Continual Learning on Noisy Data Streams via Self-Purified Replay

Chris Dongjoo Kim* Jinseo Jeong* Sangwoo Moon Gunhee Kim
Department of Computer Science and Engineering, Seoul National University, Seoul, Korea
{cdjkim, jinseo, sangwoo.moon}@vl.snu.ac.kr gunhee@snu.ac.kr
http://vision.snu.ac.kr/projects/SPR

Abstract

Continually learning in the real world must overcome many challenges, among which noisy labels are a common and inevitable issue. In this work, we present a replay-based continual learning framework that simultaneously addresses both catastrophic forgetting and noisy labels for the first time. Our solution is based on two observations; (i) forgetting can be mitigated even with noisy labels via self-supervised learning, and (ii) the purity of the replay buffer is crucial. Building on this regard, we propose two key components of our method: (i) a self-supervised replay technique named Self-Replay which can circumvent erroneous training signals arising from noisy labeled data, and (ii) the Self-Centered filter that maintains a purified replay buffer via centrality-based stochastic graph ensembles. The empirical results on MNIST, CIFAR-10, CIFAR-100, and WebVision with real-world noise demonstrate that our framework can maintain a highly pure replay buffer amidst noisy streamed data while greatly outperforming the combinations of the state-of-the-art continual learning and noisy label learning methods.

1. Introduction

The most natural form of input for an intelligent agent occurs sequentially. Hence, the ability to continually learn from sequential data has gained much attention in recent machine learning research. This problem is often coined as *continual learning*, for which three representative approaches have been proposed [57, 67, 20] including replay [52, 29, 66, 73, 70, 44], regularization [38, 91, 3], and expansion techniques [71, 89].

At the same time, learning from data riddled with noisy labels is an inevitable scenario that an intelligent agent must overcome. There have been multiple lines of work to learn amidst noisy labels such as loss regularization [84, 96, 31], data re-weighting [68, 72], label cleaning [64, 42, 61], and training procedures [85, 36].

In this work, we aim to jointly tackle the problems of *continual learning* and *noisy label classification*, which to the best of our knowledge have not been studied in prior work. Noisy labels and continual learning are inevitable for real-world machine learning, as data comes in a stream possibly polluted with label inconsistency. Hence, the two are bound to intersect; we believe exploring this intersection may glean evidence for promising research directions and hopefully shed light on the development of sustainable real-world machine learning algorithms.

We take on the replay-based approach to tackle continual learning since it has often shown superior results in terms of performance and memory efficiency even with simplicity. Yet, we discover that replaying a noisy buffer intensifies the forgetting process due to the fallacious mapping of previously attained knowledge. Moreover, existing noisy label learning approaches show great limitations when coping within the online task-free setting [2, 65, 43, 37]. In their original forms, they assume that the whole dataset is given to purify the noise and thus are hampered by a small amount of data stored only in the replay buffer to either regularize, re-weight, or decide on its validity.

We begin by backtracking the root of the problem; if we naively store a sampled set of the noisy input stream into the replay buffer, it becomes riddled with noise, worsening the amount of forgetting. Thus, we discover the key to success is maintaining a pure replay buffer, which is the major motive of our novel framework named *Self-Purified Replay* (SPR). At the heart of our framework is self-supervised learning [16, 12, 30, 24], which allows to circumvent the erroneous training signals arising from the incorrect pairs of data and labels. Within the framework, we present our novel *Self-Replay* and *Self-Centered filter* that collectively cleanse noisy labeled data and continually learn from them. The Self-Replay mitigates the noise intensified catastrophic forgetting, and the Self-Centered filter achieves a highly clean replay buffer even when restricted to a small portion of data at a time.

We outline the contributions of this work as follows.

*Equal Contribution

1. To the best of our knowledge, this is the first work to tackle *noisy labeled continual learning*. We discover noisy labels exacerbate catastrophic forgetting, and it is critical to filter out such noise from the input data stream before storing them in the replay buffer.
2. We introduce a novel replay-based framework named Self-Purified Replay (SPR), for noisy labeled continual learning. SPR can not only maintain a clean replay buffer but also effectively mitigate catastrophic forgetting with a fixed parameter size.
3. We evaluate our approach on three synthetic noise benchmarks of MNIST [41], CIFAR-10 [40], CIFAR-100 [40] and one real noise dataset of WebVision [49]. Empirical results validate that SPR significantly outperforms many combinations of the state-of-the-art continual learning and noisy label learning methods.

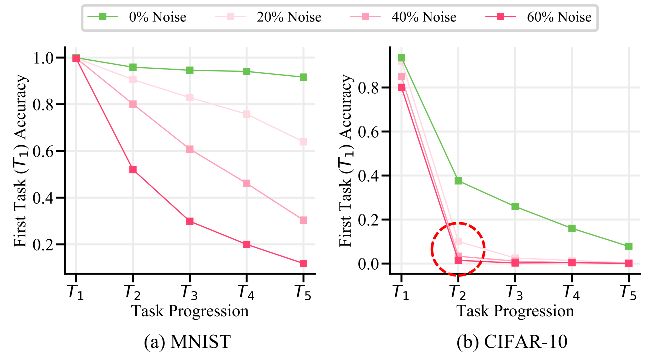


Figure 1. A noisy labeled continual learning on the symmetric noisy in (a) MNIST [41] and (b) CIFAR-10 [40] when using experience replay with the conventional reservoir sampling [94, 69]. At the end of each task, the accuracy of the first task (T_1) is plotted. It shows that the noisy labels accelerate catastrophic forgetting. Notably, the dotted red circle in (b) indicates the significantly hastened forgetting process.

2. Problem Statement

2.1. Noisy Labeled Continual Learning

We consider the problem of online task-free continual learning for classification where a sample $\{x_t, y_t\}$ enters at each time step t in a non i.i.d manner without task labels. While previous works [66, 65, 43] assume $\{x_t, y_t\}$ are correct (clean) samples, we allow the chance that a large portion of the data is falsely labeled.

2.2. Motivation: Noise induced Amnesia

We discover that if the data stream has noisy labels, it traumatically damages the continual learning model, analogous to *retrograde amnesia* [75], the inability to recall experience of the past. We perform some preliminary experiments on a sequential version of symmetric noisy MNIST and CIFAR-10 [53, 84] using experience replay with the conventional reservoir sampling technique [69, 94].

The empirical results in Figure 1 show that when trained with noisy labels, the model becomes much more prone to catastrophic forgetting [20, 57, 78, 67]. As the noise level increases from 0% to 60%, sharp decreases in accuracy are seen. Surprisingly, the dotted red circle in Figure 1(b) shows that in CIFAR-10 a fatally hastened forgetting occurs no matter the amount of noise.

We speculate that a critical issue that hinders the continual model is the corrupted replay buffer. An ideal replay buffer should shield the model from noisy labels altogether by being vigilant of all the incoming data for the maintenance of a clean buffer.

3. Approach to Noisy Labeled Continual Learning

We design an approach to continual learning with noisy labels by realizing the two interrelated subgoals as follows.

G1. *Reduce forgetting even with noisy labels*: The approach needs to mitigate catastrophic forgetting amidst learning from noisy labeled data.

G2. *Filter clean data*: The method should learn representations such that it identifies the noise as anomalies. Moreover, it should enable this from a *small amount* of data since we do not have access to the entire dataset in online continual learning.

Figure 2 overviews the proposed framework consisting of two buffers and two networks. The *delayed buffer* $\mathcal{D}$ temporarily stocks the incoming data stream, and the *purified buffer* $\mathcal{P}$ maintains the cleansed data. The *base* network addresses **G1** via self-supervised replay (Self-Replay) training (Section 3.1). The *expert* network is a key component of Self-Centered filter that tackles **G2** by obtaining confidently clean samples via centrality (Section 3.2). Both networks have the same architecture (*e.g.*, ResNet-18) with separate parameters.

Algorithm 1 outlines the training and filtering procedure. Whenever the delayed buffer $\mathcal{D}$ is full, The Self-Centered filter powered by the expert network filters the clean samples from $\mathcal{D}$ to the purified buffer $\mathcal{P}$. Then, the base network is trained via the self-supervision loss with the samples in $\mathcal{D} \cup \mathcal{P}$. The detail will be discussed in Section 3.1–3.2.

At any stage of learning, we can perform downstream tasks (*i.e.*, classification) by duplicating the base network into the inference network, adding a final softmax layer, and finetuning it using the samples in $\mathcal{P}$. Algorithm 2 outlines this inference phase.

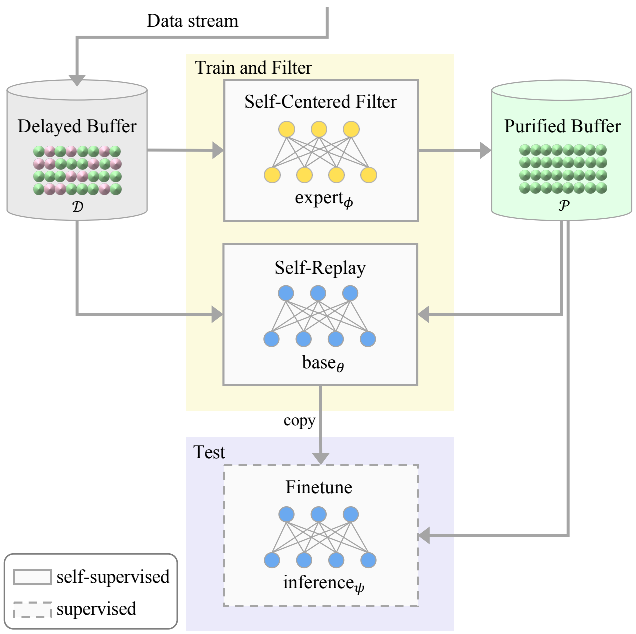


Figure 2. Illustration of the Self-Purified Replay (SPR) framework. We specify the training and filtering phase (in the yellow shade) in Algorithm 1, and the test phase (in the purple shade) in Algortihm 2.

Algorithm 1 Training and filtering phase of SPR

Input: Training data $(x_t, y_t), ..., (x_T, y_T)$ and initial parameters of base network θ.
$\mathcal{D} = \mathcal{P} = \{\}$ // Initialize delayed and purified buffer
for $t = 1$ **to** T **do**
  if $\mathcal{D}$ is full **then**
    $\mathcal{P} \leftarrow \mathcal{P} \cup$ Self-Centered Filter(D) (section 3.2)
    $\theta \leftarrow$ Self-Replay using $\mathcal{D} \cup \mathcal{P}$ (section 3.1)
    reset $\mathcal{D}$
  else
    update $\mathcal{D}$ with (x_t, y_t)
  end if
end for

Algorithm 2 Test phase of SPR

Input: Test data $(x_t, y_t), ..., (x_T, y_T)$, parameters of the base network θ, and purified buffer $\mathcal{P}$
ψ = copy(θ) // Duplicate base model to inference model
$\psi \leftarrow$ supervised finetune using $\mathcal{P}$
for $t = 1$ **to** T **do**
  downstream classification for (x_t, y_t) using ψ
end for

3.1. Self-Replay

Learning with noisy labeled data [64, 5, 54, 28] results in erroneous backpropagating signals when falsely paired x and y exist in the training set. Hence, we circumvent this error via learning only from x (without y) using contrastive self-supervised learning techniques [7, 12, 30, 24]. That is, the framework first focuses on learning general representations via self-supervised learning from *all* incoming x. Subsequently, the downstream task (*i.e*., supervised classification) finetunes the representation using only the samples in the purified buffer $\mathcal{P}$. Building on this concept in terms of continual learning leads to Self-Replay, which mitigates forgetting while learning general representations via self-supervised replay of the samples in the delayed and purified buffer ($\mathcal{D} \cup \mathcal{P}$).

Specifically, we add a projection head $g(\cdot)$ (*i.e*., a one-layer MLP) on top of the average pooling layer of the base network, and train it using the normalized temperature-scaled cross-entropy loss [12]. For a minibatch from $\mathcal{D}$ and $\mathcal{P}$ with a batch size of $B_d, B_p \in \mathbb{N}$ respectively, we apply random image transformations (*e.g*., cropping, color jitter, horizontal flip) to create two correlated views of each sample, referred to as positives. Then, the loss is optimized to attract the features of the positives closer to each other while repelling them from the other samples in the batch, referred to as the negatives. The updated objective becomes

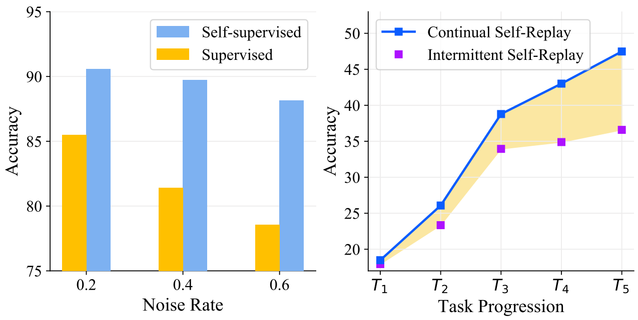
$$L_{self} = -\sum_{i=1}^{2(B_d+B_p)} \log \frac{e^{u_i^T u_j/\tau}}{\sum_{k=1}^{2(B_d+B_p)} \mathbb{1}_{k \neq i} e^{u_i^T u_k/\tau}}. \quad (1)$$

We denote (x_i, x_j) as the positives and x_k as the negatives. $u_i = \frac{g(x_i)}{||g(x_i)||_2}$ is the ℓ_2 normalized feature, and $\tau > 0$ is the temperature. Every time when the delayed buffer is full, we train the base network with this loss.

Empirical supports. Figure 3 shows some empirical results about the validity of Self-Replay for noisy labeled continual learning.

- Figure 3(a) shows a quantitative examination on downstream classification tasks. It indicates that self-supervised learning leads to a better representation, and eventually outperforms the supervised one by noticeable margins.

- Figure 3(b) exemplifies the superiority of Self-Replay in continual learning. We contrast the performances of continually trained Self-Replay (as proposed) against intermittently trained Self-Replay, which trains *offline* with only the samples in the purified buffer at the end of each task. The colored areas in Figure 3(b) indicate how much the continually learned representations alleviate the forgetting and benefit the knowledge transfers among the past and future tasks.



(a) Downstream task accuracy (b) Continual vs. Intermittent

Figure 3. **Empirical support for Self-Replay** with ResNet18 as the base network on CIFAR-10. (a) Comparison of overall accuracy of the finetuned downstream classification between self-supervised and supervised representations trained on various noise rates. The self-supervised indicates that the base network trained using only x as proposed, while the supervised means training with possibly noisy (x, y) pairs. (b) The benefits of *continual* Self-Replay over the *intermittent* Self-Replay by comparing the test set accuracy of finetuned models. The intermittent Self-Replay means training only with contents of the purified buffer up to and including the current task.

3.2. Self-Centered Filter

The goal of the Self-Centered filter is to obtain confidently clean samples; specifically, it assigns the probability of being clean to all the samples in the delayed buffer.

Expert Network. The expert network is prepared to *featurize* the samples in the delayed buffer. These features are used to compute the centrality of the samples, which is the yardstick of selecting clean samples. Inspired by the success of self-supervised learning good representations in Self-Replay, the expert network is also trained with the self-supervision loss in Eq. 1 with only difference that we use the samples in $\mathcal{D}$ only (instead of $\mathcal{D} \cup \mathcal{P}$ for the base network).

Centrality. At the core of the Self-Centered filter lies centrality [59], which is rooted in graph theory to identify the most influential vertices within a graph. We use a variant of the *eigenvector centrality* [6], which is grounded on the concept that a link to a highly influential vertex contributes to centrality more than a link to a lesser influential vertex.

First, weighted undirected graphs $G := (V, E)$ are constructed *per unique class label* in the delayed buffer. We assume that the clean samples form the largest clusters in the graph of each class. Each vertex $v \in V$ is a sample of the class, and the edge $e \in E$ is weighted by the cosine similarity between the features from the expert network. For the adjacency matrix $\mathbf{A} = (a_{v,u})_{|V| \times |V|}$. Then the eigenvector centrality is formulated as

$$c_v = \frac{1}{\lambda} \sum_{u \in N(v)} c_u = \frac{1}{\lambda} \sum_{u \in V} a_{v,u} c_u, \quad (2)$$

where $N(v)$ is the neighboring set of v, λ is a constant and $a_{v,u}$ is the truncated similarity value within $(0, 1]$. Eq. 2 can be rewritten in vector notation as $\mathbf{Ac} = \lambda\mathbf{c}$, where $\mathbf{c}$ is a vectorized centrality over V. The principal eigenvector $\mathbf{c}$ can be computed by the power method [82], and it corresponds to the eigenvector centrality for the vertices in V.

Beta Mixture Models. The centrality quantifies which samples are the most influential (or the cleanest) within the data of identical class labels. However, the identically labeled data contains both *clean* and *noisy* labeled samples, in which the noisy ones may deceptively manipulate the centrality score, leading to an indistinct division of the clean and noisy samples' centrality scores. Hence, we compute the probability of cleanliness per sample via fitting a Beta mixture model (BMM) [33] to the centrality scores as

$$p(c) = \sum_{z=1}^{Z} \pi_z p(c|z), \quad (3)$$

where $c > 0$ is the centrality score, π_z is the mixing coefficients, and $Z \in \mathbb{N}$ is the number of components. Beta distribution for $p(c|z)$ is a suitable choice due to the skewed nature of the centrality scores. We set $Z = 2$, indicating the clean and noisy components, and it is empirically the best in terms of accuracy and computation cost. We use the EM algorithm [15] to fit the BMM through which we obtain the posterior probability

$$p(z|c) = \frac{\pi_z p(c|\alpha_z, \beta_z)}{\sum_{j=1}^{Z} \pi_j p(c|\alpha_j, \beta_j)}, \quad (4)$$

where $\alpha_z, \beta_z > 0$ are the latent distribution parameters. Please refer to the appendix for details of computing $p(z|c)$.

Among the $Z = 2$ components, we can easily identify the *clean* component as the one that has the higher c scores (*i.e.*, a larger cluster). Then, the clean posterior $p(z = \text{clean}|c)$ defines the probability that centrality c belongs to the clean component, which is used as the probability to enter and exit the purified buffer, $\mathcal{P}$. After the selected samples enters our full purified buffer, the examples with the lowest $p(z = \text{clean}|c)$ are sampled out accordingly.

3.2.1 Stochastic Ensemble

Since our goal is to obtain the *most* clean samples as possible, we want to further sort out the possibly noisy samples. We achieve this by introducing a *stochastic ensemble* of BMMs, enabling a more noise robust posterior than the non-stochastic posterior $p(z = \text{clean}|c)$ in the previous section.

First, we prepare for stochastic ensembling by sampling multiple *binary* adjacency matrices $\{\mathbf{A}\}$ from a Bernoulli

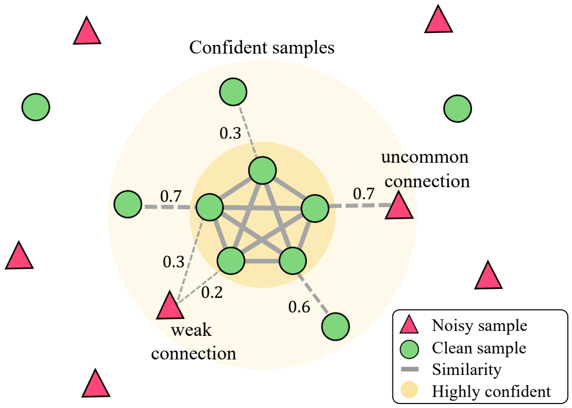


Figure 4. Illustration of graph manipulation via Stochastic Ensemble, which severs weak and uncommon connections and probabilistically focus on confident and clean data within the graph.

distribution over $\mathbf{A}$. For each class l, we impose a conditional Bernoulli distribution over $\mathbf{A}$ as

$$p(\mathbf{A}|D_l) = \prod_{d_i, d_j \in D_l} \text{Bern}\left(\mathbf{A}_{ij}|\text{ReLU}\left(\frac{d_i \cdot d_j}{||d_i||||d_j||}\right)\right), \quad (5)$$

where D_l is the set of penultimate feature of class l from the expert network. We find that it is empirically helpful to truncate the dissimilar values to 0 (ReLU) and use the cosine similarity value as the probability. We replace the zeros in $\mathbf{A}$ with a small positive value to satisfy the requirement of Perron-Frobenius theorem[1]. Then, our reformulated robust posterior probability is

$$p(z|D_l) \propto \int_{\mathbf{A}} p(z|\text{cent}(\mathbf{A}))dp(\mathbf{A}|D_l), \quad (6)$$

where $\text{cent}(\cdot)$ is the centrality scores from Eq. 2, and $p(z|\text{cent}(\mathbf{A}))$ can be obtained in the same manner as the *non-stochastic* posterior in the previous section. We approximate the integral using Monte Carlo sampling for which we use E_{max} as the sample size. Essentially, we fit the mixture models on different stochastic graphs to probabilistically carve out more confidently noisy samples by retaining the strong and dense connections while severing weak or uncommon connections. This is conceptually illustrated in Figure 4.

Empirical Supports. Figure 5 shows some empirical evidence where the stochastic ensemble addresses the two issues to achieve a noise robust posterior $p(z|D_l)$.

- First, a small portion of noisy samples are falsely confident and are consequently assigned a high centrality score. Stochastic ensembling is able to suppress these noisy samples, as indicated in Figure 5, where the mode of $p(c|z = \text{noisy}) \cdot p(z = \text{noisy})$ (red curve) is shifted to the left by a noticeable margin.
- Second, there are some cases where $p(c|z = \text{noisy}) \cdot p(z = \text{noisy})$ drops below the $p(c|z = \text{clean}) \cdot p(z = \text{clean})$ leading to a high $p(z = \text{clean}|c)$ for the noisy instances, indicated with red circles in Figure 5. The stochastic ensemble of differing $\mathbf{A}$s can mitigate such problematic cases to drown out the unexpected noise.

[1] Perron-Frobenius theorem states when $\mathbf{A}$ has positive entries, it has a unique largest real eigenvalue, whose corresponding eigenvector have strictly positive components.

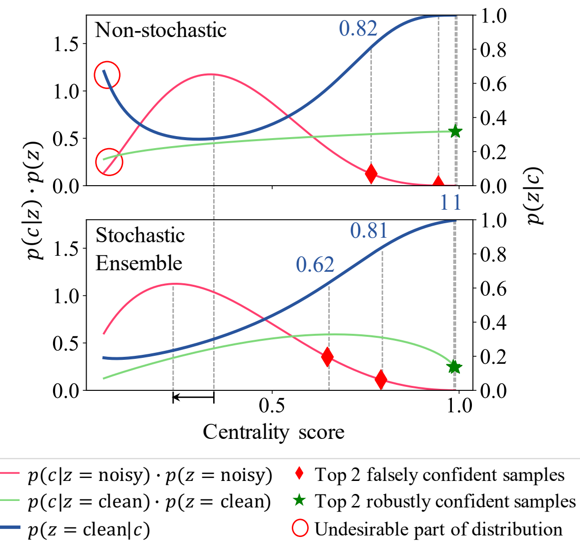


Figure 5. Comparison of non-stochastic and Stochastic Ensemble on CIFAR-10 with 40% noise. Stochastic Ensemble produces more confidently clean samples by shifting $p(c|z = \text{noisy}) \cdot p(z = \text{noisy})$ to the left, and suppressing the cases where $p(c|z = \text{noisy}) \cdot p(z = \text{noisy})$ dips below $p(c|z = \text{clean}) \cdot p(z = \text{clean})$.

4. Related Works

4.1. Continual Learning

There have been three main branches to train a model from continual data streams: regularization [51, 19, 38, 3], expansion [71, 89, 43], and replay [52, 9, 10, 69, 34]. Replay-based approaches maintain a fixed-sized memory to rehearse back to the model to mitigate forgetting. Several works [52, 9, 10] reserve the space for data samples of previous tasks, while others [73] uses a generative model. Some works [69, 34] combine rehearsal with meta-learning to find the balance between transfer and interference. We defer more comprehensive survey including all three branches of continual learning to the appendix.

Online Sequential Learning. In the online sequential learning scenario, a model can only observe the training samples once. Hence, many works propose methods for maintaining the buffer [29, 66, 37] or selecting the samples to be rehearsed [2]. Recently, [77] adopts graphs to represent relational structures between samples, and [25] employs the meta-loss for learning per-parameter learning rates along with model parameters.

Akin to our work, Gdumb [65] and MBPA++ [14] also train the model at inference time. However, greedily selecting samples to be reserved inevitably leads to degradation from noisy labeled data. Furthermore, discarding the samples that cannot enter the buffer as done in Gdumb may lead to information loss since it only relies on the buffer as its source of training.

4.2. Noisy Labels

Learning with noisy labeled data has long been studied [92, 5, 54, 35]. Several works design the noise corrected losses [86, 28, 46, 4, 84] so that the loss minimization of the whole data becomes similar to that of clean samples. Other works propose to use a noise transition matrix to correct the loss [63, 23, 31, 96]. There have been approaches that aim to suppress the contribution of noisy samples by re-weighting the loss [83, 68]. Techniques that repair labels [39, 80, 50, 74, 27, 56] or directly learn them [76, 88] are also viable options for learning from noisy labeled data. Recently, filtering methods based on training dynamics [32, 64, 58] have gained much popularity, based on the observation that models tend to learn clean data first and memorize the noisy labeled data later. Small loss sample selection [36, 72, 45] techniques by co-teaching [85, 21, 26, 90, 55, 11] identify noisy samples with multiple models in the same vein. Some works use graphs for offline learning from a large-scale noisy dataset [95, 93]. On the other hand, we use a small dataset in the delayed buffer from an online data stream without ground-truth labels; instead we adopt self-supervision to obtain features for the Self-Centered filter.

None of the works mentioned above address continual learning from noisy labeled data streams. Although [56, 47] also use self-supervised learning with noisy labeled data, they focus on the loss or prediction from the model for selecting suspicious samples. In the experiments on Table 3, we will show that training dynamics-based filtering techniques are not viable in noisy labeled continual learning. On the other hand, we provide the algorithm that identifies the clean samples while learning from a purified buffer in an online manner.

4.3. Self-supervised learning

Self-supervised learning is currently receiving an enormous amount of attention in machine learning research. The pretext task that trains a model by predicting hidden information within the data includes patch orderings [17, 60], image impainting [62], colorization [87], and rotations [22, 13], to name a few. There also have been works that utilize the contrastive loss [12, 30, 48]; especially, SimCLR [12] proposes a simplified contrastive learning method, which enables representation learning by pulling the randomly transformed samples from the same image closer while pushing ones apart from other images within the batch. Recently, this instance-wise contrastive learning is extended to prototypical contrastive learning [48] to encode the semantic structures within the data.

5. Experiments

In our evaluation, we compare SPR with other state-of-the-art models in the online task-free continual learning scenario with label noise. We test on three benchmark datasets of MNIST [41], CIFAR-10 [40] and CIFAR-100 [40] with symmetric and asymmetric random noise, and one large-scale dataset of WebVision [49] with real-world noise on the Web. We also empirically analyze Self-Replay and the Self-Centered filter from many aspects.

5.1. Experimental Design

We explicitly ground our experiment setting based on the recent suggestions for robust evaluation in continual learing [1, 18, 79] as follows. (i) *Cross-task resemblance*: Consecutive tasks in MNIST [41], CIFAR-10 [40], CIFAR-100 [40], WebVision [49] are partly correlated to contain neighboring domain concepts. (ii) *Shared output heads*: A single output vector is used for all tasks. (iii) *No test-time task labels*: Our approach does not require explicit task labels during both training and test phase, often coined as *task-free continual learning* in [66, 43, 37]. (iv) *More than two tasks*: MNIST [41], CIFAR-10 [40], CIFAR-100 [40] and WebVision [49] contain five, five, twenty, and seven tasks, respectively.

We create a synthetic noisy labeled dataset from MNIST and CIFAR-10 using two methods. First, the *symmetric* label noise assigns $\{20\%, 40\%, 60\%\}$ samples of the dataset to other labels within the dataset by a uniform probability. We then create five tasks by selecting random class pairs without replacement. Second, the *asymmetric* label noise attempts to mimic the real-world label noise by assigning other similar class labels (*e.g.*, 5 $\leftrightarrow$ 6, cat $\leftrightarrow$ dog). We use the similar classes chosen in [63] to contaminate $\{20\%, 40\%\}$ samples of the dataset with similar class pairs. Each task consists of the samples from each corrupted class pair. CIFAR-100 has 20 tasks where the *random symmetric* setting has 5 random classes per task with uniform noise across 100 classes. The *superclass symmetric* setting uses each superclass [40, 43] containing 5 classes as a task where the noise is randomized only within the classes in the superclass. In WebVision, we use the top 14 largest classes in terms of the data size, resulting in 47,784 images in total. We curate seven tasks with randomly paired classes.

	MNIST					CIFAR-10					WebVision
	symmetric			asymmetric		symmetric			asymmetric		real noise
noise rate (%)	20	40	60	20	40	20	40	60	20	40	unknown
Multitask 0% noise [8]			98.6					84.7			-
Multitask [8]	94.5	90.5	79.8	93.4	81.1	65.6	46.7	30.0	77.0	68.7	55.5
Finetune	19.3	19.0	18.7	21.1	21.1	18.5	18.1	17.0	15.3	12.4	11.9
EWC [38]	19.2	19.2	19.0	21.6	21.1	18.4	17.9	15.7	13.9	11.0	10.0
CRS [81]	58.6	41.8	27.2	72.3	64.2	19.6	18.5	16.8	28.9	25.2	19.3
CRS + L2R [68]	80.6	72.9	60.3	83.8	77.5	29.3	22.7	16.5	39.2	35.2	-
CRS + Pencil [88]	67.4	46.0	23.6	72.4	66.6	23.0	19.3	17.5	36.2	29.7	26.6
CRS + SL [84]	69.0	54.0	30.9	72.4	64.7	20.0	18.8	17.5	32.4	26.4	21.5
CRS + JoCoR [85]	58.9	42.1	30.2	73.0	63.2	19.4	18.6	21.1	30.2	25.1	19.5
PRS [37]	55.5	40.2	28.5	71.5	65.6	19.1	18.5	16.7	25.6	21.6	19.0
PRS + L2R [68]	79.4	67.2	52.8	82.0	77.8	30.1	21.9	16.2	35.9	32.6	-
PRS + Pencil [88]	62.2	33.2	21.0	68.6	61.9	19.8	18.3	17.6	29.0	26.7	26.5
PRS + SL [84]	66.7	45.9	29.8	73.4	63.3	20.1	18.8	17.0	29.6	24.0	21.7
PRS + JoCoR [85]	56.0	38.5	27.2	72.7	65.5	19.9	18.6	16.9	28.4	21.9	20.2
MIR [2]	57.9	45.6	30.9	73.1	65.7	19.6	18.6	16.4	26.4	22.1	17.2
MIR + L2R [68]	78.1	69.7	49.3	79.4	73.4	28.2	20.0	15.6	35.1	34.2	-
MIR + Pencil [88]	70.7	34.3	19.8	79.0	58.6	22.9	20.4	17.7	35.0	30.8	22.3
MIR + SL [84]	67.3	55.5	38.5	74.3	66.5	20.7	19.0	16.8	28.1	22.9	20.6
MIR + JoCoR [85]	60.5	45.0	32.8	72.6	64.2	19.6	18.4	17.0	27.6	23.5	19.0
GDumb [65]	70.0	51.5	36.0	78.3	71.7	29.2	22.0	16.2	33.0	32.5	30.4
GDumb + L2R [68]	65.2	57.7	42.3	67.0	62.3	28.2	25.5	18.8	30.5	30.4	-
GDumb + Pencil [88]	68.3	51.6	36.7	78.2	70.0	26.9	22.3	16.5	32.5	29.7	26.9
GDumb + SL [84]	66.7	48.6	27.7	73.4	68.1	28.1	21.4	16.3	32.7	31.8	30.8
GDumb + JoCoR [85]	70.1	56.9	37.4	77.8	70.8	26.3	20.9	15.0	33.1	32.2	24.2
Self-Centered filter	80.1	79.0	77.4	80.0	79.6	36.5	35.7	32.5	37.1	36.9	33.0
Self-Replay	81.5	69.2	43.0	86.3	78.9	40.1	31.4	22.4	44.1	43.2	**48.0**
SPR	**85.4**	**86.7**	**84.8**	**86.8**	**86.0**	**43.9**	**43.0**	**40.0**	**44.5**	**43.9**	40.0

Table 1. **Overall accuracy** of noisy labeled continual learning after all sequences of tasks are trained. The buffer size is set to 300, 500, 1000 for MNIST, CIFAR-10 and WebVision, respectively. Some empty slots on WebVision are due to the unavailability of clean samples required by L2R for training [68]. The results are the mean of five unique random seed experiments. We report best performing baselines on different episodes with variances in the appendix.

We fix the delayed buffer and the replay (purified) buffer size to 300, 500, 1000, 5000 for MNIST, CIFAR-10, WebVision, and CIFAR-100, respectively. The purified buffer maintains balanced classes as in [37, 65]. We fix the stochastic ensemble size, $E_{max} = 5$ unless stated otherwise. For the base model, we use an MLP with two hidden layers for all MNIST experiments and ResNet-18 for CIFAR-10, CIFAR-100, and WebVision experiments. Please refer to the appendix for experiment details.

5.2. Baselines

Since we opt for continual learning from noisy labeled data streams, we design the baselines combining existing state-of-the-art methods from the two domains of continual learning and noisy label learning.

We explore the replay-based approaches that can learn in the online task-free setting. We thus choose (i) Conventional Reservoir Sampling (CRS) [69], (ii) Maximally Interfered Retrieval (MIR) [2], (iii) Partitioning Reservoir Sampling (PRS) [37] and (iv) GDumb [65].

For noisy label learning, we select six models to cover many branches of noisy labeled classification. They include (i) SL loss correction [84], (ii) semi-supervised JoCoR [85], (iii) sample reweighting L2R [68], (iv) label repairing Pencil [88], (v) training dynamic based detection AUM [64] and (vi) cross-validation based INCV [11].

5.3. Results

Overall performance. Table 1 compares the noisy labeled continual learning performance (classification accuracy) between our SPR and baselines on MNIST, CIFAR-10 and WebVision. Additionally, Table 2 compares SPR against the best performing baselines on CIFAR-100 with random symmetric noise and superclass symmetric noise. SPR performs the best in all symmetric and asymmetric noise types with different levels of 20%, 40%, and 60% as

noise rate (%)	random symmetric 20	40	60	superclass symmetric 20	40	60
GDumb + L2R [68]	15.7	11.3	9.1	16.3	12.1	10.9
GDumb + Pencil [88]	16.7	12.5	4.1	17.5	11.6	6.8
GDumb + SL [84]	19.3	13.8	8.8	18.6	13.9	9.4
GDumb + JoCoR [85]	16.1	8.9	6.1	15.0	9.5	5.9
SPR	**21.5**	**21.1**	**18.1**	**20.5**	**19.8**	**16.5**

Table 2. **CIFAR100 results** of noisy labeled continual learning after all sequences of tasks are trained. The results are the mean of five unique random seed experiments.

	MNIST					CIFAR-10				
	symmetric			asymmetric		symmetric			asymmetric	
noise rate (%)	20	40	60	20	40	20	40	60	20	40
AUM [64]	7.0	16.0	11.7	30.0	29.5	36.0	24.0	11.7	46.0	30.0
INCV [11]	23.0	22.5	14.3	37.0	31.5	22.0	18.5	9.3	37.0	30.0
Non-stochastic	79.5	96.3	84.5	96.0	88.5	50.5	54.5	38.0	53.0	50.5
SPR	**96.0**	**96.5**	**93.0**	**100**	**96.5**	**75.5**	**70.5**	**54.3**	**69.0**	**60.0**

Table 3. **Filtered noisy label percentage** in the purified buffer (*e.g.*, out of 20% symmetric noise, SPR filters 96% of noise). We compare SPR with $E_{max} = 5$ to two other state-of-the-art label filtering methods.

well as real noise. Multitask is an upper-bound trained with an optimal setting with perfectly clean data (*i.e.*, the 0% noise rate) and offline training. Finetune is reported as a lower-bound performance since it performs online training with no continual or noisy label learning technique.

Notably, SPR works much better than L2R [68], which additionally uses 1000 clean samples for training, giving it a substantial advantage over all the other baselines. SPR also proves to be much more effective than GDumb [65], which is the most related method to ours, even when combined with different noisy label learning techniques.

Moreover, the addition of state-of-the-art noisy label techniques is not always beneficial. This may be because existing noisy label techniques usually assume a large dataset, which is required to reliably estimate the training dynamics to mitigate the noise by regularizing, repairing, and or filtering. However, the online learning setting is limited by a much smaller dataset (*i.e.*, in the purified buffer), leading to a difficult training of the noisy label techniques.

Ablation Study. To study the effectiveness of each component, two variants of our model that only use Self-Replay or the Self-Centered filter is tested. That is, the Self-Replay variant does not use any cleaning methods (*i.e.*, use conventional reservoir sampling to maintain the purified buffer). The Self-Centered filter variant finetunes a randomly initialized inference network on the purified buffer instead of finetuning it on the duplicate of the base network. Both variants outperform all the baselines (excluding L2R) in all three datasets, and combining them our model performs the best on MNIST and CIFAR-10 with all noise levels. However, WebVision is the only dataset where no synergetic effect is shown, leaving Self-Replay alone to perform the best. This may be because the WebVision contains highly abstract and noisy classes such as "Spiral" or "Cinema," making it difficult for Self-Centered filter to sample from correct clusters. Please refer to the appendix for further detail.

Purification Comparison. Table 3 compares the purification performance with the state-of-the-art noise detection methods based on the training dynamics, including AUM [64] and INCV [11]. We notice that the performance of AUM and INCV dreadfully declines when detecting label noise among only a small set of data, which is inevitable in online task-free setting, whereas SPR can filter superbly even with a small set of data. Even a non-stochastic version of our Self-Centered filter performs better than the baselines. Encouragingly, our method is further improved by introducing stochastic ensembles.

Additional Experiments. The appendix reports more experimental results, including SPR's noise-free performance, CIFAR-100 filtering performance, episode robustness, purified & delayed buffer size analysis, ablation of stochastic ensemble size, variance analysis, and data efficiency of Self-Replay.

6. Conclusion

We presented the Self-Purified Replay (SPR) framework for noisy labeled continual learning. At the heart of our framework is Self-Replay, which leverages self-supervised learning to mitigate forgetting and erroneous noisy label signals. The Self-Centered filter maintains a purified replay buffer via centrality-based stochastic graph ensembles. Experiments on synthetic and real-world noise showed that our framework can maintain a very pure replay buffer even with highly noisy data streams while significantly outperforming many combinations of noisy label learning and continual learning baselines. Our results shed light on using self-supervision to solve the problems of continual learning and noisy labels jointly. Specifically, it would be promising to extend SPR to maintain a not only pure but also more diversified purified buffer.

7. Acknowledgement

We express our gratitude for the helpful comments on the manuscript by Junsoo Ha, Soochan Lee and the anonymous reviewers for their thoughtful suggestions. This research was supported by the international coorperation program by NRF of Korea (NRF-2018K2A9A2A11080927), Basic Science Research Program through the National Research Foundation of Korea (NRF) (2020R1A2B5B03095585), and Institue of Information & communications Technology Planning & Evaluation (IITP) grant (No.2019-0-01082, SW StarLab). Gunhee Kim is the corresponding author.

References

[1] R. Aljundi. *Continual Learning in Neural Networks*. PhD thesis, Department of Electrical Engineering, KU Leuven, 2019. 6

[2] R. Aljundi, L. Caccia, E. Belilovsky, M. Caccia, M. Lin, L. Charlin, and T. Tuytelaars. Online continual learning with maximally interfered retrieval. In *NeurIPS*, 2019. 1, 6, 7

[3] R. Aljundi, R. Marcus, and T. Tuytelaars. Selfless sequential learning. In *ICLR*, 2019. 1, 5

[4] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, and K. McGuinness. Unsupervised label noise modeling and loss correction. In *ICML*, 2019. 6

[5] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien. A closer look at memorization in deep networks. In *ICML*, 2017. 3, 6

[6] P Bonacich and P Lloyd. Eigenvector-like measures of centrality for asymmetric relations. *Social Networks*, 3:191–201, 2001. 4

[7] M. Caron, I. Misra, J. Marial, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 3

[8] R. Caruaca. Multitask learning. *Machine Learning*, 28:41–75, 1997. 7

[9] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019. 5

[10] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486v4*, 2019. 5

[11] P. Chen, B. Liao, G. Chen, and S. Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, 2019. 6, 7, 8

[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 3, 6

[13] T. Chen, X. Zhai, M. Ritter, M. Lucic, and N. Houlsby. Self-supervised gans via auxiliary rotation loss. In *CVPR*, 2019. 6

[14] C. d'Autume, S. Ruder, L. Kong, and D. Yogatama. Episodic memory in lifelong language learning. In *NeurIPS*, 2019. 6

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1:1–38, 1991. 4

[16] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 1

[17] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2016. 6

[18] S. Farquhar and Y. Gal. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*, 2019. 6

[19] Enrico Fini, Stphane Lathuilire, Enver Sangineto, Moin Nabi, and Elisa Ricci. Online continual learning under extremem memory constraints. In *ECCV*, 2020. 5

[20] R. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. 1, 2

[21] Y. Ge, D. Chen, and H. Li. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *ICLR*, 2020. 6

[22] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 6

[23] J. Goldberger and E. Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017. 6

[24] J.B. Grill, F. Strub, F. Altche, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020. 1, 3

[25] G. Gupta, K. Yadav, and L. Paull. La-maml: Look-ahead meta learning for continual learning. In *NeurIPS*, 2020. 6

[26] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018. 6

[27] J. Han, P. Luo, and X. Wang. Deep self-learning from noisy labels. In *ICCV*, 2019. 6

[28] H. Harutyunyan, K. Reing, G. V. Steeg, and A. Galstyan. Improving generalization by controlling label-noise information in neural network weights. In *ICML*, 2020. 3, 6

[29] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *2019 International Conference on Robotics and Automation (ICRA)*, 2019. 1, 5

[30] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 3, 6

[31] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *NIPS*, 2018. 1, 6

[32] J. Huang, L. Qu, R. Jia, and B. Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *ICCV*, 2019. 6

[33] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Comput*, 3:79–87, 1991. 4

[34] K. Javed and M. White. Meta-learning representations for continual learning. In *NeurIPS*, 2019. 5

[35] L. Jiang, D. Huang, M. Liu, and W. Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *ICML*, 2020. 6

[36] L. jiang, Z. Zhou, T. Leung, L. Li, and L. Fei-Fei. Mentornet:learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018. 1, 6

[37] D. Kim, J. Jeong, and G. Kim. Imbalanced continual learning with partioning reservoir sampling. In *ECCV*, 2020. 1, 5, 6, 7

[38] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. In *Proceedings of the National Academy of Sciences*, 2017. 1, 5, 7

[39] J. Kremer, F. Sha, and C. Igel. Robust active label correction. In *AISTATS*, 2018. 6

[40] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Computer Science Department, University of Toronto, 2009. 2, 6

[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient based learning applied to document recognition. In *IEEE*, 1998. 2, 6

[42] K. Lee, X. He, L. Zhang, and L. Yang. Cleannet: Transfer learning for scalable image classfier training with label noise. In *CVPR*, 2018. 1

[43] S. Lee, J. Ha, D. Zhang, and G. Kim. A neural dirichlet process mixture model for task-free continual learning. In *ICLR*, 2020. 1, 2, 5, 6

[44] T. Lesort, A. Gepperth, A. Stoian, and D. Filliat. Marginal replay vs conditional replay for continual learning. In *IJCANN*, 2019. 1

[45] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020. 6

[46] J. Li, Y. Wong, Q. Zhao, and M. Kankanhalli. Learning to learn from noisy labeled data. In *CVPR*, 2019. 6

[47] J. Li, C. Xiong, and S. Hoi. Mopro: Webly supervised learning with momentum prototypes. In *ICLR*, 2021. 6

[48] J. Li, P. Zhou, C. Xiong, R. Socher, and S. C. H. Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2020. 6

[49] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv: 1708.02862*, 2017. 2, 6

[50] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L. Li. Learning from noisy labels with distillation. In *ICCV*, 2017. 6

[51] Z. Li and D. Hoiem. Learning without forgetting. In *ECCV*, 2016. 5

[52] D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017. 1, 5

[53] Y. Lyu and I.W. Tsang. Curriculum loss: Robust learning and generalization against label corruption. In *ICLR*, 2020. 2

[54] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. M. Erfani, S. Xia, S. Wijewickrema, and J. Bailey. Dimensionality-driven learning with noisy labels. In *ICML*, 2018. 3, 6

[55] E. Malach and S. Shalev-Shwartz. Decoupling "when to update" from "how to update". In *NeurIPS*, 2017. 6

[56] D. Mandal, S. Bharadwaj, and S. Biswas. A novel self-supervised re-labeling approach for training with noisy labels. In *WACV*, 2020. 6

[57] M. McCloskey and N. J. Cohen. Catastrophic interference in conncectionist networks. *Psychology of learning and motivation*, 24:109–265, 1989. 1, 2

[58] D. T. Nguyen, C. K. Mummadi, T. P. N. Ngo, T. H. P. Nguyen, L. Beggel, and T. Brox. Self: Learning to filter noisy labels with self-ensembling. In *ICLR*, 2019. 6

[59] J Nieminen. On the centrality in a graph. *Scandinavian Journal of Psychology*, 1:332–336, 1974. 4

[60] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2017. 6

[61] P. Ostyakov, E. Logacheva, R. Suvorov, V. Aliev, G. Sterkin, O. Khomenko, and S. I. Nikolenko. Label denoising with large ensembles of heterogeneous neural networks. In *ECCV*, 2018. 1

[62] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 6

[63] G. Patrini, A. Rozza, A. Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: a loss correction approach. In *CVPR*, 2017. 6

[64] G. Pleiss, T. Zhang, E. R. Elenberg, and K. Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In *NIPS*, 2020. 1, 3, 6, 7, 8

[65] A. Prabhu, P. H.S. Torr, and P. K. Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, 2019. 1, 2, 6, 7, 8

[66] Aljundi Rahaf, Min Lin, Baptiste Goujaud, and Bengio Yoshua. Gradient based sample selection for online continual learning. In *NeurIPS*, 2019. 1, 2, 5, 6

[67] R. Ratcliff. Conncectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Pscyhological review*, 97(2):285–308, 1990. 1, 2

[68] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018. 1, 6, 7, 8

[69] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *ICLR*, 2019. 2, 5, 7

[70] D. Rolnick, A. Ahuja, J. Schwarz, T. P. Lillicrap, and G. Wayne. Experience replay for continual learning. In *NeurIPS*, 2019. 1

[71] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 1, 5

[72] Y. Shen and S. Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *ICML*, 2019. 1, 6

[73] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. In *NeurIPS*, 2017. 1, 5

[74] H. Song, M. Kim, and J. Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, 2019. 6

[75] L. R. Squire. Two forms of human amnesia: an analysis of forgetting. *Journal of Neuroscience*, 6:635–640, 1981. 2

[76] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018. 6

[77] B. Tang and D. S. Matteson. Graph-based continual learning. In *ICLR*, 2021. 6

[78] S. Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in neural information processing systems*, 1996. 2

[79] G. M. van de Ven and S. T. Andreas. Three scenarios for continual learning. In *NeurIPS Continual Learning workshop*, 2019. 6

[80] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, 2017. 6

[81] J. S. Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985. 7

[82] R. von Mises and H. Pollaczek-Geiringer. Practical methods of solving equations. *Journal of Applied Mathematics and Mechanics*, 9:152–164, 1929. 4

[83] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S. Xia. Iterative learning with open-set noisy labels. In *CVPR*, 2018. 6

[84] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey. Symmetric cross entropy for robust learning with noisy labels. In *ICCV*, 2019. 1, 2, 6, 7, 8

[85] H. Wei, L. Feng, X. Chen, and B. An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, 2020. 1, 6, 7, 8

[86] Y. Xu, P. Cao, Y. Kong, and Y. Wang. L_dmi: An information-theoretic noise-robust loss function. In *NeurIPS*, 2019. 6

[87] M. Ye, X. Zhang, P. C. Yuen, and S. Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, 2019. 6

[88] K. Yi and J. Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *CVPR*, 2019. 6, 7, 8

[89] J. Yoon, E. Yang, J. Lee, and S. J. Hwang. Lifelong learning with dynamically expandable networks. In *ICLR*, 2018. 1, 5

[90] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, 2019. 6

[91] F. Zenke, B. Poole, and S. Ganguli. Continual learning through syanptic intelligence. In *ICML*, 2017. 1

[92] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017. 6

[93] HaiYang Zhang, XiMing Xing, and Liang Liu. Dualgraph: A graph-based method for reasoning about label noise. In *CVPR*, 2021. 6

[94] S. Zhang and R. Sutton. A deeper look at experience replay. *arXiv preprint arXiv:1712.01275*, 2017. 2

[95] Yaobin Zhang, Weihong Deng, Mei Wang, Jiani Hu, Xian Li, Dongyue Zhao, and Dongchao Wen. Global-local gcn: Large-scale label noise cleansing for face recognition. In *CVPR*, 2020. 6

[96] Z. Zhang and M. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NIPS*, 2018. 1, 6