A Survey of Continual Reinforcement Learning

Chaofan Pan, Xin Yang*, *Member*, *IEEE*, Yanhua Li, Wei Wei, *Member*, *IEEE*, Tianrui Li, *Senior Member*, *IEEE*, Bo An, *Senior Member*, *IEEE*, Jiye Liang, *Fellow*, *IEEE*

***Abstract*—Reinforcement Learning (RL) is an important machine learning paradigm for solving sequential decision-making problems. Recent years have witnessed remarkable progress in this field due to the rapid development of deep neural networks. However, the success of RL currently relies on extensive training data and computational resources. In addition, RL's limited ability to generalize across tasks restricts its applicability in dynamic and real-world environments. With the arisen of Continual Learning (CL), Continual Reinforcement Learning (CRL) has emerged as a promising research direction to address these limitations by enabling agents to learn continuously, adapt to new tasks, and retain previously acquired knowledge. In this survey, we provide a comprehensive examination of CRL, focusing on its core concepts, challenges, and methodologies. Firstly, we conduct a detailed review of existing works, organizing and analyzing their metrics, tasks, benchmarks, and scenario settings. Secondly, we propose a new taxonomy of CRL methods, categorizing them into four types from the perspective of knowledge storage and/or transfer. Finally, our analysis highlights the unique challenges of CRL and provides practical insights into future directions.** [1]

***Index Terms*—Continual reinforcement learning, deep reinforcement learning, continual learning, transfer learning.**

I. INTRODUCTION

REINFORCEMENT Learning (RL) has emerged as a powerful paradigm in machine learning, enabling agents to learn optimal decision-making policies by interacting with their environments [1]. The combination of RL with the representation learning capabilities of deep neural networks has led to the development of *Deep Reinforcement Learning* (DRL), which has achieved remarkable success across a wide range of domains [2]. DRL has demonstrated its potential in solving high-dimensional and complex decision-making problems, from mastering board games such as chess, shogi, and Go [3], to advancing scientific discovery by predicting protein structures [4], correcting quantum computing errors [5], and training large language models [6], [7]. It has also been applied to real-world control tasks, such as optimizing thermal power generation [8], controlling plasma configurations in tokamak nuclear fusion reactors [9], and achieving safe autonomous driving [10]. Despite these achievements, the current success of DRL has been primarily driven by its ability to learn fixed strategies for specific issues, often requiring a vast amount of training data and computational resources [11]. These pose significant challenges for the practical deployment of DRL in real-world applications. Specifically, existing DRL algorithms generally lack the ability to efficiently transfer knowledge across tasks or adapt to new environments. When faced with a new task, these algorithms typically need to start learning from scratch, leading to low sample efficiency and poor generalization performance [12]–[14].

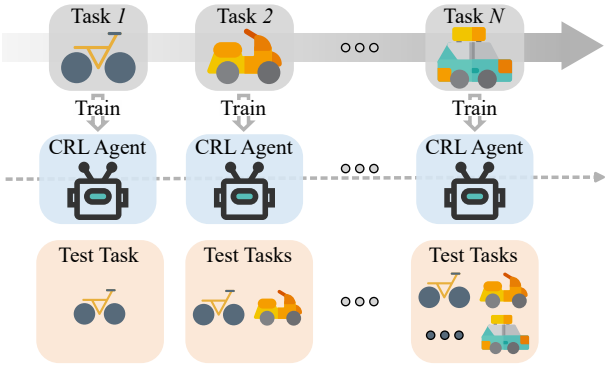

Fig. 1. The setting of CRL. Different classes arrive sequentially, and the agent is required to learn to solve the new task incrementally. After learning each task, the agent is evaluated on all previously learned tasks.

To address these challenges, researchers have been exploring methods to enable RL agents to avoid catastrophic forgetting and effectively transfer knowledge, with the ultimate goal of steering the field toward more human-like intelligence. Humans excel at leveraging prior knowledge to solve new tasks without significantly forgetting previously learned skills [15]. Inspired by this capability, the field of *Continual Learning* (CL), also referred to as lifelong learning or incremental learning, aims to develop learning systems that can adapt to new tasks while retaining knowledge from previous ones [16]–[19]. The central challenge in CL lies in achieving a balance between stability and plasticity—maintaining the stability of previously learned knowledge while allowing sufficient flexibility to adapt to new tasks. The overarching goal is to build intelligent systems that are capable of learning and adapting throughout their lifetimes, rather than starting anew for each task. Current research in CL primarily focuses on two key aspects: addressing catastrophic forgetting and enabling knowledge transfer. Catastrophic forgetting refers to

* Xin Yang is corresponding author.

Chaofan Pan, Xin Yang, and Yanhua Li are with the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu, 611130, China. E-mail: pan.chaofan@foxmail.com, yangxin@swufe.edu.cn, yyhhliyanhua@163.com.

Wei Wei and Jiye Liang are with the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi, 030006, China. E-mail: {weiwei, ljy}@sxu.edu.cn.

Tianrui Li is with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, 611756, China. E-mail: trli@swjtu.edu.cn.

Bo An is with the College of Data Science and Computing, Nanyang Technological University, 639798, Singapore. E-mail: boan@ntu.edu.sg.

the phenomenon where learning new tasks causes the model to overwrite and lose knowledge of previously learned tasks. Knowledge transfer, on the other hand, involves leveraging accumulated knowledge from past tasks to improve the learning efficiency and performance on new or even previously seen tasks. Successfully addressing both aspects is critical for developing robust, continual learning systems.

Continual Reinforcement Learning (CRL, a.k.a. *Lifelong Reinforcement Learning*, LRL) emerges from the intersection of RL and CL, aspiring to address the numerous limitations present in current RL algorithms to achieve agents that can continuously learn and adapt to a series of complex tasks [20], [21]. Fig. 1 illustrates the setting of CRL. Unlike traditional DRL, which focuses primarily on optimizing performance for a single task, CRL emphasizes maintaining and enhancing generalization capabilities across a sequence of tasks. This shift in focus is crucial for deploying RL agents in dynamic, non-stationary environments.

It is worth noting that the terms "lifelong" and "continual" are often used interchangeably in the RL literature, but their usage can vary significantly across studies, potentially leading to confusion [22]. In general, most LRL research emphasizes rapid adaptation to new tasks, while CRL research prioritizes avoiding catastrophic forgetting. In this survey, we unify the two terms under the umbrella of CRL, reflecting the broader trend in CL research to address both aspects simultaneously. A CRL agent is expected to achieve two key objectives: 1) minimizing the forgetting of knowledge from previously learned tasks and 2) leveraging prior experiences to learn new tasks more efficiently. By fulfilling these objectives, CRL holds the promise of addressing the current limitations of DRL, paving the way for RL techniques to be applied in broader and more complex domains. Ultimately, CRL aspires to achieve human-like lifelong learning capabilities, making it a compelling direction for advancing the field of RL.

Currently, a limited number of works have reviewed the field of CRL. Some surveys [18], [23] provide a comprehensive overview of CL in general, including both supervised and RL. Most notably, Khetarpal *et al.* [21] published a survey on CRL from the perspective of non-stationary RL. The survey first formulates the definition of the general CRL problem and provides a taxonomy of different CRL formulations by mathematically characterizing two key properties of non-stationarity. However, it lacks detailed comparison and discussion of some important parts in CRL, such as the challenges, benchmarks, and scenario settings, which are essential for guiding practical research. Furthermore, the number of CRL methods has been growing rapidly over the past five years. With this in mind, we aim to provide a comprehensive review of the most recent research in CRL, focusing on providing a new taxonomy of CRL methods and understanding how knowledge is stored and transferred in CRL.

In this survey, we delve into the evolving field of CRL, which seeks to bridge the gap between traditional RL and the dynamic demands of real-world environments. Our work provides a comprehensive examination of CRL, focusing on its foundational concepts, challenges, and methodologies. We explore the intricacies of CRL and identify the key challenges and benchmarks that define the field. To achieve this, we systematically review the current state of CRL research and propose a taxonomy that organizes existing methods into distinct categories. This structured approach not only clarifies the landscape of CRL but also highlights emerging trends and potential future directions. By examining the interplay between policy, experience, dynamics, and reward-focused methods, we provide a nuanced understanding of how CRL can be optimized to enhance both learning efficiency and generalization capabilities. Our analysis also extends to novel research areas that push the boundaries of CRL, offering insights into how these innovations can be harnessed to develop more sophisticated *Artificial Intelligence* (AI) systems.

TABLE I
THE STRUCTURE OF THIS SURVEY.

§ I: Introduction	
§ II Background	§ II-A: Reinforcement Learning
	§ II-B: Continual Learning
§ III Overview	§ III-A: Definition
	§ III-B: Challenges
	§ III-C: Metrics
	§ III-D: Tasks
	§ III-E: Benchmarks
	§ III-F: Scenario Settings
§ IV Methods Review	§ IV-A: Taxonomy Methodology
	§ IV-B: Policy-focused Methods
	§ IV-C: Experience-focused Methods
	§ IV-D: Dynamic-focused Methods
	§ IV-E: Reward-focused Methods
	§ IV-F: Beyound Traditional CRL
	§ IV-G: Applications
§ V Future Works	§ V-A: Task-free CRL
	§ V-B: Evaluation and Benchmark
	§ V-C: Interpretable Knowledge
	§ V-D: Large-scale Pre-trained Model
	§ V-E: Embodied Agent
§ VI: Conclusion	

The primary contributions of this survey lie in a comprehensive survey of research in CRL. Specifically:

1) *Challenges*: We highlight the unique challenges faced by CRL, emphasizing the need for a triangular balance among plasticity, stability, and scalability.
2) *Scenario Settings*: We categorize CRL scenarios into lifelong adaptation, non-stationarity learning, task incremental learning, and task-agnostic learning. This provides a standardized framework for comparing the scope of CRL methods.
3) *Taxonomy*: We present a new taxonomy of CRL methods, categorizing them based on the type of knowledge stored and/or transferred. This taxonomy includes policy-focused, experience-focused, dynamic-focused,

and reward-focused methods, providing a structured overview for understanding the diverse CRL strategies.
4) *Method Review*: We provide the most updated literature review of CRL methods, including detailed discussions of seminal works, recently published articles, and promising preprints.
5) *Open Challenges*: We discuss open challenges and future research directions in CRL, including task-free CRL, evaluation and benchmarking, interpretable knowledge, and the integration of pre-trained large models.

Table I shows the structure of this survey. The remainder is organized as follows: Section II provides a foundational overview of RL and CL, essential for understanding CRL. Section III outlines the scope of CRL, including its definition, challenges, metrics, tasks, benchmarks, and scenario settings. Section IV introduces our proposed taxonomy and presents a detailed review of existing CRL methods. These methods can be organized by the type of knowledge they store and/or transfer, including policy-focused (Section IV-B), experience-focused (Section IV-C), dynamic-focused (Section IV-D), and reward-focused methods (Section IV-E). Section V discusses open challenges and future directions in CRL, aiming to guide further research and development in this promising field. Finally, Section VI presents some concluding remarks.

II. BACKGROUND

A. Reinforcement Learning

RL is a fundamental paradigm in machine learning, where an agent interacts with an environment to learn optimal behaviors through trial and error. A common framework used to model the interaction is the *Markov Decision Process* (MDP) [24]. A standard MDP is defined as: $M = \langle \mathcal{S}, \mathcal{A}, R, \gamma, T, \rho_0, H \rangle$. Here, $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively, which constrain the form and range of states and actions. The initial state s_0 follows a probability distribution $\rho_0 \in [0, 1]$, and the agent interacts with the environment starting from one of these initial states. The transition distribution of the environment is $T(s'|s, a) \in [0, 1]$, describing the probability of transitioning to a particular state when an action is performed. The reward function is represented as $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, indicating the feedback signal after a state transition. The episode length and discount factor are denoted by $H \in \mathbb{N}$ and $\gamma \in [0, 1]$ respectively, with H typically being the maximum length from the initial state to a terminal state if the environment has one, and γ representing the attenuation of future rewards, indicating the agent's preference for immediate rewards over distant ones.

In an MDP, an agent interacts with the environment by selecting actions based on a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, which maps from the state space to the action space. This generates a trajectory $\tau = (s_0, a_0, s_1, \cdots, s_{H-1}, a_{H-1}, s_H)$, composed of states and actions at each timestep, reflecting the complete information of the interaction process. The likelihood of generating a trajectory τ under policy π is given by [1]:

$$P_\pi(\tau) := \rho_0 \prod_{t=0}^{H-1} \pi\left(a_t|s_t\right) T\left(s_{t+1}|s_t, a_t\right). \tag{1}$$

The value of a policy is the expected sum of discounted rewards [25]:

$$V_\pi := \mathbb{E}_{\tau \sim P_\pi}\left[\sum_{t=0}^{H-1} \gamma^t R\left(s_t, a_t\right)\right]. \tag{2}$$

Hence, the objective of standard RL is to find an optimal policy π^* that maximizes this value:

$$\pi^\star := \underset{\pi}{\operatorname{argmax}}\left(V_\pi\right). \tag{3}$$

RL algorithms are broadly categorized into value-based and policy-based methods. Value-based methods aim to find the optimal policy indirectly by learning and optimizing value functions, which include the state value function $V(s)$ and state-action value function $Q(s, a)$. They estimate the value of each state and state-action pair, respectively. The central challenge is accurately estimating the value functions and improving the policy based on those estimates.

In value-based approaches, *Q-learning* stands out as a typical and widely-used algorithm, which is based on the *Temporal-Difference* (TD) method [1]. It updates the estimated value of $Q(s, a)$ using the Bellman optimality equation as follows:

$$\begin{aligned} &Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \\ &\alpha\left[R(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\right], \end{aligned} \tag{4}$$

where α is the learning rate. *Q-learning* iteratively updates the Q values until convergence to the optimal action-value function $Q^\star(s, a)$. The optimal policy π^* can then be derived using a greedy algorithm that selects actions maximizing $Q^\star(s, a)$ in each state s.

When combined with deep learning, *Q-learning* employs deep neural networks to approximate action functions and strategies like experience replay to enhance stability, leading to the development of the *Deep Q Network* (DQN) algorithm [2]. Subsequent improvements such as *prioritized experience replay* [26], *Dueling Double DQN* (D3QN) [2], [27], and *Deep Recurrent Q-learning* (DRQN) [28] addressed issues like sample utilization and training instability.

Conversely, policy-based methods optimize the policy function directly. The *Policy Gradient* (PG) algorithm is a fundamental work that parameterizes the policy $\pi(a|s; \theta)$ and optimizes θ using gradient ascent to maximize the expected return [1]. The policy gradient theorem estimates the gradient of the expected return with respect to policy parameters using collected samples:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim P\pi_\theta}\left[\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t|s_t) G_t\right], \tag{5}$$

where G_t is the discounted cumulative reward from time t to the end of the trajectory, and $J(\theta)$ is the objective function of the policy.

Policy-based methods have been further enhanced through the *Actor-Critic* (AC) architecture [29], which combines the strengths of value functions and policy gradients. Improvements such as *Deep Deterministic Policy Gradient* (DDPG)

[30], *Twin Delayed DDPG* (TD3) [31], *Trust Region Policy Optimization* (TRPO) [32], *Proximal Policy Optimization* (PPO) [33], and *Soft Actor-Critic* (SAC) [34] address issues like policy uncertainty and aggressive updating.

B. Continual Learning

CL is an emerging paradigm in machine learning that focuses on incrementally updating models to adapt to new tasks while maintaining performance on previous tasks. In CL, the learner accumulates knowledge over time to enhance the model's effectiveness while saving computational resources and time through incremental modeling, providing a viable solution for a series of tasks under resource constraints [35]. The learning process in CL can be mathematically represented as follows [36]:

$$\langle h_{k-1}, U_{k-1}, D_k \rangle \rightarrow \langle h_k, U_k \rangle, \tag{6}$$

where h_{k-1} and h_k are the CL models for tasks $k-1$ and k, respectively; U_{k-1} and U_k are the auxiliary information extracted from tasks $k-1$ and k; D_k is the incremental data for task k. As the distribution of data changes, the variety of tasks increases, and the complexity of models deepens, CL faces multiple challenges.

Two of the most pressing challenges are *catastrophic forgetting* and *knowledge transfer* [16], which have garnered significant attention. Catastrophic forgetting refers to the degradation of model performance on previous tasks upon learning new ones, a phenomenon attributed to the inherent limitations of current neural network technologies, which diverge from the continuous learning mode of the human brain [15]. Knowledge transfer, on the other hand, involves leveraging knowledge from previous tasks to facilitate learning on new tasks. To mitigate catastrophic forgetting and facilitate knowledge transfer, CL approaches strive for a balance between stability, to protect acquired knowledge, and plasticity, to adapt to new tasks efficiently. This balance, often referred to as the *stability-plasticity dilemma*, is critical for the success of CL systems.

Research in CL has proposed various strategies to overcome these issues, broadly classified into three categories:

1) **Replay-based methods** [37]–[41]: These approaches retain a subset of information relevant to previous tasks to aid the model in recalling past knowledge during new task learning, addressing the challenge of forgetting while also considering the privacy implications.
2) **Regularization-based methods** [42]–[45]: These methods constrain model updates to prevent conflicts between parameters learned from old and new tasks, offering a solution that doesn't rely on data retention but may compromise model flexibility.
3) **Parameter isolation methods** [46]–[49]: These methods segregate model parameters for different tasks, avoiding conflicts and adapting the model structure dynamically to accommodate complex task sequences, thus combining the benefits of scalability and privacy preservation without the drawbacks associated with data replay or excessive regularization.

The diversity of tasks, ranging from classification to robot control, introduces variability in goals, data distributions, and input formats, further complicating the CL landscape [18], [50], [51]. CRL is a special form of this diversity, focusing on RL tasks and providing a unique perspective on the challenges and solutions in the CL field.

III. OVERVIEW

This section provides an overarching overview of the research on CRL [52], [53], focusing on its definition, evaluation, and relevant scenario settings.

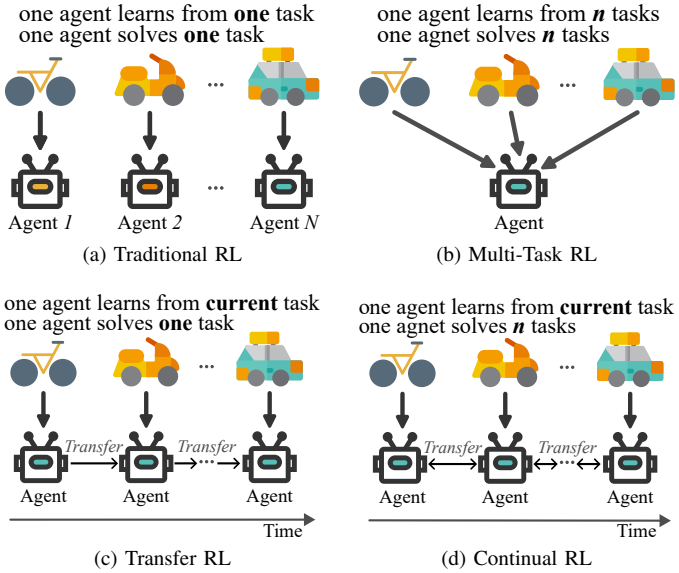

Fig. 2. A comparison of four RL paradigms.

A. Definition

The term "*Continual Reinforcement Learning*" can be broken down into two main components: "*continual*" and "*reinforcement learning*". While "*reinforcement learning*" remains the core subject of study, the term "*continual*" emphasizes the extension of traditional RL to a dynamic, multi-task framework, where agents continuously learn, adapt, and retain knowledge across various tasks.

In CRL, the learning process is typically modeled using MDPs, similar to traditional RL. The general structure includes a state space $\mathcal{S}$, an action space $\mathcal{A}$, an observation space $\mathcal{O}$, a reward function $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, a transition function $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$, and an observation function $\Omega : \mathcal{S} \rightarrow \mathcal{O}$. The most general form of CRL can be defined as [21]:

$$M_{\text{CRL}} := \langle \mathcal{S}_t, \mathcal{A}_t, R_t, T_t, \Omega_t, \mathcal{O}_t \rangle. \tag{7}$$

In this form, each function or space varies over time t, reflecting the inherent non-stationarity of CRL. In practical research, specific assumptions about this non-stationarity are commonly made. For example, t is often treated as a discrete variable representing different tasks, where the state space $\mathcal{S}$, action space $\mathcal{A}$, observation function Ω, and observation space $\mathcal{O}$ remain constant. In contrast, the reward function R and the

transition function T are typically considered as time-varying, piecewise functions of t. These assumptions regarding non-stationarity are vital to CRL, as they ensure that the learning process remains consistent.

Several related learning paradigms, such as *Multi-Task Reinforcement Learning* (MTRL) and *Transfer Reinforcement Learning* (TRL) , also aim to address multiple RL tasks. A comparison between traditional RL, MTRL, TRL, and CRL is presented in Fig. 2. The goal of MTRL is to train an agent that can handle multiple tasks simultaneously, where both source and target tasks belong to a fixed and known set [54]. TRL focuses on transferring knowledge from source tasks to target tasks, facilitating faster learning on the target tasks [55]. In contrast, CRL is designed for environments that change continuously, with tasks often arriving sequentially over time. The primary objective is to enable an agent to accumulate knowledge over extended periods and quickly adapt to new tasks as they arise. CRL shares similarities with both MTRL and TRL. MTRL typically employs a cross-task shared structure that allows the agent to handle multiple tasks simultaneously and can be viewed as a timeless version of CRL. TRL, which focuses on knowledge transfer between tasks, can be regarded as a subset of CRL, where forgetting is not considered. Thus, CRL can be considered a more generalized learning paradigm that encompasses the domains of MTRL and TRL. Even continual supervised learning and traditional RL can be viewed as special cases within the broader framework of CRL [52].

B. Challenges

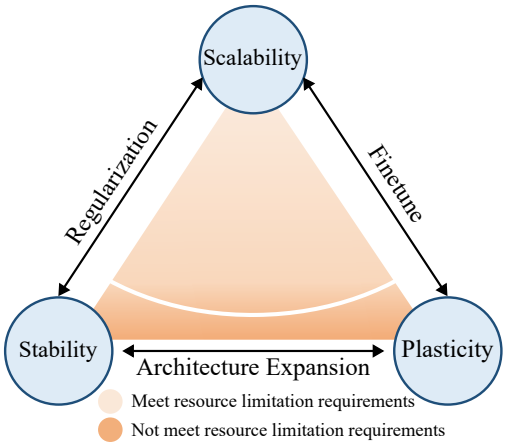

Fig. 3. The triangular balance of plasticity, stability, and scalability in CRL. Scalability determines the usability of CRL methods, while low plasticity fails to meet resource limitation requirements.

Research on CRL faces several challenges that distinguish it from traditional RL. Inspired by the CL challenges in supervised learning [18], the primary challenge in CRL can be described as achieving a triangular balance among three key aspects: plasticity, stability, and scalability. Fig. 3 illustrates the relationship between these three aspects:

1) **Stability** refers to an agent's ability to maintain performance on previously learned tasks while simultaneously learning new tasks. Stability is closely related to the problem of catastrophic forgetting, where learning new tasks causes a significant decline in performance on previously learned tasks. Addressing catastrophic forgetting is a key focus in CRL to ensure that the agent retains knowledge over time.
2) **Plasticity** refers to an agent's ability to learn new tasks after being trained on previous tasks. A critical component of plasticity is the agent's transfer ability, which enables it to leverage knowledge from previously learned tasks to enhance performance on new tasks (forward transfer) or on earlier tasks (backward transfer). High plasticity ensures that the agent can adapt to new tasks effectively.
3) **Scalability** refers to the ability of an agent to learn many tasks using limited resources. This aspect involves the efficient use of memory and computation, as well as the agent's capacity to handle increasingly complex and diverse task distributions. Scalability is particularly critical in real-world applications, where agents must efficiently adapt to a wide range of tasks.

The balance between these three aspects is crucial for the success of CRL algorithms. In the early stages of CL research, the primary focus was on stability, with significant efforts directed toward mitigating catastrophic forgetting [42], [56]. In recent years, more CL studies have recognized the importance of plasticity, emphasizing the need for effective knowledge transfer and adaptation to new tasks [53], [57]. Currently, the issue of plasticity loss has even become a crucial concern in DRL research [58], [59].

In the broader CL literature, the interplay between plasticity and stability is often referred to as the *stability-plasticity dilemma* [18], which underscores the inherent trade-off between these two aspects. Moreover, in the context of CRL, scalability emerges as an equally critical factor. Unlike supervised learning, RL algorithms typically require substantial computational resources and memory to learn complex tasks. In addition, if resource constraints are ignored, agents could theoretically store all data and train a separate model for each task. However, such an approach contradicts the principles of continual learning, which aim to develop resource-efficient algorithms that generalize across tasks [18]. Therefore, CRL research must address the unique challenge of balancing plasticity, stability, and scalability to enable agents to learn effectively in dynamic, real-world environments. This triangular balance serves as a guiding framework for the development of robust and efficient CRL algorithms.

C. Metrics

The longstanding tradition in RL has been to measure an agent's performance by recording its expected accumulated reward over an episode. While reward serves as a meaningful quantitative measure, it may not fully capture the agent's capabilities in all scenarios. For instance, the success rate better reflects the agent's ability to solve specific tasks [60]–[62]. However, both reward and success rate are metrics designed for single-task evaluation and do not adequately assess an

agent's CL ability. To address this limitation, research in CRL often incorporates common metrics from CL to evaluate the agent's performance across a sequence of tasks. These metrics are particularly useful in scenarios where task boundaries are well-defined and the number of tasks is finite. Consider a learning sequence with N tasks, where $p_{i,j} \in [0,1]$ represents the normalized performance (e.g., expected accumulated reward or success rate) of task j after the agent has been trained on task i. Using this notation, several key metrics are commonly employed in CRL:

Average Performance: This metric provides a holistic view of the agent's performance across all tasks it has encountered up to a given point in the sequence. For task i, the average performance is defined as:

$$A_i := \frac{1}{i}\sum_{j=1}^{i} p_{i,j}. \tag{8}$$

The final value, $A_N \in [0,1]$, summarizes the agent's overall performance across all N tasks. This metric is widely used in CL research to evaluate the cumulative efficacy of the learning process.

Forgetting: One of the primary goals in CRL is to prevent catastrophic forgetting, ensuring that an agent retains its performance on previously learned tasks while learning new ones. The forgetting metric quantifies the degree to which an agent's performance on a task declines after subsequent training. For task i, forgetting is defined as:

$$FG_i := \max(p_{i,i} - p_{N,i}, 0). \tag{9}$$

The overall forgetting metric, $FG \in [0,1]$, is the average of FG_i values for all tasks, providing insight into how much knowledge the agent retains over time.

Transfer: Effective transfer of knowledge is another critical aspect of CRL. Transfer can occur in two directions: forward transfer and backward transfer. **Forward transfer** measures the positive impact of learning a task on the performance of subsequent tasks. For task i, forward transfer is defined as:

$$FT_i := \frac{1}{N-i}\sum_{j=i+1}^{j\leq N} p_{i,j} - p_{i-1,j}. \tag{10}$$

The average forward transfer, denoted as $FT \in [-1,1]$, is the mean of FT_i values across all tasks. Moreover, **backward transfer** reflects the agent's ability to improve its performance on previously learned tasks after encountering new, related tasks. For task i, backward transfer is defined as:

$$BT_i := \frac{1}{N-i}\sum_{j=i-1}^{j\geq 1} p_{i,j} - p_{i-1,j}. \tag{11}$$

The average backward transfer $BT \in [-1,1]$ is the mean of BT_i values across all tasks.

Others: As shown in Table II, the above metrics provide a relatively comprehensive evaluation framework and are widely used in CRL benchmarks. Average performance captures the agent's overall learning ability, forgetting quantifies knowledge retention, and transfer evaluates the agent's ability to leverage prior knowledge for future tasks or improve past performance. Moreover, some benchmarks have introduced additional metrics such as **generalization improvement score** [63] and **performance relative to a single-task expert** [64] to more finely assess the agent's performance. However, they are not considered the scalability. Therefore, in addition to evaluation metrics related to an agent's performance, some efficiency-related metrics have been imported in recent works. For example, **model size** is the final number of parameters of a model after training on all tasks. This metric is used to measure CL methods with scalable model sizes [65]. Additionally, **sample efficiency** measures the number of samples required for an agent to achieve maximal performance [66], which can reflect the transfer ability of CRL agents from another perspective.

D. Tasks

In the domain of CRL, most agents are tasked with objectives at each step of a task sequence that aligns with the goals of RL tasks. This positions tasks as the foundational units of CRL. This section aims to introduce existing tasks within CRL and provide a succinct analysis of them.

Navigation tasks are one of the most commonly employed scenarios in CRL, often utilizing two-dimensional state spaces and a discrete set of actions. In these tasks, agents must explore unknown environments via continuous movement to reach a designated goal. Researchers frequently design task sequences based on grid-world environments [1], where rewards or environmental dynamics vary to assess CRL algorithms [71]. These tasks are relatively simple to learn and provide a lower difficulty for computationally intensive CRL algorithms. Furthermore, navigation tasks lend themselves well to environment procedural generation, which is essential for CRL. MiniGrid [72][2] is the most widely used environment library, offering a variety of map sizes and layouts for task generation. It provides preset environments like Doorkeyenv, Fourroomsenv, and Memoryenv for constructing diverse CRL task sequences. Additionally, JBW offers a testbed for lifelong learning by generating non-stationary environments within a 2D grid world. For more realistic evaluations, 3D navigation tasks, such as those based on DeepMind Lab [73], have been used to further assess CRL algorithms [74].

Control tasks are another prevalent CRL task type, typically involving three-dimensional state spaces and a discrete action set. Classic examples include the mountain car, inverted pendulum, and double pendulum tasks [3], where agents must reach specific target states (e.g., the peak of the mountain, an upright position, or a target height) using simple control commands (e.g., forward, backward, left turn, right turn) [75]. Task sequences in control tasks are often formed by altering the objectives [76] or by switching between different tasks [56], enabling the evaluation of CRL algorithms. In more complex tasks, agents are required to control robotic devices, such as robotic arms and legs [77]. These tasks involve physical properties, presenting significant challenges while also offering practical applications. Researchers typically modify

[2]https://minigrid.farama.org

[3]https://www.gymlibrary.dev/environments/classic_control

TABLE II
COMPARISON OF CONTINUAL REINFORCEMENT LEARNING BENCHMARKS.

Benchmark	3D	Number of Sequences	Length of Sequences	Partially Observable	Multi-Agent	Image Observation	Metrics[1]
CRL Maze [67]	✔	4	3	✔	✗	✔	A_N
Lifelong Hanabi [63]	✗	✗	✗	✔	✔	✗	A_N, FG, FT, GIS
Continual World [60]	✔	2	10,20	✗	✗	✗	A_N, FG, FT
L2Explorer [64]	✔	✗	✗	✔	✗	✔	FG, FT, BT, PR, SE
CORA [68][2]	✔& ✗	4	4,6,15	✔& ✗	✗	✔& ✗	A_N, FG, FT
Lifelong Manipulation [69]	✔	✗	10	✗	✗	✗	A_N
COOM [70]	✔	7	4,8,16	✔	✗	✔	A_N, FG, FT

[1] "A_N" stands for the average performance. "FG" stands for the forgetting. "FT" stands for the forward transfer. "BT" stands for the backward transfer. "GIS" stands for the generalization improvement score. "PR" stands for the performance relative to a single-task expert. "SE" stands for the sample efficiency.
[2] CORA is based on four environments with different features.

parameters such as limb length, mass, environmental friction, and gravity within control tasks to create diverse task sequences for CRL evaluation [65], [66], [78].

Video games present challenging reinforcement learning tasks, where the state space typically consists of images, and the actions are discrete. Within these environments, agents must perform complex controls to achieve specific goals, making video games an ideal testbed for evaluating the scalability of CRL algorithms in challenging scenarios [79], [80]. The Atari 2600 [4], with its collection of games that share consistent state and action spaces, is one of the most frequently used sets in DRL experiments [2], [81]. Researchers evaluate CRL algorithms by combining different games into task sequences [42], [74], [82]. For more complex tasks involving long-horizon strategy games with rich observations, some works have explored environments like MineCraft and StarCraft II [83]–[86]. However, these tasks are computationally expensive due to their large state spaces and complex task structures, requiring extended training durations [53].

Although most tasks are in simulated environments, some studies have also applied CRL algorithms to real-world robotic tasks. These include 2D navigation tasks for mobile robots [87], [88], robotic arm control tasks [89], and home robot tasks [90]. They present additional challenges, such as sensor noise, mechanical constraints, and the need for robust online learning, making them a practical direction for the further development of CRL techniques.

E. Benchmarks

Although CRL has gained increasing attention in recent years, its growth has been relatively slow compared to continual supervised learning. One of the reasons for this slow development is the difficulty of reproduction and the large amount of computation required for experiments. Another reason is the lack of standardized benchmarks and metrics for evaluation [68].

Recently, several benchmarks have been proposed for CRL. Table II provides a comparison of these benchmarks [70]. These benchmarks vary in characteristics such as the number of tasks, the length of task sequences, and the type of observations. Below, we briefly describe the notable features of these benchmarks:

- **CRL Maze** [67] [5]: A 3D environment based on ViZDoom, featuring non-stationary object-picking tasks with modified attributes such as light, textures, and objects.
- **Lifelong Hanabi** [63] [6]: A partially observable multi-agent environment based on the card game Hanabi [91]. It challenges agents to cooperate and adapt in a dynamic, multi-agent setting.
- **Continual World** [60] [7]: This benchmark comprises a set of robotic manipulation tasks derived from Meta-World [92], testing agents on various robotic manipulation tasks.
- **L2Explorer** [64] [8]: A 3D Procedural Content Generation (PCG) world that includes five tasks within a single environment, providing a highly configurable and diverse set of challenges for CRL algorithms.
- **CORA** [68] [9]: Based on four different environments, each with unique features, CORA offers a comprehensive evaluation platform for CRL algorithms.
- **Lifelong Manipulation** [69]: This benchmark includes ten manipulation tasks designed to evaluate agents at different levels of difficulty. It is easier to train compared to the Continual World.
- **COOM** [70] [10]: Another benchmark based on various ViZDoom environments, COOM focuses on embodied perception tasks, providing a robust platform for evaluating CRL algorithms.

The primary challenge in creating benchmarks for CRL lies in the design of task sequences. This is a complex engineering task that requires careful consideration of various factors, such as task difficulty, task order, and the number of tasks. Currently, there is no single ideal benchmark, and different benchmarks focus on different aspects of CRL [68]. Therefore, building a comprehensive and standardized benchmark for CRL remains an ongoing challenge.

[4]https://www.gymlibrary.dev/environments/atari

[5]https://github.com/Pervasive-AI-Lab/crlmaze
[6]https://github.com/chandar-lab/Lifelong-Hanabi
[7]https://github.com/awarelab/continual_world
[8]https://github.com/lifelong-learning-systems/l2explorer
[9]https://github.com/AGI-Labs/continual_rl
[10]https://github.com/hyintell/COOM

TABLE III
A FORMAL COMPARISON OF TYPICAL CONTINUAL REINFORCEMENT LEARNING SCENARIOS.

Scenario	Learning[1]	Evaluation
Lifelong Adaptation	$\{M_k\}_{k=1}^K$	M_K
Non-Stationarity Learning	$\{M_k\}_{k=1}^K, R_i \neq R_j$ or $T_i \neq T_j$ for $i \neq j$	$\{M_k\}_{k=1}^K$, k is available
Task Incremental Learning	$\{M_k\}_{k=1}^K, R_i \neq R_j$ and $T_i \neq T_j$ for $i \neq j$	$\{M_k\}_{k=1}^K$, k is available
Task-Agnostic Learning	$\{M_k\}_{k=1}^K$	$\{M_k\}_{k=1}^K$, k is unavailable

[1] M_k is the MDP of task k, K is the identifier of the latest task, R_i is the reward function of task i, and T_i is the transition function of task i.

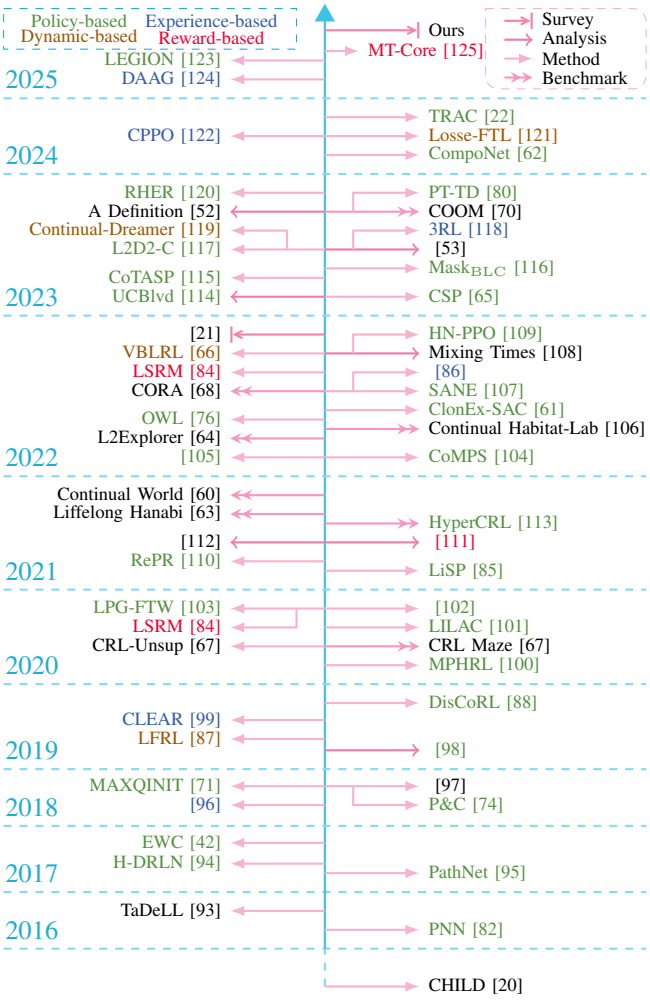

Fig. 4. Timeline illustrating the key developments, by order and interval, in the field of CRL from the end of 2016 to the present day (2025). The timeline includes methods, benchmarks, surveys, and analysis papers published in the field. At the start of the timeline, we highlight CHILD, which is described the first continual learning agent and pioneered the field of CRL. We then jump toward the middle of 2016, highlighting PNN, a stepping stone towards a full CRL agent. The dates shown in the timeline are the publication dates of the papers.

F. Scenario Settings

CRL scenarios can be categorized into four main types based on the non-stationarity property and the availability of task identities. A formal comparison of these scenarios is provided in Table III, which outlines the learning and evaluation processes for each scenario type. We summarize the key scenario types as follows:

Lifelong Adaptation: The agent is trained on a sequence of tasks, and its performance is evaluated only on new tasks. This scenario was prevalent in the early stages of CRL research [71], [126] and shares similarities with TRL, albeit with continual tasks. Lifelong adaptation can be viewed as a subproblem within the broader CRL framework, as its focus is on adapting to new tasks without addressing the full range of CRL challenges.

Non-Stationarity Learning: The tasks in the sequence differ in terms of their reward functions [65], [76] or transition functions [107], [127], but they share the same underlying logic. The agent is evaluated on all tasks in the sequence. While some studies have explored non-stationarity in action space or state space within lifelong adaptation settings [128], [129], this specific issue has not been thoroughly investigated within the broader context of CRL.

Task Incremental Learning: The tasks in the sequence differ significantly from one another in terms of both reward and transition functions [68], [70], [108]. These tasks are more distinct compared to those in non-stationary learning. Some tasks may even have different state and action spaces [130]. Moreover, a few studies have extended this scenario to include tasks from different domains [131], [132], increasing the diversity of tasks the agent must learn.

Task-Agnostic Learning: The agent is trained on a sequence of tasks without full knowledge of task labels or identities [107], [133]. The agent might not even be aware of task boundaries, requiring it to infer task changes from the data itself [80], [118], [134]. This scenario is particularly relevant to real-world applications, where agents often do not have explicit knowledge of the tasks they are solving or the states they are in.

In most CRL research, it is assumed that each task provides a sufficient number of steps for the agent to learn. Typically, the task sequence is fixed; however, some studies have explored dynamically generated task sequences [107], [135]. Recent research has also increasingly focused on scenarios where agents are unable to observe task boundaries, which adds a level of realism and complexity to the problem. Therefore, some researchers consider this scenario as CRL and refer to unsatisfactory scenarios as semi-continual RL [80]. While the above categories offer a structured view of CRL scenarios, it is important to note that the boundaries between these scenarios are not always clear-cut. Many studies integrate multiple scenarios to address more complex, general CRL problems [65], [66].

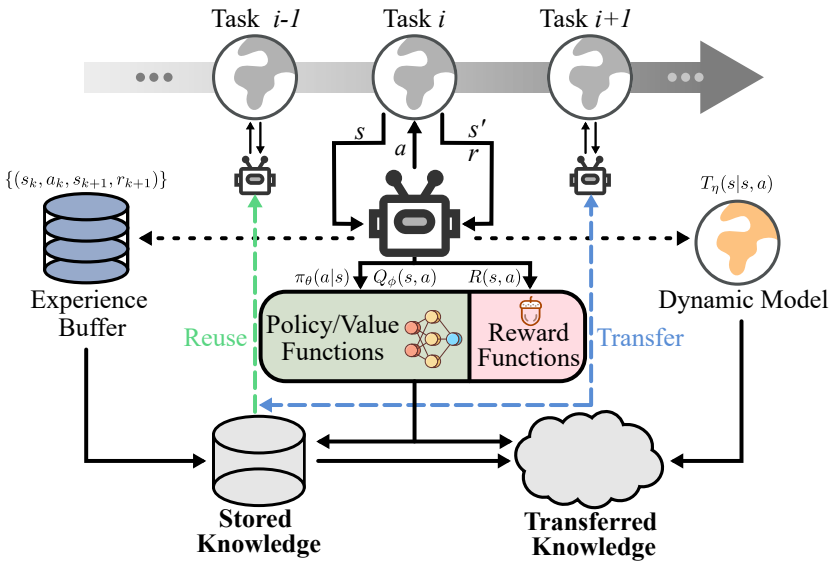

Fig. 5. Illustration of the general structure of a CRL method, organized by the knowledge that is stored and/or transferred.

IV. METHODS REVIEW

In this section, we present our taxonomy of CRL methods. Khetarpal *et al.* [21] proposed a taxonomy for CRL, classifying approaches into three categories: explicit knowledge retention, leveraging shared structures, and learning to learn. While this taxonomy provides valuable insights, it does not adequately provide the unique characteristics of CRL, and it falls short of encompassing the breadth of recent advancements in the field. To address these limitations, we propose a new taxonomy that focuses on the unique aspects of CRL, distinguishing it from traditional CL methods. Our taxonomy is grounded in the key components of RL and organizes CRL methods based on the type of knowledge they store and transfer. In addition, we provide the most updated and comprehensive review of CRL methods, including the latest advancements in the field. Fig. 4 presents a timeline with the representative methods in CRL, allowing one to evaluate the novelty and popularity of each class of methods.

A. Taxonomy Methodology

Fig. 5 illustrates the general structure of CRL methods. In this framework, an agent's knowledge can be broadly categorized into four main types: *policy*, *experience*, *dynamics*, and *reward*. While other elements in RL, such as action space and state space, can also be considered forms of knowledge, they are often overlooked in existing CRL methods. Therefore, our taxonomy primarily focuses on four categories, which are central to the design and implementation of CRL systems.

To systematically organize CRL methods, we address the following key question: **"What knowledge is stored and/or transferred?"** Based on this guiding question, we classify CRL methods into four main categories: policy-focused, experience-focused, dynamic-focused, and reward-focused. We further divide some categories into sub-categories based on how the knowledge is utilized. It is important to note that this taxonomy is not exhaustive, and many methods may span multiple categories. To facilitate a comprehensive overview of the development of CRL methods, we list representative approaches in Table IV, organized chronologically.

B. Policy-focused Methods

We start by introducing the policy-focused methods, which are the most common and fundamental in CRL. This is because the policy function or value function constitutes the core knowledge in RL, directly determining the agent's decision-making process. Among these methods, the fine-tuning strategy, where the agent inherits the policy or value function from a previous task, is widely adopted as a naive mechanism for knowledge transfer. This strategy often overlaps with other CRL approaches discussed later [65], [136]. The policy-focused methods can be further divided into three sub-categories: *policy reuse, policy decomposition, and policy merging*. Below, we provide a detailed discussion of them.

1) Policy Reuse: Policy reuse is a widely used strategy in CRL, where the agent retains and reuses complete policies from previous tasks. As illustrated in Fig. 6, the simplest approach involves storing all previously learned policies to prevent catastrophic forgetting and using them as a foundation for developing new policies. While most methods of policy reuse have limited scalability, it remains a common approach for implementing CRL agents, as it primarily focuses on knowledge transfer and adaptation.

One key form of policy reuse involves **initializing** the policy for a new task using a previously learned policy, followed by fine-tuning. The concept of "*jumpstart*" was introduced to evaluate the initial performance of an agent in a target task when knowledge is transferred from a source task [71]. The jumpstart objective [52], [80], [88] can be formalized as maximizing the expected value function of the initial state:

$$J(\pi) = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{M\in\mathcal{M}}[V_M^\pi(s_0)], \tag{12}$$

where $V_M^\pi(s_0)$ represents the value function of policy π in the initial state s_0 of task M (MDP), $\mathcal{M}$ is the set of tasks (MDPs).

TABLE IV
THE TAXONOMY OF CONTINUAL REINFORCEMENT LEARNING METHODS.

Category		Methods
§ IV-B Policy-focused	§ IV-B1: Policy Reuse	MAXQINIT [71], LFRL [87], [126], [137], [102], [138], CSP [65], SOPGOL [139], ClonEx-SAC [61], UCOI [140], SWOKS [141], LEGION [123]
	§ IV-B2: Policy Decomposition	PG-ELLA [142], [131], [143], TaDeLL [93], PNN [82], ePG-ELLA [144], MPHRL [100], LPT-FTW [103], [145], CDLRL-ZPG [132], OWL [76], SANE [107], [105], HLifeRL [146], PT-TD [80], COVERS [89], HVCL [130], RHER [120], CompoNet [62], DaCoRL [127]
	§ IV-B3: Policy Merging	EWC [42], PathNet [95], Benna-Fusi [56], P&C [74], Online-EWC [74], PC [78], DisCoRL [88], [147], VPC [148], BLIP [149], HN-PPO [109], CoTASP [115], MASK_{BLC} [116], UCBlvd [114], [150]
§ IV-C Experience-focused	§ IV-C1: Direct Replay	[96], CLEAR [99], CoMPS [104], [151], [152], 3RL [118], CPPO [122], [153], DAAG [124]
	§ IV-C2: Generative Replay	SLER [154], RePR [110], S-TRIGGER [155], [86]
§ IV-D Dynamic-focused	§ IV-D1: Direct Modeling	MOLe [156], [134], HyperCRL [113], VBLRL [66], LLIRL [157], [135], Losse-FTL [121]
	§ IV-D2: Indirect Modeling	LILAC [101], LiSP [85], 3RL [118], Continual-Dreamer [119]
§ IV-E: Reward-focused		ELIRL [158], [111], IML [159], SR-LLRL [160], LSRM [84], [161]

Building on this, MAXQINIT investigated four methods for initializing the state-action value function Q and found that the optimal initialization involves using the maximum estimated Q values across tasks in the empirical task distribution [71]:

$$\hat{Q}_{max}(s,a) = \max_{M \in \hat{\mathcal{M}}} Q_M(s,a), \tag{13}$$

where $\hat{\mathcal{M}}$ is the set of tasks the agent has sampled so far, and Q_M is the state-action value function learned from each task. Additionally, Mehimeh *et al.* [140] studied optimistic initialization of the value function. They proposed a method called *Uncertainty and Confidence aware Optimistic Initialization* (UCOI), which selectively applies optimism based on the uncertainty of state-action pairs, measured by the variability of past outcomes and confidence using PAC-MDP parameters. This selective optimism reduces unnecessary exploration and enhances learning efficiency.

Policy reuse methods can also enhance the **exploration** policy in RL algorithms by leveraging past policies. Wolczyk *et al.* [61] evaluated four exploration strategies for the SAC algorithm: random, preceding, uniform-previous, and best-return. Their proposed ClonEx-SAC method selects the best-return exploration policy due to its superior performance in knowledge transfer, particularly in task sequences with repeated tasks. Additionally, Meta-MDP [126] introduces a more sophisticated exploration policy by modeling the search for an optimal exploration policy as a meta-level MDP. This approach separates the agent's behavior into exploration and exploitation policies, allowing the exploration policy to be optimized across multiple tasks. Although it improves performance and learning efficiency, determining the appropriate exploration time remains a challenge.

Although the above methods demonstrate the effectiveness of policy reuse, their generalization capability is often constrained by the similarity between tasks. To address this limitation, some CRL methods have explored zero-shot generalization by leveraging **task composition** frameworks, such as Boolean task algebra and logical composition [102], [138], [139]. These frameworks enable agents to reuse learned policies by composing them to solve new tasks without requiring additional training.

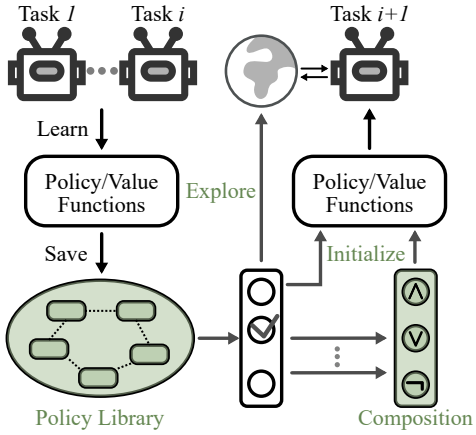

Fig. 6. The framework of policy reuse in CRL methods. Stored policies are reused to initialize new policies, enhance exploration, and improve generalization by leveraging task composition frameworks.

Task composition formalized using Boolean algebra provides a powerful mechanism for combining tasks through operations like conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$). These operations are defined over a set of tasks $\mathcal{M}$ within a Boolean algebra framework, which satisfies Boolean axioms [102]. Specifically:

1) The negation operator $\neg$ transforms a task M into a new task with a reward function $R_{\neg M}(s,a) = (R_{M_{\text{MAX}}}(s,a) + R_{M_{\text{MIN}}}(s,a)) - R_M(s,a)$, where $R_{M_{\text{MAX}}}(s,a) = \max_{M \in \mathcal{M}} R_M(s,a)$ and $R_{M_{\text{MIN}}}(s,a) = \min_{M \in \mathcal{M}} R_M(s,a)$.
2) The disjunction operator $\vee$ combines two tasks M_1 and M_2 into a task with a reward function $R_{M_1 \vee M_2}(s,a) = \max\{R_{M_1}(s,a), R_{M_2}(s,a)\}$.

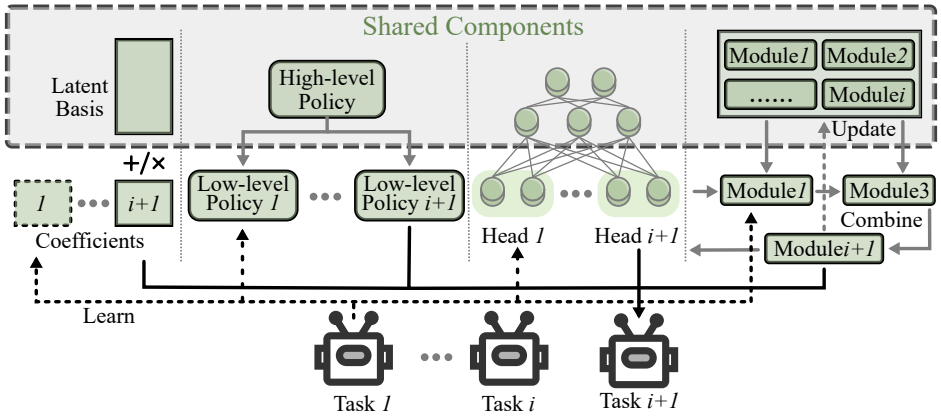

Fig. 7. The framework of policy decomposition in CRL methods. Factor decomposition, multi-head network, hierarchical decomposition, and modular architecture are used to decompose the policy into a shared base and task-specific components.

3) The conjunction operator $\wedge$ combines them into a task with a reward function $R_{M_1 \wedge M_2}(s,a) = \min\{R_{M_1}(s,a), R_{M_2}(s,a)\}$.

For goal-based tasks, this logical composition framework extends to goal-oriented value functions, enabling agents to compute optimal value functions for composed tasks directly. The extended Q-value function $\bar{Q}$ is defined as:

$$\bar{Q}(s,g,a) = \bar{R}(s,g,a) + \gamma \sum_{s'} P(s'|s,a)\bar{V}^{\bar{\pi}}(s',g), \quad (14)$$

where g is a goal state, $\bar{R}$ is the extended reward function that penalizes undesired goal states, and $\bar{V}^{\bar{\pi}}$ is the value function under policy $\bar{\pi}$. Logical operations on extended Q-value functions are defined similarly [139]:

$$\begin{aligned} &\neg(\bar{Q}^\star)(s,g,a) \\ &= (\bar{Q}^\star_{M_{\text{MIN}}}(s,g,a) + \bar{Q}^\star_{M_{\text{MAX}}}(s,g,a)) - \bar{Q}^\star(s,g,a), \\ &\vee (\bar{Q}^\star_1, \bar{Q}^\star_2)(s,g,a) = \max\{\bar{Q}^\star_1(s,g,a), \bar{Q}^\star_2(s,g,a)\}, \\ &\wedge (\bar{Q}^\star_1, \bar{Q}^\star_2)(s,g,a) = \min\{\bar{Q}^\star_1(s,g,a), \bar{Q}^\star_2(s,g,a)\}. \end{aligned} \quad (15)$$

This framework facilitates zero-shot composition, enabling agents to solve new tasks immediately by leveraging previously acquired skills. Furthermore, the *Sum Of Products with Goal-Oriented Learning* (SOPGOL) framework extends logical composition to stochastic and discounted tasks, allowing agents to determine whether a new task can be solved using existing skills or if new task-specific skills need to be learned [138]. These methods reduce the need for sample collection and training, thereby improving efficiency [102], [138].

To further improve the scalability of policy reuse, *Continual Subspace of Policies* (CSP) introduces a novel approach by maintaining a **subspace** of policies instead of learning a single policy [65]. CSP represents this subspace as a convex hull in the parameter space, with its vertices, called anchors, corresponding to the parameters of individual policies. A new policy can be derived as a convex combination of these anchors, or a new anchor can be added to the convex hull to represent a newly learned policy. This approach allows the model size to grow sublinearly with the number of tasks, achieving a better balance between performance and scalability.

2) Policy Decomposition: Policy decomposition is another widely used strategy in CRL, where the agent decomposes the policy into multiple components and reuses them in various ways. The primary challenge in policy decomposition lies in determining how to effectively decompose the policy. As shown in Fig. 7, this can be achieved through four main approaches: *factor decomposition*, *multi-head decomposition*, *modular decomposition*, and *hierarchical decomposition*.

Factor decomposition is mainly used in early CRL methods without deep learning, where the policy is decomposed into a shared base and task-specific components. This approach is from multi-task supervised learning, and has been successfully applied in CRL by *Policy Gradient Efficient Lifelong Learning Algorithm* (PG-ELLA) [142], [144]. PG-ELLA introduces a latent basis representation to model each task's parameters as a linear combination of components from a shared knowledge base. Specifically, the policy parameters for task k are represented as:

$$\boldsymbol{\theta}_k = \boldsymbol{L}\boldsymbol{s}_k, \quad (16)$$

where $\boldsymbol{L}$ is the shared latent basis and $\boldsymbol{s}_k$ are the task-specific coefficients. This formulation allows agents to accumulate and transfer knowledge across tasks.

However, PG-ELLA trains individual task policies first, which may lead to incompatibility with the shared base. To address this, *Lifelong PG: Faster Training Without forgetting* (LPG-FTW) [103] optimizes task-specific coefficients directly using policy gradients, ensuring compatibility with the shared base while leveraging shared knowledge to accelerate learning. This modification enables LPG-FTW to handle more complex dynamical systems and operate in a lifelong learning setting.

Building on the foundations of PG-ELLA, subsequent research has improved the efficiency and scalability of the factor decomposition method by the kernel method [145] or the multiple processing units assumption [144]. Furthermore, the introduction of cross-domain lifelong RL frameworks [131] enables agents to efficiently learn and generalize across multiple task domains. This is achieved by partitioning the series of tasks into task groups, such that all tasks within a particular

group $\mathcal{G}$ share a common state and action space. Then, the policy parameters for task k in group $\mathcal{G}$ are represented as:

$$\boldsymbol{\theta}_k = \boldsymbol{\Psi}^{(\mathcal{G})} \boldsymbol{L} \boldsymbol{s}_k, \tag{17}$$

where $\boldsymbol{\Psi}^{(\mathcal{G})}$ is a group-specific projection matrix that maps the shared latent basis $\boldsymbol{L}$ to the task-specific coefficients $\boldsymbol{s}_k$.

Further advancements include the integration of task descriptors for zero-shot knowledge transfer. *Task Descriptors for Lifelong Learning* (TaDeLL) assumes that task descriptors $\phi(\mathbf{m}_k)$ can be linearly factorized using a latent basis $\mathbf{D}$ over the descriptor space, coupled with the policy basis $\mathbf{L}$ to share the same coefficient vectors $\mathbf{s}_k$ [93]:

$$\phi(\mathbf{m}_k) = \mathbf{D}\mathbf{s}_k. \tag{18}$$

This allows for consistent task embeddings across policies and descriptors, enhancing learning efficiency and enabling zero-shot transfer. *Cross-Domain Lifelong Reinforcement Learning algorithm with Zero-shot Policy Generation ability* (CDLRL-ZPG) [132] further extends the zero-shot ability by constructing a linear mapping from environmental coefficients $\mathbf{q}_k^{\mathcal{D}}$ to task-specific coefficients $\mathbf{s}_k^{\mathcal{D}}$ using a matrix $W^{\mathcal{D}}$ in learned task domain $\mathcal{D}$:

$$\mathbf{s}_k^{\mathcal{D}} = W^{\mathcal{D}} \mathbf{q}_k^{\mathcal{D}}. \tag{19}$$

This mapping allows the generation of approximate optimal policy parameters for new tasks directly from environmental information, significantly improving generalization across different task domains without additional learning.

Finally, although PT-TD learning is not a factor decomposition method, it also uses a similar idea to decompose the value function, and it is also agnostic to the nature of the function approximator used [80]. PT-TD learning decomposes the value function into two components that update at different timescales: a permanent value function for preserving general knowledge and a transient value function for learning task-specific knowledge. Then the overall value function is computed additively:

$$V^{(\mathrm{PT})}(s) = V_\theta^{(\mathrm{P})}(s) + V_\phi^{(\mathrm{T})}(s), \tag{20}$$

where θ and ϕ are the parameters of the permanent function $V^{(\mathrm{P})}(s)$ and transient value function $V^{(\mathrm{T})}(s)$, respectively. The parameters of the transient value function are updated by TD learning during learning on tasks, while the parameters of the permanent value function are updated on all stored states of the task after learning. This work also provides some theoretical results for the convergence and upper bound on errors of the value functions.

Due to the advances in DRL and the increasing complexity of tasks, many CRL methods have evolved to incorporate deep neural networks. Policy function networks and value function networks can be decomposed into multiple parts, such as multiple heads and multiple modules. By combining these parts, agents can learn and generalize across tasks more effectively, enhancing scalability and performance in complex environments.

Multi-head decomposition is a common strategy in multi-task learning, where the network consists of a shared backbone and multiple heads, each responsible for a different task. In CRL, Wolczyk *et al.* [61] empirically investigated the impact of multi-head networks on the continual learning performance of SAC. By assigning separate output heads for each task, the agent facilitates the transfer of knowledge across tasks. However, freezing the backbone of the critic can hinder knowledge forward transfer, which is against the understanding of transfer in supervised learning. Additionally, the agent's performance can benefit from the resetting of the head for the critic while being damaged by the resetting of the head for the actor.

Furthermore, *cOntinual RL Without confLict* (OWL) finds that a multi-head network is suitable for dealing with the problem of interference, in which tasks have different goals (reward functions) [76]. In these cases, tasks may be fundamentally incompatible with each other and thus cannot be learned by a single policy. By employing a shared feature extractor combined with separate policy heads for each task, OWL prevents interference between tasks, as the dedicated heads allow the network to learn task-specific policies without overwriting previously acquired knowledge. Furthermore, OWL extends the multi-head network to the sequence with unknown task identifiers by modeling the head selection as a multi-armed bandit problem.

Expanding the application of the multi-head network to dynamic and non-stationary environments, *Dynamics-adaptive Continual RL* (DaCoRL) incorporates a context-conditioned multi-head design to detect and adapt to environmental changes [127]. Each head corresponds to a specific context, defined by a set of tasks with similar dynamics, allowing the network to specialize in context-specific policies. The framework dynamically expands its architecture by adding new heads when novel contexts are detected, ensuring scalability and adaptability to previously unseen scenarios.

Multi-head decomposition divides a policy network or value network into two components, which may not fully address the complexity of relationships among tasks. A more granular partitioning strategy will enable finer control over the transfer and retention of knowledge. **Modular decomposition** is an efficiency strategy in MTL [162]–[164], which leverages the composition of specialized modular deep architectures to capture compositional structures that arise in complex tasks. It has some similarities with the inner workings of the human brain and has been proven evidence of their biological plausibility [165], [166]. *Progressive Neural Networks* (PNN) has made early explorations in this direction [82], although it does not introduce the concept of modularity. PNN trains a new column of network parameters for each task, while lateral connections are established between corresponding layers of all previously trained columns. This design achieves strong forward transfer without overwriting previously learned information. However, the scalability of PNNs is a notable limitation, as the network's size grows linearly with the number of tasks. The subsequent modular methods achieve better scalability through explicit network modules.

An early effort in lifelong learning introduced a framework with a two-stage learning process, separating the reuse of existing knowledge (assimilation) from the improvement of old components or creation of new components (accommodation) [167]. It emphasizes compositionality, where tasks are

represented as combinations of reusable components. Mathematically, if a task k can be solved by reusing existing modules $\{m_i\}_{i=1}^n$ from a module set $\boldsymbol{M}$, the solution function f_k can be represented as a composition of these modules:

$$f_k(s) = m_1 \circ m_2 \circ \cdots \circ m_k(s), \quad m_i \in \boldsymbol{M}, \tag{21}$$

where $\circ$ denotes the compositional operation (e.g., functional composition, linear combination, or hierarchical stacking). Then, they extended this idea to CRL by formalizing lifelong compositional RL problems as a compositional problem graph, where each node represents a module to solve the corresponding subproblem [105]. Each module is a small neural network that takes as input the module-specific state component along with the output of the module at the previous module. The goal of a task is to find a path between nodes corresponding to a policy that maximizes the expected return. Although this method accurately captures the relations across tasks in the form of modules, it requires attempting all possible combinations of modules to find the optimal solution, which is also low in scalability.

Subsequent works have advanced the modular paradigm by focusing on dynamic and autonomous module management. *Self-Activating Neural Ensembles* (SANE) introduced a task-id-free method that automatically detects and responds to drift in the setting by maintaining an ensemble of modules [107]. It does this by activating, merging, and creating modules automatically. Each module m_i contains an actor and a critic. It is associated with an activation score $u_i(s)$, which determines its relevance to the current state s:

$$u_i(s) = |G_t - V_i(s)|, \tag{22}$$

where $V_i(s)$ is the value estimate of module i at state s and return G_t. The module with the highest activation score is selected for inference. The *Upper Confidence Bound* (UCB) for each module is used to balance exploration and exploitation:

$$V_i^{\mathrm{UCB}}(s) = V_i(s) + \alpha_u \cdot u_i(s), \tag{23}$$

where α_u is a hyperparameter that represents how wide a margin around the expected value to allow. Then, the activated module is:

$$m_{\mathrm{a}} = \arg\max_i V_i^{\mathrm{UCB}}(s). \tag{24}$$

SANE was later adapted for home robotics as SANER, which tailored the modular framework to low-data settings using attention-based interaction policies [90]. To further improve the scalability of the modular framework and avoid interference, *self-Composing policies Network architecture* (CompoNet) introduced attention mechanisms for selective knowledge transfer [62]. Each task corresponds to a module, and the modules of multiple tasks form a cascaded structure. Each module can access and compose the outputs of the previous modules by two attention heads and an internal policy to solve the current task. By allowing policies to compose themselves autonomously, CompoNet significantly reduces the memory and computational cost per task required and ensures linear growth of parameters with the number of tasks.

Inspired by the work related to hierarchical RL [168], [169], **hierarchical decomposition** is a prominent approach in policy-focused CRL that leverages the natural structure of tasks to organize policies into a hierarchy of reusable components. This approach is particularly effective in addressing complex tasks with multiple steps, as it allows agents to decompose tasks into simpler sub-policies or skills that can be reused across tasks. In hierarchical decomposition, the policy is typically structured into high-level controllers and low-level sub-policies, enabling efficient task execution and scalability in multi-task or lifelong learning settings. By organizing knowledge hierarchically, this approach also facilitates modularity, which is crucial for adapting to new tasks without interfering with previously learned ones.

A variety of approaches have been proposed to implement hierarchical decomposition in CRL, each emphasizing different mechanisms for skill discovery, knowledge storage, and transfer. For instance, *Hierarchical Deep Reinforcement Learning Network* (H-DRLN) integrates reusable *Deep Skill Networks* (DSNs) into a hierarchical framework [94]. This approach enables the agent to decompose tasks into reusable skills, reducing sample complexity and improving performance in high-dimensional environments like Minecraft. Similarly, the *Hierarchical Lifelong Reinforcement Learning framework* (HLifeRL) employs an option framework to automatically discover and store low-level skills in an option library [146]. The master policy in HLifeRL selects these options to execute tasks, allowing for efficient skill reuse and combating catastrophic forgetting. Another notable method is the *Model Primitive Hierarchical Reinforcement Learning* (MPHRL) framework, which utilizes model primitives (suboptimal world models) to decompose tasks into modular subpolicies [100]. This bottom-up approach enables the agent to learn sub-policies and a gating controller concurrently, enhancing knowledge transfer and scalability. Additionally, Hihn *et al.* [130] introduced a hierarchical information-theoretic optimality principle with the *Hierarchical Variational Continual Learning* (HVCL) framework. This framework uses a *Mixture-of-Variational-Experts* (MoVE) layer to create multiple information processing paths governed by a gating policy, facilitating specialized learning and mitigating catastrophic forgetting without requiring task-specific knowledge.

These methods share commonalities in their hierarchical structuring of knowledge but differ in their specific techniques for skill discovery and reuse. For example, H-DRLN relies on skill distillation to encapsulate multiple skills into a single network, while HLifeRL uses an explicit option framework to separate high-level decision-making from low-level execution. On the other hand, MPHRL emphasizes the use of model primitives to guide task decomposition, with a probabilistic gating mechanism to activate the appropriate subpolicy. HVCL introduces diversity objectives to enhance expert allocation and uses Wasserstein distance as a kernel for measuring expert diversity, ensuring distinct expert parameters are maintained. Despite these differences, all methods demonstrate improvements in sample efficiency, scalability, and the ability to transfer knowledge across tasks. For instance, HLifeRL and MPHRL both show that the cost of learning the first task is amortized over subsequent tasks, resulting in substantial long-term gains.

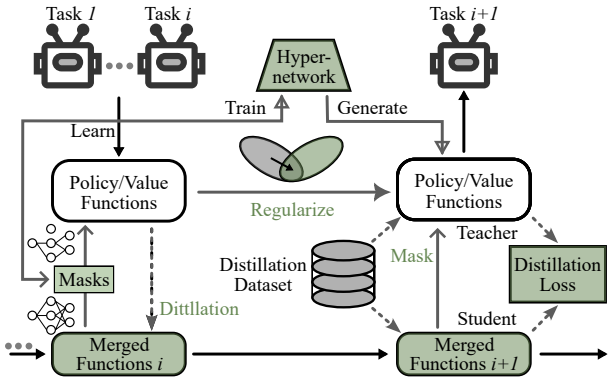

Fig. 8. The framework of policy merging in CRL methods. Distillation, masks, hypernetworks, and regularization are used to merge multiple policies into a single policy.

However, hierarchical decomposition methods face several challenges that remain open for future research. One key issue is the increasing complexity of the action space as the number of tasks and skills grows, which can strain the scalability of the hierarchical framework [146]. Additionally, the performance of these methods often depends on the quality and diversity of the discovered skills or model primitives [100]. Future work could explore more adaptive mechanisms for dynamic skill discovery and online refinement, as well as strategies to manage the complexity of growing skill libraries. Furthermore, extending these methods to real-world applications, such as robotics or natural language processing, remains an exciting avenue for research, requiring robust and efficient hierarchical frameworks capable of handling noisy and unstructured environments.

3) Policy Merging: Policy merging is a storage-sensitive strategy in CRL that focuses on merging the model of policies from multiple tasks into a single model, rather than retaining individual policies for each task. This approach is particularly useful in scenarios where memory constraint is a concern, as it allows agents to compress knowledge from multiple tasks into a more compact representation. By merging policies, agents can reduce the memory footprint and computational cost of storing and executing multiple policies, enabling more efficient and scalable continual learning. As illustrated in Fig. 8, these methods typically involve distillation, hypernetworks, or masks to combine policies and facilitate knowledge transfer across tasks.

Distillation is a common technique in supervised learning that has been adapted for CRL to merge policies and facilitate knowledge retention. This technique usually involves training a student policy on a new task to mimic the output of a teacher policy learned from previous tasks, effectively transferring knowledge from the old policies to the new ones. The *Progress & Compress* (P&C) framework [74] exemplifies this approach by integrating a knowledge base and an active column to sequentially learn tasks. After training on a new task, the active column's policy is distilled into the knowledge base using a cross-entropy loss, ensuring that previously acquired skills are preserved while facilitating forward transfer to new tasks. The framework also employs a modified version of *Elastic Weight Consolidation* (EWC) [42] to safeguard against catastrophic forgetting during the distillation process. Similarly, *Distillation for Continual Reinforcement Learning* (DisCoRL) [88] employs policy distillation to merge multiple task-specific policies into a single student policy. By generating distillation datasets from each task and using Kullback-Leibler divergence with temperature smoothing as a loss function, DisCoRL ensures effective knowledge transfer while eliminating the need for explicit task indicators at test time. These methods demonstrate the efficacy of policy distillation in achieving both knowledge retention and transfer, as evidenced by their strong empirical performance across diverse benchmarks, from Atari games to robotic navigation tasks.

Recent advancements have introduced novel techniques to enhance the distillation process. For example, an experience consistency distillation method for robotic manipulation tasks [150] combines policy distillation with experience distillation, leveraging *Fréchet Inception Distance* (FID) loss to maintain distribution consistency between original and distilled experiences. This approach not only mitigates forgetting but also optimizes memory usage by compressing experiences into a compact representation, which is then replayed during training. Furthermore, *UCB lifelong value distillation* (UCBlvd) [114] incorporates theoretical guarantees into the distillation process, ensuring sublinear regret and computational efficiency in CRL. By leveraging linear representations and *Quadratically Constrained Quadratic Programming* (QCQP), UCBlvd minimizes computational complexity while achieving effective policy merging across tasks.

These advancements also collectively highlight a trend toward leveraging distillation not only as a tool for policy merging but also as a means to improve data efficiency and scalability in CRL [94], [110], [151]. Despite their successes, challenges remain, such as optimizing the distillation process for diverse task distributions and ensuring the scalability of these methods to a larger number of tasks or more complex environments.

Some CRL methods have preliminarily explored the use of hypernetworks and masks to merge policies. **Hypernetworks** are neural networks that generate the weights of another neural network, allowing for the dynamic generation of task-specific policies [170]. The use of hypernetworks in CL has shown promise in merging policies while maintaining task-specific adaptability [171]. In CRL, *HyperNetwork-based implementation of PPO* (HN-PPO) [109] employs a hypernetwork to generate policy weights conditioned on task embeddings, enabling the agent to adapt to new tasks without discarding previously acquired knowledge. By regularizing the output of the hypernetwork, the method mitigates catastrophic forgetting and ensures stability across tasks. Similarly, Xu *et al.* [172] integrated a hypernetwork for state-conditioned action evaluation, which dynamically generates evaluators to adapt policies based on current states. This not only facilitates knowledge transfer but also enhances few-shot generalization, allowing the model to adapt more efficiently to new tasks. These methods demonstrate the versatility of hypernetworks in dynamically encoding task-specific policies within a shared architecture, reducing memory overhead while ensuring efficient

knowledge reuse. However, the computational complexity of hypernetwork training and the need for a refined training pipeline remain challenges for future research.

Masks offer another compelling avenue for policy merging by leveraging task-specific modulating masks to isolate and reuse knowledge. While the application of masks has been tested extensively in continual supervised learning for classification [46], [173], very little is known about their effectiveness in CRL [95]. One possible reason is that the previous mask methods lack the ability of knowledge transfer [116]. In order to address this limitation, a recent study has explored the use of the combination of previously learned masks to exploit previous knowledge when learning new tasks [116]. It introduces a fixed backbone network modulated by learned binary masks that selectively activate relevant parts of the network for each task. This approach not only preserves knowledge from prior tasks but also facilitates forward transfer by combining previously learned masks through linear or balanced linear compositions. Extending this method, the distributed system *Lifelong Learning Distributed Decentralized Collective* (L2D2-C) [117] enables agents to share task-specific masks in a decentralized manner, enhancing collective learning while maintaining robustness to connection drops. The results demonstrate the advantages of masks in continual multi-agent RL, which is a promising but underexplored area in CRL research.

Regularization is another effective strategy for policy merging, as it allows agents to retain knowledge from previous tasks while adapting to new ones. In supervised continual learning, regularization methods have been widely used to prevent catastrophic forgetting and are a promising direction in CRL. They do so by penalizing changes in the parameters that are important for previous tasks. Although this method is often treated as a single category in many reviews [18], [174], [175], we consider it as a subcategory of policy merging because it usually combines with other methods [109], [127], [143]. Furthermore, many regularization methods in supervised continual learning can be directly applied to CRL. The most common regularization method is EWC [42], which has been successfully applied in CRL [67], [176]. EWC employs the Fisher information matrix to identify and constrain critical parameters, effectively merging knowledge from sequential tasks without significant loss of previously acquired knowledge. This method introduces a regularization term that penalizes changes in the parameters that are important for previous tasks, thereby preserving knowledge while allowing the model to adapt to new tasks. Formally, the EWC is mathematically represented by the loss function:

$$\mathcal{L}_{\text{EWC}} = \sum_i \frac{\lambda}{2} \boldsymbol{F}_i (\theta_i - \theta_i^\star)^2, \tag{25}$$

where $\boldsymbol{F}_i$ is the Fisher information matrix for parameter θ_i, $\theta_i^\star$ is the parameter value after training on the previous task, and λ is the regularization strength. Building on EWC, P&C introduces an online version of EWC to address the computational challenges associated with traditional EWC's linear growth in complexity [74]. The online-EWC method modifies the EWC approach by updating the Fisher information matrix incrementally, allowing for efficient knowledge preservation while enabling the model to gracefully forget older tasks if needed. Instead of recalculating the Fisher information matrix for each task, online-EWC updates it incrementally. The update rule incorporates a decay factor γ, which gradually reduces the influence of older tasks:

$$\boldsymbol{F}_i^{(t)} = \gamma \boldsymbol{F}_i^{(t-1)} + \boldsymbol{F}_i^{\text{new}}, \tag{26}$$

where $\boldsymbol{F}_i^{(t)}$ is the Fisher information matrix at time t, $\boldsymbol{F}_i^{(t-1)}$ is the matrix from the previous time step, and $\boldsymbol{F}_i^{\text{new}}$ is the matrix calculated for the current task. Then, the loss function for online-EWC incorporates the updated Fisher information matrix:

$$\mathcal{L}_{\text{Online-EWC}} = \sum_i \frac{\lambda}{2} \boldsymbol{F}_i^{(t)} (\theta_i - \theta_i^*)^2. \tag{27}$$

In terms of broader regularization, Kaplanis *et al.* [56], [78] proposed models that leverage multiple timescales of learning. Their earlier work [56] introduces a synaptic model inspired by biological synapses, which incorporates dynamic variables to mitigate catastrophic forgetting without relying on task boundaries. The model's ability to retain information is encapsulated in the dynamics of the hidden variables, which help to regularize the learning process. Following this, the *Policy Consolidation* (PC) model builds on these ideas by using a cascade of hidden networks to regularize the current policy based on its historical performance, thereby preventing performance degradation in non-stationary environments [78]. In addition to these foundational methods, recent advancements such as *adapTive RegularizAtion in Continual environments* (TRAC) and *Composing Value Functions* (CFV) highlight the evolving landscape of regularization in CRL. TRAC, a parameter-free optimizer, dynamically adjusts regularization to prevent the loss of plasticity while enabling rapid adaptation to new tasks [22]. CFV, on the other hand, focuses on the theoretical underpinnings of value function composition, providing a framework for leveraging entropy-regularization to solve new tasks without further learning [98]. Overall, policy merging in CRL, particularly through regularization methods like EWC and its variants, provides powerful mechanisms for stability. Therefore, native regularization methods are still used as baselines for comparison in many CRL studies [65], [70], [119]. The direct combination of these techniques with transfer learning methods also provides a combinatorial direction for CRL.

C. Experience-focused Methods

Experience-focused methods in CRL aim to enhance the agent's ability to store and reuse experiences effectively. These methods are similar to the experience replay mechanism widely used in DRL, where experiences are stored in a buffer and replayed to stabilize training and break the correlation in data. In CRL, experience-focused methods leverage replay buffers, memory mechanisms, or experience relabeling to maintain a balance between retaining critical information from previous tasks and integrating new knowledge. Experience-focused approaches are particularly valuable in CRL, as they

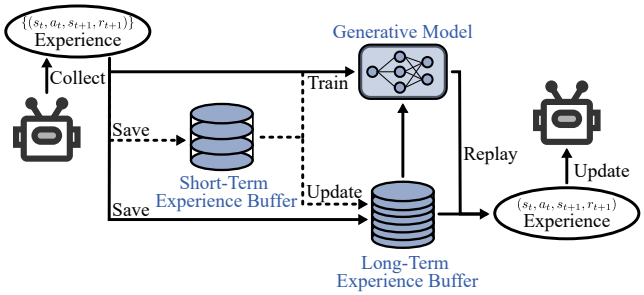

Fig. 9. The framework of experience-focused methods. Some methods use a complementary learning system, including a cross-task long-term experience buffer and a short-term experience buffer for the current task [96], [110], [118], [154]. Typical subcategories include direct replay and generative replay.

provide a direct mechanism for revisiting past knowledge without requiring task-specific information, making them versatile for both task-aware and task-agnostic scenarios. Based on whether the experience is stored or generated, experience-focused methods can be further divided into direct replay and generative replay.

1) Direct Replay: A significant body of work in this category revolves around direct replay, where the agent explicitly stores and reuses experiences from a replay buffer. Selective experience replay [96] is an early effort in this direction, which introduces strategies to prioritize the storage of experiences in a long-term memory buffer based on their importance, such as distribution matching or maximizing state-space coverage. This approach effectively mitigates catastrophic forgetting by ensuring that the agent revisits diverse and representative experiences from past tasks. Similarly, *Continual Learning with Experience And Replay* (CLEAR) [99] combines on-policy learning with off-policy replay, incorporating behavior cloning and V-Trace corrections to align the current policy with past policies. This method not only prevents forgetting but also achieves stability in multi-task settings without requiring task boundaries.

Recent advancements further extend the utility of direct replay methods by integrating them with other techniques. For instance, CoMPS [104] stores high-reward experiences in a replay buffer to support meta-policy search, enabling efficient adaptation to new tasks while leveraging past knowledge. In a similar vein, *Replay-based Recurrent RL* (3RL) [118] employs *Recurrent Neural Networks* (RNNs) to encode historical data, allowing for task-agnostic adaptation without explicit task identifiers. Moreover, relabeling, weighting, and task inference are also integrated into direct replay methods to improve sample efficiency and have been successfully applied in robotics and natural language processing [122], [152], [177].

While direct replay methods have demonstrated significant success, they often rely on the explicit storage of experiences, which can pose challenges in terms of memory constraints, scalability, and privacy. Future research could explore more efficient memory management strategies, such as dynamic memory allocation or selective forgetting, to address these limitations. Additionally, these limitations provide opportunities for an alternative approach to experience-focused methods: generative replay.

2) Generative Replay: Instead of explicitly storing experiences, generative replay methods leverage generative models to recreate or simulate previous experiences, enabling the agent to revisit and learn from past knowledge without requiring a large memory footprint. This makes generative replay particularly suited for scenarios with limited memory resources or strict privacy constraints. By synthesizing experiences on demand, these methods provide a flexible and efficient mechanism for continual learning.

A common thread among generative replay methods is their reliance on powerful generative models, such as *Variational Auto-Encoders* (VAEs) or *Generative Adversarial Networks* (GANs), to produce realistic and representative samples of past experiences. For instance, *Reinforcement-Pseudo-Rehearsal* (RePR) [110] employs a GAN-based pseudo-rehearsal mechanism to generate representative states from previous tasks, which are then rehearsed alongside new task data. This dual memory system ensures effective knowledge retention without requiring explicit storage of raw experiences, demonstrating strong performance in Atari 2600 games. Similarly, *Self-generated Long-term Experience Replay* (SLER) [154] introduces a dual memory architecture, combining short-term experience replay with the *Experience Replay Model* (ERM) to generate simulated experiences of past tasks. By retaining only essential information from prior tasks (e.g., initial states, actions, and rewards), SLER achieves significant memory efficiency while maintaining performance across complex environments like StarCraft II.

Other generative replay methods extend these ideas with unique mechanisms for triggering and managing generative processes. For example, *Self-TRIGgered GEnerative Replay* (S-TRIGGER) [155] utilizes self-triggered generative replay, where a VAE generates samples from prior environments upon detecting environmental changes using statistical tests. This adaptive mechanism enables the model to handle dynamic task transitions without manual intervention, making it particularly effective in environments with evolving distributions. Meanwhile, a model-free generative replay approach [86] incorporates a wake-sleep cycle for memory consolidation, alternating between learning from current tasks and replaying experiences generated by a VAE. This study also pointed out that hidden replay (replay for hidden features) is the most promising approach that pushed the state-of-the-art in generative replay in CRL.

Across these methods, several trends and advancements can be observed. First, the use of generative models to synthesize experiences has proven to be a versatile solution for balancing memory efficiency with the need to revisit past knowledge. Second, integrating generative replay with mechanisms like automatic task change detection or dual memory architectures has enhanced the adaptability and scalability of these approaches. However, challenges remain, particularly in ensuring the fidelity and diversity of generated experiences. The reliance on generative models introduces potential vulnerabilities, such as feature drift or inaccuracies in the generated data, which can impact long-term learning stability [86], [154], [155]. Future research could focus on improving the robustness and generalization capabilities of generative models, as well as

exploring hybrid approaches that combine generative replay with selective storage of critical experiences.

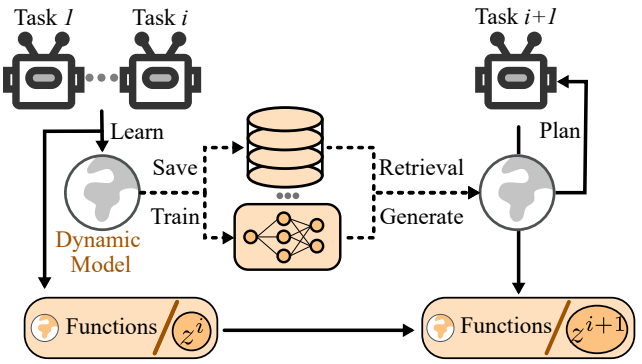

Fig. 10. The framework of dynamic-focused methods. Direct modeling (above) and indirect modeling (below) are two main categories.

D. Dynamic-focused Methods

Dynamic-focused methods in CRL are closely related to *Model-Based Reinforcement Learning* (MBRL), where the core idea is to learn a model of the environment's dynamics to predict future states and rewards. In CRL, dynamic-focused methods extend this concept to tackle non-stationary environments. They enhance the agent's ability to adapt to changing environments and tasks by modeling the environment's dynamics ($T(s'|s,a)$). As shown in Fig. 10, dynamic-focused methods can be divided into two categories: direct modeling and indirect modeling.

1) Direct Modeling: Direct modeling explicitly learns the dynamics of the environment (e.g., the transition function) based on observed state-action pairs. These methods aim to capture the underlying structure of the environment, allowing the agent to predict future states and adapt its behavior accordingly. By maintaining an explicit model of the environment, direct modeling approaches are well-suited for tasks requiring long-term planning and reasoning in changing conditions.

A common theme among direct modeling methods is the use of mixture models and probabilistic frameworks to address the challenges of catastrophic forgetting and task adaptation. Some approaches, including *Meta-learning for Online Learning* (MOLe) [156] and *LifeLong Incremental Reinforcement Learning* (LLIRL) [157], leverage the *Chinese Restaurant Process* (CRP) to dynamically instantiate new models as the environment changes. These methods maintain a library of dynamics models $\{d_{\eta_t}^{(l)}\}_{l=1}^{L}$, where each model $f_{\eta_t}^{(l)}$ represents a specific task or environment configuration. The assignment of a new observation (s_t, a_t, s_{t+1}) to one of the models is governed by a probabilistic prior. For example, the CRP prior can be expressed as:

$$P(x_t = l) = \begin{cases} \frac{n^{(l)}}{t-1+\zeta}, & \text{if } l \leq L, \\ \frac{\zeta}{t-1+\zeta}, & \text{if } l = L+1, \end{cases} \tag{28}$$

where x is the assignment of the current observation to a model, $n^{(l)}$ is the number of observations assigned to model l, and ζ is a concentration parameter controlling the likelihood of creating a new model. By employing a probabilistic mixture, such as an infinite mixture of Gaussian processes [134], they can efficiently manage task shifts by either reusing existing models for familiar dynamics or creating new ones for unseen transitions. This approach ensures that the agent retains knowledge of past dynamics while remaining flexible to adapt to new ones.

Recent advancements have extended the direct modeling paradigm to improve scalability and efficiency. For example, *Continual Reinforcement Learning via Hypernetworks* (HyperCRL) [113] employs task-conditional hypernetworks to generate task-specific dynamics models without increasing model capacity. Formally, the input of the hyper network $\mathcal{H}_{\Theta_k}$ is a task embedding e_k and the output is the parameters of the dynamics model η_k:

$$\hat{s}_{t+1} = f_{\eta_k}(s_t, a_t), \eta_k = \mathcal{H}_{\Theta_k}(e_t), \tag{29}$$

where $\mathcal{H}_{\Theta_k}$ is the hyper network parameterized by Θ_k. By using a fixed-capacity hypernetwork, HyperCRL avoids the linear growth in model complexity typically associated with maintaining multiple models, while a regularization term ensures that the hypernetwork retains predictive accuracy across tasks. Similarly, Losse-FTL [121] introduces locality-sensitive sparse encoding to enhance the efficiency of direct modeling. By using sparse nonlinear features and a *Follow-The-Leader* (FTL) objective, Losse-FTL achieves incremental updates without catastrophic forgetting, making it particularly suitable for high-dimensional environments.

2) Indirect Modeling: Indirect modeling methods do not directly model the environment's dynamics but instead use alternative representations or abstractions (e.g., latent variables) to infer or adapt to the dynamics. They allow the agent to generalize across tasks without requiring a detailed model of the environment's transitions.

One prominent example of indirect modeling is the *LIfelong Latent Actor-Critic* (LILAC) algorithm introduced [101], which leverages latent variable models to represent the evolving dynamics of non-stationary environments. LILAC formalizes a *Dynamic Parameter Markov Decision Process* (DP-MDP) framework, where latent variable z captures the task-specific parameters that shift stochastically between episodes. The agent learns to maximize the expected returns while maintaining a compact latent representation of the environment dynamics. The optimization objective can be expressed as:

$$\begin{aligned} &P(o_{1:H} = 1|\tau^{1:i-1}) \geq \\ &\mathbb{E}_{P(z^i|\tau^{1:i-1})}\left[\sum_{t=1}^{T} R(s_t, a_t; z^i) - \log \pi(a_t|s_t, z^i)\right], \end{aligned} \tag{30}$$

where $\tau^{1:i-1}$ represents the agent's past trajectory, and z^i is the latent embedding for the current task. This allows LILAC to jointly optimize for both high returns and policy entropy, ensuring adaptability to non-stationary dynamics while mitigating catastrophic forgetting. Similarly, 3RL [118] and Continual-Dreamer [119] are task-agnostic approaches that utilize latent representations to address continual learning challenges. 3RL employs recurrent memory and experience replay to maintain a contextual understanding of past tasks,

while Continual-Dreamer integrates world models with reservoir sampling to selectively replay experiences, enhancing task adaptation and reducing forgetting. These approaches share the commonality of leveraging abstract representations (recurrent structures or probabilistic models) to infer environment dynamics and enable robust learning in non-stationarity.

A key trend among indirect modeling methods is the use of intrinsic rewards to guide exploration and adaptation. For instance, both *Lifelong Skill Planning* (LiSP) [85] and Continual-Dreamer incorporate intrinsic rewards based on prediction uncertainty to encourage exploration in uncertain state-action spaces. This aligns with the broader goal of indirect modeling methods to prioritize adaptability and generalization over precise environmental modeling. Moreover, these methods often combine latent representations with auxiliary techniques, such as variational inference [101] or ensemble models [119], to enhance their robustness and scalability. However, challenges remain, particularly in handling unobserved or continuous changes in environment dynamics [101]. Future research could focus on relaxing episodic assumptions and exploring more flexible frameworks for real-time adaptation.

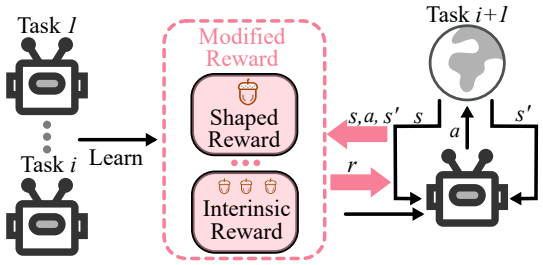

Fig. 11. The framework of reward-focused methods.

E. Reward-focused Methods

Reward-focused methods in CRL concentrate on managing and leveraging reward signals to facilitate efficient learning and adaptation to new tasks, which is similar to reward shaping in transfer reinforcement learning. These methods are particularly significant in CRL because rewards directly influence the agent's policy optimization and learning trajectory. By restructuring or reshaping reward distributions, reward-focused approaches address key challenges such as sparse or delayed rewards, knowledge transfer across tasks, and maintaining consistent learning performance over time. In general, these methods modify the reward function r_t by incorporating shaping functions or intrinsic components, which can be expressed as:

$$R_t^{\mathrm{M}} = R_t + h(s, a, s') + \alpha R_t^{\mathrm{I}}, \tag{31}$$

where $h(s, a, s')$ is a shaping function derived from external knowledge or task-specific information, and R_t^{I} is an intrinsic reward component that encourages exploration or other desirable behaviors. The weighting factor α balances the contribution of intrinsic rewards to the overall reward signal.

Several approaches exemplify the potential of reward-focused methods in CRL. For instance, the *Shaping Rewards for LifeLong RL* (SR-LLRL) algorithm [160] introduces the *Lifetime Reward Shaping* (LRS) function, which reshapes rewards based on cumulative visit counts from previous tasks. The LRS function in task i can be expressed as:

$$h_i(s, a, s') = (1 - \gamma) V_{\max} \frac{c_{i-1}(s, a, s')}{c_{i-1}(s)}, \tag{32}$$

where $c_{i-1}(s, a, s')$ and $c_{i-1}(s)$ are the cumulative visit counts of state-action pairs and states, respectively, from optimal trajectories in prior tasks. This approach accelerates learning by providing more informative rewards and addressing the inefficiencies caused by sparse or delayed rewards. Similarly, Jiang *et al.* [111] employed temporal-logic-based reward shaping, where the shaping function $h(s, a, s')$ is derived from a potential function Φ and the optimal Q-function $Q^\star$:

$$h(s, a, s') = \Phi\left(s', \arg\max_{a'} Q^\star(s', a')\right) - \Phi(s, a). \tag{33}$$

This approach leverages domain knowledge encoded in temporal logic to guide the agent's exploration and learning, even in the presence of imperfect advice.

Intrinsic rewards also play a critical role in reward-focused CRL methods. They are often used to encourage exploration, curiosity, or other desirable behaviors, and can be combined with extrinsic rewards to guide the agent's learning process. Formally, intrinsic rewards can be expressed as:

$$R_t = R_t^{\mathrm{E}} + \alpha R_t^{\mathrm{I}}, \tag{34}$$

where R_t^{E} is the extrinsic reward, R_t^{I} is the intrinsic reward, and α is a weighting factor. For example, *Intrinsically Motivated Lifelong exploration* (IML) [159] proposes a lifelong exploration mechanism that combines short-term and long-term intrinsic rewards, expressed as:

$$R_t^{\mathrm{I}} = \max(R_t^{\mathrm{D}}, 1.0) \cdot R_t^{\mathrm{L}}, \tag{35}$$

where R_t^{L} and R_t^{D} are the local and deep exploration bonuses computed by the corresponding component, respectively. The former promotes the visit of surprising states, encouraging local exploration, while the latter provides a long-term exploration signal. Similarly, Steinparz *et al.* [161] introduced *Reactive Exploration*, which uses prediction errors from observation and reward models to generate intrinsic rewards:

$$R_{t+1} = \alpha R_{t+1}^{\mathrm{S}} + \beta R_{t+1}^{\mathrm{R}} + \lambda R_{t+1}^{\mathrm{E}}, \tag{36}$$

where R_{t+1}^{S} and R_{t+1}^{R} are intrinsic rewards derived from the observation and reward models, respectively, and $\lambda = 1-\alpha-\beta$ balances the contributions of these components. This method enables agents to adapt dynamically to non-stationary environments by encouraging exploration in regions of the state space that have undergone significant changes. Additionally, *Efficient Lifelong Imitation Reinforcement Learning* (ELIRL) expands the reward-focused methods to inverse reinforcement learning, which can be used to learn the reward function from expert demonstrations [158]. Similarly to the factor decomposition method, it uses a shared latent reward component to rebuild the reward function of each task to facilitate knowledge transfer.

With the recent advancements in large language models, some DRL methods have also explored the LLM-driven reward design [178], [179]. Based on these, *Multi-granularity*

knowledge Transfer for Continual reinforcement learning (MTCore) [125] proposes utilizing the reasoning ability of the large language model for task planning, and transferring coarse-grained knowledge through intrinsic rewards. This multi-granularity knowledge transfer framework, composed of the coarse-grained knowledge (language) and the fine-grained knowledge (RL policies), extends the ability of CRL agents to transfer across more diverse tasks.

A commonality among these methods is their focus on addressing the limitations of traditional reward structures in RL. By reshaping rewards or introducing intrinsic components, they mitigate the challenges posed by sparse or delayed feedback, enhance knowledge transfer across tasks, and improve exploration efficiency. However, challenges remain in scaling these approaches to high-dimensional or continuous state-action spaces [159], [160]. Additionally, while methods like temporal-logic-based reward shaping [111] and reward machines [84] offer structured ways to incorporate domain knowledge, their reliance on predefined specifications may limit their applicability in fully autonomous settings. Future research could focus on developing more flexible and automated mechanisms for reward shaping, as well as integrating reward-focused methods with other CRL paradigms, such as policy-focused or experience-focused approaches.

F. Beyond Traditional CRL

There are several emerging research directions in CRL that extend beyond the traditional methods discussed above. These methods encompass novel techniques and pose unique challenges in CRL, including out-of-distribution detection, imitation learning, and multi-agent coordination. This section provides an overview of them and their potential impact on the field of CRL.

1) Task Detection: As described in Section III-F, the task-agnostic scenario is a common setting in CRL, where the agent is not informed of the task identities or boundaries. However, many CRL methods rely on task-specific information to guide learning and adaptation [133], [141]. Task detection methods bridge this gap by enabling agents to identify task identities or detect environmental changes without explicit supervision. These methods are particularly useful for approaches that associate specific structures or policies with individual tasks. Naturally, they can be divided into two categories: task identity detection and environment change detection.

Task identity detection methods aim to identify task labels by leveraging patterns in observations, latent representations, or task-specific features. Therefore, unsupervised or semi-supervised learning techniques can be naturally applied to this problem. For example, Jacobson *et al.* [133] proposed the use of *Familiarity AutoEncoders* (FAEs) for discovering task labels. FAEs reconstruct input data for specific tasks, assigning task labels based on the autoencoder with the highest reconstruction performance. This method avoids catastrophic forgetting of the task detector by maintaining separate models for each task. Additionally, the exploration of variational and adversarial autoencoders as FAE variants highlights the adaptability of the approach to noisy environments. Instead of using explicit task labels, *Behavior-Guided Policy Optimization* (BGPO) [177] uses behavior embeddings as task identities. It employs a self-organizing network to incrementally learn a behavior embedding space from demonstrations. By matching new behaviors to the nearest embedding, the system efficiently infers tasks without requiring predefined task structures. This approach is further enhanced by a behavior-matching intrinsic reward, which aligns generated trajectories with demonstrated behaviors. The dynamic expansion of the embedding space ensures scalability to novel tasks, making this method suited for continual robot learning. Additionally, OWL formulates task identity detection as a multi-armed bandit problem, allowing for adaptive policy selection during test time. By combining a multi-head network with shared representations, this method effectively mitigates interference between tasks.

In contrast, environment change detection methods focus on detecting environment dynamics rather than explicit task identities. Environment changes in RL can be categorized as changes in the input distribution, changes in the transition function, or changes in the reward function. Methods to detect changes in input distributions have been developed in the field of novelty and out-of-distribution detection with applications in CL [180], [181]. A commonality is the use of statistical and probabilistic techniques to detect changes in the environment. For instance, S-TRIGGER [155] employs a self-triggered generative replay mechanism, utilizing statistical analysis of reconstruction errors from VAEs to detect significant environmental changes. Similarly, LLIRL [157] leverages an infinite mixture model with online Bayesian inference to adapt to dynamic environments. By using the Chinese restaurant process for environment clustering, LLIRL can detect changes without external signals, showcasing the importance of probabilistic frameworks in managing environment dynamics.

Recently, another statistical method, *Sliced Wasserstein Online Kolmogorov-Smirnov* (SWOKS) [141], combines optimal transport methods with the Sliced Wasserstein distance and the Kolmogorov-Smirnov test to measure distances between experience distributions. This method's ability to operate online without predefined task labels highlights its suitability for real-time adaptation in complex environments. The use of distance metrics to detect task changes aligns with the statistical approaches of S-TRIGGER and LLIRL, yet SWOKS uniquely incorporates optimal transport methods to enhance detection accuracy. In contrast to these statistical approaches, *Reactive Exploration* [161] focuses on modifying reward structures to include intrinsic rewards based on prediction errors. This method utilizes the *Intrinsic Curiosity Module* (ICM) to detect changes and encourage exploration in altered regions of the state space. This reward-focused strategy provides an alternative perspective, emphasizing the role of intrinsic motivation in detecting environmental changes.

2) Offline Reinforcement Learning and Imitation Learning in CRL: Offline Reinforcement Learning (Offline RL) and *Imitation Learning* (IL) represent compelling extensions to CRL frameworks, particularly for leveraging static datasets or expert demonstrations to address the challenges of lifelong learning. Offline RL focuses on learning policies from pre-collected datasets without requiring direct interaction with the environment, which is advantageous in scenarios where real-

world interactions are costly or infeasible. Currently, there is still very little research on the integration of offline RL with CRL, but some initial studies have shown promising results. LiSP is an early step in this direction, which uses offline data to discover skills in reset-free lifelong RL, enabling long-horizon planning in an abstract skill space [85]. Its effectiveness is demonstrated in a variety of settings, including offline interactions.

A recent work, *Offline Experience Replay* (OER), formulates the *Continual Offline Reinforcement Learning* (CORL), where an agent learns a sequence of offline reinforcement learning tasks and pursues good performance on all learned tasks with a small replay buffer without exploring any of the environments of all the sequential tasks [182]. OER addresses the distribution shift problem in CORL by introducing a *Model-Based Experience Selection* (MBES) scheme. This approach filters offline data to build a replay buffer that closely aligns with the learned model, mitigating catastrophic forgetting while maintaining performance on new tasks. Additionally, offline RL has been applied in some CRL methods [105], [151], highlighting the potential of this technique to enhance CRL by integrating prior knowledge and reducing the reliance on extensive real-world interactions.

Imitation learning, on the other hand, provides a complementary approach to RL by utilizing expert demonstrations to guide the agent's learning. ELIRL [158] and *Fast Lifelong Adaptive Inverse Reinforcement learning* (FLAIR) [183] exemplify how IL can be adapted to CRL settings. ELIRL introduces a shared latent reward structure that facilitates knowledge transfer across sequential tasks while addressing catastrophic forgetting. FLAIR builds upon this by incorporating policy mixtures for fast adaptation to heterogeneous demonstrations, ensuring scalability and personalization in lifelong learning scenarios. Both approaches emphasize the importance of efficient knowledge-sharing and task-specific adaptation. In addition, behavioral cloning, as a simple imitation learning method, has been applied in many CRL methods with experience replay [61], [99], [184].

G. Applications

Although the research on CRL is shown to be promising, the application of CRL in real-world scenarios is still in its infancy. In this section, we summarize recent applications that are closely related to CRL, including robotics, autonomous driving, and game playing.

1) Robotics Learning: Robotics is a prominent application domain of CRL, where lifelong robots are required to adapt to new tasks or environments without forgetting previously learned tasks. For example, robots must continuously learn new skills as they encounter various tasks, appliances, and user preferences in home environments [90]. Traditional robot learning methods typically rely on a large amount of independent and identically distributed data, which is impractical in dynamic settings. Recent studies have focused on addressing this challenge by using offline reinforcement learning, skill learning, and distillation that enable robots to learn efficiently from limited demonstrations and adapt to new tasks while mitigating catastrophic forgetting [90], [150]–[152]. Fine-tuning pre-trained policies is also an effective strategy for continual learning in robotics. By adapting policies to new variations with minimal offline data, robots can improve their performance in dynamic environments [185].

The navigation task of mobile robots has been widely studied in CRL due to its extensive application and relative simplicity. CRL has been applied to enable mobile robots to learn sequentially in unknown environments through techniques such as policy distillation, task decomposition, and behavior self-organization [147], [176], [177]. Additionally, *Lifelong Federated Reinforcement Learning* (LFRL) leverages cloud-based systems to enhance navigation capabilities by fusing experiences from multiple robots, thereby improving generalization across different environments [87].

Recent developments have also introduced benchmarks specifically designed to evaluate the performance of CRL methods in robotic tasks. Continual World provides a structured sequence of robotic manipulation tasks that emphasize forward transfer, challenging existing algorithms to balance forgetting and transfer [60]. Furthermore, CORA provides a more comprehensive benchmark, in which sequences based on household robot tasks test agents in a more realistic visual domain and evaluate their sample efficiency [68]. Despite these advancements, some researchers have pointed out that they are too challenging for current CRL agents, and have proposed simpler benchmarks to facilitate the development of more effective methods [186], [187].

2) Game Playing: The game is a common testbed for RL algorithms. It has evolved from classical benchmarks such as GridWorld games to more complex settings such as video games with multimodal inputs [188]. These environments provide a controlled yet challenging setting for assessing the continual learning capabilities of CRL agents. Simple video games, such as Procgen [189] and Atari [81], have been widely utilized as benchmarks for evaluating CRL methods [22], [53], [74], [110]. These games offer a diverse range of tasks (different environments and play modes) that test an agent's ability to generalize and retain knowledge, making them ideal for studying the effects of catastrophic forgetting and the efficacy of knowledge transfer mechanisms. Based on these games, CORA introduces controlled variations and benchmarks that further challenge CRL agents, emphasizing the need for robust generalization and sample-efficient learning [68]. They provide structured sequences of tasks that highlight the strengths and limitations of current CRL methods, revealing that while some algorithms excel in preserving knowledge, they often struggle with adapting to new, visually complex tasks. Furthermore, HackAtari introduces controlled novelty into traditional game environments [190]. By modifying game dynamics and reward structures, it facilitates the evaluation of agents' ability to generalize and adapt to new conditions.

As CRL research progresses, more complex games such as online strategy games or 3D video games have become prominent testbeds for advanced CRL methods. These games present a higher level of complexity due to their high-dimensional state spaces and more unstable dynamic environments. In Minecraft, hierarchical approaches have been proposed to

efficiently transfer and reuse skills across tasks, addressing the sample efficiency challenge posed by the game's vast and varied environment [85], [94]. In StarCraft 2, model-free generative replay frameworks and wake-sleep mechanisms have been developed to improve continual learning efficiency [86], [191]. Additionally, COOM provides an image-based CRL benchmark based on ViZDoom for evaluating agents with embodied perception [70]. These frameworks leverage advanced modeling techniques to maintain performance across complex tasks, demonstrating significant improvements in both forward transfer and retention of previously acquired skills.

3) Others: Recently, CRL has been explored in various other fields beyond the above, showcasing its versatility and potential for broad application. In **natural language processing**, CRL has been applied to dialogue systems by integrating a transformer, enabling them to integrate new knowledge dynamically to adapt to new topics and tasks without forgetting [192]. Similarly, in controlled text generation, CRL frameworks have been utilized to allow large language models to adaptively generate text that aligns with specified attributes, such as topic or sentiment, in real-time [193]. Additionally, *Continual Proximal Policy Optimization* (CPPO) has been proposed to enhance *Reinforcement Learning from Human Feedback* (RLHF) [194] by balancing policy learning and knowledge retention, allowing language models to advise human preferences without extensive retraining [122].

In the field of **medical imaging**, CRL addresses the challenge of catastrophic forgetting by employing selective experience replay with corset compression [195]. In **finance**, CRL has been utilized for continual portfolio selection, allowing trading agents to adapt to dynamic market conditions by incrementally updating their strategies based on new data, thereby improving returns and reducing risks [196]. In the domain of **autonomous driving**, CRL has been employed to improve the adaptability of self-driving cars in partially observable environments [197]. Additionally, CRL has been applied in the field of resource allocation within **industrial internet of things networks** [198]. Here, the CRL method enables efficient resource management by continuously learning and adapting to the dynamic network conditions, thereby optimizing data transmission and energy consumption. Furthermore, CRL has been applied to **data center cooling control**, where it enhances energy efficiency by enabling systems to adapt quickly and safely to changing thermal conditions [199].

V. FUTURE WORKS

In this section, we present some open challenges and future directions in CRL, based on both retrospectives of the discussed methods and outlooks to the emerging trends of AI.

A. Task-free CRL

Most CRL methods assume that tasks are given in advance, the boundaries between tasks are clear, and the environment is stationary in a task. However, the environment may continuously change over time. Moreover, in real-world scenarios, agents are expected to learn in a non-stationary environment. Task-free CRL is a challenging problem that requires agents to learn from the environment without any explicit tasks. It is closely related to online learning and has been preliminarily explored in some works on task boundary detection and task-agnostic CRL. We believe this direction is the necessary path for RL to move towards general AI.

B. Evaluation and Benchmark

Variant evaluation metrics have been proposed to measure CRL from different but complementary perspectives, although no single metric can summarize the efficacy of a CRL approach. In addition, most metrics are drawn from the supervised CL field, which may not be suitable for CRL. Designing a set of generalized, novel metrics is beneficial for the development of CRL. Moreover, with the effervescent development of large-scale models, it is crucial to standardize evaluation from the perspectives of scalability and privacy. The appropriateness of the stored/transferred knowledge and the security of the model should also be quantified as metrics.

C. Interpretable Knowledge

Ranging from exterior experiences to policy function approximators, black-box knowledge is more accessible and predominant than interpretable and well-articulated knowledge. However, the latter is beneficial for evaluating and explaining the process of CRL. Moreover, building an interpretable knowledge base can help to transfer knowledge across various tasks and even alleviate catastrophic decision-making for high-stakes tasks such as autonomous driving.

D. Large-scale Pre-trained Model

Recently, unprecedented breakthroughs have been achieved in learning Large-scale *Pre-Trained Models* (PTMs) built on massive computation resources and attributed data. One representative example is *Large Language Models* (LLMs). Considering them as a complete knowledge base whose training is a black-box process, there are still many challenges in this direction, which are of concern to a larger CL community, including the CRL field. We briefly point out two directions that are worth exploring in the future:

1) CRL for PTMs: One important method for aligning PTMs with human preferences is RLHF [194]. CPPO [122] has been proposed to enhance RLHF with continual learning, allowing LLMs to adapt to human preferences continuously without extensive retraining. Recently, the methods represented by Deepseek-R1 have been proposed to fine-tune LLMs with RL to achieve better reasoning ability [7], [200], [201]. Although CL ability is not explicitly mentioned in these works, we believe it is crucial for PTMs' further development. We also anticipate other forms of CRL methods to be explored to further enhance the performance of PTMs.
2) PTMs for CRL: PTMs can be used as a knowledge base for CRL, which can be transferred to the target task to improve the sample efficiency and generalization ability. Existing works have shown that PTMs can be used to improve the performance of CL and RL in some

specific scenarios [202]–[204]. MT-Core first integrates an LLM into the CRL paradigm, enabling agents to multi-granularity knowledge transfer across diverse tasks [125]. We also expect more research to be conducted in this direction to explore the potential of PTMs in CRL.

E. Embodied Agent

Embodied agents are agents that interact with the environment through sensors and actuators, such as virtual agents and robots [205], [206]. The development of embodied agents with CL capabilities has gained increasing attention, particularly with the advent of LLMs [207]. Recent works have demonstrated how LLMs can enhance embodied agents by enabling efficient skill acquisition, interpretable knowledge storage, and adaptive reuse of learned behaviors [123], [208]. Such capabilities align closely with the goals of CRL, particularly in addressing catastrophic forgetting and improving generalization across tasks. Additionally, leveraging multimodal models, such as vision-language frameworks, has shown promise in improving sample efficiency and facilitating autonomous exploration in sparse reward environments [124]. While some continual embodied agents often rely on imitation learning to acquire new behaviors and adapt to novel [208]–[210], recent advancements have begun to explore the integration of CRL into this domain [123]. This shift is motivated by the need for agents to autonomously learn from dynamic, interactive environments and operate without predefined task boundaries. Despite the application of CRL to embodied agents being relatively nascent, the embodied agents offer a compelling direction for CRL research, serving as a platform to study lifelong learning in complex, real-world scenarios.

VI. CONCLUSION

In this work, we present an up-to-date and comprehensive survey of continual reinforcement learning, bridging the latest advances in methodology, evaluation, and application. We summarize challenges and future directions in this field, with an extensive analysis of how representative strategies address them from the perspective of knowledge. By sorting out the methodologies for each category, we hope that readers can easily grasp the mainstream methods, identify suitable baselines, and contribute future solutions in light of existing ones. Encouragingly, we observe a growing and widespread interest in CRL from the broad AI community, bringing novel understandings, diversified applications, and cross-directional opportunities. Based on such a holistic perspective, we expect the development of CRL to eventually empower AI agents with human-like adaptability, responding flexibly to real-world dynamics and evolving themselves in a lifelong manner.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[3] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[4] P. Bryant, G. Pozzati, and A. Elofsson, "Improved prediction of protein-protein interactions using AlphaFold2," *Nature Communications*, vol. 13, no. 1, p. 1265, 2022.

[5] J. Bausch, A. W. Senior, F. J. H. Heras, T. Edlich, A. Davies, M. Newman, C. Jones, K. Satzinger, M. Y. Niu, S. Blackwell, G. Holland, D. Kafri, J. Atalaya, C. Gidney, D. Hassabis, S. Boixo, H. Neven, and P. Kohli, "Learning high-accuracy error decoding for quantum processors," *Nature*, vol. 635, no. 8040, pp. 834–840, 2024.

[6] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *NeurIPS*, vol. 35, 2022, pp. 27 730–27 744.

[7] DeepSeek-AI, D. Guo, and D. Y. et al., "DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning," *ArXiv preprint*, vol. abs/2501.12948, 2025.

[8] X. Zhan, H. Xu, Y. Zhang, X. Zhu, H. Yin, and Y. Zheng, "Deepthermal: Combustion optimization for thermal power generating units using offline reinforcement learning," in *AAAI*, vol. 36, no. 4, 2022, pp. 4680–4688.

[9] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de las Casas, C. Donner, L. Fritz, C. Galperti, A. Huber, J. Keeling, M. Tsimpoukelli, J. Kay, A. Merle, J.-M. Moret, S. Noury, F. Pesamosca, D. Pfau, O. Sauter, C. Sommariva, S. Coda, B. Duval, A. Fasoli, P. Kohli, K. Kavukcuoglu, D. Hassabis, and M. Riedmiller, "Magnetic control of tokamak plasmas through deep reinforcement learning," *Nature*, vol. 602, no. 7897, pp. 414–419, 2022.

[10] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, 2023.

[11] O. Vinyals, I. Babuschkin, and W. M. e. a. Czarnecki, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[12] Y. Yu, "Towards sample efficient reinforcement learning," in *IJCAI*, 2018, pp. 5739–5743.

[13] Z. Ding and H. Dong, *Challenges of Reinforcement Learning*. Springer Singapore, 2020.

[14] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, and T. Hester, "Challenges of real-world reinforcement learning: definitions, benchmarks and analysis," *Machine Learning*, vol. 110, no. 9, pp. 2419–2468, 2021.

[15] D. Kudithipudi, M. Aguilar-Simon, and J. e. a. Babb, "Biological underpinnings for lifelong learning machines," *Nature Machine Intelligence*, vol. 4, no. 3, pp. 196–210, 2022.

[16] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[17] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2022.

[18] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5362–5383, 2024.

[19] X. Yang, H. Yu, X. Gao, H. Wang, J. Zhang, and T. Li, "Federated continual learning via knowledge fusion: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 8, pp. 3832–3850, 2024.

[20] M. B. Ring, "CHILD: A first step towards continual learning," *Machine Learning*, vol. 28, no. 1, pp. 77–104, 1997.

[21] K. Khetarpal, M. Riemer, I. Rish, and D. Precup, "Towards continual reinforcement learning: A review and perspectives," *Journal of Artificial Intelligence Research*, vol. 75, pp. 1401–1476, 2022.

[22] A. Muppidi, Z. Zhang, and H. Yang, "Fast TRAC: A parameter-free optimizer for lifelong reinforcement learning," in *NeurIPS*, 2024.

[23] Z. Wang, E. Yang, L. Shen, and H. Huang, "A comprehensive survey of forgetting in deep learning beyond continual learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 3, pp. 1464–1483, 2025.

[24] M. L. Puterman, "Markov decision processes," in *Stochastic Models*, 1990, vol. 2, pp. 331–434.

[25] R. J. Boucherie and N. M. Van Dijk, *Markov Decision Processes in Practice*. Springer, 2017.

[26] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *ICLR*, 2016, pp. 1–21.

[27] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," in *ICML*, vol. 48, 2016, pp. 1995–2003.

[28] M. J. Hausknecht and P. Stone, "Deep recurrent Q-Learning for partially observable mdps," in *AAAI Fall Symposia*, 2015, pp. 29–37.

[29] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *ICML*, vol. 48, 2016, pp. 1928–1937.

[30] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *ICLR*, 2016, pp. 1–14.

[31] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *ICML*, vol. 80, 2018, pp. 1582–1591.

[32] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz, "Trust region policy optimization," in *ICML*, vol. 37, 2015, pp. 1889–1897.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *ArXiv preprint*, vol. abs/1707.06347, 2017.

[34] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *ICML*, vol. 80, 2018, pp. 1856–1865.

[35] M. Mundt, Y. Hong, I. Pliushch, and V. Ramesh, "A wholistic view of continual learning with deep neural networks: Forgotten lessons and the bridge to active and open world learning," *Neural Networks*, vol. 160, pp. 306–336, 2023.

[36] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Information Fusion*, vol. 58, pp. 52–68, 2020.

[37] F. Ye and A. G. Bors, "Continual variational autoencoder via continual generative knowledge distillation," in *AAAI*, vol. 37, no. 9, 2023, pp. 10 918–10 926.

[38] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *CVPR*, 2017, pp. 5533–5542.

[39] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *NeurIPS*, vol. 30, 2017, pp. 2990–2999.

[40] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. J. Kuo, "Class-incremental learning via deep model consolidation," in *WACV*, 2020.

[41] A. Hassanpour, M. Moradikia, B. Yang, A. Abdelhadi, C. Busch, and J. Fierrez, "Differential privacy preservation in robust continual learning," *IEEE Access*, vol. 10, pp. 24 273–24 287, 2022.

[42] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[43] F. Mao, W. Weng, M. Pratama, and E. Y. K. Yee, "Continual learning via inter-task synaptic mapping," *Knowledge-Based Systems*, vol. 222, p. 106947, 2021.

[44] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2018.

[45] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *CVPR*, 2022, pp. 16 050–16 059.

[46] A. Mallya and S. Lazebnik, "PackNet: Adding multiple tasks to a single network by iterative pruning," in *CVPR*, 2018, pp. 7765–7773.

[47] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *ECCV*, 2018.

[48] F. Ye and A. G. Bors, "Lifelong generative modelling using dynamic expansion graph model," in *AAAI*, vol. 36, no. 8, 2022, pp. 8857–8865.

[49] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, and Y. Xu, "Few-shot incremental learning with continually evolved classifiers," in *CVPR*, 2021, pp. 12 455–12 464.

[50] Y.-C. Hsu, Y.-C. Liu, and Z. Kira, "Re-evaluating continual learning scenarios: A categorization and case for strong baselines," *ArXiv preprint*, vol. abs/1810.12488, 2018.

[51] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1185–1197, 2022.

[52] D. Abel, A. Barreto, B. V. Roy, D. Precup, H. P. van Hasselt, and S. Singh, "A definition of continual reinforcement learning," in *NeurIPS*, vol. 36, 2023, pp. 50 377–50 407.

[53] Z. Abbas, R. Zhao, J. Modayil, A. White, and M. C. Machado, "Loss of plasticity in continual deep reinforcement learning," in *CoLLAs*, vol. 232, 2023, pp. 620–636.

[54] N. Vithayathil Varghese and Q. H. Mahmoud, "A survey of multi-task deep reinforcement learning," *Electronics*, vol. 9, no. 9, 2020.

[55] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 13 344–13 362, 2023.

[56] C. Kaplanis, M. Shanahan, and C. Clopath, "Continual reinforcement learning with complex synapses," in *ICML*, vol. 80, 2018, pp. 2502–2511.

[57] S. Dohare, J. F. Hernandez-Garcia, Q. Lan, P. Rahman, A. R. Mahmood, and R. S. Sutton, "Loss of plasticity in deep continual learning," *Nature*, vol. 632, no. 8026, pp. 768–774, 2024.

[58] T. Klein, L. Miklautz, K. Sidak, C. Plant, and S. Tschiatschek, "Plasticity loss in deep reinforcement learning: A survey," *ArXiv preprint*, vol. abs/2411.04832, 2024.

[59] A. Juliani and J. T. Ash, "A study of plasticity loss in on-policy deep reinforcement learning," in *NeurIPS*, vol. 37, 2024, pp. 113 884–113 910.

[60] M. Wolczyk, M. Zajac, R. Pascanu, L. Kucinski, and P. Milos, "Continual world: A robotic benchmark for continual reinforcementlearning," in *NeurIPS*, vol. 34, 2021, pp. 28 496–28 510.

[61] ——, "Disentangling transfer in continual reinforcement learning," in *NeurIPS*, vol. 35, 2022, pp. 6304–6317.

[62] M. Malagon, J. Ceberio, and J. A. Lozano, "Self-composing policies for scalable continual reinforcement learning," in *ICML*, vol. 235, 2024, pp. 34 432–34 460.

[63] H. Nekoei, A. Badrinaaraayanan, A. C. Courville, and S. Chandar, "Continuous coordination as a realistic scenario for lifelong learning," in *ICML*, vol. 139, 2021, pp. 8016–8024.

[64] E. C. Johnson, E. Q. Nguyen, B. Schreurs, C. S. Ewulum, C. Ashcraft, N. M. Fendley, M. M. Baker, A. New, and G. K. Vallabha, "L2Explorer: A lifelong reinforcement learning assessment environment," *ArXiv preprint*, vol. abs/2203.07454, 2022.

[65] J. Gaya, T. Doan, L. Caccia, L. Soulier, L. Denoyer, and R. Raileanu, "Building a subspace of policies for scalable continual learning," in *ICLR*, 2023, pp. 1–28.

[66] H. Fu, S. Yu, M. Littman, and G. Konidaris, "Model-based lifelong reinforcement learning with bayesian exploration," in *NeurIPS*, vol. 35, 2022, pp. 32 369–32 382.

[67] V. Lomonaco, K. Desai, E. Culurciello, and D. Maltoni, "Continual reinforcement learning in 3D non-stationary environments," in *CVPR*, 2020.

[68] S. Powers, E. Xing, E. Kolve, R. Mottaghi, and A. Gupta, "CORA: Benchmarks, baselines, and metrics as a platform for continual reinforcement learning agents," in *CoLLAs*, vol. 199, 2022, pp. 705–743.

[69] F. Yang, C. Yang, H. Liu, and F. Sun, "Evaluations of the gap between supervised and reinforcement lifelong learning on robotic manipulation tasks," in *CoRL*, vol. 164, 2022, pp. 547–556.

[70] T. Tomilin, M. Fang, Y. Zhang, and M. Pechenizkiy, "COOM: A game benchmark for continual reinforcement learning," in *NeurIPS*, vol. 36, 2023, pp. 67 794–67 832.

[71] D. Abel, Y. Jinnai, S. Y. Guo, G. D. Konidaris, and M. L. Littman, "Policy and value transfer in lifelong reinforcement learning," in *ICML*, vol. 80, 2018, pp. 20–29.

[72] M. Chevalier-Boisvert, B. Dai, M. Towers, R. Perez-Vicente, L. Willems, S. Lahlou, S. Pal, P. S. Castro, and J. Terry, "Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks," in *NeurIPS*, 2023.

[73] C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik *et al.*, "Deepmind lab," *ArXiv preprint*, vol. abs/1612.03801, 2016.

[74] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, and R. Hadsell, "Progress & compress: A scalable framework for continual learning," in *ICML*, vol. 80, 2018, pp. 4535–4544.

[75] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. D. Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, R. Perez-Vicente,

A. Pierré, S. Schulhoff, J. J. Tai, H. Tan, and O. G. Younis, "Gymnasium: A standard interface for reinforcement learning environments," *ArXiv preprint*, vol. abs/2407.17032, 2024.

[76] S. Kessler, J. Parker-Holder, P. J. Ball, S. Zohren, and S. J. Roberts, "Same state, different task: Continual reinforcement learning without interference," in *AAAI*, vol. 36, no. 7, 2022, pp. 7143–7151.

[77] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *IROS*, 2012, pp. 5026–5033.

[78] C. Kaplanis, M. Shanahan, and C. Clopath, "Policy consolidation for continual reinforcement learning," in *ICML*, vol. 97, 2019, pp. 3242–3251.

[79] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu, "IMPALA: scalable distributed deep-RL with importance weighted actor-learner architectures," in *ICML*, vol. 80, 2018, pp. 1406–1415.

[80] N. Anand and D. Precup, "Prediction and control in continual reinforcement learning," in *NeurIPS*, vol. 36, 2023, pp. 63 779–63 817.

[81] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research*, vol. 47, no. 1, pp. 253–279, 2013.

[82] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *ArXiv preprint*, vol. abs/1606.04671, 2016.

[83] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. P. Agapiou, J. Schrittwieser, J. Quan, S. Gaffney, S. Petersen, K. Simonyan, T. Schaul, H. van Hasselt, D. Silver, T. P. Lillicrap, K. Calderone, P. Keet, A. Brunasso, D. Lawrence, A. Ekermo, J. Repp, and R. Tsing, "StarCraft II: A new challenge for reinforcement learning," *ArXiv preprint*, vol. abs/1708.04782, 2017.

[84] X. Zheng, C. Yu, and M. Zhang, "Lifelong reinforcement learning with temporal logic formulas and reward machines," *Knowledge-Based Systems*, vol. 257, p. 109650, 2022.

[85] K. Lu, A. Grover, P. Abbeel, and I. Mordatch, "Reset-free lifelong learning with skill-space planning," in *ICLR*, 2021, pp. 1–20.

[86] Z. A. Daniels, A. Raghavan, J. Hostetler, and et al., "Model-free generative replay for lifelong reinforcement learning: Application to starcraft-2," in *CoLLAs*, vol. 199, 2022, pp. 1120–1145.

[87] B. Liu, L. Wang, and M. Liu, "Lifelong federated reinforcement learning: A learning architecture for navigation in cloud robotic systems," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4555–4562, 2019.

[88] R. Traoré, H. Caselles-Dupré, T. Lesort, T. Sun, G. Cai, D. Filliat, and N. Díaz-Rodríguez, "Discorl: Continual reinforcement learning via policy distillation," in *NeurIPS DRPL workshop*, 2019.

[89] S. Liu, M. Xu, P. Huang, X. Zhang, Y. Liu, K. Oguchi, and D. Zhao, "Continual vision-based reinforcement learning with group symmetries," in *CoRL*, vol. 229, 2023, pp. 222–240.

[90] S. Powers, A. Gupta, and C. Paxton, "Evaluating continual learning on a home robot," in *CoLLAs*, vol. 232, 2023, pp. 493–512.

[91] N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes, I. Dunning, S. Mourad, H. Larochelle, M. G. Bellemare, and M. Bowling, "The Hanabi challenge: A new frontier for AI research," *Artificial Intelligence*, vol. 280, p. 103216, 2020.

[92] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning," in *CoRL*, vol. 100, 2020, pp. 1094–1100.

[93] D. Isele, M. Rostami, and E. Eaton, "Using task features for zero-shot knowledge transfer in lifelong learning," in *IJCAI*, 2016, pp. 1620–1626.

[94] C. Tessler, S. Givony, T. Zahavy, D. J. Mankowitz, and S. Mannor, "A deep hierarchical approach to lifelong learning in minecraft," in *AAAI*, vol. 31, no. 1, 2017, pp. 1553–1561.

[95] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "Pathnet: Evolution channels gradient descent in super neural networks," *ArXiv preprint*, vol. abs/1701.08734, 2017.

[96] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *AAAI*, vol. 32, no. 1, 2018, pp. 3302–3309.

[97] D. Abel, D. Arumugam, L. Lehnert, and M. L. Littman, "State abstractions for lifelong reinforcement learning," in *ICML*, vol. 80, 2018, pp. 10–19.

[98] B. van Niekerk, S. James, A. C. Earle, and B. Rosman, "Composing value functions in reinforcement learning," in *ICML*, vol. 97, 2019, pp. 6401–6409.

[99] D. Rolnick, A. Ahuja, J. Schwarz, T. P. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *NeurIPS*, vol. 32, 2019, pp. 348–358.

[100] B. Wu, J. K. Gupta, and M. Kochenderfer, "Model primitives for hierarchical lifelong reinforcement learning," in *AAMAS*, vol. 34, no. 1, 2020, pp. 1–38.

[101] A. Xie, J. Harrison, and C. Finn, "Deep reinforcement learning amidst lifelong non-stationarity," in *ICML LML Workshop*, 2020.

[102] G. N. Tasse, S. James, and B. Rosman, "A boolean task algebra for reinforcement learning," in *NeurIPS*, vol. 33, 2020, pp. 9497–9507.

[103] J. A. Mendez, B. Wang, and E. Eaton, "Lifelong policy gradient learning of factored policies for faster training without forgetting," in *NeurIPS*, vol. 33, 2020, pp. 14 398–14 409.

[104] G. Berseth, Z. Zhang, G. Zhang, C. Finn, and S. Levine, "CoMPS: Continual meta policy search," in *ICLR*, 2022, pp. 1–23.

[105] J. A. Mendez, H. van Seijen, and E. Eaton, "Modular lifelong reinforcement learning via neural composition," in *ICLR*, 2022, pp. 1–22.

[106] N. Lucchesi, A. Carta, V. Lomonaco, and D. Bacciu, "Avalanche RL: A continual reinforcement learning library," in *ICIAP*, 2022, p. 524–535.

[107] S. Powers, E. Xing, and A. Gupta, "Self-activating neural ensembles for continual reinforcement learning," in *CoLLAs*, vol. 199, 2022, pp. 683–704.

[108] M. Riemer, S. C. Raparthy, I. Cases, G. Subbaraj, M. P. Touzel, and I. Rish, "Continual learning in environments with polynomial mixing times," in *NeurIPS*, vol. 35, 2022, pp. 21 961–21 973.

[109] P. Schöpf, S. Auddy, J. Hollenstein, and A. Rodriguez-sanchez, "Hypernetwork-PPO for continual reinforcement learning," in *NeurIPS DRL Workshop*, 2022.

[110] C. Atkinson, B. McCane, L. Szymanski, and A. Robins, "Pseudo-rehearsal: Achieving deep reinforcement learning without catastrophic forgetting," *Neurocomputing*, vol. 428, pp. 291–307, 2021.

[111] Y. Jiang, S. Bharadwaj, B. Wu, R. Shah, U. Topcu, and P. Stone, "Temporal-logic-based reward shaping for continuing reinforcement learning tasks," in *AAAI*, vol. 35, no. 9, 2021, pp. 7995–8003.

[112] E. Lecarpentier, D. Abel, K. Asadi, Y. Jinnai, E. Rachelson, and M. L. Littman, "Lipschitz lifelong reinforcement learning," in *AAAI*, vol. 35, no. 9, 2021, pp. 8270–8278.

[113] Y. Huang, K. Xie, H. Bharadhwaj, and F. Shkurti, "Continual model-based reinforcement learning with hypernetworks," in *ICRA*, 2021, pp. 799–805.

[114] S. Amani, L. Yang, and C. Cheng, "Provably efficient lifelong reinforcement learning with linear representation," in *ICLR*, 2023, pp. 1–42.

[115] Y. Yang, T. Zhou, J. Jiang, G. Long, and Y. Shi, "Continual task allocation in meta-policy network via sparse prompting," in *ICML*, vol. 202, 2023, pp. 39 623–39 638.

[116] E. Ben-Iwhiwhu, S. Nath, P. K. Pilly, S. Kolouri, and A. Soltoggio, "Lifelong reinforcement learning with modulating masks," *Transactions on Machine Learning Research*, 2023.

[117] S. Nath, C. Peridis, E. Ben-Iwhiwhu, X. Liu, S. Dora, C. Liu, S. Kolouri, and A. Soltoggio, "Sharing lifelong reinforcement learning knowledge via modulating masks," in *CoLLAs*, vol. 232, 2023, pp. 936–960.

[118] M. Caccia, J. Mueller, T. Kim, L. Charlin, and R. Fakoor, "Task-agnostic continual reinforcement learning: Gaining insights and overcoming challenges," in *CoLLAs*, vol. 232, 2023, pp. 89–119.

[119] S. Kessler, M. Ostaszewski, M. P. Bortkiewicz, M. Żarski, M. Wolczyk, J. Parker-Holder, S. J. Roberts, and P. Miłoś, "The effectiveness of world models for continual reinforcement learning," in *CoLLAs*, vol. 232, 2023, pp. 184–204.

[120] Y. Luo, Y. Wang, K. Dong, Q. Zhang, E. Cheng, Z. Sun, and B. Song, "Relay hindsight experience replay: Self-guided continual reinforcement learning for sequential object manipulation tasks with sparse rewards," *Neurocomputing*, vol. 557, p. 126620, 2023.

[121] Z. Liu, C. Du, W. S. Lee, and M. Lin, "Locality sensitive sparse encoding for learning world models online," in *ICLR*, 2024, pp. 1–18.

[122] H. Zhang, Y. Lei, L. Gui, M. Yang, Y. He, H. Wang, and R. Xu, "CPPO: continual learning for reinforcement learning with human feedback," in *ICLR*, 2024, pp. 1–24.

[123] Y. Meng, Z. Bing, X. Yao, K. Chen, K. Huang, Y. Gao, F. Sun, and A. Knoll, "Preserving and combining knowledge in robotic lifelong reinforcement learning," *Nature Machine Intelligence*, vol. 7, no. 2, pp. 256–269, 2025.

[124] N. D. Palo, L. Hasenclever, J. Humplik, and A. Byravan, "Diffusion augmented agents: A framework for efficient exploration and transfer learning," in *CoLLAs*, vol. 274, 2025, pp. 268–284.

[125] C. Pan, L. Ren, Y. Feng, L. Xiong, W. Wei, Y. Li, and X. Yang, "Multi-granularity knowledge transfer for continual reinforcement learning," in *IJCAI*, 2025.

[126] F. M. Garcia and P. S. Thomas, "A Meta-MDP approach to exploration for lifelong reinforcement learning," in *NeurIPS*, vol. 32, 2019, pp. 5692–5701.

[127] T. Zhang, Z. Lin, Y. Wang, D. Ye, Q. Fu, W. Yang, X. Wang, B. Liang, B. Yuan, and X. Li, "Dynamics-adaptive continual reinforcement learning via progressive contextualization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 10, pp. 14 588–14 602, 2024.

[128] Y. Chandak, G. Theocharous, C. Nota, and P. S. Thomas, "Lifelong learning with a changing action set," in *AAAI*, vol. 34, no. 04, 2020, pp. 3373–3380.

[129] W. Ding, S. Jiang, H. Chen, and M. Chen, "Incremental reinforcement learning with dual-adaptive ϵ-greedy exploration," in *AAAI*, vol. 37, no. 6, 2023, pp. 7387–7395.

[130] H. Hihn and D. A. Braun, "Hierarchically structured task-agnostic continual learning," *Machine Learning*, vol. 112, no. 2, pp. 655–686, 2023.

[131] H. Bou-Ammar, E. Eaton, J. Luna, and P. Ruvolo, "Autonomous cross-domain knowledge transfer in lifelong policy gradient reinforcement learning," in *IJCAI*, 2015, pp. 3345–3351.

[132] Y.-M. Qian, F.-Z. Xiong, and Z.-Y. Liu, "Zero-shot policy generation in lifelong reinforcement learning," *Neurocomputing*, vol. 446, pp. 65–73, 2021.

[133] M. J. Jacobson, C. Q. Wright, N. Jiang, G. Rodriguez-Rivera, and Y. Xue, "Task detection in continual learning via familiarity autoencoders," in *SMC*, 2022, pp. 1–8.

[134] M. Xu, W. Ding, J. Zhu, Z. Liu, B. Chen, and D. Zhao, "Task-agnostic online reinforcement learning with an infinite mixture of gaussian processes," in *NeurIPS*, vol. 33, 2020, pp. 6429–6440.

[135] Z. Wang, C. Chen, and D. Dong, "A dirichlet process mixture of robust task models for scalable lifelong reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 53, no. 12, pp. 7509–7520, 2023.

[136] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *ICML*, vol. 27, 2012, pp. 17–36.

[137] D. Grbic and S. Risi, "Towards continual reinforcement learning through evolutionary meta-learning," in *GECCO*, 2019, p. 119–120.

[138] G. N. Tasse, S. James, and B. Rosman, "Logical composition in lifelong reinforcement learning," in *ICML LML Workshop*, 2020.

[139] ——, "Generalisation in lifelong reinforcement learning through logical composition," in *ICLR*, 2022, pp. 1–21.

[140] S. Mehimeh, X. Tang, and W. Zhao, "Value function optimistic initialization with uncertainty and confidence awareness in lifelong reinforcement learning," *Knowledge-Based Systems*, vol. 280, p. 111036, 2023.

[141] J. Dick, S. Nath, C. Peridis, E. Benjamin, S. Kolouri, and A. Soltoggio, "Statistical context detection for deep lifelong reinforcement learning," in *CoLLAs*, 2024.

[142] H. Bou-Ammar, E. Eaton, P. Ruvolo, and M. E. Taylor, "Online multi-task learning for policy gradient methods," in *ICML*, vol. 32, 2014, pp. 1206–1214.

[143] H. Bou-Ammar, R. Tutunov, and E. Eaton, "Safe policy search for lifelong reinforcement learning with sublinear regret," in *ICML*, vol. 37, 2015, pp. 2361–2369.

[144] Y. Zhan, H. B. Ammar, and M. E. Taylor, "Scalable lifelong reinforcement learning," *Pattern Recognition*, vol. 72, pp. 407–418, 2017.

[145] R. Mowakeaa, S.-J. Kim, and D. K. Emge, "Kernel-based lifelong policy gradient reinforcement learning," in *ICASSP*, 2021, pp. 3500–3504.

[146] F. Ding and F. Zhu, "HLifeRL: A hierarchical lifelong reinforcement learning framework," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4312–4321, 2022.

[147] R. Traoré, H. Caselles-Dupré, T. Lesort, T. Sun, N. Díaz-Rodríguez, and D. Filliat, "Continual reinforcement learning deployed in real-life using policy distillation and Sim2Real transfer," in *ICML MLRL Workshop*, 2019.

[148] C. Doyle, M. Guériau, and I. Dusparic, "Variational policy chaining for lifelong reinforcement learning," in *ICTAI*, 2019, pp. 1546–1550.

[149] Y. Shi, L. Yuan, Y. Chen, and J. Feng, "Continual learning via bit-level information preserving," in *CVPR*, 2021, pp. 16 674–16 683.

[150] C. Zhao, J. Xu, R. Peng, X. Chen, K. Mei, and X. Lan, "Experience consistency distillation continual reinforcement learning for robotic manipulation tasks," in *ICRA*, vol. 33, 2024, pp. 501–507.

[151] W. Zhou, S. Bohez, J. Humplik, N. Heess, A. Abdolmaleki, D. Rao, M. Wulfmeier, and T. Haarnoja, "Forgetting and imbalance in robot lifelong learning with off-policy data," in *CoLLAs*, vol. 199, 2022, pp. 294–309.

[152] A. Xie and C. Finn, "Lifelong robotic reinforcement learning by retaining experiences," in *CoLLAs*, vol. 199, 2022, pp. 838–855.

[153] M. Xu, X. Chen, and J. Wang, "Policy correction and state-conditioned action evaluation for few-shot lifelong deep reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2024.

[154] C. Li, Y. Li, Y. Zhao, P. Peng, and X. Geng, "SLER: Self-generated long-term experience replay for continual reinforcement learning," *Applied Intelligence*, vol. 51, no. 1, pp. 185–201, 2021.

[155] H. Caselles-Dupré, M. Garcia-Ortiz, and D. Filliat, "S-TRIGGER: Continual state representation learning via self-triggered generative replay," in *IJCNN*, 2021, pp. 1–7.

[156] A. Nagabandi, C. Finn, and S. Levine, "Deep online learning via meta-learning: Continual adaptation for model-based RL," in *ICLR*, 2019, pp. 1–15.

[157] Z. Wang, C. Chen, and D. Dong, "Lifelong incremental reinforcement learning with online bayesian inference," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 4003–4016, 2022.

[158] J. A. Mendez, S. Shivkumar, and E. Eaton, "Lifelong inverse reinforcement learning," in *NeurIPS*, vol. 31, 2018, pp. 4507–4518.

[159] N. Bougie and R. Ichise, "Intrinsically motivated lifelong exploration in reinforcement learning," in *AISC*, 2021, pp. 109–120.

[160] K. Chu, X. Zhu, and W. Zhu, "Accelerating lifelong reinforcement learning via reshaping rewards," in *SMC*, 2021, pp. 619–624.

[161] C. A. Steinparz, T. Schmied, F. Paischer, M.-c. Dinu, V. P. Patil, A. Bitto-nemling, H. Eghbal-zadeh, and S. Hochreiter, "Reactive exploration to cope with non-stationarity in lifelong reinforcement learning," in *CoLLAs*, vol. 199, 2022, pp. 441–469.

[162] E. Meyerson and R. Miikkulainen, "Beyond shared hierarchies: Deep multitask learning through soft layerordering," in *ICLR*, 2018, pp. 1–14.

[163] M. Chang, A. Gupta, S. Levine, and T. L. Griffiths, "Automatically composing representation transformations as a means for generalization," in *ICLR*, 2019, pp. 1–23.

[164] J. Pfeiffer, S. Ruder, I. Vulić, and E. Ponti, "Modular deep learning," *Transactions on Machine Learning Research*, 2023.

[165] A. Stocco, C. Lebiere, and J. R. Anderson, "Conditional routing of information to the cortex: A model of the basal ganglia's role in cognitive coordination." *Psychological Review*, vol. 117, no. 2, p. 541, 2010.

[166] A. J. Kell, D. L. Yamins, E. N. Shook, S. V. Norman-Haignere, and J. H. McDermott, "A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy," *Neuron*, vol. 98, no. 3, pp. 630–644, 2018.

[167] J. A. Mendez and E. Eaton, "Lifelong learning of compositional structures," in *ICLR*, 2021, pp. 1–25.

[168] S. Pateria, B. Subagdja, A.-h. Tan, and C. Quek, "Hierarchical reinforcement learning: A comprehensive survey," *ACM Computing Surveys*, vol. 54, no. 5, pp. 1–35, 2021.

[169] A. G. Barto and S. Mahadevan, "Recent advances in hierarchical reinforcement learning," *Discrete Event Dynamic Systems*, vol. 13, pp. 341–379, 2003.

[170] V. K. Chauhan, J. Zhou, P. Lu, S. Molaei, and D. A. Clifton, "A brief review of hypernetworks in deep learning," *Artificial Intelligence Review*, vol. 57, no. 9, p. 250, 2024.

[171] J. von Oswald, C. Henning, J. Sacramento, and B. F. Grewe, "Continual learning with hypernetworks," in *ICLR*, 2020, pp. 1–28.

[172] M. Xu, X. Chen, and J. Wang, "Policy correction and state-conditioned action evaluation for few-shot lifelong deep reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2024.

[173] M. Wortsman, V. Ramanujan, R. Liu, A. Kembhavi, M. Rastegari, J. Yosinski, and A. Farhadi, "Supermasks in superposition," in *NeurIPS*, vol. 33, 2020, pp. 15 173–15 184.

[174] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: Survey and performance evaluation on image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, p. 5513–5533, 2022.

[175] Z. Wang, E. Yang, L. Shen, and H. Huang, "A comprehensive survey of forgetting in deep learning beyond continual learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2024.
[176] N. Wang, D. Zhang, and Y. Wang, "Learning to navigate for mobile robot with continual reinforcement learning," in *CCC*, 2020, pp. 3701–3706.
[177] M. Burhan Hafez and S. Wermter, "Behavior self-organization supports task inference for continual robot learning," in *IROS*, 2021, pp. 6739–6746.
[178] M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh, "Reward design with language models," in *ICLR*, 2023, pp. 1–18.
[179] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik, B. Ichter, T. Xiao, P. Xu, A. Zeng, T. Zhang, N. Heess, D. Sadigh, J. Tan, Y. Tassa, and F. Xia, "Language to rewards for robotic skill synthesis," in *CoRL*, 2023.
[180] R. Aljundi, D. O. Reino, N. Chumerin, and R. E. Turner, "Continual novelty detection," in *CoLLAs*, vol. 199, 2022, pp. 1004–1025.
[181] X. Liu, Y. Bai, Y. Lu, A. Soltoggio, and S. Kolouri, "Wasserstein task embedding for measuring task similarities," *Neural Networks*, vol. 181, p. 106796, 2025.
[182] G. Sibo, W. Donglin, and H. Li, "OER: Offline experience replay for continual offline reinforcement learning," *ArXiv preprint*, vol. abs/2305.13804, 2023.
[183] L. Chen, S. Jayanthi, R. R. Paleja, D. Martin, V. Zakharov, and M. Gombolay, "Fast lifelong adaptive inverse reinforcement learning from demonstrations," in *CoRL*, vol. 205, 2023, pp. 2083–2094.
[184] T. Kobayashi and T. Sugino, "Reinforcement learning for quadrupedal locomotion with design of continual–hierarchical curriculum," *Engineering Applications of Artificial Intelligence*, vol. 95, p. 103869, 2020.
[185] R. Julian, B. Swanson, G. Sukhatme, S. Levine, C. Finn, and K. Hausman, "Never stop learning: The effectiveness of fine-tuning in robotic reinforcement learning," in *CoRL*, 2021, pp. 2120–2136.
[186] F. Yang, C. Yang, H. Liu, and F. Sun, "Evaluations of the gap between supervised and reinforcement lifelong learning on robotic manipulation tasks," in *CoRL*, vol. 164, 2022, pp. 547–556.
[187] A. Xie and C. Finn, "Lifelong robotic reinforcement learning by retaining experiences," in *CoLLAs*, 2022, pp. 838–855.
[188] N. Justesen, P. Bontrager, J. Togelius, and S. Risi, "Deep learning for video game playing," *IEEE Transactions on Games*, vol. 12, no. 1, pp. 1–20, 2020.
[189] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman, "Leveraging procedural generation to benchmark reinforcement learning," in *ICML*, vol. 119, 2020, pp. 2048–2056.
[190] Q. Delfosse, J. Blüml, B. Gregori, and K. Kersting, "HackAtari: Atari learning environments for robust and continual reinforcement learning," in *RLC IPRL Workshop*, 2024.
[191] I. Sur, Z. Daniels, A. Rahman, K. Faber, G. Gallardo, T. Hayes, C. Taylor, M. B. Gurbuz, J. Smith, S. Joshi, N. Japkowicz, M. Baron, Z. Kira, C. Kanan, R. Corizzo, A. Divakaran, M. Piacentino, J. Hostetler, and A. Raghavan, "System design for an integrated lifelong reinforcement learning agent for real-time strategy games," in *AIMLSystems*, 2023, pp. 1–9.
[192] C. Geishauser, C. van Niekerk, H.-c. Lin, N. Lubis, M. Heck, S. Feng, and M. Gašić, "Dynamic dialogue policy for continual reinforcement learning," in *COLING*, 2022, pp. 266–284.
[193] V. Shulev and K. Sima'an, "Continual reinforcement learning for controlled text generation," in *COLING*, 2024, pp. 3881–3889.
[194] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize with human feedback," in *NeurIPS*, vol. 33, 2020, pp. 3008–3021.
[195] G. Zheng, S. Zhou, V. Braverman, M. A. Jacobs, and V. S. Parekh, "Selective experience replay compression using coresets for lifelong deep reinforcement learning in medical imaging," in *MIDL*, 2024, pp. 1751–1764.
[196] S. Liu, B. Wang, H. Li, C. Chen, and Z. Wang, "Continual portfolio selection in dynamic environments via incremental reinforcement learning," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 1, pp. 269–279, 2023.
[197] A. Q. Md, D. Jaiswal, S. Mohan, N. Innab, R. Sulaiman, M. K. Alaoui, and A. Ahmadian, "A novel approach for self-driving car in partially observable environment using lifelong reinforcement learning," *Sustainable Energy, Grids and Networks*, vol. 38, p. 101356, 2024.
[198] Y. Wang, F. Shang, and J. Lei, "Multi-granularity fusion resource allocation algorithm based on dual-attention deep reinforcement learning and lifelong learning architecture in heterogeneous IIoT," *Information Fusion*, vol. 99, p. 101871.
[199] R. Wang, Z. Cao, X. Zhou, Y. Wen, and R. Tan, "Phyllis: Physics-informed lifelong reinforcement learning for data center cooling control," in *ACM E-Energy*, 2023, pp. 114–126.
[200] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, and D. Guo, "DeepSeekMath: Pushing the limits of mathematical reasoning in open language models," *ArXiv preprint*, vol. abs/2402.03300, 2024.
[201] J. Hu, Y. Zhang, Q. Han, D. Jiang, X. Zhang, and H.-Y. Shum, "Open-Reasoner-Zero: An open source approach to scaling up reinforcement learning on the base model," *ArXiv preprint*, vol. abs/2503.24290, 2025.
[202] J. Zhang, J. Zhang, K. Pertsch, Z. Liu, X. Ren, M. Chang, S.-H. Sun, and J. J. Lim, "Bootstrap your own skills: Learning to solve new tasks with large language model guidance," in *CoRL*, 2023.
[203] K. Chen, Y. Du, T. You, M. Islam, Z. Guo, Y. Jin, G. Chen, and P.-A. Heng, "LLM-assisted multi-teacher continual learning for visual question answering in robotic surgery," in *ICRA*, 2024, pp. 10 772–10 778.
[204] H. Bai, Y. Zhou, J. Pan, M. Cemri, A. Suhr, S. Levine, and A. Kumar, "DigiRL: Training in-the-wild device-control agents with autonomous-reinforcement learning," in *NeurIPS*, 2024.
[205] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D. Huang, Y. Zhu, and A. Anandkumar, "MineDojo: Building open-ended embodied agents with internet-scale knowledge," in *NeurIPS*, vol. 35, 2022, pp. 18 343–18 362.
[206] J. Mendez-Mendez, L. P. Kaelbling, and T. Lozano-Pérez, "Embodied lifelong learning for task and motion planning," in *CoRL*, vol. 229, 2023, pp. 2134–2150.
[207] G. Tziafas and H. Kasaei, "Lifelong robot library learning: Bootstrapping composable and generalizable skills for embodied control with language models," in *ICRA*, 2024, pp. 515–522.
[208] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," *Transactions on Machine Learning Research*, 2024.
[209] B. Kim, M. Seo, and J. Choi, "Online continual learning for interactive instruction following agents," in *ICLR*, 2024, pp. 1–18.
[210] X. Zeng, H. Luo, Z. Wang, S. Li, Z. Shen, and T. Li, "A continual learning approach for embodied question answering with generative adversarial imitation learning," in *ICASSP*, 2025, pp. 1–5.

VII. Biography Section

Chaofan Pan received the B.S. and M.S. degrees from Southwest Petroleum University, Chengdu, China, in 2020 and 2023, respectively. Now he is pursuing a Ph.D. degree from the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics. His main research interests include reinforcement learning, self-supervised learning, and continual reinforcement learning.

Xin Yang (Member, IEEE) received the Ph.D. degree in computer science from Southwest Jiaotong University, Chengdu, in 2019. He is currently a Professor at the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics. He has authored more than 100 research papers in refereed journals and conferences. His research interests include federated learning, continual learning, and multi-granularity learning.

Yanhua Li received the M.S. degree from Southwest Petroleum University, Chengdu, China, in 2022. She is currently working toward a Ph.D. degree at the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu, China. Her research interests include continual learning, multi-granularity learning, and open intent classification. She has published several papers in conferences and journals, such as AAAI and Information Fusion.

Wei Wei (Member, IEEE) received the Ph.D. degree in computer science from Shanxi University in 2012. He is currently a professor with the School of Computer and Information Technology, Shanxi University. He has authored or coauthored more than 20 journal papers in his research fields. His research interests include reinforcement learning and granular computing.

Tiranrui Li (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Southwest Jiaotong University, Chengdu, China, in 1992, 1995, and 2002, respectively. He was a post-doctoral researcher with Belgian Nuclear Research Centre, Mol, Belgium, from 2005 to 2006, and a visiting professor with Hasselt University, Hasselt, Belgium, in 2008; the University of Technology, Sydney, Australia, in 2009; and the University of Regina, Regina, Canada, in 2014. He is currently a professor and the director of the Key Laboratory of Cloud Computing and Intelligent Techniques, Southwest Jiaotong University. He has authored or coauthored more than 300 research papers in refereed journals and conferences. His research interests include Big Data, machine learning, data mining, granular computing, and rough sets.

Bo An (Senior Member, IEEE) received his Ph.D. degree in Computer Science from the University of Massachusetts, Amherst, MA, USA, in 2010. He is a President's Council Chair professor at Nanyang Technological University. His research interests include artificial intelligence, multi-agent systems, reinforcement learning, game theory, and optimization.

Jiye Liang (Fellow, IEEE) received the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1990 and 2001, respectively. He is currently a professor with the School of Computer and Information Technology, Shanxi University, Taiyuan, China, where he is also the director of the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education. He has authored or coauthored more than 200 papers in his research fields, including the Artificial Intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence, Journal of Machine Learning Research, ICML, AAAI, and so on. His current research interests include data mining, machine learning, and artificial intelligence.