Same State, Different Task:
Continual Reinforcement Learning without Interference

RL 논문 리뷰 스터디 10기 민예린
2023.03.20

Contents

1 Introduction

2 Related work

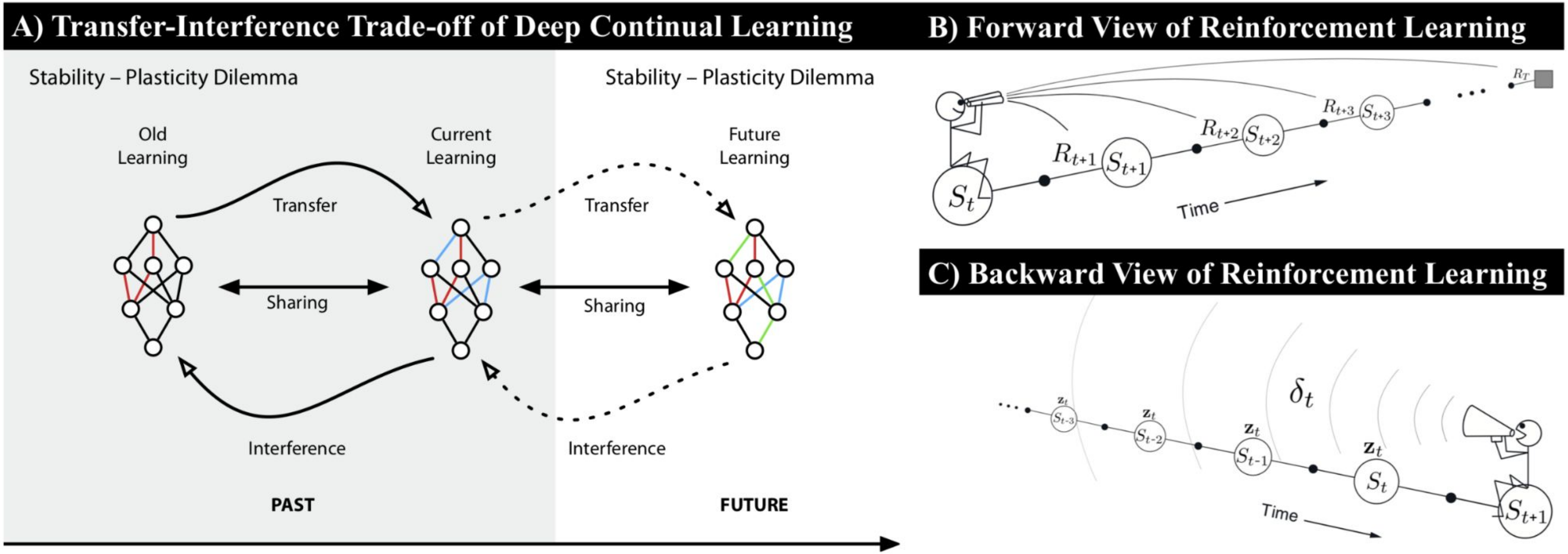
3 COntinual RL Without ConfLict

4 Experiments

Introduction

Catastrophic forgetting

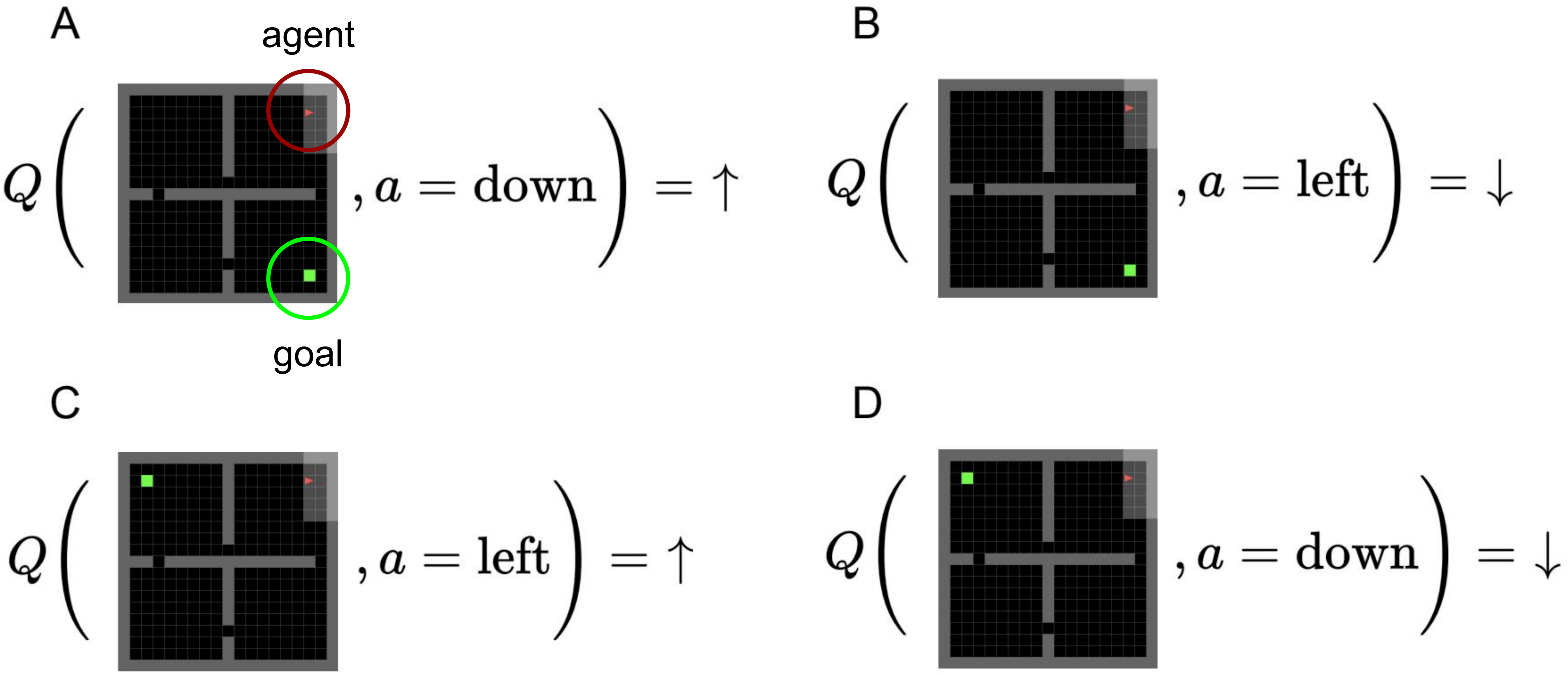
A key challenge in CL is catastrophic forgetting, which arises when performance on a previously mastered task is reduced when learning a new task.



Towards Continual Reinforcement Learning: A Review and Perspectives

Interference

we call “interference” which can in turn induce forgetting, as the agent directly optimizes for an opposing policy.



C<u>O</u>ntinual RL <u>W</u>ithout Conf<u>L</u>ict (OWL)

- Previous <u>CRL methods</u> used <u>different environments as different tasks</u> then the agents can learn that the different state spaces correspond to different optimal behaviors and so interference is rarely exhibited.

- We show that <u>existing CL methods based on single neural network</u> predictors with <u>shared replay buffers fail</u> in the presence of interference.
    - existing replay based methods such as (Rolnick et al. 2019) fail to address this issue, as the experience replay buffer will contain tuples of the same state-action pairs but different rewards for different tasks.
    - Thus, the agent will not converge.

- OWL makes use of <u>shared feature extraction layers</u>, while acting based on s<u>eparate independent policy heads.</u>

Related Work

Continual Learning

Continual Learning (CL) considers the problem of training an agent sequentially on a set of tasks while seeking to retain performance on all previous tasks.

figure 1

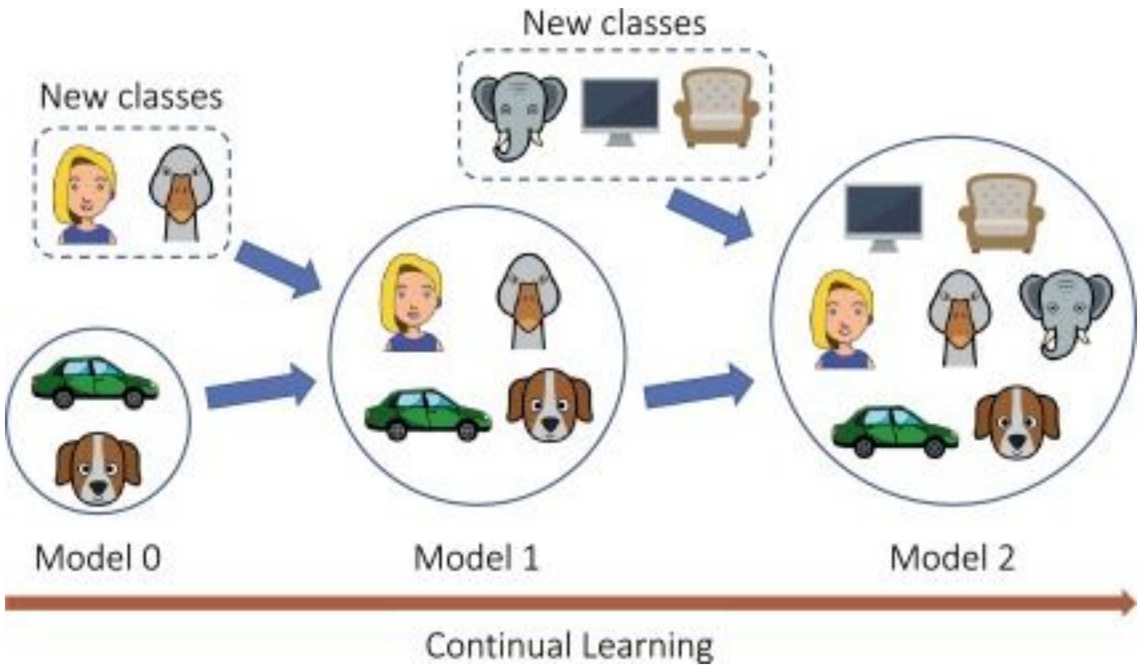


figure 2

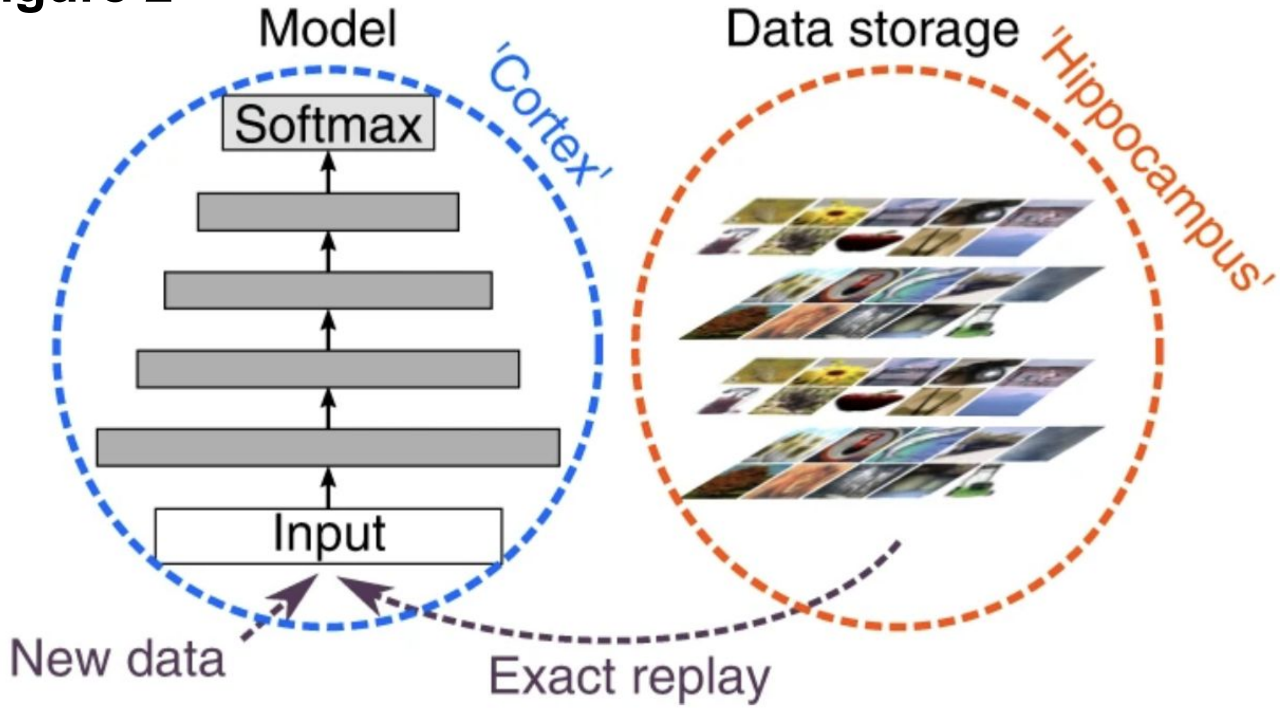
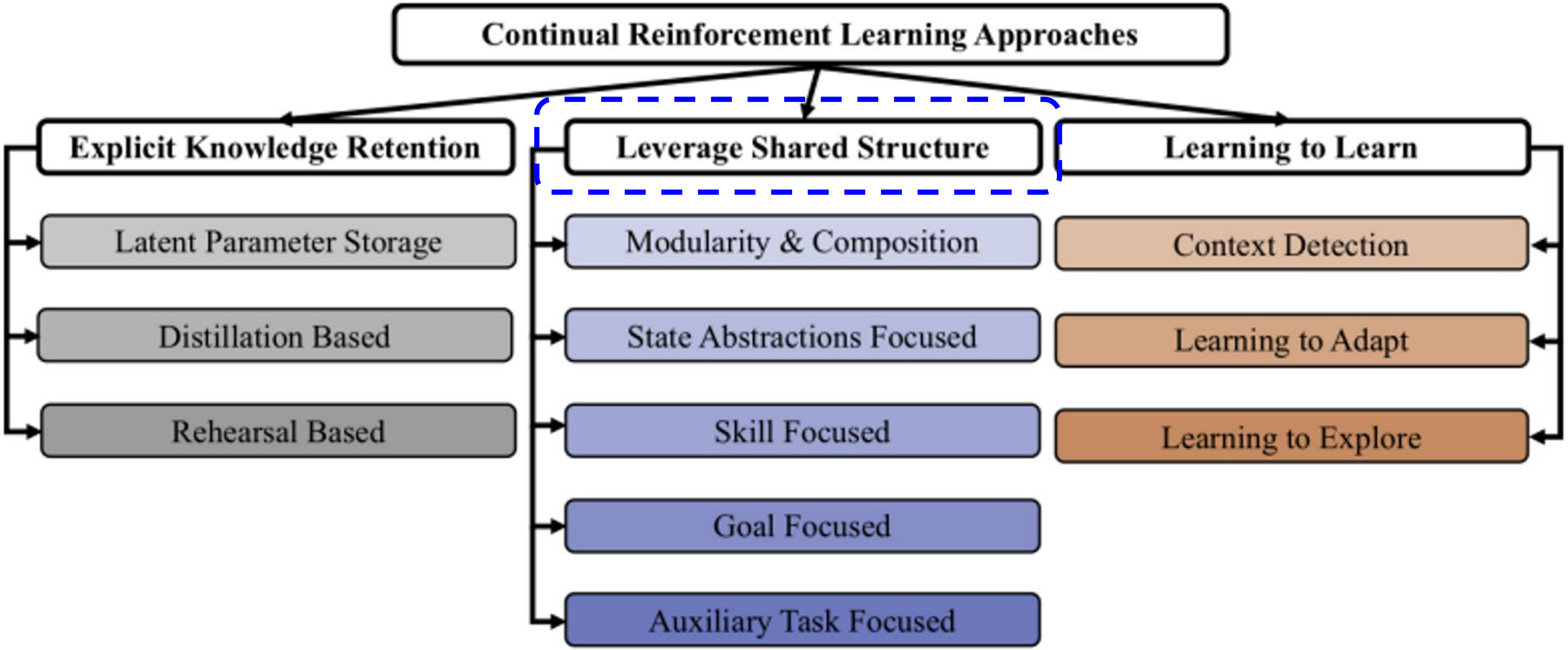


figure 1. Domain adaptation and continual learning in semantic segmentation
figure 2. Brain-inspired replay for continual learning with artificial neural networks

Continual Reinforcement Learning



Towards Continual Reinforcement Learning: A Review and Perspectives

Observation and Interference

- This observation has important consequences: methods which are task agnostic and do not condition on the task or do not use task specific parameters are susceptible to interference.

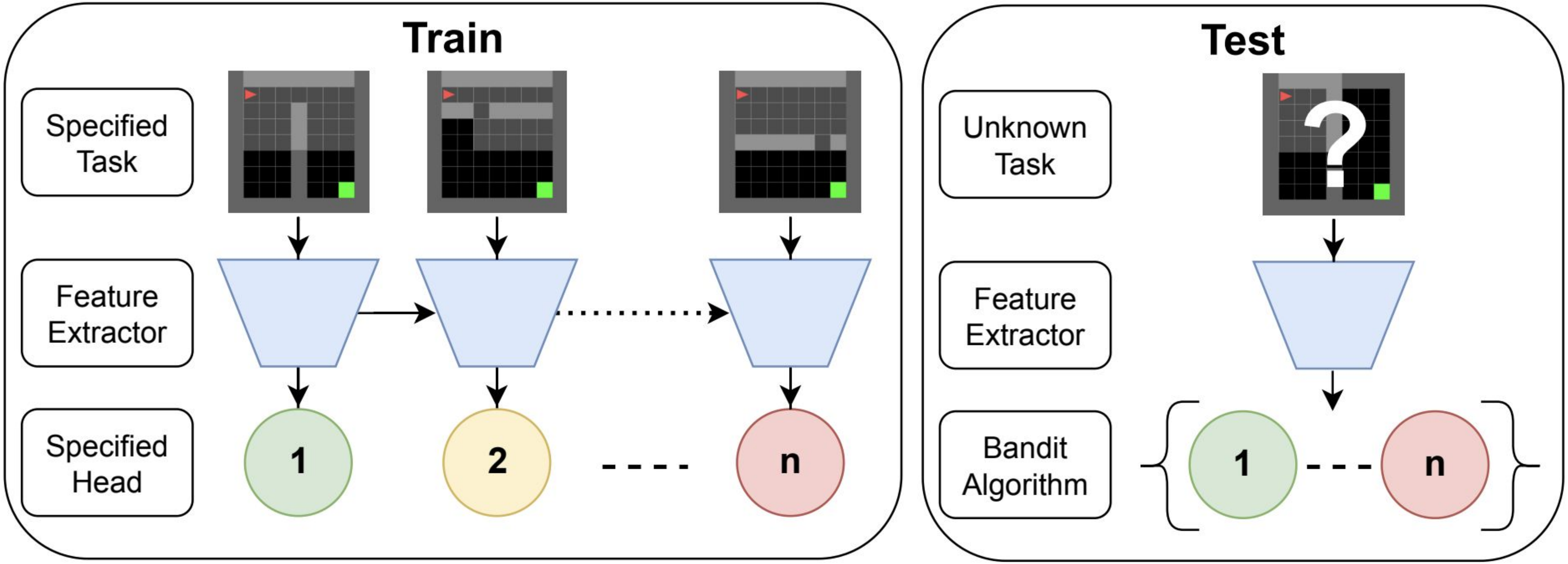
Observation 4.1. *Consider two tasks $\mathcal{T}_i$ and $\mathcal{T}_j$. Let both tasks' input distributions $p_k(X)$ share the same support but have different conditional distributions $p_k(Y|X) = \mathcal{N}(f^k(X), \beta^{-1})$, where f^k is a mean function with $f^i \neq f^j$ and β^{-1} is data noise. Then the multi-task distribution is bi-modal and using a Gaussian likelihood will result in interference.*

- Consider a partially observable MDP (POMDP) where we receive an initial observation but do not know the goal location or reward function then an agent might require different policies for each task

COntinual RL Without ConfLict (OWL)

Key insight

1) we can use a single network with a <u>shared feature extractor but multiple heads</u>, parameterized by linear layers to fit individual tasks.
2) we <u>flush the experience replay buffer when starting to learn in a new task</u>.



Factorized Q-Functions

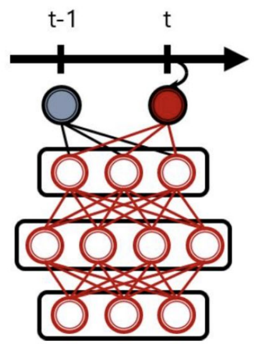
To address forgetting in the shared neural network feature extractors we use regularization methods.

Algorithm 1: OWL: Training

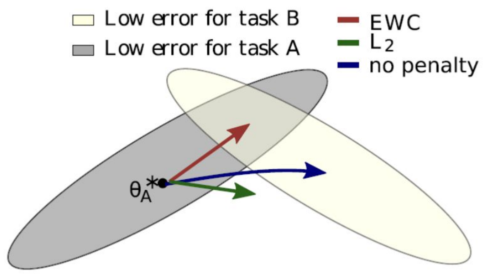
Input: Tasks $\mathcal{T} = \{\mathcal{T}_i\}_{i=1}^{M}$.
Initialize: θ and ϕ, $\Omega^Q = \Omega^\pi = \emptyset$.
for $t = 1, 2, \ldots, M$ **do**
1. See Task $\mathcal{T}_t$
shared
2. Train Q-function with parameters $\{\theta_z, \theta_i\}$ and regularization Ω^Q.
head
if $\mathcal{A}$ *is continuous* **then**
3. Train policy with parameters $\{\phi_z, \phi_i\}$ with regularization Ω^π.
4. Calculate Q-function EWC regularization and $\Omega^Q := \{\mathcal{L}^Q_{\text{EWC}}, \Omega^Q\}$.
if $\mathcal{A}$ *is continuous* **then**
5. Calculate policy EWC regularization and $\Omega^\pi := \{\mathcal{L}^\pi_{\text{EWC}}, \Omega^\pi\}$.
6. Empty the experience reply buffer $\mathcal{D} = \emptyset$.
7. Evaluate according to Algorithm 2.

- As we see more and more tasks new heads can easily be added and so we do not need to pre-specify the number of tasks or policy heads M ∈ {1, . . . , ∞}.

- slowing down learning on the weights important for those tasks.



Retraining without expansion



Elastic Weight Consoliation(Kirkpatrick et al., 2017)

https://github.com/skezle/owl/blob/a05083d7f95df9af511f2c6a95d007db75a4671a/dqn.py#L410

Selecting Policies as a Multi-Armed Bandit Problem

At test time we do not tell OWL which task it is being evaluated on.

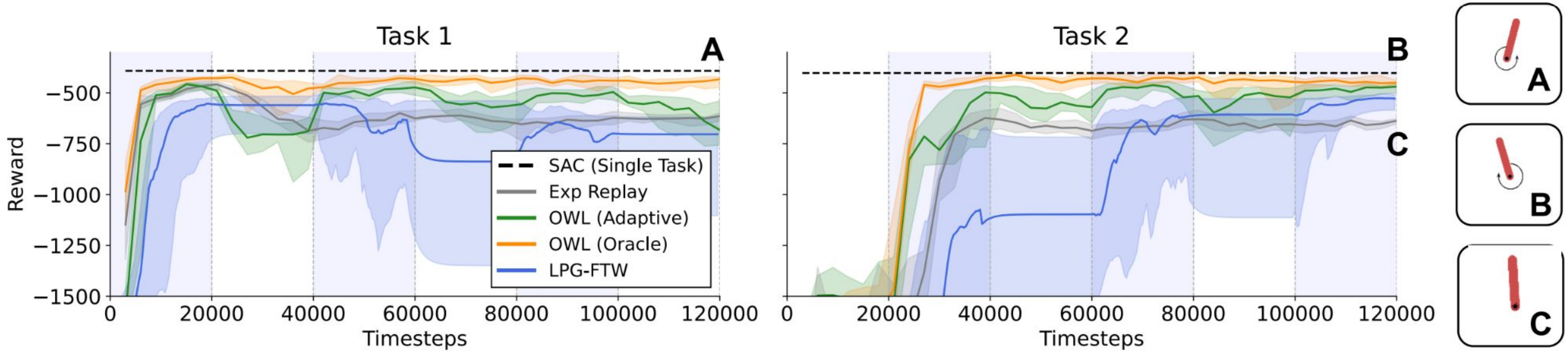
- We consider the set of arms M to be the set of policies which can be chosen to act at each time step of the test task.
- The aim is to find the policy which achieves the highest reward on a given test task.

Algorithm 2: OWL: Testing

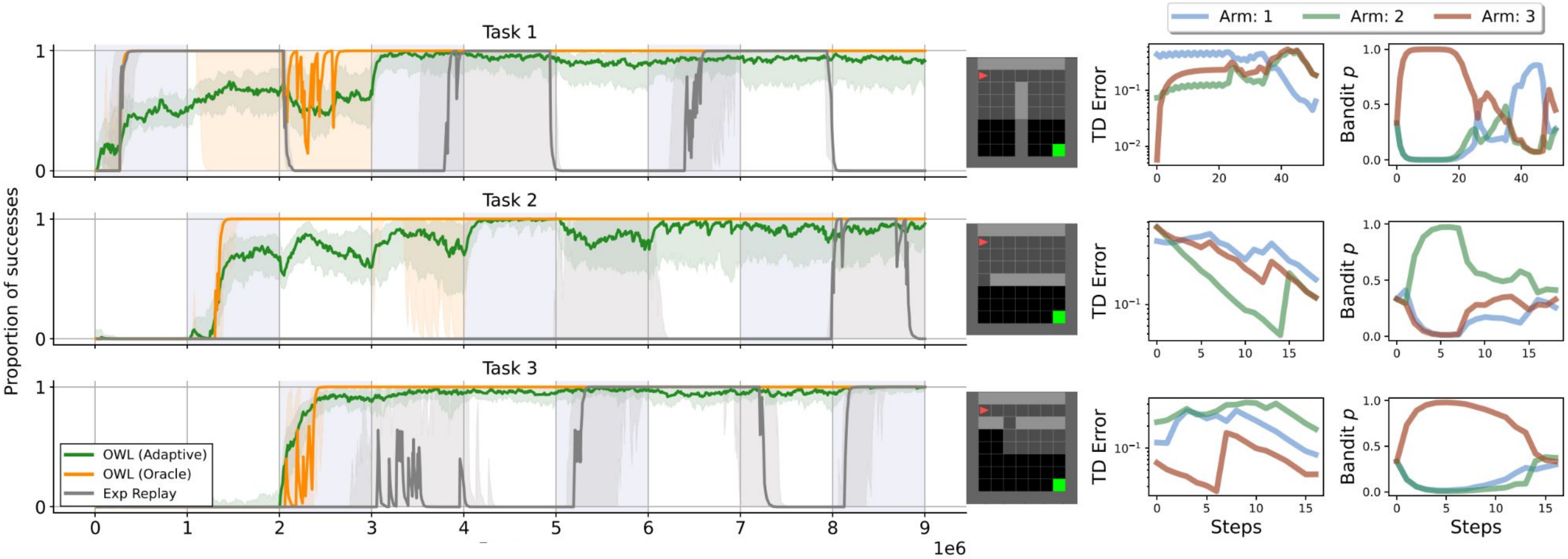
Input: tasks seen so far $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_\tau\}$, Q-functions $\{\phi_i\}_{i=1}^{M}$, step size η, maximum number of timesteps T.
Initialize: $\mathbf{p}_\phi^1$ as a uniform distribution, s_1 as the initial state of the test task.
for $\mathcal{T}_j \in \mathcal{T}$ **do**
  for $t = 1, \ldots, T-1$ **do**
    1. Select $i_t \sim \mathbf{p}_\phi^t$, and set $\pi_{\text{test}} = \pi_{\phi_{i_t}}$.
    2. Take action $a_t \sim \pi_{\text{test}}(s_t)$, and receive reward r_t and the next state s_{t+1} from $\mathcal{T}_j$.
    3. Use Equation 2 to update $\mathbf{p}_\phi^t$ with $l_{i_t}^t = \hat{G}_{\phi_t}(\theta_{t+1})$

Experiments

Experiments (1)



Experiments (2)



	SC	SC+DK
Exp Replay	0.01 (0.61, 0.00)	0.00 (0.52, 0.00)
OWL (orcl)	**0.85** (0.97, 0.72)	**0.60 (0.98, 0.44)**
OWL (adpt)	0.59 (0.75, 0.48)	**0.63 (0.79, 0.45)**
OWL - EWC (orcl)	0.45 (0.53, 0.39)	0.40 (0.48, 0.30)
OWL - EWC (adpt)	0.49 (0.60, 0.39)	**0.50 (0.62, 0.37)**
OWL - EWC + DL (orcl)	0.45 (0.53, 0.36)	0.34 (0.40, 0.29)
OWL - EWC + DL (bndt)	0.53 (0.61, 0.38)	0.39 (0.45, 0.33)
Full Rehearsal	0.99 (0.99, 0.97)	0.99 (1.00, 0.98)

- SC : Simple Crossing tasks
- DK : Door key environment
- Exp replay buffer size : 4M

Q & A