Self-Contrastive Learning: Single-Viewed Supervised Contrastive Framework Using Sub-network

Sangmin Bae[1]*, Sungnyun Kim[1]*, Jongwoo Ko[1], Gihun Lee[1], Seungjong Noh[2], Se-Young Yun[1]

[1]Graduate School of Artificial Intelligence, KAIST,
[2]SK Hynix
{bsmn0223, ksn4397, jongwoo.ko, opcrisis, yunseyoung}@kaist.ac.kr,
seungjong.noh@sk.com

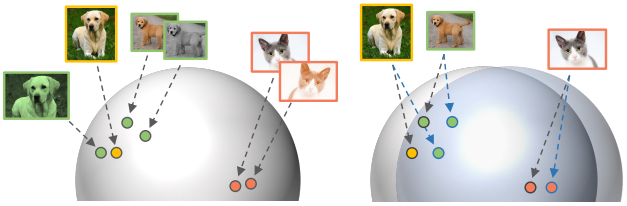
Abstract

Contrastive loss has significantly improved performance in supervised classification tasks by using a multi-viewed framework that leverages augmentation and label information. The augmentation enables contrast with another view of a single image but enlarges training time and memory usage. To exploit the strength of multi-views while avoiding the high computation cost, we introduce a multi-exit architecture that outputs multiple features of a single image in a single-viewed framework. To this end, we propose Self-Contrastive (SelfCon) learning, which self-contrasts within multiple outputs from the different levels of a single network. The multi-exit architecture efficiently replaces multi-augmented images and leverages various information from different layers of a network. We demonstrate that SelfCon learning improves the classification performance of the encoder network, and empirically analyze its advantages in terms of the single-view and the sub-network. Furthermore, we provide theoretical evidence of the performance increase based on the mutual information bound. For ImageNet classification on ResNet-50, SelfCon improves accuracy by +0.6% with 59% memory and 48% time of Supervised Contrastive learning, and a simple ensemble of multi-exit outputs boosts performance up to +1.5%. Our code is available at https://github.com/raymin0223/self-contrastive-learning.

1 Introduction

While the cross-entropy (CE) loss is the most common and powerful loss function for supervised classification tasks, lots of alternatives have been proposed to overcome the shortcomings of cross-entropy, such as high generalization error (Liu et al. 2016; Elsayed et al. 2018). Among the various approaches, Supervised Contrastive (SupCon (Khosla et al. 2020)) loss recently showed remarkable improvement in performance for large-scale benchmarks like ImageNet (Deng et al. 2009). The main idea of this loss is to make representations from the same class closer together and representations from different classes farther apart (see Figure 1a).

SupCon and its related works (Graf et al. 2021; Zheng et al. 2021; Chen et al. 2022; Li et al. 2022) have been developed on the top of a *multi-viewed* framework that leverages two core factors, augmentation and label information, when formulating the contrastive task. Additional augmented images improve the performance by enabling contrast within a single image. The *multi-viewed* framework is crucial. We indeed empirically observed that a simple extension to a *single-viewed* framework (i.e., only exploiting the label information) significantly degrades the performance on large-scale datasets (Section 5.1). However, the augmentation-based multi-view approach makes the training time and memory usage highly expensive (Chen et al. 2020; He et al. 2020; Caron et al. 2020).



(a) SupCon (b) SelfCon

Figure 1: Overview of (a) SupCon learning and (b) SelfCon learning. The anchor, positive (which is desired to be close to the anchor), and negative (which is desired to be far from the anchor) samples are represented on the feature space as yellow, green, and red points, respectively. While SupCon relies on the augmentation-based multi-views, SelfCon is a single-viewed supervised contrastive learning framework. SelfCon produces multiple features from a single instance, using the sub-network.

To implement the multi-view framework without data augmentations, we propose **Self-Contrastive (SelfCon) learning**, which uses the multi-exit architecture (Teerapittayanon, McDanel, and Kung 2016; Zhang et al. 2019a,b; Phuong and Lampert 2019) having sub-networks that produce multiple features of a single image. With the multi-exit architecture, SelfCon *self-contrasts* within multiple outputs from the different levels of a single network (see Figure 1b), making the *single-viewed* framework usable. Therefore, the multi-exit architecture efficiently replaces data augmentation by leveraging various information from different layers of a network (Zeiler and Fergus 2014).

We summarize the contributions of our paper as follows:

*These authors contributed equally.

[Section 3] We propose Self-Contrastive learning, which is the first study on a single-viewed contrastive framework exploiting multiple features from different levels of a single network.

[Section 4] We guarantee that SelfCon loss is the lower bound of label-conditional mutual information (MI) between the intermediate and the last features. To our knowledge, this is the first work to provide the MI bound for supervised contrastive learning.

[Section 5.1] SelfCon learning efficiently achieves higher classification accuracy for various benchmarks compared to CE and SupCon loss. Furthermore, SelfCon with an ensemble prediction boosts performance by a large margin.

[Section 5.2–5.4] We extensively investigate the benefits of SelfCon learning in terms of the single-viewed batch and the sub-network. Also, our empirical study of MI estimation provides evidence for the superior performance.

2 Related Works

2.1 Contrastive Learning in Supervision

After Oord, Li, and Vinyals (2018) proposed InfoNCE loss (also called a contrastive loss), contrastive learning-based algorithms began to show a remarkable improvement in image representation learning (Chen et al. 2020; He et al. 2020; Grill et al. 2020; Caron et al. 2020; Chen and He 2020). Khosla et al. (2020) extended the contrastive learning to a supervised classification task to resolve the generalization issue of cross-entropy loss. The idea of SupCon (Khosla et al. 2020), which leverages augmentation and label information on the contrastive framework, has also been extended to semantic segmentation (Wang et al. 2021) and language tasks (Gunel et al. 2020). While SupCon loss utilizes the output features from two random augmentations, our approach contrasts the features from different network paths by introducing the multi-exit framework (Teerapittayanon, McDanel, and Kung 2016; Zhang et al. 2019a). In this paper, we investigate the advantages of model-based contrastive learning within the single-viewed framework. Moreover, we offer the first proof of the MI bound for the supervised contrastive framework to theoretically explain how SelfCon improves the classification performance.

2.2 Multi-Exit Architectures

As earlier layers of the deep neural network suffer from the vanishing gradient issue (Szegedy et al. 2015; He et al. 2016), previous works have introduced a multi-exit architecture (Lee et al. 2015; Teerapittayanon, McDanel, and Kung 2016; Bolukbasi et al. 2017) that attaches sub-networks on the intermediate layers. The sub-networks have also been used to predict at any point of the network during the evaluation phase (i.e., anytime inference (Huang et al. 2017; Yang et al. 2020; Ruiz and Verbeek 2021)), as well as to leverage the information from different levels of a network which leads to the performance gain (Zeiler and Fergus 2014; Zhang et al. 2019a; Yao and Sun 2020). Recently, the knowledge distillation-based losses (Lan, Zhu, and Gong 2018; Zhang et al. 2019a,b; Phuong and Lampert 2019; Zhang et al. 2021) have been proposed to effectively train the sub-network. Motivated by these methods, we propose a novel supervised contrastive learning that self-contrasts within the multi-exit outputs. The sub-network mitigates the vanishing gradient issue and reduces the generalization error, as well as replacing the augmentation-based multi-views.

3 Self-Contrastive Learning

We propose a new supervised contrastive loss that maximizes the similarity of the outputs from different network paths by introducing the multi-exit framework. We define an encoder structure, using $\boldsymbol{F}$ as a backbone network and $\boldsymbol{G}$ as a sub-network, that shares the backbone's parameters up to some intermediate layer. $\boldsymbol{T}$ denotes the sharing layers that produce the intermediate feature. Note that $\boldsymbol{F}$ and $\boldsymbol{G}$ include the projection head after the encoder. We highlight the **positive** and **negative** pairs with respect to an **anchor** sample, following Figure 2.

SupCon loss To mitigate the weaknesses of cross-entropy, such as the reduced generalization performance and the possibility of poor margins, Khosla et al. (2020) propose a supervised version of contrastive loss that defines the positive pairs as every sample with the same ground-truth label. We reformulate the SupCon loss function as follows:

$$\mathcal{L}_{sup} = \sum_{i\in I}\Big[-\tfrac{1}{|P_i|}\sum_{p\in P_i}\boldsymbol{F}(\boldsymbol{x}_i)^\top \boldsymbol{F}(\boldsymbol{x}_p) + \log\Big(\sum_{p\in P_i} e^{\boldsymbol{F}(\boldsymbol{x}_i)^\top \boldsymbol{F}(\boldsymbol{x}_p)} + \sum_{n\in N_i} e^{\boldsymbol{F}(\boldsymbol{x}_i)^\top \boldsymbol{F}(\boldsymbol{x}_n)}\Big)\Big] \tag{1}$$

$$P_i \equiv \{p \in I \setminus \{i\} | y_p = y_i\},\ N_i \equiv \{n \in I | y_n \neq y_i\}$$

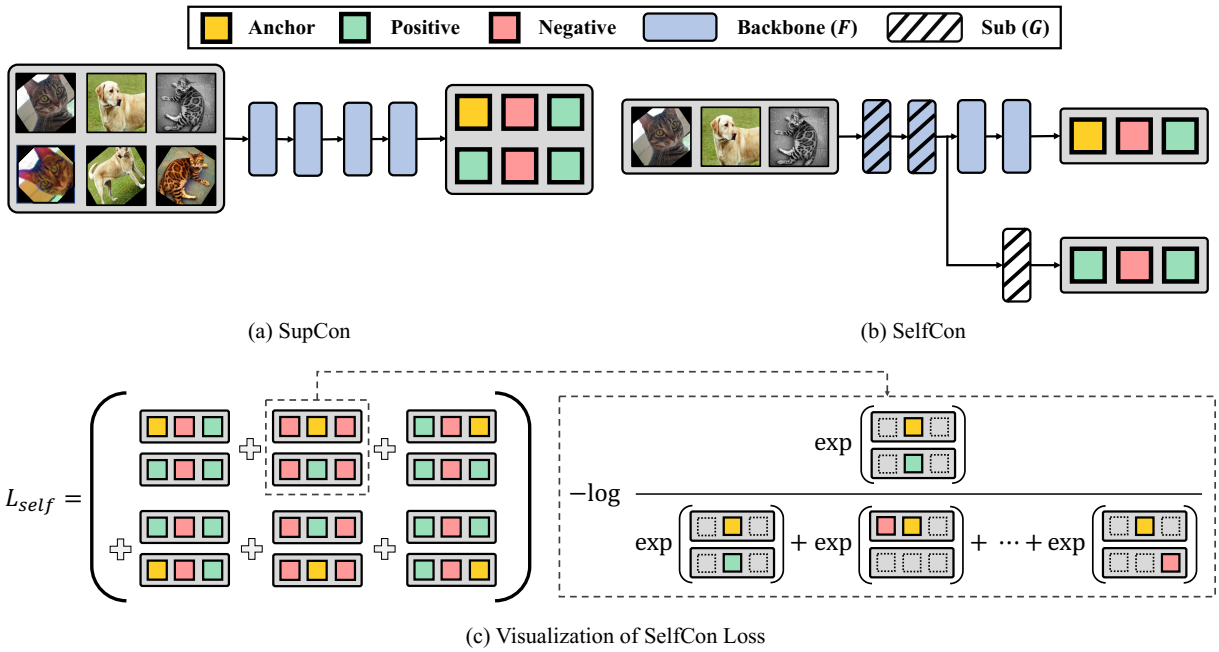
where $I \equiv \{1, \dots, 2B\}$, and B is the batch size. For brevity, we omit the temperature τ, which softens or hardens the softmax value, and the dividing constant for the summation of anchor samples (i.e., $|I|^{-1}$). I denotes a set of indices for the multi-viewed batch that concatenates the original B images and the augmented ones, i.e., $\boldsymbol{x}_{B+i}$ is an augmented pair of $\boldsymbol{x}_i$. P_i and N_i are sets of positive and negative pair indices with respect to an anchor i. Eq. 1 is a type of categorical cross-entropy loss; the numerator contains the positive pair, and the denominator contains both positive and negative pairs.

SelfCon loss We aim to maximize the similarity between the outputs from the backbone and the sub-network. To this end, we define **SelfCon loss, which forms a self-contrastive task for every output, including the features from the sub-network.**

$$\mathcal{L}_{self} = \sum_{\substack{i\in I,\\ \boldsymbol{\omega}\in\boldsymbol{\Omega}}}\Big[-\tfrac{1}{|P_{i1}||\boldsymbol{\Omega}|}\sum_{\substack{p_1\in P_{i1},\\ \boldsymbol{\omega}_1\in\boldsymbol{\Omega}}}\boldsymbol{\omega}(\boldsymbol{x}_i)^\top \boldsymbol{\omega}_1(\boldsymbol{x}_{p_1}) + \log\sum_{\boldsymbol{\omega}_2\in\boldsymbol{\Omega}}\Big(\sum_{p_2\in P_{i2}} e^{\boldsymbol{\omega}(\boldsymbol{x}_i)^\top \boldsymbol{\omega}_2(\boldsymbol{x}_{p_2})} + \sum_{n\in N_i} e^{\boldsymbol{\omega}(\boldsymbol{x}_i)^\top \boldsymbol{\omega}_2(\boldsymbol{x}_n)}\Big)\Big] \tag{2}$$

$$P_{ij} \equiv \{p_j \in I \setminus \{i\} | y_{p_j} = y_i\},\ N_i \equiv \{n \in I | y_n \neq y_i\}$$

where $I \equiv \{1, \dots, B\}$, and $\boldsymbol{\Omega} = \{\boldsymbol{F}, \boldsymbol{G}\}$ is a function set of the backbone network and the sub-network. We also omit τ



(c) Visualization of SelfCon Loss

Figure 2: (Top) Comparison of learning frameworks in terms of augmentation and architecture. In both SupCon (Khosla et al. 2020) and SelfCon, every sample of the same ground-truth label with an anchor is used as a positive pair. Specifically, in SelfCon, an anchor from the backbone network contrasts other features from the backbone, as well as the features from the sub-network. (Bottom) We visualized the SelfCon loss function to ease the understanding in Section 3. $\exp(\cdot)$ denote the exponential function of the cosine similarity between two features. Note that SupCon loss has the same form but uses the representations from the multi-viewed batch. Best seen in color.

and the dividing constant (i.e., $(|I||\boldsymbol{\Omega}|)^{-1}$). $\boldsymbol{\omega}_1$ is a function that generates positive pair, and $\boldsymbol{\omega}_2$ is for generating every contrastive pair from a multi-exit network. We include an anchor sample to the positive set when the output feature is from a different exit path, i.e., $P_{ij} \leftarrow P_{ij} \cup \{i\}$ when $\boldsymbol{\omega} \neq \boldsymbol{\omega}_j$. For example, $\boldsymbol{G}(\boldsymbol{x}_i)$ is also a positive pair for $\boldsymbol{F}(\boldsymbol{x}_i)$. Refer to Figure 2 for better understanding of contrastive task formation in the SelfCon framework.

Whereas prevalent contrastive approaches (Khosla et al. 2020; Chen et al. 2020; Grill et al. 2020) force a multi-viewed batch generated by data augmentation, the sub-network in SelfCon learning plays a role as the augmentation and provides an alternative view on the feature space. Therefore, without the additional augmented samples, we formulate our SelfCon loss function with a single-viewed batch.

We can further use multiple sub-networks, i.e., $\boldsymbol{\Omega} = \{\boldsymbol{F}, \boldsymbol{G}_1, \boldsymbol{G}_2, \dots\}$. Appendix B.4 presents the classification performance of the expanded network, but there was no significant improvement from that of a single sub-network. Thus, we have efficiently used a single sub-network throughout our paper.

4 Discussions

In this Section, we discuss theoretical evidence for the success of SelfCon learning. We summarize the discussion as follows: Selfcon learning improves the classification performance by encouraging the intermediate feature to have more label information in the last feature.

Discussion 4.1. How does SelfCon loss encourage the intermediate feature to learn the label information in the last feature? Generally, prior works (Oord, Li, and Vinyals 2018; Hjelm et al. 2018) support the success of unsupervised contrastive learning from the connection to the MI. In this sense, in Proposition 4.1, we first prove the connection between a *supervised* contrastive loss and the MI of positive pair features. In Proposition 4.2, we then provide the MI bound within a single-viewed batch using the sub-network feature.

Proposition 4.1. *Let $\boldsymbol{x}$ and $\boldsymbol{z}$ be different samples that share the same class label c. Then, with some discriminator function modeled by a neural network $\boldsymbol{F}$ and $2(K-1)$ negative sample size, SupCon loss maximizes the lower bound of conditional MI between the output features of a positive pair.*

$$\log(2K-1) - \mathcal{L}_{sup}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{F}, K) \leq \mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{F}(\boldsymbol{z})|c) \quad (3)$$

Proposition 4.2. *SelfCon loss maximizes the lower bound of MI between the output features from the backbone and the sub-network.*

$$\log(2K-1) - \mathcal{L}_{self}(\boldsymbol{x}; \{\boldsymbol{F}, \boldsymbol{G}\}, K) \leq \mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{G}(\boldsymbol{x})|c) \quad (4)$$

SupCon and SelfCon loss have a negative sample size of $2(K-1)$ because of the augmented negative pairs for SupCon and the sub-network features for SelfCon.

We extend the above MI bound to the MI between the intermediate and last feature of a backbone. Although MI is

ill-defined between the variables with deterministic mapping, previous works view the training of a neural network as a stochastic process (Shwartz-Ziv and Tishby 2017; Goldfeld et al. 2019; Saxe et al. 2019). Thus, encoder features are considered as random variables, which allows us to define and analyze the MI between the features.

Proposition 4.3. *As $\boldsymbol{F}(\boldsymbol{x})$ and $\boldsymbol{G}(\boldsymbol{x})$ are conditionally independent given the intermediate representation $\boldsymbol{T}(\boldsymbol{x})$, they formulate a Markov chain: $\boldsymbol{G} \leftrightarrow \boldsymbol{T} \leftrightarrow \boldsymbol{F}$ (Cover 1999). Then, the following is satisfied.*

$$\mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{G}(\boldsymbol{x})|c) \leq \mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{T}(\boldsymbol{x})|c) \tag{5}$$

Proposition 4.3 states that minimizing SelfCon loss, which maximizes the lower bound of MI between the features from the backbone and the sub-network, can encourage the intermediate features to learn the class-related information from the last features. Although there is indeed a gap in Eq. 5, the gap between $\mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{G}(\boldsymbol{x})|c)$ and $\mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{T}(\boldsymbol{x})|c)$ may not be large since we implement $\boldsymbol{G}(\boldsymbol{x})$ as a simple linear transformation of $\boldsymbol{T}(\boldsymbol{x})$ in practice. We empirically demonstrated the actual increment of $\mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{T}(\boldsymbol{x})|c)$ in Section 5.4.

Discussion 4.2. How does increasing $\mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{T}(\boldsymbol{x})|c)$ improve classification performance? To understand the information that SelfCon loss maximizes, we decompose the r.h.s. of Eq. 5 as follows:

$$\begin{aligned}\mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{T}(\boldsymbol{x})|c) &= \mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{T}(\boldsymbol{x}), c) - \mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); c) \\ &= \underbrace{\mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{T}(\boldsymbol{x}))}_{(\square)} + \underbrace{\mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); c|\boldsymbol{T}(\boldsymbol{x})) - \mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); c)}_{(\blacksquare)}.\end{aligned} \tag{6}$$

($\square$) implies that $\boldsymbol{T}(\boldsymbol{x})$ is distilled with refined information (not conditional with respect to c) from $\boldsymbol{F}(\boldsymbol{x})$, so the encoder can produce better representation (Hjelm et al. 2018; Bachman, Hjelm, and Buchwalter 2019). On the other hand, ($\blacksquare$) is interaction information (Yeung 1991) that measures the influence of $\boldsymbol{T}(\boldsymbol{x})$ on the amount of shared information between $\boldsymbol{F}(\boldsymbol{x})$ and c. Increasing this interaction information means the intermediate feature enhances the correlation between the last feature and the label. Therefore, when we jointly optimize ($\square + \blacksquare$), the intermediate and last features have aligned label information.

In this sense, SelfCon loss is based on the *InfoMax* principle (Linsker 1989), which is about learning to maximize the MI between the input and output of a neural network. It has been proved that *InfoMax*-based loss regularizes intermediate features and improves performance in semi-supervised (Rasmus et al. 2015) and knowledge transfer (Ahn et al. 2019) domains. Similar to the previous works, SelfCon loss increases the classification accuracy by regularizing the intermediate feature to have class-related information aligned with the last feature.

Discussion 4.3. Is SelfCon loss applicable to unsupervised representation learning? The unsupervised version of SelfCon loss is a lower bound of ($\square$) in Eq. 6. By maximizing only ($\square$), the last feature may follow the intermediate feature, learning redundant information about the input.[1] This could be the reason why SelfCon learning does *not* work in an unsupervised environment (refer to Appendix C.1). However, to mitigate this problem, we propose in Appendix C.2 a loss function to prevent the backbone from following the sub-network. For this aim, we remove the term in Eq. 2 where $\boldsymbol{\omega} = \boldsymbol{F}$ (i.e., anchor from backbone) and $\boldsymbol{\omega}_j = \boldsymbol{G}$ (i.e., contrastive pair from sub-network). This modification improves upon NT-Xent loss (Chen et al. 2020) in the unsupervised CIFAR-100 experiment.

5 Experiment

We present the image classification accuracy for standard benchmarks, such as CIFAR-10, CIFAR-100 (Krizhevsky 2009), Tiny-ImageNet (Le and Yang 2015), ImageNet-100 (Tian, Krishnan, and Isola 2019), and ImageNet (Deng et al. 2009), and extensively analyze the results. We report the mean and standard deviation of top-1 accuracy over three random seeds. We used the optimal structure and position of the sub-network, however, the overall performance was comparable to or better than the baselines. The complete implementation details and hyperparameter tuning results are presented in Appendix B.

We also have implemented SupCon with a single-viewed batch (SupCon-S) and SelfCon with a multi-viewed batch (SelfCon-M) in order to examine the independent effects of the single-view and the sub-network. Note that their loss functions only require the change of the anchor set I and corresponding positive and negative sets (i.e., P_{ij} and N_i) in Eq. 1 and 2.

5.1 Representation Learning

We measured the classification accuracy of the representation learning protocol (Chen et al. 2020), which consists of *2-stage training*: (1) pretraining an encoder network and (2) fine-tuning a linear classifier with the frozen encoder (called a linear evaluation). In Appendix D, we compared with other supervised losses in the 1-stage training framework (i.e., not decoupling the encoder pretraining and fine-tuning).

Small-scale benchmark The classification accuracy is summarized in Table 1. Interestingly, the loss functions in the single-viewed batch outperform their multi-viewed counterparts in all settings. Furthermore, our SelfCon learning, which trains using the sub-network, shows higher classification accuracy than CE and SupCon. The effects of the sub-network are analyzed in Section 5.3.

Large-scale benchmark We summarized the experimental results for the ImageNet-100, of which 100 classes were randomly sampled (Tian, Krishnan, and Isola 2019), and the full-scale ImageNet (Table 2). Our SelfCon learning that includes the sub-network consistently outperforms SupCon learning on both ImageNet-100 and ImageNet. In particular, SelfCon showed a higher efficiency ratio (i.e., cost-to-accuracy) than

[1] In supervision, a suboptimal case where $\boldsymbol{T}(\boldsymbol{x})$ becomes a sink for $\boldsymbol{F}(\boldsymbol{x})$ does not happen because the deeper layers have a larger capacity for label information (Shwartz-Ziv and Tishby 2017).

Method	Single-View	Sub-Net.	ResNet-18			ResNet-50		
			CIFAR-10	CIFAR-100	Tiny-ImageNet	CIFAR-10	CIFAR-100	Tiny-ImageNet
CE	✓		$94.7_{\pm 0.1}$	$72.9_{\pm 0.1}$	$57.5_{\pm 0.3}$	$94.9_{\pm 0.2}$	$74.8_{\pm 0.1}$	$62.3_{\pm 0.4}$
SupCon			$94.7_{\pm 0.2}$	$73.0_{\pm 0.0}$	$56.9_{\pm 0.4}$	$95.6^{\dagger}_{\pm 0.1}$	$75.5^{\dagger}_{\pm 0.2}$	$61.6_{\pm 0.2}$
SelfCon-M		✓	$95.0_{\pm 0.1}$	$74.9_{\pm 0.1}$	$59.2_{\pm 0.0}$	$95.5_{\pm 0.1}$	$76.9_{\pm 0.1}$	$63.0_{\pm 0.2}$
SupCon-S	✓		$94.9_{\pm 0.0}$	$73.9_{\pm 0.1}$	$58.4_{\pm 0.3}$	$\mathbf{95.8_{\pm 0.1}}$	$76.7_{\pm 0.1}$	$62.0_{\pm 0.2}$
SelfCon	✓	✓	$\mathbf{95.3_{\pm 0.2}}$	$\mathbf{75.4_{\pm 0.1}}$	$\mathbf{59.8_{\pm 0.4}}$	$\mathbf{95.7_{\pm 0.2}}$	$\mathbf{78.5_{\pm 0.3}}$	$\mathbf{63.7_{\pm 0.2}}$

Table 1: The results of the linear evaluation for small-scale benchmarks. Bold type is for all the values of which the standard deviation range overlaps with that of the best accuracy. We used the same batch size of 1024 and a learning rate of 0.5 as Khosla et al. (2020) did in CIFAR experiments. †: We have re-implemented SupCon and also run their official code for credibility, but the accuracy was slightly lower than their reported numbers.

Method	Single-View	Sub-Net.	ImageNet-100						ImageNet								
			ResNet-18			ResNet-50			ResNet-18			ResNet-34			ResNet-50		
			Mem.	Time	Acc.	Mem.	Time	Acc.	Mem.	Time	Acc.	Mem.	Time	Acc.	Mem.	Time	Acc.
CE	✓		-	-	83.7	-	-	86.4	-	-	69.4	-	-	72.7	-	-	$76.5^{\ddagger}$
SupCon			$\times 1.5$	$\times 2.1$	85.6	$\times 1.7$	$\times 1.9$	88.2	$\times 1.5$	$\times 2.2$	71.2	$\times 1.5$	$\times 2.1$	74.9	$\times 1.7$	$\times 2.1$	$78.0^{\ddagger}$
SelfCon-M		✓	$\times 1.6$	$\times 2.1$	85.8	$\times 1.8$	$\times 2.2$	**88.7**	$\times 1.7$	$\times 2.3$	**71.6**	$\times 1.7$	$\times 2.2$	75.5	$\times 1.8$	$\times 2.2$	78.4
SupCon-S	✓		$\times \mathbf{1.0}$	$\times \mathbf{1.0}$	84.9	$\times \mathbf{0.9}$	$\times \mathbf{0.8}$	87.8	$\times \mathbf{0.9}$	$\times \mathbf{1.0}$	70.2	$\times \mathbf{0.9}$	$\times \mathbf{1.0}$	74.4	$\times \mathbf{0.9}$	$\times \mathbf{0.9}$	77.5
SelfCon	✓	✓	$\times \mathbf{1.0}$	$\times \mathbf{1.0}$	**86.1**	$\times 1.0$	$\times 1.0$	**88.7**	$\times 1.0$	$\times \mathbf{1.0}$	71.4	$\times 1.0$	$\times \mathbf{1.0}$	**75.6**	$\times 1.0$	$\times 1.0$	**78.6**

Table 2: The classification accuracy for ImageNet-100 and ImageNet. We summarized the ratio of memory (GiB / GPU) and time (sec / step) based on those of SelfCon in each architecture. ‡: We used the results in the same setting as ours (e.g., $B = 1024$) reported by Khosla et al. (2020) (refer to Figure 4 in their original paper).

SupCon, SelfCon-M, and SupCon-S. Different from small-scale benchmarks, we observed that the training difficulty of large-scale images could degrade the performance of the single-viewed method (see SupCon-S vs. SupCon). The poor performance of SupCon-S, which consumes an amount of memory and time similar to SelfCon, reflects the superiority of SelfCon.

On large-scale benchmarks, the difference in accuracy between SelfCon and SelfCon-M was smaller than that on small-scale benchmarks. We suppose that it is mainly attributed to the over-/under-fitting problem. In fact, various factors (e.g., architecture, dataset, batch size, and training epochs) in combination can affect the bias-variance trade-off. For example, the ImageNet result on ResNet-18 appears to be affected by the underfitting from a small architecture and a huge dataset (also refer to Appendix E.2). We intensively analyzed the effects of different factors in terms of the single-view and multi-view in Section 5.2.

Ensemble prediction with sub-network The co-trained sub-network is a novel strength of SelfCon learning as an efficient and simple boosting technique. In practice, training an extra linear classifier after the frozen sub-network does not demand a high cost in the fine-tuning scheme. We can thus obtain two additional linear evaluation results by (1) fine-tuning a classifier after the sub-network output and (2) ensembling the predictions of two classifiers. Table 3 indicates that the ensemble prediction is the most powerful technique we have proposed. In particular, SelfCon can achieve a significant performance gain of +3.0% on ImageNet without requiring cost-intensive techniques such as multi-viewed batch or larger batch size (Chen et al. 2020; Khosla et al. 2020). Refer to Appendix F for the results on ResNet-18.

Method	CF-100	Tiny-IN	IN-100	IN
CE	74.8	62.3	86.4	76.5
SupCon	75.5	61.6	88.2	78.0
Backbone	78.5	63.7	88.7	78.6
Sub-network	73.3	58.9	87.6	78.5
Ensemble	**80.0**	**65.7**	**89.1**	**79.5**
└Gain (vs. CE)	+5.2	+3.4	+2.7	+3.0
└Gain (vs. SupCon)	+4.5	+4.1	+0.9	+1.5

Table 3: Classification accuracy with the classifiers after backbone, sub-network, and the ensemble of them. The ResNet-50 encoder is pretrained by the SelfCon loss function.

Downstream tasks Thus far, we have observed the SelfCon's superiority via linear evaluation performance. While our main goal is supervised classification on the target dataset, we can further use the pretrained encoder to transfer to other downstream tasks. Hence, in Table 4, we summarized the results of the downstream tasks, eight fine-grained recognition datasets and two semantic segmentation or object detection datasets, to further verify the transferability of the SelfCon's pretrained encoder. SelfCon outperforms SupCon in most of the downstream tasks, implying that ImageNet-pretrained SelfCon contains more generalized representation. Specifically, SelfCon greatly improves up to +6.8% and +4.4% for fine-grained and semantic segmentation tasks, respectively.

Method	CUB	Dogs	MIT67	Flowers	Pets	Stanford40	Cars	Aircraft
SelfCon	**62.2**	91.8	**72.3**	**85.7**	**90.6**	**77.5**	**45.2**	**39.0**
SupCon	56.8	**92.3**	65.5	82.8	89.6	76.7	40.4	37.6

(a) Fine-grained Recognition

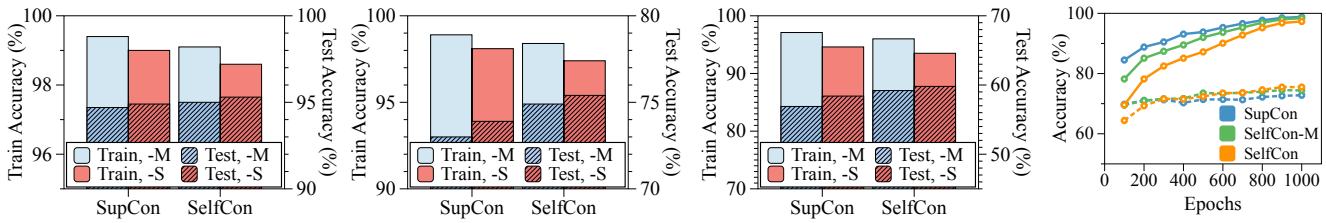
Method	VOC	COCO
SelfCon	**71.6**	**48.1**
SupCon	69.6	43.7

(b) Semantic Segmentation

Method	VOC	COCO
SelfCon	**63.0**	**29.4**
SupCon	61.8	28.8

(c) Object Detection

Table 4: Downstream task results of SelfCon and SupCon encoders. The ResNet-34 model pretrained on ImageNet is transferred. The evaluation metric is (a) linear evaluation accuracy, (b) mIoU, and (c) mAP. For the semantic segmentation, we used a DeepLabV3+ module (Chen et al. 2018), and for the object detection, we used a RetinaNet detector (Lin et al. 2017). The dataset details are in Appendix B.



(a) CIFAR-10 (b) CIFAR-100 (c) Tiny-ImageNet (d) Training Process

Figure 3: (a–c) The train and test accuracy on ResNet-18 for different views and loss functions. The accuracy is measured with a linear classifier during the linear evaluation. (d) CIFAR-100 accuracy on ResNet-18 at different epochs. The solid and dashed lines are for train and test accuracy, respectively.

5.2 Single-View vs. Multi-View

Single-view reduces generalization error. In Figures 3a-3c, SupCon shows higher train accuracy, but lower test accuracy than SupCon-S, and the same trend is observed with SelfCon-M and SelfCon (blue vs. red). Compared with single-view, multi-view from the augmented image makes the encoder amplify the memorization of data and results in overfitting to each instance. In addition, Figure 3d shows that SelfCon gradually enhances generalization ability, while SelfCon-M and SupCon achieve a little gain in test accuracy despite the fast convergence.

Multi-view is advantageous for small batch size. In supervised learning, a large batch size has been known to reduce generalization ability, which degrades performance (You, Gitman, and Ginsburg 2017; Luo et al. 2018; Wu et al. 2020). We examined whether the performance in a supervised contrastive framework is also dependent on the batch size. In Table 5, SelfCon showed the best performance in every case except for the batch size of 64. However, the multi-viewed method outperformed the single-viewed counterpart in 64-batch experiments, where underfitting may occur because of large randomness from the small batch size or the small number of positive pairs. In the ImageNet experiment on ResNet-18 (see Table 2), SelfCon-M also outperformed every method, implying that it is more important to mitigate underfitting for large-scale dataset. Conversely, in ResNet-34 and ResNet-50, SelfCon showed the best performance. In summary, multi-viewed methods may have good performance in the underfitting scenario (e.g., small batch size, small epochs, or large-scale benchmark).

Single-view is efficient in terms of memory usage and computational cost. To investigate the efficiency of a single-viewed batch against a conventional multi-viewed batch, we have compared the required memory and time cost in Table 2. Due to the additional augmented samples, the computational cost of the multi-viewed approaches is around twice as much as their single-viewed counterparts. SelfCon, on the other hand, outperformed every method with a low cost under a range of experimental conditions, while SupCon-S showed poor performance in the large-scale benchmarks. In Appendix G, we summarized the detailed numbers of the costs for SelfCon and SupCon. Although SelfCon requires the additional parameters owing to the sub-network, its memory and computation cost in practice are much more efficient.

Method	Batch Size				
	64	128	256	512	1024
CE	74.9	74.9	74.1	73.3	72.9
SupCon	74.8	73.8	72.9	72.5	73.0
SelfCon-M	**75.8**	76.5	75.9	75.0	74.9
SupCon-S	73.6	75.3	75.0	74.0	73.9
SelfCon	74.0	**76.6**	**77.0**	**75.8**	**75.4**

Table 5: The classification accuracy of CIFAR-100 on ResNet-18 with various batch sizes.

5.3 What Does the Sub-network Achieve?

Regularization effect SelfCon loss regularizes the sub-network to output similar features to the backbone network. It prevents the encoder from overfitting to the data, and it is effective in multi-viewed as well as single-viewed batches. In Figures 3a–3c, we confirm the regularization effect (i.e., lower train accuracy, but higher test accuracy) by comparing each bar of the same color. The strong regularization of the sub-network helped SelfCon (also with multi-view) outperform the SupCon counterparts. This trend can also be observed in Figure 3d and Table 5.

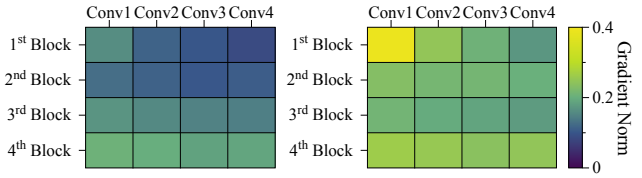


Figure 4: Gradient norm of each ResNet-18 block and convolutional layer. We computed gradients from the SupCon loss (Left) and SelfCon-M loss (Right), both from the same initialized model. All convolution layers in the block are named by order.

Figure 5: Grad-CAM (Selvaraju et al. 2017) visualizations for the feature-level multi-view generated by the sub-network. Along with the original image, each map visualizes the gradients from the sub-network (Left) and the backbone network (Right), respectively.

Mitigating the vanishing gradient SelfCon learning sends more abundant information to the earlier layers through the gradients flowing from the sub-networks. Previous works (Lee et al. 2015; Teerapittayanon, McDanel, and Kung 2016; Zhang et al. 2019a) also have pointed out that the success of the multi-exit framework owes to solving the vanishing gradient problem. In Figure 4, a large gradient flows up to the earlier layer in the SelfCon-M, whereas a large amount of the SupCon loss gradient vanishes. Note that the sub-network is positioned after the 2nd block of the ResNet-18 backbone network. Thus, there is a significant difference in the gradient norm in the 2nd block of the encoder.

Feature-level multi-view One of the advantages of SelfCon learning is that it relaxes the dependency on multi-viewed batches. This is accomplished by the multi-views on the representation space made by the parameters of the sub-network. In Figure 5, we visualize the gradient of SelfCon loss w.r.t. the intermediate layer of the backbone network (ResNet-18), right before the exit path. Both networks focus on similar but clearly different pixels of the same input image, implying that the sub-network learns another view in the feature space. As the multi-view in contrastive learning requires domain-specific augmentation, recent studies have explored domain-agnostic methods of augmentation (Lee et al. 2020; Verma et al. 2021). SelfCon could be an intriguing future work in that auxiliary networks could be an efficient substitute for data augmentation.

5.4 Mutual Information Estimation

We argue that minimizing SelfCon loss maximizes the lower bound of MI, which results in the improved classification

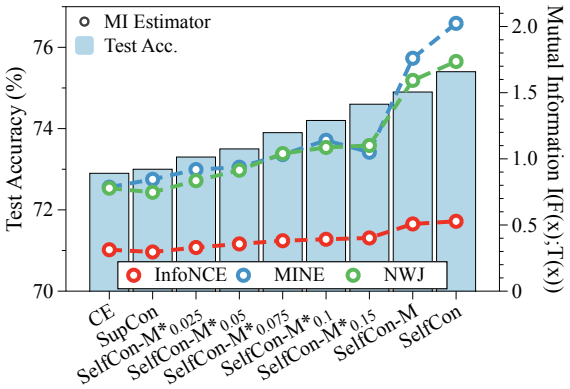


Figure 6: Test accuracy and the estimated mutual information of different methods. SelfCon-M*$_\alpha$ denotes SelfCon-M* loss with hyperparameter α. When $\alpha \geq 0.2$, the test accuracy was similar to that of SelfCon-M.

performance presented in Section 4. To empirically confirm this claim, we design an interpolation between SupCon and SelfCon-M loss as follows:

$$\mathcal{L}_{self\text{-}m*} = \frac{1}{1+\alpha}\mathcal{L}_{sup} + \frac{\alpha}{1+\alpha}\mathcal{L}_{self\text{-}m}\Big|_{\boldsymbol{\omega}=\boldsymbol{G}} \tag{7}$$

If $\alpha = 0$, $\mathcal{L}_{self\text{-}m*}$ is equivalent to the SupCon loss, and if $\alpha = 1$, then $\mathcal{L}_{self\text{-}m*}$ is *almost* the same as SelfCon-M loss. We cannot make the exact interpolation because SelfCon-M has contrastive pairs from the sub-network, whereas SupCon does not.

In Figure 6, we estimated MI with ResNet-18 and CIFAR-100 using various estimators: InfoNCE (Oord, Li, and Vinyals 2018), MINE (Belghazi et al. 2018), and NWJ (Nguyen, Wainwright, and Jordan 2010). We measured $\mathcal{I}(\boldsymbol{F}(\boldsymbol{x}); \boldsymbol{T}(\boldsymbol{x}))$ because it is difficult to estimate the conditional MI. We observed a clear increasing trend for both MI and the test accuracy as the contribution of SelfCon becomes larger (i.e., increasing α). After SelfCon loss increases the correlation between $\boldsymbol{F}(\boldsymbol{x})$ and $\boldsymbol{T}(\boldsymbol{x})$, the rich information in earlier features enables the encoder to output a better representation because the intermediate feature is also the input for the subsequent layers. Refer to Appendix H for a detailed SelfCon-M* loss formulation and the exact numbers.

6 Conclusion

We have proposed a single-viewed supervised contrastive framework called Self-Contrastive learning, which self-contrasts the multiple features from a multi-exit architecture. By replacing the augmentation with the sub-network, SelfCon enables the encoder to contrast within multiple features from a single image while significantly reducing the computational cost. We verified by extensive experiments that SelfCon loss outperforms CE and SupCon loss. We analyzed the success of SelfCon learning by exploring the effect of single-view and sub-network, such as the regularization effect, computational efficiency, or ensemble prediction. In addition, we theoretically proved that SelfCon loss regularizes the intermediate features to learn the label information in the last feature, as our MI estimation experiment has supported.

Acknowledgments

This research was supported by SK Hynix AICC K20.06_Unsupervised DL Model Error Estimation (90%). This work was also supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST), 10%).

References

Ahn, S.; Hu, S. X.; Damianou, A.; Lawrence, N. D.; and Dai, Z. 2019. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9163–9171.

Bachman, P.; Hjelm, R. D.; and Buchwalter, W. 2019. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*.

Belghazi, M. I.; Baratin, A.; Rajeswar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, R. D. 2018. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*.

Bolukbasi, T.; Wang, J.; Dekel, O.; and Saligrama, V. 2017. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, 527–536. PMLR.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 801–818.

Chen, M.; Fu, D. Y.; Narayan, A.; Zhang, M.; Song, Z.; Fatahalian, K.; and Ré, C. 2022. Perfectly Balanced: Improving Transfer and Robustness of Supervised Contrastive Learning. In *International Conference on Machine Learning*, 3090–3122. PMLR.

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

Chen, X.; and He, K. 2020. Exploring Simple Siamese Representation Learning. *arXiv preprint arXiv:2011.10566*.

Cover, T. M. 1999. *Elements of information theory*. John Wiley & Sons.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Elsayed, G.; Krishnan, D.; Mobahi, H.; Regan, K.; and Bengio, S. 2018. Large margin deep networks for classification. *Advances in neural information processing systems*, 31.

Goldfeld, Z.; van den Berg, E.; Greenewald, K. H.; Melnyk, I.; Nguyen, N.; Kingsbury, B.; and Polyanskiy, Y. 2019. Estimating Information Flow in Deep Neural Networks. In *ICML*.

Graf, F.; Hofer, C.; Niethammer, M.; and Kwitt, R. 2021. Dissecting supervised constrastive learning. In *International Conference on Machine Learning*, 3821–3830. PMLR.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Gunel, B.; Du, J.; Conneau, A.; and Stoyanov, V. 2020. Supervised Contrastive Learning for Pre-trained Language Model Fine-tuning. *arXiv preprint arXiv:2011.01403*.

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.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Huang, G.; Chen, D.; Li, T.; Wu, F.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. *Technical Report TR-2009, University of Toronto*.

Lan, X.; Zhu, X.; and Gong, S. 2018. Knowledge distillation by on-the-fly native ensemble. *arXiv preprint arXiv:1806.04606*.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7: 7.

Lee, C.-Y.; Xie, S.; Gallagher, P.; Zhang, Z.; and Tu, Z. 2015. Deeply-supervised nets. In *Artificial intelligence and statistics*, 562–570. PMLR.

Lee, K.; Zhu, Y.; Sohn, K.; Li, C.-L.; Shin, J.; and Lee, H. 2020. I-mix: A domain-agnostic strategy for contrastive representation learning. *arXiv preprint arXiv:2010.08887*.

Li, T.; Cao, P.; Yuan, Y.; Fan, L.; Yang, Y.; Feris, R. S.; Indyk, P.; and Katabi, D. 2022. Targeted supervised contrastive learning for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6918–6928.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Linsker, R. 1989. An application of the principle of maximum information preservation to linear systems. In *Advances in neural information processing systems*, 186–194.

Liu, W.; Wen, Y.; Yu, Z.; and Yang, M. 2016. Large-margin softmax loss for convolutional neural networks. *arXiv preprint arXiv:1612.02295*.

Luo, P.; Wang, X.; Shao, W.; and Peng, Z. 2018. Towards understanding regularization in batch normalization. *arXiv preprint arXiv:1809.00846*.

Nguyen, X.; Wainwright, M. J.; and Jordan, M. I. 2010. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11): 5847–5861.

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

Phuong, M.; and Lampert, C. H. 2019. Distillation-based training for multi-exit architectures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1355–1364.

Rasmus, A.; Valpola, H.; Honkala, M.; Berglund, M.; and Raiko, T. 2015. Semi-supervised learning with ladder networks. *arXiv preprint arXiv:1507.02672*.

Ruiz, A.; and Verbeek, J. 2021. Anytime inference with distilled hierarchical neural ensembles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9463–9471.

Saxe, A. M.; Bansal, Y.; Dapello, J.; Advani, M.; Kolchinsky, A.; Tracey, B. D.; and Cox, D. D. 2019. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12): 124020.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Shwartz-Ziv, R.; and Tishby, N. 2017. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Teerapittayanon, S.; McDanel, B.; and Kung, H.-T. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2464–2469. IEEE.

Tian, Y.; Krishnan, D.; and Isola, P. 2019. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*.

Verma, V.; Luong, T.; Kawaguchi, K.; Pham, H.; and Le, Q. 2021. Towards domain-agnostic contrastive learning. In *International Conference on Machine Learning*, 10530–10541. PMLR.

Wang, W.; Zhou, T.; Yu, F.; Dai, J.; Konukoglu, E.; and Van Gool, L. 2021. Exploring cross-image pixel contrast for semantic segmentation. *arXiv preprint arXiv:2101.11939*.

Wu, J.; Hu, W.; Xiong, H.; Huan, J.; Braverman, V.; and Zhu, Z. 2020. On the noisy gradient descent that generalizes as sgd. In *International Conference on Machine Learning*, 10367–10376. PMLR.

Yang, L.; Han, Y.; Chen, X.; Song, S.; Dai, J.; and Huang, G. 2020. Resolution adaptive networks for efficient inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2369–2378.

Yao, A.; and Sun, D. 2020. Knowledge transfer via dense cross-layer mutual-distillation. In *European Conference on Computer Vision*, 294–311. Springer.

Yeung, R. W. 1991. A new outlook on Shannon's information measures. *IEEE transactions on information theory*, 37(3): 466–474.

You, Y.; Gitman, I.; and Ginsburg, B. 2017. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhang, D.; Zhang, H.; Tang, J.; Hua, X.-S.; and Sun, Q. 2021. Self-regulation for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6953–6963.

Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; and Ma, K. 2019a. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3713–3722.

Zhang, L.; Tan, Z.; Song, J.; Chen, J.; Bao, C.; and Ma, K. 2019b. SCAN: A scalable neural networks framework towards compact and efficient models. *arXiv preprint arXiv:1906.03951*.

Zheng, M.; Wang, F.; You, S.; Qian, C.; Zhang, C.; Wang, X.; and Xu, C. 2021. Weakly supervised contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10042–10051.