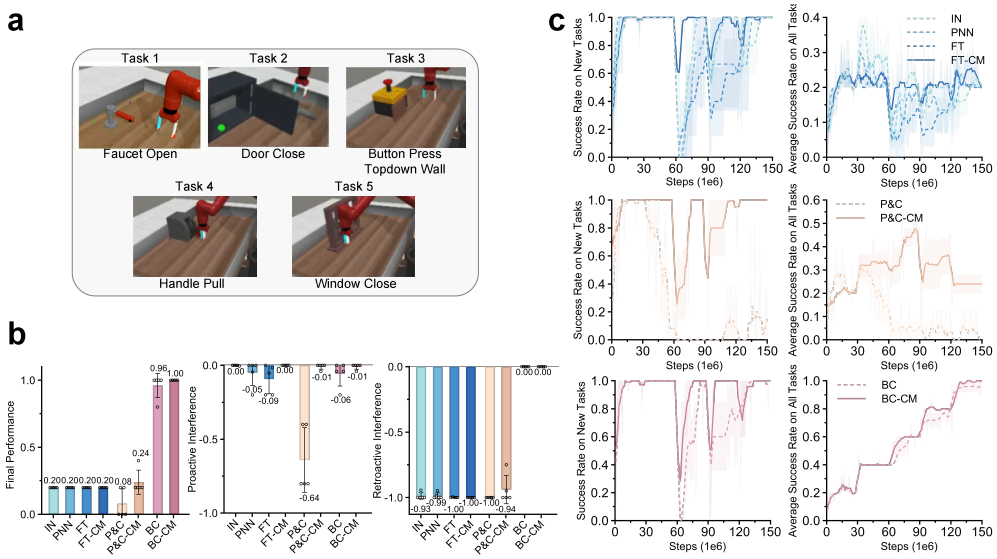
Supplementary Materials: Neuro-inspired cascaded memory coordination enables continual generalization

Zijian Gao[1,2], Xingxing Zhang[3*], Liyuan Wang[4], Bo Ding[1,2], Xinjun Mao[1,2], Huaimin Wang[1,2], Kele Xu[1,2*]

[1]College of Computer Science and Technology, National University of Defense Technology, Changsha, China.
[2]State Key Laboratory of Complex & Critical Software Environment, Changsha, China.
[3]Department of Computer Science and Technology, Tsinghua University, Beijing, China.
[4]Department of Psychological and Cognitive Sciences, Tsinghua University, Beijing, China.

*Corresponding author(s). E-mail(s): xxzhang1993@gmail.com; xukelele@nudt.edu.cn;
Contributing authors: gaozijian19@nudt.edu.cn; liyuanwang@tsinghua.edu.cn; dingbo@nudt.edu.cn; xjmao@nudt.edu.cn; hmwang@nudt.edu.cn;

Contents



Supplementary Fig. 1 Robustness of cascaded memory coordination on an alternative task order. a, Five-task Meta-World CRL sequence ordered by difficulty hierarchy [1]. **b,** Final performance, proactive interference (PI), and retroactive interference (RI) across methods. CM variants consistently improve PI while maintaining competitive final performance and controlled RI. **c,** Learning curves showing success on the current task and average success across all tasks. CM variants improve new-task acquisition and cumulative performance across the alternative task sequence. Shaded regions and error bars denote mean ± s.e.m. across 5 random seeds.

1 Extended Experimental Results

1.1 Extended results in continual reinforcement learning

In the main text, we evaluate cascaded memory coordination on a Meta-World task sequence chosen to induce strong inter-task interference. To test whether the observed gains persist under milder transfer conditions, we additionally consider an alternative five-task sequence (Supplementary Fig. 1a), following the analysis of Ahn et al. [1]. This ordering reduces negative transfer between consecutive tasks and therefore provides a complementary continual reinforcement learning regime.

Supplementary Fig. 1b summarizes final performance, proactive interference (PI), and retroactive interference (RI) on this alternative sequence. Relative to the task order used in the main text, proactive interference is substantially weaker overall, indicating more favorable forward-transfer conditions and correspondingly smaller performance gaps between methods. Even in this setting, however, cascaded memory coordination continues to improve performance across optimization regimes. In the regularization-based setting, P&C [2] achieves a final success rate of 0.08, whereas P&C-CM improves this to 0.24. In the replay-based setting, BC [3] already performs strongly, but BC-CM further improves final success from 0.96 to 1.00.

The corresponding learning curves are shown in Supplementary Fig. 1c. Although adaptation is generally easier than in the main-text task sequence, CM variants still

exhibit faster acquisition of new tasks and improved cumulative performance. These gains are accompanied by consistently improved PI across methods. For example, the PI of P&C improves from -0.64 to -0.01 after introducing the cascaded architecture. Importantly, these gains do not come at the expense of stronger forgetting: RI remains controlled and, in the replay-based setting, BC-CM maintains RI = 0.00.

Taken together, these results show that the benefits of cascaded memory coordination are not restricted to highly adversarial task orderings. Even when interference is reduced through task sequencing, the cascaded architecture continues to improve adaptation efficiency while preserving previously learned policies.

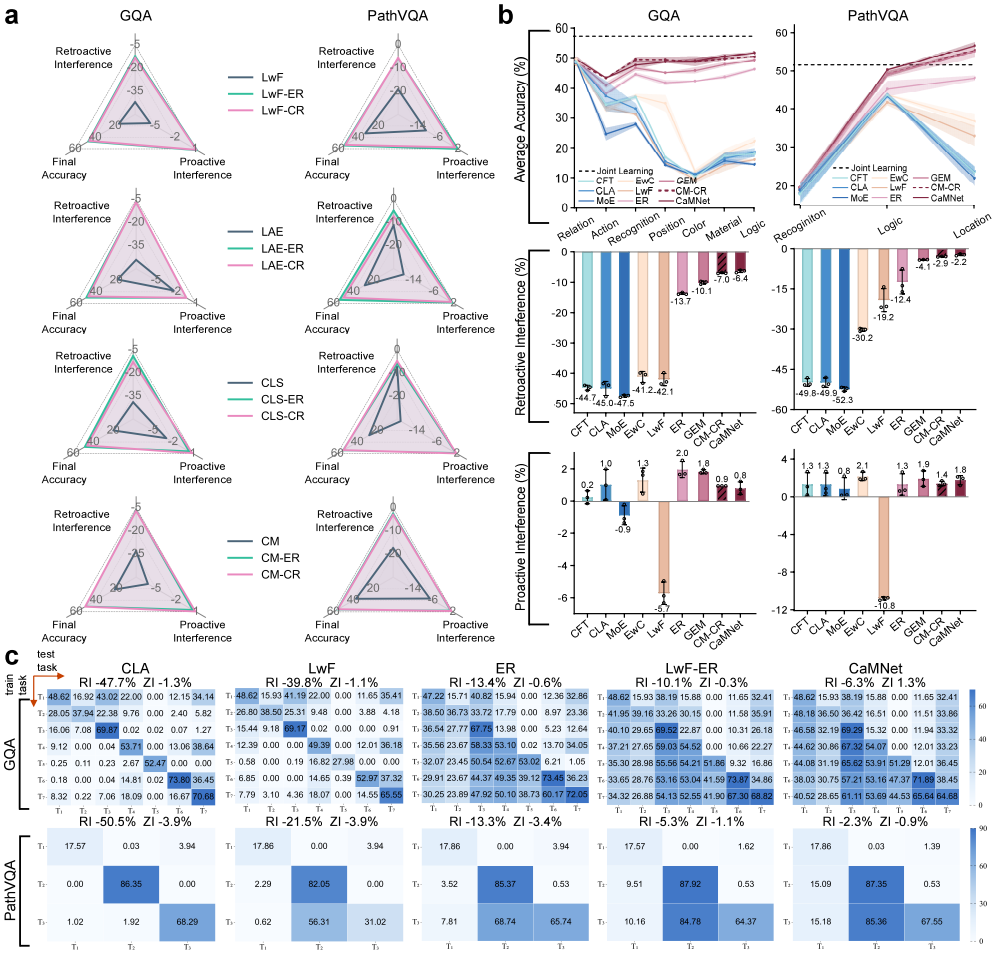
1.2 Extended results on additional multimodal benchmarks

In the main text, we analyze continual generalization primarily on VQAv2 and SVLC. Here, we extend the evaluation to the remaining multimodal benchmarks used in this work, namely GQA [4] and PathVQA [5], to test whether the same trends hold across additional reasoning domains. Results are summarized in Supplementary Fig. 2.

We first evaluate the effect of cue-driven reactivation (CR) on both benchmarks. Supplementary Fig. 2a summarizes final accuracy, retroactive interference (RI), and proactive interference (PI) for four representative frameworks (LwF [6], LAE [7], CLS [8], and CM) under three memory settings: no replay, experience replay (ER), and CR. Across both GQA and PathVQA, CR consistently expands the performance envelope relative to replay-free variants. This effect is strongest within the cascaded architecture. On GQA, CM-CR reduces RI from –33.0% to –7.0% while maintaining final accuracy comparable to CM-ER. On PathVQA, CM-CR improves RI from –17.4% to –2.9%, approaching the stability of CM-ER (–2.4%) without storing task-specific samples. These results indicate that compact textual cues can support effective memory reactivation across diverse multimodal reasoning settings.

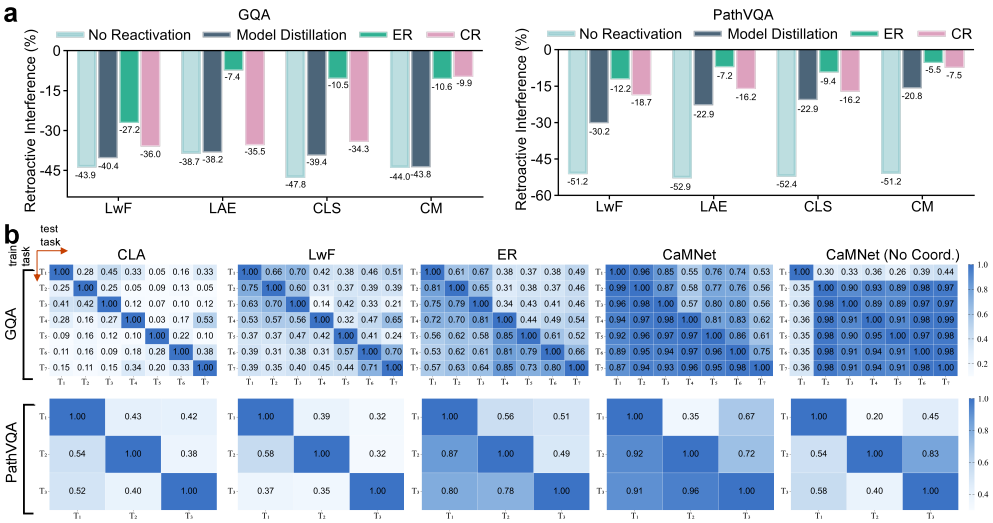
Supplementary Fig. 2b shows the corresponding average-accuracy trajectories together with RI and PI for GQA and PathVQA. Across both datasets, CaMNet maintains stable performance as tasks accumulate, whereas baselines such as LwF [6] and Continual-LoRA (CLA) [10] show progressively larger degradation. On PathVQA, CaMNet also exceeds the joint-learning baseline on the recognition task. Although this task is not the smallest by sample count, it is substantially harder to optimize, requiring fine-grained discrimination of pathological visual patterns. Under joint learning, optimization is more easily dominated by tasks such as logic and location, whereas CaMNet learns recognition in a dedicated stage before integrating it into the shared representation. This structured training process preserves task-specific features during acquisition and leads to markedly stronger recognition performance.

To further examine the relationship between retention and zero-shot generalization, Supplementary Fig. 2c reports pairwise accuracy matrices for GQA and PathVQA. Across both benchmarks, CaMNet combines improved backward retention with stable zero-shot transfer. On GQA, for example, CaMNet achieves positive zero-shot interference (ZI: 1.3%), whereas LwF [6] yields negative ZI (–1.1%). This pattern mirrors the corresponding RI scores, again indicating that methods with stronger retention also tend to generalize more reliably to unseen future tasks. Together, these results extend the main-text findings to additional multimodal benchmarks and support the



Supplementary Fig. 2 Extended multimodal results on GQA and PathVQA. a, Radar plots summarizing final accuracy, retroactive interference (RI), and proactive interference (PI) for four continual learning frameworks (LwF [6], LAE [7], CLS [8], and CM) under three memory settings: no replay, experience replay (ER) [9], and cue-driven reactivation (CR). **b,** Average accuracy trajectories across tasks (curves) and interference metrics (bar charts) on GQA and PathVQA. The dashed line denotes the upper bound obtained by joint learning. The dashed line denotes the upper bound obtained by joint learning. **c,** Pairwise accuracy matrices (A_{ij}), where each entry denotes the test performance on task j after training up to task i. Retroactive interference (RI) and zero-shot interference (ZI) scores are reported above each matrix for selected baselines and CaMNet. Error bars indicate mean ± s.e.m. across 3 random seeds.

view that coordinated memory reinstatement promotes continual generalization across diverse reasoning domains.



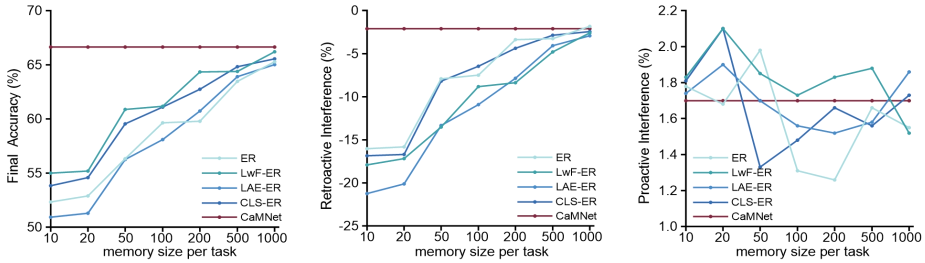
Supplementary Fig. 3 Extended mechanistic analyses of cue-driven cascaded dynamics. **a,** Retroactive interference (RI) under no reactivation, model distillation, experience replay (ER), and cue-driven reinstatement (CR) across LwF [6], LAE [7], CLS [8], and CM frameworks on GQA and PathVQA. **b,** Cosine similarity matrices of task representations across tasks on GQA and PathVQA, comparing CaMNet, baseline methods, and an ablated variant without coordination.

1.3 Extended mechanistic analyses on additional multimodal benchmarks

We next examine whether the mechanistic observations from the main text also hold on GQA and PathVQA. Supplementary Fig. 3 summarizes two additional analyses: late-reactivation recovery and representational alignment across task-specific modules.

To assess the restorative role of cue-driven reactivation (CR), we repeat the late-reactivation ablation on GQA and PathVQA, enabling reactivation only at the final task. This isolates the extent to which sparse cues can recover previously acquired knowledge after substantial degradation. As shown in Supplementary Fig. 3a, the effectiveness of late reactivation depends strongly on the underlying memory organization. In monolithic architectures such as LwF [6], CR yields only partial recovery. On GQA, LwF-CR improves RI from –43.9% to –36.0%, and on PathVQA from –51.2% to –18.7%, but remains substantially below ER [9]. By contrast, the cascaded memory (CM) architecture responds much more strongly to CR. On PathVQA, CM-CR reaches RI = –7.5%, approaching ER (–5.5%) and substantially improving over the no-reactivation baseline (–51.2%). A similar pattern is observed on GQA. These results further support the view that sparse textual cues become substantially more effective when combined with hierarchical cascaded organization.

We also analyze cosine similarity matrices between task-specific modules to examine the representational basis of this behavior. Supplementary Fig. 3b shows that the CaMNet variant without coordination exhibits a pronounced separation between the first module and later modules, indicating that early structure remains weakly connected to subsequent representations. With full coordination, by contrast, off-diagonal



Supplementary Fig. 4 Performance under varying replay memory budgets on VQAv2. Line plots show final accuracy (left), retroactive interference (middle), and proactive interference (right) for replay-based baselines (ER, LwF-ER, LAE-ER, and CLS-ER) as the memory size per task increases from 10 to 1000 stored samples. CaMNet (red) does not use an episodic replay buffer and is shown as a constant reference. Increasing replay capacity steadily improves final accuracy and retroactive interference for replay-based methods. Even at the largest memory budget, however, CaMNet remains higher in final accuracy and less negative in retroactive interference, while maintaining broadly comparable proactive interference without storing past samples.

similarity is preserved across both early and late modules, showing that later representations remain aligned with the initial scaffold rather than drifting into isolated subspaces. This more coherent geometry is consistent with schema-dependent assimilation, in which new knowledge is incorporated into an existing representational framework rather than learned in isolation. Together, these supplementary analyses indicate that the recovery and alignment mechanisms identified in the main text generalize across additional multimodal benchmarks.

1.4 Extended comparisons with replay-based baselines

In the main text, we compare CaMNet with replay-based methods under a standard memory budget of 20 samples per task. To further assess the data efficiency of our approach, we extend this comparison by varying the replay buffer size from 10 to 1000 samples per task for representative replay-based baselines, including ER [9], LwF-ER [6], LAE-ER [7], and CLS-ER [8]. Results on the VQAv2 benchmark are shown in Supplementary Fig. 4.

As expected, increasing the replay budget steadily improves both final accuracy and retroactive interference for all replay-based methods. Nevertheless, CaMNet, which operates without episodic replay and instead relies on cue-driven reinstatement, remains competitive across all three metrics. In particular, its final accuracy exceeds that of all replay-based baselines even when they are allocated 1000 stored samples per task, and its retroactive interference remains less negative than the best-performing replay variant at the same memory budget. The reported proactive interference remains broadly comparable across methods, indicating that CaMNet preserves stable plasticity without requiring large replay buffers.

These results show that the benefits of CaMNet are not explained by matching the storage scale of replay-based approaches. Even when replay methods are given substantially larger memory budgets, structured internal reinstatement remains a highly data-efficient alternative to explicit sample buffering.

Supplementary Tab. 1 Overall performance on VQAv2 using the multimodal large language model BLIP-2 [11]. Results are reported under the same continual task ordering as in the main experiments.

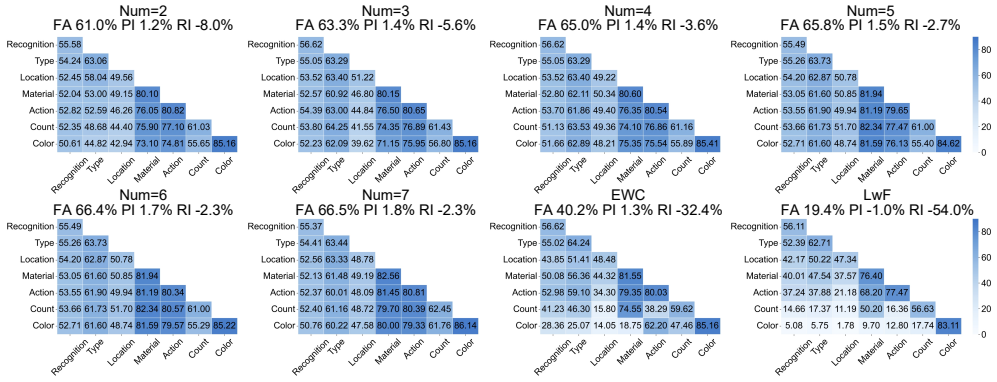
Method	*Recognition* → *Type* → *Location* → *Material* → *Action* → *Count* → *Color*		
	Final Accuracy	Proactive Interference (PI)	Retroactive Interference (RI)
Joint Learning	53.76	-	-
Continual-LoRA [10]	41.09	**4.53**	–19.03
MoE [12]	41.74	3.70	–18.21
ZAF [13]	49.82	3.87	–7.20
ER [9]	47.77	2.22	–8.53
LwF-ER [6]	49.64	2.71	–6.92
LAE-ER [7]	49.31	2.52	–6.83
CLS-ER [8]	49.42	2.60	–7.05
GEM [14]	46.70	0.64	–6.45
EWC [15]	44.35	3.18	–10.82
LwF [6]	43.37	2.69	–14.22
CM-CR	50.36	2.96	–6.37
CaMNet (Ours)	**50.89**	2.82	**–5.59**

1.5 Performance on multimodal large language models

To examine whether the proposed approach scales to larger multimodal backbones, we further evaluate CaMNet on BLIP-2 [11] under the same continual task ordering used for VQAv2. Supplementary Tab. 1 summarizes the overall results.

Across replay-free baselines, CaMNet achieves the highest final accuracy. In particular, it outperforms regularization-based methods such as EWC [15] (44.35%) and LwF [6] (43.37%), indicating that cascaded coordination remains effective even with a larger multimodal language model backbone. CaMNet also exceeds replay-based methods such as ER [9] (47.77%) and GEM [14] (46.70%), reaching a final accuracy of 50.89% while narrowing the gap to the joint-learning upper bound (53.76%) without storing past samples.

This performance is accompanied by favorable interference behavior. CaMNet attains the lowest retroactive interference among all compared continual learning methods (RI = −5.59%), indicating the strongest retention of previously learned tasks, while maintaining positive proactive interference (PI = 2.82%). The cue-driven variant CM-CR also performs strongly, achieving 50.36% final accuracy with RI = −6.37% and PI = 2.96%. Together, these results suggest that cascaded memory coordination with cue-driven reinstatement remains effective at the scale of multimodal large language models, providing a replay-free alternative when memory, privacy, or storage constraints limit explicit buffering.



Supplementary Fig. 5 Ablation study on memory capacity constraints. Pairwise accuracy matrices on VQAv2 for CaMNet with different numbers of memory modules ($N = 2$ to 7), together with EWC and LwF for reference. As N increases, final accuracy (FA) improves, retroactive interference (RI) decreases, and proactive interference (PI) increases modestly. Even with only two modules, CaMNet remains substantially stronger than EWC and LwF baselines.

2 Extended Analysis and Ablation Studies

2.1 Effect of memory capacity on interference dynamics

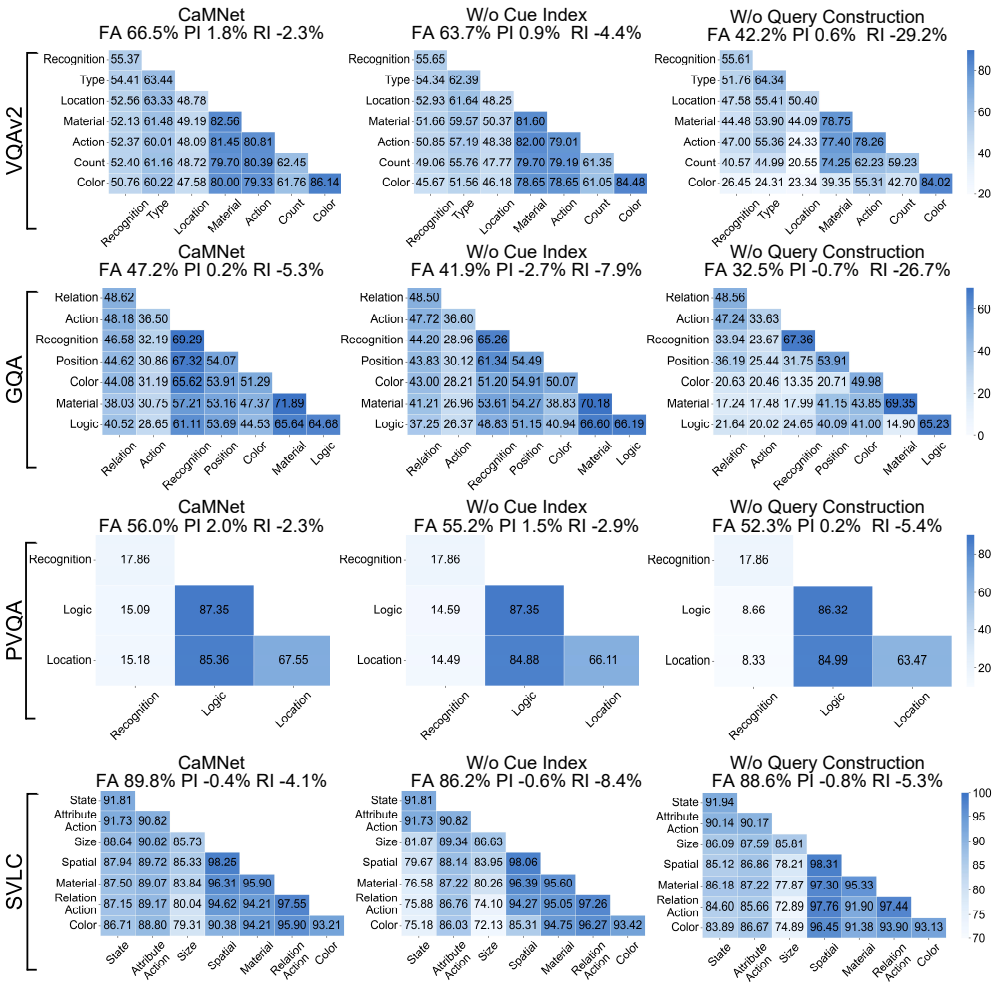
We investigate how memory capacity affects continual learning behavior by varying the number of available memory modules, denoted by N (shown as *Num* in Supplementary Fig. 5). On the VQAv2 benchmark, we vary N from 2 to 7, corresponding to progressively larger memory capacity relative to the task sequence.

Supplementary Fig. 5 shows a clear relationship between memory capacity and interference. As N increases, final accuracy (FA) improves and retroactive interference (RI) is reduced. Specifically, FA increases from 61.0% at $N = 2$ to 66.5% at $N = 7$, while RI improves from -8.0% to -2.3%. Proactive interference (PI) also increases modestly, from 1.2% to 1.8%, indicating that additional capacity slightly improves plasticity as well as retention.

At the same time, these gains exhibit diminishing returns. Beyond moderate capacity (for example, $N = 5$), further expansion yields only marginal improvements in both FA and RI. This pattern suggests that continual generalization in CaMNet is not driven simply by unbounded parameter growth, but by effective coordination and reuse of memory modules.

Importantly, CaMNet remains substantially stronger than replay-free baselines even under severe capacity constraints. With only two memory modules, it achieves 61.0% FA, compared with 40.2% for EWC [15] and 19.4% for LwF [6]. Thus, the advantage of CaMNet cannot be explained by storage capacity alone. Rather, the cascaded architecture and cue-guided coordination allow the model to maintain selective access to relevant prior knowledge even when representational resources are limited.

Taken together, these results show that memory capacity and interference resolution are closely related, but that CaMNet derives its effectiveness from structured coordination rather than capacity expansion alone.



Supplementary Fig. 6 Ablation study on explicit memory addressing and cue-guided query construction. Removing the cue index (middle column) tests the role of explicit memory addressing, whereas replacing structured query construction with naive cue concatenation (right column) tests the importance of semantically coherent probing. Across benchmarks, both ablations reduce final accuracy and worsen interference, with the largest degradation observed on the more complex reasoning benchmarks VQAv2 and GQA.

2.2 Importance of explicit memory addressing and cue-guided query construction

To clarify the contribution of cue-driven coordination, we ablate two components of CaMNet: explicit memory addressing through task-specific cue indices, and structured cue-guided query construction. Supplementary Fig. 6 summarizes the resulting changes in final accuracy (FA), proactive interference (PI), and retroactive interference (RI) across VQAv2, GQA, PathVQA, and SVLC.

Effect of explicit memory addressing.

In CaMNet, each task is associated with a cue index that allows the coordination hub to selectively probe the corresponding memory pathway. To test the importance of this addressing mechanism, we remove the cue index and instead rely on non-selective reuse of prior memory states. This ablation degrades performance across all benchmarks. On VQAv2, FA decreases from 66.5% to 63.7%, and on GQA, PI shifts from 0.2% to –2.7%, indicating reduced forward transfer. Memory retention is also weakened: on SVLC, RI worsens from –4.1% to –8.4%. These results indicate that explicit memory addressing is important for targeted reactivation and for limiting interference between old and new tasks.

Effect of cue-guided query construction.

We next examine the role of structured query construction by replacing the cue-guided query with a naive concatenation of the cue and input text. This ablation produces a much larger degradation, especially on the more complex reasoning benchmarks. On VQAv2 and GQA, RI deteriorates to –29.2% and –26.7%, respectively, accompanied by substantial drops in FA. This pattern suggests that effective reactivation depends not only on retrieving the correct memory pathway, but also on constructing a semantically coherent probe that remains aligned with the current input. By contrast, the effect is smaller on PathVQA and SVLC, where task structure is simpler and the language space is more constrained. Even in these settings, however, the full CaMNet configuration remains the strongest overall.

Together, these ablations show that cue-driven coordination depends on both components: explicit addressing determines which prior memory should be engaged, whereas query construction determines how that memory is functionally probed. Removing either component weakens continual performance, but disrupting query construction is especially harmful in benchmarks requiring more diverse and compositional reasoning.

2.3 Impact of the consolidation factor α

We examine the effect of the consolidation factor α in the fast–slow memory mechanism, which controls the update rate of the slow memory relative to the fast task-adaptive parameters. Results are summarized in Supplementary Tab. 2. Across the tested range, CaMNet remains robust and consistently competitive relative to its CM-CR counterpart, indicating that the dual-timescale mechanism improves performance over a broad set of consolidation rates.

Varying α reveals a clear trade-off between plasticity and stability. Smaller values of α promote faster integration of newly learned parameters into the slow memory, improving adaptation and final accuracy but also increasing interference. For example, at $\alpha = 0.80$, CaMNet achieves the highest final accuracy (FA = 66.73) and the strongest proactive interference (PI = 1.89), but also shows the most negative retroactive interference (RI = -2.38). By contrast, larger values of α slow consolidation and reduce interference. As α increases to 0.95, RI improves to -0.23, but this comes with reduced plasticity (PI = 0.63) and lower final accuracy (FA = 65.76).

Consolidation Parameter α	0.80	0.825	0.85	0.875	0.90	0.925	0.95	CM-CR
Final Accuracy	66.73	66.03	66.64	66.41	66.42	66.12	65.76	65.66
Proactive Interference	1.89	1.85	1.70	1.21	0.94	0.81	0.63	1.5
Retroactive Interference	–2.38	–2.32	–2.12	–1.56	–1.13	–0.89	–0.23	–2.64

Supplementary Tab. 2 Effect of the consolidation factor α on continual learning performance. Varying α reveals a trade-off between plasticity and stability: smaller values favor higher proactive interference and final accuracy, whereas larger values reduce retroactive interference.

Intermediate values provide a balanced operating regime, maintaining high final accuracy together with positive forward transfer and controlled forgetting. In our experiments, we use $\alpha = 0.85$ as the default setting, as it preserves near-maximum final accuracy while avoiding the more aggressive stability–plasticity bias associated with the extreme settings. Overall, these results indicate that α governs a smooth continuum between rapid adaptation and long-term stability, rather than admitting a single universally optimal value.

Supplementary Tab. 3 Robustness evaluation across an alternative task ordering on VQAv2. Final accuracy (FA), proactive interference (PI), and retroactive interference (RI) are reported for a reversed conceptual curriculum.

Method	*Color* → *Material* → *Count* → *Location* → *Action* → *Type* → *Recognition*		
	Final Accuracy	Proactive Interference	Retroactive Interference
Joint Learning	67.72	-	-
Continual-LoRA [10]	48.28	–3.90	–16.90
MoE [12]	44.51	–4.54	–20.54
ZAF [13]	51.76	**0.68**	–18.18
ER [9]	58.77	–0.90	–6.99
LwF-ER [6]	56.88	–1.34	–6.34
LAE-ER [7]	56.64	–1.67	–6.98
CLS-ER [8]	59.97	–1.91	–5.58
GEM [14]	59.08	–0.64	-6.10
EWC [15]	48.95	–6.57	-16.50
LwF [6]	53.30	–1.29	–14.09
CM-CR	62.27	–0.75	–1.75
CaMNet (Ours)	**63.69**	–0.62	**-1.57**

2.4 Robustness of CaMNet to task orderings

To assess whether the effectiveness of CaMNet depends on a particular curriculum, we evaluate it under alternative task orderings that differ substantially from those used in the main text. In VQAv2, we use the reversed conceptual sequence *Color* → *Material* → *Count* → *Location* → *Action* → *Type* → *Recognition*. In SVLC, we use

Supplementary Tab. 4 Robustness evaluation across an alternative task ordering on SVLC. Final accuracy (FA), proactive interference (PI), and retroactive interference (RI) are reported for a permuted conceptual curriculum.

Method	*Spatial* → *Material* → *State* → *Attribute_Action* → *Size* → *Relation_Action* → *Color*		
	Final Accuracy	Proactive Interference	Retroactive Interference
Joint Learning	90.05	-	-
Continual-LoRA [10]	69.55	–0.23	–27.25
MoE [12]	72.50	–0.04	–23.74
ZAF [13]	88.34	**0.38**	–5.38
ER [9]	87.84	–0.49	–5.42
LwF-ER [6]	88.07	–0.62	–5.78
LAE-ER [7]	88.84	–0.77	–5.79
CLS-ER [8]	88.91	–0.70	–5.13
GEM [14]	88.58	–0.42	–5.66
EWC [15]	75.19	–0.38	–21.47
LwF [6]	73.00	–0.10	–23.1
CM-CR	88.79	–0.07	–4.88
CaMNet (Ours)	**89.19**	–0.14	**–4.52**

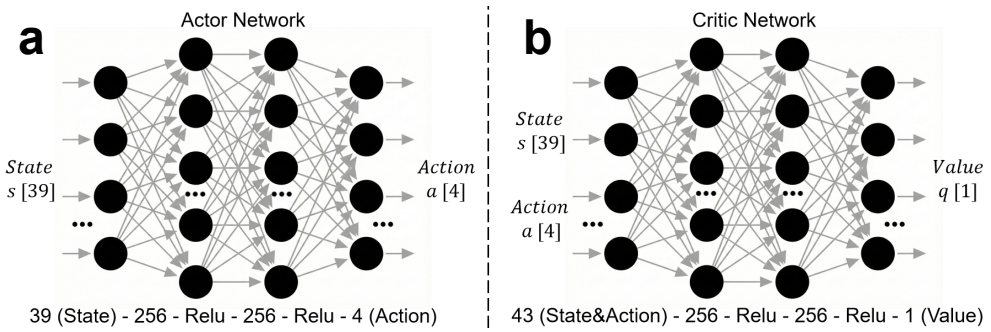
the permuted sequence *Spatial* → *Material* → *State* → *Attribute_Action* → *Size* → *Relation_Action* → *Color*.

Results are reported in Supplementary Tab. 3 and Supplementary Tab. 4. Across both benchmarks, CaMNet remains highly robust under these alternative curricula. On VQAv2, it achieves the highest final accuracy (FA = 63.69%) and the lowest retroactive interference (RI = –1.57%) among all evaluated methods, outperforming replay-based methods such as ER [9] (58.77%, RI = –6.99%) and regularization-based baselines such as EWC [15] (48.95%, RI = –16.50%). On SVLC, CaMNet attains 89.19% final accuracy with limited forgetting (RI = –4.52%), remaining competitive with or better than all baselines under the reordered curriculum.

Importantly, this strong retention is not achieved at the expense of plasticity. Across both datasets, CaMNet maintains proactive interference values close to zero while remaining non-destructive (VQAv2: –0.62; SVLC: –0.14), indicating stable acquisition of new tasks without severe forward inhibition. These results show that CaMNet is robust to substantial task-order variations and does not rely on a specific curriculum to maintain strong continual learning performance.

3 Implementation details

We implement CaMNet and all baseline methods using the PyTorch framework. All experiments are conducted on NVIDIA A100 (80GB) and RTX 4090 GPUs. Our implementation covers two domains: Continual Reinforcement Learning (CRL) and Multimodal Vision–Language Continual Learning. In the following, we describe the model architectures, baseline configurations, and training hyperparameters in detail to ensure reproducibility.



Supplementary Fig. 7 Overview of the network framework. (**a**) The Actor network. (**b**) The Critic network.

3.1 Continual reinforcement learning

3.1.1 Model framework

For CRL experiments, we adopt the Soft Actor-Critic (SAC) algorithm [16] as the underlying reinforcement learning solver for each task. Both the policy (Actor) and value (Critic) functions are parameterized as multi-layer perceptrons (MLPs) with identical hidden geometries (two hidden layers of 256 units with ReLU activations), while differing in their input–output specifications:

- **Actor Network (Supplementary Fig. 7a).** The Actor takes as input a 39-dimensional state vector s, consisting of robot proprioceptive information and object coordinates (without visual observations). It outputs the parameters of a Gaussian distribution over the 4-dimensional action space a, which controls 3D end-effector displacement and gripper actu ation.
- **Critic Network (Supplementary Fig. 7b).** The Critic estimates the soft Q-function $Q(s, a)$. It receives a concatenated input of the 39-dimensional state and the 4-dimensional action, yielding a 43-dimensional input vector, which is mapped to a scalar Q-value q through the hidden layers. To mitigate overestimation bias, we employ clipped double Q-learning, instantiating two independent critic networks with identical architectures but different random initializations.

Crucially, the progressive memory expansion of the cascaded memory architecture is applied exclusively to the hidden layers of the Actor network. We follow a multi-head design commonly used in CRL: while intermediate representations expand in a cascaded manner to preserve task-specific features, each task is associated with an independent output head that maps the shared cascaded features to the action space of that task. The Critic networks are used solely for value estimation during training and do not participate in the cascaded memory expansion.

3.1.2 Baselines

To rigorously evaluate the effectiveness of cascaded memory coordination, we integrate the proposed Cascaded Memory (CM) architecture into a representative set of CRL

baselines and compare each baseline with its CM-augmented counterpart. All methods share the same network architectures, training protocol, and random seeds to ensure a fair comparison.

- **Naive baseline.**
  - *Fine-tuning* sequentially trains a single network across tasks without any mechanism to mitigate interference. It serves as a lower bound for performance and highlights the extent of retroactive interference under unconstrained learning.
- **Architecture-based methods.**
  - *Incremental Networks (IN)* [17] dynamically expand network capacity by allocating additional parameters for new tasks, thereby reducing interference through parameter separation.
  - *Progressive Neural Networks (PNN)* [18] freeze parameters associated with previously learned tasks and instantiate a new network column for each new task. Forward transfer is enabled through lateral connections to earlier columns, while backward adaptation is explicitly prevented.
- **Regularization-based methods.**
  - *Progress & Compress (P&C)* [2] decouples learning into a fast progress phase and a slow compress phase using a dual-network architecture. Knowledge acquired by the progress network is consolidated into a shared knowledge base via regularization, such as EWC [15] or knowledge distillation [19]. Following standard practice, the compression step is performed every 10^6 environment steps.
- **Replay-based methods.**
  - *Behavioral Cloning (BC)* [3] maintains an expert replay buffer containing transition samples from previous tasks. A distillation objective encourages the current policy to reproduce behaviors stored in the buffer, thereby preserving historical knowledge. We implement BC using a KL-divergence-based distillation loss and set the expert buffer size to 10,000 transitions each task.

3.1.3 Implementation details

All models are optimized using the Adam optimizer. Detailed hyperparameters for general training and algorithm-specific configurations are provided in Supplementary Tab. 5.

3.2 Multimodal continual learning

3.2.1 Model framework

For all vision–language continual learning experiments, we adopt the pre-trained BLIP framework [20] as the pre-trained backbone and initialize all models from the official checkpoints. Depending on the downstream task, we consider two architectural

Supplementary Tab. 5 Hyperparameter settings for Continual Reinforcement Learning experiments.

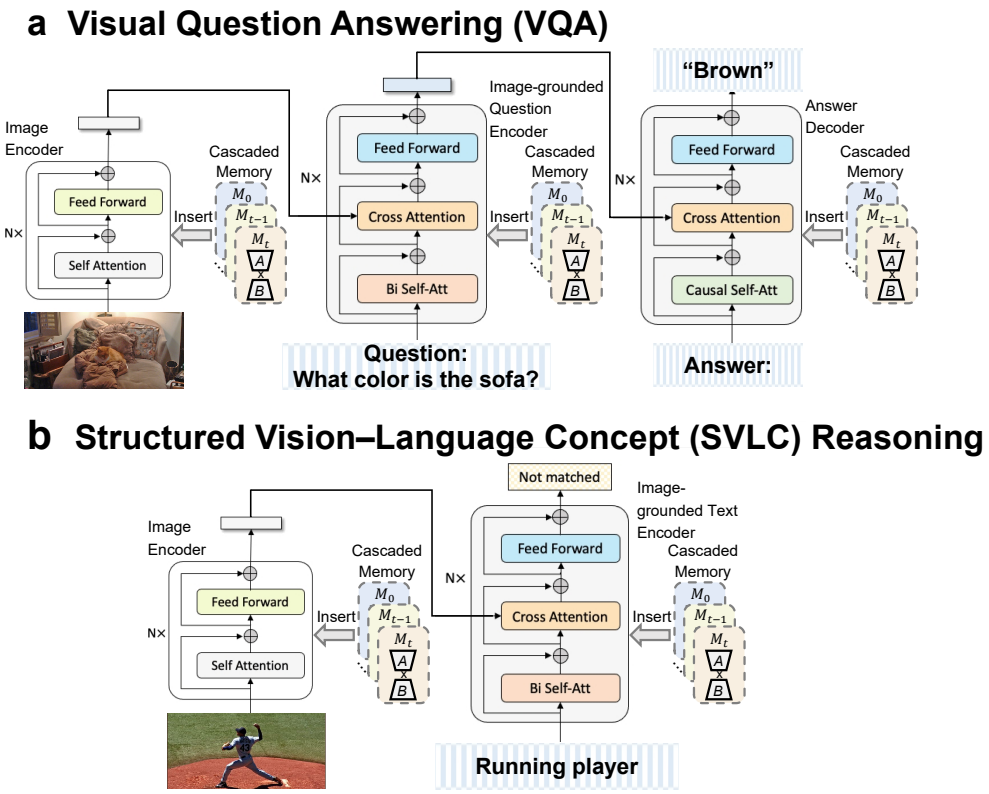
Description	Value
General Hyperparameters	
Maximum episode length	500
Environment steps per task	3×10^6
Evaluation steps	1×10^5
Gradient updates per environment step	1
Discount factor	0.99
Algorithm-Specific Hyperparameters	
Policy learning rate	3×10^{-4}
Q-function learning rate	3×10^{-4}
Replay buffer size	10^6
Mini batch size	64
Policy min. std	e^{-20}
Policy max. std	e^2
Soft target interpolation	5×10^{-3}
Entropy coefficient	Automatic Tuning

variants: a generative formulation for visual question answering (VQA) (Supplementary Fig. 8a) and a discriminative formulation for structured vision–language concept (SVLC) reasoning (Supplementary Fig. 8b).

VQA formulation.

VQA is modeled as an open-ended text generation problem and consists of three components.

- **Image encoder (f_ν):** The visual encoder is a ViT-B/16 transformer with 12 layers, 12 attention heads, and a hidden dimension of 768, which extracts patch-level visual representations from the input image.
- **Image-grounded question encoder (f_q):** The question encoder follows the BERT-base architecture. To enable multimodal fusion, cross-attention layers are inserted in each transformer block, allowing textual representations to attend to visual features produced by f_ν.
- **Answer decoder (h_ω):** The decoder adopts a causal Transformer architecture and generates answers autoregressively. Its cross-attention modules attend to the multimodal representations output by f_q, ensuring that generation is conditioned on integrated image–text features. A language modeling head projects the decoder outputs to the vocabulary space for next-token prediction.



Supplementary Fig. 8 Overview of the model framework. (**a**) The visual question answering (VQA) task. (**b**) The structured vision–language concept (SVLC) reasoning task. Adapted from Ref. [20].

SVLC formulation.

SVLC is formulated as a binary classification task that determines whether a structured textual description matches the visual input. The architecture shares the same image encoder (f_ν) as the VQA setup and includes:

- **Image-grounded text encoder (f_t):** A BERT-base encoder augmented with cross-attention layers to fuse visual features into textual representations. The final multimodal representation is obtained from the CLS token.
- **Image–text matching head (h_ψ):** A lightweight multilayer perceptron ($768 \to 768 \to 2$) that produces logits for binary match/non-match prediction.

Parameter-efficient memory instantiation.

To enable task-specific adaptation with minimal overhead, memory modules in CaMNet are instantiated using Low-Rank Adaptation (LoRA) [10] with rank $r = 16$. This design introduces only a small fraction of additional trainable parameters. For

VQA, the memory modules contain 10.03M parameters compared with the 361.48M-parameter frozen backbone (2.77%). For SVLC, the memory overhead is 6.19M parameters over a 223.94M backbone (2.76%).

LoRA adapters are injected uniformly across the architecture. In the image encoder, LoRA is applied to all linear projections in both self-attention and feed-forward layers. In the text encoders and decoders, LoRA modules are attached to the query, key, and value projections of self- and cross-attention layers, as well as to the feed-forward network projections. This uniform deployment ensures consistent task adaptation while preserving the integrity of the frozen backbone.

3.2.2 Baselines

To evaluate the effectiveness of CaMNet under a controlled and fair setting, we compare it against a comprehensive suite of continual learning baselines. All methods share the same backbone architecture, training protocol, and random seeds. To strictly control parameter efficiency, all baselines—except for the fully fine-tuned *Continual-FT* and the adapter-based *MoE*—are implemented using LoRA as the underlying adaptation mechanism. The evaluated methods are grouped as follows:

- **Naive baseline.**
  - *Continual Fine-tuning (CFT)*: A naive fine-tuning strategy that sequentially updates the entire backbone on new tasks without any regularization or memory mechanism. This baseline serves as a lower bound for catastrophic forgetting.
- **Parameter-efficient adaptation methods.**
  - *Continual-LoRA (CLA)* [10]: A parameter-efficient continual learning baseline that updates only low-rank adaptation matrices [10] while keeping the backbone frozen.
  - *MoE-Adapters (MoE)* [12]: A modular architecture that introduces Mixture-of-Experts [21] adapters to disentangle task representations via sparse expert routing.
- **Regularization-based methods.**
  - *Learning without Forgetting (LwF)* [6]: A distillation-based approach that constrains the current model to match the outputs of previous task models using only data from the current task.
  - *Elastic Weight Consolidation (EWC)* [15]: A synaptic regularization method that penalizes changes to parameters deemed important for previous tasks, as estimated by the Fisher Information Matrix.
- **Replay-based methods.**
  - *Experience Replay (ER)* [9]: A rehearsal-based method that stores a buffer of past samples and replays them alongside current task data.
  - *Learning without Forgetting with ER(LwF-ER)* [6]: An extension of LwF [6] that performs distillation using replayed samples rather than relying solely on current task data.

– *Learning Accumulation Ensemble with ER (LAE-ER)* [7]: A replay-augmented variant of the Learning Accumulation Ensemble framework, which introduces a slow-learning model to accumulate knowledge via distillation on stored samples.
– *Complementary Learning Systems with ER (CLS-ER)* [8]: A dual-model approach inspired by complementary learning systems theory [22], employing fast and slow learners to consolidate knowledge through replay.
– *Gradient Episodic Memory (GEM)* [14]: A constraint-based replay method that enforces gradient updates not to increase the loss on stored episodic memories.
– *Zero-shot Antidote to Forgetting (ZAF)* [13]: A rehearsal-based approach that mitigates forgetting by replaying pre-processed, task-aligned wild data instead of raw historical samples.

3.2.3 Implementation details

To ensure a rigorous and reproducible comparison, we align our training configuration with established protocols in recent continual vision–language studies [13, 23]. Unless otherwise specified, all experiments use a batch size of 32 and are optimized with AdamW (weight decay 0.05). A cosine annealing scheduler is employed to adjust the learning rate throughout training. For standard VQA benchmarks (VQAv2 and GQA), models are trained for 15 epochs per task, while for the medical PathVQA benchmark, the training duration is extended to 30 epochs per task to account for higher task complexity. The regularization coefficient λ_{CR} is set to 1.0 for VQAv2, SVLC, and PathVQA, and reduced to 0.1 for GQA following prior practice.

Learning rate and parameterization.

We differentiate learning rate schedules based on whether methods employ parameter-efficient adaptation or full-parameter optimization. For CaMNet and all LoRA-based baselines, the initial learning rate is set to 1.25×10^{-3} for VQAv2, GQA, and SVLC, and to 2.0×10^{-3} for PathVQA to ensure stable low-rank adaptation. For the Continual-FT baseline, which updates the entire backbone, a smaller learning rate of 3×10^{-5} is used. All LoRA-based methods use a fixed rank of $r = 16$. For replay-based baselines (e.g., ER [9] and GEM [14]), the episodic memory buffer is restricted to 20 samples per task in standard benchmarks. In the embodied humanoid setting, the buffer is further limited to 5 samples per task to reflect the few-shot regime. The consolidation factor α governing CaMNet's fast–slow integration is fixed to 0.85 across all main experiments.

Embodied humanoid setting.

In the embodied experiments, we mitigate overfitting risks inherent to few-shot adaptation by interleaving a small subset of samples from the final VQAv2 task (*Color*) into the training stream of subsequent tasks, thereby stabilizing feature representations. Training is restricted to a single epoch per task, and the consolidation coefficient is reduced to $\alpha = 0.2$ to preserve sufficient plasticity under limited optimization steps. All embodied data collection and evaluation are conducted on a Unitree G1 humanoid robot, equipped with an NVIDIA Jetson Orin NX edge processor for on-device inference and an Intel RealSense D435i camera for visual perception.

Supplementary Tab. 6 The complete set of task-specific cues used for query generation in VQAv2, GQA, and PathVQA benchmarks. Note that xxx serves as a placeholder for object names or specific attributes.

Benchmark	Category	Cue Templates
VQAv2	Recognition	What is the name of xxx; What is xxx; Who is xxx; What are the xxx; What does the xxx; What is on the xxx; What is in the xxx; Which one is xxx
	Type	What type of xxx; What kind of xxx
	Location	Where is the xxx; Where are the xxx
	Material	What material is the xxx made of; What material are the xxx made of
	Action	Are the xxx doing sth.; Is the xxx doing sth.; What is the xxx doing sth.; What are the xxx doing sth.; Are xxx doing sth.; Is xxx doing sth.
	Count	How many xxx; How many are the xxx
	Color	What color is xxx; What color are xxx
GQA	Relation	What is the name of xxx; Where is xxx; Is the xxx; What is xxx; What's xxx; What kind of xxx; Who is xxx; What are the xxx
	Action	What is the xxx doing sth.; What's the xxx doing sth.; What are the xxx doing sth.
	Recognition	What type of; What kind of; What is xxx doing sth.; Which xxx is it; Is this
	Position	On which side of the xxx is the xxx; On which side of the xxx; Which side is xxx on; Is the xxx in the xxx part of the xxx; Are the xxx in the xxx part of the xxx; Is the xxx on the left; Is the xxx on the right
	Color	What color are the; What color is the; Of what color is xxx; What is the color of the xxx; Which color does the xxx have; Which color do the xxx have; Which color are the xxx have; Which color is the xxx have
	Material	What material is the xxx made of; What material are the xxx made of; Which material is the xxx made of; Which material are the xxx made of; What are the xxx made of; What is the xxx made of; What makes up the xxx; Are the xxx and the xxx made of the same material; Is the xxx made of xxx
	Logic	What do both the xx and the xx have in common; Is the xxx than the xxx; Are the xx and xxx; Are all the xx; Are both the xx and xxx the same/different xxx; What is common to the xxx and xxx; Do the xxx and xxx have xxx in common; Do the xxx and xxx have different xxx
PathVQA	Recognition	What is xxx; What is the xxx; What are xxx; What are the xxx; What is present; What does; What shows; What was xxx; What was the xxx; What were xxx; What were the xxx; What is characterized by xxx; What xxx shows; What illustrates; What has; What is seen; What is there; What are there
	Logic	Are the; Is the; Is xxx present; Do xxx present; Is this; Is there; Does xxx show; Do the xxx show; Does this xxx show; Does this image
	Location	Where is; Where are; Where is this; Where is this from; Where does this belong to; Where is this part in the figure
Real-word Embodied Task	Recognition	What is this?
	Count	How many xxx are there?
	Material	What is the xxx made of?
	State	Is there anything on the xxx? Is there anything in the xxx? Is xxx in the usage?

Cue specification.

Cue-driven reinstatement relies on a predefined set of textual cues corresponding to semantic task categories. Specifically:

- **VQAv2** [24]: 24 cues spanning 7 categories, including concise templates for *Color*, *Material*, *Count*, *Location*, and *Type* (2 cues each), and more diverse formulations for *Action* (6 cues) and *Recognition* (8 cues).
- **GQA** [4]: 49 cues across 7 reasoning types, with richer coverage for *Relation* and *Material* (9 cues each), followed by *Logic* and *Color* (8 cues each), *Position* (7 cues), *Recognition* (5 cues), and *Action* (3 cues).

- **PathVQA** [5]: 36 medical-domain cues, grouped into *Judge* (10 cues), *Where* (6 cues), and *What* (20 cues) templates.
- **Embodied data**: 7 cues spanning *Recognition*, *Count*, *Material*, and *State*.

Representative examples include `'What material is the [object] made of?'` for material queries and `'On which side of the [object] is the [object]?'` for Spatial Relation reasoning. The full list of cue templates is provided in Supplementary Tab. 6.

Cue-guided query construction.

To transform static cues into task-aligned queries, we employ a generative language model. In all experiments, we use Qwen2.5-3B Instruct [25] to produce grammatically coherent, context-aware queries.

For VQA benchmarks, the model is prompted to rewrite an existing question to match a target cue type:

> *"Generate a new question for me, replace the following question with one that begins with '{cue}', and make it grammatically correct: '{question}'. Only provide the new question, which must differ from the original."*

For the SVLC benchmark, existing textual descriptions are transformed using:

> *"Generate a new description for me, replace the following description with one that begins with the concept word '{cue}', and make it as short as possible (no more than five words): '{description}'. Only provide the new description, which must differ from the original."*

Importantly, the cue-driven mechanism is agnostic to the choice of the underlying language model. While Qwen2.5-3B-Instruct is used for experimental consistency, query generation can be delegated to alternative large language models or external API services. This decouples language reasoning from local inference, avoiding additional storage or computational overhead on the deployed system.

4 Benchmark descriptions

Continual reinforcement learning benchmarks.

For continual reinforcement learning (CRL), we evaluate cascaded memory architecture on the Meta-World benchmark [26], which consists of a diverse set of robotic manipulation tasks requiring precise continuous control. To examine performance under different levels of task interference, we construct two task sequences with distinct curriculum characteristics. The primary sequence, used in the main text, follows a challenging ordering:

$$Faucet\,Open \rightarrow Push \rightarrow Button\,Press\,Topdown \rightarrow Sweep\,Into \rightarrow Button\,Press\,Wall,$$

where successive tasks involve substantially different interaction dynamics, such as rotating articulated objects, pushing rigid bodies, and pressing buttons from varying orientations. In the supplementary experiments, we additionally consider an

Supplementary Tab. 7 Detailed statistics for VQAv2, GQA, PathVQA, and SVLC benchmarks.

Benchmark	Task	Train	Val	Test	Total
VQAv2	Recognition	147,648	6,010	6,065	159,723
	Type	26,563	1,260	1,211	29,034
	Location	14,921	710	702	16,333
	Material	4,787	203	200	5,190
	Action	35,185	1,572	1,448	38,205
	Count	68,564	2,930	2,905	74,399
	Color	57,303	2,590	2,451	62,344
GQA	Relation	80,000	5,496	5,504	91,000
	Action	10,536	893	899	12,328
	Recognition	36,548	2,593	2,585	41,726
	Position	67,340	4,913	4,929	77,182
	Color	59,013	4,521	4,567	68,101
	Material	15,851	1,354	1,409	18,615
	Logic	27,658	2,132	2,193	31,982
PathVQA	Recognition	8,083	2,565	2,753	13,401
	Logic	9,806	3,135	3,391	16,332
	Location	1,316	409	432	2,157
SVLC	State	4,716	596	568	5,880
	Attribute Action	3,930	462	539	4,931
	Size	14,357	1,883	1,776	18,016
	Spatial Relation	24,576	2,931	3,143	30,650
	Material	11,492	1,363	1,450	14,305
	Relation Action	10,036	1,331	1,366	12,733
	Color	59,916	7,520	7,685	75,121

alternative Meta-World task sequence with a milder interference profile:

$$Faucet\,Open \rightarrow Door\,Close \rightarrow Button\,Press\,Topdown\,Wall \rightarrow Handle\,Pull \rightarrow Window\,Close.$$

This ordering is adopted to reduce negative transfer between consecutive tasks, following the analysis of Ahn et al. [1], who identified curriculum-induced interference as a primary source of performance degradation in CRL. By evaluating both sequences, we assess the robustness of cascaded memory under curricula with differing interference profiles.

Multimodal continual learning benchmarks.

For vision–language continual learning, we consider two reasoning scenarios: visual question answering (VQA) and structured vision–language concept (SVLC) reasoning.

In the VQA scenario, we construct three continual benchmarks derived from VQAv2 [24], GQA [4], and PathVQA [5]. Tasks are defined based on semantic question types and organized into fixed concept-based sequences to induce controlled

distributional shifts. Specifically, VQAv2 is organized into a 7-task sequence:

$$Recognition \rightarrow Type \rightarrow Location \rightarrow Material \rightarrow Action \rightarrow Count \rightarrow Color.$$

The GQA benchmark, which emphasizes compositional reasoning, follows the sequence:

$$Relation \rightarrow Action \rightarrow Recognition \rightarrow Position \rightarrow Color \rightarrow Material \rightarrow Logic.$$

For the medical PathVQA benchmark, we adopt a 3-task sequence consisting of:

$$Recognition \rightarrow Logic \rightarrow Location,$$

reflecting the dominant reasoning patterns in pathology-focused VQA.

For the SVLC scenario, we adopt the benchmark introduced by Smith et al. [23], which is built from Visual Genome (VG) [27] and Visual Attributes in the Wild (VAW) [28]. This benchmark evaluates fine-grained visual concept understanding and is organized into a 7-task sequence:

$$State \rightarrow Attribute\ Action \rightarrow Size \rightarrow Spatial \rightarrow Material \rightarrow Relation\ Action \rightarrow Color.$$

Supplementary Tab. 7 reports the sample statistics for each benchmark and task. Across VQAv2 and GQA, tasks exhibit pronounced class imbalance (e.g., recognition-oriented tasks contain substantially more samples than material or logic reasoning), while PathVQA reflects a smaller, domain-specific dataset with limited training data. The SVLC benchmark is comparatively balanced across most concept types, with fewer samples for action- and state-related categories.

Embodied humanoid benchmark.

For embodied evaluation, we conduct real-world experiments using a humanoid robot platform. The benchmark focuses on four fundamental visual–semantic capabilities: *Recognition*, *Count*, *State*, and *Material*. Unlike large-scale synthetic or web-based datasets, this setting operates in a few-shot regime, with all data collected directly from the robot's onboard camera. In total, the dataset comprises 877 training samples and 276 test samples. The *Recognition* task spans 19 object categories and includes 385 training and 70 test samples. The *Count* task covers 9 categories with 194 training and 78 test samples. The binary *State* task uses 156 training and 65 test instances, while the *Material* task involves 12 categories supported by 142 training and 63 test samples. This constrained data regime is designed to evaluate the ability of continual learning methods to adapt efficiently under limited supervision.

References

[1] Ahn, H., Hyeon, J., Oh, Y., Hwang, B., Moon, T.: Prevalence of negative transfer in continual reinforcement learning: Analyses and a simple baseline. In: International Conference on Learning Representations (2025)

[2] Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y.W., Pascanu, R., Hadsell, R.: Progress & compress: A scalable framework for continual learning. In: Proceedings of the International Conference on Machine Learning, pp. 4528–4537 (2018)

[3] Torabi, F., Warnell, G., Stone, P.: Behavioral cloning from observation. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 4950–4957. AAAI Press, ??? (2018)

[4] Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6700–6709 (2019)

[5] He, X., Zhang, Y., Mou, L., Xing, E., Xie, P.: Pathvqa: 30000+ questions for medical visual question answering. arXiv preprint arXiv:2003.10286 (2020)

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

[7] Gao, Q., Zhao, C., Sun, Y., Xi, T., Zhang, G., Ghanem, B., Zhang, J.: A unified continual learning framework with general parameter-efficient tuning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 11483–11493 (2023)

[8] Arani, E., Sarfraz, F., Zonooz, B.: Learning fast, learning slow: A general continual learning method based on complementary learning system. In: International Conference on Learning Representations (2022)

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

[10] Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., *et al.*: Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2021)

[11] Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: Proceedings of the International Conference on Machine Learning, pp. 19730–19742 (2023)

[12] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. arXiv preprint arXiv:1701.06538 (2017)

[13] Gao, Z., Zhang, X., Xu, K., Mao, X., Wang, H.: Stabilizing zero-shot prediction: A novel antidote to forgetting in continual vision-language tasks. Advances in Neural Information Processing Systems **37**, 128462–128488 (2024)

[14] Lopez-Paz, D., Ranzato, M.A.: Gradient episodic memory for continual learning. In: Advances in Neural Information Processing Systems, vol. 30 (2017)

[15] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., *et al.*: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)

[16] Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proceedings of the International Conference on Machine Learning, pp. 1861–1870 (2018)

[17] Zhou, G., Sohn, K., Lee, H.: Online incremental feature learning with denoising autoencoders. In: Proceedings of the International Conference on Artificial Intelligence and Statistics, pp. 1453–1461 (2012)

[18] Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)

[19] Hinton, G., Vinyals, O., Dean, J., et al.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 **2**(7) (2015)

[20] Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: Proceedings of the International Conference on Machine Learning, pp. 12888–12900 (2022)

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

[22] 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)

[23] Smith, J.S., Cascante-Bonilla, P., Arbelle, A., Kim, D., Panda, R., Cox, D., Yang, D., Kira, Z., Feris, R., Karlinsky, L.: Construct-vl: Data-free continual structured vl concepts learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14994–15004 (2023)

[24] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6904–6913 (2017)

[25] Hui, B., Yang, J., Cui, Z., Yang, J., Liu, D., Zhang, L., Liu, T., Zhang, J., Yu, B., Lu, K., et al.: Qwen2.5-coder technical report. arXiv preprint arXiv:2409.12186

(2024)

[26] Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., Levine, S.: Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In: Proceedings of the Conference on Robot Learning, pp. 1094–1100 (2020)

[27] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D.A., *et al.*: Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision **123**, 32–73 (2017)

[28] Pham, K., Kafle, K., Lin, Z., Ding, Z., Cohen, S., Tran, Q., Shrivastava, A.: Learning to predict visual attributes in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13018–13028 (2021)