Adaptive Continual Learning: Rapid Adaptation and Knowledge Refinement

Jin Du
School of Statistics
University of Minnesota-Twin Cities
Minneapolis, MN 55455, USA
du000142@umn.edu

Yuhong Yang
School of Statistics
University of Minnesota-Twin Cities
Minneapolis, MN 55455, USA
yangx374@umn.edu

Jie Ding
School of Statistics
University of Minnesota-Twin Cities
Minneapolis, MN 55455, USA
dingj@umn.edu

Abstract

Continual learning (CL) is an emerging research area aiming to emulate human learning throughout a lifetime. Most existing CL approaches primarily focus on mitigating catastrophic forgetting, a phenomenon where performance on old tasks declines while learning new ones. However, human learning involves not only retaining knowledge but also quickly recognizing the current environment, recalling related knowledge, and refining it for improved performance. In this work, we introduce a new problem setting, Adaptive CL, which captures these aspects in an online, recurring task environment without explicit task boundaries or identities. We propose the LEARN algorithm to efficiently explore, recall, and refine knowledge in such environments. We provide theoretical guarantees from two perspectives: online prediction with tight regret bounds and asymptotic consistency of knowledge. Additionally, we present a scalable implementation that requires only first-order gradients for training deep learning models. Our experiments demonstrate that the LEARN algorithm is highly effective in exploring, recalling, and refining knowledge in adaptive CL environments, resulting in superior performance compared to competing methods.

1 Introduction

Inspired by the process of human lifelong learning, Continual Learning (CL), also referred to as Lifelong Learning, aims to develop models that can sequentially learn tasks, simultaneously preserving and consolidating existing knowledge. The primary focus of CL approaches is on preventing catastrophic forgetting [1, 2], a phenomenon where the performance of previously learned tasks declines as new tasks are learned [3]. Traditional CL literature [4–8] mainly addresses a sequence of tasks with known task identities. In recent years, however, the focus has shifted towards more challenging scenarios in CL research, with growing interest in one scenario called task-free CL [9–13], where task identities and boundaries are unknown during training. In these instances, it becomes crucial for the learner to comprehend the current environment and incorporate new information without catastrophic forgetting, a more challenging problem due to the lack of task information.

While numerous technical approaches have been developed in CL to mitigate forgetting during the learning of new tasks, an underexplored area is the enhancement of machine performance on recurring

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

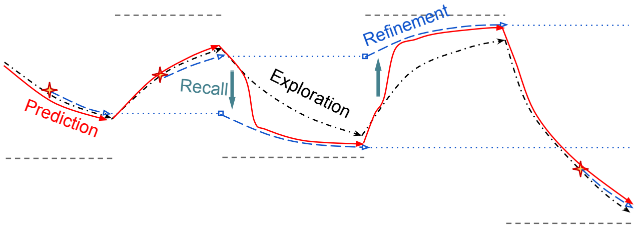


Figure 1: An illustration of the proposed LEARN algorithm: 1) **Exploration** (black lines): the fast learner updates using the new data. 2) **Recall** (green arrows): the output of the fast learner is dynamically mixed with those of the slow learner 3) **Refinement** (blue lines): each model in the slow learner is updated with varying learning rates. The ground truth is marked as gray dashed lines.

tasks through the swift task recognition and recall of prior information. This is a critical aspect of human lifelong learning. As humans encounter changing environments, they can swiftly recollect associated memory and adapt their learning when tasks switch, gradually building a knowledge base to improve their effectiveness and proficiency in recurring tasks. We believe such a learning process involves three key components: **quick recognition of new environments**, **recall of related knowledge**, and **refinement of existing knowledge**. Consequently, this highlights the need to explore the broader scope of CL problems that not only address catastrophic forgetting but also incorporate the above three human-inspired capabilities.

In this paper, we introduce Adaptive CL, a novel problem designed to capture the multi-dimensional nature of human learning. This framework involves learning from dynamic online environments with recurring tasks, while task boundaries and identities remain unknown. To perform well, learners are expected to make three interrelated decisions 1) swiftly identify the current task, overcoming challenges posed by unknown task boundaries and identities, 2) recall knowledge on previous tasks to adapt effectively to recurring tasks, and 3) refine existing knowledge by integrating new information, thereby improving future performance on the same task.

To tackle the Adaptive CL problem, we will develop a novel approach that automatically unifies the above three decisions with provable performance guarantees. A critical reader may question why not adopt a straightforward approach of treating the above decisions as three separate steps: change-point detection, hypothesis-testing, and online update. Although this heuristic could provide a viable approach, errors in separate steps may propagate and accumulate over time, resulting in a disconnected, inefficient, and complex continual learning process. Instead, we propose a unified algorithm called LEARN (Lifelong Exploration, recAll, and Refinement of kNowledge). LEARN consists of three stages: exploration, recall, and refinement, as illustrated in Figure 1. During the exploration stage (black dashed lines), a fast learner is updated according to new data observations; In the recall stage (green arrows), the prediction (red lines) leverages recent information and stored knowledge by mixing the fast learner with the slow learner, enabling swift adaptation to familiar tasks; In the refinement stage (blue dashed lines), the slow learner updates the stored knowledge for future use by integrating the fast learner. Furthermore, we present a comprehensive theoretical foundation by offering guarantees from dual perspectives: performance-level regret bound that ensures near-optimal decision-making, and knowledge-level consistency to the ground truth in hindsight. These guarantees ensure that our approach is both powerful in prediction and interpretable in gained knowledge in the Adaptive CL context. For application in large-scale models such as deep neural networks, we introduce a Gaussian mixture model (GMM) approximation of the proposed LEARN, which requires only first-order gradients for training deep learning models. This approximation enables efficient training while preserving the efficiency and interpretability of our approach, making it a practical solution for real-world applications. Our experimental evaluation illustrates the success of the LEARN algorithm in several simulated and benchmark data cases.

1.1 Main contributions

In this paper, our contributions are highlighted as follows:

• We propose Adaptive CL, a novel framework that cooperates with a broader set of human learning characteristics: rapid task recognition, efficient recall of related knowledge, and continuous refinement of knowledge.

• We develop the LEARN algorithm, a tailored solution that addresses the aforementioned challenges through a three-step process that encompasses exploration, recall, and refinement. We establish the theoretical foundations for the LEARN algorithm by delivering performance-level regret bounds and knowledge-level asymptotic consistency.

• We provide a scalable implementation of our algorithm, which facilitates its usage in deep learning. Our proposed LEARN algorithm exhibits a superior ability to explore, recall, and refine knowledge in Adaptive CL settings, surpassing the performance of competing methods.

1.2 Related work

Continual learning Continual Learning (CL) targets learning in dynamic environments with restricted historical data access. Many existing works have achieved significant success in preventing catastrophic forgetting, namely preserving performance on old tasks while learning new tasks. Existing CL approaches can primarily be divided into three categories: regularization, replay, and dynamic architecture. Regularization-based methods [4, 14–16] minimize forgetting by imposing constraints on critical parameters from previous tasks. Replay approaches generate pseudo samples [5] or store actual samples [6, 17] of prior tasks to implicitly safeguard essential parameters. Alternatively, stored data can be used to constrain optimization [7, 18, 19], preventing gradient updates in crucial directions. Finally, dynamic architecture methods either train separate masks of a dense neural network [8, 20, 21] or maintain dynamic model structures [22, 23]. Experimental results demonstrate the superior performance of these methods in efficiently retaining knowledge and preventing catastrophic forgetting when training in a changing environment. Many recent works have theoretically investigated the cause of forgetting, specifically the impact of factors such as task similarity and ordering on generalization performance [24, 25]. Additionally, the semi-supervised and unsupervised CL settings have also been studied [26, 27].

Task-free continual learning Task-free CL presents a more complex scenario than traditional CL, as it deals with unknown task boundaries and identities during training. In this setting, learners must retain knowledge to prevent catastrophic forgetting and quickly recognize the current task. Existing works have proposed replay-based and dynamic architecture methods. Replay-based methods [11, 28, 29] maintain a small buffer of previous data and replay a small batch every step. The dynamic architecture approach expands the number of models upon detecting a new task using the Dirichlet process [10] or discrepancy distances [13, 30].

Both traditional and task-free CL contribute to a profound understanding of how to learn without catastrophic forgetting during training, and they demonstrate promising results in experiments. In this paper, we aim to enhance the CL framework by integrating training and testing stages to better emulate realistic human learning scenarios. The learner must not only adapt to the changing environment but also efficiently exploit knowledge by recalling and consolidating relevant information. This dual objective is analogous to the need for both exploration and exploitation in reinforcement learning [31].

Bayesian Learning Bayesian learning [32] aims to obtain a posterior distribution, capturing more information such as prediction uncertainty than a point estimate. It employs Bayes' rule, where the posterior distribution is determined by the likelihood and prior distribution. However, calculating the marginal distribution, which requires the integration of joint likelihood, is computationally expensive. Consequently, Bayesian deep learning focuses on scalable approximate inference methods, such as Variational Inference [33, 34], Laplace approximation [35], and stochastic gradient Langevin dynamics [36]. Despite its popularity and powerful applications, Bayesian learning may not be the most suitable choice in changing environments. Methods like the Dirichlet process [37], a Bayesian nonparametric model for clustering and density estimation, still require data generated from a stationary distribution. To address issues in environments with arbitrary or adversarial changes, Ding et al. [38] proposed a recursive update that mixes with a uniform distribution at every step, along with theoretical performance guarantees. This recursive update inspires our algorithm due to its similarity to the changing environments and Adaptive CL.

Expert learning Expert learning, a sub-field of online learning, focuses on sequentially combining advice from multiple experts to make predictions [39, 40]. In static environments, Exponential Weights (EW) is a popular no-regret strategy, where the average loss converges to that of the best expert in hindsight [41–43]. In the more challenging non-stationary environments, the Fixed-Share algorithm achieves no-regret compared to the non-stationary sequence of optimal experts in hindsight by maintaining each mixing weight above a certain level [44]. In recurring non-stationary environments, Bousquet and Warmuth [45] proposed an efficient algorithm called Mixing-Past-Posteriors (MPP). Building on this work, Koolen et al. [46] provides a Bayesian interpretation of MPP and improves the regret bound with a modified mixing scenario. Although our setting is distinct from expert learning, our algorithm draws inspiration from Fixed-Share and MPP to track a changing environment by recalling knowledge, namely mixing with knowledge.

2 Adaptive CL

2.1 Problem formulation

Many recent works [9–11] have extended traditional Continual Learning (CL) to the task-free CL setting, where task boundaries and identities are unknown during training. However, these approaches, which learn from a training set and evaluate on a separate test set, do not fully reflect human learning capabilities of swift adaptation to previously encountered tasks and continuous knowledge refinement.

To better emulate human cognition, we introduce a novel problem setting, Adaptive CL, characterized by an online, recurring task environment without explicit task boundaries or identities. This setting poses challenges in rapidly recognizing, adapting, and refining knowledge in response to changes in the task distribution. These abilities are essential for improved performance and realistic CL, closely resembling human learning capabilities.

We assume a sequential data stream $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ for time $t = 1, \dots, T$. The learner is asked to predict the label $\hat{y}_t$ with input x_t based on historical data, $\{x_i, y_i\}_{i=1}^{t-1}$. In Adaptive CL, (x_t, y_t) independently follows unknown distribution $\mathcal{D}_t$, where the sequence of distributions $\mathcal{D}_1, \dots, \mathcal{D}_T$ consists of m_T distinct types of distributions and has $k_T - 1$ change points. namely

$$k_T \triangleq 1 + \sum_{t=1}^{T-1} \mathbb{1}(\mathcal{D}_{t+1} \neq \mathcal{D}_t) < T, \quad m_T \triangleq \text{Card}(\{\mathcal{D}_t\}_{t=1}^T) < k_T,$$

where $\text{Card}(\cdot)$ denotes the set cardinality. For simplicity, we omit the subscript T in m_T and k_T, and assume the m **modes**, namely distinct distributions, are $\{\tilde{\mathcal{D}}_j\}_{j=1}^m$, which means $\{\tilde{\mathcal{D}}_t\}_{t=1}^T = \{\mathcal{D}_t\}_{t=1}^T$. In Adaptive CL, our objectives are two-fold: 1) achieve a small cumulative loss by enabling the learner to swiftly adapt to previously learned tasks, and 2) ensure that the learned knowledge converges to the underlying ground truth.

2.2 Questions in mathematical form

Swift adaptation to previously learned tasks and knowledge refinement are crucial for effectively navigating dynamic environments and making accurate predictions in the face of changing and recurring tasks. To better quantify these objectives, we mathematically formulate them as two key questions:

Question 2.1 (Regret Bound)**.** Given a model class $\mathcal{M} \triangleq \{M(\,\cdot\,;\theta) : \mathcal{X} \mapsto \mathcal{Y}, \theta \in \Theta\}$, what is the *optimal upper bound* for the cumulative expected regret with respect to the best competitors from hindsight? The cumulative expected regret for a randomized algorithm $\mathcal{A}$ is defined as

$$\mathbb{E}\left[\text{Regret}_T\right] \triangleq \sum_{t=1}^{T} \mathbb{E}\left[l_t^{\mathcal{A}}\right] - \sum_{t=1}^{T} \min_{\theta \in \Theta} \mathbb{E}\left[l_t(\theta)\right],$$

where $l_t(\theta) \triangleq L\left(M(x_t;\theta), y_t\right)$ for a given loss function L, and $l_t^{\mathcal{A}} \triangleq \mathbb{E}_{\theta \sim \pi_t(\mathcal{A})}[l_t(\theta)]$ with $\pi_t(\mathcal{A})$, a distribution over Θ produced by algorithm $\mathcal{A}$.

A small regret bound in Question 2.1 implicitly guarantees the swift adaptation to previously learned tasks, which is essential for performance improvement. Otherwise, the learner will learn from scratch for every recurring task, which is sub-optimal. To better understand knowledge, we propose the following question regarding convergence. In addition, we denote the frequency of mode $\tilde{\mathcal{D}}_j$ as $\text{freq}_{T,j} \triangleq \sum_{t=1}^T \mathbb{1}(\mathcal{D}_t = \tilde{\mathcal{D}}_j)/T$, and $[n] \triangleq \{1, \dots, n\}$ for any $n \in \mathbb{N}$.

Question 2.2 (Knowledge Convergence)**.** Will the knowledge converge to the ground truth? The convergence can be characterized by the following two conditions. Suppose algorithm $\mathcal{A}$ maintains a density $g_t(\theta)$ over Θ, which represents the learned knowledge, at each time $t \in [T]$. We say the knowledge mass *outside the ground truth vanishes*, if

$$\lim_{\varepsilon \to 0} \lim_{T \to \infty} \mathbb{E}\left[\int_{\theta: d(\theta, \cup_{t=1}^{T} \mathcal{C}_t) \geq \varepsilon} g_T(\theta) d\theta\right] = 0,$$

where the set of minimizers $\mathcal{C}_t \triangleq \arg\min_{\theta \in \Theta} \mathbb{E}[l_t(\theta)]$, and set distance $d(\theta, A) \triangleq \min_{a \in A} \|\theta - a\|_2$ for any $A \subseteq \Theta$. Moreover, suppose $\text{freq}_{T,j}$, the frequency of j-th mode $\tilde{\mathcal{D}}_j$, converges to q_j for $j \in [m]$ with $\sum_{j \in [m]} q_j = 1$. We say the knowledge *converges* to the ground truth, if for any $j \in [m]$,

$$\lim_{\varepsilon \to 0} \lim_{T \to \infty} \mathbb{E}\left[\int_{\theta: d(\theta, \mathcal{B}_j) < \varepsilon} g_T(\theta) d\theta\right] = q_j,$$

where mode minimizers $\mathcal{B}_j \triangleq \arg\min_{\theta \in \Theta} \mathbb{E}_{(x,y) \sim \tilde{\mathcal{D}}_j} \left[L(M(x;\theta), y)\right]$ and $\mathcal{B}_1, \ldots, \mathcal{B}_m$ are disjoint.

Knowledge convergence, as defined in Question 2.2, ensures the knowledge converges to the underlying ground truth, which captures knowledge refinement. To the best of our knowledge, existing literature has not fully addressed one crucial aspect of human learning, where the learner should swiftly adapt to previously learned tasks. This feature is implicitly guaranteed by the regret bound formulated in Question 2.1. Additionally, the understanding of knowledge refinement has been limited in the literature. Our work, through Question 2.2, aims to contribute to a deeper understanding of knowledge convergence and its role in Adaptive CL.

3 LEARN algorithm

In this section, we introduce LEARN (Lifelong Exploration, recAll, and Refinement of kNowledge), a novel algorithm designed to address the challenges of adaptive CL through exploration, knowledge recall, and refinement. This approach facilitates swift adaptation and ongoing consolidation of knowledge. The intuition and detailed explanation of LEARN can be found in Section 3.1, while theoretical guarantees demonstrating its effectiveness are provided in Section 3.2. In Section 3.3, we discuss the scalable implementation of LEARN, highlighting its applicability in deep learning.

3.1 Algorithm description

The LEARN algorithm, as shown in Algorithm 1, consists of two main components: fast and slow learners. The fast learner absorbs new data in the adaptive CL environment through exploration, using tempered Bayesian updates [47–49]. The slow learner consolidates previously learned information, laying the foundation for swift recognition and adaptation to recurring tasks for later knowledge recall. The LEARN operates in three steps: 1) *exploration*, 2) *recall*, and 3) *refinement*. During exploration, the algorithm processes new data with the fast learner. In the recall stage, the fast learner recollects stored knowledge through mixing with the slow learner, facilitating rapid adaptation to previously learned tasks. Finally, in the refinement stage, the information from the fast learner is integrated into the slow learner using a mixing process to enhance the quality of the stored information.

When receiving the input x_t at time t, the agent randomly samples $\hat{\theta}_t$ from the fast learner f_{t-1} and provides prediction $\hat{y}_t = M(x_t, \hat{\theta}_t)$. Upon receiving the true label y_t, the fast learner is updated in the exploration stage, Line 5, using the tempered Bayesian update with temperature η:

$$\tilde{f}_t(\theta) \propto f_{t-1}(\theta) \exp\{-\eta l_t(\theta)\}.$$

While for stationary data, this update leads the fast learner to converge exponentially fast to the point mass on the minimizer, it is an undesirable feature in non-stationary environments due to the long time required to increase the exponentially small probability on the new minimizer. To address this, in the recall stage, the fast learner recalls the slow learner g_{t-1} with recalling ratio $\alpha_t \in [0, 1]$ (Line 6) as:

$$f_t(\theta) \leftarrow (1 - \alpha_t)\tilde{f}_t(\theta) + \alpha_t g_{t-1}(\theta).$$

This mixing step enables the agent to swiftly adapt to previously learned tasks, as the mass of g_{t-1} near the corresponding minimizer is relatively large. In the refinement stage (Line 7), the slow learner g_{t-1} is consolidated with fast learner $\tilde{f}_t$ using a learning rate $\gamma_t \in [0, 1]$:

$$g_t(\theta) \leftarrow g_{t-1}(\theta) - \gamma_t \left\{g_{t-1}(\theta) - \tilde{f}_t(\theta)\right\}.$$

The exploration, recall, and refinement stages of LEARN collectively promote rapid adaptation and enhanced performance on previously learned tasks. To further elucidate the adaptability and knowledge convergence, we will delve into its theoretical underpinnings in the subsequent analysis.

Algorithm 1 LEARN: Lifelong Exploration, recAll, and Refinement of kNowledge

```
Input  Model class $\mathcal{M} = \{M(\cdot;\theta) : \theta \in \Theta\}$, data $\{(x_t, y_t)\}_{t=1}^T$, mixing $\{\alpha_t\}_{t=1}^T$, forgetting $\{\gamma_t\}_{t=1}^T$, step size $\eta > 0$, loss function $L : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_{\geq 0}$.
Output  Fast learner $\tilde{f}_T$, slow learner $g_T$.
Initialization: $f_0(\theta) = g_0(\theta) = 1/\mathrm{Vol}(\Theta)$.
for $t = 1 \to T$ do
    Receive $x_t$, randomly sample $\hat{\theta}_t$ from density $f_{t-1}$, and predict $\hat{y}_t = M(x_t; \hat{\theta}_t)$.
    Receive $y_t$ and corresponding loss $l_t(\theta) \triangleq L(M(x_t;\theta), y_t)$.
    Exploration: $\tilde{f}_t(\theta) \leftarrow f_{t-1}(\theta) \exp\{-\eta l_t(\theta)\}$, and normalize $\tilde{f}_t$.
    Recall: $f_t(\theta) \leftarrow (1-\alpha_t)\tilde{f}_t(\theta) + \alpha_t g_{t-1}(\theta)$.
    Refinement: $g_t(\theta) \leftarrow (1-\gamma_t) g_{t-1}(\theta) + \gamma_t \tilde{f}_t(\theta)$.
```

3.2 Theoretical analysis and insights into the adaptiveness

In our theoretical analysis, we first tackle Question 2.1 by providing a regret upper bound for Algorithm 1 in Proposition 3.1. The technical details are included in Appendix.

Proposition 3.1. *Assume set $\Theta \subseteq \mathbb{R}^d$ is compact with $\sup_{\theta\in\Theta} \|\theta\|_2 \leq D$, and $|l_t(\theta) - l_t(\theta')| \leq Z_t \|\theta - \theta'\|_2$ for all $\theta, \theta' \in \Theta$, with $\mathbb{E}[Z_t^2] \leq v^2$. Then there exists $\eta_{opt} > 0$, stated in the Appendix, such that Algorithm 1 with $\alpha_t = k/T$ and $\gamma_t = 1/t$ yields an expected cumulative regret*

$$\begin{aligned}\mathbb{E}\left[\mathit{Regret}_T\right] \leq & Dv\sqrt{2T\left(md\log\frac{DvT}{2} + 2k\log\frac{T}{k} + k\log mk + md\right)} + 1 \\ = & O\left(Dv\sqrt{T}\sqrt{md\log\{DvT\} + k\log\{mT\}}\right).\end{aligned} \tag{1}$$

Leveraging adaptation to reduce dimensionality costs. In the upper bound, we observe two distinct sources of loss. Excluding the shared $Dv\sqrt{T}$, the first term, $\sqrt{md\log\{DvT\}}$, is dimension-related and signifies the cost of learning a new distribution, occurring m times. This dimension-related aspect is particularly crucial in deep learning, where large dimensions are commonplace. The fact that it is not linked to k indirectly demonstrates the rapid adaptation and knowledge refinement capabilities of LEARN. The second term, $\sqrt{k\log mT}$, is dimension-free and encapsulates the information needed to identify task boundaries and the cost associated with retaining the current distribution. This component has also been recognized and explored in expert learning literature [46, 50].

Next, we turn our attention to the knowledge refinement, specifically the convergence addressed in Question 2.2, by presenting Proposition 3.2. This crucial result demonstrates that the knowledge in Algorithm 1 indeed attains the desired convergence properties.

Proposition 3.2 (Convergence)**.** *Under the assumptions in Proposition 3.1. Suppose $\{\mathbb{E}[l_t]\}_{t=1}^T$ is uniformly strict, namely for any $\varepsilon > 0$, there exists $\delta > 0$ such that,*

$$\min_{1\leq t\leq T} \inf_{\theta: d(\theta, \mathcal{C}_t)\geq\varepsilon} \{\mathbb{E}[l_t](\theta) - \min_{\theta'} \mathbb{E}[l_t](\theta')\} \geq \delta.$$

If $k = o(T/\log T)$, then there exists $\eta_{opt,T}$ such that Algorithm 1 with $\alpha_t = k/T$ and $\gamma_t = 1/t$ has the following properties:

1. *For any $\varepsilon > 0$,*

$$\lim_{T\to\infty} \mathbb{E} \int_{\theta\in\Theta: d(\theta, \cup_t \mathcal{C}_t)\geq\varepsilon} g_T(\theta) = 0,$$

   where $\mathcal{C}_t \triangleq \arg\min_{\theta\in\Theta} \mathbb{E}[l_t(\theta)]$.

2. *If further assume for $\lim_{T\to\infty} \mathit{freq}_{T,j} = q_j$ and the minimizer $\{\mathcal{B}_j\}_{j=1}^m$ are disjoint. Then*

$$\lim_{\varepsilon\to 0} \lim_{T\to\infty} \mathbb{E} \int_{\theta\in\Theta: d(\theta, \mathcal{B}_j)\leq\varepsilon} g_T(\theta) = q_j,$$

   where $\mathcal{B}_j \triangleq \arg\min_{\theta\in\Theta} \mathbb{E}_{(x,y)\sim\tilde{\mathcal{D}}_j} \left[L(M(x;\theta), y)\right]$.

From black-box to knowledge building. The convergence result presented in Proposition 3.2 provides a mathematical insight into knowledge building, unlike many existing heuristic black-box CL approaches. Our analysis illuminates the core mechanisms that underpin the adaptive capabilities of the LEARN algorithm, fostering a comprehensive understanding of its inner workings.

In summary, Propositions 3.1 and 3.2 address Questions 2.1 and 2.2, respectively. LEARN effectively adapts to previously learned tasks and refines its knowledge base, exhibiting key aspects of human learning and making it suitable for various real-world applications. However, Algorithm 1 may not be scalable for large-scale deep learning tasks due to its requirement for density integration. To tackle this, the following subsection introduces an approximation of LEARN that employs an efficient approximation method, enhancing its scalability and compatibility with deep learning applications, thereby extending its applicability.

3.3 Scalable implementation

In the previous subsection, we introduced LEARN in Algorithm 1 and provided theoretical guarantees. However, this approach faces scalability challenges in large-scale deep learning tasks. To address this issue, we present Scalable LEARN in Algorithm 2, an efficient approximation using Gaussian Mixture Models (GMMs). While Variational Inference (VI) [33] is a popular technique for approximating target distributions in deep learning literature, it is not well-suited for our problem setting due to the recursive form in Algorithm 1. GMM, on the other hand, offers a more straightforward and effective solution while preserving the core properties and adaptability of the LEARN algorithm. The detailed technical derivation is included in Appendix.

Algorithm 2 Scalable LEARN

```
Input  Model class $\mathcal{M} = \{M(\cdot;\theta) : \theta \in \Theta\}$, data $\{(x_t, y_t)\}_{t=1}^T$, mixing $\{\alpha_t\}_{t=1}^T$, step size $\eta > 0$, variance $\sigma^2$, patience $\tau \in [0,1]$, $Q > 0$, loss function $L : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_{\geq 0}$.
Output Knowledge $\mathcal{G}_T = \{(r_{T,i}, \beta_{T,i}) \in [0,1] \times \mathbb{R}^d\}_{i=1}^{m_T}$ of tuples of slow weight and parameter.
Initialization: fast learner $\theta_0 \sim \mathrm{Unif}(\Theta)$, predictive weight $w_{0,0} = 1$, cache weight $r_{0,0} = 1$, slow learner $\mathcal{G}_0 = \emptyset$, patience $q_0 = 0$.
for $t = 1 \to T$ do
    Receive $x_t$ and predict $\hat{y}_t = w_{t-1,0} M(x_t; \theta_{t-1}) + \sum_{i=1}^{m_{t-1}} w_{t-1,i} M(x_t; \beta_{t-1,i})$.
    Receive $y_t$ and corresponding loss $l_t(\theta) \triangleq L(M(x_t;\theta), y_t)$.
    Exploration of fast learner: $\theta_t \leftarrow \theta_{t-1} - \eta\sigma^2 \nabla l_t(\theta_{t-1})$.
    Exploration and knowledge recall for adaptation, and then normalize $w_{t,i}$:
        $w_{t,i} \leftarrow \{(1-\alpha) w_{t-1,i} + \alpha r_{t-1,i}\} \exp\{-\eta l_t(\beta_{t-1,i})\}, \; (i = 0, \dots, m_{t-1}, \beta_{t-1,0} \triangleq \theta_{t-1})$
    Refinement of knowledge: for $i \in [m_{t-1}]$
        $r_{t,i} \leftarrow r_{t-1,i} - \frac{1}{t}(r_{t-1,i} - w_{t,i}), \quad \beta_{t,i} \leftarrow \beta_{t-1,i} - \eta\sigma^2 \frac{w_{t,i}}{t r_{t,i}} \nabla l_t(\beta_{t-1,i})$.
    Update patience: $q_t \leftarrow q_{t-1} + \max\{0, w_{t,0} - \tau\}$
    if patience $q_t > Q$ then
        Consolidate knowledge with cache, and initialize cache:
            $\mathcal{G}_t \leftarrow \mathcal{G}_{t-1} \cup \{(r_{t,0}, \theta_t)\}. \quad q_t = r_{t,0} = 0$
```

In Algorithm 1, we focus on two densities: the fast learner $\tilde{f}_t$ and the slow learner g_t. Assume there are m_t Gaussian models, $\mathcal{N}(\beta_{t,i}, \sigma^2 I_d)$, constituting our knowledge base, and an additional Gaussian model, $\mathcal{N}(\theta_t, \sigma^2 I_d)$, for exploration. We approximate the fast and slow learner as the weighted averages of these $m_t + 1$ Gaussian models: $(\beta_{t,0} \triangleq \theta_t)$

$$\tilde{f}_t(\theta) \approx \sum_{i=0}^{m_t} w_{t,i} \mathcal{N}(\beta_{t,i}, \sigma^2 I_d), \quad g_t(\theta) \approx \sum_{i=0}^{m_t} r_{t,i} \mathcal{N}(\beta_{t,i}, \sigma^2 I_d),$$

where $\{w_{t,i}\}_{i=1}^{m_t}$ denote predictive mixing weights, $\{r_{t,i}\}_{i=1}^{m_t}$ represent slow weights, and $w_{t,0}$ and $r_{t,0}$ denote the cache weights to be consolidated. For simplicity, we only consider the first-order Taylor expansion of loss l_t, implying that the variance σ^2 remains unchanged. By substituting the approximation into the update rules in Algorithm 1, we obtain the weights update stated in Algorithm 2, and updates for θ_t and $\beta_{t,i}$ as:

$$\theta_{t+1} \leftarrow \theta_t - \eta\sigma^2 \nabla l_{t+1}(\theta_t), \quad \beta_{t+1,i} \leftarrow \beta_{t,i} - \eta\sigma^2 \frac{w_{t+1,i}}{\sum_{\tau=1}^{t+1} w_{\tau,i}} \nabla l_{t+1}(\beta_{t,i}).$$

In Algorithm 2, the fast learner is first updated using gradient descent (Line 5). After this update, predictive weights facilitate swift adaptation to prior tasks by recalling knowledge and exploring; this is achieved by mixing with slow weights and multiplying by their corresponding performances in Line 6. This process ensures that the agent remains responsive to new information.

In the context of knowledge consolidation, slow weights are first updated in Line 7. New information is selectively consolidated into the knowledge by applying gradient descent with varying step sizes (Line 7). These step sizes are determined by the relevance to the current data, namely the ratio of the predictive weight to the sum of historical predictive weights. This method enables the knowledge to refine by absorbing different amounts of current data while preventing forgetting.

In order to detect new modes, we monitor the patience, which is the sum of cache predictive weights $\max\{w_{t,0} - \tau, 0\}$ (Lines 8 to 10). When the patience surpasses a predetermined threshold Q, the current cache weight and fast learner are consolidated into the slow learner as a new component. This step ensures that the algorithm effectively responds to any new modes.

4 Experimental evaluation

We conduct extensive experiments to evaluate the performance, the ability to adapt to learned tasks, and the knowledge quality. Recall that an Adaptive CL scenario consists of a data stream from an online, non-stationary environment with potentially recurring tasks and unknown task boundaries or identities. To emphasize the challenge of the problem, we create multiple tasks with distinct original labels, which are then re-labeled within the same label region–otherwise, the task boundaries and identity could be inferred directly from the labels. In the following experiments, each data point is presented only once with batch size 5.

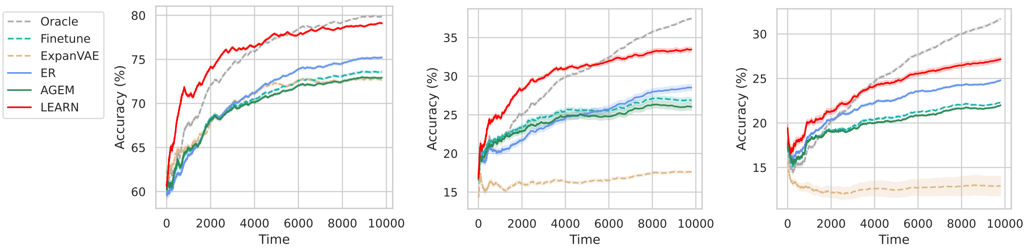
Datasets. CIFAR10 [51] consists of color images in 10 classes, with 6000 images per class. We create 5 tasks from CIFAR10 by splitting the dataset into 5 subsets according to labels $(0/1, 2/3, \ldots, 8/9)$, and then convert the labels in the region of $0/1$ by taking modulus with respect to 2, which brings the challenges of inferring task boundaries and identities. Each task is randomly split into 20 segments with 500 data per segment. By shuffling and combining all the 100 segments, we obtain the **Adaptive CIFAR10 scenario**. **CIFAR100** [51] consists of color images in 100 classes containing 600 images each. Like before, we create 10 tasks from CIFAR100 by splitting it according to labels so that there are 10 classes per task. We then obtain the **Adaptive CIFAR100 scenario** with 10 segments per task, in a way similar to CIFAR10. **Mini-ImageNet** [52] contains 100 classes, and we obtain the **Adaptive Mini-ImageNet scenario** similar to the Adaptive CIFAR100.

Compared methods. Except for our method, LEARN, we further evaluate the following methods, where * indicates unrealistic baselines: 1) **Finetune** with a neural network naively trained on the new data. 2) **Oracle*** as the performance upper bound consisting of multiple models, where one corresponds to one task, with known task identities during training and testing. 3) **ExpVAE*** (Expansion+VAE) consisting of (classifier, generator) tuples which is popular in dynamic expansion with mixture models in Task-free CL [10, 13, 30]. We assume the task identity is known during training. However, during the prediction stage, the task identity must be inferred by the generators, Variational Autoencoder (VAE) [53]. 4) **ER** (Experience Replay) [54] with reservoir sampling [55] guaranteeing the past data uniformly stored in the buffer. When training on a new batch, a replayed batch sampled from the buffer is combined with the new batch, implicitly alleviating forgetting. 5) **A-GEM** (Averaged Gradient Episodic Memory) [18] which stores samples in memory, and projects the gradient on current data onto the orthogonal space of the one on replayed data. The implementation details such as network architecture and hyperparameters are included in Appendix.

Table 1: Comparison of Average Accuracy (%) (mean ± se) from 10 runs.

Method	CIFAR10	CIFAR100	Mini-ImageNet
Oracle*	79.90 ± 0.06	37.45 ± 0.10	31.69 ± 0.11
ExpVAE*	72.69 ± 0.14	17.61 ± 0.14	12.93 ± 1.08
Finetune	73.58 ± 0.11	26.89 ± 0.32	22.35 ± 0.03
ER	75.22 ± 0.10	28.53 ± 0.39	24.81 ± 0.09
A-GEM	72.92 ± 0.11	26.04 ± 0.45	22.00 ± 0.07
LEARN	$\mathbf{79.11 \pm 0.11}$	$\mathbf{33.46 \pm 0.28}$	$\mathbf{27.16 \pm 0.25}$

Metrics. We consider three metrics: 1) **Average Accuracy**: the cumulative accuracy divided by the total time. 2) **Knowledge Accuracy**: the mean of test accuracy over all tasks. 3) **Adaptiveness**: the weighted average of accuracy, where the weights decay geometrically with factor $\gamma \in [0, 1]$ and



(a) CIFAR10 (b) CIFAR100 (c) Mini-ImageNet

Figure 2: The running average accuracy of all compared methods on the three Adaptive scenarios from 10 runs. The dashed line indicates non-Adaptive CL methods with task identity information.

reinitialize whenever the task changes, detailed in Appendix. Larger adaptiveness means a faster speed to recall related information.

Results. As illustrated in Figure 2 and Table 1, the average accuracy of LEARN markedly surpasses that of competing methods across all scenarios. Amongst the methods, ER is a simple but strong comparator. Although ExpVAE utilizes the information of task identities in training, the dependency on VAE to recognize current task in the test stage significantly harm the performance due to the complexity of training VAE. Interestingly, as seen in Figure 2, LEARN outperforms Oracle in the initial stage because each model is trained separately, without knowledge transfer during initialization.

Table 2: Comparisons of Knowledge Accuracy (%) (mean ± se) from 10 runs.

Method	CIFAR10	CIFAR100	Mini-ImageNet
Oracle*	82.64 ± 2.98	44.55 ± 1.29	39.42 ± 1.40
ExpVAE*	82.40 ± 3.03	44.21 ± 1.32	39.71 ± 1.35
Finetune	60.65 ± 2.78	12.11 ± 1.57	14.32 ± 2.05
ER	74.47 ± 2.47	18.27 ± 1.43	16.49 ± 2.14
A-GEM	58.34 ± 2.97	12.08 ± 1.55	14.29 ± 2.06
LEARN	$\mathbf{78.08 \pm 2.53}$	$\mathbf{29.28 \pm 2.91}$	$\mathbf{29.28 \pm 1.53}$

Table 2 measures the mean of the test accuracy over all tasks, representing the quality of knowledge refinement. The refined knowledge in LEARN, namely a mixture of models, is significantly better than competing methods. Table 3 measures the Adaptiveness, defined in Metrics, As shown in the table, LEARN has the largest adaptiveness, showing the ability to adapt to learned tasks more efficiently. It is worth noting that the competing CL approaches in the literature were not designed and optimized for the Adaptive CL scenario, leading to less satisfactory performance. A more comprehensive discussion of the experimental results will be provided in the Appendix.

Table 3: Comparison of Adaptiveness (mean±se) with $\gamma = 0.99$ from 10 runs in 10^{-2} scale.

Method	CIFAR10	CIFAR100	Mini-ImageNet
Oracle*	79.75 ± 0.06	37.05 ± 0.09	31.54 ± 0.11
ExpVAE*	72.76 ± 0.16	17.36 ± 0.13	12.90 ± 1.07
Finetune	72.62 ± 0.11	24.90 ± 0.30	21.07 ± 0.02
ER	74.64 ± 0.11	26.87 ± 0.36	23.62 ± 0.09
A-GEM	71.86 ± 0.13	24.11 ± 0.41	20.72 ± 0.07
LEARN	$\mathbf{78.04 \pm 0.14}$	$\mathbf{31.81 \pm 0.32}$	$\mathbf{26.18 \pm 0.26}$

5 Conclusion

In this work, we proposed a realistic and challenging problem, Adaptive CL, and two mathematically defined characteristics: performance and knowledge quality. To address the problem, we propose a unified LEARN algorithm that simultaneously recognizes the current task, recalls related information, and refines knowledge. The theoretical analysis of LEARN guarantees near-optimal performance and asymptotically consistent knowledge. To be efficient in deep learning, we propose a scalable implementation. Experimental results show that LEARN significantly surpasses the baseline methods in multiple aspects. The **Appendix** contains additional details on the implementation details, more extensive ablation studies, and all the technical proofs. We do not envision any negative social impact of the developed approach.

Limitations: While our work provides a theoretical foundation, there are several limitations worth further investigation. Future studies could 1) examine more efficient ways to recall learned knowledge such as context-dependent mixing weight, 2) investigate the second-order expansion of the loss l_t and updatable variance in deriving Algorithm 2, and 3) extend the current supervised settings to semi-supervised, unsupervised, and reinforcement settings.

References

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

[2] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[3] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

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

[5] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

[6] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[7] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

[8] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

[9] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019.

[10] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. *arXiv preprint arXiv:2001.00689*, 2020.

[11] Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. *Advances in Neural Information Processing Systems*, 34: 29193–29205, 2021.

[12] Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems*, 34:16131–16144, 2021.

[13] Fei Ye and Adrian G Bors. Learning an evolved mixture model for task-free continual learning. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1936–1940. IEEE, 2022.

[14] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017.

[15] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. *Advances in neural information processing systems*, 30, 2017.

[16] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[17] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[18] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[19] Yunhui Guo, Mingrui Liu, Tianbao Yang, and Tajana Rosing. Improved schemes for episodic memory-based lifelong learning. *Advances in Neural Information Processing Systems*, 33: 1023–1035, 2020.

[20] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–82, 2018.

[21] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557. PMLR, 2018.

[22] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[23] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3375, 2017.

[24] Haruka Asanuma, Shiro Takagi, Yoshihiro Nagano, Yuki Yoshida, Yasuhiko Igarashi, and Masato Okada. Statistical mechanical analysis of catastrophic forgetting in continual learning with teacher and student networks. *Journal of the Physical Society of Japan*, 90(10):104001, 2021.

[25] Sen Lin, Peizhong Ju, Yingbin Liang, and Ness Shroff. Theory on forgetting and generalization of continual learning. *arXiv preprint arXiv:2302.05836*, 2023.

[26] Xiaofan Yu, Yunhui Guo, Sicun Gao, and Tajana Rosing. Scale: Online self-supervised lifelong learning without prior knowledge. *arXiv preprint arXiv:2208.11266*, 2022.

[27] Alessandro Achille, Tom Eccles, Loic Matthey, Chris Burgess, Nicholas Watters, Alexander Lerchner, and Irina Higgins. Life-long disentangled representation learning with cross-domain latent homologies. *Advances in Neural Information Processing Systems*, 31, 2018.

[28] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

[29] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems 32*, pages 11849–11860. Curran Associates, Inc., 2019.

[30] Fei Ye and Adrian Gheorghe Bors. Task-free continual learning via online discrepancy distance learning. In *Advances in Neural Information Processing Systems (NeurIPS)*. MIT Press, 2022.

[31] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[32] William M Bolstad and James M Curran. *Introduction to Bayesian statistics*. John Wiley & Sons, 2016.

[33] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

[34] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[35] David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[36] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688, 2011.

[37] Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.

[38] Jie Ding, Jiawei Zhou, and Vahid Tarokh. Asymptotically optimal prediction for time-varying data generating processes. *IEEE Transactions on Information Theory*, 65(5):3034–3067, 2018.

[39] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[40] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[41] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[42] Volodimir G Vovk. Aggregating strategies. *Proc. of Computational Learning Theory, 1990*, 1990.

[43] Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

[44] Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine learning*, 32(2): 151–178, 1998.

[45] Olivier Bousquet and Manfred K Warmuth. Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research*, 3(Nov):363–396, 2002.

[46] Wouter M Koolen, Dmitry Adamskiy, and Manfred K Warmuth. Putting bayes to sleep. In *NIPS*, pages 135–143, 2012.

[47] Tim Erven, Wouter M Koolen, Steven Rooij, and Peter Grünwald. Adaptive hedge. *Advances in Neural Information Processing Systems*, 24, 2011.

[48] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[49] Nial Friel and Anthony N Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):589–607, 2008.

[50] James Robinson and Mark Herbster. Improved regret bounds for tracking experts with memory. *Advances in Neural Information Processing Systems*, 34:7625–7636, 2021.

[51] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[52] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[53] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[54] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

[55] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.