Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Brain-inspired hierarchical modularity for general continual learning

Liyuan Wang

liyuanwang@tsinghua.edu.cn

Tsinghua University https://orcid.org/0000-0002-3869-8155

Hongwei Yan

Tsinghua University

Kanglei Zhou

Tsinghua University

Qi Cheng

Tsinghua University

Weiyi Dong

Tsinghua University

Chunyan Lan

Tsinghua University

Guanglong Sun

Tsinghua University

Jun Zhou

Tsinghua University

Qian Li

Sun Yat-sen University

Yi Zhong

Tsinghua University

Article

Keywords:

Posted Date: November 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11334458/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: There is **NO** Competing Interest.

Brain-inspired hierarchical modularity for general continual learning

Hongwei Yan[1,2,3†], Kanglei Zhou[2,4†], Qi Cheng[2,4], Weiyi Dong[2,4], Chunyan Lan[2,4], Guanglong Sun[1,2,3], Jun Zhou[1,2,3], Qian Li[5], Yi Zhong[1,2,3] and Liyuan Wang[2,4*]

[1]School of Life Sciences, Tsinghua University, Beijing, China.
[2]IDG/McGovern Institute for Brain Research, Tsinghua University, Beijing, China.
[3]Tsinghua-Peking Center for Life Sciences, Beijing, China.
[4]Department of Psychological and Cognitive Sciences, Tsinghua University, Beijing, China.
[5]Zhongshan School of Medicine, Sun Yat-sen University Shenzhen Campus, Shenzhen, China.

*Corresponding author(s). E-mail(s): liyuanwang@tsinghua.edu.cn;
†These authors contributed equally to this work.

Abstract

Continual learning, the ability to learn from experience while retaining and adapting prior knowledge, is central to intelligence operating in changing environments. However, existing efforts in continual learning often assume offline task-wise training and clear task boundaries, leaving a substantial gap towards general continual learning of the external world under online, uncertain, and evolving data streams. In this regime, intelligent systems must selectively separate conflicting experience to reduce interference while integrating compatible experience to promote generalization. Inspired by the organization of the *Drosophila* learning and memory system, we identify a hierarchical modular principle that coordinates both functions through expert specialization and ensemble integration. Controlled biological simulation and formal analysis support this principle. We further instantiate it into a lightweight modular adaptation framework for pretrained foundation models, combining brain-inspired random expansion for expert routing with diversified modular integration across spatial and temporal scales. Across visual recognition, vision-language understanding, ego-exo video

understanding, and embodied vision-language-action learning, our method consistently improves continual learning under online and uncertain data streams, with gains exceeding 50 percentage points over replay-free alternatives in embodied manipulation. These findings establish hierarchical modularity as a biologically grounded computational principle for learning from evolving experience.

Keywords: neuro-inspired learning, continual learning, learning and memory, catastrophic forgetting, adaptability

1 Introduction

Continual learning (CL) [1, 2] is central to intelligence that develops through experience, enabling knowledge to be acquired, accumulated, and adapted over time. In biological organisms [3, 4], learning from sequentially arriving experience supports both immediate responses to environmental changes and progressive development throughout the lifespan. A similar capability is increasingly central to artificial intelligence (AI). Moving beyond intelligence acquired primarily from static human-curated data demands AI systems capable of learning continuously from their own experience, despite the persistent challenges of catastrophic forgetting [1, 5] and loss of plasticity [6, 7]. Recent perspectives on an "era of experience" [8] and autonomous, open-ended intelligence [9, 10] envision increasingly general agents whose capabilities develop through persistent interaction with the external world. Emerging directions in self-improving agents [11] and test-time training [12, 13] similarly point towards systems that continue to refine their behaviour and internal knowledge after deployment.

Most existing efforts in AI systems formulate CL in a simplified regime, typically within a narrow task setting and with largely offline training, clear task boundaries, and auxiliary task identities [14–16]. These assumptions have enabled remarkable progress in addressing catastrophic forgetting through synaptic regularization [6, 17], memory replay [18, 19], and dynamic architecture [15, 20]. However, real-world experience typically arrives online under uncertain and evolving data distributions across diverse models, modalities, and application scenarios. We refer to this broader regime as *general continual learning* (GCL) [2, 21]. Its central challenge extends beyond retaining past knowledge to a more fundamental question: **how should learning be organized as data distributions evolve over time?** Addressing this challenge requires selectively separating conflicting experience to reduce interference while integrating compatible experience to exploit shared structure and promote generalization.

Biological organisms naturally learn under such dynamic conditions, serving as a desirable reference for GCL. Among model organisms, *Drosophila* is particularly tractable because its learning circuits combine rich adaptive behaviour with increasingly detailed anatomical characterization from whole-brain connectomes and cross-connectome cell typing [3, 4, 22–25]. In the olfactory learning and memory system, sparse, largely random projections expand sensory representations in

Kenyon cells and support pattern separation [26–29], while downstream learning and memory are distributed across differentiated compartments with distinct spatial and temporal characteristics [30–34]. Computationally, we relate these biological mechanisms to two classical paradigms of modular machine learning: *mixture-of-experts* (MoE) [35, 36] and *ensemble learning* (EL) [37, 38]. MoE promotes specialization across dissimilar distributions to reduce interference, whereas EL integrates diversified information over related distributions to promote generalization. Importantly, the *Drosophila* system coordinates these MoE- and EL-like functions through a hierarchical learning and memory organization rather than deploying them independently [39, 40]. This hierarchical coordination provides a biological reference for jointly organizing specialization and integration in GCL.

Here we introduce FlyGCL, a unified brain-inspired framework for GCL with pretrained foundation models. In *Drosophila*, learning and memory operate downstream of relatively stable sensory processing. Analogously, FlyGCL retains the pretrained backbone as a stable representational substrate and organizes lightweight, parameter-efficient learning downstream. Brain-inspired random expansion of pretrained representations improves instance-level routing among specialized experts, while differentiated adaptive modules and multi-timescale predictions introduce complementary diversity across spatial and temporal axes for ensemble integration. These components realize hierarchical coordination between routing-based specialization and spatial-temporal integration. Controlled olfactory simulation experiments demonstrate their additive benefits, with hierarchical coordination consistently outperforming either component alone across disjoint, blurry, and joint data streams. The modular design accommodates different lightweight learning modules and is broadly applicable across pretrained backbones and learning settings. Computational analyses further characterize the complementary gains of specialization and integration, and the benefit of organizing them over stable pretrained representations (Methods).

We evaluate FlyGCL across various forms of real-world continual experience, spanning visual recognition, vision-language understanding, ego-exo video understanding, and embodied vision-language-action learning, all under online and uncertain data streams (Supplementary Tabs. S1 and S2). FlyGCL consistently improves CL performance across these scenarios and pretrained models. The gains are particularly pronounced in embodied vision-language-action learning. Across spatial, object-centric, goal-conditioned, and long-horizon manipulation, FlyGCL achieves final average success rates of 83.1%, 86.1%, 94.5%, and 79.1%, respectively, exceeding the strongest replay-free CL baseline on each benchmark by 49.5–58.8 percentage points. These results support hierarchical modularity as a biologically grounded computational principle for organizing learning from evolving experience.

2 Results

We study *general continual learning* (GCL) under online, uncertain, and evolving data streams across diverse models, modalities, and application scenarios. Its central challenge is to organize incoming experience by separating conflicting distributions to reduce interference while integrating compatible ones to promote generalization

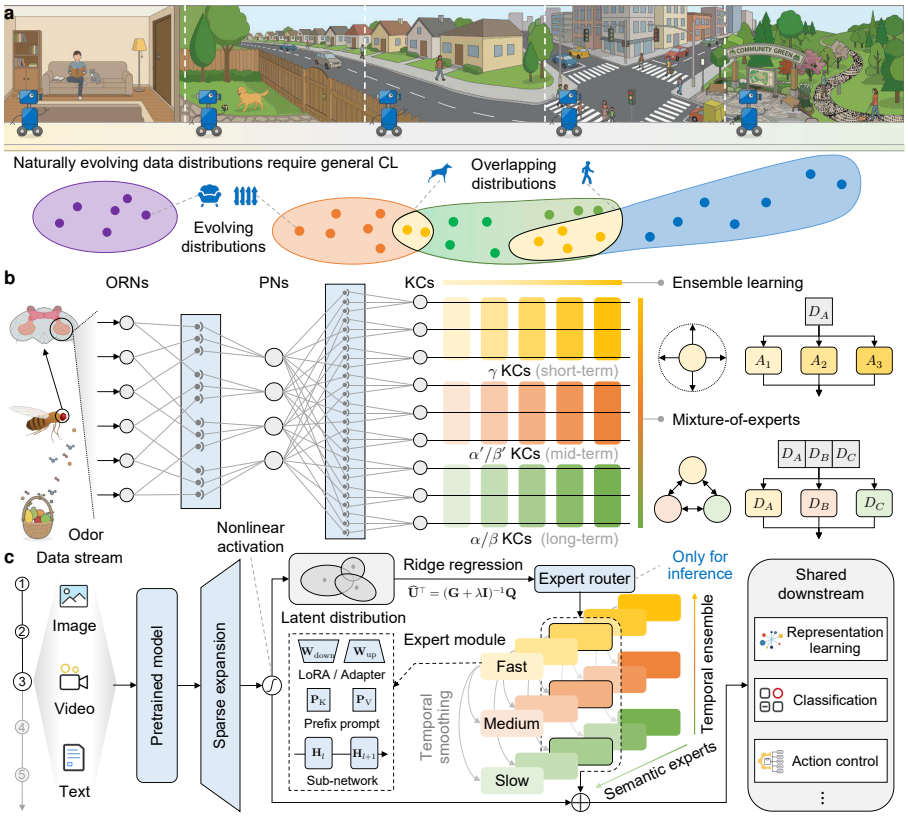


Fig. 1: **Brain-inspired hierarchical modular framework for general continual learning (GCL).** a: Real-world environments induce online and uncertain streams, where data distributions evolve and previously seen concepts may recur over time. b: Biological inspiration from the *Drosophila* olfactory learning and memory system. Olfactory receptor neurons (ORNs) project to projection neurons (PNs) and are sparsely expanded into Kenyon cells (KCs), motivating hierarchical modular learning. The upper branch illustrates ensemble learning, where multiple readouts A_1, A_2, A_3 share the same distribution D_A, while the lower branch illustrates mixture-of-experts, where different distributions D_A, D_B, D_C are assigned to specialized experts. c: Brain-inspired framework for GCL. Online data streams are processed by a pretrained model and adapted through sparse expansion with nonlinear activation, expert routing, and multi-timescale expert integration, where routing is solved by closed-form ridge regression in the latent space. The resulting representations support diverse downstream tasks.

(Fig. 1a). FlyGCL addresses this challenge through brain-inspired hierarchical modularity. We first examine its biological and computational basis, and then evaluate its generality under perceptual and embodied learning scenarios.

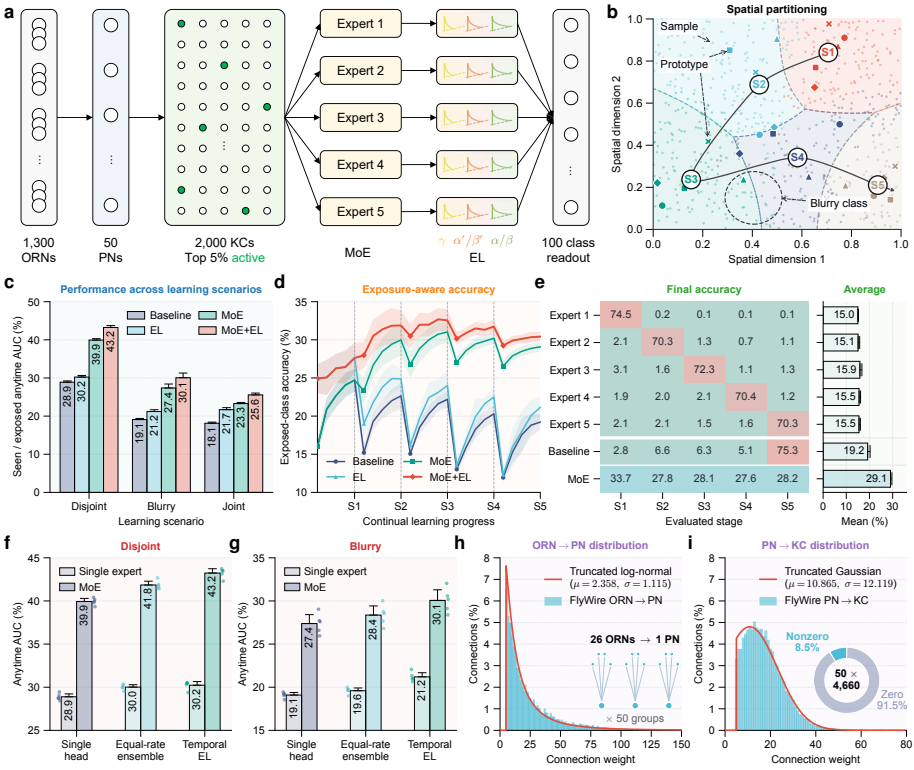


Fig. 2: **Biologically grounded evaluation of brain-inspired hierarchical modularity.** a: *Drosophila*-inspired architecture with ORN-PN compression, sparse PN-KC expansion, spatially differentiated experts, and temporal ensemble learning. ORNs, olfactory receptor neurons; PNs, projection neurons; KCs, Kenyon cells; MoE, mixture-of-experts; EL, ensemble learning. b: Spatial partitioning of 100 classes into five regions according to prototype proximity. c: Area under the exposed-class anytime accuracy curve (AUC) of the baseline, EL, MoE, and MoE+EL under disjoint, blurry, and joint settings. d: Exposed-class accuracy of the four methods over the blurry setting. e: Final accuracy of individual experts, baseline, and MoE across five spatial regions. f and g: Anytime AUC with a single head, equal-rate ensemble, and temporal EL, with or without MoE, under disjoint and blurry settings, respectively. h: FlyWire ORN-PN connection weights and their truncated log-normal approximation; inset, 50 groups with 26 ORNs converging onto one PN. i: FlyWire PN-KC connection weights and their truncated Gaussian approximation; inset, connection sparsity. All results are averaged over five independent runs. Error bars and shaded areas denote 95% confidence intervals.

2.1 Biological and Computational Basis of Brain-Inspired Hierarchical Modularity

The *Drosophila* olfactory system provides a compact biological model of learning from continuously varying sensory experience (Fig. 1b). Odor signals are encoded by around 1300 olfactory receptor neurons (ORNs) of 50 groups and around 150 projection neurons (PNs) of 50 types [41], and then transmitted through sparse, largely

random PN-KC projections into a substantially expanded population of around 2,000 Kenyon cells (KCs) [26, 27, 42]. This transformation produces sparse, distributed representations that reduce overlap between odor patterns and support pattern separation [28, 29]. Downstream of this relatively stable sensory representation, learning and memory are organized through the γ, α'/β', and α/β KC lobes, each partitioned into five anatomically differentiated compartments (a total of 15 modules across three lobes) regulated by distinct dopaminergic and output pathways [30–32]. These compartments provide spatial diversification within each lobe, while the three lobes contribute preferentially to short-term memory, intermediate memory consolidation, and long-term memory, respectively [33, 34]. The mushroom body therefore organizes learning and memory along two complementary axes: spatial compartmentalization within KC lobes and temporal differentiation across them.

We interpret this biological organization computationally as a hierarchy of two complementary paradigms in modular machine learning (Figs. 1b and 1c). At the first level, sparse random expansion separates sensory representations and enables selective recruitment of downstream pathways, providing a biological analogue of *mixture-of-experts* (MoE) routing for reducing interference between dissimilar distributions. At the second level, differentiated compartments provide parallel spatial memory pathways, while the γ, α'/β', and α/β lobes span progressively longer memory timescales. Coordinating information across these spatial and temporal dimensions resembles *ensemble learning* (EL), which exploits diversity and integration to improve generalization over related distributions. Computationally, MoE-like specialization separates conflicting experience, whereas EL-like integration combines compatible information across spatial and temporal memory components. Their hierarchical coordination yields the central principle of GCL: separating conflicting experience while integrating compatible experience.

FlyGCL instantiates this biological organization in pretrained foundation models (Fig. 1c, Methods). The pretrained backbone serves as a relatively stable representational substrate, analogous to upstream sensory processing, while brain-inspired random expansion of its representations enables instance-level routing among specialized experts. Diversified adaptive components provide spatial variation, whereas prediction heads with different effective memory windows provide temporal variation in each expert. Fast predictions emphasize recent experience, slow predictions preserve information over longer timescales, and intermediate heads bridge the two, forming a computational analogue of short-term, consolidation-related, and long-term memory. This hierarchical design can be implemented with lightweight adaptation such as prompts, adapters, or LoRA [43–45], allowing FlyGCL to operate across different pretrained models and learning scenarios. Computational analysis further supports the complementary roles of specialization and integration, and the benefit of coordinating them on stable pretrained representations (Fig. 1c, Methods).

To examine the hierarchical modular principle in a controlled biologically grounded setting, we construct a continual olfactory learning model that follows the population scale and modular organization of the *Drosophila* olfactory system (Fig. 2a, Supplementary Sec. B.1). Following prior task-driven models of olfactory learning [41, 46], the network contains 1,300 ORNs, 50 types of PNs, and

2,000 KCs. ORN-PN and PN-KC connections are sampled using statistics derived from FlyWire [47] (Figs. 2h and 2i), and only the 5% most active KCs are retained for each odor. The sensory pathway remains fixed, restricting CL to downstream learning and memory components. We organize these components along the same spatial-temporal hierarchy: five parallel experts represent memory pathways differentiated by spatial regions, while three temporal heads capture fast, intermediate, and slow effective memory scales. We generate 100 classes from prototypes in a 50-dimensional sensory space and assign nearby prototypes to five equally sized spatial regions. Each online data stream contains 50,000 unique samples presented over five stages, ranging from strictly separated classes in Disjoint to increasing cross-stage overlap in Blurry and Joint (50% and 100% classes overlap, respectively).

We compare four targeted baselines that isolate the two dimensions of this organization: a naive baseline with a single adaptive pathway; MoE with five spatially differentiated experts and an expert router based on accumulated stage prototypes; EL with three prediction heads operating at distinct effective timescales; and the hierarchical model combining five experts with three temporal heads each (Fig. 2a). Across stream configurations, MoE and EL provide complementary benefits, while their hierarchical combination consistently performs best (Figs. 2c and 2d). Under the disjoint setting, the area under the curve (AUC) performance increases from 28.9% for the baseline to 39.9% with MoE and 43.2% with MoE+EL. Under the blurry setting, MoE+EL reaches 30.1%, compared with 19.1% for the baseline, and remains consistently stronger over the course of learning. The same ordering holds under the joint setting, where MoE+EL also outperforms either component alone.

The expert analysis provides direct evidence of specialization (Fig. 2e). Each expert is most accurate in its corresponding region, whereas its accuracy is low elsewhere. Routing these specialized experts produces a mean final accuracy of 29.1% across regions, compared with 19.2% for the shared baseline. Temporal diversity provides an additional consistent gain (Figs. 2f and 2g). For example, under the blurry setting with MoE, anytime AUC increases from 27.4% with a single head to 28.4% with three equal-rate heads and 30.1% when the heads use different learning rates. This progression holds both with and without MoE in the disjoint and blurry settings, supporting distinct contributions from expert specialization and temporal integration.

2.2 General Continual Learning for Visual and Vision-Language Perception

Real-world intelligent systems continuously encounter changing perceptual experience, from evolving visual concepts to multimodal observations grounded in language. Continual visual recognition and continual vision-language learning therefore provide representative scenarios for studying how models preserve shared structure while adapting to distribution-specific changes over time. Although both scenarios have been widely studied in conventional CL, existing efforts largely rely on offline and disjoint task sequences. We revisit them under more realistic online and blurry data streams.

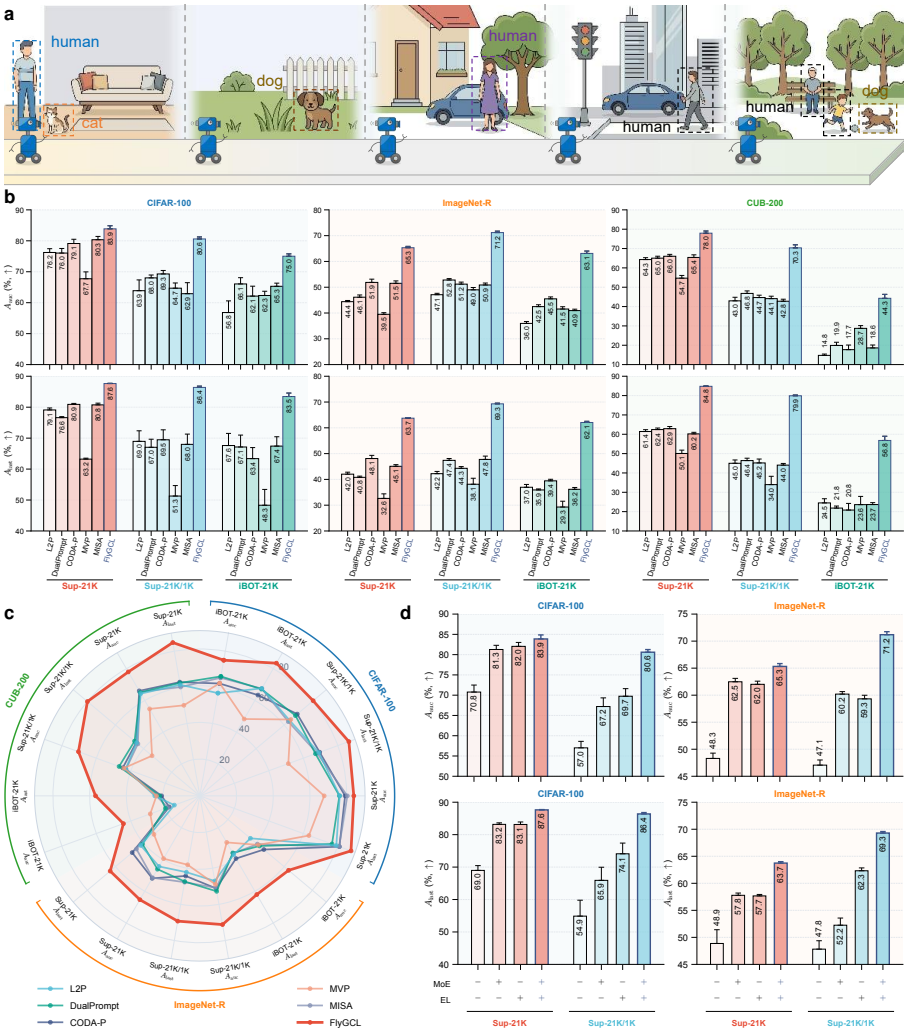


Fig. 3: **Results on continual visual recognition benchmarks.** a: Illustration of real-world vision scenarios, where agents continuously encounter diverse objects across evolving sessions. b: Overall performance comparison with state-of-the-art baselines. c: Unified performance summary across datasets, pretrained models, and evaluation metrics. d: Ablation analysis of mixture-of-experts (MoE) and ensemble learning (EL). In b–d, FlyGCL uses LoRA modules as adaptive experts. All results are averaged over five independent runs. Error bars indicate the standard error of the mean. Complete results for prompt-, adapter-, and LoRA-based instantiations are reported in Supplementary Tabs. S3 and S4.

Continual Visual Recognition. We first evaluate FlyGCL on continual visual recognition under online and blurry data streams, where samples from newly introduced and previously observed classes are probabilistically interleaved [21, 48], producing uncertain and evolving distributions over time (Fig. 3a). We consider CIFAR-100 [49], ImageNet-R [50], and CUB-200 [51] datasets, spanning generic

object recognition, distribution-shifted concepts, and fine-grained categories. Comparisons include representative pretrained-based CL methods, such as L2P [52], DualPrompt [53], and CODA-Prompt [54], as well as online CL methods MVP [21] and MISA [48]. We report average anytime performance (A_{auc}) and final average performance (A_{last}). Across these benchmarks, FlyGCL consistently achieves the strongest final and anytime performance (Figs. 3b and 3c, Supplementary Tab. S3): in the primary comparison, its A_{auc}/A_{last} exceed the strongest baseline by 3.5%/6.9%, 13.8%/18.7%, and 12.6%/24.6% on CIFAR-100, ImageNet-R, and CUB-200, respectively. The performance gains are particularly clear on ImageNet-R and CUB-200, where distribution shifts and fine-grained distinctions place greater demands on selective adaptation.

We next test whether this advantage depends on the pretrained representation or the adaptation interface. The primary comparison covers three backbone settings: a model pretrained on ImageNet-21K (Sup-21K), a model pretrained on ImageNet-21K and subsequently adapted to ImageNet-1K (Sup-21K/1K) [55–57], and a self-supervised iBOT model pretrained on ImageNet-21K (iBOT-21K) [58] (Fig. 3b). We additionally evaluate self-supervised checkpoints pretrained on ImageNet-1K, including iBOT [58], DINO [59], and MoCo v3 [60] (Supplementary Tab. S3). Across these settings, FlyGCL supports prompt-, adapter-, and LoRA-based adaptation [43–45, 61], and retains strong performance across the resulting combinations. This consistency shows that the proposed hierarchical modularity is not tied to a particular pretrained representation or tuning interface.

Finally, we isolate the roles of the two components of FlyGCL design. Removing either MoE or EL reduces performance across datasets and pretrained representations, whereas their combination performs best (Fig. 3d, Supplementary Tab. S4). Expert routing provides differentiated adaptation paths for heterogeneous visual distributions, while temporal ensemble integration stabilizes predictions as related experience recurs. These results support that hierarchical specialization and integration provide complementary gains for continual visual recognition.

Continual Vision-Language Learning. We next extend GCL to vision-language models, where CL must accommodate evolving visual concepts while preserving the cross-modal semantic structure acquired during large-scale pretraining (Fig. 4a). Compared with visual recognition, CL introduces an additional challenge: changes in visual representations may disrupt their correspondence with language and weaken the shared semantic space that supports multimodal generalization. We evaluate FlyGCL with pretrained CLIP [62] on CIFAR-100 and ImageNet-R, comparing with classical CL methods such as EWC [17] and LwF [63], as well as CLIP-based CL methods including CLAP4CLIP [64] and MG-CLIP [65]. FlyGCL achieves the strongest overall performance across both benchmarks (Figs. 4b and 4c, Supplementary Tab. S5), reaching A_{auc}/A_{last} of 85.2/79.6% on CIFAR-100 and 84.7/79.3% on ImageNet-R, while also maintaining favorable forgetting and backward transfer. These results show that hierarchical modularity remains effective when GCL extends from visual prediction to continually evolving cross-modal representations.

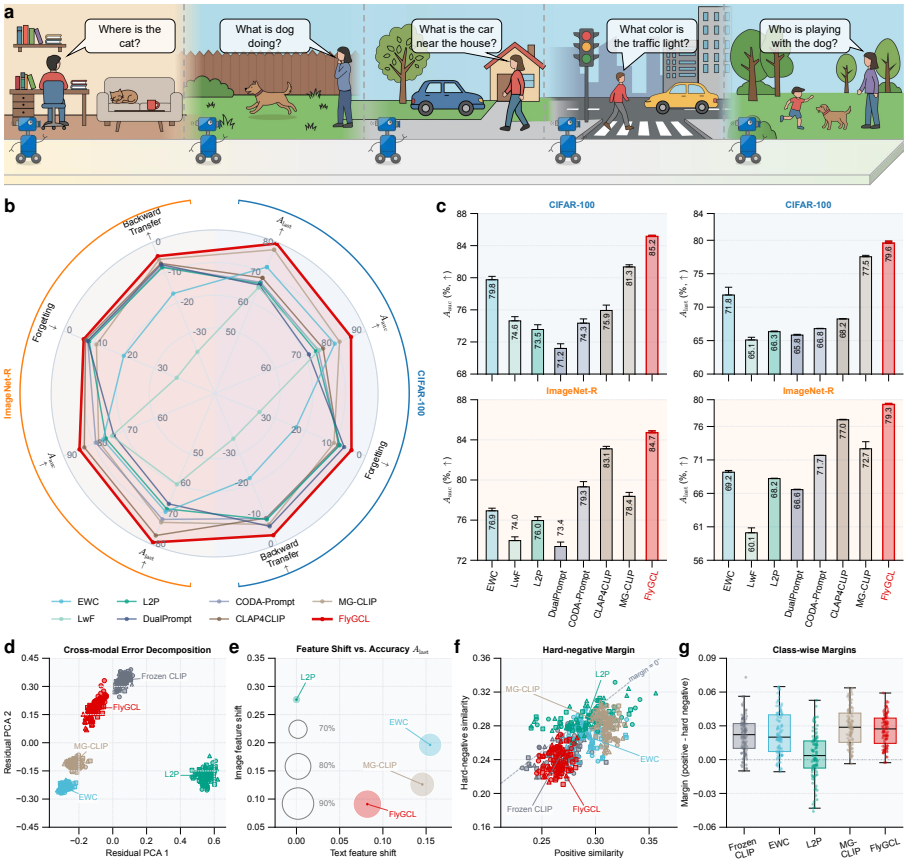


Fig. 4: **Results on continual vision-language benchmarks.** a: Illustration of real-world vision-language scenarios, where an embodied agent continuously encounters evolving visual contexts and language queries. b: Unified performance summary on CIFAR-100 and ImageNet-R using A_{auc}, A_{last}, forgetting, and backward transfer. c: Performance comparison on CIFAR-100 and ImageNet-R. d: Principal component analysis (PCA) projection of representations on CIFAR-100. e: Vision-language feature drift relative to frozen CLIP. Bubble size denotes A_{last}. f: Hard-negative separation by positive-pair and hard-negative similarities. g: Class-wise vision-language margin, computed as positive-pair similarity minus hard-negative similarity. All results are averaged over five independent runs. Error bars indicate the standard error of the mean. Detailed numerical results are reported in Supplementary Tab. S5.

We examine how different continual learners reshape the pretrained image-text representation space. Principal component analysis (PCA) visualization of the representation residuals reveals distinct adaptation trajectories relative to frozen CLIP (Fig. 4d). FlyGCL exhibits a more constrained representation shift than competing approaches, while retaining the highest final performance. The joint decomposition of image- and text-feature shifts shows the same pattern (Fig. 4e): FlyGCL remains closer to the pretrained cross-modal space without sacrificing adaptation accuracy.

This balance is consistent with the hierarchical design of FlyGCL, where selective expert adaptation accommodates distribution-specific changes while temporal integration limits unnecessary displacement of the pretrained representation.

We further analyze whether this preservation extends to the semantic relationships that underpin image-text recognition. For each image, we compare its similarity to the matched text with that of the hardest negative text (Fig. 4f). CL generally narrows this separation relative to frozen CLIP, whereas FlyGCL better preserves the positive-to-hard-negative margin. The class-wise analysis confirms that this advantage is broadly distributed across semantic categories rather than driven by a small subset of classes (Fig. 4g). The representation- and similarity-level analyses indicate that FlyGCL adapts to evolving visual distributions while better retaining the pretrained image-text geometry, providing a mechanistic explanation for its stronger continual vision-language learning performance.

2.3 General Continual Learning for Embodied Perception and Action

Real-world intelligent systems must perceive changing environments while learning continuously from embodied and human-centered experience. Continual ego-exo video understanding and continual vision-language-action learning represent two important settings in this direction, spanning the interpretation of evolving human activities and the acquisition of action policies through multimodal interaction. These scenarios remain relatively underexplored in conventional CL, despite their direct relevance to long-running intelligent agents. Their video and interaction streams are inherently online, temporally continuous, and distributionally blurry, closely matching the uncertain and evolving experience targeted by GCL. We therefore examine whether FlyGCL can extend from perception to embodied understanding and action.

Continual Ego-Exo Video Understanding. We extend GCL to continual ego-exo video understanding [67], where agents must continuously learn evolving activities from heterogeneous first- and third-person observations (Fig. 5a). Compared with image-based GCL, continual video learning introduces additional temporal and viewpoint variations: the same activity can exhibit substantially different visual dynamics across egocentric and exocentric views, while subject, skill, and motion patterns evolve continuously throughout the data stream. We evaluate FlyGCL on EgoExoLearn [66] and EgoExo-Fitness [68], covering continual skill assessment, action anticipation, and action classification. Across these settings, FlyGCL achieves consistently strong overall performance on both A_{auc} and A_{last} (Figs. 5b and 5c, Supplementary Tabs. S6 to S9), showing that the brain-inspired hierarchical modularity remains effective when GCL extends from static visual recognition to temporally evolving and cross-view video representations.

The performance gains are consistent across distinct forms of embodied video understanding. On EgoExoLearn skill assessment, FlyGCL reaches $A_{\text{auc}}/A_{\text{last}}$ of $81.3\%/82.6\%$, outperforming replay-, regularization-, and prompt-based methods. On EgoExo-Fitness skill assessment, FlyGCL reaches $62.1\%/62.7\%$, compared with $58.0\%/57.9\%$ for the strongest competing method in the main comparison (Fig. 5c).

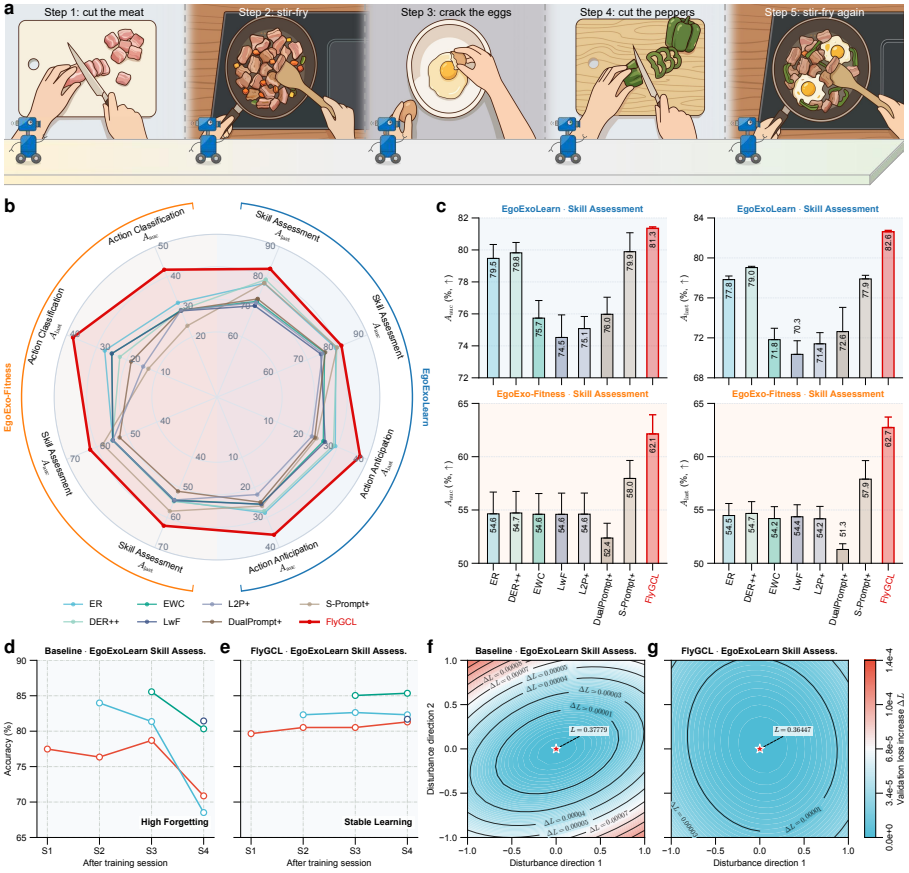


Fig. 5: **Results on continual ego-exo video understanding benchmarks.** a: Illustration of continual ego-exo video understanding, where embodied agents encounter evolving activities and viewpoints over time. b: Unified performance summary on EgoExoLearn and EgoExo-Fitness using A_{auc} and A_{last}. For skill assessment on EgoExoLearn and EgoExo-Fitness, results are averaged over the Relation Network (RN)- and Triplet Loss (TL)-based ego-exo model variants [66]. c: Performance comparison for continual skill assessment on EgoExoLearn and EgoExo-Fitness. d, e: Stage-wise accuracy of baseline and FlyGCL on EgoExoLearn. f, g: Local loss landscapes under parameter perturbations for baseline and FlyGCL. All results are averaged over three independent runs. Error bars indicate the standard error of the mean. Detailed numerical results are reported in Supplementary Tabs. S6 to S9.

Additional evaluations on sequence verification and guidance-based execution verification (Supplementary Tabs. S10 and S11) further demonstrate that the benefit of hierarchical modularity extends across distinct video tasks and output spaces.

We further examine how different continual learners evolve as new ego-exo sessions arrive. The stage-wise trajectories (Figs. 5d and 5e) reveal substantially different forgetting patterns: the sequential fine-tuning baseline progressively degrades on previously learned sessions, exhibiting substantial forgetting on EgoExoLearn

skill assessment, whereas FlyGCL largely preserves earlier performance and maintains stable learning throughout the continual stream. The local loss landscapes provide complementary evidence (Figs. 5f and 5g): FlyGCL converges to a flatter basin and exhibits lower sensitivity to parameter perturbations than the sequential fine-tuning baseline. These results suggest that hierarchical modularity improves not only average continual performance but also the robustness of the learned solution, with specialized experts accommodating heterogeneous temporal and viewpoint-specific patterns while temporal integration limits destructive interference as related embodied experience recurs.

Continual Vision-Language-Action Learning. We further extend GCL to continual vision-language-action learning, where embodied agents must connect visual observations and language instructions to actions as environments, goals, and interaction states evolve over time (Fig. 6a). Unlike recognition or representation learning, continual updates in this setting affect an entire interaction policy. The agent must preserve visuolinguistic grounding while adapting its visuomotor behaviour to new experience, even as related situations recur without clear task boundaries. Changes in perception or action generation can alter subsequent states and compound over an interaction trajectory, placing demands on both stable representation learning and policy adaptation. We evaluate FlyGCL on LIBERO-Spatial, -Object, -Goal, and -Long [69] under the primary online GCL stream and a complementary offline protocol. In addition to EWC, LwF, L2P+, and DualPrompt+, we compare with sequential fine-tuning (SeqFT), LoRA fine-tuning (SeqLoRA), and PackNet [70].

In the online setting, task distributions overlap and recur throughout the stream, requiring the agent to acquire new visuomotor behaviours while retaining earlier ones. On LIBERO-Spatial and LIBERO-Object, FlyGCL reaches $A_{\text{last}}/A_{\text{auc}}$ of 83.1%/81.6% and 86.1%/86.0%, exceeding the strongest replay-free baselines by 49.5%/37.5% and 49.8%/35.9%, respectively (Figs. 6b and 6c). The same pattern holds for goal-conditioned and long-horizon manipulation (Extended Data Figs. 1a and 1b and Supplementary Tab. S12). FlyGCL also maintains strong forward and backward transfer across the four suites, indicating that it can incorporate new interaction patterns without sacrificing previously acquired behaviours. These results extend the benefit of hierarchical modularity from perceptual representations to instruction-conditioned action policies.

We also evaluate the four suites under the offline protocol, in which each task is trained for multiple passes before the learner proceeds to the next one. FlyGCL remains consistently strong across spatial, object-centric, goal-conditioned, and long-horizon manipulation (Extended Data Fig. 2 and Supplementary Tab. S13). This result shows that its effectiveness is not limited to rapid online transitions. It also applies when task changes are more structured but each task still contains continuously varying visual states, action trajectories, and interaction outcomes.

The rollout analyses further show how this stability affects task execution (Figs. 6d and 6e, Extended Data Figs. 1c and 1d). We compare FlyGCL with regularization-based EWC and task-expert-based DualPrompt+. After subsequent updates, both methods can lose a critical part of behaviours that they execute successfully immediately after learning. Their failures range from spatial grounding and

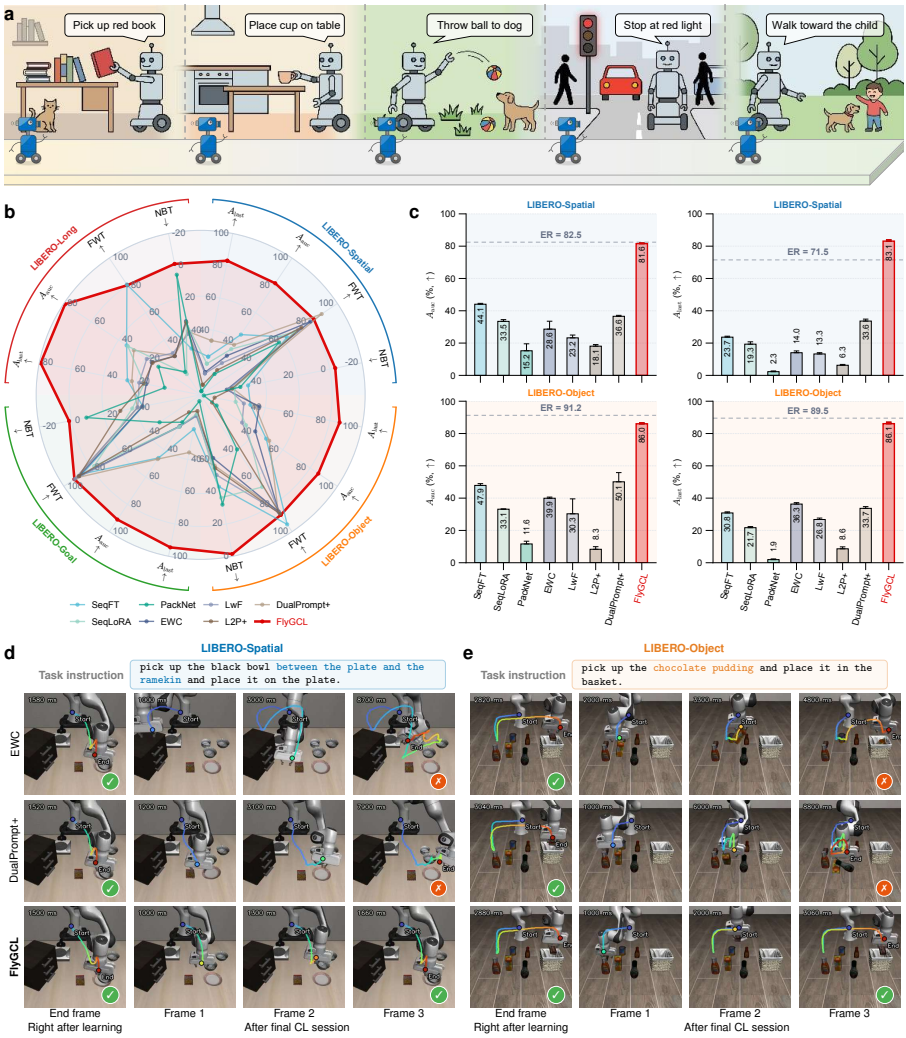


Fig. 6: **Results on continual vision-language-action benchmarks.** a: Illustration of continual embodied interaction, where agents follow evolving language instructions across changing environments and tasks. b: Unified performance summary across LIBERO benchmarks and CL metrics. c: Performance comparison with state-of-the-art baselines on LIBERO-Spatial and LIBERO-Object using A_{auc} and A_{last}. d: Qualitative rollouts on LIBERO-Spatial, comparing task execution immediately after learning and after the final CL session for EWC, DualPrompt+, and FlyGCL. e: Qualitative rollouts on LIBERO-Object under the same protocol. Results are averaged over three runs. Error bars denote the standard error of the mean.

object selection, as illustrated by DualPrompt+ missing the target position of the black bowl and EWC selecting the wrong object instead of the chocolate pudding, to incomplete goal execution and long-horizon action sequences. In the latter cases, EWC pursues an incorrect goal state, while DualPrompt+ completes the placement

step but fails to close the drawer. FlyGCL more consistently preserves complete task execution across subsequent sessions, consistent with expert routing separating task-specific visuomotor changes and temporal integration retaining useful information across recurring experience.

3 Discussion

This work identifies hierarchical modularity as a unified computational principle for GCL under online, uncertain, and evolving data streams. Beyond preserving previous knowledge, GCL demands selectively separating conflicting experience to reduce interference while integrating compatible experience to promote generalization. Inspired by the *Drosophila* olfactory learning and memory system, FlyGCL coordinates both functions through expert specialization and ensemble integration. Controlled olfactory experiments demonstrate their additive benefits, with hierarchical coordination consistently outperforming either component. FlyGCL improves GCL across visual recognition, vision-language understanding, ego-exo video understanding, and embodied vision-language-action learning, spanning diverse pretrained representations and parameter-efficient adaptation mechanisms. Some simplified brain-inspired components were explored in our earlier conference paper [71]. The present work proposes a holistic model of the biological learning and memory system with formal analysis, biological simulation and substantially broader validation across models, modalities, and application scenarios (Supplementary Information).

From an AI perspective, GCL reflects a broader transition from intelligence acquired from static human-curated data to sequentially arriving experience. Recent perspectives on the "era of experience" [8], autonomous machine intelligence [9], and open-ended learning [10] similarly envision agents that continually extend their capabilities through interaction with the external world. Such agents must not only accumulate experience, but also determine what should be reused, separated, or integrated as distributions change. GCL provides a concrete learning paradigm for this process by bringing CL closer to the online, uncertain, and evolving conditions faced by real-world agents. This requirement arises in long-running systems such as embodied robots, autonomous vehicles, personalized assistants, and healthcare or scientific monitoring systems, where perception, knowledge, and behaviour must be continually updated without predefined task boundaries. FlyGCL provides a biologically grounded realization of this idea by organizing learning according to relationships among evolving experiences.

The biological implications are equally important. Recent whole-brain connectomes and cross-connectome cell typing have provided increasingly detailed maps of the *Drosophila* nervous system [23–25], but how this anatomical organization supports learning from changing experience remains less understood. Our computational model offers a functional interpretation of the olfactory learning and memory system by linking sparse expansion and differentiated memory pathways to specialization and integration. Controlled olfactory experiments further show that their hierarchical coordination improves learning as experience shifts from disjoint to increasingly recurrent distributions. These results suggest testable roles for the underlying circuit

organization: differentiated pathways may reduce interference between dissimilar experiences, whereas coordinated integration may preserve shared structure and improve generalization across related experiences. In this way, computational modelling can complement connectomics by linking anatomical organization to functional principles of learning and memory.

More broadly, our study exemplifies a bidirectional NeuroAI framework in which biological organization inspires computational principles, while computational models generate testable hypotheses for neuroscience. The hierarchical organization of *Drosophila* olfactory learning and memory motivates a unified view of MoE and EL as complementary computational paradigms for specialization and integration in GCL. In turn, our results suggest specific biological predictions: sparse expansion and differentiated downstream pathways should improve separation and reduce interference between dissimilar experiences; memory pathways operating across distinct spatial and temporal scales should contribute complementary information when related experiences recur; and their hierarchical coordination should be particularly beneficial when conflicting and compatible experiences coexist over time. Extending these principles to embodied GCL also aligns with the NeuroAI vision of an "embodied Turing test", in which intelligence develops through continuous sensorimotor interaction with the external world [72].

Several directions remain beyond the scope of this study. Our biological model captures the overall organization of the *Drosophila* olfactory learning and memory system, leaving richer neuromodulatory dynamics, mechanisms of memory consolidation, and behavioural feedback for future computational and experimental investigation. Meanwhile, our benchmarks extend GCL towards online, multimodal, and embodied experience, while longer-term open-ended interaction may additionally require active exploration, evolving objectives, and adaptive allocation of learning resources. FlyGCL currently concentrates plasticity in lightweight modules over relatively stable pretrained representations. Understanding how hierarchical specialization and integration can operate as foundation-model representations themselves continue to evolve remains an important direction. Looking ahead, incorporating richer biological mechanisms and extending GCL towards autonomous agents capable of learning from open-ended experience and recursively improving their own learning processes may help advance a more general science of intelligence that develops through experience.

4 Methods

4.1 Problem Formulation

General Continual Learning. Conventional CL [1, 6, 15] often studies well-separated tasks under largely offline training, with previous-task data unavailable during subsequent updates. In contrast, GCL considers single-pass, non-stationary streams with uncertain and blurry data distributions, without clear task boundaries or reliable task identities [2, 21, 73]. Formally, the data stream consists of T sessions, $\mathcal{S} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_T\}$, where each session is given by $\mathcal{D}_t = \{(\mathbf{o}_{t,i}, \mathbf{u}_{t,i})\}_{i=1}^{N_t}$. Here,

$\mathbf{o}_{t,i} \in \mathcal{O}_t$ denotes an observation, and $\mathbf{u}_{t,i} \in \mathcal{U}_t$ denotes its associated learning signal. This notation provides a unified description of the GCL settings studied in this work: $\mathbf{o}$ may be instantiated as an image, an image-text pair, a video clip, an ego-exo multi-view sequence, or an embodied visual-language state, while $\mathbf{u}$ may correspond to a class label, semantic target, retrieval correspondence, temporal annotation, quality score, action, trajectory, or task-success signal. A standard model consists of a backbone $f_\theta(\cdot)$ and an output module $g_\psi(\cdot)$. For notational convenience, we denote the resulting predictor as $F_{\theta,\psi} = g_\psi \circ f_\theta$, i.e., $\hat{\mathbf{u}} = F_{\theta,\psi}(\mathbf{o}) = g_\psi(f_\theta(\mathbf{o}))$. The learning objective is to obtain a unified mapping from $\mathcal{O} = \bigcup_{t=1}^{T} \mathcal{O}_t$ to $\mathcal{U} = \bigcup_{t=1}^{T} \mathcal{U}_t$, while learning online and retaining previously acquired knowledge.

At session t, samples are drawn from a local distribution $(\mathbf{o}_{t,i}, \mathbf{u}_{t,i}) \sim P_t(\mathbf{o}, \mathbf{u})$, where P_t can evolve gradually or abruptly, overlap with previous distributions, and recur later in the stream. In classification-based GCL, such overlap is commonly instantiated by the Si-Blurry setting [21, 48], which decomposes the global label space into a disjoint subset $\mathcal{Y}^D$ and a blurry subset $\mathcal{Y}^B$, with $\mathcal{Y} = \mathcal{Y}^D \cup \mathcal{Y}^B$ and $\mathcal{Y}^D \cap \mathcal{Y}^B = \varnothing$. Classes in $\mathcal{Y}^D$ are primarily associated with specific sessions, whereas classes in $\mathcal{Y}^B$ can reappear across multiple sessions. The disjoint ratio $r_D = |\mathcal{Y}^D|/|\mathcal{Y}|$ controls the degree of session-specific separation, while the blurry sample ratio r_B controls the frequency of recurring samples. Beyond classification, the same principle also applies more broadly, where semantic concepts, temporal states, skill levels, or action patterns may be partially shared across sessions.

Specialization and Integration. GCL requires determining when incoming knowledge should be shared and when it should be separated. Local distributions may share task-relevant structure in representations, semantics, temporal dynamics, viewpoints, or behaviours, while differing in gradients, decision boundaries, temporal alignments, or action policies. Related distributions should therefore share information to promote generalization, whereas conflicting distributions should be separated to reduce interference. A fully shared predictor $F_{\theta,\psi}$ can exploit common structure but is vulnerable to interference, whereas fully isolated predictors preserve distribution-specific knowledge at the cost of useful transfer. We formalize this trade-off with modular machine learning. Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ denote a set of trainable modules, such as prompts, adapters, LoRA branches, task heads, video-specific modules, or policy modules. When modules have their own output heads, we denote them by $\Psi = \{\psi_1, \psi_2, \ldots, \psi_K\}$. Given a representation $\mathbf{h} = f_\theta(\mathbf{o})$, a routing function produces module weights $\pi(\mathbf{o}) = r_\eta(\mathbf{h}) \in \Delta^{K-1}$. We use $F_{\theta,\Omega,\Psi}$ to denote the stream-level modular predictor induced by the backbone, modules, output heads, and routing or aggregation mechanism.

MoE supports specialization by assigning inputs to adaptive modules. For example, with $k^* = \arg\max_k \pi_k(\mathbf{o})$, the routed prediction can be written as

$$F_{\mathrm{MoE}}(\mathbf{o}) = F_{\theta,\omega_{k^*},\psi_{k^*}}(\mathbf{o}), \tag{1}$$

which reduces interference by limiting updates and predictions to the selected module. EL instead integrates multiple predictors,

$$F_{\mathrm{EL}}(\mathbf{o}) = \mathcal{A}\left(F_{\theta,\omega_1,\psi_1}(\mathbf{o}), \ldots, F_{\theta,\omega_K,\psi_K}(\mathbf{o})\right), \tag{2}$$

where $\mathcal{A}(\cdot)$ denotes an aggregation function. This improves robustness and generalization for similar distributions by combining diverse but related predictions.

However, directly combining MoE and EL is non-trivial because they favor different distributional structures [40]. EL promotes generalization when predictors capture identical or closely related distributions [74], whereas effective MoE routing relies on structured separation among heterogeneous distributions [75]. In GCL, compatible and conflicting distributions may coexist: overly exclusive routing can suppress transfer among related distributions, whereas overly broad ensembling can mix incompatible modules and reintroduce interference. GCL therefore requires hierarchical coordination between specialization and integration, separating conflicting distributions while integrating compatible ones.

4.2 Theoretical Analysis

Decomposing Hierarchical GCL Risk. The preceding formulation shows that effective GCL requires coordinating specialization and integration under uncertain and evolving data streams. We analyze this requirement using the stream-level modular predictor $F_{\theta,\Omega,\Psi}$ defined above, which comprises the backbone f_θ, trainable modules Ω, output modules Ψ, and the associated routing or aggregation operations. Let $F^\star$ denote the ideal stream-level predictor obtained if the latent structure of the local distributions $\{P_t\}_{t=1}^T$ were known. We define the expected GCL risk as

$$\mathcal{R}_{\mathrm{GCL}}(F_{\theta,\Omega,\Psi}) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{(\mathbf{o},\mathbf{u})\sim P_t}\left[\ell\left(F_{\theta,\Omega,\Psi}(\mathbf{o}), \mathbf{u}\right)\right], \tag{3}$$

where $\ell(\cdot)$ denotes the task-specific loss. Because $F^\star$ is unavailable in CL, the practical objective is to approximate this ideal predictor from the observed stream $\mathcal{S}$ under non-stationary distribution shifts.

Theorem 1 (Hierarchical Decomposition of GCL Risk) *Let $F_{\theta,\Omega,\Psi}$ be a hierarchical modular predictor learned from $\mathcal{S} = \{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$, where $\mathcal{D}_t = \{(\mathbf{o}_{t,i}, \mathbf{u}_{t,i})\}_{i=1}^{N_t}$. Under an additive excess-risk decomposition relative to $F^\star$, the expected GCL risk is upper bounded by*

$$\mathcal{R}_{\mathrm{GCL}}(F_{\theta,\Omega,\Psi}) \lesssim \mathcal{R}_{\mathrm{fit}}(F_{\theta,\Omega,\Psi}) + \underbrace{\mathcal{E}_{\mathrm{sep}}(F_{\theta,\Omega,\Psi})}_{\textit{separation error}} + \underbrace{\mathcal{E}_{\mathrm{int}}(F_{\theta,\Omega,\Psi})}_{\textit{integration error}} + \underbrace{\mathcal{C}_{\mathrm{coord}}(F_{\theta,\Omega,\Psi})}_{\textit{coordination cost}}. \tag{4}$$

Here, the empirical fitting term is defined on the observed stream as

$$\mathcal{R}_{\mathrm{fit}}(F_{\theta,\Omega,\Psi}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}\ell\left(F_{\theta,\Omega,\Psi}(\mathbf{o}_{t,i}), \mathbf{u}_{t,i}\right). \tag{5}$$

The remaining terms are excess-risk components: $\mathcal{E}_{\mathrm{sep}}(F_{\theta,\Omega,\Psi})$ is induced by insufficient separation of conflicting local distributions, $\mathcal{E}_{\mathrm{int}}(F_{\theta,\Omega,\Psi})$ is induced by insufficient integration of

compatible predictions, and $\mathcal{C}_{\mathrm{coord}}(F_{\theta,\Omega,\Psi})$ *is the additional cost introduced by coordinating routing and integration.*

The proof is provided in Supplementary Sec. A.1. The three terms characterize distinct failure modes of hierarchical modular learning. The separation error $\mathcal{E}_{\mathrm{sep}}$ captures residual interference when samples with conflicting gradients, decision boundaries, temporal alignments, or policies share parameters. The integration error $\mathcal{E}_{\mathrm{int}}$ captures the residual estimation gap when related local distributions or compatible predictors are treated in isolation. The coordination cost $\mathcal{C}_{\mathrm{coord}}$ arises from jointly performing routing and integration, and is defined as the positive excess loss of the hierarchical modular predictor relative to the ideal stream-level predictor:

$$\mathcal{C}_{\mathrm{coord}}(F_{\theta,\Omega,\Psi}) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{(\mathbf{o},\mathbf{u})\sim P_t}\left[\ell\left(F_{\theta,\Omega,\Psi}(\mathbf{o}),\mathbf{u}\right) - \ell\left(F^{\star}(\mathbf{o}),\mathbf{u}\right)\right]_{+}, \tag{6}$$

where $[\cdot]_{+}$ denotes the positive part. This term accounts for over-separation of related distributions, over-integration of incompatible modules, routing errors, and aggregation mismatch. Effective GCL therefore requires coordinating separation and integration rather than treating routing and aggregation as independent operations.

Separation and Integration Gains. The decomposition in Eq. (4) indicates that hierarchical modular learning is beneficial only when the gains from separation and integration outweigh the additional cost of coordinating them. We define the routing gain and ensemble gain as

$$\Delta_{\mathrm{route}} = \mathcal{E}_{\mathrm{sep}}(F_{\theta,\psi}) - \mathcal{E}_{\mathrm{sep}}(F_{\mathrm{MoE}}), \quad \Delta_{\mathrm{ens}} = \mathcal{E}_{\mathrm{int}}(F_{\mathrm{single}}) - \mathcal{E}_{\mathrm{int}}(F_{\mathrm{EL}}), \tag{7}$$

where $F_{\theta,\psi}$ is the fully shared predictor, F_{MoE} is the routed modular predictor in Eq. (1), F_{single} denotes a predictor using a single adaptive path, and F_{EL} is the integrated predictor in Eq. (2). Δ_{route} measures the reduction in separation error obtained by routing conflicting samples to different adaptive modules, while Δ_{ens} measures the reduction in integration error obtained by aggregating compatible predictors.

Proposition 1 (Separation and Integration Gains) *Assume that routing provides a non-negative separation gain* $\Delta_{\mathrm{route}} \geq 0$ *and aggregation provides a non-negative integration gain* $\Delta_{\mathrm{ens}} \geq 0$*. Then the GCL risk of a hierarchical modular predictor* $F_{\theta,\Omega,\Psi}$ *satisfies*

$$\begin{aligned}\mathcal{R}_{\mathrm{GCL}}(F_{\theta,\Omega,\Psi}) \lesssim & \mathcal{R}_{\mathrm{fit}}(F_{\theta,\Omega,\Psi}) + \mathcal{E}_{\mathrm{sep}}(F_{\theta,\psi}) + \mathcal{E}_{\mathrm{int}}(F_{\mathrm{single}}) \\ & - \Delta_{\mathrm{route}} - \Delta_{\mathrm{ens}} + \mathcal{C}_{\mathrm{coord}}(F_{\theta,\Omega,\Psi}).\end{aligned} \tag{8}$$

Consequently, hierarchical modular learning improves the risk bound whenever

$$\Delta_{\mathrm{route}} + \Delta_{\mathrm{ens}} > \mathcal{C}_{\mathrm{coord}}(F_{\theta,\Omega,\Psi}). \tag{9}$$

The proof is provided in Supplementary Sec. A.2. Proposition 1 makes the trade-off in hierarchical modular learning explicit. Routing reduces interference

by separating conflicting local distributions, whereas integration improves robustness by combining compatible predictions. The two operations are nevertheless coupled: overly exclusive routing may separate related distributions, while overly broad aggregation may mix incompatible modules. Effective GCL therefore requires coordinating separation and integration rather than optimizing either operation in isolation.

Pretraining-Supported Coordination. The condition in Eq. (9) shows that the benefit of hierarchical modular learning depends on both the gains from routing and aggregation and the cost of coordinating them. This coordination cost is particularly relevant in GCL, where the distinction between related and conflicting local distributions may be uncertain. We formalize this effect by separating the probability of an imperfect modular decision from its resulting prediction penalty.

Proposition 2 (Pretraining-Supported Coordination) *Let* ϵ_{route} *denote the degree of routing error,* ϵ_{agg} *denote the mismatch introduced by aggregating modular predictions, and* $\Delta_{\text{mis}}(f_\theta)$ *denote the prediction penalty of assigning an input to a suboptimal module under the representation induced by* f_θ*. Then the coordination cost is bounded by*

$$\mathcal{C}_{\text{coord}}(F_{\theta,\Omega,\Psi}) \leq \epsilon_{\text{route}}\Delta_{\text{mis}}(f_\theta) + \epsilon_{\text{agg}}. \tag{10}$$

Moreover, if a pretrained backbone provides a more stable and semantically organized representation than a representation learned from scratch on the online stream, then

$$\Delta_{\text{mis}}(f_\theta^{\text{pre}}) < \Delta_{\text{mis}}(f_\theta^{\text{scratch}}), \tag{11}$$

which reduces the coordination cost in Eq. (10)*.*

The proof is provided in Supplementary Sec. A.3. Proposition 2 explains how pretrained representations reduce the cost of imperfect coordination in hierarchical GCL. A routing error is more harmful when the selected module produces predictions that differ substantially from the appropriate one, whereas this penalty is reduced when the pretrained backbone maps related inputs into a stable semantic space. Pretrained representations therefore contribute not only positive transfer and resistance to forgetting, but also greater robustness to imperfect routing and integration. This supports hierarchical MoE-EL as a lightweight modular design over stable pretrained foundation models.

Implications for FlyGCL. The analysis above identifies three requirements for GCL: separating conflicting local distributions, integrating compatible predictions, and limiting the coordination cost between them. These considerations motivate hierarchical modularity as the design principle of FlyGCL, particularly over stable pretrained representations. We next instantiate this principle in a brain-inspired GCL framework and describe its architecture and optimization.

4.3 FlyGCL Model

Model Overview. FlyGCL is a unified brain-inspired framework for GCL with pretrained foundation models. Its design follows the organization of the *Drosophila* olfactory learning and memory system at three functional levels. Sparse random

expansion provides a distributed representation analogous to the PN-KC pathway and supports selective recruitment of downstream pathways. Spatially differentiated experts provide parallel memory pathways, while output heads with different update timescales capture complementary short- and long-term information. Therefore, stable representation, expert routing, and temporal integration form a computational hierarchy for separating conflicting experience and integrating compatible predictions.

Pretraining-based CL commonly introduces parameter-efficient tuning components, such as prompts, adapters, and LoRA, as lightweight experts over a pretrained backbone. Each expert provides a specialized learning pathway parameterized by $\boldsymbol{\omega}$ and accumulates spatially differentiated knowledge, producing outputs $f_{\boldsymbol{\theta}}(\boldsymbol{x};\ \boldsymbol{\omega})$. Under single-pass blurry data streams, expert-based learning must determine both the appropriate expert for each input and how to maintain reliable predictions under limited and imbalanced online supervision. FlyGCL addresses these challenges through random-expanded analytic routing and temporal ensemble integration within each routed expert. Let $f_{\boldsymbol{\theta}}(\cdot)$ denote a pretrained backbone and $\boldsymbol{h} = f_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathbb{R}^d$ its representation. The trainable expert pool is $\Omega = \{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_K\}$, and the prediction of expert E_k is denoted by $F_{\boldsymbol{\theta},\boldsymbol{\omega}_k,\boldsymbol{\psi}_k}(\boldsymbol{x})$, where $\boldsymbol{\psi}_k$ is its output head.

Random-Expanded Analytic Router. To improve expert routing, FlyGCL uses a random-expanded analytic router inspired by sparse expansion in the fruit fly mushroom body. Given $\boldsymbol{h} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$, we apply a fixed random projection followed by nonlinear activation:

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \sigma\left(f_{\boldsymbol{\theta}}(\boldsymbol{x})\boldsymbol{R}\right) = \sigma(\boldsymbol{h}\boldsymbol{R}) \in \mathbb{R}^M, \tag{12}$$

where $\boldsymbol{R} \in \mathbb{R}^{d\times M}$ is a random matrix, $M > d$, and $\sigma(\cdot)$ is an element-wise activation function. The expanded feature $\boldsymbol{\varphi}(\boldsymbol{x})$ is used for instance-level expert routing rather than final prediction, preserving the flexibility of downstream adaptive experts.

During online training, for each incoming batch $\mathcal{B}_i$ from session t, we compute the expanded feature matrix $\boldsymbol{\Phi}_i \in \mathbb{R}^{B\times M}$ and update two statistics:

$$\boldsymbol{G} \leftarrow \boldsymbol{G} + \boldsymbol{\Phi}_i^\top \boldsymbol{\Phi}_i, \quad \boldsymbol{Q} \leftarrow \boldsymbol{Q} + \boldsymbol{\Phi}_i^\top \boldsymbol{C}_t, \tag{13}$$

where $\boldsymbol{G} \in \mathbb{R}^{M\times M}$ captures second-order feature correlations, $\boldsymbol{Q} \in \mathbb{R}^{M\times K}$ stores expert-wise feature statistics, and $\boldsymbol{C}_t \in \mathbb{R}^{B\times K}$ denotes the expert assignment target for the current session. The router matrix $\boldsymbol{U} \in \mathbb{R}^{K\times M}$ is obtained by the closed-form ridge solution

$$\widehat{\boldsymbol{U}}^\top = (\boldsymbol{G} + \lambda \boldsymbol{I})^{-1}\boldsymbol{Q}, \tag{14}$$

where $\lambda > 0$ is the regularization parameter. At inference time, the routing score and selected expert are computed as

$$\boldsymbol{s}(\boldsymbol{x}) = \boldsymbol{\varphi}(\boldsymbol{x})\widehat{\boldsymbol{U}}^\top, \quad \hat{E}(\boldsymbol{x}) = \arg\max_{k\le K} s_k(\boldsymbol{x}). \tag{15}$$

The routed prediction is

$$F_{\text{route}}(\boldsymbol{x}) = F_{\boldsymbol{\theta},\boldsymbol{\omega}_{\hat{E}},\boldsymbol{\psi}_{\hat{E}}}(\boldsymbol{x}). \tag{16}$$

At training time, the selected expert is updated using the task-specific learning signal. For a sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ assigned to expert $\hat{E}$, the online objective is

$$\mathcal{L}_i = \ell_i \left(F_{\boldsymbol{\theta},\boldsymbol{\omega}_{\hat{E}},\boldsymbol{\psi}_{\hat{E}}}(\boldsymbol{x}_i), \boldsymbol{y}_i \right), \tag{17}$$

where ℓ_i can be instantiated as a classification, contrastive, regression, ranking, verification, imitation, or policy-learning loss. Because the router is updated through accumulated statistics and solved in closed form, it avoids iterative router training and is well suited to the single-pass dynamic data streams.

Temporal Ensemble-based Experts. The prediction of a routed expert depends not only on its learned representation, but also on the stability of its output head as the data distribution evolves. FlyGCL therefore equips each expert with multiple output heads operating at different effective timescales. For expert E_k that accumulates spatially differentiated knowledge, we maintain an online head $\boldsymbol{\psi}_k^{(0)}$ and n shadow heads $\{\boldsymbol{\psi}_k^{(j)}\}_{j=1}^{n}$ updated with different exponential moving average (EMA) rates. For a linear output head $\boldsymbol{\psi} = (\boldsymbol{W}, \boldsymbol{b})$, the j-th EMA head of expert E_k is updated as

$$\boldsymbol{W}_k^{(j)} \leftarrow \alpha_j \boldsymbol{W}_k^{(j)} + (1-\alpha_j)\boldsymbol{W}, \quad \boldsymbol{b}_k^{(j)} \leftarrow \alpha_j \boldsymbol{b}_k^{(j)} + (1-\alpha_j)\boldsymbol{b}. \tag{18}$$

Different EMA rates α_j induce distinct effective memory timescales: faster-updating heads prioritize recent observations, resembling short-term memory in the γ lobe of *Drosophila*, whereas progressively slower heads integrate information over longer timescales, yielding more stable decision boundaries under distribution shift and paralleling the more persistent memory supported by the α'/β' and α/β lobes.

At inference, after the analytic router selects expert $\hat{E}$, FlyGCL computes predictions from the online and EMA heads of this expert and aggregates them:

$$F_{\text{FlyGCL}}(\boldsymbol{x}) = \mathcal{A}\left(F_{\boldsymbol{\theta},\boldsymbol{\omega}_{\hat{E}},\boldsymbol{\psi}_{\hat{E}}^{(0)}}(\boldsymbol{x}), F_{\boldsymbol{\theta},\boldsymbol{\omega}_{\hat{E}},\boldsymbol{\psi}_{\hat{E}}^{(1)}}(\boldsymbol{x}), \ldots, F_{\boldsymbol{\theta},\boldsymbol{\omega}_{\hat{E}},\boldsymbol{\psi}_{\hat{E}}^{(n)}}(\boldsymbol{x}) \right), \tag{19}$$

where $\mathcal{A}(\cdot)$ combines the head predictions, using normalized softmax weights when weighted temporal aggregation is adopted. This temporal ensemble improves decoding robustness without removing the specialization created by expert routing.

For visual recognition settings, we further introduce a lightweight gate to calibrate the integrated temporal output. We construct an analytic class head from the frozen pretrained representation reusing the same fixed random expansion and ridge solution described above, obtaining class evidence $\boldsymbol{z}_{\text{an}}(\boldsymbol{x})$. We calibrate the temporally aggregated class probabilities $\boldsymbol{p}_{\text{temp}}(\boldsymbol{x}) = F_{\text{FlyGCL}}(\boldsymbol{x})$ according to

$$z_{\text{gate},c}(\boldsymbol{x}) = \log p_{\text{temp},c}(\boldsymbol{x}) + \lambda_t z_{\text{an},c}(\boldsymbol{x}), \qquad \forall c \in \mathcal{C}(\boldsymbol{x}), \tag{20}$$

where $\lambda_t = \lambda_{\max}(t/T)^2$, t/T denotes the normalized progress through the continual stream, $\lambda_{\max} \geq 0$ is the final calibration strength, and $\mathcal{C}(\boldsymbol{x})$ denotes the category set. The analytic evidence enhances classes supported by the stable pretrained representation and suppresses unsupported alternatives, without selecting experts or adding another temporal head. This class-dependent regulation may functionally resemble the modulatory role of dopamine neurons (DANs) over mushroom-body output pathways [28, 29]. The calibration is applied with temporal integration, while expert selection remains separately determined by the analytic router.

4.4 Experimental Setups

Datasets. We evaluate FlyGCL across four CL scenarios: visual recognition, vision-language learning, ego-exo video understanding, and embodied vision-language-action learning. For visual recognition, we use CIFAR-100 [49], containing 60,000 images from 100 classes (50,000/10,000 training/test images); ImageNet-R [50], containing 30,000 artistic and non-photographic renditions from 200 ImageNet classes; and CUB-200 [51], containing 11,788 images from 200 fine-grained bird species. For vision-language learning, we construct CLIP-based continual benchmarks on CIFAR-100 and ImageNet-R using the same visual streams, with each class represented by the textual prompt `a photo of a {class name}`. For ego-exo video understanding, we use EgoExoLearn [66], which contains 120 hours of egocentric execution and exocentric demonstration videos with gaze and multimodal annotations for cross-view association, planning, and skill assessment, and EgoExo-Fitness [68], which provides synchronized ego-exo fitness videos with two-level temporal boundaries and interpretable action-judgement annotations. For embodied vision-language-action learning, we use LIBERO [69], a benchmark of language-conditioned robotic manipulation comprising LIBERO-Spatial, -Object, -Goal, and -Long. The first three suites each contain 10 tasks emphasizing spatial relations, object-centric manipulation, and goal-conditioned behaviour, respectively, while LIBERO-Long contains 10 long-horizon tasks derived from LIBERO-100. We adopt $r_D = 50\%$ and $r_B = 30\%$ across all scenarios (except $r_B = 10\%$ for visual recognition following MVP [21] and MISA [48]). Sessions are constructed using class-, action-semantic-, or task-level partitions, with category-based partitioning used for the single-category EgoExo-Fitness benchmark. Detailed benchmark setup and implementations are provided in Supplementary Sec. B.

Evaluation Metrics. We report two commonly used CL metrics across all benchmarks: final average performance A_{last} and average anytime performance A_{auc}. Let $R_{t,j}$ denote the task-specific performance on session j after the model has learned through session t. Depending on the benchmark, $R_{t,j}$ is instantiated as accuracy, ranking accuracy, Top-5 recall, etc. The average anytime performance is defined as

$$A_{\text{auc}} = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{t}\sum_{j=1}^{t} R_{t,j}\right), \tag{21}$$

which measures performance throughout CL. The final average performance is defined as

$$A_{\text{last}} = \frac{1}{T}\sum_{j=1}^{T} R_{T,j}, \tag{22}$$

which measures performance retained after learning the entire stream. Benchmark-specific metric instantiations and notation are detailed in Supplementary Sec. C.

Baseline Methods. We compare FlyGCL with representative CL methods across four application scenarios, with sequential fine-tuning (SeqFT) included as a common baseline throughout. For continual image recognition, we additionally consider the regularization-based methods EWC [17] and LwF [63], the parameter efficiently tuning methods L2P [52] and DualPrompt [53], and the online CL methods MVP [21] and MISA [48]. For continual vision-language learning with CLIP-based models, we compare against EWC, LwF, L2P, DualPrompt, CLAP4CLIP [64], and MG-CLIP [65]. For continual ego-exo video understanding, we include EWC, LwF, L2P+, DualPrompt+, S-Prompt+ [76], and the replay-based methods Experience Replay (ER) [77] and DER++ [18]. For these adapter-based variants, we replace the original prompt modules with adapters, with "+" indicating this modification. For continual embodied vision-language-action learning, we compare against EWC, LwF, L2P+, DualPrompt+, ER, and PackNet [70]. Collectively, these baselines cover sequential fine-tuning, regularization-based, replay-based, and task-specific CL methods.

Implementation Details. We implement FlyGCL on pretrained backbones adopted by each benchmark to ensure fair comparison with existing CL methods. For visual recognition, we use Vision Transformer (ViT-B/16) backbones with ImageNet-based pretraining, including supervised ImageNet-21K pretraining (Sup-21K), ImageNet-21K pretraining followed by ImageNet-1K fine-tuning (Sup-21K/1K), and self-supervised iBOT pretraining on ImageNet-21K (iBOT-21K). For vision-language learning, we use the pretrained OpenAI CLIP model with a ViT-B/16 image encoder. For ego-exo video understanding and embodied vision-language-action learning, we follow the backbone, input preprocessing, and evaluation pipeline of the corresponding benchmarks, replacing only the continual adaptation component across methods. All methods use the same data stream, session order, and online update budget. For prompt-, adapter-, and LoRA-based methods, we match the number of trainable modules to FlyGCL whenever applicable, so that comparisons primarily reflect the continual coordination strategy rather than model capacity. Hyperparameters are selected on the validation split of the first stream setting and then kept fixed across sessions. We report the mean and standard error over multiple random seeds. Detailed optimizer settings, learning rates, batch sizes, training budgets, and benchmark-specific implementations are provided in Supplementary Sec. D.

Data Availability

All benchmark datasets used in this paper are publicly available from their original sources. For continual visual recognition, we use CIFAR-100 (https://www.cs.toronto.edu/~kriz/cifar.html), ImageNet-R (https://github.com/hendrycks/imagenet-r), and CUB-200 (https://www.vision.caltech.edu/datasets/cub_

200_2011/). For continual ego-exo video understanding, we use EgoExoLearn (https://github.com/OpenGVLab/EgoExoLearn) and EgoExo-Fitness (https://github.com/iSEE-Laboratory/EgoExo-Fitness). For continual vision-language-action learning, we use the LIBERO benchmark suite (https://github.com/Lifelong-Robot-Learning/LIBERO), including LIBERO-Long, -Spatial, -Object and -Goal. The connectome data used to parameterize the olfactory model were obtained from FlyWire [78] (https://codex.flywire.ai). The synthetic odor streams used for the biologically grounded analysis are generated by the released code. The EgoExoLearn and EgoExo-Fitness videos were collected and released by their original creators. No human data were collected in this study.

Code Availability

The implementation code, configuration files, and evaluation protocols are available at https://github.com/THU-NeuroML/FlyGCL under the MIT License.

Acknowledgments

This work was supported by the NSFC Project (No. T2622023, No. 62406160 and No. 62595773), the Beijing Natural Science Foundation (No. L247011), the Beijing Nova Program (No. 202604841279), and the Beijing Major Science and Technology Project (No. Z251100008425003).

Author Contributions Statement

H.Y., K.Z. and L.W. conceived the project. H.Y., K.Z. and L.W. designed the computational framework. H.Y. and K.Z. performed the main experiments, assisted by Q.C. and W.D. H.Y., J.Z., G.S., Q.L., Y.Z., and L.W. contributed to the biological motivation and interpretation. H.Y., K.Z. and L.W. analyzed the results. H.Y., K.Z., and L.W. wrote the paper. All authors discussed the results and revised the manuscript. L.W. supervised the project.

Competing Interests Statement

The authors declare no competing interests.

References

[1] 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 **46**(8), 5362–5383 (2024)

[2] De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(7), 3366–3385 (2021)

[3] Davis, R.L.: Learning and memory using drosophila melanogaster: a focus on advances made in the fifth decade of research. Genetics **224**(4), 085 (2023)

[4] Li, F., Lindsey, J.W., Marin, E.C., Otto, N., Dreher, M., Dempsey, G., Stark, I., Bates, A.S., Pleijzier, M.W., Schlegel, P., *et al.*: The connectome of the adult drosophila mushroom body provides insights into function. eLife **9**, 62576 (2020)

[5] McClelland, J.L., McNaughton, B.L., O'Reilly, R.C.: Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. Psychological Review **102**(3), 419 (1995)

[6] Wang, L., Zhang, M., Jia, Z., Li, Q., Bao, C., Ma, K., Zhu, J., Zhong, Y.: Afec: Active forgetting of negative transfer in continual learning. In: Advances in Neural Information Processing Systems, vol. 34 (2021)

[7] Dohare, S., Hernandez-Garcia, J.F., Lan, Q., Rahman, P., Mahmood, A.R., Sutton, R.S.: Loss of plasticity in deep continual learning. Nature **632**(8026), 768–774 (2024)

[8] Silver, D., Sutton, R.S.: Welcome to the era of experience. In: Designing an Intelligence. MIT Press (2025)

[9] LeCun, Y.: A Path Towards Autonomous Machine Intelligence. OpenReview. Version 0.9.2 (2022). https://openreview.net/forum?id=BZ5a1r-kVsf

[10] Hughes, E., Dennis, M.D., Parker-Holder, J., Behbahani, F., Mavalankar, A., Shi, Y., Schaul, T., Rocktäschel, T.: Position: Open-endedness is essential for artificial superhuman intelligence. In: International Conference on Machine Learning, vol. 235, pp. 20597–20616 (2024)

[11] Zhang, J., Hu, S., Lu, C., Lange, R.T., Clune, J.: Darwin gödel machine: Open-ended evolution of self-improving agents. In: International Conference on Learning Representations (2026)

[12] Zweiger, A., Pari, J., Guo, H., Kim, Y., Agrawal, P.: Self-adapting language models. Advances in Neural Information Processing Systems **38**, 74084–74115 (2026)

[13] Behrouz, A., Zhong, P., Mirrokni, V.: Titans: Learning to memorize at test time. Advances in Neural Information Processing Systems **38**, 113506–113543 (2026)

[14] Zhou, K., Hao, Z., Wang, L., Liang, X.: Adaptive score alignment learning for continual perceptual quality assessment of 360-degree videos in virtual reality. IEEE Transactions on Visualization and Computer Graphics **31**(5), 2880–2890 (2025)

[15] Wang, L., Xie, J., Zhang, X., Su, H., Zhu, J.: Hide-pet: continual learning via hierarchical decomposition of parameter-efficient tuning. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(8), 6687–6702 (2025)

[16] Wang, L., Zhang, X., Li, Q., Zhu, J., Zhong, Y.: Coscl: Cooperation of small continual learners is stronger than a big one. In: Proceedings of the European Conference on Computer Vision, pp. 254–271 (2022)

[17] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114**(13), 3521–3526 (2017)

[18] Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark experience for general continual learning: a strong, simple baseline. Advances in Neural Information Processing Systems **33**, 15920–15930 (2020)

[19] Zhou, K., Wang, L., Zhang, X., Shum, H.P., Li, F.W., Li, J., Liang, X.: Magr: Manifold-aligned graph regularization for continual action quality assessment. In: European Conference on Computer Vision, vol. 15069, pp. 375–392 (2024)

[20] Wang, L., Xie, J., Zhang, X., Huang, M., Su, H., Zhu, J.: Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. Advances in Neural Information Processing Systems **36**, 69054–69076 (2023)

[21] Moon, J.-Y., Park, K.-H., Kim, J.U., Park, G.-M.: Online class incremental learning on stochastic blurry task boundary via mask and visual prompt tuning. In: IEEE/CVF International Conference on Computer Vision, pp. 11731–11741 (2023)

[22] Modi, M.N., Shuai, Y., Turner, G.C.: The drosophila mushroom body: From architecture to algorithm in a learning circuit. Annual Review of Neuroscience **43**, 465–484 (2020)

[23] Winding, M., Pedigo, B.D., Barnes, C.L., Patsolic, H.G., Park, Y., Kazimiers, T., Fushiki, A., Andrade, I.V., Khandelwal, A., Valdes-Aleman, J., *et al.*: The connectome of an insect brain. Science **379**(6636), 9330 (2023)

[24] Lin, A., Yang, R., Dorkenwald, S., Matsliah, A., Sterling, A.R., Schlegel, P., Yu, S.-c., McKellar, C.E., Costa, M., Eichler, K., *et al.*: Network statistics of the whole-brain connectome of drosophila. Nature **634**(8032), 153–165 (2024)

[25] Schlegel, P., Yin, Y., Bates, A.S., Dorkenwald, S., Eichler, K., Brooks, P., Han, D.S., Gkantia, M., Dos Santos, M., Munnelly, E.J., *et al.*: Whole-brain annotation and multi-connectome cell typing of drosophila. Nature **634**(8032), 139–152 (2024)

[26] Caron, S.J., Ruta, V., Abbott, L.F., Axel, R.: Random convergence of olfactory inputs in the drosophila mushroom body. Nature **497**(7447), 113–117 (2013)

[27] Honegger, K.S., Campbell, R.A., Turner, G.C.: Cellular-resolution population imaging reveals robust sparse coding in the drosophila mushroom body. Journal of Neuroscience **31**(33), 11772–11785 (2011)

[28] Aso, Y., Sitaraman, D., Ichinose, T., Kaun, K.R., Vogt, K., Belliart-Guérin, G., Plaçais, P.-Y., Robie, A.A., Yamagata, N., Schnaitmann, C., *et al.*: Mushroom body output neurons encode valence and guide memory-based action selection in drosophila. eLife **3**, 04580 (2014)

[29] Dasgupta, S., Stevens, C.F., Navlakha, S.: A neural algorithm for a fundamental computing problem. Science **358**(6364), 793–796 (2017)

[30] Aso, Y., Hattori, D., Yu, Y., Johnston, R.M., Iyer, N.A., Ngo, T.-T., Dionne, H., Abbott, L., Axel, R., Tanimoto, H., *et al.*: The neuronal architecture of the mushroom body provides a logic for associative learning. eLife **3**, 04577 (2014)

[31] Aso, Y., Rubin, G.M.: Dopaminergic neurons write and update memories with cell-type-specific rules. elife **5**, 16135 (2016)

[32] Cohn, R., Morantte, I., Ruta, V.: Coordinated and compartmentalized neuromodulation shapes sensory processing in drosophila. Cell **163**(7), 1742–1755 (2015)

[33] Handler, A., Graham, T.G., Cohn, R., Morantte, I., Siliciano, A.F., Zeng, J., Li, Y., Ruta, V.: Distinct dopamine receptor pathways underlie the temporal sensitivity of associative learning. Cell **178**(1), 60–75 (2019)

[34] Cervantes-Sandoval, I., Martin-Peña, A., Berry, J.A., Davis, R.L.: System-like consolidation of olfactory memories in drosophila. Journal of Neuroscience **33**(23), 9846–9854 (2013)

[35] Mu, S., Lin, S.: A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. arXiv preprint arXiv:2503.07137 (2025)

[36] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation **3**(1), 79–87 (1991)

[37] Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q.: A survey on ensemble learning. Frontiers of Computer Science **14**(2), 241–258 (2020)

[38] Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(10), 993–1001 (2002)

[39] Wang, L., Zhang, X., Li, Q., Zhang, M., Su, H., Zhu, J., Zhong, Y.: Incorporating neuro-inspired adaptability for continual learning in artificial intelligence. Nature Machine Intelligence **5**(12), 1356–1368 (2023)

[40] Wang, L., Li, Q.: Convergent multi-modular architecturefor adaptive learning in drosophila and artificial intelligence. iScience **28**(11) (2025)

[41] Wang, P.Y., Sun, Y., Axel, R., Abbott, L., Yang, G.R.: Evolving the olfactory system with machine learning. Neuron **109**(23), 3879–3892 (2021)

[42] Fulton, K.A., Zimmerman, D., Samuel, A., Vogt, K., Datta, S.R.: Common principles for odour coding across vertebrates and invertebrates. Nature Reviews Neuroscience **25**(7), 453–472 (2024)

[43] Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021)

[44] Rebuffi, S.-A., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. NeurIPS **30** (2017)

[45] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)

[46] Shen, Y., Dasgupta, S., Navlakha, S.: Reducing catastrophic forgetting with associative learning: a lesson from fruit flies. Neural Computation **35**(11), 1797–1819 (2023)

[47] Dorkenwald, S., McKellar, C.E., Macrina, T., Kemnitz, N., Lee, K., Lu, R., Wu, J., Popovych, S., Mitchell, E., Nehoran, B., *et al.*: Flywire: online community for whole-brain connectomics. Nature Methods **19**(1), 119–128 (2022)

[48] Kang, Z., Wang, L., Zhang, X., Alahari, K.: Advancing prompt-based methods for replay-independent general continual learning. In: International Conference on Learning Representations (2025)

[49] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Technical report, Citeseer (2009)

[50] Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., *et al.*: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: IEEE/CVF International Conference on Computer Vision, pp. 8340–8349 (2021)

[51] Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology (2011)

[52] 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: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 139–149 (2022)

[53] Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.-Y., Ren, X., Su, G., Perot, V., Dy, J., *et al.*: Dualprompt: Complementary prompting for rehearsal-free continual learning. In: European Conference on Computer Vision, pp. 631–648 (2022)

[54] Smith, J.S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., Kira, Z.: Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11909–11919 (2023)

[55] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., *et al.*: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)

[56] Ridnik, T., Ben-Baruch, E., Noy, A., Zelnik-Manor, L.: Imagenet-21k pretraining for the masses. arXiv preprint arXiv:2104.10972 (2021)

[57] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

[58] Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: Image bert pre-training with online tokenizer. In: International Conference on Learning Representations (2021)

[59] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: IEEE/CVF International Conference on Computer Vision, pp. 9650–9660 (2021)

[60] Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: IEEE/CVF International Conference on Computer Vision, pp. 9640–9649 (2021)

[61] Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)

[62] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763 (2021)

[63] Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(12), 2935–2947 (2017)

[64] Jha, S., Gong, D., Yao, L.: Clap4clip: Continual learning with probabilistic finetuning for vision-language models. Advances in Neural Information Processing Systems **37**, 129146–129186 (2024)

[65] Huang, L., Cao, X., Lu, H., Meng, Y., Yang, F., Liu, X.: Mind the gap: Preserving and compensating for the modality gap in clip-based continual learning. In: IEEE/CVF International Conference on Computer Vision, pp. 3777–3786 (2025)

[66] Huang, Y., Chen, G., Xu, J., Zhang, M., Yang, L., Pei, B., Zhang, H., Dong, L., Wang, Y., Wang, L., *et al.*: Egoexolearn: A dataset for bridging asynchronous ego-and exo-centric view of procedural activities in real world. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22072–22086 (2024)

[67] Yan, H., Zhou, K., Liu, Y., Shi, Q., Zhong, Y., Wang, L.: Ce^{4}l: Continual ego, exo, and ego-exo learning. In: International Conference on Machine Learning (2026)

[68] Li, Y.-M., Huang, W.-J., Wang, A.-L., Zeng, L.-A., Meng, J.-K., Zheng, W.-S.: Egoexo-fitness: Towards egocentric and exocentric full-body action understanding. In: European Conference on Computer Vision, pp. 363–382 (2024)

[69] Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems **36**, 44776–44791 (2023)

[70] Mallya, A., Lazebnik, S.: Packnet: Adding multiple tasks to a single network by iterative pruning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 7765–7773 (2018)

[71] Yan, H., Sun, G., Zhou, K., Li, Q., Wang, L., Zhong, Y.: Flyprompt: Brain-inspired random-expanded routing with temporal-ensemble experts for general continual learning. In: International Conference on Learning Representations (2026)

[72] Zador, A.M., Escola, S., Richards, B., Ölveczky, B., Bengio, Y., Boahen, K., Botvinick, M., Chklovskii, D., Churchland, A., Clopath, C., DiCarlo, J.J., Ganguli, S., Hawkins, J., Kording, K., Koulakov, A., LeCun, Y., Lillicrap, T., Marblestone, A., Olshausen, B.A., Pouget, A., Savin, C., Sejnowski, T.J., Simoncelli, E., Solla, S.A., Sussillo, D., Tolias, A.S., Tsao, D.: Catalyzing next-generation artificial intelligence through neuroai. Nature Communications **14**, 1597 (2023)

[73] Koh, H., Kim, D., Ha, J.-W., Choi, J.: Online continual learning on class incremental blurry task configuration with anytime inference. arXiv preprint arXiv:2110.10031 (2021)

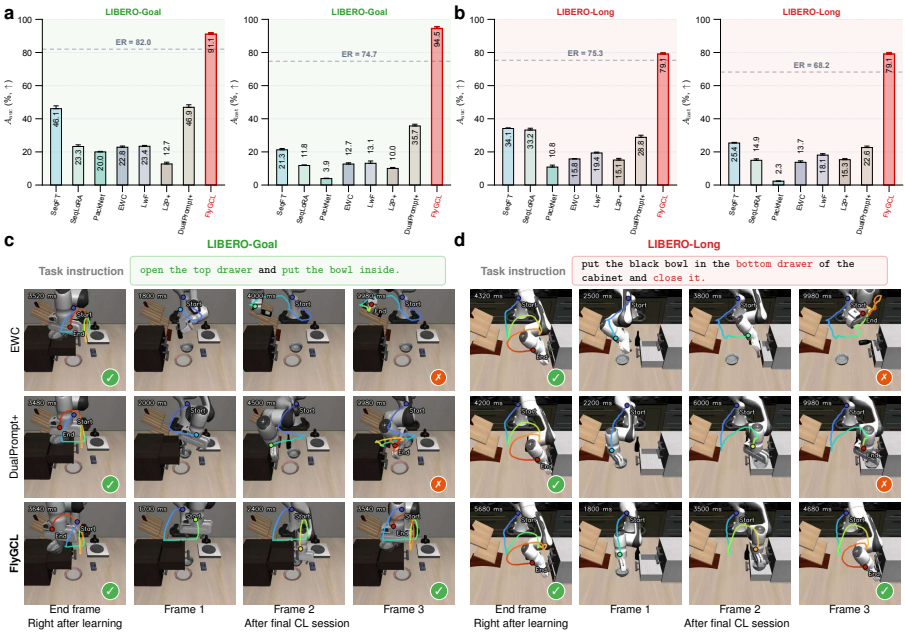
[74] Allen-Zhu, Z., Li, Y.: Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. In: International Conference on Learning Representations (2023)

[75] Chen, Z., Deng, Y., Wu, Y., Gu, Q., Li, Y.: Towards understanding the mixture-of-experts layer in deep learning. Advances in Neural Information Processing Systems **35**, 23049–23062 (2022)

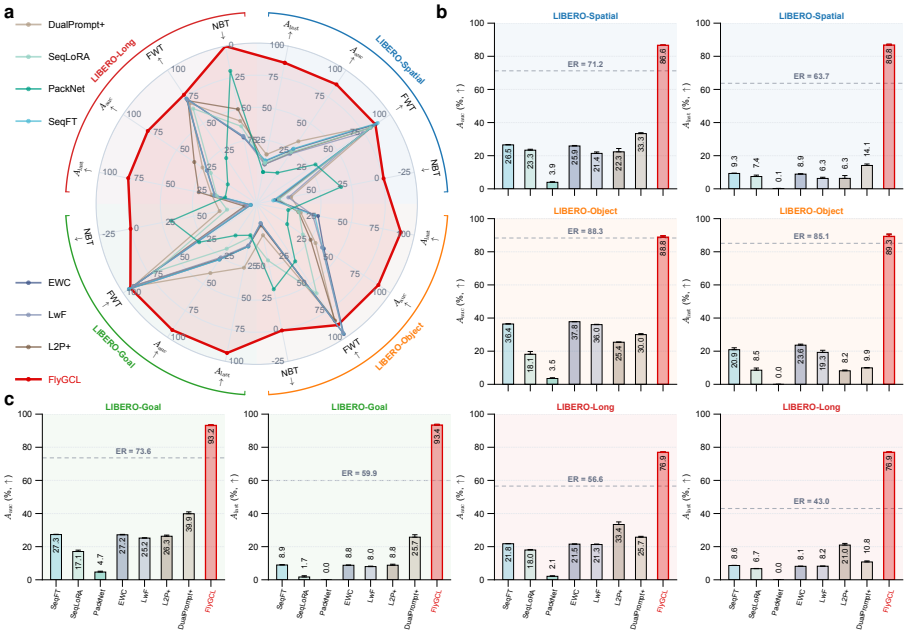
[76] Wang, Y., Huang, Z., Hong, X.: S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. Advances in Neural Information Processing Systems **35**, 5682–5695 (2022)

[77] Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T.P., Wayne, G.: Experience replay for continual learning. In: Advances in Neural Information Processing Systems, vol. 32 (2019)

[78] Dorkenwald, S., Matsliah, A., Sterling, A.R., Schlegel, P., Yu, S.-c., McKellar, C.E., Lin, A., Costa, M., Eichler, K., Yin, Y., *et al.*: Neuronal wiring diagram of an adult brain. Nature **634**(8032), 124–138 (2024)



Extended Data Fig. 1: **Extended results on continual vision-language-action benchmarks.** a: Performance comparison with state-of-the-art baselines on LIBERO-Goal using A_{auc} and A_{last}. b: Performance comparison with state-of-the-art baselines on LIBERO-Long using A_{auc} and A_{last}. c: Qualitative rollouts on LIBERO-Goal, comparing task execution immediately after learning and after the final CL session for EWC, DualPrompt+, and FlyGCL. d: Qualitative rollouts on LIBERO-Long under the same protocol. Results are averaged over three independent runs. Error bars denote the standard error of the mean.



Extended Data Fig. 2: **Offline results on continual vision-language-action benchmarks.** a: Unified performance summary across LIBERO-Long, LIBERO-Spatial, LIBERO-Goal, and LIBERO-Object under multiple continual learning metrics. b: Performance comparison with state-of-the-art baselines on LIBERO-Spatial and LIBERO-Object using A_{auc} and A_{last}. c: Performance comparison with state-of-the-art baselines on LIBERO-Goal and LIBERO-Long using A_{auc} and A_{last}. Results are averaged over three independent runs. Error bars denote the standard error of the mean.

A Proofs

A.1 Proof of Theorem 1

Proof Let $F_{\theta,\Omega,\Psi}$ be the stream-level modular predictor and let $F^\star$ be the ideal stream-level predictor. For brevity, we write F for $F_{\theta,\Omega,\Psi}$ in this proof. By definition, the expected GCL risk over the evolving stream is

$$\mathcal{R}_{\mathrm{GCL}}(F) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{(\mathbf{o},\mathbf{u})\sim P_t}\left[\ell(F(\mathbf{o}),\mathbf{u})\right]. \tag{S23}$$

The empirical fitting risk on the observed stream $\mathcal{S} = \{\mathcal{D}_1,\ldots,\mathcal{D}_T\}$ is

$$\mathcal{R}_{\mathrm{fit}}(F) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t} \ell(F(\mathbf{o}_{t,i}),\mathbf{u}_{t,i}). \tag{S24}$$

The gap between the expected stream risk and the empirical fitting risk can be written as

$$\mathcal{R}_{\mathrm{GCL}}(F) - \mathcal{R}_{\mathrm{fit}}(F). \tag{S25}$$

Under hierarchical modular learning, this residual gap has three sources. First, if conflicting local distributions are assigned to insufficiently separated adaptive modules, their updates and predictions interfere with each other. We denote the corresponding excess risk by $\mathcal{E}_{\mathrm{sep}}(F)$. Second, if compatible local distributions or predictors are treated in isolation, the model loses potential transfer and suffers from larger estimation variance. We denote this excess risk by $\mathcal{E}_{\mathrm{int}}(F)$. Third, even if routing and integration are individually useful, their combination may be imperfect. The positive excess loss of the hierarchical modular predictor over the ideal stream-level predictor is

$$\mathcal{C}_{\mathrm{coord}}(F) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{(\mathbf{o},\mathbf{u})\sim P_t}\left[\ell(F(\mathbf{o}),\mathbf{u}) - \ell(F^\star(\mathbf{o}),\mathbf{u})\right]_+. \tag{S26}$$

By the assumed additive excess-risk decomposition, the residual gap in Eq. (S25) is upper bounded by the sum of these three non-negative components:

$$\mathcal{R}_{\mathrm{GCL}}(F) - \mathcal{R}_{\mathrm{fit}}(F) \lesssim \mathcal{E}_{\mathrm{sep}}(F) + \mathcal{E}_{\mathrm{int}}(F) + \mathcal{C}_{\mathrm{coord}}(F). \tag{S27}$$

Adding $\mathcal{R}_{\mathrm{fit}}(F)$ to both sides gives

$$\mathcal{R}_{\mathrm{GCL}}(F) \lesssim \mathcal{R}_{\mathrm{fit}}(F) + \mathcal{E}_{\mathrm{sep}}(F) + \mathcal{E}_{\mathrm{int}}(F) + \mathcal{C}_{\mathrm{coord}}(F). \tag{S28}$$

Substituting back $F = F_{\theta,\Omega,\Psi}$ yields Eq. (4). This completes the proof. □

A.2 Proof of Proposition 1

Proof For brevity, we write F_{hier} for $F_{\theta,\Omega,\Psi}$. From Theorem 1, the expected GCL risk of a hierarchical modular predictor satisfies

$$\mathcal{R}_{\mathrm{GCL}}(F_{\mathrm{hier}}) \lesssim \mathcal{R}_{\mathrm{fit}}(F_{\mathrm{hier}}) + \mathcal{E}_{\mathrm{sep}}(F_{\mathrm{hier}}) + \mathcal{E}_{\mathrm{int}}(F_{\mathrm{hier}}) + \mathcal{C}_{\mathrm{coord}}(F_{\mathrm{hier}}). \tag{S29}$$

By the definition of the routing gain in Eq. (7), we have

$$\Delta_{\mathrm{route}} = \mathcal{E}_{\mathrm{sep}}(F_{\theta,\psi}) - \mathcal{E}_{\mathrm{sep}}(F_{\mathrm{MoE}}), \tag{S30}$$

which gives

$$\mathcal{E}_{\mathrm{sep}}(F_{\mathrm{MoE}}) = \mathcal{E}_{\mathrm{sep}}(F_{\theta,\psi}) - \Delta_{\mathrm{route}}. \tag{S31}$$

Since the hierarchical predictor uses routing as its separation mechanism, its separation error is upper bounded by that of the routed modular predictor, i.e.,

$$\mathcal{E}_{\text{sep}}(F_{\text{hier}}) \leq \mathcal{E}_{\text{sep}}(F_{\text{MoE}}). \tag{S32}$$

Combining Eqs. (S31) and (S32), we obtain

$$\mathcal{E}_{\text{sep}}(F_{\text{hier}}) \leq \mathcal{E}_{\text{sep}}(F_{\theta,\psi}) - \Delta_{\text{route}}. \tag{S33}$$

Similarly, by the definition of the ensemble gain in Eq. (7), we have

$$\Delta_{\text{ens}} = \mathcal{E}_{\text{int}}(F_{\text{single}}) - \mathcal{E}_{\text{int}}(F_{\text{EL}}), \tag{S34}$$

which gives

$$\mathcal{E}_{\text{int}}(F_{\text{EL}}) = \mathcal{E}_{\text{int}}(F_{\text{single}}) - \Delta_{\text{ens}}. \tag{S35}$$

Since the hierarchical predictor uses aggregation as its integration mechanism, its integration error is upper bounded by that of the ensemble predictor, i.e.,

$$\mathcal{E}_{\text{int}}(F_{\text{hier}}) \leq \mathcal{E}_{\text{int}}(F_{\text{EL}}). \tag{S36}$$

Combining Eqs. (S35) and (S36), we obtain

$$\mathcal{E}_{\text{int}}(F_{\text{hier}}) \leq \mathcal{E}_{\text{int}}(F_{\text{single}}) - \Delta_{\text{ens}}. \tag{S37}$$

Substituting Eqs. (S33) and (S37) into Eq. (S29) yields

$$\begin{aligned}\mathcal{R}_{\text{GCL}}(F_{\text{hier}}) \lesssim & \mathcal{R}_{\text{fit}}(F_{\text{hier}}) + \mathcal{E}_{\text{sep}}(F_{\theta,\psi}) + \mathcal{E}_{\text{int}}(F_{\text{single}}) \\ & - \Delta_{\text{route}} - \Delta_{\text{ens}} + \mathcal{C}_{\text{coord}}(F_{\text{hier}}).\end{aligned} \tag{S38}$$

Substituting back $F_{\text{hier}} = F_{\theta,\Omega,\Psi}$ gives Eq. (8).

Finally, comparing this bound with the corresponding bound without routing and integration gains,

$$\mathcal{R}_{\text{fit}}(F_{\theta,\Omega,\Psi}) + \mathcal{E}_{\text{sep}}(F_{\theta,\psi}) + \mathcal{E}_{\text{int}}(F_{\text{single}}), \tag{S39}$$

shows that hierarchical modular learning improves the bound whenever

$$\Delta_{\text{route}} + \Delta_{\text{ens}} > \mathcal{C}_{\text{coord}}(F_{\theta,\Omega,\Psi}). \tag{S40}$$

This proves Eq. (9) and completes the proof. □

A.3 Proof of Proposition 2

Proof For brevity, we write F_{hier} for $F_{\theta,\Omega,\Psi}$. The coordination cost measures the excess loss introduced when routing and integration do not exactly match the ideal stream-level organization. We decompose this cost into two sources: mismatch due to imperfect routing and mismatch due to imperfect aggregation.

Let ϵ_{route} denote the degree of routing error, i.e., the probability or expected weight that an input is assigned to a suboptimal module. Let $\Delta_{\text{mis}}(f_\theta)$ denote the maximum expected loss increase caused by such a suboptimal assignment under the representation induced by f_θ. Then the routing-induced part of the coordination cost is bounded by

$$\mathcal{C}_{\text{route}}(F_{\text{hier}}) \leq \epsilon_{\text{route}} \Delta_{\text{mis}}(f_\theta). \tag{S41}$$

Let ϵ_{agg} denote the residual mismatch caused by aggregating modular predictions that are not perfectly compatible. Then the total coordination cost is bounded by the sum of the routing-induced mismatch and the aggregation-induced mismatch:

$$\mathcal{C}_{\text{coord}}(F_{\text{hier}}) \leq \mathcal{C}_{\text{route}}(F_{\text{hier}}) + \epsilon_{\text{agg}}. \tag{S42}$$

Combining Eqs. (S41) and (S42), we obtain

$$\mathcal{C}_{\text{coord}}(F_{\text{hier}}) \leq \epsilon_{\text{route}} \Delta_{\text{mis}}(f_\theta) + \epsilon_{\text{agg}}. \tag{S43}$$

Substituting back $F_{\mathrm{hier}} = F_{\theta,\Omega,\Psi}$ gives Eq. (10).

It remains to show why pretrained representations reduce the mismatch penalty. By definition, $\Delta_{\mathrm{mis}}(f_\theta)$ measures the expected prediction penalty when an input is routed to a suboptimal module. If a representation maps related inputs closer together and makes conflicting inputs more separable, then the predictions produced by nearby or partially mismatched modules are less divergent for related inputs, while routing ambiguity is reduced for conflicting inputs. A pretrained backbone provides such a more stable and semantically organized representation than a representation learned from scratch on the limited online stream. Therefore, the mismatch penalty under a pretrained backbone is smaller:

$$\Delta_{\mathrm{mis}}(f_\theta^{\mathrm{pre}}) < \Delta_{\mathrm{mis}}(f_\theta^{\mathrm{scratch}}). \tag{S44}$$

Substituting Eq. (S44) into Eq. (S43) shows that pretraining reduces the upper bound of $\mathcal{C}_{\mathrm{coord}}$, which proves Eq. (11) and completes the proof. □

B Additional Experimental Setups

B.1 Biological Simulation

Odor Classes and Spatial Regions. We generate 100 class prototypes independently and uniformly in $[0, 1)^{50}$. Training and test inputs are sampled from the same space and assigned to their nearest prototype by squared Euclidean distance. The prototypes are then grouped by proximity into five regions of 20 classes using capacity-constrained clustering. For each seed, the formal benchmark contains 10,000 training and 2,000 test inputs from each region. The fixed test set therefore contains 10,000 inputs, while every training stream contains 50,000 inputs.

Continual Odor Streams. The stream represents a learner moving through the five spatial regions and acquiring experience along this trajectory. Each region defines the home stage of its classes. In Disjoint, the 10,000 inputs from each region are shuffled within their home stage and the five stages are presented sequentially. Blurry and Joint soften the boundaries between neighboring periods of experience without altering the inputs or labels. For each home stage, a specified subset of classes remains disjoint, while 10% of the inputs from the other classes is removed, pooled across stages, globally shuffled, and reassigned under equal stage quotas. Blurry retains 10 disjoint classes (50%) in each region, whereas Joint applies this procedure to all classes. In every case, the model encounters data from all five regions in sequence, each input appears exactly once, and the total learning budget is unchanged.

Biologically Grounded Architecture. The sensory network follows the population-level compression and expansion described in biological and task-driven accounts of the *Drosophila* olfactory system [26, 41]. Each of the 50 input channels is replicated across 26 ORNs and pooled by one PN, giving 1,300 ORNs and 50 PNs. During training, independent Gaussian noise with standard deviation 0.1 is added to the ORN responses. Evaluation is noise-free. Within each group, ORN-PN weights are sampled from a truncated log-normal distribution fitted to FlyWire connections between matched olfactory types and normalized to sum to one. PN activity is rectified and projected to 2,000 KCs through a sparse PN-KC connection matrix. We randomly set 91.5% of its entries to zero, sample the remaining 8.5% from a truncated Gaussian distribution fitted to the nonzero FlyWire PN-KC connection weights, and then keep

the resulting matrix fixed. After rectification, only the 100 largest KC responses (top-5% of the 2,000 KCs) are retained for each odor. The sensory pathway remains fixed throughout continual learning, so that all learning occurs in downstream components and the effects of learning and memory organization can be evaluated independently of changes in sensory encoding.

Baseline Variants. MoE uses five independently initialized experts with distinct views of KC activity, representing spatially differentiated memory pathways. During training, we associate each stage with one expert as a simple way to activate experts successively along the stream. This association is specific to the experimental protocol rather than a requirement for known task identities or predefined stage semantics: experts could instead be activated sequentially after a fixed number of observed samples or by another online switching rule. In parallel, the model maintains the mean KC representation of each observed stage. At evaluation, an input is routed to the active expert whose mean representation has the highest cosine similarity to its KC activity. The router is updated only from inputs observed so far and does not use class or region labels at inference.

The Baseline uses one bias-free 100-class readout trained at a learning rate of 10^{-3}. EL replaces it with three independently initialized heads trained at learning rates 10^{-2}, 10^{-3}, and 10^{-4}, representing fast, intermediate, and slow effective memory scales. MoE uses five experts with one head each, and MoE+EL uses three such heads within every expert. Each head is optimized with its own cross-entropy loss, and their class probabilities are averaged within the selected expert at inference. The incremental EL ablation further compares these models with three independently initialized heads all trained at 10^{-3}, separating the effect of ensembling from that of different update rates.

Training and Evaluation. All configurations share the same sensory encoding, observations, supervision, and online learning budget, without replay, so that performance differences primarily reflect the organization of the downstream learning and memory components. Models are optimized online with Adam using a batch size of 64. Each run comprises 50,000 inputs, with performance evaluated every 2,000 inputs at 25 post-update checkpoints. At each checkpoint, the exposed class set contains all classes represented by at least one input observed up to that point, and accuracy is evaluated on the corresponding test examples. Exposed-class anytime AUC is computed as the trapezoidal integral of this checkpoint-wise accuracy over the number of observed inputs, normalized by the evaluated interval. Results are averaged over five independently generated datasets, streams, and model initializations, and reported with 95% confidence intervals.

B.2 Continual Stream Construction

Across benchmarks, sessions are constructed from the semantic unit most closely aligned with the prediction target (Supplementary Tab. S1). Visual recognition and vision-language benchmarks use image classes, whereas ego-exo video tasks use action classes or procedure-level task groups to preserve the temporal and cross-view structure of the original annotations. LIBERO uses individual manipulation tasks as the partition unit. All stream construction is performed within the official data

Table S1: Detailed continual stream construction and evaluation metrics for all benchmarks. Partition Unit specifies the semantic or distributional unit used to construct sessions. Task Metric denotes the benchmark-specific evaluation measure, while Reported CL Metrics denote its continual evaluation over the stream.

Scenario	Benchmark / Task	Partition Unit	Setting	T	r_D / r_B	Task Metric	Reported CL Metrics
Visual recognition	CIFAR-100	Image class	GCL	5	50% / 10%	Accuracy	A_{last}, A_{auc}
	ImageNet-R	Image class	GCL	5	50% / 10%	Accuracy	A_{last}, A_{auc}
	CUB-200	Image class	GCL	5	50% / 10%	Accuracy	A_{last}, A_{auc}
Vision-language learning	CIFAR-100	Image class	GCL	5	50% / 30%	Accuracy	A_{last}, A_{auc}
	ImageNet-R	Image class	GCL	5	50% / 30%	Accuracy	A_{last}, A_{auc}
Ego-exo video understanding	EgoExoLearn: Skill Assessment	Action class	GCL	4	50% / 30%	Ranking accuracy	A_{last}, $\bar{F}_T$, A_{auc}
	EgoExoLearn: Action Anticipation	Procedure task	GCL	8	50% / 30%	Top-5 recall	A_{last}, A_{auc}
	EgoExo-Fitness: Skill Assessment	Action class	GCL	4	50% / 30%	Ranking accuracy	A_{last}, $\bar{F}_T$, A_{auc}
	EgoExo-Fitness: Action Classification	Action class	GCL	4	50% / 30%	Accuracy	A_{last}, $\bar{F}_T$, A_{auc}
	EgoExo-Fitness: Sequence Verification	Action class	GCL	4	50% / 30%	ROC-AUC, mAP	A_{last}, $\bar{F}_T$, A_{auc}
	EgoExo-Fitness: Guidance-based Verification	Action class	GCL	4	50% / 30%	Classification accuracy, F1	A_{last}, $\bar{F}_T$, A_{auc}
Vision-language-action learning	LIBERO-Spatial, -Object, -Goal, -Long	Manipulation task	Offline CL	10	–	Success rate	A_{last}, A_{auc}, FWT, NBT
	LIBERO-Spatial, -Object, -Goal, -Long	Manipulation task	GCL	10	50% / 30%	Success rate	A_{last}, A_{auc}, FWT, NBT

splits, so redistribution across sessions does not transfer samples between training and evaluation sets.

For GCL evaluation, each semantic unit is assigned a home session and the stream is made blurry by allowing samples from the recurring component to appear outside that session according to r_B, while the disjoint component remains primarily session-specific according to r_D. Each training sample is processed under the benchmark-specific online budget, and evaluation after session t covers all sessions exposed up to that point. The complementary offline LIBERO protocol instead presents the ten manipulation tasks sequentially and permits multiple passes over each task before moving to the next, while retaining the same task order and evaluation procedure.

C Additional Evaluation Metrics

This section details the task-specific base metrics and additional CL metrics. Consistent with the notation in the main paper, let $\mathcal{E}_j = \{(\mathbf{o}_{j,n}, \mathbf{u}_{j,n})\}_{n=1}^{N_j}$ denote the evaluation set of session j, where $\mathbf{o}_{j,n}$ is an observation and $\mathbf{u}_{j,n}$ is its associated learning signal. Let $\hat{\mathbf{u}}_{j,n}^{(t)}$ denote the prediction obtained after the model has learned through session t. Each task-specific metric defined below instantiates the session-level performance $R_{t,j}$ used in the CL aggregates in the main paper.

C.1 Task Metrics

Accuracy-Based Metrics. For visual recognition, vision-language classification, and action classification, the learning signal is a categorical label $u_{j,n} \in \mathcal{U}_j$, and we use classification accuracy:

$$\mathrm{Acc}_{t,j} = \frac{1}{N_j} \sum_{n=1}^{N_j} \mathbf{1}\left[\hat{u}_{j,n}^{(t)} = u_{j,n}\right]. \tag{S45}$$

Skill Assessment. For skill assessment, the learning signal specifies the relative quality ordering of two video observations. Let $\mathcal{P}_j$ denote the set of annotated ordered pairs, where $(a, b) \in \mathcal{P}_j$ indicates that $\mathbf{o}_{j,a}$ exhibits higher skill than $\mathbf{o}_{j,b}$. Given the predicted skill scores $\hat{q}_{j,a}^{(t)}$ and $\hat{q}_{j,b}^{(t)}$, ranking accuracy is defined as

$$\mathrm{RAcc}_{t,j} = \frac{1}{|\mathcal{P}_j|} \sum_{(a,b)\in\mathcal{P}_j} \mathbf{1}\left[\hat{q}_{j,a}^{(t)} > \hat{q}_{j,b}^{(t)}\right]. \tag{S46}$$

Action Anticipation. For action anticipation, the learning signal is the future verb or noun class, and we report class-mean Top-K recall with $K = 5$ separately for verb and noun prediction. Let $\mathcal{C}$ denote the evaluated class set, $\mathcal{E}_{j,c} = \{n : u_{j,n} = c\}$ denote the evaluation samples belonging to class c, and $\mathrm{TopK}(\mathbf{s}_{j,n}^{(t)})$ denote the K classes with the largest predicted scores. The metric is defined as

$$\mathrm{R@}K_{t,j} = \frac{1}{|\mathcal{C}|} \sum_{c\in\mathcal{C}} \frac{1}{|\mathcal{E}_{j,c}|} \sum_{n\in\mathcal{E}_{j,c}} \mathbf{1}\left[c \in \mathrm{TopK}\left(\mathbf{s}_{j,n}^{(t)}\right)\right]. \tag{S47}$$

Sequence and Guidance-Based Verification. For binary sequence verification, the learning signal indicates whether an observed execution sequence is valid. We report ROC-AUC and mAP. Let $\mathcal{E}_j^+$ and $\mathcal{E}_j^-$ denote the positive and negative sample sets, respectively, and let $\hat{s}_{j,n}^{(t)}$ be the predicted verification score. ROC-AUC is defined as

$$\text{ROC-AUC}_{t,j} = \frac{1}{|\mathcal{E}_j^+||\mathcal{E}_j^-|} \sum_{p\in\mathcal{E}_j^+} \sum_{n\in\mathcal{E}_j^-} \mathbf{1}\left[\hat{s}_{j,p}^{(t)} > \hat{s}_{j,n}^{(t)}\right]. \tag{S48}$$

Following the standard ranking-based evaluation, mAP is computed as

$$\mathrm{mAP}_{t,j} = \frac{1}{|\mathcal{Q}_j|} \sum_{q\in\mathcal{Q}_j} \mathrm{AP}_q^{(t)}, \tag{S49}$$

where $\mathcal{Q}_j$ denotes the evaluated verification queries or groups. For guidance-based execution verification, we additionally report classification accuracy as defined in Eq. (S45) and F1 score:

$$\mathrm{F1}_{t,j} = \frac{2\,\mathrm{Prec}_{t,j}\,\mathrm{Rec}_{t,j}}{\mathrm{Prec}_{t,j} + \mathrm{Rec}_{t,j}}, \tag{S50}$$

$$\mathrm{Prec}_{t,j} = \frac{\mathrm{TP}_{t,j}}{\mathrm{TP}_{t,j} + \mathrm{FP}_{t,j}}, \tag{S51}$$

$$\mathrm{Rec}_{t,j} = \frac{\mathrm{TP}_{t,j}}{\mathrm{TP}_{t,j} + \mathrm{FN}_{t,j}}. \tag{S52}$$

Embodied Vision-Language-Action. For embodied vision-language-action, the learning signal corresponds to an action or trajectory and the evaluation target is

successful task completion. Let $z_{j,n}^{(t)} \in \{0,1\}$ indicate whether the learned policy successfully completes episode observation $\mathbf{o}_{j,n}$ after learning through session t. The success rate is defined as

$$\mathrm{SR}_{t,j} = \frac{1}{N_j} \sum_{n=1}^{N_j} z_{j,n}^{(t)}. \tag{S53}$$

At each evaluation checkpoint, each independently trained policy is evaluated for 100 episodes per manipulation task. Metrics are computed separately for each training run and then summarized across three independent runs.

C.2 Continual Learning Metrics

Each higher-is-better task metric above is instantiated as $R_{t,j}$ in the main-paper definitions of A_{last} and A_{auc}, where $R_{t,j}$ denotes the performance on session j after learning through session t. For selected benchmarks, we additionally report average forgetting ($\bar{F}_T$), forward transfer (FWT), and negative backward transfer (NBT).

Average Forgetting. For a higher-is-better base metric, $\bar{F}_T$ is defined as

$$\bar{F}_T = \frac{1}{T-1} \sum_{j=1}^{T-1} \left(\max_{t \in \{j,\ldots,T-1\}} R_{t,j} - R_{T,j} \right), \tag{S54}$$

which measures the average degradation from the best historical performance on each previously observed session to its final performance.

Forward Transfer measures how knowledge acquired from earlier continual sessions improves performance on a new session before it is learned. Let b_j denote the performance of an untrained or reference model on session j. We define

$$\mathrm{FWT} = \frac{1}{T-1} \sum_{j=2}^{T} \left(R_{j-1,j} - b_j \right), \tag{S55}$$

where higher values indicate stronger transfer from previous to future sessions.

Negative Backward Transfer measures the effect of subsequent learning on previously learned sessions relative to their performance immediately after acquisition:

$$\mathrm{NBT} = \frac{1}{T-1} \sum_{j=1}^{T-1} \left(R_{j,j} - R_{T,j} \right), \tag{S56}$$

where lower values indicate less interference to previously learned sessions.

Algorithm 1 FlyGCL: online training and inference.

Require: Frozen pretrained backbone $f_{\boldsymbol{\theta}}$, expert pool $\Omega = \{\boldsymbol{\omega}_k\}_{k=1}^{K}$, online heads $\{\boldsymbol{\psi}_k^{(0)}\}_{k=1}^{K}$, EMA heads $\{\boldsymbol{\psi}_k^{(j)}\}_{j=1}^{n}$ with rates $\{\alpha_j\}_{j=1}^{n}$, fixed random matrix $\boldsymbol{R} \in \mathbb{R}^{d\times M}$, ridge parameter λ, and online iterations I over each batch

$\boldsymbol{G} \leftarrow \boldsymbol{0}_{M\times M},\ \boldsymbol{Q} \leftarrow \boldsymbol{0}_{M\times K}$
for each session $t = 1, \ldots, T$ **do**
  $k \leftarrow$ index of the active expert for session t
  for each incoming batch $\mathcal{B}_i = \{(\boldsymbol{x}, \boldsymbol{y})\}$ of session t **do** ▷ single pass
    for $r = 1, \ldots, I$ **do**
      $\mathcal{L} \leftarrow \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{B}_i} \ell\big(F_{\boldsymbol{\theta},\boldsymbol{\omega}_k,\boldsymbol{\psi}_k^{(0)}}(\boldsymbol{x}), \boldsymbol{y}\big)$ ▷ task-specific loss
      Update $\boldsymbol{\omega}_k$ and $\boldsymbol{\psi}_k^{(0)}$ by gradient descent on $\mathcal{L}$ ▷ $\boldsymbol{\theta}$ frozen
      for $j = 1, \ldots, n$ **do**
        $\boldsymbol{\psi}_k^{(j)} \leftarrow \alpha_j \boldsymbol{\psi}_k^{(j)} + (1-\alpha_j)\boldsymbol{\psi}_k^{(0)}$ ▷ temporal ensemble
      end for
    end for
    $\boldsymbol{\Phi}_i \leftarrow \sigma\big(f_{\boldsymbol{\theta}}(\boldsymbol{X}_i)\boldsymbol{R}\big)$ ▷ random expansion
    $\boldsymbol{G} \leftarrow \boldsymbol{G} + \boldsymbol{\Phi}_i^\top \boldsymbol{\Phi}_i,\ \boldsymbol{Q} \leftarrow \boldsymbol{Q} + \boldsymbol{\Phi}_i^\top \boldsymbol{C}_t$ ▷ $\boldsymbol{C}_t$: one-hot targets of expert k
  end for
end for

function PREDICT($\boldsymbol{x}$)
  $\widehat{\boldsymbol{U}}^\top \leftarrow (\boldsymbol{G} + \lambda \boldsymbol{I})^{-1}\boldsymbol{Q}$ ▷ closed-form analytic router
  $\hat{E} \leftarrow \arg\max_{k\le K} \big[\sigma\big(f_{\boldsymbol{\theta}}(\boldsymbol{x})\boldsymbol{R}\big)\widehat{\boldsymbol{U}}^\top\big]_k$
  $\boldsymbol{p}^{(j)} \leftarrow F_{\boldsymbol{\theta},\boldsymbol{\omega}_{\hat{E}},\boldsymbol{\psi}_{\hat{E}}^{(j)}}(\boldsymbol{x})$ for $j = 0, \ldots, n$
  return $\mathcal{A}\big(\boldsymbol{p}^{(0)}, \ldots, \boldsymbol{p}^{(n)}\big)$
end function

D Additional Implementation Details

The complete training and inference procedure of FlyGCL is summarized in Supplementary Alg. 1. The visual-recognition experiments use frozen ViT-B/16 backbones with supervised or self-supervised ImageNet pretraining, as specified in Methods and Supplementary Tab. S2. The vision-language experiments use the OpenAI CLIP ViT-B/16 model with frozen pretrained encoders and update only the method-specific lightweight modules and output components. For ego-exo video understanding, we retain each benchmark's pretrained feature extraction and task-specific prediction heads, changing only the continual adaptation module. The embodied vision-language-action experiments initialize a DiT flow-matching policy from a LIBERO-90 pretrained checkpoint (built upon DINOv2 vision encoder and CLIP text encoder), and use the same observation and policy pipeline across methods within each suite. Within each benchmark and task, methods share the same stream, session order, data preprocessing, mini-batch schedule, and evaluation checkpoints.

Optimization settings that of each method are listed in Supplementary Tab. S2. FlyGCL freezes the pretrained backbone and updates only the active adaptive expert and its output heads. The random-expanded router is updated from accumulated sufficient statistics as feature covariance and mean. Unless a method intrinsically requires stored samples, no replay memory is used. Method-specific regularization coefficients, prompt or LoRA dimensions, and other configurations follow the corresponding implementations and are held fixed across sessions after selection on the initial validation stream. All experiments were conducted on a Linux server running Ubuntu 22.04 on Intel Xeon Gold 6530 CPU, with each run using a single NVIDIA GeForce RTX 5090 GPU (32 GB memory), Python 3.10, PyTorch, and CUDA 12.8.

E Extensions beyond the Conference Version

An earlier conference paper, FlyPrompt [71], explored a preliminary brain-inspired GCL framework based on random-expanded expert routing and temporal-ensemble prompt experts. That study focused on continual image classification with prompt-based adaptation over pretrained vision models. The present work substantially extends this preliminary study in its scientific formulation, biological grounding, model generality, theoretical analysis, and empirical scope.

First, FlyGCL develops a broader hierarchical modular principle for GCL. The present work formulates GCL around two complementary requirements: separating conflicting experience to reduce interference and integrating compatible experience to promote generalization. We relate these requirements to MoE and EL, respectively, and study how their hierarchical coordination supports learning under online, uncertain, and evolving data distributions. This formulation generalizes the original algorithmic design into a broader principle for organizing CL.

Second, the biological grounding is substantially expanded. FlyPrompt focused on sparse random expansion and multi-timescale memory in the *Drosophila* olfactory system. FlyGCL develops a more complete correspondence with the hierarchical organization of olfactory learning and memory, including sparse sensory expansion, spatially differentiated downstream pathways, and learning and memory processes operating across distinct temporal scales. We further introduce a controlled biologically grounded olfactory learning model that directly evaluates spatial specialization, temporal integration, and their hierarchical coordination under continual odor streams with different degrees of distributional recurrence.

Third, FlyGCL extends the model and theoretical scope beyond prompt-based visual learning. The hierarchical design can be instantiated with different lightweight learning modules, including prompts, adapters, LoRA branches, video-specific modules, task heads, policy adapters, and action heads, while retaining pretrained backbones as relatively stable representational substrates. We further develop a new theoretical analysis of hierarchical specialization and integration, decomposing GCL risk into separation, integration, and coordination terms and analyzing when the gains from routing and aggregation outweigh their coordination cost. The analysis also characterizes how stable pretrained representations can reduce the penalty of imperfect coordination.

The empirical evaluation is also substantially expanded within visual CL. Beyond reproducing the original image-classification setting, FlyGCL is evaluated across multiple pretrained representations and lightweight learning interfaces, including prompts, adapters, and LoRA. We further analyze the complementary contributions of MoE-based specialization and EL-based integration across different datasets and pretrained backbones. These experiments examine whether the proposed principle remains effective beyond a particular prompt design or backbone initialization, and establish its generality across diverse pretrained visual representations.

We then extend GCL from unimodal visual recognition to multimodal and temporal understanding. FlyGCL is evaluated on continual vision-language learning with pretrained CLIP models, where learning must preserve cross-modal semantic structure while accommodating evolving visual concepts. Beyond overall performance, we analyze changes in image-text representation spaces and the preservation of semantic relations under continual updates. We further evaluate continual ego-exo video understanding, where temporal dynamics, viewpoint shifts, and skill-related variations introduce substantially more complex distribution changes than static image classification. These settings test hierarchical specialization and integration across multimodal semantics, temporal structure, and cross-view experience.

Finally, we extend GCL to embodied vision-language-action learning, where continual updates directly affect sequential decision-making and action policies. FlyGCL is evaluated on language-conditioned robotic manipulation under evolving task distributions, providing a substantially more challenging setting than recognition-oriented benchmarks. Together with the controlled olfactory simulation, these experiments expand the empirical scope from static visual classification to biological modeling, multimodal perception, temporal multi-view understanding, and embodied action. They therefore validate the proposed hierarchical modular principle across a much broader range of models, modalities, and CL scenarios than the conference study.

Table S2: Method-level optimization configurations across benchmark groups. Each method is listed separately within the benchmark group where it is evaluated. All methods within the same benchmark follow the same random seed, stream construction, session order, online update budget, and evaluation schedule. Hyperparameters are selected on the validation split of the first stage stream and then fixed across sessions. Opt., optimizer. LR, learning rate. WD, weight decay. Mem., memory buffer capacity for replay-based ER and DER++.

Benchmark	Method	Backbone	Opt.	LR	WD	Batch	Mem.	Notes
Visual Recog.	SeqFT	ViT-B/16	Adam	5×10^{-3}	0	64	–	Sequentially fine-tunes the full network.
	EWC [17]	ViT-B/16	Adam	5×10^{-3}	0	64	–	Uses regularization based on the Fisher information matrix.
	LwF [63]	ViT-B/16	Adam	5×10^{-3}	0	64	–	Uses output distillation from the previous model.
	L2P [52]	ViT-B/16	Adam	5×10^{-3}	0	64	–	Uses a learnable prompt pool for continual adaptation.
	DualPrompt [53]	ViT-B/16	Adam	5×10^{-3}	0	64	–	Uses task-shared and task-specific prompts.
	MVP [21]	ViT-B/16	Adam	5×10^{-3}	0	64	–	Uses GCL-oriented prompt adaptation.
	MISA [48]	ViT-B/16	Adam	5×10^{-3}	0	64	–	Initializes adaptation with pretrained prompts.
	FlyGCL	ViT-B/16	Adam	5×10^{-3}	0	64	–	Uses adapter, prompt, or LoRA experts; the random expansion dimension and regularization parameter for regression are set to 10,000, and the temporal-ensemble EMA decay rates are 0.9 and 0.99.
Vision Language	SeqFT	CLIP ViT-B/16	AdamW	5×10^{-4}	5×10^{-4}	64	–	Sequentially fine-tunes the CLIP vision encoder.
	EWC [17]	CLIP ViT-B/16	AdamW	5×10^{-4}	5×10^{-4}	64	–	Uses regularization based on the Fisher information matrix.
	LwF [63]	CLIP ViT-B/16	AdamW	5×10^{-4}	5×10^{-4}	64	–	Uses output distillation from the previous model.
	L2P [52]	CLIP ViT-B/16	AdamW	5×10^{-4}	5×10^{-4}	64	–	Uses a learnable prompt pool for continual adaptation.
	DualPrompt [53]	CLIP ViT-B/16	AdamW	5×10^{-4}	5×10^{-4}	64	–	Uses task-shared and task-specific prompts.
	CLAP4CLIP [64]	CLIP ViT-B/16	AdamW	5×10^{-4}	5×10^{-4}	64	–	Provides a CLIP-specific continual-learning baseline.
	MG-CLIP [65]	CLIP ViT-B/16	AdamW	5×10^{-4}	5×10^{-4}	64	–	Provides a CLIP-specific alignment baseline.
	FlyGCL	CLIP ViT-B/16	AdamW	5×10^{-4}	5×10^{-4}	64	–	Uses LoRA experts.
EgoExo-Learn &-Fitness	SeqFT	I3D / CLIP encoder	AdamW	10^{-4}	5×10^{-4}	32	–	Sequentially fine-tunes the full network.
	ER [77]	I3D / CLIP encoder	AdamW	10^{-4}	5×10^{-4}	32	10%	Uses reservoir-based experience replay.
	DER++ [18]	I3D / CLIP encoder	AdamW	10^{-4}	5×10^{-4}	32	10%	Combines experience replay with logit-level distillation.
	EWC [17]	I3D / CLIP encoder	AdamW	10^{-4}	5×10^{-4}	32	–	Uses regularization based on the Fisher information matrix.
	LwF [63]	I3D / CLIP encoder	AdamW	10^{-4}	5×10^{-4}	32	–	Uses output distillation from the previous model.
	L2P+ [52]	I3D / CLIP encoder	AdamW	10^{-4}	5×10^{-4}	32	–	Uses a learnable adapter pool for continual adaptation.
	DualPrompt+ [53]	I3D / CLIP encoder	AdamW	10^{-4}	5×10^{-4}	32	–	Uses task-shared and task-specific adapters.
	S-Prompt+ [76]	I3D / CLIP encoder	AdamW	10^{-4}	5×10^{-4}	32	–	Uses adapter-based adaptation and routing for sequential video tasks.
	FlyGCL	I3D / CLIP encoder	AdamW	5×10^{-4}	5×10^{-4}	32	–	Uses adapter experts.
LIBERO	SeqFT	DiT flow-matching	Adam	10^{-4}	10^{-6}	32	–	Sequentially fine-tunes the full network.
	SeqLoRA	DiT flow-matching	Adam	10^{-4}	10^{-6}	32	–	Sequentially updates LoRA parameters.
	PackNet [70]	DiT flow-matching	Adam	10^{-4}	10^{-6}	32	–	Isolates parameters through model pruning.
	ER [77]	DiT flow-matching	Adam	10^{-4}	10^{-6}	32	10%	Uses reservoir-based experience replay.
	EWC [17]	DiT flow-matching	Adam	10^{-4}	10^{-6}	32	–	Uses regularization based on the Fisher information matrix.
	LwF [63]	DiT flow-matching	Adam	10^{-4}	10^{-6}	32	–	Uses output distillation from the previous model.
	L2P+ [52]	DiT flow-matching	Adam	10^{-4}	10^{-6}	32	–	Uses a learnable adapter pool for continual adaptation.
	DualPrompt+ [53]	DiT flow-matching	Adam	10^{-4}	10^{-6}	32	–	Uses task-shared and task-specific adapters.
	FlyGCL	DiT flow-matching	Adam	10^{-4}	10^{-6}	32	–	Uses adapter experts.

F Additional Results

Table S3: Performance comparison on continual visual recognition benchmarks under the GCL setting using ViT-B/16 backbones with different pretraining. We report average anytime accuracy A_{auc} (↑), final average accuracy A_{last} (↑), and forgetting F (↓). SL, use small backbone learning rate (1/100 of classifier learnign rate).

Method	CIFAR-100			ImageNet-R			CUB-200		
	A_{auc}(↑)	A_{last}(↑)	F(↓)	A_{auc}(↑)	A_{last}(↑)	F(↓)	A_{auc}(↑)	A_{last}(↑)	F(↓)
Backbone: Sup-21K									
SeqFT	19.71±3.39	10.42±4.92	32.25±16.74	7.51±3.94	2.29±0.85	74.79±23.89	3.47±0.41	1.49±0.42	54.01±24.57
Linear Probe	49.69±6.09	23.07±7.33	11.59±1.63	29.24±1.26	16.87±3.14	18.95±1.44	28.96±2.46	17.33±3.08	28.82±3.02
SeqFT w/ SL	64.90±7.18	62.06±1.89	21.09±3.82	47.20±1.47	39.60±2.43	33.99±7.14	56.16±4.32	56.50±3.08	29.05±1.73
EWC	55.94±8.36	58.13±1.77	41.13±4.59	36.29±1.51	33.92±1.86	50.28±12.28	40.27±4.03	35.57±3.25	68.73±13.00
LwF	62.14±8.35	71.59±3.20	18.47±7.95	37.91±1.60	36.66±2.35	48.01±13.05	41.08±3.94	37.00±3.24	67.05±13.30
L2P	76.23±2.73	79.11±1.43	11.53±1.44	44.40±1.03	42.03±1.72	19.16±1.92	64.30±2.18	61.42±2.13	29.15±3.19
DualPrompt	76.04±3.32	76.62±0.74	11.42±0.91	46.13±1.94	40.80±1.04	18.24±4.34	65.03±2.24	62.43±1.78	27.18±2.91
CODA-P	79.13±3.06	80.91±0.70	9.95±1.30	51.87±2.81	48.09±2.75	18.49±2.44	66.01±2.20	62.90±2.46	27.43±3.42
MVP	67.74±4.96	63.22±0.69	33.19±2.33	39.50±1.41	32.63±3.95	43.39±4.23	54.69±3.14	50.07±3.86	47.61±5.73
MISA	80.35±2.39	80.75±1.24	9.67±1.39	51.52±2.09	45.08±1.43	21.46±4.25	65.40±3.01	60.20±1.82	30.06±3.70
FlyGCL-Prompt	83.77±2.21	87.59±0.38	**4.31**±0.82	60.22±1.18	59.62±0.50	**10.74**±1.69	**78.02**±2.74	**84.93**±0.50	3.30±1.72
FlyGCL-Adapter	83.65±2.31	87.29±0.21	4.83±0.59	63.91±1.22	62.64±0.24	12.13±1.57	77.61±2.78	84.37±0.11	**3.27**±1.60
FlyGCL-LoRA	**83.87**±2.22	**87.64**±0.19	4.46±0.62	**65.32**±1.17	**63.74**±0.58	11.79±2.20	77.95±2.49	84.79±0.42	3.55±1.96
Backbone: Sup-21K/1K									
L2P	63.88±7.79	68.96±7.63	14.88±6.27	47.10±1.21	42.22±1.94	35.88±4.51	42.96±4.13	45.00±3.83	35.05±9.31
DualPrompt	68.02±2.08	67.04±5.84	19.18±5.13	52.80±1.21	47.39±1.60	30.73±7.45	46.80±2.89	46.39±2.76	35.40±8.16
CODA-P	69.29±2.52	69.47±7.19	19.82±6.85	51.20±1.76	44.30±1.50	35.58±5.52	44.66±2.73	45.18±4.50	35.73±8.95
MVP	64.69±3.77	51.29±7.56	46.98±8.15	48.99±2.01	38.12±5.20	51.11±2.86	44.10±2.81	33.97±9.62	62.93±8.05
MISA	62.91±7.96	67.99±7.41	11.38±2.76	50.87±1.69	47.75±2.87	28.66±8.19	42.76±2.33	44.05±1.94	33.95±8.63
FlyGCL-Prompt	80.72±1.61	86.74±0.70	**3.09**±0.86	67.13±1.21	65.78±0.69	**12.10**±1.92	69.82±3.66	79.16±1.18	3.41±3.61
FlyGCL-Adapter	**82.79**±1.71	**87.42**±0.81	3.61±1.09	**73.09**±1.44	**70.97**±0.92	13.22±2.06	**71.46**±4.02	**81.23**±1.68	3.25±2.51
FlyGCL-LoRA	80.60±1.45	86.39±0.93	3.17±1.12	71.19±1.24	69.30±0.61	12.32±2.46	70.35±3.54	79.91±0.88	**2.10**±3.58
Backbone: iBOT-21K									
L2P	56.82±8.42	67.61±8.76	12.58±6.24	35.97±1.62	36.95±2.44	34.17±7.54	14.76±1.53	24.51±4.82	19.09±8.05
DualPrompt	66.06±4.52	67.14±8.60	19.22±4.22	42.48±1.62	35.91±0.88	37.96±9.50	19.90±3.68	21.84±2.35	30.18±8.79
CODA-P	62.13±7.17	63.38±7.98	22.20±5.61	45.50±1.66	39.44±1.35	41.21±6.19	17.72±5.33	20.82±7.66	28.98±9.54
MVP	62.33±3.06	48.32±11.42	50.23±11.27	41.55±1.98	29.29±5.03	62.17±6.10	28.73±3.18	23.62±9.51	62.20±5.81
MISA	65.30±2.28	67.43±6.75	17.34±6.20	40.94±1.22	36.16±1.58	35.62±11.69	18.62±3.36	23.66±2.21	21.18±10.00
FlyGCL-Prompt	78.29±1.98	**85.49**±2.13	2.74±1.38	61.08±1.99	61.61±0.37	**16.32**±3.03	34.60±6.33	48.13±4.35	5.44±5.33
FlyGCL-Adapter	**79.27**±1.97	85.45±2.04	2.65±1.39	**64.82**±1.41	**64.41**±0.06	17.02±3.20	**46.31**±6.04	**60.75**±3.11	**4.75**±4.54
FlyGCL-LoRA	75.02±1.69	83.45±2.50	**2.34**±1.52	63.11±2.05	62.07±1.02	16.46±4.10	44.27±4.57	56.80±4.87	6.25±3.83
Backbone: iBOT-1K									
L2P	53.17±7.08	62.28±8.19	16.47±7.80	38.29±2.65	39.86±0.95	33.01±8.96	19.20±2.21	31.21±5.24	21.26±11.09
DualPrompt	52.39±3.21	53.56±6.10	20.45±3.88	45.76±1.63	39.19±0.65	35.14±7.72	29.32±3.15	30.53±5.33	32.70±9.00
CODA-P	59.29±4.03	61.30±6.73	22.72±6.32	49.56±1.57	42.64±2.78	40.23±6.19	27.57±2.83	33.61±4.52	27.71±9.65
MVP	57.52±3.62	44.08±12.42	53.47±12.87	44.76±2.23	34.93±4.48	56.26±5.49	33.81±3.50	26.32±9.97	62.61±6.88
MISA	54.31±2.91	55.89±5.10	19.16±4.04	43.91±3.95	40.09±1.24	33.52±10.12	27.76±2.69	33.74±2.11	29.35±6.60
FlyGCL-Prompt	72.83±2.77	**81.88**±3.12	3.47±1.74	62.78±1.62	62.77±1.08	**15.10**±1.44	45.98±6.11	59.28±2.18	**4.75**±4.44
FlyGCL-Adapter	**73.38**±2.84	81.14±2.53	3.78±2.05	**66.12**±1.17	**64.07**±0.82	16.43±1.74	**53.45**±5.36	**64.90**±1.85	5.72±4.25
FlyGCL-LoRA	66.59±5.95	76.56±5.71	**2.52**±2.11	64.57±1.75	62.02±0.82	16.55±2.22	52.25±5.52	63.54±1.90	5.13±4.29
Backbone: DINO-1K									
L2P	47.98±7.38	59.13±6.32	14.82±6.81	35.81±1.37	36.58±1.31	33.00±7.22	21.18±2.01	32.47±6.10	24.23±9.93
DualPrompt	52.12±4.01	55.71±6.11	18.69±5.27	43.03±1.12	35.40±1.40	37.89±7.65	27.80±4.21	29.49±4.24	31.35±10.61
CODA-P	54.69±4.49	58.91±5.43	19.43±5.83	45.16±2.05	38.23±2.02	41.26±6.20	29.22±2.97	31.85±7.47	32.69±10.65
MVP	53.64±3.91	41.02±12.09	54.70±12.47	41.78±2.15	32.00±4.22	56.54±6.00	33.44±3.43	26.02±10.29	64.21±7.82
MISA	52.03±3.07	55.98±4.26	17.76±4.28	41.26±3.25	37.50±1.62	34.38±10.60	27.13±3.31	33.08±4.10	25.77±10.77
FlyGCL-Prompt	66.93±5.69	**78.07**±5.66	3.13±1.20	59.48±1.43	58.92±0.84	16.32±2.17	46.33±6.20	60.50±2.62	**4.20**±4.88
FlyGCL-Adapter	**67.92**±4.71	77.67±5.29	3.02±2.60	**62.03**±1.50	**61.16**±1.09	**15.95**±1.96	**54.34**±5.17	**66.45**±1.92	4.49±4.32
FlyGCL-LoRA	67.00±3.82	76.12±3.02	**2.97**±2.89	61.61±1.75	60.01±1.23	16.26±2.60	52.45±4.96	64.07±1.91	4.34±3.94
Backbone: MoCo v3-1K									
L2P	28.17±7.08	39.07±11.31	19.68±15.19	17.43±1.71	16.27±5.43	43.59±4.64	12.42±2.31	20.00±7.36	26.38±11.63
DualPrompt	53.33±4.65	58.20±7.73	14.79±4.09	36.69±1.74	30.24±1.94	35.07±5.93	19.88±3.35	21.93±4.30	28.42±9.47
CODA-P	53.47±3.42	58.55±7.19	16.97±6.15	39.89±2.71	31.72±4.86	43.27±2.27	20.09±2.52	24.10±6.48	30.90±10.77
MVP	54.33±4.56	40.84±14.21	55.42±15.00	36.45±2.35	26.37±6.04	55.87±4.26	28.48±3.34	23.56±9.78	62.97±5.67
MISA	57.00±6.06	62.18±3.94	16.19±5.26	38.85±4.27	33.47±0.95	34.42±8.60	25.02±4.39	27.68±4.35	35.50±7.97
FlyGCL-Prompt	67.29±4.65	79.67±3.83	**1.42**±1.92	59.00±1.58	59.68±0.64	**15.29**±2.46	44.69±6.32	59.30±2.31	**4.27**±4.98
FlyGCL-Adapter	73.39±3.02	**82.49**±1.29	2.57±1.83	**61.82**±1.46	59.61±0.78	17.45±2.49	**51.38**±3.41	**62.53**±1.36	7.12±4.20
FlyGCL-LoRA	**73.45**±1.96	81.99±1.52	3.02±1.52	61.40±1.49	**59.75**±0.83	15.81±2.55	50.26±5.11	61.53±3.06	7.28±2.21

Table S4: Ablation study of FlyGCL under the GCL setting using ViT-B/16 backbones pre-trained on ImageNet-21K (Sup-21K) and fine-tuned on ImageNet-1K (Sup-21K/1K). We report average anytime accuracy A_{auc} (↑), final average accuracy A_{last} (↑), forgetting F (↓), and negative backward transfer (NBT, ↓).

Method	Component		CIFAR-100				ImageNet-R			
	MoE	EL	A_{auc} (↑)	A_{last} (↑)	F (↓)	NBT (↓)	A_{auc} (↑)	A_{last} (↑)	F (↓)	NBT (↓)
Backbone: Sup-21K										
	×	×	68.31±4.29	65.95±2.85	40.26±3.40	27.15±2.07	43.12±1.74	43.74±5.28	50.73±3.45	49.28±3.64
	✓	×	81.84±1.95	83.83±1.62	9.51±1.75	0.16±2.69	56.91±1.36	53.61±1.16	18.15±1.71	16.53±1.44
FlyGCL-Prompt	×	✓	81.91±1.98	83.72±1.21	10.40±1.61	0.43±2.55	56.68±1.25	53.67±1.22	18.53±2.27	16.62±2.13
	✓	✓	**83.77**±2.21	**87.59**±0.38	**4.31**±0.82	**−4.23**±1.71	**60.22**±1.18	**59.62**±0.50	**10.74**±1.69	**9.00**±1.67
	×	×	66.78±4.75	64.98±4.72	39.27±3.16	26.16±5.99	46.36±1.79	46.44±2.97	54.02±6.96	52.33±6.85
	✓	×	81.60±1.61	83.09±2.14	10.38±2.29	1.51±3.74	61.04±1.36	56.23±0.87	21.29±3.25	19.79±3.12
FlyGCL-Adapter	×	✓	81.19±1.94	82.19±1.87	12.56±2.24	2.53±3.30	60.22±1.45	61.20±0.58	15.98±2.43	14.19±2.17
	✓	✓	**83.65**±2.31	**87.29**±0.21	**4.83**±0.59	**−4.09**±2.10	**63.91**±1.22	**62.64**±0.24	**12.13**±1.57	**10.72**±1.44
	×	×	70.77±3.86	68.98±3.30	33.24±3.85	21.43±4.65	48.30±2.19	48.87±5.69	54.24±4.25	52.97±4.42
	✓	×	81.27±2.30	83.17±1.05	10.08±1.17	0.53±1.98	62.48±1.41	57.78±0.93	20.04±2.99	18.60±2.57
FlyGCL-LoRA	×	✓	82.04±2.17	83.14±1.72	10.80±1.78	1.13±2.97	62.01±1.30	57.67±0.60	20.76±3.17	18.67±2.89
	✓	✓	**83.87**±2.22	**87.64**±0.19	**4.46**±0.62	**−5.06**±1.89	**65.32**±1.17	**63.74**±0.58	**11.79**±2.20	**10.30**±1.98
Backbone: Sup-21K/1K										
	×	×	57.64±3.84	56.17±9.31	57.11±8.94	42.63±7.39	42.51±1.79	43.84±3.94	59.07±6.71	57.55±6.54
	✓	×	71.08±2.93	71.47±6.50	18.68±5.11	4.74±8.76	55.47±1.05	48.13±1.89	31.95±3.66	30.98±3.53
FlyGCL-Prompt	×	✓	72.77±3.14	69.35±6.50	23.01±4.58	11.43±6.48	56.47±1.22	47.48±2.08	37.84±8.69	36.48±8.51
	✓	✓	**80.72**±1.61	**86.74**±0.70	**3.09**±0.86	**−10.71**±2.94	**67.13**±1.21	**65.78**±0.69	**12.10**±1.92	**10.88**±1.95
	×	×	60.42±2.78	57.58±8.68	59.11±10.34	44.65±7.77	49.06±2.22	48.99±5.26	59.55±8.94	58.13±8.57
	✓	×	72.72±3.30	72.83±7.56	18.24±5.41	6.26±7.93	62.76±1.76	54.83±2.36	32.58±5.81	31.89±5.81
FlyGCL-Adapter	×	✓	74.60±1.93	71.77±5.14	21.95±4.90	11.04±5.45	62.13±1.49	64.01±2.05	24.25±4.34	23.18±4.25
	✓	✓	**82.79**±1.71	**87.42**±0.81	**3.61**±1.09	**−6.78**±2.42	**73.09**±1.44	**70.97**±0.92	**13.22**±2.06	**12.42**±2.06
	×	×	57.01±3.58	54.88±10.98	56.11±11.59	39.63±10.37	47.06±2.13	47.82±3.48	54.80±7.17	53.29±7.26
	✓	×	67.20±4.77	65.91±9.04	19.01±5.99	7.29±6.13	60.20±1.03	52.25±2.96	32.76±5.47	31.67±5.00
FlyGCL-LoRA	×	✓	69.74±4.18	74.10±7.38	17.34±6.33	4.25±5.36	59.32±1.49	62.32±1.18	22.85±3.69	21.74±3.65
	✓	✓	**80.60**±1.45	**86.39**±0.93	**3.17**±1.12	**−10.06**±2.56	**71.19**±1.24	**69.30**±0.61	**12.32**±2.46	**11.33**±2.38

Table S5: Performance comparison on continual vision-language benchmarks under the GCL setting. We report final average accuracy A_{last} (%, ↑), average anytime accuracy A_{auc} (%, ↑), average forgetting F (%, ↓), and backward transfer BWT (%, ↑).

Method	CIFAR-100				ImageNet-R			
	A_{last}(↑)	A_{auc}(↑)	F(↓)	BWT(↑)	A_{last}(↑)	A_{auc}(↑)	F(↓)	BWT(↑)
SeqFT	73.38±1.51	79.96±0.68	21.12±2.22	−19.85±2.30	72.74±0.76	80.45±0.58	17.90±1.29	−15.89±1.43
EWC [17]	71.82±2.01	79.75±0.73	22.98±2.60	−21.95±2.46	69.17±0.44	76.93±0.45	19.94±1.05	−16.95±1.20
LwF [63]	65.1±0.69	74.63±0.90	35.15±0.73	−35.02±0.80	60.11±1.27	73.96±0.65	37.46±1.98	−36.26±1.86
L2P [52]	66.34±0.16	73.55±1.03	8.59±1.18	−8.29±1.20	68.21±0.10	75.96±0.65	8.49±0.26	−8.30±0.21
DualPrompt [53]	65.83±0.19	71.19±1.01	7.12±1.05	−6.02±1.19	66.56±0.11	73.37±0.77	7.98±0.15	−7.36±0.19
CODA-Prompt [54]	66.79±0.08	74.34±0.92	8.72±1.19	−8.60±1.20	71.68±0.00	79.32±0.90	8.00±0.36	−8.00±0.36
CLAP4CLIP [64]	68.24±0.06	75.92±1.16	9.07±0.74	−8.79±0.78	77.01±0.11	83.11±0.42	7.29±0.41	−6.98±0.45
MG-CLIP [65]	77.54±0.36	81.35±0.46	10.48±0.54	−6.64±0.46	72.67±1.83	78.37±0.68	10.91±1.07	−5.66±1.19
FlyGCL (Ours)	**79.59**±0.48	**85.15**±0.22	**4.70**±0.59	**−3.08**±0.53	**79.30**±0.25	**84.72**±0.31	**6.67**±0.66	**−4.58**±0.72

Table S6: Performance comparison on the continual skill assessment benchmark (EgoExoLearn). We report final average ranking accuracy A_{last} (%, ↑), average forgetting $\bar{F}_T$ (%, ↓), and average anytime ranking accuracy A_{auc} (%, ↑).

Method	RAAN + RN (Ego-exo)			RAAN + TL (Ego-exo)			RAAN (Ego-only)		
	A_{last} (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)	A_{last} (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)	A_{last} (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)
SeqFT	72.45±2.21	10.97±1.64	75.76±2.00	71.63±2.13	13.20±1.94	75.87±2.03	72.94±2.15	11.30±1.35	76.19±1.84
ER [77]	77.39±0.50	3.13±2.83	79.14±1.40	78.27±0.73	2.39±0.95	79.80±1.61	77.24±0.56	3.94±2.30	79.07±1.48
DER++ [18]	78.60±0.30	1.17±1.70	79.50±1.08	79.47±0.07	1.09±0.81	80.14±1.16	79.46±0.26	0.98±1.24	79.98±0.95
EWC [17]	72.59±1.95	11.32±2.26	75.72±1.85	71.05±2.00	14.22±2.44	75.75±1.96	73.34±2.28	10.62±1.75	76.29±1.77
LwF [63]	68.87±2.58	12.52±3.20	74.42±1.65	71.83±2.11	0.49±2.24	74.63±3.04	72.61±1.73	2.21±3.63	75.13±2.98
L2P+ [52]	71.43±2.56	12.52±1.20	74.98±1.65	71.38±0.97	14.22±3.81	75.18±0.85	72.03±1.48	11.69±3.64	75.58±0.93
DualPrompt+ [53]	70.04±5.90	9.26±3.04	74.91±2.37	75.21±0.70	7.45±2.50	77.04±1.10	75.90±0.44	6.02±2.02	77.42±0.91
S-Prompt+ [76]	77.75±0.41	**0.00**±0.21	79.61±2.00	78.05±0.78	**0.13**±0.17	80.17±2.10	79.17±0.30	**0.05**±0.03	80.49±1.76
FlyGCL (Ours)	**82.66**±0.22	0.31±0.22	**81.38**±0.17	**82.59**±0.23	0.31±0.25	**81.32**±0.15	**82.27**±0.47	0.38±0.28	**81.30**±0.11

Table S7: Performance comparison on the continual action anticipation benchmark (EgoExoLearn). We report final average Top-5 recall A_{last} (%, ↑) and average anytime Top-5 recall A_{auc} (%, ↑) for verb (-V) and noun (-N) prediction. Avg. denotes the average over Ego-V, Ego-N, Exo-V, and Exo-N.

Method	A_{last} (↑)					A_{auc} (↑)				
	Ego-V	Ego-N	Exo-V	Exo-N	Avg.	Ego-V	Ego-N	Exo-V	Exo-N	Avg.
Ego-exo										
SeqFT	31.25±1.15	19.26±0.67	30.30±0.91	18.82±0.54	24.91±0.82	31.60±0.33	20.60±0.90	30.32±0.63	20.07±0.95	25.65±0.70
ER [77]	34.49±0.53	25.29±0.24	32.70±0.15	24.75±0.23	29.31±0.29	33.44±0.28	24.44±1.00	32.03±0.31	23.80±1.16	28.43±0.69
DER++ [18]	33.70±0.37	24.55±0.53	32.37±0.26	24.13±0.35	28.69±0.38	32.92±0.33	23.57±1.55	31.73±0.38	23.18±1.54	27.85±0.95
EWC [17]	31.29±1.04	19.96±0.48	30.64±0.95	19.44±0.25	25.33±0.68	31.54±0.20	20.67±1.03	30.04±0.44	20.11±1.17	25.59±0.71
LwF [63]	30.57±0.21	21.38±1.46	31.19±1.32	20.58±1.16	25.93±1.04	30.58±0.38	21.14±0.81	30.08±0.21	20.50±0.94	25.57±0.58
L2P+ [52]	28.51±1.89	16.19±0.80	26.12±0.93	15.21±0.62	21.51±1.06	28.85±0.62	17.53±0.81	26.35±1.12	16.74±1.03	22.37±0.90
DualPrompt+ [53]	29.50±2.37	16.02±2.03	28.31±1.69	16.02±1.81	22.47±1.98	32.66±1.95	18.19±3.01	30.57±3.17	18.25±2.90	24.92±2.76
S-Prompt+ [76]	31.83±2.24	16.49±2.43	27.64±1.46	16.36±2.02	23.08±2.04	**35.09**±4.24	18.97±3.54	31.56±4.18	19.08±3.43	26.17±3.85
FlyGCL (Ours)	**35.21**±0.52	**34.18**±0.60	**39.18**±2.26	**41.65**±1.28	**37.55**±1.15	32.90±0.32	**32.46**±0.28	**38.75**±0.77	**38.85**±0.47	**35.74**±0.37
Ego-only										
SeqFT	29.83±1.05	20.77±0.99	30.47±0.85	20.34±0.81	25.35±0.93	29.36±0.31	22.03±1.48	28.81±0.25	21.59±1.57	25.45±0.90
ER [77]	31.15±0.49	26.47±0.47	**30.84**±0.23	**26.50**±0.21	28.74±0.35	30.41±0.40	26.04±1.37	**30.25**±0.32	**26.03**±1.55	**28.18**±0.91
DER++ [18]	31.31±0.36	26.91±0.69	30.68±0.22	26.16±0.56	28.77±0.46	30.29±0.57	26.15±2.08	29.40±0.65	25.54±1.86	27.85±1.29
EWC [17]	29.72±1.12	20.63±1.00	30.16±0.37	20.19±0.84	25.18±0.83	29.31±0.18	22.02±1.22	28.69±0.33	21.56±1.29	25.39±0.76
LwF [63]	29.84±0.77	22.24±0.77	29.75±0.75	21.79±0.74	25.91±0.76	29.19±0.41	22.52±0.92	28.76±0.58	22.09±1.03	25.64±0.73
L2P+ [52]	28.76±1.14	16.24±1.18	27.88±1.20	15.50±1.34	22.10±1.21	28.77±0.40	17.29±1.05	27.24±0.53	16.86±0.84	22.54±0.70
DualPrompt+ [53]	27.86±1.30	17.54±2.14	27.36±0.85	17.42±1.92	22.54±1.55	28.77±1.09	19.73±3.41	27.93±1.98	19.92±3.56	24.09±2.51
S-Prompt+ [76]	30.41±1.11	18.36±1.72	27.99±1.62	18.23±1.35	23.75±1.45	29.36±1.48	21.54±3.61	28.58±2.89	21.73±3.56	25.30±2.89
FlyGCL (Ours)	**34.97**±0.19	**33.75**±0.45	28.53±0.92	18.27±0.21	**28.88**±0.33	**32.66**±0.23	**31.75**±0.34	27.82±0.22	15.85±0.55	27.02±0.07

Table S8: Performance comparison on the continual skill assessment benchmark (EgoExo-Fitness). We report final average ranking accuracy $\bar{A}_{\text{last}}$ (%, ↑), average forgetting $\bar{F}_T$ (%, ↓), and average anytime ranking accuracy A_{auc} (%, ↑).

Method	RAAN (Ego-only)			RAAN+RN (Ego-exo)			RAAN+TL (Ego-exo)		
	$\bar{A}_{\text{last}}$ (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)	$\bar{A}_{\text{last}}$ (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)	$\bar{A}_{\text{last}}$ (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)
SeqFT	54.26±1.87	1.55±1.02	54.62±3.36	54.21±1.98	1.60±1.07	54.52±3.58	54.22±1.89	1.63±0.99	54.58±3.39
ER [77]	54.62±1.86	1.02±0.57	54.68±3.48	54.36±1.99	1.39±0.97	54.64±3.57	54.60±1.92	0.78±0.40	54.64±3.50
DER++ [18]	54.68±1.74	**0.45**±0.06	54.78±3.41	54.73±2.11	**0.65**±0.56	54.71±3.57	54.62±1.71	**0.27**±0.14	54.73±3.42
EWC [17]	54.22±1.86	1.60±1.02	54.62±3.34	54.17±2.00	1.69±1.19	54.62±3.34	54.22±1.87	1.60±1.00	54.58±3.39
LwF [63]	54.28±1.87	1.36±0.89	54.67±3.30	54.31±2.00	1.34±0.90	54.52±3.54	54.42±1.90	1.14±0.66	54.67±3.34
L2P+ [52]	54.33±1.96	1.53±0.92	54.62±3.38	54.12±2.08	1.67±1.09	54.60±3.44	54.23±1.94	1.62±1.01	54.62±3.41
DualPrompt+ [53]	50.94±1.34	3.05±3.20	52.30±2.22	51.64±0.34	1.97±2.10	52.36±2.52	50.95±1.27	3.21±3.06	52.38±2.27
S-Prompt+ [76]	57.85±3.97	1.07±0.20	57.22±3.64	58.04±2.63	1.19±0.85	57.98±3.13	57.75±3.35	1.79±0.33	57.94±2.70
FlyGCL (Ours)	**62.47**±1.04	0.89±1.15	**61.60**±3.14	**62.48**±2.06	1.94±1.42	**62.29**±3.24	**62.99**±1.25	1.09±1.04	**61.99**±2.97

Table S9: Performance comparison on the continual action classification benchmark (EgoExo-Fitness). We report final average accuracy A_{last} (%, ↑), average forgetting $\bar{F}_T$ (%, ↓), and average anytime accuracy A_{auc} (%, ↑).

Method	Ego-exo			Ego-only		
	A_{last} (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)	A_{last} (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)
SeqFT	24.97±10.60	−1.82± 1.56	28.86±15.63	25.00± 8.53	−7.75± 1.43	28.70±12.93
ER [77]	27.31± 5.69	**−4.14**± 4.72	31.35±10.32	28.40± 3.97	**−11.78**± 3.39	30.45± 9.47
DER++ [18]	22.29± 3.30	−1.67± 1.33	29.00±10.32	21.13± 4.10	−6.29± 2.34	27.67± 9.54
EWC [17]	24.93±10.58	−1.91± 1.63	28.83±15.62	25.08± 8.43	−8.07± 1.38	28.66±12.87
LwF [63]	24.98±10.25	−3.35± 2.60	28.67±15.34	25.43± 9.30	−9.36± 4.65	28.64±12.60
L2P+ [52]	14.53± 1.85	0.99± 1.41	28.86± 0.61	13.44± 2.17	0.14± 0.58	25.88± 6.16
DualPrompt+ [53]	18.54± 6.55	20.02± 7.52	28.85±12.07	17.10± 6.72	19.76± 6.97	28.31± 8.97
S-Prompt+ [76]	12.79± 4.11	14.43±12.66	23.72±12.77	11.02± 5.14	6.40± 8.90	23.21± 7.75
FlyGCL (Ours)	**37.86**±12.85	7.02± 6.41	**42.34**±16.42	**39.61**±13.23	2.08± 5.41	**41.05**±14.34

Table S10: Performance comparison on the continual sequence verification benchmark (EgoExo-Fitness). We report final average performance $\bar{A}_{\text{last}}$ (%, ↑), average forgetting $\bar{F}_T$ (%, ↓), and average anytime performance A_{auc} (%, ↑) in terms of ROC-AUC and mAP.

Method	ROC-AUC			mAP		
	$\bar{A}_{\text{last}}$ (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)	$\bar{A}_{\text{last}}$ (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)
Ego-exo						
SeqFT	83.42±2.64	5.75±3.88	87.72±0.53	53.43±5.11	22.58±5.26	70.37±4.29
ER [77]	86.02±0.52	4.49±1.38	89.40±0.52	57.26±4.45	20.02±4.96	72.26±4.21
DER++ [18]	89.58±1.21	4.28±1.86	**92.83**±1.01	64.55±3.20	19.68±5.75	79.29±2.94
EWC [17]	83.65±2.45	5.55±4.07	87.83±0.61	53.34±5.05	22.73±5.45	70.38±4.25
LwF [63]	84.22±1.72	5.61±3.07	88.44±0.66	55.90±5.78	21.13±6.00	71.76±4.35
L2P+ [52]	83.95±2.39	5.37±3.71	88.01±0.30	53.98±5.38	22.38±5.03	70.75±4.67
DualPrompt+ [53]	84.85±2.12	6.89±3.64	89.92±0.93	55.33±3.31	24.49±6.99	73.69±3.43
S-Prompt+ [76]	86.67±1.31	6.39±1.18	91.43±1.27	62.05±6.47	22.11±3.62	78.52±6.40
FlyGCL (Ours)	**89.64**±2.09	**4.22**±2.51	92.62±1.91	**76.77**±8.82	**11.23**±8.12	**85.25**±4.56
Ego-only						
SeqFT	83.64±3.57	5.55±3.02	86.61±1.66	51.37±3.65	24.25±5.26	68.86±3.15
ER [77]	83.71±3.30	6.29±2.67	88.09±1.38	53.92±4.55	23.00±6.10	70.18±2.79
DER++ [18]	86.81±2.55	6.19±2.93	89.94±0.63	57.08±4.12	23.46±6.24	74.19±1.62
EWC [17]	82.59±2.24	5.56±2.48	86.61±1.66	50.70±3.61	24.21±5.41	68.86±3.15
LwF [63]	84.52±3.87	5.37±4.24	87.73±0.72	53.02±4.86	23.02±5.64	69.48±3.51
L2P+ [52]	83.32±3.55	5.96±2.81	87.17±1.32	51.93±4.15	23.78±5.41	68.73±3.25
DualPrompt+ [53]	85.24±2.92	4.50±3.21	88.62±1.21	51.93±4.07	25.23±4.69	70.85±2.81
S-Prompt+ [76]	84.37±8.05	7.53±6.94	85.57±0.39	53.15±6.46	26.03±6.97	67.19±2.50
FlyGCL (Ours)	**92.52**±3.89	**3.84**±2.43	**95.40**±2.12	**86.29**±8.62	**7.72**±5.82	**92.08**±4.48

Table S11: Performance comparison on the continual guidance-based execution verification benchmark (EgoExo-Fitness). We report final average performance A_{last} (%, ↑), average forgetting $\bar{F}_T$ (%, ↓), and average anytime performance A_{auc} (%, ↑) in terms of classification accuracy and F1 score.

Method	Cls Acc			F1		
	A_{last} (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)	A_{last} (↑)	$\bar{F}_T$ (↓)	A_{auc} (↑)
Ego-exo						
SeqFT	58.12± 6.20	−1.65± 2.86	53.45± 5.69	66.96± 8.18	−2.62± 4.11	59.80± 7.88
ER [77]	59.43± 8.86	−2.93± 1.25	53.99± 6.13	67.73±10.81	−3.18± 2.56	60.45± 8.31
DER++ [18]	58.75± 6.14	**−3.23**± 9.79	53.87± 1.71	65.05±8.81	**−4.15**± 13.17	59.04± 2.68
EWC [17]	57.99± 9.36	−1.65± 2.32	53.55± 6.20	66.06±11.89	−1.64± 4.21	59.96± 8.45
LwF [63]	56.87± 9.15	−1.35± 2.81	53.28± 6.07	65.01±11.50	−1.31± 3.08	59.89± 8.69
L2P+ [52]	55.49± 9.44	0.36± 4.87	52.26± 5.53	62.80±13.09	1.03± 8.27	58.13± 7.61
DualPrompt+ [53]	58.08± 8.84	−0.16± 7.36	53.90± 6.91	64.53±12.01	0.02± 9.51	59.42±11.08
S-Prompt+ [76]	58.35±16.81	12.68±12.34	72.11±10.07	62.02±20.48	13.77±14.29	76.43±11.26
FlyGCL (Ours)	**76.27**± 2.52	3.33± 5.50	**78.98**± 7.00	**85.75**± 2.70	−0.97± 6.23	**84.90**± 7.60
Ego-only						
SeqFT	58.74± 6.40	−1.58± 8.49	55.03± 6.34	65.53± 7.42	−2.00±10.54	60.75± 9.02
ER [77]	65.04±17.74	−0.92±16.56	58.31± 8.06	69.37±22.33	2.06±21.76	63.36±10.65
DER++ [18]	64.07± 8.32	−1.46± 8.67	58.72± 5.55	70.26± 9.96	−0.89±11.08	64.25± 7.31
EWC [17]	57.62±13.90	0.11±14.34	54.60± 8.50	62.64±17.55	1.98±18.53	59.78±12.01
LwF [63]	**66.92**± 6.46	**−5.80**± 7.55	58.19± 5.80	**75.14**± 7.40	**−6.76**± 9.31	64.67± 7.98
L2P+ [52]	58.30± 7.16	−1.83±10.72	54.51± 5.09	64.81± 8.35	−1.90±13.86	59.93± 7.64
DualPrompt+ [53]	51.63±20.06	8.55±21.36	51.31± 8.34	51.69±30.21	15.71±34.57	54.47±12.67
S-Prompt+ [76]	53.82± 2.43	6.56± 2.67	53.91± 4.16	60.46± 3.86	8.54± 4.23	60.07± 6.51
FlyGCL (Ours)	65.87± 8.90	11.98± 8.95	**72.57**± 8.64	73.13± 9.75	8.74± 9.77	**77.34**± 8.58

Table S12: Performance comparison on continual embodied vision-language-action learning benchmarks under the GCL setting (LIBERO). We report final average success rate A_{last} (%, ↑), average anytime success rate A_{auc} (%, ↑), forward transfer (FWT, %, ↑), and negative backward transfer (NBT, %, ↓).

(a) LIBERO-Spatial

Method	A_{last} (↑)	A_{auc} (↑)	FWT (↑)	NBT (↓)
SeqFT	23.67±1.02	44.06±0.92	63.17±2.79	50.58±2.03
SeqLoRA	19.33±2.51	33.45±1.86	69.00±3.21	59.23±0.21
PackNet [70]	2.30±0.76	15.24±7.46	57.33±2.60	55.38±1.25
ER [77]	71.50±1.40	82.46±2.14	80.67±1.25	13.50±1.11
EWC [17]	14.03±1.93	28.64±8.47	80.17±2.43	63.83±1.15
LwF [63]	13.27±1.35	23.24±2.99	71.17±1.15	62.57±3.37
L2P+ [52]	6.27±0.89	18.09±1.70	73.67±1.61	60.56±1.03
DualPrompt+ [53]	33.63±2.08	36.58±1.03	**88.44**±1.50	60.03±2.00
FlyGCL (Ours)	**83.12**±1.42	81.58±0.64	82.75±3.84	**−3.40**±1.59

(b) LIBERO-Object

Method	A_{last} (↑)	A_{auc} (↑)	FWT (↑)	NBT (↓)
SeqFT	30.83±1.16	47.88±1.91	**94.67**±1.97	43.10±2.47
SeqLoRA	21.70±1.20	33.15±0.26	67.33±0.93	36.33±1.71
PackNet [70]	1.93±0.75	11.64±2.92	39.83±1.25	32.03±1.89
ER [77]	**89.50**±0.50	**91.25**±1.12	87.83±2.93	7.67±3.06
EWC [17]	36.33±1.51	39.94±1.22	85.50±1.77	60.84±5.01
LwF [63]	26.77±1.53	30.30±16.01	86.07±1.73	50.32±1.53
L2P+ [52]	8.63±2.08	8.33±2.79	87.00±0.75	54.49±3.71
DualPrompt+ [53]	33.67±1.86	50.11±9.86	84.67±1.02	56.07±1.24
FlyGCL (Ours)	86.11±1.56	86.03±1.01	87.73±1.30	1.40±1.55

(c) LIBERO-Goal

Method	A_{last} (↑)	A_{auc} (↑)	FWT (↑)	NBT (↓)
SeqFT	21.27±1.33	46.09±2.98	86.33±1.89	44.18±2.33
SeqLoRA	11.80±0.65	23.27±1.73	77.83±0.75	37.66±3.00
PackNet [70]	3.90±0.44	20.00±0.56	30.00±1.93	9.23±1.07
ER [77]	74.73±1.32	82.01±1.11	84.33±1.31	29.68±2.25
EWC [17]	12.67±1.29	22.82±1.27	88.83±0.29	43.84±0.76
LwF [63]	13.13±2.42	23.39±0.92	92.17±1.58	41.03±2.57
L2P+ [52]	10.03±1.02	12.73±1.83	89.00±3.18	30.24±3.03
DualPrompt+ [53]	35.67±1.81	46.92±2.64	90.67±4.50	40.88±2.66
FlyGCL (Ours)	**94.50**±1.95	**91.11**±1.46	**92.25**±2.33	−1.51±0.41

(d) LIBERO-Long

Method	A_{last} (↑)	A_{auc} (↑)	FWT (↑)	NBT (↓)
SeqFT	25.37±0.66	34.12±0.70	**80.67**±1.41	51.31±3.08
SeqLoRA	14.93±1.51	33.21±1.86	15.17±1.25	35.18±1.25
PackNet [70]	2.33±0.08	10.85±2.04	16.83±1.48	5.56±0.82
ER [77]	68.20±0.29	75.29±1.31	45.33±0.62	16.85±1.48
EWC [17]	13.73±1.53	15.76±0.19	28.33±1.32	34.33±3.06
LwF [63]	18.07±1.51	19.36±0.90	30.50±0.26	42.17±1.75
L2P+ [52]	15.30±1.11	15.09±1.74	28.33±3.54	35.05±7.07
DualPrompt+ [53]	22.63±1.53	28.81±2.18	28.67±1.24	40.70±2.96
FlyGCL (Ours)	**79.13**±1.41	**79.12**±1.11	80.15±2.42	−1.15±1.01

Table S13: Performance comparison on continual embodied vision-language-action learning benchmarks under the offline CL setting (LIBERO). We report final average success rate A_{last} (%, ↑), average anytime success rate A_{auc} (%, ↑), forward transfer (FWT, %, ↑), and negative backward transfer (NBT, %, ↓).

(a) LIBERO-Spatial

Method	A_{last} (↑)	A_{auc} (↑)	FWT (↑)	NBT (↓)
SeqFT	9.27±0.35	26.53±0.41	88.07±0.93	86.88±0.75
SeqLoRA	7.43±1.46	23.33±0.97	81.37±1.07	82.35±2.00
PackNet [70]	0.13±0.23	3.94±0.59	29.97±3.52	33.04±3.80
ER [77]	63.70±0.87	71.23±2.51	**88.40**±0.46	24.76±3.72
EWC [17]	8.87±0.42	25.87±0.53	86.43±1.56	85.49±1.57
LwF [63]	6.27±1.23	21.39±1.52	74.30±2.26	74.90±1.30
L2P+ [52]	6.30±2.95	22.34±3.47	81.47±4.42	83.16±1.52
DualPrompt+ [53]	14.10±1.55	33.34±0.93	84.67±2.04	72.71±2.17
FlyGCL (Ours)	**86.77**±0.84	**86.61**±0.26	86.03±0.99	**−0.61**±1.11

(b) LIBERO-Object

Method	A_{last} (↑)	A_{auc} (↑)	FWT (↑)	NBT (↓)
SeqFT	20.90±2.00	36.41±0.28	96.03±0.40	84.68±0.32
SeqLoRA	8.53±2.25	18.13±2.93	58.87±3.62	55.05±2.66
PackNet [70]	0.00±0.00	3.51±0.80	28.93±6.51	32.15±7.24
ER [77]	85.10±0.10	88.31±1.93	95.03±1.02	9.56±2.86
EWC [17]	23.63±1.07	37.80±0.13	**96.93**±0.68	84.08±1.11
LwF [63]	19.27±2.21	36.03±0.32	95.57±0.64	84.61±0.66
L2P+ [52]	8.17±0.75	25.41±0.21	84.53±1.21	82.83±2.54
DualPrompt+ [53]	9.93±0.31	30.00±1.02	83.83±1.66	75.03±2.01
FlyGCL (Ours)	**89.27**±2.37	**88.82**±1.55	88.50±1.95	**−0.50**±0.53

(c) LIBERO-Goal

Method	A_{last} (↑)	A_{auc} (↑)	FWT (↑)	NBT (↓)
SeqFT	8.93±0.42	27.31±0.28	**94.60**±0.53	95.28±0.74
SeqLoRA	1.70±1.31	17.07±1.42	70.67±1.76	76.49±2.22
PackNet [70]	0.00±0.00	4.68±0.88	28.90±5.48	32.10±6.10
ER [77]	59.93±5.58	73.60±3.94	93.83±0.45	28.24±5.80
EWC [17]	8.77±0.25	27.17±0.43	94.53±1.24	95.32±1.08
LwF [63]	7.97±0.46	25.18±0.29	90.40±0.10	92.10±0.44
L2P+ [52]	8.80±1.06	26.29±1.19	90.07±1.63	90.32±1.81
DualPrompt+ [53]	25.73±2.54	39.88±1.96	91.03±1.39	70.57±3.72
FlyGCL (Ours)	**93.40**±0.72	**93.17**±0.59	93.20±1.04	**−0.03**±0.56

(d) LIBERO-Long

Method	A_{last} (↑)	A_{auc} (↑)	FWT (↑)	NBT (↓)
SeqFT	8.63±0.15	21.77±0.25	72.97±1.43	71.48±1.76
SeqLoRA	6.73±0.21	17.98±0.55	64.03±0.83	63.62±1.08
PackNet [70]	0.00±0.00	2.10±0.55	17.70±4.39	19.67±4.87
ER [77]	43.03±0.76	56.56±1.54	71.27±2.19	21.93±5.17
EWC [17]	8.13±0.15	21.53±0.16	73.37±1.01	72.48±1.11
LwF [63]	8.20±0.10	21.29±0.53	72.37±2.72	71.30±3.08
L2P+ [52]	20.97±1.71	33.38±2.70	71.37±1.46	50.02±2.93
DualPrompt+ [53]	10.77±1.08	25.71±0.91	68.90±0.92	59.39±1.69
FlyGCL (Ours)	**76.93**±0.40	**76.92**±0.47	**76.93**±1.14	**0.02**±2.21