Unbiased and Efficient Self-Supervised Incremental Contrastive Learning

Cheng Ji
Beihang University
Beijing, China
jicheng@act.buaa.edu.cn

Jianxin Li
Beihang University
Beijing, China
lijx@act.buaa.edu.cn

Hao Peng
Beihang University
Beijing, China
penghao@act.buaa.edu.cn

Jia Wu
Macquarie University
Sydney, Australia
jia.wu@mq.edu.au

Xingcheng Fu
Beihang University
Beijing, China
fuxc@act.buaa.edu.cn

Qingyun Sun
Beihang University
Beijing, China
sunqy@act.buaa.edu.cn

Philip S. Yu
University of Illinois at Chicago
Chicago, USA
psyu@uic.edu

ABSTRACT

Contrastive Learning (CL) has been proved to be a powerful self-supervised approach for a wide range of domains, including computer vision and graph representation learning. However, the incremental learning issue of CL has rarely been studied, which brings the limitation in applying it to real-world applications. Contrastive learning identifies the samples with the negative ones from the noise distribution that changes in the incremental scenarios. Therefore, only fitting the change of data without noise distribution causes bias, and directly retraining results in low efficiency. To bridge this research gap, we propose a self-supervised *Incremental Contrastive Learning* (ICL) framework consisting of (i) a novel Incremental InfoNCE (NCE-II) loss function by estimating the change of noise distribution for old data to guarantee no bias with respect to the retraining, (ii) a meta-optimization with deep reinforced Learning Rate Learning (LRL) mechanism which can adaptively learn the learning rate according to the status of the training processes and achieve fast convergence which is critical for incremental learning. Theoretically, the proposed ICL is equivalent to retraining, which is based on solid mathematical derivation. In practice, extensive experiments in different domains demonstrate that, without retraining a new model, ICL achieves up to 16.7× training speedup and 16.8× faster convergence with competitive results.

CCS CONCEPTS

• **Computing methodologies → Online learning settings**; • **Information systems → Data streams; Data stream mining**.

KEYWORDS

Contrastive learning, incremental learning, self-supervised learning.

1 INTRODUCTION

Contrastive Learning (CL) is a widely used self-supervised learning approach across a wide range of domains, such as computer vision (CV) [5, 11], neural language processing (NLP) [19, 24, 41], and graph representation learning (GRL) [32, 38, 44]. The main idea of contrastive learning is to make representations of similar samples close and distinct samples far away through a noise contrastive estimation (NCE) [9, 10] loss function with the given noise distribution [41]. Moreover, in real-world application scenarios, online systems are expected to constantly face new data and learn incrementally, which limits the applicability of contrastive learning. The data distribution and noise distribution are incrementally observed and the estimation by NCE is thus biased. Nevertheless, there is little work to study the incremental learning issue of contrastive learning (ICL, Incremental Contrastive Learning).

Incremental learning aims to learn the new data while not forgetting the old data (see the formal definition in Definition 3). It requires the learning model to face the stability-plasticity dilemma [27] with the following three characteristics: **(a) stability for the old data,** which means that the model should contain the ability to remember and update the knowledge of the old data, **(b) plasticity for the new data,** which requires that the model should be able to adapt to the new data, **(c) efficiency of the training process,** which helps the model update quickly in real-world applications, especially in a streaming environment.

Table 1: Comparison of different strategies.

	Stability	Plasticity	Efficiency
Inference	×	×	✓
Fine-tuning	×	×	✓
Retraining	✓	✓	×
ICL (Ours)	✓	✓	✓

[1] ✓ : "unbiased" or "high", × : "biased" or "low".

The major challenge of contrastive learning in the incremental setting is how to estimate the change of the noise distribution. The noise distribution in contrastive learning is used to sample the negative ones and is related to and close to the data distribution [10, 41]. Therefore, as the data change, the noise distribution also changes. Based on the trained model, the old data has been learned through NCE by contrasting the negative samples from the original noise distribution. However, there is a bias in the estimation after giving the new data since the noise distribution used for sampling the negative samples has changed. The existing NCE methods cannot be applied to fit the change (we provide the bias analysis in Section 3). Regardless of the trained model, retraining seems to be the best way due to no bias in the process. However, it has a serious time-consuming problem. Therefore, in this paper, we seek to answer the question: *can contrastive learning approaches be able to estimate the old data through an unbiased NCE w.r.t. retraining, while learning the new data efficiently?*

The naive strategies (e.g., inference, fine-tuning, and retraining) fail to answer this question due to the above three requirements of incremental learning as shown in Table 1:

- **Poor Stability.** Without a carefully designed NCE strategy, both inference and fine-tuning lead to a biased estimate due to the change in the noise distribution. In addition, directly fine-tuning the model on new data causes catastrophic forgetting [26].
- **Weak Plasticity.** Inference is not capable of learning the new data and fine-tuning is over-transferring, ignoring the contribution of old data to the noise distribution. Moreover, bias will continue to accumulate in practical online applications.
- **Low Efficiency.** While retraining a new model with all data does not seem to have the above issues, the time cost is unacceptable in real-world applications.

However, there is little work on incremental learning in CL, except for applying the replay and distillation technique [4, 22]. Moreover, existing incremental learning approaches are hard to use for self-supervised CL because they focus mainly on class-incremental learning [25] and task-incremental learning [34]. Without any information of labels and tasks, self-supervised incremental contrastive learning is still facing a research gap.

Contributions. To this end, we propose an unbiased and efficient self-supervised *Incremental Contrastive Learning* (ICL) framework. First, we design an **I**ncremental **I**nfo**NCE** (**NCE-II**) loss function to fit the change of noise distribution. Furthermore, we accelerate the convergence by a meta-optimization algorithm with a

reinforced Learning Rate Learning (LRL) mechanism. Finally, we conduct thorough experiments on four datasets of CV and GRL to show the efficiency and effectiveness of ICL. The main contributions of this paper are summarized as follows:

- Leveraging a new metric for measuring the change of the noise distribution, we design a novel NCE-II loss to theoretically achieve an unbiased ICL with respect to retraining. To our best knowledge, it is the first attempt to study the issue of unbiased incremental learning in self-supervised CL.
- We propose a meta-optimization algorithm with LRL to adaptively learn the learning rates according to the status of the training process for fast convergence.
- Extensive experiments demonstrate that ICL maintains high efficiency in terms of training time and convergence epoch. In addition, as an unbiased and efficient approach, ICL still achieves competitive results.

2 BACKGROUND AND PROBLEM FORMULATION

We first provide the background and problem formulation of contrastive learning and incremental contrastive learning.

DEFINITION 1 (CONTRASTIVE LEARNING). *Given the set of input data $X := \{x_i \in X\}_{i=1}^N$ of size N and an encoder $\phi(\cdot) : X \to Z$ mapping the inputs from the data space X into the latent space Z, contrastive learning aims to train the encoder with a contrastive loss function $\mathcal{L}$ designed to identify the positive sample x_i^+ for a given x_i.*

In this paper, we consider the most commonly used contrastive learning framework [5, 11, 24, 44] with the **I**nfo**NCE** (denoted as **NCE-I** in this paper) as the loss function $\mathcal{L}$.

DEFINITION 2 (INFONCE (NCE-I)). *Given the set of input data $X := \{x_i \in X\}_{i=1}^N$ and an encoder $\phi(\cdot)$, the InfoNCE (NCE-I) loss is defined as:*

$$\mathcal{L}_i^{I,X} = -\log \frac{f(x_i, x_i^+)}{f(x_i, x_i^+) + K\mathbb{E}_{x_i^- \sim p_n^X} f(x_i, x_i^-)}, \quad (1)$$

where $f(\cdot, \cdot) := \exp(sim(\phi(\cdot), \phi(\cdot))/\tau)$ with $sim(\cdot, \cdot)$ as the similarity measurement and τ as the temperature parameter, x_i^+ represents the positive sample generated by a data augmentation module, x_i^- is one of the negative samples from a noise distribution, and K is a hyperparameter representing the ratio of negative samples to positive samples.

Noise Distribution. The InfoNCE loss follows the noise contrastive estimation (NCE) principle [9, 10], which learns to deduce the properties of data X by comparing the difference to the reference (noise) data Y. The noise data Y is an i.i.d. sample $\{y_1, y_2, \ldots, y_K\}$ from a random variable with noise distribution p_n. In this paper, the negative ones are from the data distribution by a uniform sampling which can be approximated by Monte Carlo sampling [35]. We randomly sample a mini-batch of $K + 1$ inputs and treat the others as the negative samples for each one.

DEFINITION 3 (INCREMENTAL CONTRASTIVE LEARNING). *Given an encoder $\phi(\cdot)$ trained on old data $X := \{x_i \in X\}_{i=1}^N$ with a contrastive learning approach and the new data $\Delta X := \{x_{N+i} \in$*

$X\}_{i=1}^{\Delta N}$ that has not been observed, the incremental contrastive learning aims to refine the encoder for adapting to the new data without forgetting the knowledge of old data, i.e., stability and plasticity.

Noise Distribution Change. As mentioned above, the negative ones in contrastive learning are randomly sampled from the dataset which changes in the incremental setting. That is, the noise distribution changes.

Finally, we denote $X' := \{x_i \in X \cup \Delta X\}$ as all data. For generality, we focus on one incremental step and note that the old data X represent the data used for training the encoder before and preserved by the online system through a memory queue or replay technique. Therefore, the study in this paper can be conveniently implemented to a number of contrastive learning and incremental learning methods.

3 METHODOLOGY

In this section, we propose an unbiased and efficient self-supervised *Incremental Contrastive Learning* (ICL) framework.

3.1 Overall Framework

The proposed ICL framework consists of the following components:

(1) **Augmentation.** Given each input x_i (e.g., an image or a graph), two augmentations $q_1(\cdot|x_i)$ and $q_2(\cdot|x_i)$ are applied to x_i to obtain a positive pair (x_i, x_i^+). For different domains of datasets (i.e., CV and GRL), different augmentation strategies are applied (Section 4.1.3).

(2) **Encoder.** An encoder $\phi(\cdot) : \mathcal{X} \rightarrow \mathcal{Z}$ is used to learn the latent representations for each positive pair (x_i, x_i^+). Specifically, a ResNet-18 [12] and a GCN [15] are used for CV an GRL respectively.

(3) **Incremental InfoNCE (NCE-II).** To resolve the problems mentioned in Section 1, we design a novel loss function, named **NCE-II**, to eliminate the bias caused by the change of noise distribution. Furthermore, the proposed NCE-II maintains equivalence with retraining, and the error bound of final empirical risk tends to zero (Section 3.2).

(4) **Meta-optimization with Learning Rate Learning (LRL).** In order to further improve the efficiency of the training process, which is crucial for incremental learning, we proposed a new meta-learning optimization algorithm with a reinforced learning rate learning mechanism for fast convergence (Section 3.3).

In the following sections, we introduce the proposed objective function (NCE-II) and optimization algorithm (meta-optimization with LRL).

3.2 Objective Function: Unbiased Estimation

We split the final objective function into two parts: one for old data and the other for new data. The new data ΔX can be learned by NCE-I with the noise distribution $p_n^{X'}$:

$$\mathcal{L}_i^{I,X'} = -\log \frac{f(x_i, x_i^+)}{f(x_i, x_i^+) + K\mathbb{E}_{x_i^- \sim p_n^{X'}} f(x_i, x_i^-)}. \qquad (2)$$

However, for stability as discussed in Section 1, InfoNCE loss cannot serve as the objective function for the old data. Given the encoder trained with the old data X, it is necessary to find an optimization method for X, so that the entire training process (including the incremental learning phase) with X is unbiased.

3.2.1 Motivation: Change of Noise Distribution. In order to eliminate the deviation caused by the change of the noise distribution in the learning of the old data, we first propose a metric for measuring the change ratio of the noise distribution.

To explore how noise distribution works in contrastive learning, we first rewrite the InfoNCE loss into the form of softmax-based categorical cross-entropy:

$$\mathcal{L}_i^{I,X} = \sum_{x_j \in X} \mathbb{1}_{i=j} \cdot -\log \left(\frac{f(x_i, x_j^+)}{\sum_{x_k \in X} f(x_i, x_k^+)} \right) \qquad (3)$$

$$= \sum_{x_j \in X} \mathbb{1}_{i=j} \cdot -\log \left(\text{softmax}_X(f(x_i, x_j^+)) \right). \qquad (4)$$

The softmax term in Eq. (4) represents the final prediction result of the model, in which the sum of predicted probabilities for all classes changes with the noise distribution, that is, $\sum_{x_k \in X} f(x_i, x_k^+) \rightarrow \sum_{x_k \in X \cup \Delta X} f(x_i, x_k^+)$. Therefore, we propose to use the ratio of the sum of predicted probabilities for new classes to the one for old classes.

DEFINITION 4 (**CHANGE RATIO OF NOISE DISTRIBUTION**). *Given the noise distribution p_n^X of old data X and $p_n^{\Delta X}$ of new data ΔX, the change ratio is represented by*

$$r_i^{X \rightarrow \Delta X} = \frac{\sum_{x_j \in \Delta X \cup \{x_i\}} f(x_i, x_j^+)}{\sum_{x_k \in X} f(x_i, x_k^+)} \qquad (5)$$

$$= \frac{f(x_i, x_i^+) + K\mathbb{E}_{x_i^- \sim p_n^{\Delta X}} f(x_i, x_i^-)}{f(x_i, x_i^+) + K\mathbb{E}_{x_i^- \sim p_n^X} f(x_i, x_i^-)}. \qquad (6)$$

With the proposed change ratio in Eq. (6), we can measure how much the noise distribution changes. If the new data maintain the same noise distribution as the old data (i.e., $p_n^{\Delta X} = p_n^X$), the change ratio $r_i^{X \rightarrow \Delta X}$ equals 1. The deviation of $r_i^{X \rightarrow \Delta X}$ from 1 reflects the degree of change of noise distribution.

3.2.2 Objective Function: NCE-II Loss. Next, we design a novel contrastive loss function, leveraging the change ratio of the noise distribution.

DEFINITION 5 (**INCREMENTAL INFONCE (NCE-II)**). *Given the old data X and new data ΔX, the loss function of the incremental contrastive learning for old data is defined as*

$$\mathcal{L}_i^{II} = \log \left(\alpha \cdot r_i^{X \rightarrow \Delta X} + (1 - \alpha) \cdot 1 \right) \qquad (7)$$

$$= \log \left(\alpha \cdot \frac{f(x_i, x_i^+) + K\mathbb{E}_{x_i^- \sim p_n^{\Delta X}} f(x_i, x_i^-)}{f(x_i, x_i^+) + K\mathbb{E}_{x_i^- \sim p_n^X} f(x_i, x_i^-)} + (1 - \alpha) \cdot 1 \right), \quad (8)$$

where constant $\alpha = \frac{\Delta N}{N + \Delta N} \in [0, 1)$ represents the growth ratio of the data.

In NCE-II loss, the change ratio for each sample $r_i^{\Delta X \rightarrow X}$ is weighted by a coefficient α, which reflects the increment of the data. Specifically, when there is no new data (i.e., $\alpha = 0$) or the noise distribution remains the same (i.e., $r_{i,\Delta X \rightarrow X} = 1$), the NCE-II equals 0, which is in line with the intuition.

Table 2: Bias of different strategies.

	Old Data	New Data
Inference	$\log r_i^{X \rightarrow X'}$	$\mathcal{L}_i^{I,X'}$
Fine-tuning	$\log r_i^{X \rightarrow X'}$	$\log r_i^{\Delta X \rightarrow X'}$
Retraining	0	0
ICL (Ours)	0	0

3.2.3 No Bias: Equivalence with Retraining. Given NCE-II, the encoder is trained on the old data X with two contrastive loss functions (Eq.(2) and Eq.(8)), the sum of which is equivalent to the loss function of retraining on all data X'.

THEOREM 1. *For old data $x_i \in X$, the NCE-II with new data ΔX plus InfoNCE only with X is equivalent to the one with all data X', i.e., $\mathcal{L}_i^{I,X'} = \mathcal{L}_i^{I,X} + \mathcal{L}_i^{II}$.*

PROOF. The proof is given in the online repositories[1]. □

Finally, the objective function of ICL is defined as

$$\mathcal{L} = \sum_{x_i \in X} \mathcal{L}_i^{II} + \sum_{x_i \in \Delta X} \mathcal{L}_i^{I,X'}, \tag{9}$$

which is an unbiased estimation of both old data and new data w.r.t. retraining. In Table 2, we provide the comparison of bias between different strategies to show the superiority of the proposed method.

3.2.4 Bound Analysis. Although the proposed loss function NCE-II is equal to the difference of InfoNCE with old data and all data, the final empirical risk $\mathcal{R}$ during the entire training process is nevertheless different due to the change of data distribution. Therefore, we provide the bound analysis to guarantee the correctness of the proposed NCE-II.

THEOREM 2. *The difference between the empirical risk of the method with the proposed NCE-II and the retraining with InfoNCE throughout the training process is bounded by $\alpha \mathcal{R}^{old}$, where $\mathcal{R}^{old} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i^{I,old}$ approaches zero as $\mathcal{L}_i^{I,old}$ is minimized after the previous training process on old data X and the growth ratio $\alpha \in [0, 1)$. Then we have $\alpha \mathcal{R}^{old} \rightarrow 0$.*

PROOF. The proof is given in the online repositories. □

3.2.5 Complexity Analysis. Given the number of epochs ΔT in the incremental learning process, the time complexity of the proposed method is $O(\Delta T((2K + 1)N + (K + 1)\Delta N)C)$, where C represents for the time consuming of the encoder and function $f(\cdot, \cdot)$. As the parameters of the encoder have been optimized, the number of epochs is much smaller than the one of the retraining process. We further guarantee it in the next section.

---

[1]Proofs are at https://github.com/RingBDStack/ICL-Incremental-InfoNCE.

3.3 Optimization: Efficient Adaption

In order to further improve the efficiency of the proposed method and ensure that the number of epochs in the incremental learning process is smaller, we next propose a meta-optimization algorithm and a Learning Rate Learning (LRL) mechanism by reinforcement learning (RL) to quickly adapt to the new data.

Formally, the optimization of the encoder is defined as

$$\theta_{t+1} \leftarrow \theta_t - lr * \nabla_\theta \mathcal{L}(\phi(X; \theta_t)), \tag{10}$$

where lr is the learning rate and $\nabla_\theta \mathcal{L}(\phi(X; \theta_t))$ is the gradient in the time step t.

3.3.1 Meta-optimization. For plasticity, we treat the optimization of ICL for new data as transfer learning (the new data may contain new classes or from different domains) and propose a meta-optimization strategy for fast adaption, which is formulated as a form of model agnostic meta-learning (MAML) [8].

We treat the old data X as the support set and the new data ΔX as the query set. In the meta-training stage, we calculate the loss on the support set by Eq.(8) and gain the new parameters θ' in a few gradient descent steps:

$$\theta' = \theta - lr_s * \frac{\partial \sum_{x_i \in X} \mathcal{L}_\theta^{II}(\phi(x_i; \theta))}{\partial \theta}, \tag{11}$$

where lr_s is the learning rate of the meta-training process on the support set X. For fairness, the number of steps is set to $\max\{\lceil(1 - \alpha)/\alpha\rceil, 1\}$ to ensure that the total number of samples trained in one epoch and the size of all data are as equal as possible, under the premise that there is at least one support sample for each query.

In the meta-testing stage, after obtaining the adapted parameters θ', we update the parameters θ on the query set with

$$\theta \leftarrow \theta - lr_q * \frac{\partial \sum_{x_i \in \Delta X} \mathcal{L}_\theta^{I,X'}(\phi(x_i; \theta'))}{\partial \theta}, \tag{12}$$

where lr_q is the learning rate of the meta-testing process on the query set ΔX.

3.3.2 Learning Rate Learning (LRL). The learning rate for gradient descent is commonly set as a hyperparameter and needs to be manually searched for the optimal value. However, it is a challenge to determine the values of two learning rates (lr_q and lr_s) in our proposed meta-optimization. It is (a) unreasonable to simply set the two values to the same or use the previous one because the two learning rates are used for different training processes and data, (b) hard to search for the optimal value due to the large searching space, and (c) the change rules of learning rates should be automatically learned according to the status of the training environment instead of applying the human-designed ones. Therefore, we aim to find a solution that can adaptively control learning rates while speeding up the convergence process.

We proposed a Learning rate Learning (LRL) mechanism to learn the two learning rates (lr_s and lr_q) adaptively according to the current state of the encoder in a reinforcement learning approach [14, 20, 43]. In our reinforcement learning setting, we propose to use a function $\mathcal{F}$ to generate the state s_t which encodes the observation of the training process (the inputs X and the parameters θ_t) at the time step t:

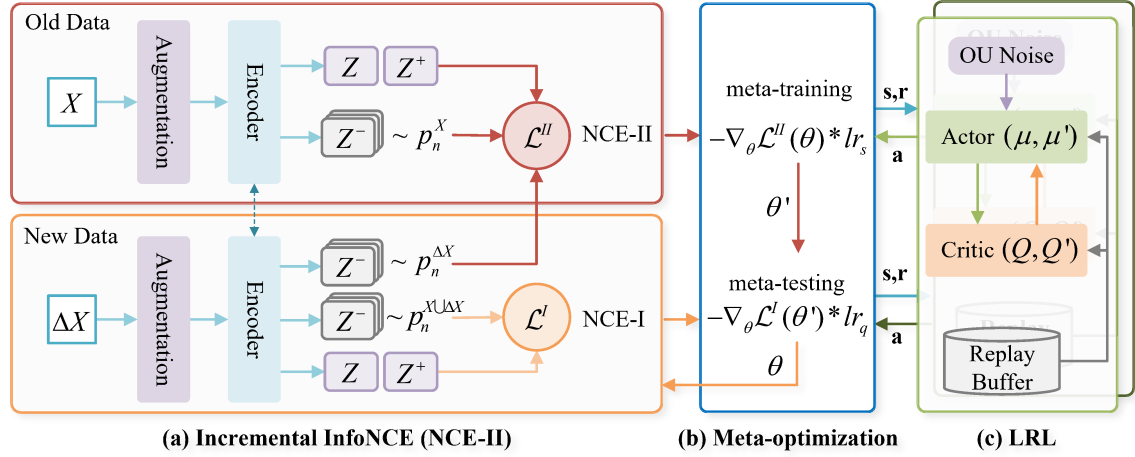$$s_t = \mathcal{F}(X, \theta_t). \tag{13}$$

Figure 1: An overview of ICL. (a) The proposed Incremental InfoNCE (NCE-II) to fit the change of noise distribution p_n. (b) The proposed meta-optimization where the old data X is treated as the support set in the meta-training stage and the new data ΔX is treated as the query set in the meta-testing stage. (c) The proposed Learning Rate Learning (LRL) mechanism which uses the current loss as state and loss decrement as reward to generate the learning rates (lr_s and lr_q) as actions for meta-optimization.

The action a_t is defined as the learned learning rate based on the state s_t and $a_t \in \mathbb{R}$ is a continuous value.

To resolve the continuous action space issue, [43] uses an actor-critic algorithm [36, 39] to generate the learning rate, employing two neural networks: the actor network for learning rate generation and the critic network for criticizing the actions. However, because the inputs of the actor network are ordered and related (i.e., not i.i.d.), the reinforcement learning process is therefore unstable and problematic [28]. Therefore, we propose to use the Deep Deterministic Policy Gradient (DDPG) to learn the two learning rates respectively, which contains the experience replay mechanism to resolve the i.i.d. issue and the separate target network mechanism for a stable training [21].

In general, DDPG networks $\mathcal{D} = \{\mu, Q, \mu', Q'\}$ consist of the (on-line) actor network $\mu(s|\theta^\mu)$ and (online) critic network $Q(s, a|\theta^Q)$, and the target actor network μ' and the target critic network Q' with the same initial weights as the online ones. We first select an action a_t in step t as

$$a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t, \qquad (14)$$

where $\mathcal{N}$ is the Ornstein-Uhlenbeck (OU) process [40] to generate noise for action exploration. Next, the action a_t is executed (updating encoder's parameters with a_t as the learning rate) and the new state s_{t+1} and reward r_t are observed. As the aim of LRL is to accelerate the convergence, we define the reward r_t as the loss decrement:

$$r_t = \mathcal{L}_t - \mathcal{L}_{t+1}. \qquad (15)$$

Finally, we optimize the the critic network using Temporal-Difference (TD) learning with a replay buffer and actor network with the chain rule [21]. The target networks are updated by a momentum mechanism. Please refer to the online repositories for more details.

4 EXPERIMENT

Extensive experiments across two domains, computer vision and graph representation learning, are conducted to demonstrate the high efficiency of the proposed unbiased ICL[2] framework. In addition, ICL also achieves competitive results. We further provide the incremental setting analysis and ablation study.

4.1 Experimental Settings

4.1.1 Datasets. We use four benchmark datasets across computer vision (CV) and graph representation learning (GRL). For CV, we use the following two datasets: (a) ImageNet is a benchmark dataset consisting of around 1.28 million images [6]. In order to reduce the influence of additional factors for an accurate efficiency evaluation, we extract a subset with 2 classes, named ImageNet-2, to manage that the entire training process can be completed on 1 GPU within 6 hours. (b) MNIST is a handwritten digit dataset [16]. Similarly, we extract a subset with 2 classes, named MNIST-2. For GRL, we use the following two datasets: (a) PROTEINS is a bioinformatics dataset with 1,113 molecular graphs [29]. (b) REDDIT is a social network dataset consisting of 2,000 graphs in 2 classes [29]. We use Local Degree Profile (LDP) algorithm [2] to generate node features.

4.1.2 Baselines and ICL Variants. We compare four baselines including the simple strategies (retraining and fine-tuning) and commonly used techniques (replay and distillation).

- **Retraining** uses both old data and new data to train a new encoder. Since the aim of this study is to find an approach for unbiased estimation, we first compare ICL with retraining which serves as the baseline of efficiency and the upper bound of effectiveness.
- **Fine-tuning** updates the parameters of the pre-trained encoder using the new data.

---

[2]Code is available at https://github.com/RingBDStack/ICL-Incremental-InfoNCE.

Table 3: Classification results of running time and convergence epoch on CV datasets. (bold: best; <u>underlined</u>: runner-up)

Dataset Growth Rate Metric	ImageNet-2						MNIST-2						Avg. Rank	
	$\alpha=0.3$		$\alpha=0.5$		$\alpha=0.7$		$\alpha=0.3$		$\alpha=0.5$		$\alpha=0.7$			
	Time	Epoch	Time	Epoch	Time	Epoch	Time	Epoch	Time	Epoch	Time	Epoch	Time	Epoch
Retraining	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	-	-
Fine-tuning	<u>10.7x</u>	3.2x	3.9x	2.0x	2.2x	1.6x	<u>12.3x</u>	<u>3.6x</u>	5.5x	2.6x	<u>3.8x</u>	<u>2.7x</u>	<u>2.3</u>	<u>3.6</u>
Replay	3.4x	2.0x	2.0x	1.6x	<u>3.4x</u>	2.7x	2.8x	2.6x	3.5x	2.5x	3.2x	2.4x	4.3	4.5
Distillation	6.9x	2.9x	2.1x	1.4x	2.2x	1.5x	7.8x	3.0x	4.4x	<u>2.8x</u>	3.1x	2.2x	4.0	4.8
ICL (Ours)	**11.2x**	**16.8x**	**6.2x**	**13.2x**	**5.9x**	**12.7x**	**14.2x**	**10.6x**	**16.7x**	**12.7x**	**6.4x**	**7.1x**	**1.0**	**1.0**
ICL (w/o LRL)	9.6x	14.7x	3.4x	5.3x	1.1x	2.2x	1.0x	1.0x	2.4x	2.5x	1.5x	2.3x	5.2	4.0
ICL (w/o M)	9.2x	15.7x	<u>4.8x</u>	<u>8.1x</u>	1.7x	<u>3.8x</u>	1.1x	1.3x	1.2x	1.5x	1.0x	1.2x	5.3	4.5
ICL (w/o M+LRL)	9.8x	<u>16.4x</u>	1.6x	2.7x	1.0x	1.3x	1.7x	1.7x	1.3x	1.3x	1.2x	1.3x	5.7	5.3

Table 4: Classification results of running time and convergence epoch on GRL datasets. (bold: best; <u>underlined</u>: runner-up)

Dataset Growth Rate Metric	PROTEINS						REDDIT						Avg. Rank	
	$\alpha=0.3$		$\alpha=0.5$		$\alpha=0.7$		$\alpha=0.3$		$\alpha=0.5$		$\alpha=0.7$			
	Time	Epoch	Time	Epoch	Time	Epoch	Time	Epoch	Time	Epoch	Time	Epoch	Time	Epoch
Retraining	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	1.0x	-	-
Fine-tuning	**25.7x**	8.0x	**8.1x**	<u>3.9x</u>	**2.7x**	1.9x	**14.4x**	1.1x	**5.5x**	<u>2.9x</u>	1.2x	0.9x	**1.7**	3.8
Replay	2.9x	2.1x	1.6x	1.0x	1.4x	1.4x	2.9x	2.1x	2.0x	1.4x	1.3x	1.1x	4.8	5.8
Distillation	<u>18.2x</u>	7.3x	5.2x	2.7x	0.9x	0.8x	2.4x	1.5x	2.7x	1.6x	1.1x	0.9x	4.8	5.3
ICL (Ours)	10.1x	<u>10.5x</u>	<u>5.8x</u>	**6.1x**	<u>2.6x</u>	**2.7x**	2.7x	2.5x	<u>2.9x</u>	**3.1x**	2.3x	<u>3.2x</u>	<u>3.0</u>	**1.8**
ICL (w/o LRL)	3.8x	**14.6x**	1.6x	1.7x	1.0x	1.1x	3.4x	<u>3.3x</u>	2.7x	<u>2.9x</u>	**2.6x**	**3.6x**	3.8	<u>3.0</u>
ICL (w/o M)	2.6x	3.6x	2.1x	2.9x	1.4x	<u>2.1x</u>	3.0x	3.2x	1.4x	1.6x	1.2x	1.3x	5.0	3.7
ICL (w/o M+LRL)	2.5x	2.9x	1.9x	2.0x	1.1x	1.2x	<u>5.9x</u>	**5.8x**	2.2x	2.4x	<u>2.5x</u>	2.5x	4.1	4.0

Furthermore, since ICL is under the self-supervised setting which lacks related research, we survey the replay and distillation technique used in incremental learning [3, 4, 7, 22, 33].

- **Replay** uses the previously seen data (partially stored old data) and the new data to train the encoder for avoiding catastrophic forgetting. classes,
- **Distillation** learns a more compact encoder from the old one to prevent over-drift of representations from previous data when learning new ones.

For fairness, we use the exact same network architecture for all methods.

We further introduce the following three variants of ICL to verify the effectiveness of the components (LRL and meta-optimization):

- ICL without LRL mechanism (w/o LRL), i.e., with NCE-II and meta-optimization.
- ICL without meta-optimization (w/o M), i.e., with NCE-II and LRL only generating one learning rate.
- ICL without meta-optimization and LRL mechanism (w/o M+LRL), i.e., only with NCE-II.

4.1.3 Implementation Details. We adopt a two-stage scheme [17, 18, 44]. In the training process, for fairness, all of the methods use the same single encoder to generate the representations. In the testing process, an extra SVM classifier [44] is trained with the fixed embedding for evaluation. In order to evaluate the efficiency of the methods more accurately and reduce the influence of external factors, we conduct each experiment on one single Nvidia V100 32G GPU for 5 times independently. We record the mean value as the final accuracy for each experiment and omit the standard deviations (all deviations are around 0.01). (a) The encoder we have used for CV is ResNet-18 [12] and a two-layer Graph Convolution Network (GCN) [15] with 32 hidden units is used for GRL. (b) We choose {random resizing, 224×224-pixel cropping, random color jittering, random grayscale conversion, random horizontal flip} [11, 42] as the data augmentation for images, and randomly sample one from {random node dropping, random node attribute masking, random subgraph selection} [44] for GRL. (c) The similarity measurement we have used is the cosine similarity function $sim(z_i, z_j) := z_i \cdot z_j / ||z_i|| \cdot ||z_j||$. (d) The state generated by function $\mathcal{F}$ in LRL is defined as the average loss of the current training process on the mini-batch data. (e) A two-layer LSTM with 20 hidden units is used as the actor network and a three-layer neural network (NN) with 10 hidden units is used as the critic network.

4.1.4 Parameters Setting. For common hyper-parameters, the number of negative samples $K = 31$ (i.e., a batch size $bs = 32$), the initial learning rate is searched from $lr \in \{10^{-3}, 10^{-4}, 10^{-5}\}$, the temperature parameters $\tau = 0.1$, and the patience to wait for convergence is 50 epochs (i.e., the process is terminated when the loss no longer drops for 50 epochs). For LRL, the momentum term $m = 10^{-3}$.

4.2 Efficiency Analysis: How fast ICL is?

We first perform the image and graph-level classification on CV and GRL datasets. To simulate different incremental scenarios, we randomly split each dataset into the old data and new data according to the given growth ratio α. As an unbiased approach, the proposed ICL framework achieves high efficiency and fast convergence. Specifically, we have the following observations.

High Efficiency. The proposed ICL framework has a great superiority in terms of training time consumption as shown in Table 3 and Table 4, where we use the speedup of training time compared with retraining $s_{e,i} = time_{retrain}/time_i$ for method i as the metric. (a) For CV datasets, ICL achieves a speedup of up to 16.7× w.r.t. retraining. Moreover, without the meta-optimization and LRL (i.e., only applying NCE-II), ICL still brings us a speedup of up to 9.8×. Overall, it is obvious that ICL is the most efficient approach in all of the cases on CV datasets, even compared with fine-tuning and distillation which exclude the need of training old data. (b) For GRL datasets, ICL also achieves a speedup of up to 10.1×, and still 5.9× without the meta-optimization and LRL. It is worth mentioning that ICL can beat replay and distillation with only NCE-II.

Fast Convergence. The proposed ICL framework gives the fastest convergence speed. We use the speedup of convergence epoch compared with the retraining $s_{c,i} = epoch_{retrain}/epoch_i$ for method i as its metric. (a) For CV datasets, ICL achieves a speedup of up to 16.8× compared with retraining and the improvements are significant in all cases. Moreover, ICL only with NCE-II (w/o M+LRL) still gains a speedup of up to 16.4×. Similar to training time, the convergence speedup of ICL exceeds all methods. (b) For GRL datasets, the experimental results are identical. ICL with its variants achieves a speedup of up to 14.6×.

Therefore, the proposed ICL maintains significant advantages in reducing training time and accelerating convergence. Furthermore, although the time complexity of ICL is larger, the results still show the superiority of ICL in terms of efficiency, which contributed to the LRL mechanism.

4.3 Effectiveness Analysis: Does ICL impede model?

As an unbiased and efficient approach, the proposed ICL also achieves competitive performance as shown in Figure 2 and Figure 3. It is clearly observed that applying ICL to a contrastive learning approach will not impede the representation ability of the model. Specifically, the difference between ICL and the others falls in $[-0.0137, +0.0126]$ with an average improvement of 0.002.

4.4 Incremental Setting Analysis: When to use ICL?

We study the change of speedup of training time and convergence epoch with the variation of the growth ratio α. For fairness, we compare the unbiased methods (ICL and retraining) and report the results in Figure 4. Specifically, we have the following findings.

Consistent Efficiency. We vary α from 0.1 to 0.9 at 0.1 intervals, i.e., simulating the amount of new data from 1/9 of the old data to 9 times the old data. Overall, ICL achieves a 2.2×-18.3× speedup of training time and a 2.7×-22.9× speedup of convergence epoch.

Adaption Ability with a Large Ratio of New Data. The superiority of ICL in training time is obvious even when α is large. Moreover, from Table 3 and Table 4, ICL consistently achieves faster convergence while the other baselines nearly equal the retraining. Thus, ICL is more practical in scenarios where there is a large ratio of new data.

Limited Superiority by the Ratio of New Data. As α increases, the speedup of ICL becomes smaller due to a large amount of new data. Thus, overmuch new data leads to the weakening of incremental learning strategies. However, ICL nevertheless gains a 2.2×-5.3× speedup when the amount of the new data is 9 times the old one.

4.5 Ablation Study: How ICL works?

We further investigate the ablation study of the three important components: the NCE-II loss, the meta-optimization algorithm, and the LRL mechanism. We provide the loss change compared with NCE-I and Adam in Figure 5.

NCE-II Loss is unbiased and useful. From Table 3 and Table 4, it is noticed that ICL only with NCE-II (i.e., without meta-optimization and LRL) still achieves a faster training and convergence speed, up to 9.8× and 16.4× respectively. From Figure 5, NCE-II's decrease rate of loss is faster than only applying NCE-I, which reflects that NCE-II helps to quickly adapt to new data with an unbiased estimation.

Meta-optimization is essential. As shown in Table 3 and Table 4, without meta-optimization, the efficiency is significantly lowered. Moreover, comparing ICL only with NCE-II and ICL with meta-optimization, the efficiency is improved. Thus, meta-optimization contributes considerably to fast adaption of new data.

Learning Rate Learning has a vital contribution to the improvement of efficiency. From Table 3 and Table 4, without LRL mechanism, the degree of improvement on efficiency is reduced. In Figure 5, it is obvious that the loss declines faster with LRL compared with the Adam algorithm.

5 RELATED WORK

5.1 Contrastive Learning

Contrastive learning (CL) is a self-supervised approach that aims at learning to discriminate the samples by contrasting the negative ones [13, 17, 23]. Contrastive Predictive Coding (CPC) proposed an InfoNCE loss to maximize the mutual information between the sample and its positive one [37, 41]. Recently, InfoNCE loss has been widely used in computer vision (CV). SimCLR and MoCo subsequently use contrastive learning to generate the representations of images through different data augmentation methods [5, 11] and achieve comparable results with state-of-the-art supervised approaches. Furthermore, contrastive learning is also well researched in graph representation learning (GRL). Graph Contrastive Coding (GCC) and GrpahCL apply InfoNCE to learn the representation of node and graph [32, 44]. The above studies follow a similar framework to Definition 1 which is the backbone of ICL.

5.2 Incremental Learning

Incremental learning is a learning process where the new data is continuously coming from the environment [1, 30, 31]. Most studies
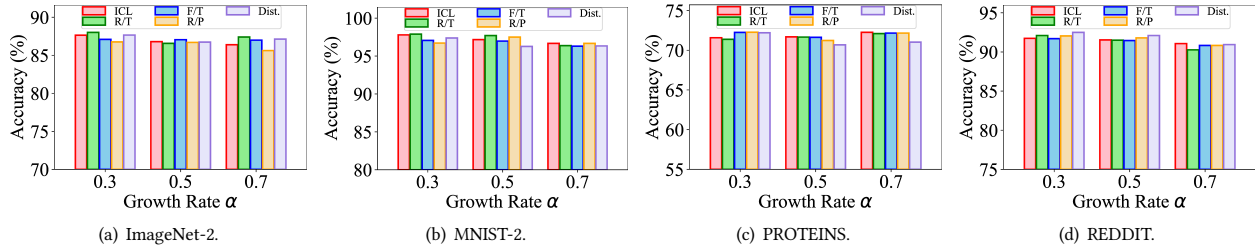
(a) ImageNet-2. (b) MNIST-2. (c) PROTEINS. (d) REDDIT.

Figure 2: Classification results of old data on four datasets. (R/T: Retraining; F/T: Fine-tuning; R/P: Replay; Dist.: Distillation)
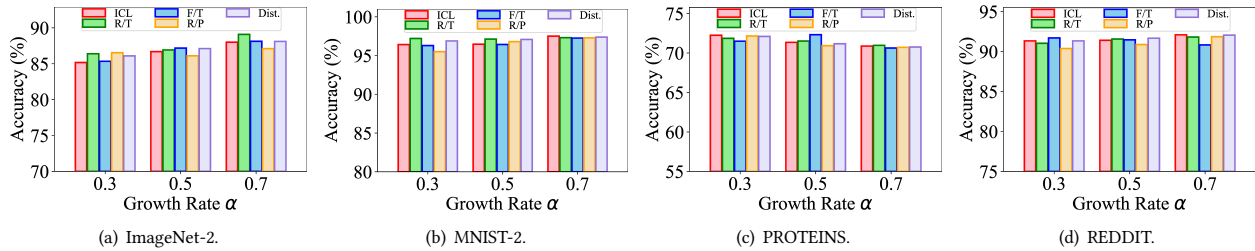

(a) ImageNet-2. (b) MNIST-2. (c) PROTEINS. (d) REDDIT.

Figure 3: Classification results of new data on four datasets. (R/T: Retraining; F/T: Fine-tuning; R/P: Replay; Dist.: Distillation)

(a) ImageNet-2 (Time). (b) ImageNet-2 (Epoch).
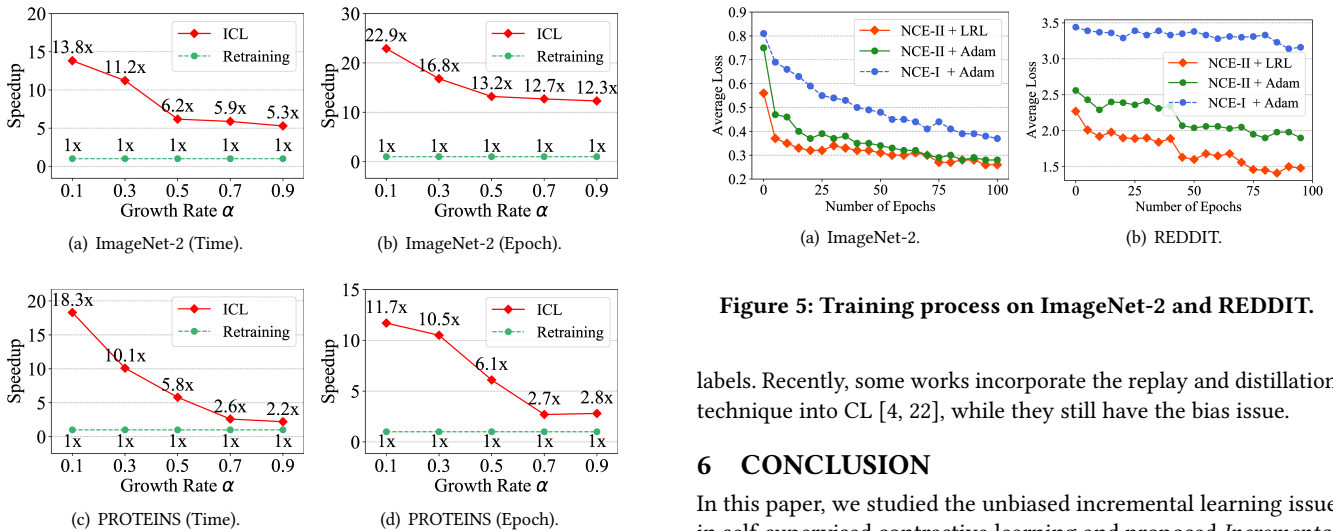
(c) PROTEINS (Time). (d) PROTEINS (Epoch).

Figure 4: Speedup of unbiased methods (ICL and retraining) with the variation of α. For fairness, the biased ones (fine-tuning, replay, and distillation) are excluded.

(a) ImageNet-2. (b) REDDIT.

Figure 5: Training process on ImageNet-2 and REDDIT.

labels. Recently, some works incorporate the replay and distillation technique into CL [4, 22], while they still have the bias issue.

6 CONCLUSION

In this paper, we studied the unbiased incremental learning issue in self-supervised contrastive learning and proposed *Incremental Contrastive Learning* (ICL) framework. Specifically, we designed an Incremental InfoNCE (NCE-II) loss function to give an unbiased estimation of noise distribution change in incremental scenarios. Moreover, we proposed a meta-optimization algorithm with Learning Rate Learning (LRL) mechanism to achieve fast convergence. The experiments demonstrated the efficiency and effectiveness of the proposed ICL framework.

ACKNOWLEDGMENTS

of incremental learning focus on supervised learning. For example, iCaRL proposed an incremental classifier and representation learning approach for supervised incremental learning [33]. In addition, [3] proposed an end-to-end incremental learning method for class-incremental issue. However, these methods are hard to implement into self-supervised contrastive learning due to the lack of

REFERENCES

[1] RR Ade and PR Deshmukh. 2013. Methods for incremental learning: a survey. *International Journal of Data Mining & Knowledge Management Process* 3, 4 (2013), 119.

[2] Chen Cai and Yusu Wang. 2018. A simple yet effective baseline for non-attributed graph classification. *arXiv preprint arXiv:1811.03508* (2018).

[3] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. 2018. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*. 233–248.

[4] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. 2021. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 9516–9525.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning (ICML)*. PMLR, 1597–1607.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition (CVPR)*. Ieee, 248–255.

[7] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. 2020. SEED: Self-supervised Distillation For Visual Representation. In *International Conference on Learning Representations (ICLR)*.

[8] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning (ICML)*. PMLR, 1126–1135.

[9] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 297–304.

[10] Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-Contrastive Estimation of Unnormalized Statistical Models, with Applications to Natural Image Statistics. *Journal of machine learning research* 13, 2 (2012).

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. 9729–9738.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 770–778.

[13] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies* 9, 1 (2020), 2.

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

[15] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[17] Jianxin Li, Cheng Ji, Hao Peng, Yu He, Yangqiu Song, Xinmiao Zhang, and Fanzhang Peng. 2021. RWNE: A Scalable Random-Walk based Network Embedding Framework with Personalized Higher-order Proximity Preserved. *Journal of Artificial Intelligence Research* 71 (2021), 237–263.

[18] Jianxin Li, Tianchen Zhu, Haoyi Zhou, Qingyun Sun, Chunyang Jiang, Shuai Zhang, and Chunming Hu. 2022. AIQoSer: Building the efficient Inference-QoS for AI Services. In *2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.

[19] Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, et al. 2022. A Survey on Deep Learning Event Extraction: Approaches and Applications. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[20] Qian Li, Hao Peng, Jianxin Li, Jia Wu, Yuanxing Ning, Lihong Wang, S Yu Philip, and Zheng Wang. 2021. Reinforcement learning-based dialogue guided event extraction to exploit argument relations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2021), 520–533.

[21] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning.. In *International Conference on Learning Representations (ICLR)*.

[22] Zhiwei Lin, Yongtao Wang, and Hongxiang Lin. 2021. Continual Contrastive Self-supervised Learning for Image Classification. *arXiv preprint arXiv:2107.01776* (2021).

[23] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* (2021).

[24] Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *International Conference on Learning Representations (ICLR)*.

[25] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. 2021. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3589–3599.

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

[27] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. 2013. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology* 4 (2013), 504.

[28] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[29] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. 2020. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663* (2020).

[30] Hao Peng, Jianxin Li, Yangqiu Song, and Yaopeng Liu. 2017. Incrementally learning the hierarchical softmax function for neural language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[31] Hao Peng, Renyu Yang, Zheng Wang, Jianxin Li, Lifang He, S Yu Philip, Albert Y Zomaya, and Rajiv Ranjan. 2021. Lime: Low-cost and incremental learning for dynamic heterogeneous information networks. *IEEE Trans. Comput.* 71, 3 (2021), 628–642.

[32] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 1150–1160.

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

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

[35] Alexander Shapiro. 2003. Monte Carlo sampling methods. *Handbooks in operations research and management science* 10 (2003), 353–425.

[36] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *International conference on machine learning (ICML)*. PMLR, 387–395.

[37] Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Xingcheng Fu, Cheng Ji, and S Yu Philip. 2022. Graph structure learning with variational information bottleneck. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4165–4174.

[38] Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Yuanxing Ning, Philip S Yu, and Lifang He. 2021. Sugar: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism. In *Proceedings of the Web Conference 2021*. 2081–2091.

[39] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems (NeurIPS)* 12 (1999).

[40] George E Uhlenbeck and Leonard S Ornstein. 1930. On the theory of the Brownian motion. *Physical review* 36, 5 (1930), 823.

[41] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints* (2018), arXiv–1807.

[42] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 3733–3742.

[43] Chang Xu, Tao Qin, Gang Wang, and Tie-Yan Liu. 2017. Reinforcement learning for learning rate control. *arXiv preprint arXiv:1705.11159* (2017).

[44] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), 5812–5823.