DUEL: Adaptive Duplicate Elimination on Working Memory for Self-Supervised Learning

Won-Seok Choi[1], **Dong-Sig Han**[1], **Hyundo Lee**[1], **Junseok Park**[1], **Byoung-Tak Zhang**[12]
[1]Seoul National University, [2]AIIS
{wchoi,dshan,hdlee,jspark,btzhang}@bi.snu.ac.kr

Abstract

In Self-Supervised Learning (SSL), it is known that frequent occurrences of the *collision* in which target data and its negative samples share the same class can decrease performance. Especially in real-world data such as crawled data or robot-gathered observations, collisions may occur more often due to the duplicates in the data. To deal with this problem, we claim that sampling negative samples from the adaptively debiased distribution in the memory makes the model more stable than sampling from a biased dataset directly. In this paper, we introduce a novel SSL framework with adaptive Duplicate Elimination (DUEL) inspired by the *human working memory*. The proposed framework successfully prevents the downstream task performance from degradation due to a dramatic inter-class imbalance.

1 Introduction

In Self-Supervised Learning (SSL), there is a possibility that target data and its negative samples' class information are partially duplicated during the sampling process. This phenomenon is called *collision* and it leads to the degradation of the latent space's representability [1, 2]. When data is provided in the real world such as crawled images from the web or robot-gathered vision data, the agent may face many duplicates and they can cause collisions frequently.

Human working memory [3, 13, 14] has a Central Executive System (CES) which manages the limited memory efficiently to enhance the task performance. Some of the major roles of CES are as follows: *inhibition of dominant signals* and *updating recent signal* in memory [13]. In this paper, we claim that a more efficient memory with an adaptive controller akin to the human working memory is essential to reduce the collisions for more robust training with a biased dataset.

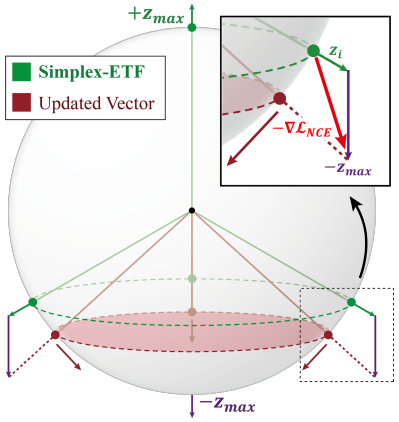

Figure 1: Visualization of **Observation 1**. If the dataset is biased with a class $c_{\max}$, representations of other classes (green) come closer to the opposite of $z_{\max}$ (purple).

In this paper, we first evaluate the previous SSL frameworks' robustness when the data is highly biased with a specific class. Based on the results, we introduce a novel SSL framework with adaptive Duplicate Elimination (DUEL) to imitate human working memory. The proposed framework performs training of the feature extractor and removing the most duplicated data from the memory with the current feature extractor simultaneously. We compare our proposed framework to previous popular frameworks in a biased dataset adapted from the common vision dataset.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Related Work

The main difference between the Self-Supervised Learning frameworks is the method of selecting the negative samples. SimCLR [4, 5] uses other data in the same batch as negative samples. MoCo [10, 7] has the external memory to store representative information. BYOL [8] and SimSiam [6] can be trained with only positives for training by using bootstrapping.

Recently, there were analyses on the relationship between Noise Contrastive Estimation (NCE) loss and supervised loss. Ash et al. [1] found that the upper bound of the supervised learning loss can be derived with two different terms: NCE loss without collisions and intra-class variances. They claimed when the collision occurs frequently, it may increase the intra-class variance and loosen the upper bound. Awasthi et al. [2] followed the formulas from Ash et al. [1] and proved the representations which optimizes the NCE loss form the simplex *Equiangular Tight Frame (ETF)*.

Definition 1 (Simplex-ETF)**.** The normalized representations $z_1, \cdots, z_k$ and their classes $c_1, \cdots, c_k \in \mathcal{C}$ form simplex-ETF when they satisfy the following property.

$$\forall i, j,\ z_i^T z_j = \begin{cases} 1 & c_i = c_j \\ -1/|\mathcal{C}| & c_i \neq c_j \end{cases} \tag{1}$$

In this work, we used the simplex-ETF to analyze the robustness of SSL frameworks.

3 Analysis on InfoNCE Loss with Biased Dataset

In this section, we analyze which shape the representations form to optimize the NCE loss with a biased dataset by expanding previous works' approaches [1, 2, 11]. Let a dataset $\mathcal{D}$ contain a pair of data x and its implicit class $\hat{c}$, $d = (x, \hat{c}) \sim \mathcal{D}$. The implicit class $\hat{c}$ has its marginal distribution $\hat{c} \sim \rho$. In general, the positive sample d^+ has the same class $\hat{c}$ as d and negative samples $d^-_{1:k}$ are drawn in the i.i.d. manner from the same distribution ρ. NCE loss is derived as below.

$$\mathcal{L}_{\text{NCE}}(f) = \mathbb{E}_{d,d^+,d^-_{1:k}}\left[\ell\left(\{f(x)^T\left(f(x^+) - f(x_i^-)\right)\}_{i=1}^k\right)\right] \tag{2}$$

f is the feature extractor which projects the data x onto a hypersphere. ℓ is a logistic loss function $\ell(\boldsymbol{v}) = \log(1 + \sum \exp(-v_i))$ which is widely used recently.

3.1 Gradient of Representation on NCE Loss with Biased Distribution

Let a class $c_{\max}$ occurs more frequently than others. Then the probability of choosing $c_{\max}$ is $\rho_{\max}$, and otherwise $\rho_{\min} = \frac{1-\rho_{\max}}{|\mathcal{C}|-1}$. In practice, representations of data with each class form the clusters whose mean vectors represent the ETF-like shape. If mean vectors get closer to each other, the chance of overlapping among them will increase and it will decrease the downstream task performance.

Observation 1 (Non-convergence to simplex-ETF with biased data)**.** *The representations optimized by the NCE loss will not converge to the simplex-ETF when the data is biased with class $c_{\max} \in \mathcal{C}$.*

We compute the gradient of the NCE loss with respect to each vector on the simplex-ETF with the equation in Khosla et al. [11]. The gradient of each case is computed as Equation 3.

$$\frac{\partial \mathcal{L}_{\text{NCE},i}}{\partial z_i} \propto \begin{cases} z_i \cdot (-1 + \rho_{\max} - \rho_{\min}) & c_i = c_{\max} \\ -z_i + z_{\max} \cdot (\rho_{\max} - \rho_{\min}) & c_i \neq c_{\max} \end{cases} \tag{3}$$

As a result, the gradient will contain a non-zero $z_{\max}$ term when $c_i \neq c_{\max}$. This means that all vectors except $z_{\max}$ will get closer to the $-z_{\max}$ direction after updates. This indicates that frequent collision will disturb the training of general representation of the data. Details on this observation is provided in Appendix A.2. The visualization is also shown in Figure 1.

4 Adaptive Duplicate Elimination (DUEL) with Working Memory

For real-world agents such as humans, dealing with biases caused by physical accessibility is important. The working memory solves the problem by updating recent data while reducing the intensity of dominant information [13, 14]. Inspired by this paradigm, we propose a memory control policy that replaces duplicated data with current data to reduce biases in the dataset.

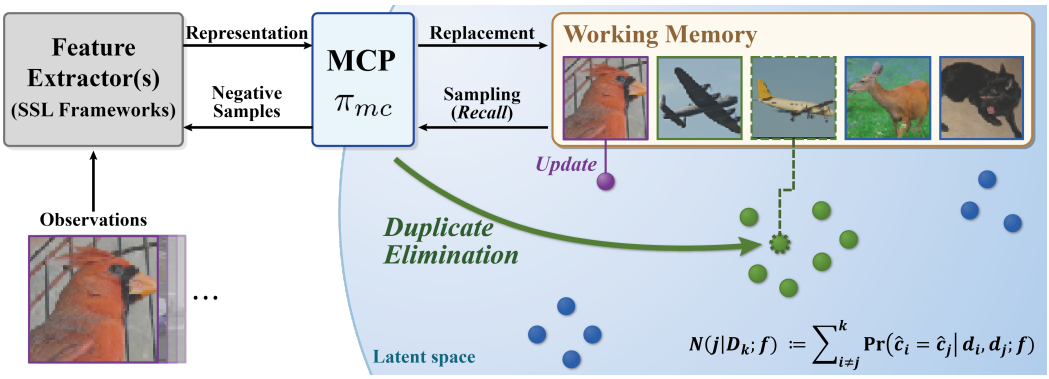

Figure 2: Visualization of general DUEL framework. Our method stores various data for the negative samples by adaptive Duplicate Elimination. A memory control policy selects the most duplicated sample in memory (green) and replaces it with current data (purple).

4.1 Memory Control Policy (MCP)

In the perspective of working memory, both storing recent data and avoiding the dominance of specific information are essential. This implies that the MCP should focus on choosing the most duplicated data for replacement and also needs a measurement of information to define the replacement criterion. In this case, the agent only can get the cosine similarity of data pairs in SSL, so we define a pair-wise collision probability with a score function to measure the estimated number of duplicates in memory.

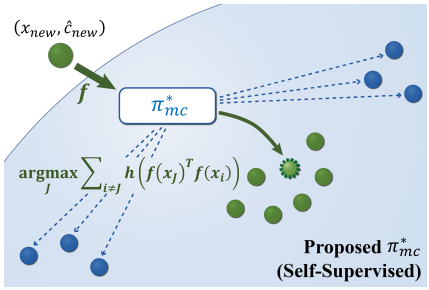

Figure 3: π^*_{mc} in SSL.

Definition 2 (Pair-wise collision probability)**.** Let there be two data $d_i = (x_i, \hat{c}_i)$ and $d_j = (x_j, \hat{c}_j)$. The probability that the implicit classes $\hat{c}_i$ and $\hat{c}_j$ are the same with feature extractor f and a score function h can be defined as below.

$$Pr(\hat{c}_i = \hat{c}_j | d_i, d_j; f) = h(f(x_i)^T f(x_j)) \tag{4}$$

There can be various h functions that satisfy the properties of cosine similarity and probability. This pair-wise collision probability can be applied to the MCP to compute the expected number of duplicates of each sample in a set of data $D_k = \{d_i\}_{i=1}^k$ with Equation 5.

$$N(j|D_k; f) := \sum_{i \neq j}^{k} Pr(\hat{c}_i = \hat{c}_j | d_i, d_j; f) \tag{5}$$

With $N(\cdot|D_k; f)$ function, we can design a simple MCP to remove samples that have duplicated information in the memory.

Definition 3 (Naïve MCP)**.** Let there be a poilcy π^*_{mc} that handles data one by one. In replacement, the policy chooses J-th sample as $\arg\max_J N(J|D_k; f)$ and replaces it with the current input.

Figure 3 shows the behavior of proposed MCP π^*_{mc}. The π^*_{mc} finds the dense area (green) of the latent space and ejects the most duplicated element (dotted outline). The sparse region (blue) is not influenced by this replacement and it will increase or maintain the variety of the memory structure.

4.2 DUEL Framework for Biased Dataset

In DUEL framework, adaptive duplicated elimination with the model f and training of the model parameters θ are executed simultaneously. The procedure of our framework is shown in Figure 2. The framework shares the training part with previous works, so main difference of our framework is choosing the most duplicated data in memory $\mathcal{M}$. For experiments, we use the DUEL framework with π^*_{mc} in Definition 3 which selects J-th element with the highest value of $N(J|\mathcal{M}; f, \theta)$. We use the simple policy that processes each data on-the-fly, but there also can be more complex policies from such as reinforcement learning that can update data in a batch at once. General DUEL algorithm and our implementation with π^*_{mc} are shown in Appendix A.1.

Table 1: Top-k accuracy. (CIFAR-10, %)

Methods	Top-1 Bias factor ($\rho_{\max}/\rho_{\min}$)				Top-10 Bias factor ($\rho_{\max}/\rho_{\min}$)			
	$\times 1.0$	$\times 3.0$	$\times 9.0$	$\times 27.0$	$\times 1.0$	$\times 3.0$	$\times 9.0$	$\times 27.0$
MoCoV2[7]	31.38	32.38	27.68	23.78	84.55	84.32	81.74	78.40
SimCLR[4]	39.85	36.94	35.58	27.17	88.38	87.20	84.34	76.98
SimSiam[6]	18.55	16.14	18.72	20.11	76.45	73.58	76.58	78.91
D-MoCo (ours)	37.12	34.45	**38.20**	28.13	87.22	85.48	**87.59**	**82.99**
D-SimCLR (ours)	**42.51**	**40.79**	37.14	**33.36**	**89.33**	**88.66**	82.49	80.51

Table 2: Downstream task accuracy. (%)

Methods	Bias factor ($\rho_{\max}/\rho_{\min}$)			
	$\times 1.0$	$\times 3.0$	$\times 9.0$	$\times 27.0$
MoCoV2	54.25	55.32	49.54	42.17
SimCLR	63.34	61.19	61.06	49.02
D-MoCo	58.15	56.59	**62.10**	52.06
	(+3.9)	(+1.27)	(+12.56)	**(+9.89)**
D-SimCLR	**65.39**	**63.69**	60.99	**56.60**
	(+2.05)	(+2.50)	(-0.07)	**(+7.58)**

Table 3: **Ablation:** h function. ($\times 27.0$, %)

Methods	Linear	Gaussian	Quadratic
D-MoCo	52.06	56.81	57.32
D-SimCLR	56.60	-	-

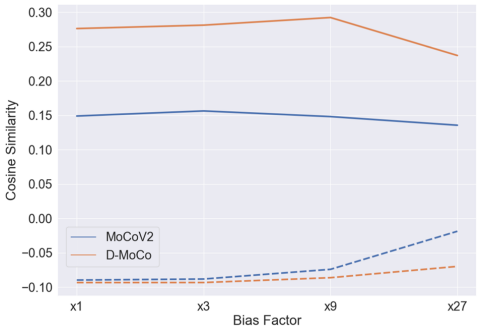

Figure 4: Average intra- (solid) and inter-class (dotted) cosine similarity. (MoCo-based)

5 Experiment

The DUEL's ultimate goal is both effective and efficient framework which can remove duplicates in memory adaptively and train its feature extractor at the same time. We design experiments to validate our approach's robustness in the biased environment.

Experiment setting. We implement the artificial dataset with biased class distribution ρ in Section 3.1: $\rho_{\max}$ with $c_{\max}$ and otherwise, $\rho_{\min}$. Both proposed and previous frameworks are compared with various bias factors $\rho_{\max}/\rho_{\min}$. More details on the training are explained in Appendix A.3.

Comparison with competitive frameworks. We first evalute our approach to the popular SSL frameworks. Table 1 shows the Top-k accuracies of each model. Our approaches outperform the previous frameworks in all environments with various bias factors. There are large gains of about 5% in average with the bias factor $\times 27$ in both D-MoCo and D-SimCLR than their origin framework respectively. Table 2 also shows our framework is robust in *every* environment including the unbiased dataset. It implies that the processed dataset still contains *implicit* biases which humans hardly discriminate, and our framework can remove them effectively. Figure 4 also supports proposed framework can extract more robust representations than original framework: average inter-class cosine similarity of proposed model is smaller, and conversely, average intra-class similarity is bigger.

Ablation study: h function. We test our framework with several h functions: Linear (default), Quadratic, and Gaussian. The visualizations of these functions are also shown in Appendix A.3. MoCo-based model performs robustly with all function types, but SimCLR-based one fails when the function is not linear. We carefully claim that this function will affect the behavior of MCP in early-stage learning, but more analysis is needed to prove this phenomenon.

6 Discussion

In this work, we suggest a novel framework with adaptive memory control which imitates the working memory's behavior. Our framework outperforms its original SSL framework, especially in a severely biased dataset. The robustness of proposed framework supports our claim that adaptive methods in gathering negative samples are essential for more stable SSL training.

Acknowledgments and Disclosure of Funding

This work was partly supported by the Institute of Information & Communications Technology Planning & Evaluation (2015-0-00310-SW.StarLab/20%, 2019-0-01371-BabyMind/20%, 2021-0-02068-AIHub/10%, 2021-0-01343-GSAI/10%, 2022-0-00951-LBA/20%, 2022-0-00953-PICA/20%) grant funded by the Korean government.

References

[1] J. Ash, S. Goel, A. Krishnamurthy, and D. Misra. Investigating the role of negatives in contrastive representation learning. In *2022 International Conference on Artificial Intelligence and Statistics*, February 2022.

[2] P. Awasthi, N. Dikkala, and P. Kamath. Do more negative samples necessarily hurt in contrastive learning? *arXiv preprint arXiv:2205.01789*, 2022.

[3] A. Baddeley. Working memory. *Science*, 255(5044):556–559, 1992.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[5] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33: 22243–22255, 2020.

[6] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758, June 2021.

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

[8] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

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

[10] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[11] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] A. Miyake, N. P. Friedman, M. J. Emerson, A. H. Witzki, A. Howerter, and T. D. Wager. 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, 2000.

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

A Appendix

A.1 Algorithms

Algorithm 1 and Algorithm 2 describe our proposed framework, called DUEL. The replacement in the working memory can be interpreted with the decision-making problem. In this case, the decisions with the policy are the indices of the data in the working memory. Various algorithms which can be separated into the smaller decision-making processes (e.g. sorting or Reinforcement Learning) can be used for the replacement procedure. Also, there is an alternative way to implement our framework: updating memory with current data first and sampling only from memory for the training. However, we do not conduct experiments with those setups because we avoid considerable modifications to the previous frameworks for a fair comparison. In this work, we have shown that our framework is general and also performs more robust than its original model, so experiments with those various techniques will be conducted in future work.

Algorithm 1 DUEL Framework

Model : feature extractor f, memory $\mathcal{M}$, memory control policy π_{mc}
Input : biased dataset $\mathcal{D}$, batch size B, learning rate η, intial memory $\mathcal{M}_0$
Output : Trained parameter θ^*
$\theta \leftarrow \theta_0$
$\mathcal{M} \leftarrow \mathcal{M}_0$
while θ is not converged **do**
    $\{(d_b, d_b^+)\}_{b=1}^B \leftarrow \text{sample}(\mathcal{D})$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{NCE}}(\{d_b\}_{b=1}^B, \{d_b^+\}_{b=1}^B, \{d_b^+\}_{b=1}^B \cup \mathcal{M}; f, \theta)$
    $\theta \leftarrow \theta - \text{Optimizer}(\nabla_\theta \mathcal{L}, \eta)$
    for $b \in \{1, \cdots, B\}$ **do**
        $\mathcal{M} \leftarrow \mathcal{M} \cup \{d_b\}$
        $J \leftarrow \arg\max_J \pi_{mc}(\mathcal{M}, J; f, \theta)$
        $\mathcal{M} \leftarrow \mathcal{M} / \{d_J\}$
    end for
    $\pi_{mc} \leftarrow \text{UpdatePolicy}(\pi_{mc})$
end while
$\theta^* \leftarrow \theta$

Algorithm 2 DUEL Framework with naïve MCP

Model : feature extractor f, memory $\mathcal{M}$
Input : biased dataset $\mathcal{D}$, batch size B, learning rate η, intial memory $\mathcal{M}_0$
Output : Trained parameter θ^*
$\theta \leftarrow \theta_0$
$\mathcal{M} \leftarrow \mathcal{M}_0$
while θ is not converged **do**
    $\{(d_b, d_b^+)\}_{b=1}^B \leftarrow \text{sample}(\mathcal{D})$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{NCE}}(\{d_b\}_{b=1}^B, \{d_b^+\}_{b=1}^B, \{d_b^+\}_{b=1}^B \cup \mathcal{M}; f, \theta)$
    $\theta \leftarrow \theta - \text{Optimizer}(\nabla_\theta \mathcal{L}, \eta)$
    for $b \in \{1, \cdots, B\}$ **do**
        $J \leftarrow \arg\max_J N(J|\mathcal{M}; f, \theta)$ in Equation 5 ▷ naïve MCP π_{mc}^*
        $\mathcal{M} \leftarrow (\mathcal{M} \cup \{d_b\}) / \{d_J\}$
    end for
end while
$\theta^* \leftarrow \theta$

A.2 Proofs

Observation 1 (Non-convergence to simplex-ETF with biased dataset)**.** *Let the representation of data form simplex-ETF. Then representations will not converge to the simplex-ETF when the data is biased with specific class* $c_{\max}$.

Proof. In Khosla et al. [11], the authors formulated the gradient of the NCE Loss. We ignored the temperature parameter τ for the ease of formulation.

$$\frac{\partial \mathcal{L}_i}{\partial z_i} = \sum_{p \in P(i)} z_p(P_{ip} - X_{ip}) + \sum_{n \in N(i)} z_n P_{in} \tag{6}$$

Let $c_i = c_{max}$. In this case, P_{ip}, X_{ip} and P_{in} can be computed as below. Note that the P_{ip} and P_{in} is the same with p and n each.

$$P^+ = P_{ip} = \frac{exp(1)}{k\{\rho_{\max} exp(1) + (1-\rho_{\max}) exp(-1/|\mathcal{C}|)\}}, X_{ip} = \frac{1}{k\rho_{\max}} \tag{7}$$

$$P^- = P_{in} = \frac{exp(-1/|\mathcal{C}|)}{k\{\rho_{\max} exp(1) + (1-\rho_{\max}) exp(-1/|\mathcal{C}|)\}} \tag{8}$$

Then the gradient of z_i can be derived as

$$\frac{\partial \mathcal{L}_i}{\partial z_i} = k\rho_{\max} z_i \left(P^+ - \frac{1}{k\rho_{\max}} \right) + P^- \cdot \sum_{n \in N(i)} z_n \tag{9}$$

By the property of the ETF, the sum of all different vectors is 0.

$$= z_i \cdot k\rho_{\max} \left(P^+ - \frac{1}{k\rho_{\max}} \right) - z_i \cdot k\rho_{\min} P^- \tag{10}$$

$$= z_i(k\rho_{\max} P^+ - k\rho_{\min} P^- - 1) = z_i \cdot kP^-(-1 + \rho_{\max} - \rho_{\min}) \tag{11}$$

So the gradient of the loss function is parallel to the feature vector when $c_i = c_{\max}$. However, if $c_i \neq c_{\max}$, the gradient will be computed differently.

$$P^+ = P_{ip} = \frac{exp(1)}{k\{\rho_{\min} exp(1) + (1-\rho_{\min}) exp(-1/|\mathcal{C}|)\}}, X_{ip} = \frac{1}{k\rho_{\min}} \tag{12}$$

$$P^- = P_{in} = \frac{exp(-1/|\mathcal{C}|)}{k\{\rho_{\min} exp(1) + (1-\rho_{\min}) exp(-1/|\mathcal{C}|)\}} \tag{13}$$

$$\frac{\partial \mathcal{L}_i}{\partial z_i} = k\rho_{\min} z_i \left(P^+ - \frac{1}{k\rho_{\min}} \right) + P^- \cdot \sum_{n \in N(i)} z_n \tag{14}$$

$$= z_i(k\rho_{\min} P^+ - 1 - k\rho_{\min} P^-) + z_{\max} \cdot k(\rho_{\max} - \rho_{\min}) P^- \tag{15}$$

$$= z_i \cdot (-kP^-) + z_{\max} \cdot kP^-(\rho_{\max} - \rho_{\min}) \tag{16}$$

□

Theorem 1 (Safety of naïve MCP)**.** *Let the latent space form a simplex ETF. Suppose that there is a new data* d_{new} *and* π^*_{mc} *should replace a sample with* d_{new}*. Let sample chosen by* π^*_{mc} *among the pool* D_k *be* d_J*. Let* p_d *with the replaced pool be* $p_{d|\pi}$*. After then replacement,* $p_d \leq p_{d|\pi}$ *is satisfied for any pool* D_k*.*

Proof.

$$p_d = \sum_i \sum_j Pr(\hat{c}_i \neq \hat{c}_j | d_i, d_j; f) \tag{17}$$

$$= \sum_i \sum_j \{1 - Pr(\hat{c}_i = \hat{c}_j | d_i, d_j; f)\} \tag{18}$$

To show $p_d \leq p_{d|\pi}$, we show that p_d increases when d_{new} replaces d_J. During replacement, $(k-1)$ samples will not be changed, so we defined Δp_d and $\Delta p_{d|\pi}$ to ignore the remaining ones' relationship. In the end, proving $p_d \leq p_{d|\pi}$ is the same as proving $\Delta p_d \leq \Delta p_{d|\pi}$.

i) $\hat{c}_{\max} = \hat{c}_{new}$

Let $c_{\max}$ be the implicit class with the maximum number of samples that share the representation. By the definition, $\arg\max_J N(J|D_k; f) = \sum_i Pr(\hat{c}_i = \hat{c}_J|d_i, d_j; f)$ selects an element with the class $\hat{c}_{\max}$. $\hat{c}_{\max} = \hat{c}_{new}$ means that two data d_{new} and d_J are *latent indistinguishable*, so $\Delta p_d = \Delta p_{d|\pi}$.

ii) $\hat{c}_{\max} \neq \hat{c}_{new}$

In this case, $n_{\hat{c}_J} = n_{\hat{c}_{\max}} \geq n_{\hat{c}_{new}}$ is satisfied.

$$\Delta p_{d|\pi} = \sum_{i \neq J} \{1 - Pr(\hat{c}_i = \hat{c}_{new}|\cdot)\} + 1 - Pr(\hat{c}_{new} = \hat{c}_{new}|\cdot) \tag{19}$$

$$= (n_{\hat{c}_{new}} - 1)\{1 - h(-1/|\mathcal{C}|)\} + (k - n_{\hat{c}_{new}})\{1 - h(1)\} + 1 - Pr(\hat{c}_J = \hat{c}_J|\cdot) \tag{20}$$

$$\geq (n_{\hat{c}_J} - 1)\{1 - h(-1/|\mathcal{C}|)\} + (k - n_{\hat{c}_J})\{1 - h(1)\} + 1 - Pr(\hat{c}_J = \hat{c}_J|\cdot) \tag{21}$$

$$= \sum_{i \neq J} \{1 - Pr(\hat{c}_i = \hat{c}_J|\cdot)\} + 1 - Pr(\hat{c}_J = \hat{c}_J|\cdot) = \Delta p_d \tag{22}$$

□

Theorem 2 (Nearly latent indistinguishable)**.** *Let two samples be nearly latent indistinguishable,* $Pr(\hat{c}_i = \hat{c}_j|d_i, d_j; f) \geq \alpha \geq 0$*. Then there exists the unique* α^* *that satisfies* $Pr(\hat{c}_i = \hat{c}_j|d_i, d_j; f) \geq \alpha \leftrightarrow f(x_i)^T f(x_j) \geq \alpha^*$.

Proof. By the definition of f, there exists $\theta \in [0, \pi]$ that satisfies $\cos\theta = f(x_i)^T f(x_j)$.

$$h(f(x_i)^T f(x_j)) = h(\cos\theta) \tag{23}$$

Note that cosine function is continuous and monotonic decreasing in the inverval $[0, \pi]$. Then,

$$h(-1) = h(\cos(\pi)) = 0, h(1) = h(\cos(0)) = 1. \tag{24}$$

By the property of the continuous function, $(h \circ \cos)$ is continuous and monotonic decreasing function. Let $Pr(\hat{c}_i = \hat{c}_j|\cdot) = \alpha$ and $0 \leq \alpha \leq 1$. Then we can get a unique θ^* that satisfies $h(\cos(\theta^*)) = \alpha$ by using intermediate value theorem. If we set $\alpha^* = \cos(\theta^*)$, the proposition is proved. □

A.3 Details of Experiments

Training setting. To validate that the proposed frameworks are robust with biased dataset, we design several experiments. We use only the CNN layers of the ResNet-50 [9] as the backbone. We modify the first CNN kernel like Chen et al. [4]. We add just one linear layer for the projection layer. The dimension of the projected vectors is 256. We use Adam optimizer [12] and Cosine scheduler [7] for decaying learning rate. Initial learning rate is set to 0.05. We trained for 40k steps with the same batch size 256 for each experiment. The temperature parameter τ is set to 0.7 for all experiments. We use fixed, weak augmentations which contain only color jittering, grayscale and horizontal filp. As images are small, we do not apply cropping or resizing to the images.

Implementation details. To implement our proposed models, we need to connect the memory structure to previous frameworks while not harming their properties. Designing D-MoCo is relatively easier than D-SimCLR because MoCoV2 already has the memory in it. We simply add the policy to the model directly. We use memory for D-MoCo which can hold 2048 representations of images. However, in D-SimCLR, the memory needs to hold raw images instead of their representations because the model does not use the momentum encoder. Also, for the memory efficiency, we use the stop gradient to the representations from the memory. D-SimCLR contains memory with 512 images. Figure 5 shows the shape of score functions we use for the experiments. We use a gaussian kernel $h(x) = \exp(-(x-1)^2/\tau)$ with temperature parameter $\tau = 1$ and normalize it to fit $h(x) \in [0, 1]$.

Evaluation metrics. We evaluate trained models with several metrics. Firstly, we use linear probing as the downstream task. We train the linear layer with SGD optimizer for 100 epochs each. The momentum for SGD is 0.9 and weight decay parameter is 10^{-6}. Initial learning rate for this task is 0.01. We also measure the Top-k accuracies with different k-s (1,10).

Resource usage. Table 4 shows the usage of time and VRAM for both the model and the memory $\mathcal{M}$. Note that D-MoCo only needs a small amount of memory and some affordable time, contrary to its robustness against biases. D-SimCLR needs additional time because it stores images instead of their representations and it should extract representations again with newly updated encoder.

Table 4: Resource usage.

Methods	Time (h)	Memory (MB)
MoCoV2	2.7	10,415
SimCLR	4.1	15,573
SimSiam	4.0	15,541
D-MoCo	3.4	10,415
D-SimCLR	7.7	19,065

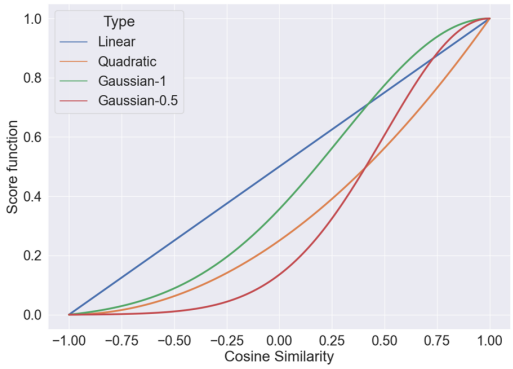

Figure 5: Visualization of score functions.