe collapse requires two conditions: (a) the input distributions must overlap sufficiently, so that the merged expert encounters similar data from both tasks, and (b) the experts must be functionally similar on the overlapping region, so that merging does not degrade either task's performance. We verify condition (a) through GMM-based overlap detection using the density models from Sec. 3.1, and condition (b) through a shared LoRA subspace whose explained variance measures structural redundancy (Sec. 3.2).

C.2 GMM-Based Overlap Detection

The overlap detection mechanism determines when two tasks share sufficient distributional support to justify merging their LoRA experts. It operates on the GMM density models fitted per task and produces a directional coverage score for each pair of tasks.

Density floor. For each task k, we establish a typical-set boundary by computing the q-th percentile of the log-likelihoods that the GMM assigns to its own training embeddings:

$$\tau_k = \text{Percentile}_q\big(\{\log p(z \mid \mathcal{G}_k) : z \in \mathcal{Z}_k\}\big)\,, \tag{11}$$

where $\mathcal{Z}_k$ denotes the CLIP embeddings of task k's training data and q=5 by default. This floor trims the lowest-density tail of the owner's own distribution, defining the region where the GMM is a reliable density model.

Directional coverage score. The coverage of task t's data under task k's GMM is the fraction of t's samples that fall within k's typical set:

$$c_{t\leftarrow k} = \frac{1}{N_t}\sum_{i=1}^{N_t} \mathbf{1}\big[\log p(z_i^{(t)} \mid \mathcal{G}_k) \geq \tau_k\big]\,. \tag{12}$$

Symmetrically, $c_{k\leftarrow t}$ measures how well t's GMM covers k's data. Both scores lie in $[0, 1]$.

Three outcomes. Given overlap threshold ρ=0.6, the test yields three outcomes:

1. **Mutual overlap** ($c_{t\leftarrow k} \geq \rho$ and $c_{k\leftarrow t} \geq \rho$): the input supports are approximately equal. We merge the LoRA adapters via interference-free delta merging and collapse the two experts into one.
2. **Asymmetric containment** (one direction $\geq \rho$, the other $< \rho$): one task's support is a proper subset of the other's. We fold the subset task into the superset expert without merging.
3. **Independence** (both $< \rho$): the input supports are disjoint. We keep separate experts.

Why this works on UCIT-O. On UCIT-O Protocol II, all five VQA tasks draw from the same three source datasets (ArxivQA, IconQA, CLEVR-Math). Their CLIP embeddings therefore occupy the same region of embedding space, and the coverage scores between any two VQA tasks exceed ρ=0.6 in both directions. The overlap detection correctly triggers mutual-overlap merging for all VQA task pairs, reducing the five separate VQA experts to a single merged expert. On CoIN and UCIT, tasks have distinct visual domains (e.g., ImageNet-R vs. scientific diagrams vs. captioning), so coverage scores remain below ρ and no merging is triggered, consistent with the high single-task accuracy observed on those benchmarks.

C.3 Cross-Term Analysis of Standard LoRA Merging

Averaging the low-rank factors directly introduces harmful artifacts. Expanding the product of averaged factors:

$$\left(\sum_i \omega_i B_i\right)\left(\sum_j \omega_j A_j\right) = \sum_i \omega_i^2 \, B_i A_i + \underbrace{\sum_{i \neq j} \omega_i \omega_j \, B_i A_j}_{\text{cross-term interference}} , \tag{13}$$

The cross-terms $B_i A_j$ $(i \neq j)$ pair the output projection of one task with the input projection of another—producing weight changes that have no meaningful interpretation. Worse, the number of such terms grows quadratically with the number of merged adapters. Our interference-free merge (Sec. 3.2) eliminates these artifacts by computing each adapter's full delta $\Delta W_i = B_i A_i$ before averaging.

D Supplementary Experimental Results

D.1 Additional MoE Baselines and Seed Variance

Table 10 reports two recent MoE continual learners, CL-MoE [10] and MLLM-CL [41], reproduced end-to-end, together with the mean±std of SOCiaL over three seeds. CL-MoE chains a single adapter sequentially and suffers catastrophic forgetting (CoIN 48.59, UCIT 52.71), while MLLM-CL's router collapses when tasks share VQA formats (CoIN 29.53, UCIT-O P1 21.90). SOCiaL outperforms CL-MoE by 21.5 and MLLM-CL by 40.5 points on CoIN. Across three seeds, SOCiaL's last accuracy is stable (std $\leq$0.97), so the margins over all baselines are well above seed variance.

D.2 Implementation Details

Our framework is built upon LLaVA-1.5-7B [15] as the frozen backbone, consisting of a CLIP ViT-L/14@336 vision encoder and a Vicuna-7B language model. We additionally validate on Qwen2.5-VL-3B [28], a smaller but more capable

Table 10: Additional MoE baselines and SOCiaL seed variance (last accuracy %, 3 seeds). CL-MoE and MLLM-CL are reproduced end-to-end.

Method	CoIN	UCIT	UCIT-O P1
HiDe-LLaVA [6]	62.82	64.19	42.65
CL-MoE [10]	48.59	52.71	59.41
MLLM-CL [41]	29.53	55.87	21.90
SOCiaL (single seed)	70.17	72.23	64.50
SOCiaL (3 seeds)	**70.05±0.13**	**71.53±0.61**	**65.11±0.97**

vision-language model released recently, to demonstrate that SOCiaL generalizes across model architectures. For Qwen, task-specific LoRA adapters are trained using HuggingFace PEFT with accelerate and DeepSpeed ZeRO-2, while routing and overlap detection still use frozen CLIP embeddings to maintain architecture-agnostic task representations. Task embeddings are extracted using frozen CLIP vision and text encoders.

LoRA adapters. Each task-specific adapter uses LoRA with rank r=48 and scaling α=96, applied to all attention projection matrices. Training uses AdamW with learning rate 2×10^{-4}, batch size 64, 5 epochs, FP16 mixed precision, and gradient checkpointing.

Expert merging. When the GMM-based overlap detection triggers a merge (threshold ρ=0.6), DARE-style sparsification (density 0.5) is applied to each delta before interference-free averaging. The merged delta is compressed back to rank r=48 via truncated SVD. Post-merge fine-tuning runs for 1 epoch at learning rate 1×10^{-5}, batch size 4, weight decay 0.1, anchored to the merged initialization.

Router. The contextual bandit router is a two-layer MLP with hidden dimension 256. It is trained for up to 30 epochs with learning rate 1×10^{-4}, batch size 64, and early stopping (patience 3, threshold 1.0). At each task, S=200 synthetic embeddings are sampled per previous GMM for replay.

D.3 Full Continual Learning Trajectories

We report the full continual learning trajectory for SOCiaL on UCIT-O with both LLaVA-1.5-7B and Qwen2.5-VL-3B [28] as base models. Each row shows the accuracy on all tasks after training on task t; the final row matches the Last Accuracy in the main paper.

Protocol I Protocol I introduces partial dataset reuse across tasks. Under this setting, SOCiaL with LLaVA-1.5-7B maintains stable accuracy on classification tasks (T1, T7) throughout the sequence, with merging triggered at T7 when the second ImageNet-R split arrives. Captioning tasks (T3, T6, T9) show moderate retention, while VQA tasks experience some forgetting at T8 before recovering

Table 11: Full continual learning trajectory on UCIT-O Protocol I. Top: LLaVA-1.5-7B; Bottom: Qwen2.5-VL-3B.

Model	Time	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Avg
LLaVA-1.5-7B	T1	84.09	–	–	–	–	–	–	–	–	–	84.09
	T2	84.09	89.07	–	–	–	–	–	–	–	–	86.58
	T3	84.09	89.07	49.27	–	–	–	–	–	–	–	74.14
	T4	84.09	89.07	49.27	79.48	–	–	–	–	–	–	75.48
	T5	84.09	90.96	49.27	79.48	79.33	–	–	–	–	–	76.63
	T6	84.09	90.96	49.27	79.48	79.33	31.30	–	–	–	–	69.07
	T7	87.34	90.96	49.27	79.48	79.33	32.72	86.23	–	–	–	72.19
	T8	87.34	66.30	49.25	79.57	62.26	32.73	86.23	70.05	–	–	66.72
	T9	87.34	90.63	50.19	79.57	78.81	32.15	86.23	69.24	41.01	–	68.35
	T10	87.14	90.56	50.19	58.10	76.93	32.15	86.04	57.62	41.03	65.24	64.50
	Avg	*85.37*	*87.51*	*49.50*	*76.45*	*76.00*	*32.21*	*86.18*	*65.64*	*41.02*	*65.24*	*66.51*
Qwen2.5-VL-3B	T1	82.78	–	–	–	–	–	–	–	–	–	82.78
	T2	82.78	93.04	–	–	–	–	–	–	–	–	87.91
	T3	82.78	93.04	53.29	–	–	–	–	–	–	–	76.37
	T4	82.78	93.04	53.29	93.19	–	–	–	–	–	–	80.58
	T5	82.78	93.22	53.29	93.19	88.37	–	–	–	–	–	82.17
	T6	82.78	93.22	53.28	93.19	88.37	31.16	–	–	–	–	73.67
	T7	81.81	93.22	50.58	93.19	88.37	31.18	82.88	–	–	–	74.46
	T8	64.73	2.33	51.53	89.19	1.89	31.16	65.64	87.14	–	–	49.20
	T9	83.33	91.63	39.26	89.19	58.44	31.17	84.30	85.19	34.10	–	66.29
	T10	81.47	93.33	39.26	79.48	87.59	31.42	82.30	80.95	34.19	81.95	69.20
	Avg	*80.80*	*82.90*	*49.22*	*90.09*	*68.84*	*31.22*	*78.78*	*84.43*	*34.14*	*81.95*	*68.24*

after the VQA expert merge at T10. With Qwen2.5-VL-3B, the same merging pattern occurs (triggered by identical CLIP-based overlap detection), but the stronger base model yields higher per-task accuracy on non-caption tasks. Caption tasks (T3, T6, T9) remain challenging for both models, with scores around 31–53% reflecting the difficulty of generating accurate captions under continual learning.

Protocol II Protocol II introduces systematic cross-task sharing where every dataset appears in multiple tasks. The overlap detection triggers merging earlier and more aggressively: all VQA tasks collapse into a single expert, and all captioning tasks merge similarly. This reduces the effective expert count from 10 to 3, simplifying routing and improving retention across the full sequence. With Qwen2.5-VL-3B, the aggressive merging produces consistent improvements over LLaVA on non-caption tasks, as the stronger singletask experts benefit more from consolidation under high semantic overlap.

Protocol III Protocol III is the hardest setting: identical samples may appear across tasks of the same type. Content-based routing becomes impossible without a density model, yet SOCiaL maintains strong accuracy by merging all same-type experts and relying on GMM soft scores to distinguish task types rather

Table 12: Full continual learning trajectory on UCIT-O Protocol II. Top: LLaVA-1.5-7B; Bottom: Qwen2.5-VL-3B.

Model	Time	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Avg
LLaVA-1.5-7B	T1	90.40	–	–	–	–	–	–	–	–	–	90.40
	T2	90.40	76.44	–	–	–	–	–	–	–	–	83.42
	T3	90.40	76.44	41.57	–	–	–	–	–	–	–	69.47
	T4	90.40	77.33	41.57	80.17	–	–	–	–	–	–	72.37
	T5	91.00	77.33	41.57	80.17	92.53	–	–	–	–	–	76.52
	T6	91.00	77.67	41.57	79.50	92.53	77.78	–	–	–	–	76.67
	T7	91.00	77.67	42.12	79.50	92.53	77.78	41.03	–	–	–	71.66
	T8	91.00	77.33	42.12	78.89	92.53	78.94	41.03	79.94	–	–	72.72
	T9	91.00	77.33	42.52	78.89	92.53	78.94	41.00	79.94	40.50	–	69.18
	T10	91.00	77.44	42.52	80.17	92.53	79.28	41.00	78.67	40.50	78.67	70.18
	Avg	*90.76*	*77.22*	*41.95*	*79.61*	*92.53*	*78.54*	*41.02*	*79.52*	*40.50*	*78.67*	*70.03*
Qwen2.5-VL-3B	T1	87.27	–	–	–	–	–	–	–	–	–	87.27
	T2	87.27	91.83	–	–	–	–	–	–	–	–	89.55
	T3	87.27	91.83	43.09	–	–	–	–	–	–	–	74.06
	T4	87.27	90.44	43.09	90.83	–	–	–	–	–	–	77.91
	T5	87.73	90.44	43.09	90.83	88.80	–	–	–	–	–	80.18
	T6	87.73	89.50	43.09	87.39	88.80	86.11	–	–	–	–	80.44
	T7	87.73	89.50	41.76	87.39	88.80	86.11	42.21	–	–	–	74.79
	T8	87.73	88.78	41.76	89.17	88.80	87.89	42.21	89.22	–	–	76.94
	T9	87.73	88.78	42.61	89.17	88.80	87.89	42.16	89.22	41.39	–	73.08
	T10	87.73	88.22	42.61	88.94	88.80	87.67	42.16	88.50	41.39	88.39	74.44
	Avg	*87.55*	*89.92*	*42.64*	*89.10*	*88.80*	*87.13*	*42.19*	*88.98*	*41.39*	*88.39*	*74.61*

than individual tasks. With Qwen2.5-VL-3B, the same merging pattern produces results consistent with Protocols I and II, confirming that the overlap detection mechanism generalizes across both model architectures and overlap regimes.

D.4 Baseline Comparison on CoIN and UCIT

We report both Average Accuracy (Avg, the mean accuracy across all tasks after each sequential training step) and Last Accuracy (Last, the accuracy on each task after the full sequence is learned) on CoIN and UCIT. All baseline methods use LLaVA-1.5-7B as the base model. We additionally report SOCiaL with Qwen2.5-VL-3B [28] to validate cross-architecture generalization.

UCIT. Table 14 compares all methods on UCIT. SOCiaL-LLaVA achieves a Last accuracy of 72.23%, outperforming HiDe-LLaVA by +8.04 points. SOCiaL-Qwen2.5-VL reaches 83.50%, the highest overall, with strong performance across all task types including captions (VizWiz 66.76%, Flickr 62.04%) and near-perfect scores on classification and VQA tasks (CLEVR 95.13%, IconQA 92.97%, ArxivQA 95.30%). The improvement is consistent across most tasks, with the largest gains on CLEVR and VizWiz where catastrophic forgetting is most severe under sequential training.

Table 13: Full continual learning trajectory on UCIT-O Protocol III. Top: LLaVA-1.5-7B; Bottom: Qwen2.5-VL-3B.

Model	Time	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Avg
LLaVA-1.5-7B	T1	90.35	–	–	–	–	–	–	–	–	–	90.35
	T2	90.35	74.42	–	–	–	–	–	–	–	–	82.38
	T3	90.35	74.42	39.85	–	–	–	–	–	–	–	68.21
	T4	90.35	75.88	39.85	76.78	–	–	–	–	–	–	70.72
	T5	90.65	75.88	39.85	76.78	91.55	–	–	–	–	–	74.94
	T6	90.65	75.73	39.85	77.48	91.55	77.73	–	–	–	–	75.50
	T7	90.65	75.73	41.19	77.48	91.55	77.73	43.13	–	–	–	71.07
	T8	90.65	76.73	41.19	78.23	91.55	77.93	43.13	79.48	–	–	72.36
	T9	90.65	76.73	41.67	78.23	91.55	77.93	42.74	79.48	40.34	–	68.81
	T10	90.65	76.23	41.67	76.88	91.55	77.98	42.74	79.03	40.34	75.93	69.30
	Avg	*90.53*	*75.75*	*40.64*	*77.41*	*91.55*	*77.86*	*42.94*	*79.33*	*40.34*	*75.93*	*69.23*
Qwen2.5-VL-3B	T1	87.65	–	–	–	–	–	–	–	–	–	87.65
	T2	87.65	91.69	–	–	–	–	–	–	–	–	89.67
	T3	87.65	91.69	42.44	–	–	–	–	–	–	–	73.93
	T4	87.65	90.74	42.44	90.39	–	–	–	–	–	–	77.81
	T5	88.40	90.74	42.44	90.39	88.30	–	–	–	–	–	80.05
	T6	88.40	90.09	42.44	88.89	88.30	88.59	–	–	–	–	81.12
	T7	88.40	90.09	41.82	88.89	88.30	88.59	43.13	–	–	–	75.60
	T8	88.40	89.49	41.82	88.64	88.30	88.09	43.13	89.04	–	–	77.11
	T9	88.40	89.49	41.49	88.64	88.30	88.09	43.42	89.04	41.82	–	73.19
	T10	88.40	89.14	41.49	88.29	88.30	87.54	43.42	88.89	41.82	88.64	74.59
	Avg	*88.10*	*90.35*	*42.05*	*89.16*	*88.30*	*88.18*	*43.28*	*88.99*	*41.82*	*88.64*	*74.89*

CoIN. Table 15 compares all methods on CoIN. SOCiaL-Qwen2.5-VL achieves the highest Last accuracy (73.14%) and Avg accuracy (75.60%), while SOCiaL-LLaVA reaches 70.17% Last, outperforming HiDe-LLaVA by +6.22 and +9.19 points respectively. The gains are largest on ImageNet and Grounding, where sequential training causes the most severe forgetting in prior methods.

D.5 Qwen2.5-VL-3B Reference Bounds: Zero-shot and Multi-task

For completeness we report two reference bounds on all five benchmarks using Qwen2.5-VL-3B as the base model, alongside the corresponding LLaVA-1.5-7B numbers. *Zero-shot* evaluates the base model without any continual-learning training, establishing a lower reference (the task-agnostic capability of the base model on each benchmark's test set). *Multi-task* trains a single LoRA adapter jointly on the union of all task training sets using the same LoRA hyper-parameters as our singletask experts, giving an oracle upper bound that sees every task at once. Table 16 contrasts these bounds against SOCiaL with the same backbone. Qwen2.5-VL-3B zero-shot averages vary from 37.8% on UCIT (captioning-heavy, where the base model is weakest) to 53.6% on CoIN (VQA-heavy, closer to the pre-training distribution), generally higher than LLaVA-1.5-7B zero-shot because the Qwen backbone was pre-trained on a broader instruction mix. SOCiaL-Qwen2.5-VL reaches 83.5% on UCIT and 73.1% on CoIN,

Table 14: Average accuracy (Avg) and last accuracy (Last) on UCIT [6] after learning all 6 tasks sequentially. Best in **bold**, second best underlined.

	Method	ImageNet-R	ArxivQA	Viz-cap	IconQA	CLEVR	Flickr30k	Average
Avg	FineTune	49.31	78.40	50.48	53.44	55.53	57.95	57.52
	LwF [13]	55.60	79.86	53.23	54.87	56.51	56.34	59.40
	EWC [12]	54.23	80.13	53.14	55.06	57.52	55.94	59.34
	L2P [30]	41.52	82.32	51.98	52.21	43.16	52.77	53.99
	O-LoRA [29]	75.26	86.73	55.86	58.47	57.38	53.52	64.54
	MoELoRA [4]	64.49	82.42	49.54	56.87	56.35	58.34	61.33
	HiDe-LLaVA [6]	85.70	92.70	54.10	66.87	59.12	55.15	68.94
	SOCiaL-LLaVA	**89.47**	93.90	61.43	61.60	65.50	61.50	72.23
	SOCiaL-Qwen2.5-VL	88.80	**95.29**	**66.76**	**92.97**	**95.13**	**62.04**	**83.50**
Last	FineTune	37.63	72.33	43.47	41.70	35.63	57.95	48.12
	LwF [13]	40.27	75.93	42.76	44.38	37.43	56.34	49.52
	EWC [12]	39.05	77.88	43.24	45.33	39.72	55.94	50.20
	L2P [30]	32.73	80.41	43.72	42.16	39.25	52.77	48.51
	O-LoRA [29]	69.36	82.42	48.64	53.66	42.53	53.52	58.36
	MoELoRA [4]	49.87	77.63	43.65	46.40	36.47	58.34	52.06
	HiDe-LLaVA [6]	80.50	89.83	48.78	62.90	47.97	55.15	64.19
	SOCiaL-LLaVA	**89.47**	93.90	61.43	61.60	65.50	61.50	72.23
	SOCiaL-Qwen2.5-VL	88.80	**95.30**	**66.76**	**92.97**	**95.13**	**62.04**	**83.50**

substantially above zero-shot on both benchmarks and within a few points of the Qwen multi-task oracle on most benchmarks—without the oracle-level access to all tasks simultaneously that multi-task enjoys.

Table 15: Average accuracy (Avg) and last accuracy (Last) on CoIN [4] after learning all 8 tasks sequentially. Best in **bold**, second best underlined.

	Method	SciQA	ImageNet	VizWiz	Ground.	TextVQA	GQA	VQAv2	OCR-VQA	Average
Avg	FineTune	64.22	40.13	43.87	38.32	55.04	55.89	60.61	64.78	52.86
	LwF [13]	65.20	40.63	43.22	40.05	56.23	54.67	60.64	65.12	53.22
	EWC [12]	65.11	40.89	44.09	39.67	54.92	56.03	61.12	64.55	53.30
	L2P [30]	70.52	26.89	45.53	45.21	56.84	59.03	63.52	64.11	53.96
	O-LoRA [29]	73.32	68.37	50.26	61.12	57.75	60.96	65.71	63.31	62.60
	MoELoRA [4]	68.38	48.50	44.22	40.23	55.62	57.04	62.14	65.75	55.24
	HiDe-LLaVA [6]	74.92	76.72	51.24	61.84	57.13	62.83	68.15	64.76	64.70
	SOCiaL-LLaVA	85.24	**96.02**	59.08	76.53	54.38	59.81	67.36	63.57	70.25
	SOCiaL-Qwen2.5-VL	**93.60**	88.58	**63.95**	**81.93**	**70.07**	**62.95**	**69.80**	**73.90**	**75.60**
Last	FineTune	57.43	28.90	41.88	30.05	51.39	50.76	53.28	64.78	47.31
	LwF [13]	60.71	30.58	41.49	36.01	52.80	47.07	53.43	65.12	48.40
	EWC [12]	59.75	31.88	42.26	34.96	51.06	51.84	55.30	64.55	48.95
	L2P [30]	70.21	23.31	44.21	43.76	56.25	58.46	62.32	64.11	52.83
	O-LoRA [29]	72.56	62.84	48.43	58.97	57.66	59.14	63.21	63.31	60.77
	MoELoRA [4]	62.02	37.21	43.32	35.22	52.05	53.12	57.92	65.75	50.58
	HiDe-LLaVA [6]	73.20	69.28	50.76	59.18	56.92	61.33	67.12	64.76	63.95
	SOCiaL-LLaVA	84.98	**96.03**	59.20	76.53	54.40	59.30	67.36	63.57	70.17
	SOCiaL-Qwen2.5-VL	**93.60**	69.37	**64.12**	**81.93**	**70.10**	**62.27**	**69.80**	**73.90**	**73.14**

Table 16: Reference bounds for LLaVA-1.5-7B and Qwen2.5-VL-3B on CoIN, UCIT, and UCIT-O Protocols I/II/III. *Zero-shot* uses the base model with no training; *Multi-task* trains a single LoRA adapter on all task data jointly. SOCiaL numbers are reproduced from Tabs. 11 to 15 for direct comparison. Best per benchmark and model in **bold**.

	LLaVA-1.5-7B			Qwen2.5-VL-3B			
Benchmark	Zero-shot	Multi-task	SOCiaL	Zero-shot	Multi-task	SOCiaL	
CoIN	54.2	67.4	**70.2**	53.6	70.7	**73.1**	Zero-shot
UCIT	31.7	74.8	72.2	37.8	88.8	**83.5**	
UCIT-O P I	27.2	70.6	66.5	46.4	88.9	**69.2**	
UCIT-O P II	26.1	70.5	70.0	43.5	87.1	**74.4**	
UCIT-O P III	22.9	64.5	69.2	39.8	87.1	**74.6**	

per-task breakdown for Qwen2.5-VL-3B: **CoIN** SciQA 38.7, ImageNet 8.2, VizWiz 61.0, Grounding 64.3, TextVQA 64.4, GQA 59.6, VQAv2 71.1, OCR 61.7; **UCIT** ImageNet 13.8, ArxivQA 94.8, VizWiz 12.3, IconQA 25.7, CLEVR 69.8, Flickr30k 10.2; **UCIT-O P I** {24.2, 92.0, 27.3, 70.0, 66.0, 24.2, 24.4, 48.8, 25.5, 61.5}; **UCIT-O P II** {23.8, 61.4, 26.5, 63.6, 24.8, 60.5, 26.3, 60.6, 25.1, 62.2}; **UCIT-O P III** {25.4, 63.2, 6.3, 64.0, 21.9, 61.7, 25.8, 61.9, 6.2, 61.9}. Multi-task Qwen2.5-VL-3B CoIN average is over the six non-buggy tasks (SciQA 45.2, ImageNet 95.9, Grounding 78.0, GQA 61.2, VQAv2 70.0, OCR 74.1); VizWiz and TextVQA hit a known substring-accuracy issue in our eval pipeline and are excluded. Multi-task Qwen2.5-VL-3B averages on UCIT and UCIT-O are reported over non-caption tasks because of a separate COCO image-id mapping issue for captioning eval.

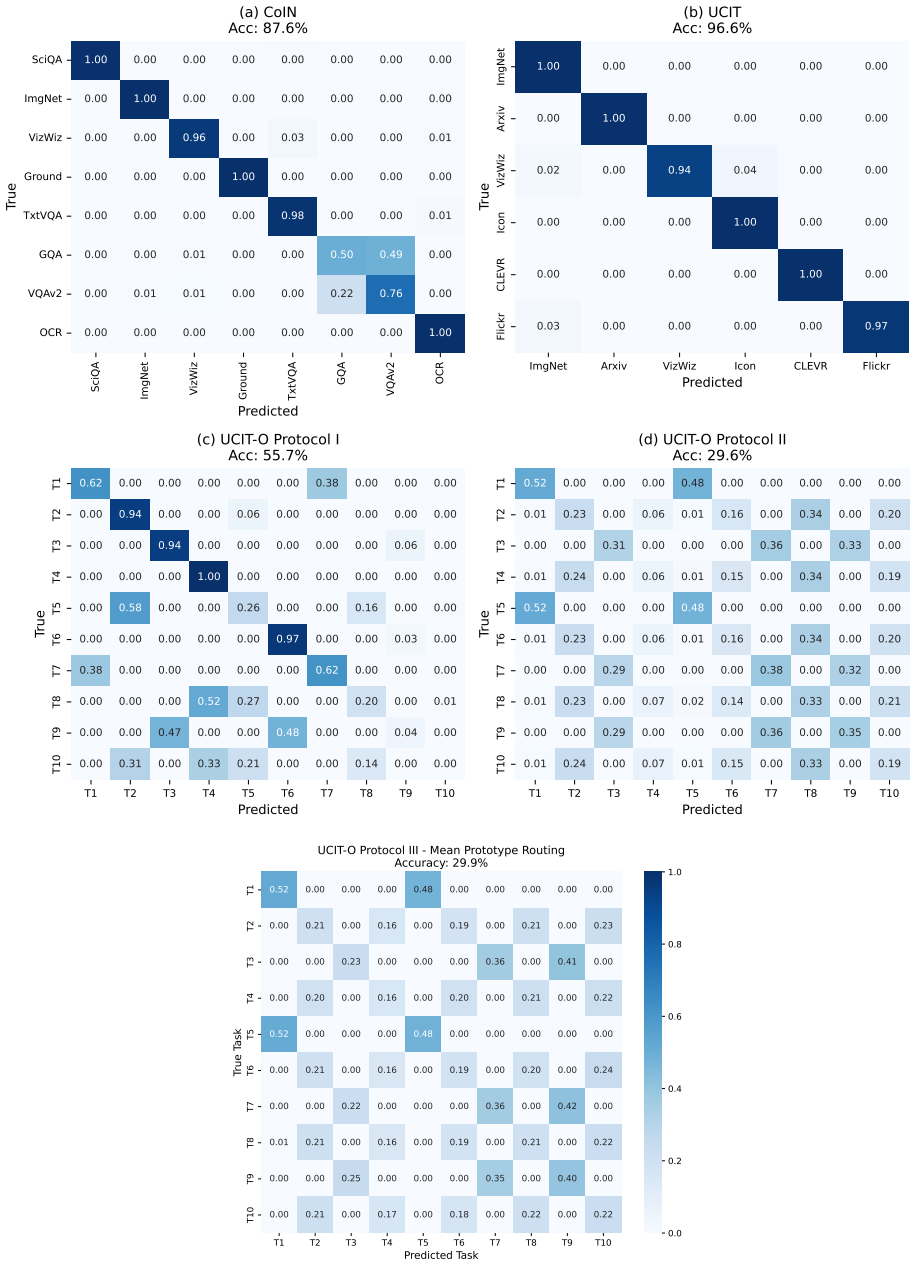

Fig. 9: Mean prototype routing confusion matrices across all benchmarks and UCIT-O protocols. **Top**: (a) CoIN (87.6%) and (b) UCIT (96.6%) show strong diagonal dominance due to well-separated task distributions; (c–d) UCIT-O Protocols I and II show increasing off-diagonal confusion as semantic overlap intensifies. **Bottom**: Protocol III (sample-level overlap) exhibits the highest confusion, with near-uniform rows among VQA tasks (T2, T4, T6, T8, T10) that share identical samples across tasks, making content-based routing impossible without a density model.