# AN EXPERT KNOWLEDGE COMPOSITION METHOD FOR CONTINUAL REINFORCEMENT LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Continual reinforcement learning requires an agent to continuously adapt to sequentially arriving tasks while retaining and reusing previously acquired knowledge. However, when past interaction data are no longer accessible, effectively organizing historical knowledge, identifying transferable knowledge components, and preventing the knowledge repository from growing unboundedly with the number of tasks remain key challenges. To address these issues, we propose an expert knowledge composition method for continual reinforcement learning (EKC-CRL). The proposed method represents each historical task as a group of expert knowledge vectors, enabling a more fine-grained characterization of task-specific knowledge than a single task-level representation. When adapting to a new task, the model performs two-level expert routing to select and compose relevant historical experts, while maintaining an independent set of current experts to acquire task-specific new knowledge. We further introduce routing-balancing and expert-diversity regularization to improve expert utilization and reduce knowledge redundancy across experts. As tasks arrive sequentially, learned experts are incorporated into a historical knowledge pool. Once the pool reaches its capacity limit, Hungarian matching is employed to align experts across tasks, followed by the merging of similar knowledge groups. In this way, the model can achieve flexible reuse of historical knowledge and effective new-task learning without storing past interaction trajectories. Experimental results demonstrate that our method consistently outperforms state-of-the-art approaches across multiple continual reinforcement learning benchmarks.

## 1 INTRODUCTION

Reinforcement learning (RL) enables agents to learn decision-making policies through interaction with their environments and adapt their behavior to task requirements. Whereas conventional RL typically focuses on training for a fixed task, continual reinforcement learning (CRL) requires agents to keep learning as tasks evolve while retaining and reusing previously acquired knowledge Khetarpal et al. (2022); Abel et al. (2023). As the task sequence progresses, learning new tasks may overwrite previously acquired knowledge and degrade performance on earlier tasks, a phenomenon known as catastrophic forgetting. With prolonged training, networks may also gradually lose their ability to acquire new knowledge, a phenomenon known as loss of plasticity Dohare et al. (2024); Muppidi et al. (2024). The former undermines knowledge retention, whereas the latter limits knowledge acquisition; together, they exacerbate the stability–plasticity trade-off in CRL. A central challenge in CRL is therefore to organize and reuse previously acquired knowledge effectively without continued access to historical data, while mitigating catastrophic forgetting and enhancing plasticity Parisi et al. (2019); De Lange et al. (2022).

Existing CRL methods primarily focus on retaining historical information and transferring knowledge. Experience replay methods provide ongoing training signals for previous tasks by storing, generating, or replaying past interactions Zhang et al. (2023); Shin et al. (2017); regularization-based methods mitigate catastrophic forgetting by constraining changes to parameters that are important for previous tasks Kirkpatrick et al. (2017); Zenke et al. (2017); and architecture-based methods reduce interference by allocating task-specific components or modules Mallya & Lazebnik (2018); Lee et al. (2020); Wang et al. (2025). Meanwhile, some studies have begun to explore how to explicitly preserve and reuse knowledge encoded in the parameters learned on previous tasks. For example,



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

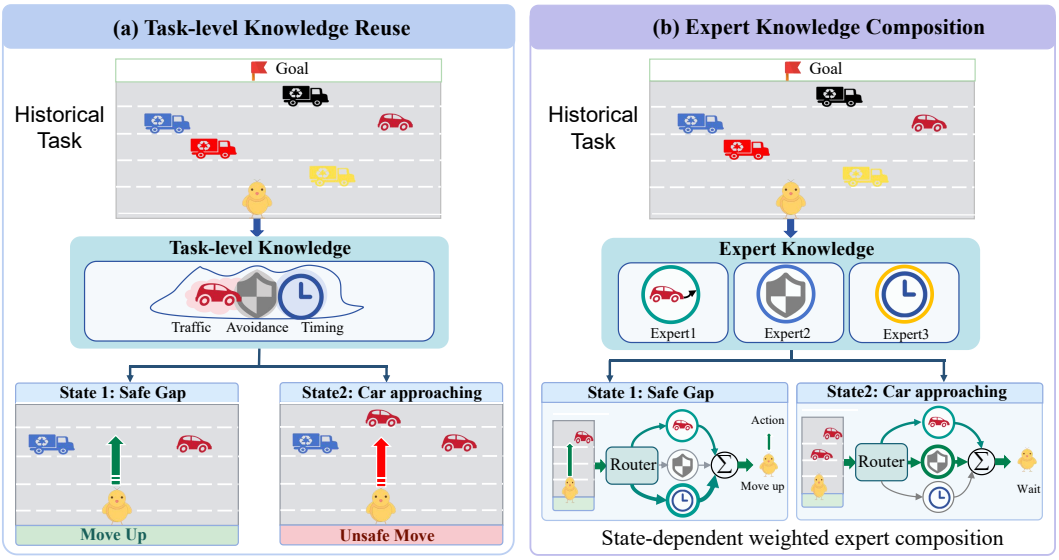



Figure 1: Motivation for expert knowledge composition. Task-level knowledge reuse represents multiple knowledge components as a single task-level representation, whereas expert knowledge composition selectively combines relevant knowledge components for more flexible knowledge reuse.

CKA-RL represents the knowledge acquired from each task as a task-specific knowledge vector and retrieves and merges historical knowledge during subsequent learning Hu et al. (2025). Modular and self-composing policy methods organize previously learned policy capabilities into composable units to support learning new tasks Mendez et al. (2022); Malagon et al. (2024); Meng et al. (2025). These methods can preserve and reuse knowledge acquired from different tasks, but they typically use tasks as the basic units of knowledge representation. Although such representations preserve differences across tasks, different types of knowledge within the same task remain bundled together, making it difficult for the model to identify and compose relevant knowledge components Jacobs et al. (1991); Shazeer et al. (2017); Hihn & Braun (2023); Obando Ceron et al. (2024).

Consider the Freeway environment illustrated in Fig. 1 Machado et al. (2018). The agent must account for factors such as traffic flow, potential collision risk, and crossing opportunities when deciding whether to move forward or wait. As shown in Fig. 1(a), task-level knowledge reuse encodes multiple types of knowledge acquired from a historical task into a single representation, which then participates as a whole in subsequent policy construction. Because the internal knowledge components cannot be adjusted independently, such a representation provides limited flexibility when only part of the historical knowledge is relevant. To enable finer-grained reuse, we instead decompose task knowledge into multiple expert knowledge components and compose them selectively, as illustrated in Fig. 1(b). By shifting the basic unit of knowledge reuse from a task-level representation to multiple expert knowledge components, the model can combine different historical knowledge components more flexibly, enabling more selective and effective knowledge reuse.

Building on this idea, we propose Expert Knowledge Composition for Continual Reinforcement Learning (EKC-CRL). EKC-CRL represents each historical task as a group of expert knowledge vectors and composes relevant historical experts for fine-grained knowledge reuse, while learning task-specific knowledge through a separate set of current experts. To encourage expert diversity and prevent routing collapse, we introduce expert-diversity regularization and a load-balancing constraint on expert routing, which promote differentiated expert representations and balanced knowledge utilization. As the task sequence grows, EKC-CRL further maintains a fixed-capacity historical knowledge pool by matching and merging similar expert knowledge groups, preventing historical knowledge storage from growing unboundedly with the number of tasks. Experiments on the evaluated CRL benchmarks with multiple random seeds show that EKC-CRL achieves higher final performance and forward transfer than the compared baselines.

Our main contributions are summarized as follows:

- We propose EKC-CRL, which represents historical task knowledge as groups of expert knowledge vectors and enables fine-grained expert knowledge composition for knowledge reuse and new-task adaptation.
- We introduce expert-diversity and routing-balancing regularization, together with a bounded-capacity knowledge merging strategy, to promote effective expert specialization and control the growth of the historical knowledge pool.
- Experiments on multiple CRL benchmarks with multiple random seeds demonstrate that EKC-CRL achieves superior final performance and forward transfer compared with the evaluated baselines.

## 2 Related Work

### 2.1 Continual Reinforcement Learning

Continual reinforcement learning (CRL) requires agents to adapt to sequentially changing tasks while retaining and reusing previously acquired knowledge Wolczyk et al. (2021); Wang et al. (2024). Existing studies have mainly focused on reducing interference between sequential tasks and improving knowledge transfer. Replay-based methods mitigate catastrophic forgetting by preserving historical experiences through stored or generated trajectories Rolnick et al. (2019); Shin et al. (2017); Guo et al. (2026); Yue et al. (2025). Regularization-based methods, such as EWC Kirkpatrick et al. (2017) and SI Zenke et al. (2017), constrain updates to parameters important for previous tasks, while architecture-based approaches, including PackNet Mallya & Lazebnik (2018) and MaskNet Ben-Iwhiwhu et al. (2023), reduce interference by isolating task-specific components. Beyond mitigating forgetting, recent CRL studies investigate how previously acquired knowledge can be transferred and reused when learning new tasks. Early approaches, such as Distral Teh et al. (2017) and successor features with generalized policy improvement (SF & GPI) Barreto et al. (2017), exploit shared structures across tasks to accelerate adaptation. More recent modular approaches represent reusable capabilities as composable units, including Modular Lifelong RL Mendez et al. (2022), CompoNet Malagon et al. (2024), and LEGION Meng et al. (2025). CKA-RL Hu et al. (2025) further explores explicit knowledge representation by encoding each task as a task-specific parameterized knowledge vector and retrieving historical knowledge during continual adaptation. However, existing approaches either represent knowledge at a coarse task level or rely on predefined reusable modules, which limits fine-grained decomposition and composition of knowledge acquired from previous tasks.

### 2.2 Mixture-of-Experts and Dynamic Routing

Mixture-of-experts (MoE) models improve model capacity and specialization by combining multiple expert networks with input-dependent routing Jacobs et al. (1991). Recent studies have developed sparse and soft routing mechanisms to efficiently activate experts, including Sparsely-Gated MoE Shazeer et al. (2017), Switch Transformer Fedus et al. (2022), and Soft MoE Puigcerver et al. (2024). Expert allocation and load balancing have also been investigated through approaches such as Expert Choice Zhou et al. (2022) and BASE Lewis et al. (2021). In reinforcement learning, MoE architectures have been explored to improve policy capacity and adaptation under complex environments. Obando Ceron et al. Obando Ceron et al. (2024), MENTOR Huang et al. (2025), and Willi et al. Willi et al. (2024) investigate expert-based policies in deep RL, visual RL, and multitask RL, respectively, demonstrating the effectiveness of routing-based specialization for non-stationary decision-making. Beyond multitask settings, expert-based architectures have also been studied in continual learning. MoVE Hihn & Braun (2023) employs dynamic gating and diversity objectives to encourage expert specialization, while Li et al. Li et al. (2025) analyze the role of expert specialization in continual learning. However, existing expert-based approaches primarily focus on learning specialized experts or reducing interference, without explicitly modeling historical knowledge as reusable expert components that can be dynamically composed across sequential tasks.

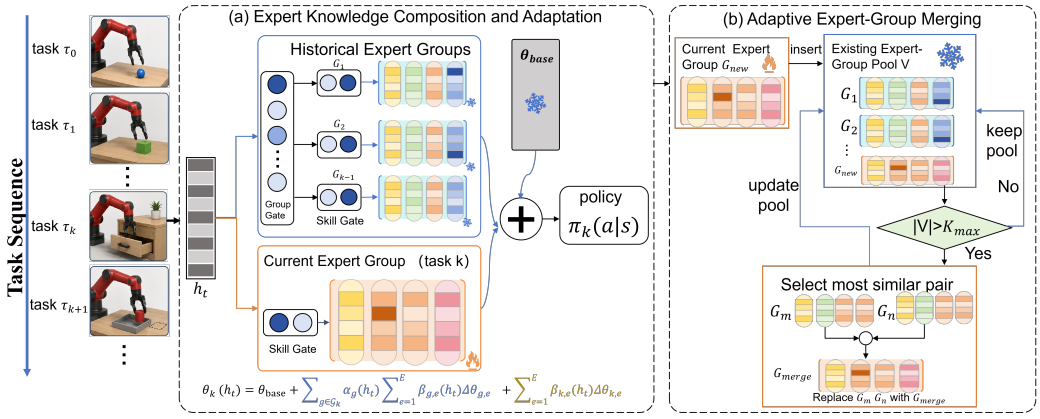



Figure 2: Overall framework of the proposed method. On task $\tau_k$, the model dynamically composes historical knowledge and current-task knowledge through expert routing. As shown in (a), historical knowledge consists of multiple expert groups and is selected through two-level routing; the current task maintains an independent set of learnable experts. The two types of knowledge, together with the fixed base parameters $\theta_{\text{base}}$, jointly constitute the current policy. As shown in (b), after a task is completed, the current expert group is added to the historical knowledge pool. When the pool exceeds its capacity limit, the most similar knowledge groups are selected for merging through expert matching.

## 3 Problem Definition

Standard reinforcement learning is formulated as a **markov decision process (MDP)**: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively. The transition probability $p(s'|s,a)$ describes the probability of reaching state $s'$ after taking action $a$ in state $s$, and $r(s,a)$ represents the immediate reward. The discount factor is denoted by $\gamma \in [0,1)$. At each timestep $t$, the agent observes the current state $s_t$, selects an action $a_t \sim \pi_\theta(\cdot|s_t)$, and transitions to the next state $s_{t+1} \sim p(\cdot|s_t, a_t)$. The policy parameters $\theta$ are optimized to maximize the expected discounted return:

$$J(\theta) = \mathbb{E}_{\pi_\theta, p}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]. \tag{1}$$

This objective measures the long-term cumulative reward obtained by the policy through interactions with the environment.

**Continual reinforcement learning (CRL)** considers a sequence of tasks that arrive sequentially: $\mathcal{T} = (\tau_0, \tau_1, \ldots, \tau_{N-1})$, where each task $\tau_k$ is associated with a Markov decision process (MDP): $\mathcal{M}_k = (\mathcal{S}_k, \mathcal{A}, p_k, r_k, \gamma)$. Here, $\mathcal{S}_k$ denotes the state space of task $\tau_k$, while $p_k$ and $r_k$ represent the transition probability and reward function, respectively. Tasks arrive according to a predefined order, and the transition time between tasks is assumed to be known. When learning the current task $\tau_k$, the agent interacts with the corresponding environment and updates its policy using data collected from the current task. Meanwhile, previously collected trajectories from earlier tasks are no longer accessible. Unlike independently optimizing each task, CRL requires the agent to continuously adapt to new tasks while retaining and reusing useful knowledge acquired from previous tasks. Therefore, a key challenge in CRL is how to effectively organize and exploit historical knowledge to facilitate future adaptation when direct access to previous interaction data is unavailable.

## 4 Expert Knowledge Composition for Continual Reinforcement Learning

We propose an expert knowledge composition framework for continual reinforcement learning, as illustrated in Fig. 2. The framework consists of historical knowledge representation and routing, current-expert adaptation, and adaptive knowledge merging, which are detailed in Sections 4.1–4.3.

### 4.1 Expert Knowledge Representation and Routing

Task-level knowledge representations may entangle multiple capabilities learned within the same task, limiting fine-grained knowledge reuse. To enable more flexible knowledge composition, we represent historical knowledge as groups of expert parameter offsets rather than a single task-level knowledge representation.

Before learning task $k$, let $\mathcal{H}_k = \{V_g\}_{g=1}^{G}$ denote the active historical knowledge pool after consolidation, where $G = |\mathcal{H}_k| \leq P$. Each knowledge group is defined as $V_g = [v_{g,1}, v_{g,2}, \ldots, v_{g,E}]$, where $E$ denotes the number of experts and $v_{g,e}$ represents the frozen knowledge offset captured by expert $e$ in historical group $g$ relative to the fixed base parameters $\theta_{\text{base}}$. A knowledge group may correspond to either a single previously learned task or a consolidated representation of multiple tasks. Compared with a single task-level representation, expert-based representation preserves more fine-grained knowledge components for subsequent composition.

When learning a new task, the model composes historical knowledge through hierarchical routing. The group-level router determines the contribution of different historical knowledge groups, while the expert-level router selects experts within each group. Meanwhile, a separate set of trainable current experts $V_k = [v_{k,1}, \ldots, v_{k,E}]$ is maintained to capture task-specific knowledge from the current task.

Let $\alpha_g(h_t)$, $\beta_{g,e}(h_t)$, and $\beta_{k,e}(h_t)$ denote the historical-group, historical-expert, and current-expert routing weights, respectively. The routing weights are normalized using softmax. The composed policy parameters for state representation $h_t$ are formulated as:

$$\theta_k(h_t) = \theta_{\text{base}} + \sum_{g=1}^{G} \alpha_g(h_t) \sum_{e=1}^{E} \beta_{g,e}(h_t) v_{g,e} + \sum_{e=1}^{E} \beta_{k,e}(h_t) v_{k,e}. \tag{2}$$

The second term represents reusable historical knowledge composed from previous tasks, while the third term captures new knowledge acquired by current experts. This formulation enables knowledge reuse while preserving adaptation to new tasks.

### 4.2 Current-Expert Adaptation

Historical experts provide reusable knowledge acquired from previous tasks, but directly adapting these experts to new tasks may overwrite previously learned capabilities and weaken knowledge retention. To preserve historical knowledge while maintaining the ability to learn new task-specific behaviors, we introduce a separate set of trainable current experts $V_k = [v_{k,1}, \ldots, v_{k,E}]$. During training, historical experts and base parameters remain fixed, while current experts are optimized using data from the current task. At the transition to the next task, the learned current experts are incorporated into the historical knowledge pool as a new knowledge group.

Since historical experts contain already optimized representations, jointly normalizing historical and current expert weights may bias routing toward historical knowledge and reduce the learning signal received by current experts. Therefore, historical and current experts are normalized independently, allowing current experts to maintain an independent adaptation pathway.

To further improve the utilization of current experts, we introduce two regularization terms. First, routing may concentrate on only a small subset of experts, preventing other experts from receiving sufficient optimization signals. We therefore introduce a routing-balance constraint:

$$\mathcal{L}_{\text{bal}} = E \sum_{e=1}^{E} \left( \frac{1}{B} \sum_{i=1}^{B} \beta_{k,e}(h_i) \right)^2, \tag{3}$$

where $B$ denotes the training batch size. This constraint encourages a more balanced usage of current experts within each batch.

Second, although multiple experts are maintained, they may converge to similar representations and fail to capture complementary knowledge components. To encourage expert specialization and reduce redundancy, we introduce an expert-diversity constraint. Let $D_k^{(\ell)}$ denote the matrix of nor-

malized expert vectors at layer $\ell$. The diversity loss is defined as:

$$\mathcal{L}_{\text{div}} = \frac{1}{|\mathcal{L}_A|} \sum_{\ell \in \mathcal{L}_A} \frac{\left\| D_k^{(\ell)} (D_k^{(\ell)})^\top - I \right\|_F^2}{E(E-1)}, \tag{4}$$

where $\mathcal{L}_A$ denotes the parameter layers involved in expert composition. By reducing similarity among expert representations, this constraint encourages different experts to capture complementary knowledge components. Combining the reinforcement learning objective with the two regularization terms, the overall training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{RL}} + \lambda_{\text{bal}} \mathcal{L}_{\text{bal}} + \lambda_{\text{div}} \mathcal{L}_{\text{div}}, \tag{5}$$

where $\mathcal{L}_{\text{RL}}$ denotes the PPO loss formulated as the negative policy optimization objective, and $\lambda_{\text{bal}}$ and $\lambda_{\text{div}}$ control the strengths of the routing-balance and expert-diversity regularization terms.

### 4.3 Expert Knowledge Merging

As the number of tasks increases, storing a separate expert knowledge group for each task leads to an unbounded growth of the historical knowledge pool. To maintain a compact and scalable knowledge repository, we impose a capacity constraint $P$ on the historical knowledge pool. When the number of historical knowledge groups exceeds this capacity, similar groups are merged until the pool size satisfies the constraint.

However, expert correspondence across different knowledge groups cannot be determined by expert indices. Since experts are learned independently from different tasks, the same expert index may capture different knowledge components in different groups. Directly averaging experts with identical indices may therefore mix unrelated knowledge and degrade the quality of the merged representation. To address this issue, we align experts between two knowledge groups before merging by formulating expert matching as a one-to-one assignment problem.

For two knowledge groups $V_g$ and $V_{g'}$, the optimal expert correspondence is obtained by maximizing the similarity between matched experts:

$$\pi^*_{g,g'} = \arg\max_{\pi} \frac{1}{E} \sum_{e=1}^{E} \cos\left(v_{g,e}, v_{g',\pi(e)}\right), \tag{6}$$

where $\pi$ denotes a one-to-one expert correspondence. Here, cosine similarity is computed from layer-wise L2-normalized descriptors of the expert parameter offsets, while merging averages the original parameter offsets. Based on the optimal matching, the average cosine similarity between the matched experts is used to measure the similarity between two knowledge groups, and the most similar pair is selected for merging. After establishing the expert correspondence, the matched expert knowledge vectors are merged by averaging, i.e., $v_{\text{new},e} = \frac{1}{2}(v_{g^*,e} + v_{g'^*,\pi^*(e)})$, where $e = 1, \ldots, E$. The merging process is repeated until the historical knowledge pool satisfies the capacity constraint. By explicitly aligning experts before consolidation, the proposed merging strategy reduces the effect of expert permutation differences and preserves the structure of reusable knowledge during long-term continual learning.

## 5 Experiments

### 5.1 Experimental Setup

**Training Details.** We evaluate EKC-CRL on Atari continual reinforcement learning benchmarks, including *SpaceInvaders* and *Freeway* Machado et al. (2018). Following CKA-RL Hu et al. (2025), we use the same task sequences, where each environment configuration is treated as an independent task and only current-task interaction data are available during learning. All methods are trained with PPO under identical settings, using $\gamma = 0.99$, GAE $\lambda = 0.95$, and standard Atari preprocessing with four stacked frames. For EKC-CRL, knowledge reuse is applied only to the Actor, with the Critic reinitialized for each new task. We set the number of experts to $E = 4$, the knowledge pool capacity to $P = 5$, and use $\lambda_{\text{bal}} = 0.01$ and $\lambda_{\text{div}} = 0.05$. Other implementation hyperparameters are provided in Appendix C.

Table 1: Performance comparison on Atari continual reinforcement learning benchmarks. PERF. denotes final performance and FWT. denotes forward transfer. Results are reported as mean ± standard deviation over 10 random seeds. For averaged results across environments, the standard deviations are aggregated using root mean square. The best results are highlighted in bold, and the second-best results are underlined.

| Method | SpaceInvaders | | Freeway | | Average | |
|---|---|---|---|---|---|---|
| | PERF. | FWT. | PERF. | FWT. | PERF. | FWT. |
| Baseline | $0.4154 \pm 0.0624$ | $0.0000 \pm 0.0000$ | $0.2514 \pm 0.0554$ | $0.0000 \pm 0.0000$ | $0.3334 \pm 0.0590$ | $0.0000 \pm 0.0000$ |
| FT-N | $0.9068 \pm 0.0776$ | $0.6789 \pm 0.1913$ | $0.7766 \pm 0.1217$ | $0.6287 \pm 0.1642$ | $0.8417 \pm 0.1021$ | $0.6538 \pm 0.1783$ |
| ProgNet | $0.4568 \pm 0.0663$ | $-0.0214 \pm 0.1116$ | $0.3716 \pm 0.0805$ | $0.1285 \pm 0.1724$ | $0.4142 \pm 0.0737$ | $0.0536 \pm 0.1452$ |
| PackNet | $0.2684 \pm 0.0470$ | $-0.1582 \pm 0.1662$ | $0.2404 \pm 0.0724$ | $-0.0705 \pm 0.2159$ | $0.2544 \pm 0.0610$ | $-0.1144 \pm 0.1927$ |
| MaskNet | $0.0000 \pm 0.0000$ | $-0.5651 \pm 0.1462$ | $0.0808 \pm 0.0230$ | $-0.4789 \pm 0.2314$ | $0.0404 \pm 0.0163$ | $-0.5220 \pm 0.1935$ |
| CReLUs | $0.9061 \pm 0.0673$ | $0.6853 \pm 0.1632$ | $0.7954 \pm 0.0803$ | $0.6544 \pm 0.1221$ | $0.8508 \pm 0.0741$ | $0.6699 \pm 0.1441$ |
| CompoNet | $0.9167 \pm 0.0192$ | $0.6441 \pm 0.0574$ | $0.7590 \pm 0.1602$ | $0.6222 \pm 0.2084$ | $0.8378 \pm 0.1141$ | $0.6331 \pm 0.1528$ |
| CbpNet | $\underline{0.9247 \pm 0.0150}$ | $\underline{0.7121 \pm 0.0549}$ | $0.7646 \pm 0.0709$ | $0.6411 \pm 0.0739$ | $0.8446 \pm 0.0512$ | $0.6766 \pm 0.0651$ |
| CKA-RL | $0.9246 \pm 0.0131$ | $0.6952 \pm 0.0764$ | $\underline{0.7994 \pm 0.0534}$ | $\underline{0.6771 \pm 0.0616}$ | $\underline{0.8620 \pm 0.0389}$ | $\underline{0.6862 \pm 0.0694}$ |
| **Ours** | $\mathbf{0.9295 \pm 0.0132}$ | $\mathbf{0.7277 \pm 0.0552}$ | $\mathbf{0.8052 \pm 0.0226}$ | $\mathbf{0.6844 \pm 0.0438}$ | $\mathbf{0.8674 \pm 0.0185}$ | $\mathbf{0.7061 \pm 0.0498}$ |

**Baselines.** We compare EKC-CRL with nine baselines, including Baseline, FT-N Wolczyk et al. (2021), ProgNet Rusu et al. (2016), PackNet Mallya & Lazebnik (2018), MaskNet Ben-Iwhiwhu et al. (2023), CReLUs Abbas et al. (2023), CompoNet Malagon et al. (2024), CbpNet Dohare et al. (2024), and CKA-RL Hu et al. (2025). Detailed descriptions and implementation details of these baselines are provided in Appendix B.

**Evaluation Metrics.** Following CKA-RL Hu et al. (2025) and previous continual reinforcement learning studies Wolczyk et al. (2021), we use final performance (PERF.) and forward transfer (FWT.) as the primary evaluation metrics. For a continual task sequence containing $N$ tasks, let $p_i(t) \in [0, 1]$ denote the success rate on task $i$ at global training step $t$. Each task is trained for $\Delta$ environment steps, resulting in a total of $T = N\Delta$ training steps. The final performance is defined as the average success rate across all tasks at the end of training: $P(T) = \frac{1}{N}\sum_{i=1}^{N} p_i(T)$. Forward transfer measures the extent to which continual learning facilitates knowledge transfer across tasks. Let $p_i^b(t) \in [0, 1]$ denote the success rate of a reference model trained independently from scratch on task $i$. For task $i$, the normalized area under the learning curve is computed as:

$$\mathrm{AUC}_i = \frac{1}{\Delta}\int_{(i-1)\Delta}^{i\Delta} p_i(t)\,\mathrm{d}t, \qquad \mathrm{AUC}_i^b = \frac{1}{\Delta}\int_{0}^{\Delta} p_i^b(t)\,\mathrm{d}t. \tag{7}$$

The forward transfer on task $i$ and the overall forward transfer are defined as:

$$\mathrm{FT}_i = \frac{\mathrm{AUC}_i - \mathrm{AUC}_i^b}{1 - \mathrm{AUC}_i^b}, \qquad \mathrm{FWT} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{FT}_i. \tag{8}$$

### 5.2 Main Results

Table 1 compares EKC-CRL with existing continual reinforcement learning methods on the *SpaceInvaders* and *Freeway* task sequences. EKC-CRL achieves the highest average PERF. and FWT. among all evaluated methods, reaching 0.8674 and 0.7061, respectively. On the *SpaceInvaders* sequence, EKC-CRL obtains a PERF. of 0.9295 and an FWT. of 0.7277, while achieving 0.8052 and 0.6844 on the *Freeway* sequence. These results demonstrate the effectiveness of EKC-CRL in maintaining performance and facilitating knowledge transfer across sequential tasks. Compared with CKA-RL, EKC-CRL improves the average PERF. from 0.8620 to 0.8674 and the average FWT. from 0.6862 to 0.7061. The improvement in FWT. is more evident on the *SpaceInvaders* sequence, where EKC-CRL increases FWT. from 0.6952 to 0.7277. Compared with parameter-isolation methods, EKC-CRL achieves higher average PERF. and FWT. than ProgNet, PackNet, and MaskNet, showing the advantage of dynamically composing reusable knowledge over allocating task-specific parameters. Moreover, EKC-CRL consistently outperforms FT-N on both task sequences, indicating that composing knowledge from multiple historical sources is more effective than transferring knowledge only from the preceding task. Overall, the results validate that EKC-CRL effectively improves both final performance and forward transfer under sequential task learning.

### 5.3 Ablation Study

Table 2 presents the ablation results on the Freeway task sequence to evaluate the contribution of each component in EKC-CRL. Compared with uniform routing, adaptive routing improves both PERF. and FWT., indicating that flexible expert routing enables more effective historical knowledge reuse by selectively combining different knowledge components. Removing either the balance loss or the diversity loss degrades performance, demonstrating that both constraints contribute to effective expert utilization. In particular, removing the diversity loss leads to a larger performance drop, suggesting that differentiated expert representations are important for avoiding redundant knowledge and improving expert composition. The balance loss further prevents routing from concentrating on a small subset of experts, enabling more balanced knowledge utilization. When both regularization terms are removed, performance decreases further, confirming their complementary effects. Overall, EKC-CRL achieves the best performance among all variants, validating the effectiveness of adaptive routing combined with expert diversity and routing-balance regularization.

Table 2: Ablation study of EKC-CRL on the Freeway task sequence. Route, Bal., and Div. denote routing, balance, and diversity losses.

| Variant | Route | Bal. | Div. | PERF. | FWT. |
|---|---|---|---|---|---|
| Uniform Routing | – | – | – | 0.7348 | 0.6133 |
| w/o Both | ✓ | – | – | 0.7437 | 0.6154 |
| w/o Diversity | ✓ | ✓ | – | 0.7641 | 0.6349 |
| w/o Balance | ✓ | – | ✓ | 0.7844 | 0.6577 |
| **EKC-CRL** | ✓ | ✓ | ✓ | **0.8052** | **0.6844** |

### 5.4 Hyperparameter Sensitivity Analysis

**Effect of the number of experts $E$.** The number of experts $E$ determines the granularity of expert-level knowledge representation and the number of parameter directions available for composition. As shown in Fig. 3(a), increasing $E$ generally improves both PERF. and FWT., although performance fluctuates across different settings. When only a small number of experts is used, the model has limited capacity to represent diverse knowledge components, which restricts fine-grained knowledge reuse. As $E$ increases, additional experts provide more parameter directions, allowing the router to select and combine more suitable knowledge components during continual adaptation. When the number of experts becomes sufficiently large, the performance improvement gradually saturates, indicating that additional experts provide limited extra benefits. Considering both performance and model complexity, we set $E = 4$ as the default configuration in subsequent experiments.

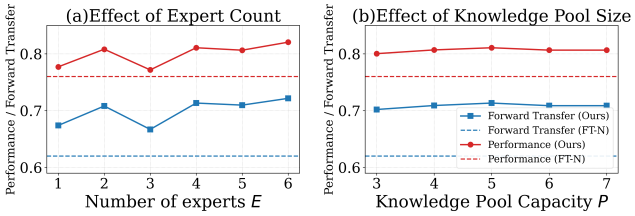



Figure 3: Sensitivity analysis of expert number $E$ and pool capacity $P$.

**Effect of the historical knowledge pool capacity $P$.** The historical knowledge pool capacity $P$ determines the maximum amount of historical expert knowledge that can be retained and reused during continual learning. As shown in Fig. 3(b), increasing $P$ from 3 to 5 improves both final performance and forward transfer, indicating that a larger knowledge pool allows the model to preserve more useful historical knowledge and provides richer information for cross-task knowledge reuse. When $P$ is small, the limited capacity requires more frequent consolidation of historical knowledge groups, which may reduce the granularity of retained knowledge and limit effective knowledge reuse. However, further increasing the capacity beyond 5 does not lead to additional improvements in either performance or forward transfer. This suggests that after retaining a sufficient number of historical knowledge groups, newly added knowledge provides limited benefits and may introduce unnecessary redundancy. Therefore, we set $P = 5$ as the default historical knowledge pool capacity, achieving a balance between knowledge retention and storage efficiency.

### 5.5 Further Discussion

**Catastrophic forgetting analysis.** Table 3 further evaluates catastrophic forgetting on the Freeway task sequence using the average adjacent-task forgetting rate. EKC-CRL obtains a forgetting rate of 0.31, which is comparable to CKA-RL (0.30) and lower than CompoNet (0.34) and CbpNet (0.32). Meanwhile, EKC-CRL achieves the highest PERF. and FWT. of 0.8052 and 0.6844. These results indicate that EKC-CRL improves final performance and forward transfer without a marked increase in forgetting. A low forgetting rate alone, however, does not necessarily imply strong continual learning performance. Although MaskNet and PackNet exhibit lower forgetting rates, their PERF. and FWT. remain substantially lower. Therefore, forgetting should be evaluated jointly with final performance and forward transfer in continual learning.

Table 3: Comparison on the Freeway task sequence. Best and second-best results are bold and underlined, respectively.

| Method | PERF.↑ | FWT.↑ | Forgetting↓ |
|---|---|---|---|
| ProgNet | 0.3716 | 0.1285 | 0.30 |
| PackNet | 0.2404 | -0.0705 | <u>0.03</u> |
| MaskNet | 0.0808 | -0.4789 | **0.01** |
| CReLUs | 0.7954 | 0.6544 | 0.28 |
| CompoNet | 0.7590 | 0.6222 | 0.34 |
| CbpNet | 0.7646 | 0.6411 | 0.32 |
| CKA-RL | <u>0.7994</u> | <u>0.6771</u> | 0.30 |
| **EKC-CRL** | **0.8052** | **0.6844** | 0.31 |

**Resource efficiency and scalability.** To examine scalability under increasing task sequences, Fig. 4 compares the policy parameter counts and inference latency of different methods. Architecture-expansion approaches such as ProgNet and CompoNet continuously introduce new task-specific modules, resulting in nearly linear growth of both parameters and inference cost as more tasks are encountered. Although these methods reduce interference by allocating additional capacity, their resource requirements increase with the length of the task sequence. In contrast, EKC-CRL maintains a bounded historical knowledge pool and merges similar expert knowledge groups when the capacity limit is reached. Therefore, the number of policy parameters gradually increases during early learning and becomes stable after the knowledge pool reaches its predefined capacity. Compared with CKA-RL, EKC-CRL introduces additional computation due to maintaining multiple expert directions and performing expert composition. However, this overhead remains controlled while enabling more flexible knowledge reuse and achieving better final performance and forward transfer. These results demonstrate that EKC-CRL provides a scalable solution for continual reinforcement learning with long task sequences.

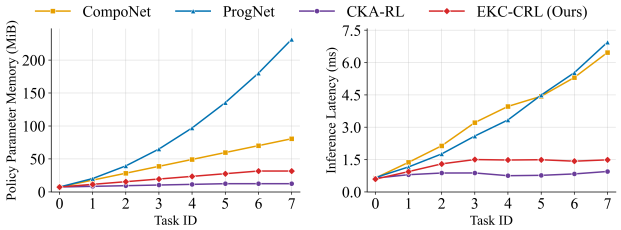



Figure 4: Scalability analysis of parameter count and inference latency with increasing tasks.

## 6 Conclusion

We propose EKC-CRL, an expert knowledge composition framework for continual reinforcement learning. EKC-CRL addresses the limitation of coarse-grained knowledge reuse by representing historical task knowledge as groups of expert knowledge components, enabling fine-grained knowledge decomposition and composition across tasks. It further maintains independent current experts for acquiring task-specific knowledge and introduces routing-balance and expert-diversity regularization to improve expert utilization. To support long-term continual learning, EKC-CRL employs expert matching and knowledge merging to maintain a bounded historical knowledge pool while preserving reusable knowledge. Extensive experiments on multiple continual reinforcement learning benchmarks demonstrate that EKC-CRL consistently improves final performance and forward transfer compared with existing methods, showing the effectiveness of expert-level knowledge composition for sequential task learning. By shifting from task-level knowledge preservation to expert-level knowledge composition, EKC-CRL enables more flexible knowledge reuse in continual learning. Future work will explore more adaptive knowledge management strategies, including automatically adjusting expert capacity and knowledge pool size according to task characteristics, as well as extending expert composition to larger-scale and more diverse continual reinforcement learning environments and applications.

### AI USE STATEMENT

Generative AI tools were used for manuscript English polishing (grammar correction and sentence rephrasing). All AI-assisted text has been manually reviewed and revised by the authors to ensure academic accuracy and consistency. The authors are fully responsible for the entire manuscript content provided.

### ETHICS STATEMENT

This work does not involve human subjects, private data, or sensitive information. All experiments are conducted on publicly available reinforcement learning benchmarks and follow their original usage guidelines.

### REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide implementation details, experimental configurations, and baseline descriptions in the Appendix. The source code will be made publicly available after the review process.

## REFERENCES

Zaheer Abbas, Rosie Zhao, Joseph Modayil, Adam White, and Marlos C Machado. Loss of plasticity in continual deep reinforcement learning. In *Conference on lifelong learning agents*, pp. 620–636. PMLR, 2023.

David Abel, André Barreto, Benjamin Van Roy, Doina Precup, Hado Van Hasselt, and Satinder Singh. A definition of continual reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 50377–50407, 2023.

André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado P. van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4055–4065. Curran Associates, Inc., 2017.

Eseoghene Ben-Iwhiwhu, Saptarshi Nath, Praveen K. Pilly, Soheil Kolouri, and Andrea Soltoggio. Lifelong reinforcement learning with modulating masks. *Transactions on Machine Learning Research*, 2023.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022.

Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 632:768–774, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Ting Guo, Yangrui Fan, Caixia Cui, Jiye Liang, Jiao Zhao, and Da Wang. Topological anomaly quantification for semi-supervised graph anomaly detection. In *The Fourteenth International Conference on Learning Representations*, 2026.

Heinke Hihn and Daniel A. Braun. Hierarchically structured task-agnostic continual learning. *Machine Learning*, 112(2):655–686, 2023.

Jinwu Hu, Zihao Lian, Zhiquan Wen, Chenghao Li, Guohao Chen, Xutao Wen, Bin Xiao, and Mingkui Tan. Continual knowledge adaptation for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 36283–36313, 2025.

Suning Huang, Zheyu Aqa Zhang, Tianhai Liang, Yihan Xu, Zhehao Kou, Chenhao Lu, Guowei Xu, Zhengrong Xue, and Huazhe Xu. MENTOR: Mixture-of-experts network with task-oriented perturbation for visual reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 26143–26161. PMLR, 2025.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. In *International Conference on Learning Representations*, 2020.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. BASE layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, pp. 6265–6274. PMLR, 2021.

Hongbo Li, Sen Lin, Lingjie Duan, Yingbin Liang, and Ness Shroff. Theory on mixture-of-experts in continual learning. In *International Conference on Learning Representations*, 2025.

Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

Mikel Malagon, Josu Ceberio, and Jose A. Lozano. Self-composing policies for scalable continual reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34432–34460. PMLR, 2024.

Arun Mallya and Svetlana Lazebnik. PackNet: Adding multiple tasks to a single network by iterative pruning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773. IEEE, 2018.

Jorge A. Mendez, Harm van Seijen, and Eric Eaton. Modular lifelong reinforcement learning via neural composition. In *International Conference on Learning Representations*, 2022.

Yuan Meng, Zhenshan Bing, Xiangtong Yao, Kejia Chen, Kai Huang, Yang Gao, Fuchun Sun, and Alois Knoll. Preserving and combining knowledge in robotic lifelong reinforcement learning. *Nature Machine Intelligence*, 7(2):256–269, 2025.

Aneesh Muppidi, Zhiyu Zhang, and Heng Yang. Fast TRAC: A parameter-free optimizer for lifelong reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 51169–51195, 2024.

Johan Samir Obando Ceron, Ghada Sokar, Timon Willi, Clare Lyle, Jesse Farebrother, Jakob Nicolaus Foerster, Gintare Karolina Dziugaite, Doina Precup, and Pablo Samuel Castro. Mixtures of experts unlock parameter scaling for deep RL. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38520–38540. PMLR, 2024.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Joan Puigcerver, Carlos Riquelme Ruiz, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *International Conference on Learning Representations*, 2024.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, volume 32, pp. 348–358. Curran Associates, Inc., 2019.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, volume 30, pp. 2990–2999, 2017.

Yee Whye Teh, Victor Bapst, Wojciech M. Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4496–4506. Curran Associates, Inc., 2017.

Da Wang, Lin Li, Wei Wei, Qixian Yu, Jianye Hao, and Jiye Liang. Improving generalization in offline reinforcement learning via adversarial data splitting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 50822–50839. PMLR, 2024.

Da Wang, Yi Ma, Ting Guo, Hongyao Tang, Wei Wei, and Jiye Liang. Fans: A flatness-aware network structure for generalization in offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 149075–149098, 2025.

Timon Willi, Johan Samir Obando Ceron, Jakob Nicolaus Foerster, Gintare Karolina Dziugaite, and Pablo Samuel Castro. Mixture of experts in a mixture of RL settings. *Reinforcement Learning Journal*, 3:1072–1105, 2024.

M. Wolczyk, M. Zajac, R. Pascanu, L. Kucinski, and P. Milos. Continual world: A robotic benchmark for continual reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 28496–28510, 2021.

William Yue, Bo Liu, and Peter Stone. t-DGR: A trajectory-based deep generative replay method for continual learning in decision making. In *Proceedings of The 3rd Conference on Lifelong Learning Agents*, volume 274 of *Proceedings of Machine Learning Research*, pp. 481–497. PMLR, 2025.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 3987–3995. PMLR, 2017.

Tiantian Zhang, Kevin Zehua Shen, Zichuan Lin, Bo Yuan, Xueqian Wang, Xiu Li, and Deheng Ye. Replay-enhanced continual reinforcement learning. *Transactions on Machine Learning Research*, 2023.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, pp. 7103–7114, 2022.

**Algorithm 1:** The Pipeline of EKC-CRL

**Input:** Task sequence $\{\tau_0, \ldots, \tau_{N-1}\}$, initial parameters $\theta_0$, pool capacity $P$, experts per task $E$
**Output:** Learned policies $\{\pi_0, \ldots, \pi_{N-1}\}$

**Base Policy Learning:** $M_0 \leftarrow \mathrm{PPOTrain}(\tau_0; \theta_0)$
$\theta_{\mathrm{base}} \leftarrow \mathrm{BaseActor}(M_0)$, $\mathcal{H} \leftarrow \emptyset$
**for** $k = 1, \ldots, N-1$ **do**
  **if** $k > 1$ **then**
    **Knowledge Pool Update:** $\mathcal{H} \leftarrow \mathcal{H} \cup \{V_{k-1}\}$
  **while** $|\mathcal{H}| > P$ **do**
    **Expert Matching:** $(g, g', \pi^\star) \leftarrow \mathrm{BestHungarianPair}(\mathcal{H})$
    **Knowledge Merging:** $V_{\mathrm{merge}} \leftarrow \frac{1}{2}\big(V_g + \pi^\star(V_{g'})\big)$
    $\mathcal{H} \leftarrow (\mathcal{H} \setminus \{V_g, V_{g'}\}) \cup \{V_{\mathrm{merge}}\}$
  **Task Initialization:** $\theta_{k,e}^{\mathrm{cur}} \leftarrow \theta_{\mathrm{base}} + \epsilon_{k,e}$, $e = 1, \ldots, E$; initialize routing networks $\phi_k$ ;
  **Policy Construction:** $v_{k,e} \equiv \theta_{k,e}^{\mathrm{cur}} - \theta_{\mathrm{base}}$
  $\theta_k(h) \leftarrow \theta_{\mathrm{base}} + \sum_{g=1}^{|\mathcal{H}|} \alpha_g(h) \sum_{e=1}^{E} \beta_{g,e}(h) v_{g,e} + \sum_{e=1}^{E} \beta_{k,e}(h) v_{k,e}$
  **Policy Optimization:** Optimize current experts and routing networks with PPO, routing-balance, and diversity losses; freeze the base and historical experts
  **Expert Decomposition:** $V_k \leftarrow [\theta_{k,e}^{\mathrm{cur}} - \theta_{\mathrm{base}}]_{e=1}^{E}$

## A Algorithm Details

Algorithm 1 summarizes the overall training pipeline of EKC-CRL. The first task $\tau_0$ is used to learn a shared base policy, from which the base actor parameters $\theta_{\mathrm{base}}$ are extracted and fixed during subsequent continual learning. EKC-CRL maintains a historical knowledge pool $\mathcal{H}$ that stores compressed expert knowledge from previously learned tasks.

Task $\tau_0$ learns the fixed base actor and does not produce an expert group $V_0$. The historical pool is therefore empty when training the first expert task $\tau_1$. After each task $\tau_k$, the learned current experts are expressed as parameter offsets from the base actor, forming $V_k$. At the beginning of the next task, $V_k$ is added to the inherited historical pool. If the pool then exceeds capacity $P$, Hungarian matching aligns the most similar groups before they are merged. The base actor and historical experts remain frozen during current-task training.

## B Baseline Details

We compare EKC-CRL with nine continual reinforcement learning baselines. A brief description of each competing method is provided below.

- **Baseline.** The baseline trains an independent policy from scratch for each task without transferring knowledge from previous tasks. It provides a reference performance without continual knowledge reuse.
- **FT-N** Wolczyk et al. (2021). FT-N sequentially fine-tunes the policy learned from the previous task on the current task. It enables direct parameter transfer but suffers from catastrophic forgetting due to unrestricted parameter updates.
- **Progressive Networks (ProgNet)** Rusu et al. (2016). ProgNet expands the network by adding task-specific columns for new tasks while keeping previous columns fixed. Lateral connections allow the new columns to reuse previously learned representations.
- **PackNet** Mallya & Lazebnik (2018). PackNet allocates separate parameter subsets for different tasks through iterative pruning and parameter freezing. By isolating task-specific parameters, it reduces interference between sequential tasks.
- **MaskNet** Ben-Iwhiwhu et al. (2023). MaskNet learns task-specific masks over a shared neural network and uses these masks to preserve previously acquired knowledge. The method improves continual adaptation by controlling parameter usage across tasks.

Table 4: Environment and training configurations.

| Configuration | Value |
|---|---|
| Environment | ALE/Freeway-v5, ALE/SpaceInvaders-v5 |
| Number of tasks | 8 / 10 |
| Environment steps per task | 1,000,000 |
| Parallel environments | 8 |
| Input resolution | $4 \times 84 \times 84$ |
| Frame skip | 4 |
| Frame stacking | 4 |
| Optimizer | Adam |
| Learning rate | $2.5 \times 10^{-4}$ |
| Rollout length | 128 |
| Batch size | 1024 |
| Update epochs | 4 |
| Discount factor $\gamma$ | 0.99 |
| GAE coefficient $\lambda$ | 0.95 |
| PPO clip coefficient | 0.1 |
| Entropy coefficient | 0.01 |
| Value loss coefficient | 0.5 |

- **CReLUs** Abbas et al. (2023). CReLUs addresses the loss of plasticity in continual reinforcement learning by introducing the Concatenated ReLU activation function. It improves the ability of neural networks to acquire new knowledge during long task sequences by maintaining richer activation patterns.
- **CompoNet** Malagon et al. (2024). CompoNet represents previously learned policies as reusable components and learns new tasks by composing existing modules with newly trained components. It enables knowledge reuse through modular policy composition.
- **CbpNet** Dohare et al. (2024). CbpNet maintains neural network plasticity by periodically replacing underutilized units during continual learning. This mechanism helps preserve the model's ability to adapt to future tasks.
- **CKA-RL** Hu et al. (2025). CKA-RL represents task knowledge as task-specific knowledge vectors and retrieves previously learned knowledge during adaptation to new tasks. It further performs knowledge merging to facilitate transfer across sequential tasks.

For fair comparison, all methods use the same PPO backbone, observation preprocessing, task sequences, and training budgets. Unless a baseline requires specific architectural components, the remaining optimization settings are kept consistent across methods.

# C Additional Implementation Details

## C.1 Environment and Training Settings

We evaluate EKC-CRL on two Atari continual reinforcement learning benchmarks, including *Freeway-v5* and *SpaceInvaders-v5* from the Arcade Learning Environment (ALE). Each environment configuration is treated as an individual task, and tasks are presented to the agent in a predefined sequential order. During continual training, the agent only has access to data collected from the current task, while trajectories from previous tasks are unavailable. All methods are trained using PPO as the underlying reinforcement learning algorithm. Unless otherwise specified, the same optimization settings are applied to all compared methods to ensure a fair comparison. The main environment and training configurations are summarized in Table 4.

## C.2 EKC-CRL Configuration

EKC-CRL introduces additional hyperparameters related to expert composition and knowledge management. The number of experts $E$ controls the granularity of knowledge decomposition, while the historical knowledge pool capacity $P$ determines the maximum number of retained knowledge

Table 5: EKC-CRL and network configurations.

| Configuration | Value |
|---|---|
| Number of experts $E$ | 4 |
| Historical pool capacity $P$ | 5 |
| Balance coefficient $\lambda_{\text{bal}}$ | 0.01 |
| Diversity coefficient $\lambda_{\text{div}}$ | 0.05 |
| Weight perturbation std. | 0.001 |
| Bias perturbation std. | 0.0001 |
| Encoder output dimension | 512 |
| Actor network | $512 \to 512 \to action_dim$ |
| Critic network | $512 \to 1$ |
| Initialization | Orthogonal initialization |

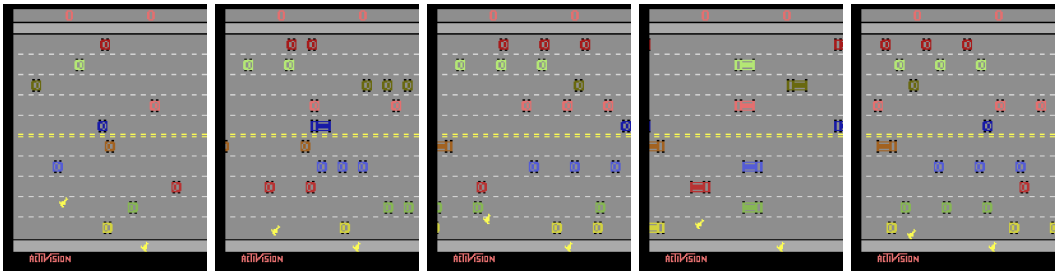


Figure 5: Example observations from different task modes in the Freeway environment.

groups. During continual training, the base parameters and historical experts remain fixed, while only the current experts and routing-related modules are optimized using data from the current task. After completing a task, the learned current experts are incorporated into the historical knowledge pool for future knowledge reuse. The main EKC-CRL and network configurations are provided in Table 5.

### C.3 Atari Environment Details

We evaluate EKC-CRL on the game-mode task sequences of *ALE/Freeway-v5* and *ALE/SpaceInvaders-v5*, following the benchmark configuration used by CKA-RL Hu et al. (2025). Each game mode constitutes one task, and the modes are encountered in numerical order. The figures show representative RGB observations from five selected modes in each environment; they do not depict every task in the sequence.

**Freeway.** The agent moves a chicken across a road while avoiding traffic. A successful crossing yields a reward. We use modes 0–7, which vary vehicle types, traffic density, and vehicle speeds. Figure 5 shows representative observations from modes 0, 1, 2, 3, and 6.

**Mode 0.** The default traffic configuration.

**Mode 1.** Traffic becomes denser and faster, and trucks appear in a lane near the center of the road.

**Mode 2.** Relative to Mode 1, traffic increases further and the trucks move faster.

**Mode 3.** Trucks appear across all lanes, with high vehicle speeds retained in part of the traffic.

**Mode 4.** There are no trucks; vehicle speeds vary during an episode.

**Mode 5.** Mode 1 is extended with changing vehicle speeds and closely spaced groups of vehicles.

**Mode 6.** The clustered, variable-speed traffic of Mode 5 becomes denser.

**Mode 7.** Trucks appear in every lane, and their speeds vary during an episode.

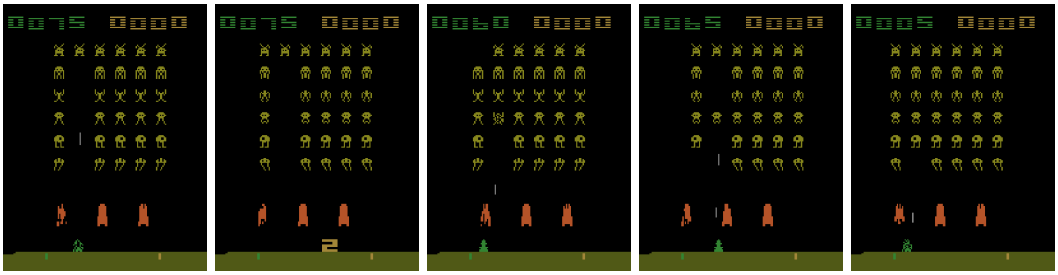


Figure 6: Example observations from different task modes in the SpaceInvaders environment.

**SpaceInvaders.** The agent controls a laser cannon, shooting invaders while avoiding incoming bombs. We use modes 0–9, which vary shield movement, bomb trajectories and speeds, and invader visibility. Figure 6 shows representative observations from modes 0, 1, 2, 3, and 7.

**Mode 0.** The default configuration has stationary shields and standard bomb trajectories.

**Mode 1.** Shields move horizontally instead of remaining fixed.

**Mode 2.** Incoming bombs follow zigzag trajectories.

**Mode 3.** Moving shields are combined with zigzagging bombs.

**Mode 4.** Bombs descend faster than in the default configuration.

**Mode 5.** Moving shields are combined with faster bombs.

**Mode 6.** Zigzagging bombs also descend faster.

**Mode 7.** Moving shields, zigzagging bombs, and faster bomb descent are combined.

**Mode 8.** Invaders periodically become invisible.

**Mode 9.** Periodic invader invisibility is combined with moving shields.

# D Additional Experiments

## D.1 Sensitivity Analysis of Loss Weights

The proposed EKC-CRL introduces two regularization terms to improve expert utilization: the diversity constraint controlled by $\lambda_{\text{div}}$ and the routing-balance constraint controlled by $\lambda_{\text{bal}}$. We analyze the influence of these coefficients on the Freeway task sequence, as shown in Figure 7.

For the diversity coefficient, increasing $\lambda_{\text{div}}$ from 0.025 to the default value 0.05 improves both PERF. and FWT., indicating that encouraging differentiated expert representations helps the model learn more complementary knowledge components. However, further increasing the coefficient to 0.25 leads to degraded performance, suggesting that overly strong diversity regularization restricts expert adaptation and limits the acquisition of task-specific knowledge. The default setting achieves a favorable trade-off between expert specialization and adaptation.

For the balance coefficient, increasing $\lambda_{\text{bal}}$ from 0.002 to 0.01 consistently improves performance, demonstrating the benefit of preventing routing concentration on a small subset of experts. Larger values maintain comparable results but do not provide additional gains, indicating that excessive routing constraints may reduce the flexibility of expert selection. Overall, EKC-CRL remains stable across a broad range of regularization weights, while the default coefficients provide the best balance between expert diversity, utilization, and adaptation.

## D.2 Expert Specialization Analysis

To understand whether the proposed expert decomposition can discover complementary knowledge components within each task, we further analyze the relationships among experts learned for the

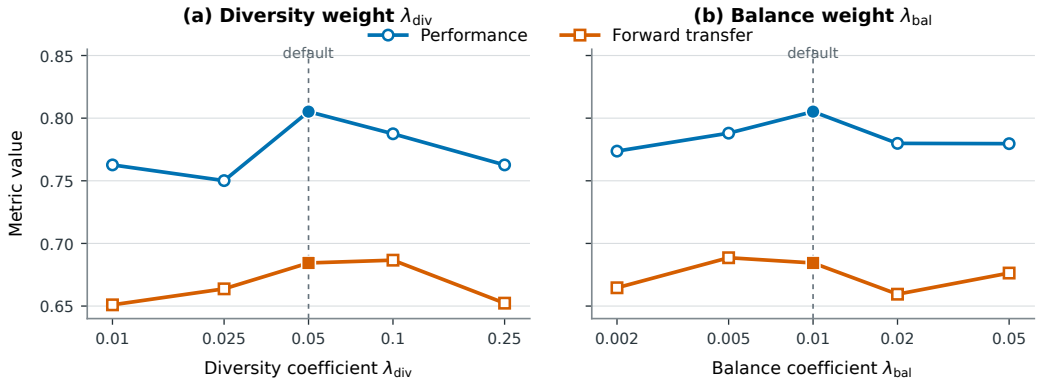



Figure 7: Sensitivity analysis of the diversity coefficient $\lambda_{\text{div}}$ and routing-balance coefficient $\lambda_{\text{bal}}$ on the Freeway task sequence.

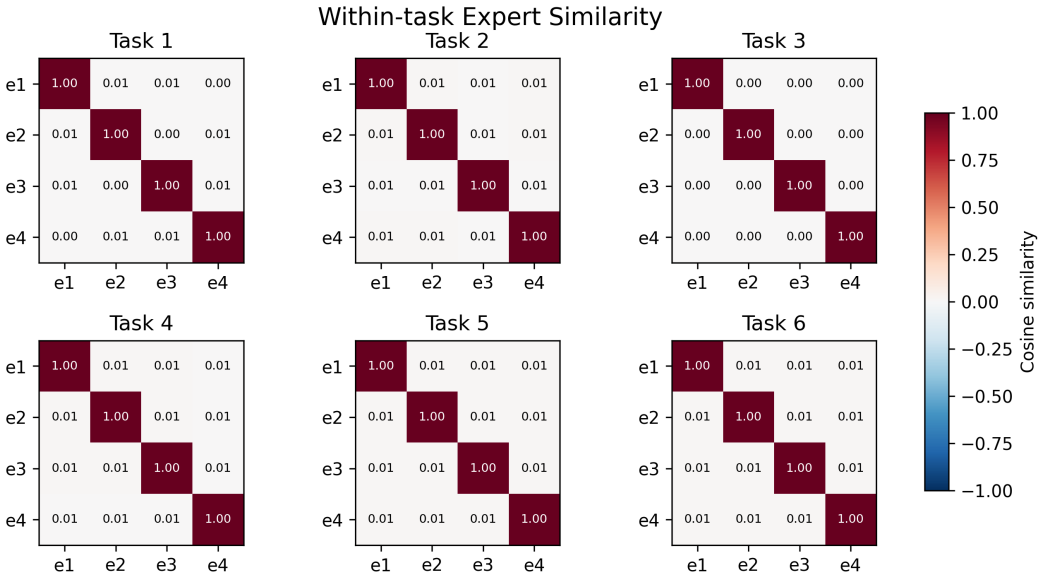



Figure 8: Within-task cosine similarity matrices among experts on six representative tasks. Low off-diagonal similarity indicates that different experts learn diverse knowledge representations.

same task. Specifically, we compute the cosine similarity between expert knowledge vectors within each task, where a higher similarity indicates that two experts capture more similar parameter directions and potentially redundant information. Figure 8 presents the within-task expert similarity matrices for six representative tasks in the Freeway task sequence. The diagonal entries are equal to one since each expert is compared with itself. In contrast, the off-diagonal similarities between different experts remain consistently close to zero across all evaluated tasks. This indicates that experts associated with the same task do not converge to similar parameter directions, but instead capture distinct knowledge components. Such a separation suggests that the expert decomposition does not simply replicate multiple copies of the same task-specific policy representation; rather, it encourages different experts to specialize in different aspects of task solving. This property is particularly important for expert knowledge composition. Maintaining diverse expert representations allows the model to select and combine complementary knowledge components during continual adaptation. Without sufficient expert diversity, multiple experts would provide highly correlated information, reducing the effective capacity of the expert pool and making expert composition less meaningful. The observed low off-diagonal similarity further verifies that the proposed diversity regularization successfully promotes expert specialization while preserving independent knowledge directions.

# E THEORETICAL ANALYSIS

This section provides theoretical analysis of the knowledge merging mechanism in EKC-CRL. We focus on three properties of expert knowledge management: permutation-invariant expert matching, stability of expert merging, and bounded active knowledge storage.

## E.1 PERMUTATION-INVARIANT EXPERT MATCHING

Experts learned from different tasks do not necessarily preserve the same ordering. Therefore, directly merging experts with identical indices may introduce incorrect correspondences. EKC-CRL employs Hungarian matching to identify the optimal correspondence between expert groups before knowledge merging.

For two expert groups $V = [v_1, \ldots, v_E]$ and $W = [w_1, \ldots, w_E]$, we define their optimal matching score as:

$$S(V, W) = \max_{\pi} \frac{1}{E} \sum_{e=1}^{E} \cos(v_e, w_{\pi(e)}), \tag{9}$$

where $\pi$ denotes a one-to-one permutation between experts. As in the main text, cosine similarity is computed from layer-wise L2-normalized descriptors of the expert parameter offsets.

**Lemma 1 (Permutation Invariance of Expert Matching).** For arbitrary permutations $\sigma$ and $\tau$ applied to two expert groups, the optimal matching score satisfies:

$$S(V_\sigma, W_\tau) = S(V, W). \tag{10}$$

*Proof.* The permutations $\sigma$ and $\tau$ only change the indices of experts without altering the expert representations themselves. For any feasible matching between $V$ and $W$, there exists an equivalent matching between $V_\sigma$ and $W_\tau$ with identical cosine similarity values by applying the corresponding inverse permutations. Therefore, the set of achievable matching scores remains unchanged after re-indexing. Since the Hungarian algorithm finds the maximum-weight matching, the optimal matching score is invariant to expert permutations.

## E.2 STABILITY OF EXPERT KNOWLEDGE MERGING

After expert alignment, EKC-CRL merges matched expert vectors by averaging their parameters. We analyze the perturbation introduced by this merging operation. For two matched expert vectors $v_a$ and $v_b$, the merged vector is defined as:

$$v_m = \frac{1}{2}(v_a + v_b). \tag{11}$$

Let$r_a = \|v_a\|_2$, $r_b = \|v_b\|_2$, and $c = \cos(v_a, v_b)$. The deviation between the merged vector and one of the original expert vectors can be written as:

$$\begin{aligned} \|v_m - v_a\|_2^2 &= \frac{1}{4}\|v_b - v_a\|_2^2 \\ &= \frac{1}{4}\left(r_a^2 + r_b^2 - 2r_a r_b c\right). \end{aligned} \tag{12}$$

**Lemma 2 (Conditional Stability of Expert Merging).** Assume that two matched expert vectors satisfy $r_a, r_b \le R$, $|r_a - r_b| \le \delta$, and $\cos(v_a, v_b) \ge \rho$. The perturbation introduced by averaging the two expert vectors is bounded by:

$$\|v_m - v_a\|_2 \le \frac{1}{2}\sqrt{\delta^2 + 2R^2(1-\rho)}. \tag{13}$$

*Proof.* The squared distance between the merged vector and $v_a$ can be rewritten as:

$$\begin{aligned} \|v_m - v_a\|_2^2 &= \frac{1}{4}\left(r_a^2 + r_b^2 - 2r_a r_b c\right) \\ &= \frac{1}{4}\left((r_a - r_b)^2 + 2r_a r_b(1-c)\right). \end{aligned} \tag{14}$$

According to the assumptions, the norm difference satisfies $|r_a - r_b| \leq \delta$, and the cosine similarity satisfies $c \geq \rho$. Moreover, since $r_a, r_b \leq R$,

$$2r_a r_b(1-c) \leq 2R^2(1-\rho). \tag{15}$$

Therefore,

$$\|v_m - v_a\|_2^2 \leq \frac{1}{4}\left(\delta^2 + 2R^2(1-\rho)\right). \tag{16}$$

Taking the square root on both sides gives

$$\|v_m - v_a\|_2 \leq \frac{1}{2}\sqrt{\delta^2 + 2R^2(1-\rho)}. \tag{17}$$

The same bound applies to $\|v_m - v_b\|_2$. Therefore, when matched experts have similar directions and bounded norm differences, the merging operation introduces only a limited parameter perturbation.

### E.3 Bounded Active Knowledge Pool

As the number of tasks increases, continuously retaining all historical expert groups would cause the active knowledge storage to grow with the task sequence. EKC-CRL limits the number of retained knowledge groups by maintaining a fixed historical knowledge pool capacity $P$.

**Lemma 3 (Bounded Active Knowledge Storage).** After knowledge consolidation, the active historical knowledge pool of EKC-CRL requires at most $\mathcal{O}(PEd)$ memory, where $E$ is the number of experts per knowledge group and $d$ denotes the dimension of each expert vector.

*Proof.* The historical knowledge pool contains at most $P$ knowledge groups after consolidation. Each group consists of $E$ expert vectors, and each expert vector has dimension $d$. Therefore, the storage requirement of the active knowledge pool is: $P \times E \times d$, which corresponds to: $\mathcal{O}(PEd)$. This bound applies to the active historical knowledge pool after consolidation and does not include archived checkpoints or temporary memory used during pool construction.