DUEL: Duplicate Elimination on Active Memory for Self-Supervised Class-Imbalanced Learning

Won-Seok Choi[1], Hyundo Lee[1], Dong-Sig Han[1], Junseok Park[1], Heeyeon Koo[2], Byoung-Tak Zhang[1,3,*]

[1]Seoul National University
[2]Yonsei University
[3]AI Institute of Seoul National University (AIIS)
{wchoi, hdlee, dshan, jspark, btzhang}@bi.snu.ac.kr

Abstract

Recent machine learning algorithms have been developed using well-curated datasets, which often require substantial cost and resources. On the other hand, the direct use of raw data often leads to overfitting towards frequently occurring class information. To address class imbalances cost-efficiently, we propose an active data filtering process during self-supervised pre-training in our novel framework, Duplicate Elimination (DUEL). This framework integrates an active memory inspired by human working memory and introduces distinctiveness information, which measures the diversity of the data in the memory, to optimize both the feature extractor and the memory. The DUEL policy, which replaces the most duplicated data with new samples, aims to enhance the distinctiveness information in the memory and thereby mitigate class imbalances. We validate the effectiveness of the DUEL framework in class-imbalanced environments, demonstrating its robustness and providing reliable results in downstream tasks. We also analyze the role of the DUEL policy in the training process through various metrics and visualizations.

Introduction

Recent machine learning algorithms are heavily influenced by the quantity and quality of data. However, when agents collect data in real-world environments, the class distribution of the unprocessed data is long-tailed, indicating that data from certain classes are acquired much more frequently than others (Liu et al. 2019). When trained on such raw data without any processing, deep learning models tend to overfit to these *frequent* classes. Therefore, adaptive data refinement during the training process is essential to mitigate class imbalances cost-efficiently. Traditional methods have been developed based on resampling (Buda, Maki, and Mazurowski 2018; Pouyanfar et al. 2018) and reweighting (Cao et al. 2019; Cui et al. 2019; Tan et al. 2020) techniques. However, these methods require class information for each data point, which increases the cost of preprocessing and labeling. Even semi-supervised approaches (Wei et al. 2021; Kim et al. 2020) still require a fine-tuned support set that can reflect the target data distribution. To address these issues, recent research (Yang and Xu 2020; Liu et al. 2021) has proposed self-supervised pretraining techniques that can be trained with minimally processed data and demonstrate improved performance in class-imbalanced environments.

Self-supervised learning (Chen et al. 2020a; He et al. 2020; Zbontar et al. 2021), which is the modernized form of metric learning (Khosla et al. 2020; Sohn 2016), has the advantage of acquiring positive samples via augmentation which reflects the inductive bias in the data. Similarly, by using data in either a mini-batch or a memory as negative samples, SSL methods do not require class information to collect such negatives. However, in a class-imbalanced environment, we have observed that the relationships within and between classes are unevenly reflected when training an SSL model. This imbalance leads to performance degradation in both mini-batch and memory-based methods. Based on this empirical evidence, we claim that an active memory that can alleviate the imbalances between latent classes is necessary for self-supervised class-imbalanced learning.

In this context, we mimic a well-known human cognitive process to achieve an active memory. Human working memory (Baddeley and Logie 1999; Baddeley 2012) is a prominent cognitive concept that explains how humans deal with extreme class-imbalances via an active data filtering process. Figure 1.A shows the mechanism of human working memory. The Central Executive System (CES), which is a supervisory subsystem of working memory, *inhibits dominant information* (Miyake et al. 2000; Wongupparaj, Kumari, and Morris 2015) from perceived data and remembers it while maximizing the amount of information (Baddeley and Logie 1999). These cognitive phenomena support our hypothesis that eliminating the most duplicated data will increase the distinctiveness information within memory.

To compute the amount of the information, we define *distinctiveness* information, which measures how different a data point is from other data points. We also introduce Hebbian Metric Learning (HML) which directly optimizes *distinctiveness* information while reducing information among co-fired similar data inspired by the characteristics of Hebbian learning (Hebb 2005; Löwel and Singer 1992). We show that to generalize HML in class-imbalanced environments, an active memory is essential. In this case, a policy for managing the memory should maximize the distinctiveness information in the memory.

*Corresponding author.

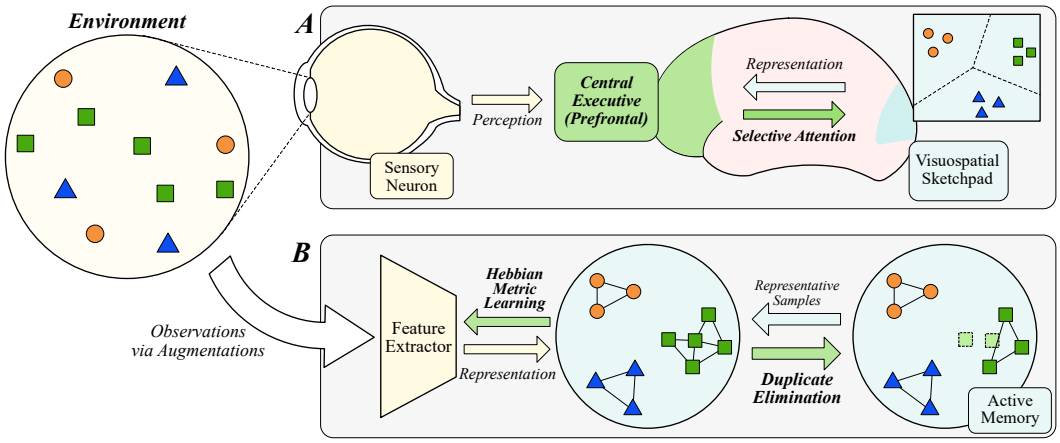


Figure 1: Visualizations of the concepts of working memory and our proposed DUEL framework. (A) Real-world agent *perceives* data from the environment and maps the representation to solve the task. Working memory finds semantically duplicated signals and reduces them to maximizes the total amount of information. (B) Inspired by this cognitive process, we design the Duplicate Elimination (DUEL) framework. With mutual duplication probability, the representations form a graph structure (center) and are filtered out (right) to gradually maximize the distinctiveness information.

As an implementation of our method, we propose the Duplicate Elimination (DUEL) framework, a novel SSL framework tailored for class-imbalanced environments. Figure 1.B provides a conceptual visualization of the proposed DUEL framework. The framework consists of two components: an active memory and a feature extractor. The feature extractor is trained with both the current data and the additional data from the active memory, while the active memory eliminates the most redundant data with the extracted representations. By iteratively updating both components, the DUEL framework can extract robust representations even in highly class-imbalanced environments.

This study provides the following contributions:

- **Memory-integrated Hebbian Metric Learning.** We define HML which optimizes the information-based metric between the data from a Hebbian perspective. We show that an active memory that maximizes distinctiveness information is essential to extend HML to class-imbalanced environments.
- **Memory Management Policy.** Inspired by working memory, we design a memory management policy that eliminates the most duplicated element in the memory.
- **DUEL Framework.** We propose the DUEL framework for self-supervised class-imbalanced learning. To simulate class-imbalanced environments, we assume that one *dominant* class occurs more frequently than others. In class-imbalanced environments, performance degradation has been observed with conventional self-supervised learning methods. On the other hand, even with the dramatically class-imbalanced data, the DUEL framework maintains stable performance in downstream tasks. We also validate the DUEL framework with more realistic environments with long-tailed class distributions and observe consistent results.

Revisiting Metric Learning from a Hebbian-based Perspective

In this section, we discuss Hebbian Metric Learning (HML), which allows us to represent the optimization problem of both the feature extractor and the memory from the same perspective. HML consists of Hebbian information and distinctiveness information terms, which aim to make representations of data with the same latent class similar while maximizing the diversity of information.

Problem Definition

The data distribution $\mathcal{D}$ is a joint distribution of the observation $x \in \mathcal{X}$ and its corresponding *latent class* $c \in \mathcal{C}$ (Saunshi et al. 2019; Ash et al. 2021; Awasthi, Dikkala, and Kamath 2022). Data with each latent class c has a distinct data distribution $\mathcal{D}_c$, and latent classes constitute the class distribution $c \sim \rho$. In this case, the joint distribution $p(x, c)$ can be expressed as follows:

$$p(x, c) := \rho(c) \cdot \mathcal{D}_c(x).$$

Our objective is to fit an estimated distribution $q(x, c; f)$ with a feature extractor $f : \mathcal{X} \to \mathcal{Z}$ to the true distribution $p(x, c)$ by minizing the Kullback-Leibler divergence $D_{KL}(p(x, c)||q(x, c; f))$. However, since latent class is not directly accessible, indirect methods such as metric learning is needed to compute $q(x, c; f)$.

Hebbian Metric Learning

To represent $p(x, c)$ and $q(x, c; f)$ without directly using latent class, we define the probability that two data samples share the same latent class as mutual duplication probability. In this case, $q(c_i = c_j|x_i, x_j; f)$ can be computed through a similarity metric of the representations of two data samples in the latent space.

Definition 1 (Mutual duplication probability)**.** Let (x_i, c_i), $(x_j, c_j) \sim \mathcal{D}$. The mutual duplication probability $q(c_i = c_j | x_i, x_j; f)$ with feature extractor f is defined as follows:

$$q(c_i = c_j | x_i, x_j; f) := \text{sim}^*(f(x_i), f(x_j)). \quad (1)$$

sim^* denotes an arbitrary metric function which satisfies the property as the probability: the range should be bounded to $[0, 1]$. With mutual duplication probability, the duplication density functions P and Q are derived via Bayes' theorem.

$$\begin{aligned} P(x_i, x_j) &:= p(x_i, x_j | c_i = c_j) \\ &= \frac{p(c_i = c_j | x_i, x_j) p(x_i) p(x_j)}{\mathbb{E}_{x_k \sim \mathcal{D}} [p(c_i = c_k | x_i, x_k)]} \end{aligned}$$

$$\begin{aligned} Q(x_i, x_j; f) &:= q(x_i, x_j | c_i = c_j; f) \\ &= \frac{q(c_i = c_j | x_i, x_j; f) p(x_i) p(x_j)}{\mathbb{E}_{x_k \sim \mathcal{D}} [q(c_i = c_k | x_i, x_k; f)]} \end{aligned}$$

P and Q represent normalized joint distributions of two data samples sharing the same class. Through the use of *message passing* (Pearl 1988), we can represent the two joint distributions, $p(x, c)$ and $q(x, c; f)$, using their density functions P and Q. In Proposition 1, we show that minimizing $D_{\text{KL}}(p(x, c) || q(x, c; f))$ is equivalent to minimizing $D_{\text{KL}}(P||Q)$ and it becomes Hebbian Metric Learning.

Proposition 1 (Hebbian Metric Learning)**.** Minimizing $D_{\text{KL}}(p(x, c) || q(x, c; f))$ is equivalent to minimizing $\mathcal{L}_{\text{HML}}(f; \mathcal{D})$, which can be derived as:

$$\begin{aligned} \arg\min_f D_{\text{KL}}(p||q) &= \arg\min_f D_{\text{KL}}(P||Q) \\ &= \arg\min_f \underbrace{(\mathcal{I}_h(f; \mathcal{D}) - \mathcal{I}_d(f; \mathcal{D}))}_{\mathcal{L}_{\text{HML}}(f; \mathcal{D})} \end{aligned}$$

where $\mathcal{I}_h(f; \mathcal{D})$ and $\mathcal{I}_d(f; \mathcal{D})$ are denoted as *Hebbian* information and *Distinctiveness* information respectively.

$$\mathcal{I}_h(f; \mathcal{D}) := \mathbb{E}_{x_i \sim \mathcal{D}} [I_h(x_i; f, \mathcal{D})] \quad (2)$$

$$\mathcal{I}_d(f; \mathcal{D}) := \mathbb{E}_{x_i \sim \mathcal{D}} [\mathcal{I}_d(x_i; f, \mathcal{D})] \quad (3)$$

$$I_h(x_i; f, \mathcal{D}) := \mathbb{E}_{x_j \sim \mathcal{D}_i^+} [-\log q(c_i = c_j | x_i, x_j; f)]$$

$$\mathcal{I}_d(x_i; f, \mathcal{D}) := -\log \left(\mathbb{E}_{x_j \sim \mathcal{D}} [q(c_i = c_j | x_i, x_j; f)] \right)$$

$\mathcal{D}_i^+$ represents the distribution of data that belong to the same latent class of x_i. For each data x_i, Hebbian information $I_h(x_i; f, \mathcal{D})$ is defined as mean information of mutual duplication probability with positive samples from $\mathcal{D}_i^+$. In Hebbian learning (Hebb 2005; Löwel and Singer 1992), the learning process strengthens the connections between similar data, which means that the Hebbian information between the two data should be minimized.

On the other hand, for each data x_i, distinctiveness information $\mathcal{I}_d(x_i; f, \mathcal{D})$ is estimated information of the proportion of class c_i from the distribution $\mathcal{D}$. The expected value of distinctiveness information $\mathcal{I}_d(x_i; f, \mathcal{D})$ becomes a measurement indicating how diversely the latent class information is distributed within the data distribution. For agents, it is essential to acquire as much information as possible from the observations in the given environment to form diverse representations. This property, known as the distinctiveness effect (Parker, Wilding, and Akerman 1998; Waddill and McDaniel 1998), becomes crucial in extracting the richest representation from the data.

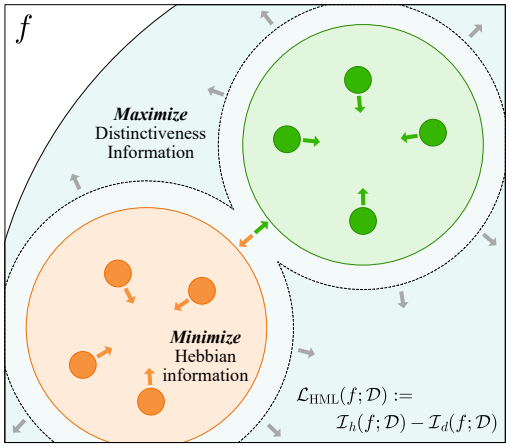


Figure 2: Conceptual Visualization of Hebbian Metric Learning. HML minimizes the Hebbian information while maximizing the distinctiveness information.

The optimization for $\mathcal{L}_{\text{HML}}$ can be interpreted as finding f^* that minimizes Hebbian information among similar data, while preventing collapsed representation by maximizing $\mathcal{I}_d$ as a regularization term. Figure 2 visualizes the concept of Hebbian Metric Learning.

Memory-integrated HML for Class-imbalanced Environment

Conventional metric learning frameworks (Khosla et al. 2020; Sohn 2016; Caron et al. 2020; Chen et al. 2020a,b; He et al. 2020) often assume that $\mathcal{D}$ is an *oracle* with evenly distributed class information: $\forall c \in \mathcal{C}, \rho_c = 1/|\mathcal{C}|$. However, when the accessible data is unrefined, data with some dominant classes may occur more frequently than others, resulting in class-imbalances which can hinder the formation of robust representations in SSL. To deal with these imbalances, maintaining a memory which stores the data selectively can be a breakthrough. Thus we extend Hebbian Metric Learning with a memory for the empirical distribution $\mathcal{D}'$ with class distribution ρ by introducing a memory $\mathcal{M}$.

Proposition 2 (HML Bound)**.** Let $\mathcal{D}$ and $\mathcal{D}'$ be the oracle and the empirical data distribution, respectively. Then the upper bound of ideal HML loss is formulated as below:

$$\begin{aligned} &\mathcal{L}_{\text{HML}}(f; \mathcal{D}) \leq \\ &\underbrace{\lambda \cdot \mathcal{I}_h(f; \mathcal{D}') - \mathcal{I}_d(f; \mathcal{D}', \mathcal{M}) + |\mathcal{I}_d(f; \mathcal{D}', \mathcal{M}) - \mathcal{I}_d(f; \mathcal{D})|}_{\mathcal{L}_{\text{M-HML}}(f, \mathcal{M}; \mathcal{D}')} \end{aligned} \quad (4)$$

where $\mathcal{I}_d(f; \mathcal{D}', \mathcal{M}) := \mathbb{E}_{x_i \sim \mathcal{D}'} [\mathcal{I}_d(x_i; f, \mathcal{M})]$ denotes the empirical distinctiveness information in the memory $\mathcal{M}$ and $\lambda = 1/(|\mathcal{C}| \cdot \rho_{\min})$.

The proof is provided in Appendix A. We denote the upper bound in Equation 4 as $\mathcal{L}_{\text{M-HML}}(f, \mathcal{M}; \mathcal{D}')$, and adopt it as an objective function to minimize. In Theorem 1, we show that the optimal feature extractor f^* that minimizes $\mathcal{L}_{\text{M-HML}}$ with optimal memory $\mathcal{M}^*$ is also optimal for $\mathcal{L}_{\text{HML}}$ with oracle data distribution.

Theorem 1 (Optimality of M-HML)**.** *Assume that an optimal memory $\mathcal{M}^* \simeq \mathcal{D}$ exists. With the memory $\mathcal{M}^*$, the ideal loss $\mathcal{L}_{HML}$ and empirical loss $\mathcal{L}_{M\text{-}HML}$ shares the optimal feature extractor f^*:*

$$\mathcal{L}_{M\text{-}HML}(f^*, \mathcal{M}^*; \mathcal{D}') = \mathcal{L}_{HML}(f^*; \mathcal{D}) = -\log|\mathcal{C}|$$

where mutual duplication probability with f^* satisfies the following property:

$$q(c_i = c_j | x_i, x_j; f^*) = \begin{cases} 1 & c_i = c_j \\ 0 & c_i \neq c_j. \end{cases}$$

Proof Sketch. We show that the bound of difference between two losses becomes zero with optimal feature extractor f^* and memory $\mathcal{M}^*$. The bound contains two terms: $|\lambda \cdot \mathcal{I}_h(f; \mathcal{D}') - \mathcal{I}_h(f; \mathcal{D})|$ and $|\mathcal{I}_d(f; \mathcal{D}', \mathcal{M}) - \mathcal{I}_d(f; \mathcal{D})|$.

We find f which satisfies $|\lambda \cdot \mathcal{I}_h(f; \mathcal{D}') - \mathcal{I}_h(f; \mathcal{D})| = 0$ by setting $q(c_i = c_j | x_i, x_j; f) = 1$ for $c_i = c_j$. We show that $|\mathcal{I}_d(f^*; \mathcal{D}', \mathcal{M}^*) - \mathcal{I}_d(f^*; \mathcal{D})| = 0$ with optimal feature extrator f^*. Then $|\mathcal{L}_{\text{M-HML}}(f^*, \mathcal{M}; \mathcal{D}') - \mathcal{L}_{\text{HML}}(f^*; \mathcal{D})| = 0$ is satisfied and $\mathcal{L}_{\text{HML}}(f^*; \mathcal{D}) = -\log|C|$.

By utilizing Theorem 1 and Proposition 1, the optimal feature extractor f^* which minimizes $D_{\text{KL}}(p||q)$ is also optimal for $\mathcal{L}_{\text{M-HML}}(f^*, \mathcal{M}^*; \mathcal{D}')$ with the memory $\mathcal{M}^*$. In the next section, we describe a procedural methodology to effectively optimize $\mathcal{L}_{\text{M-HML}}$ by adopting an active memory based on distinctiveness information $\mathcal{I}_d$.

Duplicate Elimination on Active Memory with Hebbian Metric Learning

Memory Management Policy

In the previous section, we propose the objective function $\mathcal{L}_{\text{M-HML}}$ for both the memory and feature extractor. Since memory is a finite set which stores limited number of the incoming data, optimizing $\mathcal{M}$ is a discrete process of deciding which data to store. Therefore, we split the objective function $\mathcal{L}_{\text{M-HML}}$ into objective function of memory while fixing the feature extractor, and objective function of feature extractor while fixing the memory. In this case, we set the memory $\mathcal{M}$ to hold K representative data points: $\mathcal{M} \in \mathcal{X}^K$.

$$f^* := \arg\min_f \ (\lambda \cdot \mathcal{I}_h(f; \mathcal{D}') - \mathcal{I}_d(f; \mathcal{D}', \mathcal{M})) \tag{5}$$

$$\begin{aligned} \mathcal{M}^* &:= \arg\min_{\mathcal{M} \in \mathcal{X}^K} |\mathcal{I}_d(f; \mathcal{D}', \mathcal{M}) - \mathcal{I}_d(f; \mathcal{D})| \\ &= \arg\max_{\mathcal{M} \in \mathcal{X}^K} \mathcal{I}_d(f; \mathcal{D}', \mathcal{M}) \end{aligned} \tag{6}$$

To remove $\mathcal{I}_d(f; \mathcal{D})$ term with oracle distribution $\mathcal{D}$ in Equation 6, we assume that $\mathcal{I}_d(f; \mathcal{D}) \geq \mathcal{I}_d(f; \mathcal{D}', \mathcal{M})$ and simplify Equation 6 as $\arg\max_{\mathcal{M} \in \mathcal{X}^K} \mathcal{I}_d(f; \mathcal{D}', \mathcal{M})$, which implies maximizing the distinctiveness information in the memory. To optimize the memory, we introduce an active memory $\mathcal{M}_\pi$ that is procedurally updated by a memory management policy π. Since finding an optimal policy π^* is NP-hard, we design its approximated policy inspired by the human cognitive process.

Algorithm 1: DUEL Framework with the policy π_{DUEL}

Model : feature extractor f_θ, memory $\mathcal{M}$
Input : empirical data distribution $\mathcal{D}'$, batch size B, memory size K, learning rate η
Output : trained feature extractor f_{θ^*}

$\theta \leftarrow \theta_0$
$\mathcal{M} \leftarrow \mathcal{M}_0$
while θ is not converged **do**
  $\{(x_b, x_b^+)\}_{b=1}^B \leftarrow \text{Sample}(\mathcal{D}')$
  Compute $\mathcal{L}_{\text{InfoNCE}}(\{(x_b, x_b^+)\}_{b=1}^B, \mathcal{M}; f_\theta)$
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{InfoNCE}}$
  for $b \in \{1, \cdots, B\}$ **do**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{x_b\}$
    $J \leftarrow \arg\min_{j \in \{1..(K+1)\}} \mathcal{I}_d(x_j; f_\theta, \mathcal{M})$ ▷ π_{DUEL}
    $\mathcal{M} \leftarrow \mathcal{M} \setminus \{x_J\}$
  end for
end while

Duplicate Elimination Policy on Active Memory

Working memory is an active memory connected to human sensory-motor neurons, allowing humans to selectively concentrate on necessary information from the environment to achieve their goals. In order to mimic the behavior of CES, which is inhibiting the dominant information (Miyake et al. 2000; Wongupparaj, Kumari, and Morris 2015), we design a memory management policy π_{DUEL} based on distinctiveness information.

Definition 2 (Duplicate Elimination)**.** Let the new data x_{new} be provided to active memory $\mathcal{M}_\pi$. The DUEL policy π_{DUEL} is a policy which chooses the J-th element $x_J \in \mathcal{M}_\pi$ with the minimum value of $\mathcal{I}_d(x_j; f, \mathcal{M}_\pi)$.

$$J = \arg\min_{j \in \{1..K\}} \mathcal{I}_d(x_j; f, \mathcal{M}_\pi) \tag{7}$$

π_{DUEL} in Definition 2 replaces the element with the least distinctiveness information, which is the most duplicated element in the memory. The replacement is carried out gradually, one element at a time. We show that the process of π_{DUEL} is *safe* in the sense that it increases total amount of information as in Appendix A. Figure 3 illustrates the behavior of π_{DUEL}. π_{DUEL} finds the densest area (green) of the latent space and ejects the most duplicated element (dotted outline). The plural region (blue) is not influenced by this replacement and leaving this region intact will increase distinctiveness information in the memory.

Since reducing memory usage and time comsumption of π_{DUEL} is crucial to implement our model, we optimize the policy π_{DUEL} to minimize the time consumption with an affordable amount of additional resources. Details of the implementation and analyses on the resource usage are in Ap-

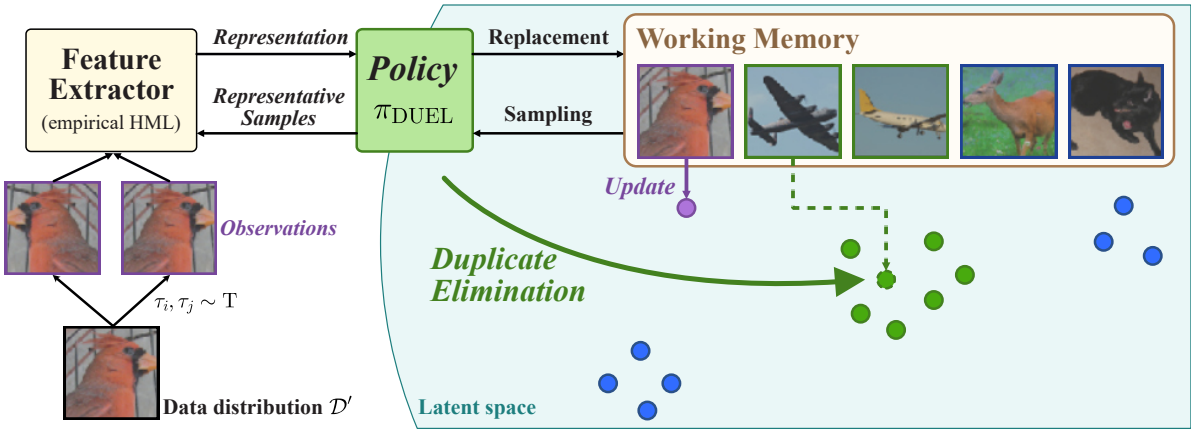


Figure 3: Visualization of general DUEL framework. Our method stores various data for the negative samples by Duplicate Elimination. The DUEL policy selects the most duplicated sample in memory (green) and replaces it with current data (purple).

pendix D. With π_{DUEL}, we now propose the Duplicate Elimination (DUEL) framework.

DUEL Framework

The DUEL framework is an self-supervised learning framework which optimizes $\mathcal{L}_{\text{M-HML}}$ in Equation 4. The procedure of our framework is shown in Figure 3. To sample positive samples from $\mathcal{D}_i^+$, we use augmentation methods (Chen et al. 2020a) rather than maintaining multiple bins of each class (Sohn 2016). Our framework can be effectively applied to class-imbalanced environment without any class information, due to the inductive bias introduced by augmentation.

Algorithm 1 summarizes the training procedure of DUEL framework. We utilize the InfoNCE loss $\mathcal{L}_{\text{InfoNCE}}$ instead of $\mathcal{L}_{\text{M-HML}}$ to train the feature extractor because $\mathcal{L}_{\text{InfoNCE}}$ is equivalent to $\mathcal{L}_{\text{M-HML}}$ under certain conditions: (1) $q(c_i = c_j|x_i, x_j; f) = \exp((f(x_i)^\top f(x_j) - 1)/\tau)$ and (2) $\lambda = 1$. More details and mathematical support regarding the relationship between our DUEL framework and conventional SSL models are provided in the Appendix B.

After each training step of the feature extractor, the duplication elimination step begins. In the duplication elimination step, selected data according to the policy π_{DUEL} is replaced with the current data. These two steps repeat iteratively until the termination condition is satisfied. We also provide the general form of memory-integrated Hebbian Metric Learning algorithm in Appendix C.

Experiments

The goal of our framework is to learn a robust representation given unrefined and instantaneous data sampled from a class-imbalanced distribution. Thus, we validate our framework in class-imbalanced environments.

Experiment setting In our experiments, we use the ResNet-50 (He et al. 2016) as a backbone of the feature extractor. We choose MoCoV2 (Chen et al. 2020b), SimCLR (Chen et al. 2020a), and Barlow Twins (Zbontar et al. 2021) as baselines. We implement our DUEL frameworks based on MoCoV2 and SimCLR by adding the DUEL process, denoted as D-MoCo and D-SimCLR respectively. Hyperparameters for all models are unified for fair comparison. After the training, we evaluate each model with downstream tasks such as linear probing with class-balanced datasets to prove each model can extract generalized representations. More details for hyperparameters and the experiments are provided in the Appendix E.

Class-imbalanced environment We design a two-step data generator with predefined datasets to describe a class-imbalanced environment. A dataset D is partitioned into D_c with each class $c \in \mathcal{C}$. In every experiment, we assume that one class, denoted as $c_{\max}$, occurs much more frequently than others. The occurrence probability of the most frequent class is denoted as $\rho_{\max}$. The probabilities of the remaining classes are identically set as $\rho_{\min} = \frac{1}{|C|-1}(1 - \rho_{\max})$. Then the data is sampled from the environment in two steps:

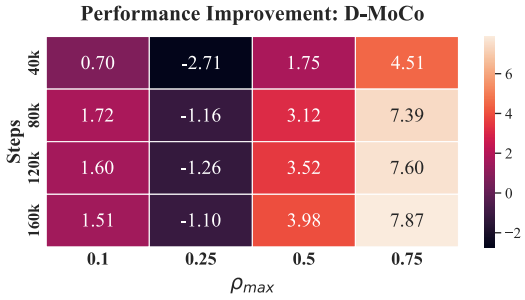
1. *Class Sampling*: $c \sim \rho$, $\rho_c = \begin{cases} \rho_{\max} & c = c_{\max} \\ \rho_{\min} & c \neq c_{\max} \end{cases}$
2. *Data Sampling*: $x \sim \text{Uniform}(D_c)$.

We utilize CIFAR-10 (Krizhevsky, Hinton et al. 2009) and STL-10 (Coates, Ng, and Lee 2011) for experiments. We also use ImageNet-LT (Liu et al. 2019), which has a long-tailed class distribution, to validate our framework in a more realistic environment. See Appendix E for more details.

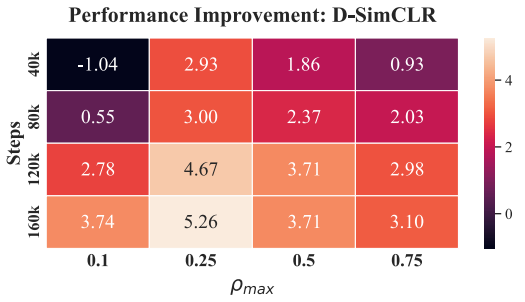
Class-imbalanced learning with SSL frameworks To validate our approaches, we first conduct experiments with conventional SSL models in class-imbalanced environments. Table 1 shows that SSL models suffer from the performance degradation, especially when the class distribution is highly imbalanced. However, our frameworks can prevent the performance loss compared to their origin models. The visualization of the performance improvement during the training process in Figure 4 implies that the DUEL process gradually improves the robustness of representations. More experiments and results are described in Appendix F.

Method	STL-10 Class Probability $\rho_{\max(\rho_{\min})}$				CIFAR-10 Class Probability $\rho_{\max(\rho_{\min})}$			
	0.1 (0.1)	0.25 (0.083)	0.5 (0.056)	0.75 (0.028)	0.1 (0.1)	0.25 (0.083)	0.5 (0.056)	0.75 (0.028)
MoCoV2	**79.59**±6.57	**79.32**±1.02	77.28±2.68	75.34±1.11	80.99±1.85	**82.50**±1.12	76.54±1.97	70.12±1.40
SimCLR	70.80±1.21	69.59±0.50	67.20±1.10	68.75±2.08	82.28±1.26	78.90±1.60	76.67±1.96	72.81±1.39
Barlow Twins	77.48±3.10	78.21±2.82	72.47±0.56	71.98±2.12	51.72±0.30	51.59±1.44	54.89±0.93	53.54±0.99
BYOL	68.42±0.80	64.04±1.41	58.78±3.49	61.82±2.77	67.87±3.26	69.32±3.52	67.50±1.63	60.40±2.40
D-MoCo (ours)	**79.20**±3.56	76.03±2.06	**78.65**±2.23	**77.23**±0.48	82.58±3.48	81.40±2.24	**80.53**±3.04	**77.99**±1.71
D-SimCLR (ours)	75.89±2.90	72.68±3.53	78.23±4.38	74.13±1.04	**82.82**±1.10	**82.37**±1.53	79.56±2.61	75.17±3.49

Table 1: Linear probing accuracies with various settings. (3 times, %)



(a) D-MoCo (Compared to MoCoV2)



(b) D-SimCLR (Compared to SimCLR)

Figure 4: Visualization of the performance enhancement in the linear probing task. In both D-MoCo and D-SimCLR, accuracies are gradually improved during the training steps. Especially in D-MoCo, the DUEL process can prevent the dramatical performance degradation with high $\rho_{\max}$.

Analysis of the robustness of representation Additionally, we measure and compare how well the representations extracted from the DUEL framework and the baseline's feature extractor cluster are formed. We use intra-class variance and inter-class similarity as measurements for this purpose. The intra-class variance and inter-class similarity are described in Equation 8 and 9, respectively.

$$\bar{\mathrm{v}}_{\text{intra}} := \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \mathbb{E}_{x_c \sim \mathcal{D}_c} \left[(\bar{r}_c^\top f(x_c) - 1)^2 \right] \tag{8}$$

$$\bar{\mathrm{s}}_{\text{inter}} := \frac{1}{|\mathcal{C}| \cdot (|\mathcal{C}| - 1)} \sum_{c \in \mathcal{C}} \sum_{c' \neq c} (\bar{r}_c^\top \bar{r}_{c'}) \tag{9}$$

Metric	Method	Class Probability $\rho_{\max(\rho_{\min})}$ 0.1 (0.1)	0.5 (0.056)	0.75 (0.028)
Class Entropy (↑)	MoCo	2.2991	1.8394	1.0523
	D-MoCo	2.2988	**2.1654**	**1.8306**
Intra-class Variance (↓)	MoCo	0.7879	0.7853	0.7689
	D-MoCo	0.7750	0.7633	0.7699
Inter-class Similarity (↓)	MoCo	−0.1005	−0.0534	0.0723
	D-MoCo	**−0.1033**	**−0.0846**	**−0.0513**

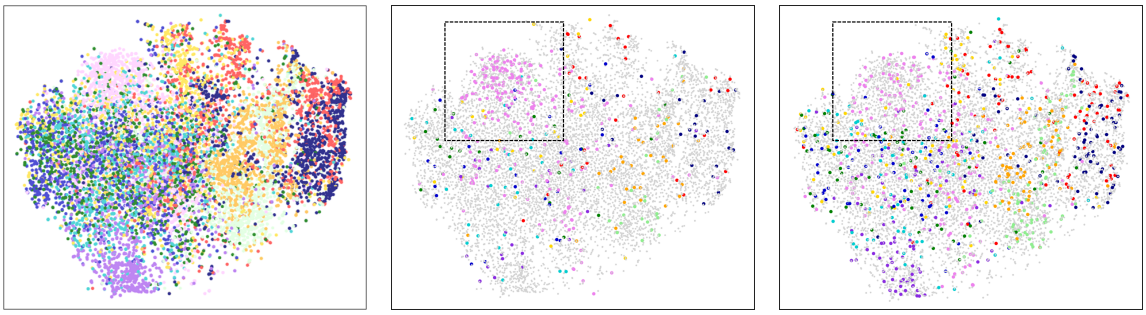
Table 2: Quantitative analysis of the behavior of MoCo and D-MoCo with various metrics. (CIFAR-10)

$\bar{r}_c$ is a centroid of each class on a hypersphere: $\bar{r}_c = \mathbb{E}_{x_c \sim \mathcal{D}_c} [f(x_c)] / ||\mathbb{E}_{x_c \sim \mathcal{D}_c} [f(x_c)]||_2$. Intra-class variance indicates how densely the representations of the same class are gathered, while inter-class similarity indicates how far apart the centroids of each class are. From the perspective of a classification task, both low intra-class variance and low inter-class similarity signify the robustness of representations. Table 2 presents the quantitative results for MoCo and D-MoCo. In both cases, the intra-class variance is preserved in every environment. However, the inter-class similarity of MoCo dramatically increases in extreme situation with $\rho_{\max} = 0.75$. It implies that our framework extracts more distinguishable representation than MoCo when the data is class-imbalanced.

The role of the DUEL policy We analyze the properties of the data stored in the memory to show that the DUEL policy can effectively mitigate the class imbalances. In Table 2, we observe that the entropy of the class distribution within the memory of D-MoCo is consistently higher than that of MoCo. This indicates that the DUEL process can maintain the diversity of the class information even in extremely class-imbalanced environments. We also visualize the policy of the DUEL framework with t-SNE (Van der Maaten and Hinton 2008) in Figure 5. Even in the presence of the frequent class (pink) (Figure 5b), the proposed framework filters out the duplicates and stores diverse data in the memory (Figure 5c).

Related Work

Class-imbalanced learning Class-imbalanced learning is a methodology for effective learning when class informa-



(a) t-SNE (D-MoCo). (b) Batch-wise observation with $\mathcal{D}'$. (c) Updated memory with π_{DUEL}.

Figure 5: t-SNE visualization of the active data filtering process with DUEL policy. (a) The representations extracted by the trained model along with their corresponding class. (b) The agent faces a dominant class (pink) that occurs more frequently than others. (c) The DUEL policy π_{DUEL} replaces duplicated data with newer data and maximizes the distinctiveness information.

tion is unevenly distributed in the data. To address this challenge, various techniques such as data resampling to smooth out class distributions (Buda, Maki, and Mazurowski 2018; Pouyanfar et al. 2018) and specialized loss functions (Cao et al. 2019; Cui et al. 2019; Tan et al. 2020) have been employed. However, these approaches have limitations, as they require class information for each data point and may struggle to perform stably in extremely class-imbalanced environments. Recent research in class-imbalanced learning (Yang and Xu 2020; Liu et al. 2021) has shown that the self-supervised pretraining technique is more robust in class-imbalanced environments, even without explicit class information. To improve adaptation to extreme class-imbalanced environments, we have proposed an SSL framework with an additional active memory.

Self-supervised learning Self-supervised learning has been proposed in different paradigms depending on the loss function and model architecture. For example, InfoNCE-based SSL models (Chen et al. 2020a,b; Oord, Li, and Vinyals 2018), for instance, can be considered as an extension of traditional metric learning that does not use class information. In the case of BYOL (Grill et al. 2020), training process is based on knowledge distillation on the student model with a teacher model that is updated with momentum. Recently, several methods have been introduced, including Barlow Twins (Zbontar et al. 2021), which perform metric learning by matching distributions on the latent space (Bardes, Ponce, and LeCun 2021; Liu et al. 2022; Chen and He 2021). To validate the DUEL framework, we compared representative models from each paradigm, primarily based on InfoNCE.

Dealing with negative samples In contrastive SSL, numerous studies have highlighted the significant impact of properly configuring negative samples on the model performance. Since self-supervised learning fundamentally extracts negative samples in an i.i.d. manner, the influence of the number of negative samples on training has been investigated (Arora et al. 2019; Ash et al. 2021; Awasthi, Dikkala, and Kamath 2022). Subsequently, techniques such as generating virtual data using interpolation between samples (Kalantidis et al. 2020) and applying penalties to elements within negative samples that share the same class information for debiasing have been employed (Chuang et al. 2020). In addition, methods using mutual dependencies among elements within the same batch to adjust the degree of learning for each triplet have also been proposed (Tian 2022). Our filtering algorithm has improved performance by encouraging the maximization of distinctiveness information among negative samples, especially for data distributions containing class imbalances.

Conclusion

With respect to self-supervised class-imbalanced learning, we mainly claim that an active memory is essential to robustly generalize to instantaneous and class-imbalanced data without class information. We first introduce the Hebbian Metric Learning which optimizes both distinctiveness and Hebbian information. As an implementation of memory-integrated HML, we propose the Duplicate Elimination framework inspired by the working memory. We validate the DUEL framework with class-imbalanced environments and analyze the behavior of the framework. Our novel framework gradually maximizes the distinctiveness information in the memory, which leads to the preservation of the robustness despite dramatic class imbalance.

Limitations As we discuss, finding the optimal memory management policy π^* is difficult to achieve in practice. Although the DUEL policy provides sufficient robustness, one can argue that our policy does not perform *optimally* in some situations. We claim that further investigations on HML and distinctiveness information will be pivotal in comprehending the behavior of SSL and determining the best policy.

Acknowledgments

This work was partly supported by the IITP (2021-0-02068-AIHub/15%, 2021-0-01343-GSAI/20%, 2022-0-00951-LBA/25%, 2022-0-00953-PICA/25%) and NRF (RS-2023-00274280/15%) grant funded by the Korean government.

References

Arora, S.; Khandeparkar, H.; Khodak, M.; Plevrakis, O.; and Saunshi, N. 2019. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*.

Ash, J. T.; Goel, S.; Krishnamurthy, A.; and Misra, D. 2021. Investigating the role of negatives in contrastive representation learning. *arXiv preprint arXiv:2106.09943*.

Awasthi, P.; Dikkala, N.; and Kamath, P. 2022. Do more negative samples necessarily hurt in contrastive learning? In *International Conference on Machine Learning*, 1101–1116. PMLR.

Baddeley, A. 2012. Working memory: Theories, models, and controversies. *Annual review of psychology*, 63: 1–29.

Baddeley, A. D.; and Logie, R. H. 1999. Working memory: The multiple-component model.

Bardes, A.; Ponce, J.; and LeCun, Y. 2021. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*.

Buda, M.; Maki, A.; and Mazurowski, M. A. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106: 249–259.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33: 9912–9924.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020b. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15750–15758.

Chuang, C.-Y.; Robinson, J.; Lin, Y.-C.; Torralba, A.; and Jegelka, S. 2020. Debiased contrastive learning. *Advances in neural information processing systems*, 33: 8765–8775.

Coates, A.; Ng, A.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 215–223. JMLR Workshop and Conference Proceedings.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9268–9277.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33: 21271–21284.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hebb, D. O. 2005. *The organization of behavior: A neuropsychological theory*. Psychology press.

Kalantidis, Y.; Sariyildiz, M. B.; Pion, N.; Weinzaepfel, P.; and Larlus, D. 2020. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33: 21798–21809.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33: 18661–18673.

Kim, J.; Hur, Y.; Park, S.; Yang, E.; Hwang, S. J.; and Shin, J. 2020. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *Advances in neural information processing systems*, 33: 14567–14579.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Liu, H.; HaoChen, J. Z.; Gaidon, A.; and Ma, T. 2021. Self-supervised learning is more robust to dataset imbalance. *arXiv preprint arXiv:2110.05025*.

Liu, X.; Wang, Z.; Li, Y.-L.; and Wang, S. 2022. Self-Supervised Learning via Maximum Entropy Coding. *Advances in Neural Information Processing Systems*, 35: 34091–34105.

Liu, Z.; Miao, Z.; Zhan, X.; Wang, J.; Gong, B.; and Yu, S. X. 2019. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2537–2546.

Löwel, S.; and Singer, W. 1992. Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity. *Science*, 255(5041): 209–212.

Miyake, A.; Friedman, N. P.; Emerson, M. J.; Witzki, A. H.; Howerter, A.; and Wager, T. D. 2000. The unity and diversity of executive functions and their contributions to complex "frontal lobe" tasks: A latent variable analysis. *Cognitive psychology*, 41(1): 49–100.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Parker, A.; Wilding, E.; and Akerman, C. 1998. The von Restorff effect in visual object recognition memory in humans and monkeys: The role of frontal/perirhinal interaction. *Journal of cognitive neuroscience*, 10(6): 691–703.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann.

Pouyanfar, S.; Tao, Y.; Mohan, A.; Tian, H.; Kaseb, A. S.; Gauen, K.; Dailey, R.; Aghajanzadeh, S.; Lu, Y.-H.; Chen, S.-C.; et al. 2018. Dynamic sampling in convolutional neural networks for imbalanced data classification. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, 112–117. IEEE.

Saunshi, N.; Plevrakis, O.; Arora, S.; Khodak, M.; and Khandeparkar, H. 2019. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, 5628–5637. PMLR.

Sohn, K. 2016. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29.

Tan, J.; Wang, C.; Li, B.; Li, Q.; Ouyang, W.; Yin, C.; and Yan, J. 2020. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11662–11671.

Tian, Y. 2022. Understanding Deep Contrastive Learning via Coordinate-wise Optimization. In *Advances in Neural Information Processing Systems*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Waddill, P. J.; and McDaniel, M. A. 1998. Distinctiveness effects in recall. *Memory & Cognition*, 26: 108–120.

Wei, C.; Sohn, K.; Mellina, C.; Yuille, A.; and Yang, F. 2021. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10857–10866.

Wongupparaj, P.; Kumari, V.; and Morris, R. G. 2015. The relation between a multicomponent working memory and intelligence: The roles of central executive and short-term storage functions. *Intelligence*, 53: 166–180.

Yang, Y.; and Xu, Z. 2020. Rethinking the value of labels for improving class-imbalanced learning. *Advances in neural information processing systems*, 33: 19290–19301.

Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; and Deny, S. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, 12310–12320. PMLR.