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 Jonathan Wu[3]

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

Project page: https://shownx.github.io/ECCV26-SOCIAL/

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, 48] 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, 43, 46, 47]. Continual learning methods, from regularization-based approaches [12,13,20,23,24,36], LoRA based approaches [8, 17, 18, 45], 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

⋆Work done during an internship at Amazon.

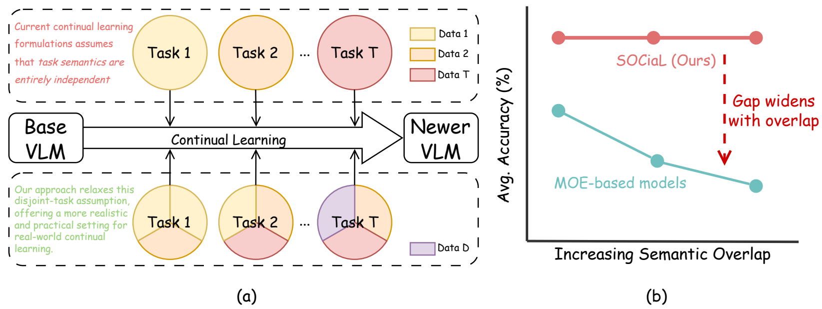

Fig. 1: **(a)** Existing continual learning benchmarks assume that sequential tasks occupy separate domains (top). In practice, tasks often share images, concepts, or language patterns (bottom). **(b)** As inter-task overlap increases, HiDe [6] degrades sharply: its standard adapter merging mixes the 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.

routinely reuse visual concepts, language patterns, or training samples from earlier ones. We call this more realistic setting *natural continual learning*: tasks arrive 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, 45] 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*. [44] 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.

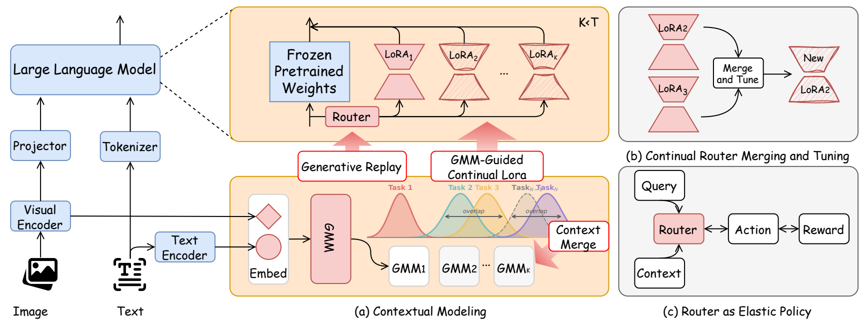

Fig. 2: SOCiaL framework. SOCiaL leverages a per-task GMM (a) to serve triple duty: it models the semantic context of each task, guides merging of LoRAs within the same semantic context, and generates synthetic embeddings for the router's continual updates. When semantic overlap is detected, the corresponding 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.

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 ($B_i A_i$) 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, 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. 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: 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 multi-modal 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 $\mathcal{Z}_k = \{z_i\}_{i=1}^{N_k}$:

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

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.

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. Expanding the product of averaged factors,

$$\big(\textstyle\sum_i \omega_i B_i\big)\big(\textstyle\sum_j \omega_j A_j\big) = \textstyle\sum_i \omega_i^2 \, B_i A_i + \underbrace{\sum_{i \neq j} \omega_i \omega_j \, B_i A_j}_{\text{cross-term interference}} , \tag{5}$$

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 with no meaningful interpretation, and their number grows quadratically with the number of merged adapters. 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:

$$\Delta \bar{W}^{(\ell)} \approx \bar{B}^{(\ell)} \bar{A}^{(\ell)} , \quad \bar{B}^{(\ell)} = U_r \sqrt{\Sigma_r} , \;\; \bar{A}^{(\ell)} = \sqrt{\Sigma_r}\, V_r^{\top} , \tag{6}$$

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:

$$\mathcal{L}_{\text{ft}} = \mathcal{L}_{\text{CE}}(\bar{w};\, \mathcal{D}_t) + \lambda \sum_{\ell} \left\| \bar{w}^{(\ell)} - \bar{w}^{(\ell)}_{\text{init}} \right\|_2^2 , \tag{7}$$

where $\bar{w}_{\text{init}}$ denotes the merged weights from Eq. (6) (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.

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 $a \in \{1, \dots, E_t\}$, 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:

$$\mathcal{L}_\pi = -\mathbb{E}_{z, a \sim \pi_\phi}\big[\log \pi_\phi(a \mid z)\, (r(z, a) - b_\psi(z))\big] - \lambda_H\, H[\pi_\phi(\cdot \mid z)] , \tag{8}$$

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.

4 Experimental Setup

We evaluate on CoIN [4] (8 tasks), UCIT [6] (6 tasks), and our proposed UCIT-O, a 10-task benchmark with three protocols of increasing semantic overlap that we release publicly on the project page (Tab. 1).

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. 6
      Fine-tune merged adapter; unify GMMs ▷ Eqs. 7,4
    end if
  end for
  Train router π_ϕ on $\mathcal{D}_t$ + GMM replay ▷ Eq. 8
end for

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

UCIT-O benchmark. Built on UCIT's six vision-language datasets (ImageNet-R, ArxivQA, VizWiz, CLEVR-Math, IconQA, and Flickr30k), UCIT-O arranges them into 10 sequential tasks—2 classification, 5 visual question answering, and 3 captioning—and deliberately injects three types of semantic overlap: *domain overlap*, where tasks reuse the same dataset with disjoint subsets (*e.g.*, separate ImageNet-R class splits); *task-type overlap*, where the same reasoning pattern recurs across visual domains (VQA tasks that recombine ArxivQA, IconQA, and CLEVR-Math later in the sequence); and *cross-modal overlap* across caption generation styles. Three protocols progressively increase this overlap: *Selective* (Protocol I) shares content in only some tasks; *Universal* (Protocol II) makes every task overlap with multiple others through systematic dataset sharing in an alternating task-type layout; and *Sample-Level* (Protocol III) allows identical samples to appear across tasks of the same type, the most realistic and challenging case as it forces the model to rely on task context rather than visual content alone. For dataset statistics, each task contains roughly 20K training and 2K test samples drawn under a fixed seed, and tasks that combine multiple source datasets sample equally from each to prevent bias toward any single domain. We score classification and VQA by exact-match accuracy and captioning by BLEU/METEOR/ROUGE-L/CIDEr.

Baselines include regularization-based methods (LwF [13], EWC [12], L2P [30], O-LoRA [29]) and MoE-based methods (MoELoRA [4], HiDe-LLaVA [6]). All

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** ↑7.35

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** ↑8.04

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 supplementary material. 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 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** $^{\uparrow 3.07}$
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** $^{\uparrow 4.74}$
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** $^{\uparrow 9.77}$

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-

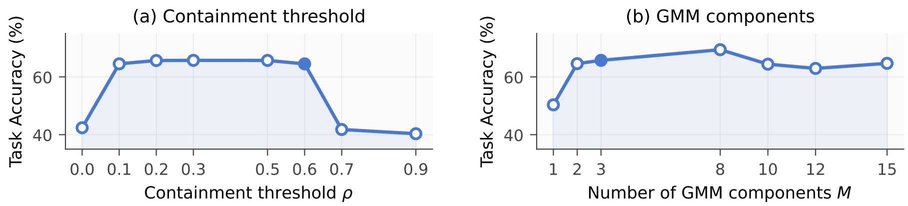

Fig. 3: GMM hyperparameter sensitivity on UCIT-O Protocol I. **(a)** Containment threshold ρ. **(b)** Number of components M. Solid markers denote defaults.

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

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**

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**

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 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. (7)) 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. Table 6 evaluates prototype and router variants 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.

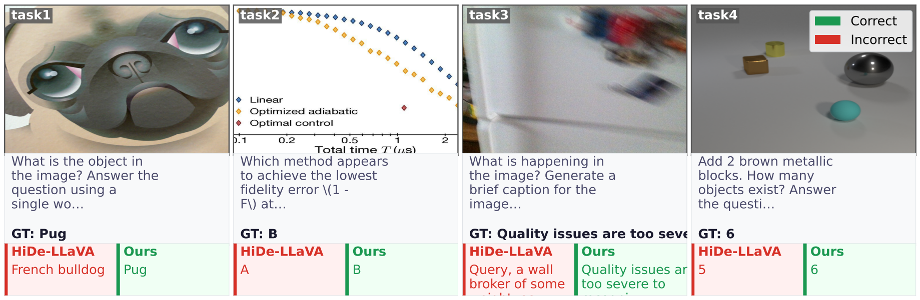

Fig. 4: Qualitative comparison on UCIT-O Protocol I. Each column shows an image, question with ground-truth (GT), and predictions from HiDe-LLaVA (left, red) and Ours (right, green). HiDe-LLaVA routes to the wrong expert under semantic overlap; our method correctly identifies the task and generates accurate 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.: Hide-llava: 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. arXiv preprint arXiv:2602.19449 (2026)
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., Chen, X., Li, Z., Fang, J., An, D., Xu, X., Tu, Z., Xing, Y.: Salient concept-aware generative data augmentation. arXiv preprint arXiv:2510.15194 (2025)
43. 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)
44. Zhou, Y., Wu, Y., Wei, Y.: Resolving conflicts in lifelong learning via aligning updates in subspaces. arXiv preprint arXiv:2512.08960 (2025)
45. 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)
46. 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)
47. 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)
48. 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)