OPERA: Omni-Supervised Representation Learning with Hierarchical Supervisions

Anonymous authors
Paper under double-blind review

Abstract

The pretrain-finetune paradigm in modern computer vision facilitates the success of self-supervised learning, which tends to achieve better transferability than supervised learning. However, with the availability of massive labeled data, a natural question emerges: *how to train a better model with **both** self and full supervision signals?* In this paper, we propose **O**mni-su**PE**rvised **R**epresentation le**A**rning with hierarchical supervisions (**OPERA**) as a solution. We provide a unified perspective of supervisions from labeled and unlabeled data and propose a unified framework of fully supervised and self-supervised learning. We extract a set of hierarchical proxy representations for each image and impose self and full supervisions on the corresponding proxy representations. Extensive experiments on both convolutional neural networks and vision transformers demonstrate the superiority of OPERA in image classification, segmentation, and object detection.[1]

1 Introduction

Learning good representations is a significant yet challenging task in deep learning (Chen & He, 2021; Zheng et al., 2021; He et al., 2020). Researchers have developed various ways to adapt to different supervisions, such as fully supervised (Oh et al., 2018; Kim et al., 2020b; Wang et al., 2016; Verma et al., 2019), self-supervised (Wang & Gupta, 2015; Ye & Shen, 2020; Grill et al., 2020; Chen et al., 2020a), and semi-supervised learning (Xu et al., 2021; Zhang et al., 2021; Wang et al., 2022b). They serve as fundamental procedures in various tasks including image classification (Deng et al., 2019; Zhang et al., 2018; Yun et al., 2019), semantic segmentation (Grill et al., 2020; Strudel et al., 2021), and object detection (He et al., 2017; Yang et al., 2019; Carion et al., 2020).

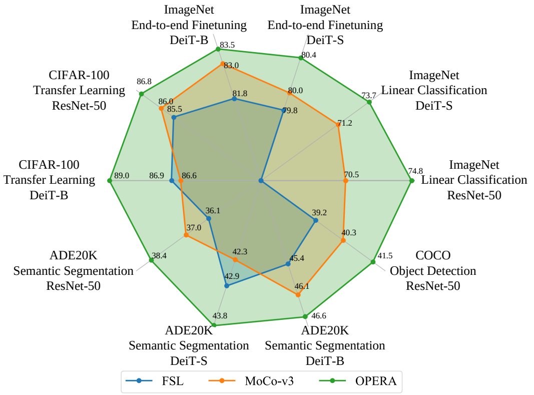
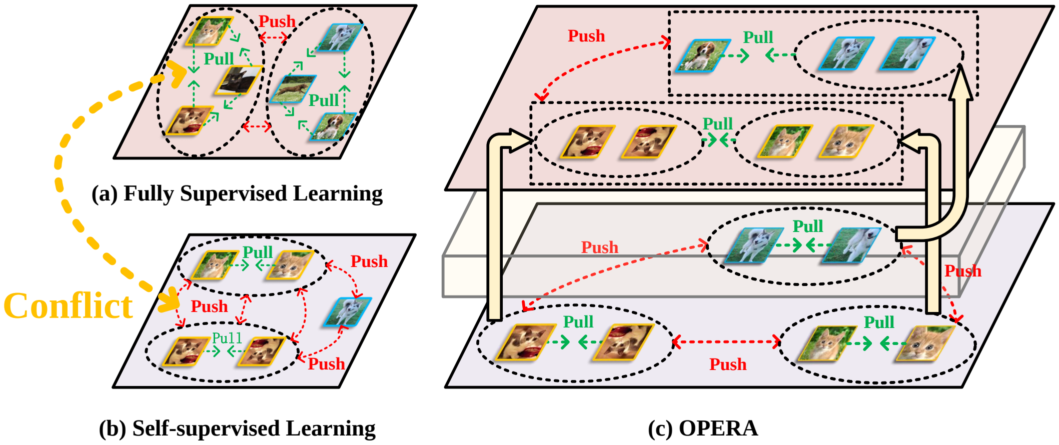


Figure 1: The proposed OPERA outperforms both fully supervised and self-supervised counterparts on various downstream tasks.

Fully supervised learning (FSL) has always been the default choice for representation learning, which learns from discriminating samples with different ground-truth labels. However, this dominance begins to fade with the rise of the pretrain-finetune paradigm in modern computer vision. Under such a paradigm, researchers usually pretrain a network on a large dataset first and then transfer it to downstream tasks (He et al., 2021; Chu et al., 2021; He et al., 2020; Chen & He, 2021). This advocates transferability more than discriminativeness of the learned representations. This preference nurtures the recent success of self-supervised learning (SSL) methods with contrastive objective (He et al., 2020; Xie et al., 2021; Grill et al., 2020; Chen et al., 2020a; Wang & Qi, 2022). They require two views (augmentations) of the same image to be consistent and distinct from other images in the representation space. This instance-level supervision is said to obtain more general and thus transferable representations (Ericsson et al., 2021; Islam et al., 2021). The ability to learn without human-annotated labels also greatly popularizes self-supervised contrastive learning. Despite

[1] Code is provided in the supplementary material.



(a) Fully Supervised Learning

(b) Self-supervised Learning

(c) OPERA

Figure 2: Comparisons of different learning strategies. Fully supervised learning (a) and self-supervised learning (b) constrain images at the class level and instance level, respectively. They conflict with each other for different images from the same class. OPERA imposes hierarchical supervisions on hierarchical spaces and uses a transformation to resolve the supervision conflicts.

its advantages, we want to explore *whether combining self-supervised signals*[2] *with fully supervised signals further improves the transferability*, given the already availability of massive annotated labels (Russakovsky et al., 2015; Lin et al., 2014; Abu-El-Haija et al., 2016; Caesar et al., 2020).

We find that a simple combination of the self and full supervisions results in contradictory training signals. To address this, in this paper, we provide **O**mni-su**PE**rvised **R**epresentation le**A**rning with hierarchical supervisions (**OPERA**) as a solution, as demonstrated in Figure 2. We unify full and self supervisions in a similarity learning framework where they differ only by the definition of positive and negative pairs. Instead of directly imposing supervisions on the representations, we extract a hierarchy of proxy representations to receive the corresponding supervision signals. Extensive experiments are conducted with both convolutional neural networks (He et al., 2016) and vision transformers (Dosovitskiy et al., 2020) as the backbone model. We pretrain the models using OPERA on ImageNet-1K (Russakovsky et al., 2015) and then transfer them to various downstream tasks to evaluate the transferability. We report image classification accuracy with both linear probe and end-to-end finetuning on ImageNet-1K. We also conduct experiments when transferring the pretrained model to other classification tasks, semantic segmentation, and object detection. Experimental results demonstrate consistent improvements over FSL and SSL on all the downstream tasks, as shown in Figure 1. Additionally, we show that OPERA can outperform the counterpart methods even with fewer pretraining epochs (e.g., fewer than 150 epochs), demonstrating good data efficiency.

2 Related Work

Fully Supervised Representation Learning. Fully supervised representation learning (FSL) utilizes the ground-truth labels of data to learn a discriminative representation space. The general objective is to maximize the discrepancies of representations from different categories and minimize those from the same class. The softmax loss is most widely used for fully supervised representation learning (He et al., 2016; Liu et al., 2021; Deng et al., 2019; Wang et al., 2018). Various loss functions are further developed in deep metric learning (Kim et al., 2020b; Wang et al., 2019; Hu et al., 2014; Movshovitz-Attias et al., 2017; Teh et al., 2020), but are doubtful to achieve better performance for general representation learning (Musgrave et al., 2020; Boudiaf et al., 2020; Zhai & Wu, 2018). As fully supervised objectives entail strong constraints, the learned representations are usually more suitable for the specialized classification task and thus lag behind on transferability (Zhao et al., 2020; Ericsson et al., 2021; Islam et al., 2021). To alleviate this, many works devise various data augmentation methods to expand the training distribution (Zhang et al., 2018; Kim et al., 2020a;

[2] We mainly focus on self-supervised contrastive learning. In the rest of the paper, we use self-supervised learning to refer to self-supervised contrastive learning unless otherize specified for simplicity.

Chen et al., 2022; Venkataramanan et al., 2022). Recent works also explore adding more layers after the representation to avoid direct supervision (Vo & Hays, 2019; Wang et al., 2022c). Differently, we focus on effectively combining full supervision with self-supervision to improve transferability.

Self-supervised Representation Learning. Self-supervised representation learning (SSL) attracts increasing attention in recent years due to its ability to learn meaningful representation without human-annotated labels. The main idea is to train the model to perform a carefully designed label-free pretext task. Early self-supervised learning methods devised various pretext tasks including image restoration (Vincent et al., 2008; Zhang et al., 2016; Pathak et al., 2016), prediction of image rotation (Gidaris et al., 2018), and solving jigsaw puzzles (Noroozi & Favaro, 2016). They achieve fair performance but still cannot equal fully supervised learning until the arise of self-supervised contrastive learning (He et al., 2020; Chen et al., 2020a; Grill et al., 2020). The pretext task of contrastive learning is instance discrimination, i.e., to identify different views (augmentations) of the same image from those of other images. Contrastive learning methods (Chen & He, 2021; Xie et al., 2021; Wang et al., 2022a; Xie et al., 2020; Liu et al., 2020; Chen et al., 2021a; Hou et al., 2021; Liang et al., 2021) demonstrate even better transferability than fully supervised learning. This superiority is said to result from their focus on learning lower-level and thus more general features (Zhao et al., 2020; Ericsson et al., 2021; Islam et al., 2021). Very recently, masked image modeling (MIM) (He et al., 2021; Zhou et al., 2021; Xie et al., 2022) emerges as a strong competitor to contrastive learning, which trains the model to correctly predict the masked parts of the input image. In this paper, we mainly focus on contrastive learning in self-supervised learning. Our framework can be extended to other pretext tasks by inserting a new task space in our hierarchy.

Omni-supervised Representation Learning: It is worth mentioning that some existing studies have attempted to combine FSL and SSL (Radosavovic et al., 2018; Nayman et al., 2022; Wei et al., 2022). Radosavovic et al. (2018) first trained an FSL model and then performed knowledge distillation on unlabeled data. Wei et al. (2022) adopted an SSL pretrained model to generate instance labels and compute an overall similarity to train a new model. Nayman et al. (2022) proposed to finetune an SSL pretrained model using ground-truth labels in a controlled manner to enhance its transferability. Nevertheless, they do not consider the hierarchical relations between the self and full supervision. Also, they perform SSL and FSL sequentially in separate stages. Differently, OPERA unifies them in a universal perspective and imposes the supervisions on different levels of the representations. Our framework can be trained in an end-to-end manner efficiently with fewer epochs.

3 Proposed Approach

In this section, we first present a unified perspective of self-supervised learning (SSL) and fully supervised learning (FSL) under a similarity learning framework. We then propose OPERA to impose hierarchical supervisions on the corresponding hierarchical representations for better transferability. Lastly, we elaborate on the instantiation of the proposed OPERA framework.

3.1 Unified Framework of Similarity Learning

Given an image space $\mathcal{X} \subset \mathcal{R}^{H\times W\times C}$, deep representation learning trains a deep neural network as the map to their a representation space $\mathcal{Y} \subset \mathcal{R}^{D\times 1}$. Fully supervised learning and self-supervised learning are two mainstream representation learning approaches in modern deep learning. FSL utilizes the human-annotated labels as explicit supervision to train a discriminative classifier. Differently, SSL trains models without ground-truth labels. The widely used contrastive learning (*e.g.*, MoCo-v3 (Chen et al., 2021b)) obtains meaningful representations by maximizing the similarity between random augmentations of the same image.

Generally, FSL and SSL differ in both the supervision form and optimization objective. To integrate them, we first provide a unified similarity learning framework to include both training objectives:

$$J(\mathcal{Y},\mathcal{P},\mathcal{L}) = \sum_{\mathbf{y}\in\mathcal{Y},\mathbf{p}\in\mathcal{P},l\in\mathcal{L}} [-w_p \cdot I(l_{\mathbf{y}}, l_{\mathbf{p}}) \cdot s(\mathbf{y},\mathbf{p}) + w_n \cdot (1 - I(l_{\mathbf{y}}, l_{\mathbf{p}})) \cdot s(\mathbf{y},\mathbf{p})], \quad (1)$$

where $w_p \geq 0$ and $w_n \geq 0$ denote the coefficients of positive and negative pairs, $l_{\mathbf{y}}$ and $l_{\mathbf{p}}$ are the labels of the samples, and $s(\mathbf{y},\mathbf{p})$ defines the pairwise similarity between $\mathbf{y}$ and $\mathbf{p}$. $I(a, b)$ is an indicator function which outputs 1 if $a = b$ and 0 otherwise. $\mathcal{L}$ is the label space, and $\mathcal{P}$ can be the

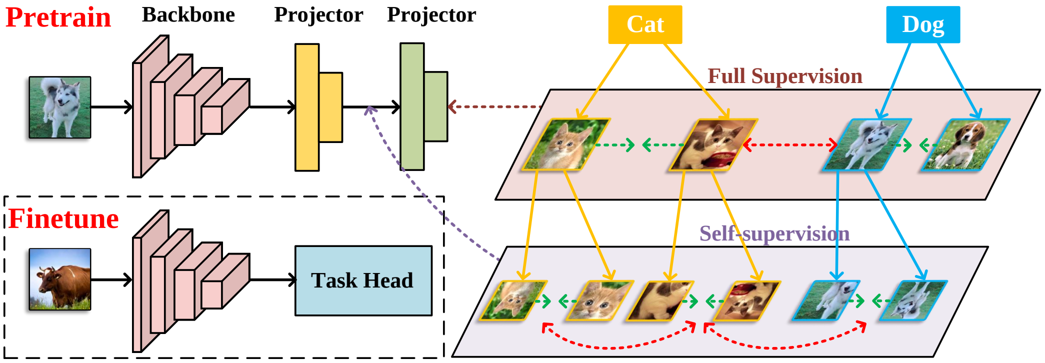


Figure 3: An illustration of the proposed OPERA framework. We impose perform SSL and FSL on the corresponding proxy representations, respectively. OPERA combines both supervisions to balance instance-level and class-level information for the backbone in an end-to-end manner.

same as $\mathcal{Y}$, a transformation of $\mathcal{Y}$, or a learnable class prototype space. For example, to obtain the softmax objective widely employed in FSL (He et al., 2016; Touvron et al., 2021), we can set:

$$w_p = 1, w_n = \frac{exp(s(\mathbf{y},\mathbf{p}))}{\sum_{l_{\mathbf{p}'}\neq l_{\mathbf{y}}} exp(s(\mathbf{y},\mathbf{p}'))}, \tag{2}$$

where $s(\mathbf{y},\mathbf{p}) = \mathbf{y}\cdot\mathbf{p}$, and $\mathbf{p}$ to be the row vector in the classifier matrix $\mathbf{W}$. For the infoNCE loss used in contrastive learning (Van den Oord et al., 2018; He et al., 2020; Khosla et al., 2020), we set:

$$w_p = \frac{1}{\tau}\frac{\sum_{l_{l_{\mathbf{p}'}\neq \mathbf{y}}} exp(s(\mathbf{y},\mathbf{p}')/\tau)}{exp(s(\mathbf{y},\mathbf{p})/\tau) + \sum_{l_{\mathbf{p}'}\neq l_{\mathbf{y}}} exp(s(\mathbf{y},\mathbf{p}')/\tau)}, w_n = \frac{1}{\tau}\frac{exp(s(\mathbf{y},\mathbf{p})/\tau)}{exp(s(\mathbf{y},\mathbf{p})/\tau) + \sum_{l_{\mathbf{p}'}\neq l_{\mathbf{y}}} exp(s(\mathbf{y},\mathbf{p}')/\tau)} \tag{3}$$

where τ is the temperature hyper-parameter. We refer to Wang et al. (2019) for more details.

Under the unified training objective Eq. (1), the main difference between FSL and SSL lies in the definition of the label space $\mathcal{L}^{full}$ and $\mathcal{L}^{self}$. For the labels $l^{full} \in \mathcal{L}^{full}$ in FSL, $l_i^{full} = l_j^{full}$ only if they are from the same ground-truth category. For the labels $l^{self} \in \mathcal{L}^{self}$ in SSL, $l_i^{self} = l_j^{self}$ only if they are the augmented views of the same image.

3.2 Hierarchical Supervisions on Hierarchical Representations

With the same formulation of the training objective, a naive way to combine the two training signals is to simply add them:

$$\begin{aligned} J^{naive}(\mathcal{Y},\mathcal{P},\mathcal{L}) = \sum_{\mathbf{y}\in\mathcal{Y},\mathbf{p}\in\mathcal{P},l\in\mathcal{L}} [&-w_p^{self}\cdot I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})\cdot s(\mathbf{y},\mathbf{p}) + w_n^{self}\cdot(1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}))\cdot s(\mathbf{y},\mathbf{p}) \\ &-w_p^{full}\cdot I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full})\cdot s(\mathbf{y},\mathbf{p}) + w_n^{full}\cdot(1 - I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}))\cdot s(\mathbf{y},\mathbf{p})]. \end{aligned} \tag{4}$$

For $\mathbf{y}$ and $\mathbf{p}$ from the same class, i.e., $I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}) = 0$ and $I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) = 1$, the training loss is:

$$J^{naive}(\mathbf{y},\mathbf{p},\mathbf{l}) = (w_n^{self} - w_p^{full})\cdot s(\mathbf{y},\mathbf{p}). \tag{5}$$

This indicates the two training signals are contradictory and may neutralize each other. This is particularly harmful if we adopt similar loss functions for fully supervised and self-supervised learning, i.e., $w_n^{self} \approx w_p^{full}$, and thus $J^{naive}(\mathbf{y},\mathbf{p},\mathbf{l}) \approx 0$.

Existing methods (Nayman et al., 2022; Wei et al., 2022; Wang et al., 2022c) address this by subsequently imposing the two training signals. They tend to first obtain a self-supervised pretrained model and then use the full supervision to tune it. Differently, we propose a more efficient way to adaptively balance the two weights so that we can simultaneously employ them:

$$J^{adap}(\mathbf{y},\mathbf{p},\mathbf{l}) = (w_n^{self}\cdot\alpha - w_p^{full}\cdot\beta)\cdot s(\mathbf{y},\mathbf{p}), \tag{6}$$

where α and β are modulation factors that can be dependent on $\mathbf{y}$ and $\mathbf{p}$ for more flexibility. However, it remains challenging to design the specific formulation of α and β.

Considering that the two label spaces are entangled and demonstrate a hierarchical structure:

$$I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}) = 1 \implies I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) = 1, \tag{7}$$

i.e., the two augmented views of the same image must share the same category label, we transform the image representation into proxy representations in an instance space and a class space to construct a hierarchical structure. Formally, we apply two transformations $\mathcal{Y}$ sequentially:

$$\mathcal{Y}^{self} = g(\mathcal{Y}), \quad \mathcal{Y}^{full} = h(\mathcal{Y}^{self}), \tag{8}$$

where $g(\cdot)$ and $h(\cdot)$ denote the mapping functions. We extract the class representations following the instance representations since full supervision encodes higher-level features than self-supervision.

We then impose the self and full supervision on the instance space and class space, respectively, to formulate the overall training objective for the proposed OPERA:

$$J^{O}(\mathcal{Y}, \mathcal{P}, \mathcal{L}) = J^{self}(\mathcal{Y}^{self}, \mathcal{P}^{self}, \mathcal{L}^{self}) + J^{full}(\mathcal{Y}^{full}, \mathcal{P}^{full}, \mathcal{L}^{full}). \tag{9}$$

We will show in the next subsection that this objective naturally implies Eq. (6), which implicitly and adaptively balances the self and full supervisions in the representation space.

3.3 Omni-supervised Representation Learning

To effectively combine the self and full supervision to learn representations, OPERA further extracts a set of proxy representations hierarchically to receive the corresponding training signal, as illustrated in Figure 3. Despite its simplicity and efficiency, it is not clear how it achieves balances between the two supervision signals and how it resolves the contradiction demonstrated in Eq. (5).

To thoroughly understand the effect of Eq. (9) on the image representations, we project it back on the representation space $\mathcal{Y}$ and obtain an equivalent training objective in $\mathcal{Y}$.

Proposition 1. *Assume using linear projection as the transformation between representation spaces, i.e., $g(\boldsymbol{y}) = \boldsymbol{W}_g \boldsymbol{y}$ and $h(\boldsymbol{y}) = \boldsymbol{W}_h \boldsymbol{y}$, where $\boldsymbol{W}_g$ and $\boldsymbol{W}_h$ are learnable parameters. Optimizing Eq. (9) is equivalent to optimizing the following objective on the original representation space $\mathcal{Y}$:*

$$\begin{aligned} J(\mathcal{Y}, \mathcal{P}, \mathcal{L}) = \sum_{\mathbf{y}\in\mathcal{Y}, \mathbf{p}\in\mathcal{P}, l\in\mathcal{L}} [& I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}) \cdot I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) \cdot (-w_p^{self}\alpha(\boldsymbol{W}_g) - w_p^{full}\beta(\boldsymbol{W}_g, \boldsymbol{W}_h)) \cdot s(\mathbf{y}, \mathbf{p}) \\ & + (1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) \cdot (w_n^{self}\alpha(\boldsymbol{W}_g) - w_p^{full}\beta(\boldsymbol{W}_g, \boldsymbol{W}_h)) \cdot s(\mathbf{y}, \mathbf{p}) \\ & + (1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot (1 - I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full})) \cdot (w_n^{self}\alpha(\boldsymbol{W}_g) + w_n^{full}\beta(\boldsymbol{W}_g, \boldsymbol{W}_h)) \cdot s(\mathbf{y}, \mathbf{p})], \end{aligned} \tag{10}$$

where $\alpha(\boldsymbol{W}_g)$ and $\beta(\boldsymbol{W}_g, \boldsymbol{W}_h)$ are scalars related to the transformation parameters.

We give detailed proof in Appendix A.

Remark. *Proposition 1 only considers the case without activation functions. We conjecture that the mappings $g(\cdot)$ and $h(\cdot)$ only influence the form of $\beta(\cdot, \cdot)$ without altering the final conclusion.*

Proposition 1 induces two corollaries as proved in Appendix B and Appendix C.

Corollary 1. *The loss weight w on a pair of samples $(\mathbf{y}, \mathbf{p})$ satisfies:*

$$w(l_{\mathbf{y}}^{self} = l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} = l_{\mathbf{p}}^{full}) \leq w(l_{\mathbf{y}}^{self} \neq l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} = l_{\mathbf{p}}^{full}) \leq w(l_{\mathbf{y}}^{self} \neq l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} \neq l_{\mathbf{p}}^{full}). \tag{11}$$

Corollary 2. *We resolve the contradictory in Eq. (5) by adaptively adjusting the loss weight by*

$$w_n^{self} \cdot \alpha(\boldsymbol{W}_g) - w_p^{full} \cdot \beta(\boldsymbol{W}_g, \boldsymbol{W}_h). \tag{12}$$

Corollary 1 ensures that the learned representations are consistent with how humans perceive the similarities of images, i.e., the similarities between different images of the same class should be larger than those between images of different classes but smaller than those between the views of the same images. Corollary 2 demonstrates the ability of OPERA to adaptively balance the training signals of self and full supervisions.

OPERA can be trained in an end-to-end manner using both self and full supervisions. We extract proxy representations in hierarchical spaces to receive the corresponding training signals. For inference, we discard the proxy representations and directly add the task head on the image representation space $\mathcal{Y}$. We give an example of an instantiation of OPERA in Appendix D.

Table 1: Top-1 and top-5 accuracies (%) under the linear classification protocol on ImageNet.

Method	Batch Size	Pretraining	Finetuning	Backbone	Top-1 Acc	Top-5 Acc
MoCo-v1	256	200	100	R50	60.6	-
MoCo-v2	256	200	100	R50	67.5	-
MoCo-v2	256	800	100	R50	71.1	-
SimCLR	4096	100	1000	R50	69.3	89.0
SimSiam	256	800	100	R50	71.3	-
BYOL	4096	1000	80	R50	74.3	91.6
MoCo-v3†	1024	300	90	R50	70.5	90.0
OPERA	1024	300	90	R50	**74.8**	**91.9**
MoCo-v3†	1024	300	90	DeiT-S	71.2	90.3
OPERA	1024	300	90	DeiT-S	**73.7**	**91.3**

Table 2: Top-1 and top-5 accuracies (%) under the end-to-end finetuning protocol on ImageNet.

Method	Batch Size	Pretraining	Finetuning	Backbone	Top-1 Acc	Top-5 Acc
Supervised	1024	-	300	DeiT-S	79.8	95.0
Supervised	1024	-	300	DeiT-B	81.8	95.6
DINO†	1024	300	300	DeiT-B	82.8	96.3
MoCo-v3†	1024	300	100	DeiT-S	78.8	94.6
OPERA	1024	300	100	DeiT-S	**80.0**	**95.1**
MoCo-v3†	1024	300	150	DeiT-S	79.1	94.6
OPERA	1024	300	150	DeiT-S	**80.4**	**95.3**
MoCo-v3†	1024	300	200	DeiT-S	80.0	95.2
OPERA	1024	300	200	DeiT-S	**80.8**	**95.5**
MoCo-v3†	1024	300	150	DeiT-B	82.1	95.9
OPERA	1024	300	150	DeiT-B	**82.6**	**96.2**
MoCo-v3†	2048	300	150	DeiT-B	82.7	96.3
OPERA	2048	300	150	DeiT-B	**83.1**	**96.4**
MoCo-v3†	4096	300	150	DeiT-B	83.0	96.3
OPERA	4096	300	150	DeiT-B	**83.5**	**96.5**

4 Experiments

In this section, we conducted extensive experiments to evaluate the performance of the proposed OPERA framework. We pretrained the network using OPERA on the ImageNet-1K (Russakovsky et al., 2015) (IN) dataset and then evaluated its performance on different tasks. We also provide in-depth ablation studies to analyze the effectiveness of OPERA. All experiments were conducted with the PyTorch (Paszke et al., 2019) library using RTX 3090 GPUs.

4.1 Experimental Setup

Datasets. We pretrain our model on the training set of ImageNet-1K (Russakovsky et al., 2015) containing 1,200,000 samples of 1,000 categories. We then evaluate the linear probe and end-to-end finetuning performance on the validation set consisting of 50,000 images. For transferring to other classification tasks, we adopt CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), Oxford Flowers-102 (Nilsback & Zisserman, 2008), and Oxford-IIIT-Pets (Parkhi et al., 2012). For other downstream tasks, we use ADE20K (Zhou et al., 2019) for semantic segmentation and COCO (Lin et al., 2014) for object detection and instance segmentation.

Implementation Details. We mainly applied our OPERA to MoCo-v3 (Chen et al., 2021b). We added an extra MLP block after the predictor of the online network, which is composed of two fully-connected layers with a batch normalization layer and a ReLU layer. The hidden dimension of the MLP block was set to 256 while the output dimension was $1,000$. We trained ResNet50 (He et al., 2016) (R50) and DeiTs (Touvron et al., 2021) (DeiT-S and DeiT-B) as our backbone with

Table 3: Top-1 accuracy (%) of the transfer learning on other classification datasets.

Method	Pretraining	Finetuning	Backbone	CIFAR-10	CIFAR-100	Flowers-102	Pets
Supervised†	300	100	R50	97.6	85.5	95.6	92.2
MoCo-v3†	300	100	R50	97.8	86.0	93.7	90.0
OPERA	300	100	R50	**98.2**	**86.8**	**95.6**	**92.7**
Supervised†	300	100	DeiT-S	98.4	86.9	95.4	93.0
MoCo-v3†	300	100	DeiT-S	97.9	86.6	90.3	90.1
OPERA	300	100	DeiT-S	**98.6**	**89.0**	**95.5**	**93.3**

Table 4: Experimental results of semantic segmentation on ADE20K (160k schedule).

Method	Pretraining	Backbone	Batch Size	mIoU	mAcc	aAcc
Supervised	300	R50	1024	36.1	45.4	77.5
MoCo-v3†	300	R50	1024	37.0	47.0	77.6
OPERA	300	R50	1024	**37.9**	**48.1**	**77.9**
OPERA	300	R50	4096	**38.4**	**48.5**	**78.1**
Supervised	300	DeiT-S	1024	42.9	53.9	80.3
MoCo-v3†	300	DeiT-S	1024	42.3	53.5	80.6
OPERA	300	DeiT-S	1024	**43.6**	**54.4**	**80.9**
OPERA	300	DeiT-S	4096	**43.8**	**54.6**	**80.9**
Supervised	300	DeiT-B	1024	45.4	56.5	81.4
MoCo-v3†	300	DeiT-B	1024	44.4	55.1	81.5
OPERA	300	DeiT-B	1024	**45.2**	**55.9**	**81.9**
MoCo-v3†	300	DeiT-B	2048	45.2	55.5	81.9
OPERA	300	DeiT-B	2048	**45.9**	**56.7**	**82.0**
MoCo-v3†	300	DeiT-B	4096	46.1	56.7	82.1
OPERA	300	DeiT-B	4096	**46.6**	**57.2**	**82.1**

a batch size of 1024, 2048, and 4096. We adopted LARS (You et al., 2017) as the optimizer for R50 and AdamW (Loshchilov & Hutter, 2018) for DeiT. We set the other settings the same as the original MoCo-v3 for fair comparisons. In the following experiments, † denotes our reproduced results with the same settings. The bold number highlights the improvement of OPERA compared with the associated method, and the red number indicates the best performance.

4.2 Main Results

Linear Probe Evaluation on ImageNet. We evaluated OPERA using the linear probe protocol, where we trained a classifier on top of the frozen representation. We also compared OPERA with existing SSL methods including MoCo-v1 (He et al., 2020), MoCo-v2 (Chen et al., 2020b), SimCLR (Chen et al., 2020a), SimSiam (Chen & He, 2021), and BYOL (Grill et al., 2020), as shown in Table 1. We achieved 74.8% and 73.7% top-1 accuracy using R50 and DeiT-S, respectively. This demonstrates the discriminative ability of the learned representations using OPERA.

End-to-end Finetuning on Imagenet. Having pretrained, we finetuned the backbone on the training set of ImageNet. We provide the results in Table 2 with diverse batch sizes and end-to-end finetuning epochs. We see that OPERA consistently achieves better performance under the same setting compared with the MoCo-v3 baseline and DINO (Caron et al., 2021).

Transfer to Other Classification Tasks. We transferred the pretrained network to other classification tasks including CIFAR-10, CIFAR-100, Oxford Flowers-102, and Oxford-IIIT-Pets. We fixed the finetuning epochs to 100 following Chen et al. (2021b) and reported the top-1 accuracy in Table 3. We observe that OPERA obtains better results on four datasets with both R50 and DeiT-S. Though MoCo-v3 does not show consistent improvement compared to supervised training on these tasks, our OPERA demonstrates clear superiority. The results demonstrate that OPERA learns generic representations from ImageNet which can widely transfer to smaller classification datasets.

Transfer to Semantic Segmentation. We also transferred the OPERA-pretrained network to semantic segmentation on ADE20K, which aims at classifying each pixel of an image. We adopted the MMSegmentaion (Contributors, 2020) codebase to conduct the experiments under the same

Table 5: Experimental results of object detection and instance segmentation on the COCO dataset. (Mask R-CNN, R50-FPN, 1 × schedule)

Method	Pretraining	Batch Size	$\mathbf{AP}^{bb}$	$\mathbf{AP}^{bb}_{50}$	$\mathbf{AP}^{bb}_{75}$	$\mathbf{AP}^{mk}$	$\mathbf{AP}^{mk}_{50}$	$\mathbf{AP}^{mk}_{75}$
Rand. Init.	-	1024	31.0	49.5	33.2	28.5	46.8	30.4
Supervised	300	1024	38.2	58.8	41.4	34.7	55.7	37.2
MoCo-v3†	300	1024	38.9	58.8	42.4	35.2	56.0	37.7
OPERA	300	1024	**39.2**	**59.2**	**42.6**	**35.9**	**56.2**	**38.1**
OPERA	300	4096	**39.3**	**59.3**	**42.9**	**36.0**	**56.4**	**38.1**

Table 6: Experimental results of object detection and instance segmentation on the COCO dataset (Mask R-CNN, R50-FPN, 2 × schedule).

Method	Pretraining	Backbone	$\mathbf{AP}^{bb}$	$\mathbf{AP}^{bb}_{50}$	$\mathbf{AP}^{bb}_{75}$	$\mathbf{AP}^{mk}$	$\mathbf{AP}^{mk}_{50}$	$\mathbf{AP}^{mk}_{75}$
Rand. Init.	-	1024	36.7	56.7	40.0	33.7	53.8	35.9
Supervised	300	1024	39.2	59.6	42.8	35.4	56.4	37.9
MoCo-v3†	300	1024	40.3	60.0	44.3	36.5	57.4	39.0
OPERA	300	1024	**41.2**	**60.7**	**45.0**	**36.9**	**57.7**	**39.5**
OPERA	300	4096	**41.5**	**61.2**	**45.5**	**37.3**	**58.2**	**39.9**

setting. Specifically, we equipped R50 with FCN (Shelhamer et al., 2017) and DeiTs with UPerNet (Xiao et al., 2018). We used a learning schedule of 160k. We provided the experimental results in Table 4. We observe consistent improvements over both supervised learning and MoCo-v3 with both R50 and DeiTs. Particularly, MoCo-v3 performs worse than the supervised model with DeiT-S (-0.6 mIoU) while OPERA still outperforms supervised learning with a large margin (+0.9 mIoU).

Transfer to Object Detection and Instance Segmentation. We further evaluated the transferability of OPERA to object detection and instance segmentation on COCO. We performed finetuning and evaluation on COCO$_{train2017}$ and COCO$_{val2017}$, respectively, using the MMDetection (Chen et al., 2019) codebase. We adopted Mask R-CNN (He et al., 2017) with R50-FPN as the detection model. We reported the performance using the 1 × schedule (12 epochs) and 2 × schedule (24 epochs) in Tables 5 and 6, respectively. We observe that both OPERA and MoCo-v3 demonstrate remarkable advantages compared with random initialization as well as supervised learning on both object detection and instance segmentation. Additionally, OPERA further improves MoCo-v3 by a relatively large margin on both training schedules, indicating that OPERA can generalize well on detection and instance segmentation datasets.

4.3 Ablation Study

To further understand the proposed OPERA, we conducted various ablation studies to evaluate its effectiveness. We mainly focus on end-to-end finetuning on ImageNet for representation discriminativeness and semantic segmentation on ADE20K for representation transferability evaluation. We fixed the number of finetuning epochs to 100 for ImageNet and used a learning schedule of 160k based on UPerNet (Xiao et al., 2018) on ADE20K.

Arrangements of Supervisions. As discussed in the paper, the arrangements of supervisions are significant to the quality of the representation. We thus conducted experiments with different arrangements of supervisions to analyze their effects, as illustrated in Figure 4. We maintained the basic structure of contrastive learning and impose the fully-supervised training signal on three different positions. Note that Figure 4 only shows the online network of the framework. Specifically, arrangement A obtains the class-level representation from the backbone and directly imposes the fully-supervised learning signal. Differently, arrangement B simultaneously extracts the class-level representation and the instance-level representation with an MLP structure from the projector. Arrangement C denotes the proposed OPERA framework in our main experiments. The experimental results are shown in the right of Figure 4. We observe that arrangement A achieves the highest classification performance on ImageNet. This is because the full supervision is directly imposed on the backbone feature, which extracts more class-level information during pretraining. However, both arrangements A and B perform much worse on the downstream semantic segmentation task. They ignore the underlying hierarchy of the supervisions and do not apply the stronger supervision (full supervision) after the weaker supervision (self-supervision). The learned representation tends to abandon more instance-level information but obtain more task-specific knowledge, which is not

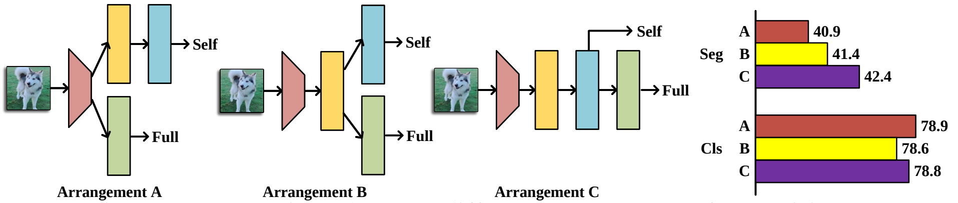


Figure 4: Comparisons between different arrangements of supervisions.

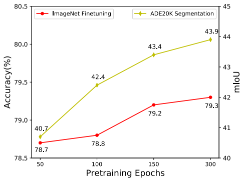


Figure 5: Pretraining epochs.

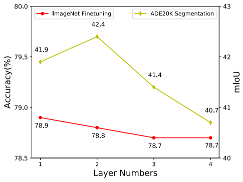


Figure 6: Layer numbers of MLP.

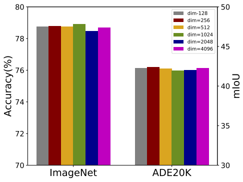


Figure 7: Embedding dimensions.

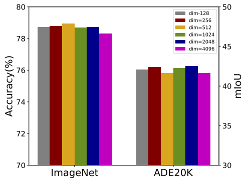


Figure 8: Hidden dimensions of MLP.

beneficial to the transfer learning tasks. Instead, our OPERA (arrangement C) achieves a better balance of class-level and instance-level information learning.

Pretraining Epochs. We conducted experiments with different pretraining epochs on ImageNet and provided corresponding results in Figure 5. We observe that both tasks perform better with longer pretraining epochs. Particularly, the performance on semantic segmentation is more sensitive to the number of pretraining epochs compared with ImageNet finetuning, indicating that it takes longer for learning instance-level knowledge. Note that the finetuning accuracy reaches 78.7% with only 50 pretraining epochs, which demonstrates the efficiency of OPERA.

Layer Numbers of MLP. We evaluated OPERA with different numbers of fully-connected layers in the final MLP block, as illustrated in Figure 6. We observe that the classification performance generally decreases with more layers deployed. This demonstrates that the class-level supervision is weakened after the MLP block so that the model extracts less class-level information with more layers. For semantic segmentation, the mIoU improves (+0.5) when the layer number increases from 1 to 2, indicating that weaker class-level supervision boosts the transferability of the representation. Still, the performance drops with more layers due to the less effect of the class-level supervision.

Embedding Dimensions. The embedding dimension in our framework measures the output size of the online network projector. We tested the performance using a dimension of 128, 256, 512, 1024, 2048, and 4096 for the embedding and provide the results in Figure 7. We observe that the ImageNet accuracy gradually increases before the embedding dimension reaches 512. In addition, the model achieves the best segmentation performance when the dimension is 256. This indicates that larger dimensions do not necessarily enhance the results because of the information redundancy. Therefore, we adopted the embedding dimension of 256 in the main experiments for the best trade-off between model performances and training efficiency.

Hidden Dimensions of MLP. The hidden dimension of MLP corresponds to the output size of the first linear layer. We fixed the other settings and used a dimension of 128, 256, 512, 1024, 2048, and 4096 for comparison, as shown in Figure 8. We see that enlarging the hidden dimension would not necessarily benefit the two tasks, indicating that OPERA is not sensitive to the hidden dimensions of MLP. Therefore, we employ a dimension of 256 for the main experiments.

5 Conclusion

In this paper, we have presented an omni-supervised representation learning with hierarchical supervisions (OPERA) framework to effectively combine fully-supervised and self-supervised contrastive learning. We provide a unified perspective of both supervisions and impose the corresponding supervisions on the hierarchical proxy representations in an end-to-end manner. We have conducted extensive experiments on classification and other downstream tasks including semantic segmentation and object detection to evaluate the effectiveness of our framework. The experimental results have demonstrated the superior classification and transferability of OPERA over both fully supervised learning and self-supervised contrastive learning. In the future, we will seek to integrate other self-supervised signals such as masked image modeling to further improve the performance.

REFERENCES

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv*, abs/1609.08675, 2016.

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. *arXiv*, abs/2204.07141, 2022.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2021.

Malik Boudiaf, Jérôme Rony, Imtiaz Masud Ziko, Eric Granger, Marco Pedersoli, Pablo Piantanida, and Ismail Ben Ayed. Metric learning: cross-entropy vs. pairwise losses. In *ECCV*, 2020.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pp. 11621–11631, 2020.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pp. 213–229, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *CVPR*, pp. 9650–9660, 2021.

Jie-Neng Chen, Shuyang Sun, Ju He, Philip HS Torr, Alan Yuille, and Song Bai. Transmix: Attend to mix for vision transformers. In *CVPR*, pp. 12135–12144, 2022.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv*, abs/1906.07155, 2019.

Kai Chen, Lanqing Hong, Hang Xu, Zhenguo Li, and Dit-Yan Yeung. Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In *ICCV*, pp. 7546–7554, 2021a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pp. 1597–1607, 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, pp. 15750–15758, 2021.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv*, abs/2003.04297, 2020b.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *CVPR*, pp. 9640–9649, 2021b.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers, 2021.

MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark, 2020.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pp. 4690–4699, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *CVPR*, pp. 5414–5423, 2021.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, pp. 21271–21284, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *CVPR*, pp. 2961–2969, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv*, abs/2111.06377, 2021.

Ji Hou, Saining Xie, Benjamin Graham, Angela Dai, and Matthias Nießner. Pri3d: Can 3d priors help 2d representation learning? In *ICCV*, pp. 5693–5702, 2021.

Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *CVPR*, pp. 1875–1882, 2014.

Ashraful Islam, Chun-Fu Richard Chen, Rameswar Panda, Leonid Karlinsky, Richard Radke, and Rogerio Feris. A broad study on the transferability of visual representations with contrastive learning. In *ICCV*, pp. 8845–8855, 2021.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *NeurIPS*, pp. 18661–18673, 2020.

Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *ICML*, pp. 5275–5285, 2020a.

Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. Proxy anchor loss for deep metric learning. In *CVPR*, pp. 3238–3247, 2020b.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Hanxue Liang, Chenhan Jiang, Dapeng Feng, Xin Chen, Hang Xu, Xiaodan Liang, Wei Zhang, Zhenguo Li, and Luc Van Gool. Exploring geometry-aware contrast and clustering harmonization for self-supervised 3d object detection. In *ICCV*, pp. 3293–3302, 2021.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pp. 740–755, 2014.

Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *arXiv*, abs/2011.13677, 2020.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 10012–10022, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv*, abs/1608.03983, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *ICCV*, pp. 360–368, 2017.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *ECCV*, 2020.

Niv Nayman, Avram Golbert, Asaf Noy, Tan Ping, and Lihi Zelnik-Manor. Diverse imagenet models transfer better. *arXiv*, abs/2204.09134, 2022.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pp. 69–84, 2016.

Seong Joon Oh, Kevin P Murphy, Jiyan Pan, Joseph Roth, Florian Schroff, and Andrew C Gallagher. Modeling uncertainty with hedged instance embeddings. In *ICLR*, 2018.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pp. 8026–8037, 2019.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pp. 2536–2544, 2016.

Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *CVPR*, pp. 4119–4128, 2018.

H. Robbins and S. Monro. *A Stochastic Approximation Method*. Herbert Robbins Selected Papers, 1985.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *TPAMI*, 2017.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pp. 7262–7272, 2021.

Eu Wern Teh, Terrance DeVries, and Graham W Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *ECCV*, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pp. 10347–10357, 2021.

Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, abs/1807.03748, 2018.

Shashanka Venkataramanan, Ewa Kijak, Laurent Amsaleg, and Yannis Avrithis. Alignmixup: Improving representations by interpolating aligned features. In *CVPR*, pp. 19174–19183, 2022.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICLR*, pp. 6438–6447, 2019.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pp. 1096–1103, 2008.

Nam Vo and James Hays. Generalization in metric learning: Should the embedding layer be embedding layer? In *WACV*, pp. 589–598, 2019.

Faqiang Wang, Wangmeng Zuo, Liang Lin, David Zhang, and Lei Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *CVPR*, pp. 1288–1296, 2016.

Feng Wang, Huiyu Wang, Chen Wei, Alan Yuille, and Wei Shen. Cp2: Copy-paste contrastive pretraining for semantic segmentation. *arXiv*, abs/2203.11709, 2022a.

Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, pp. 5265–5274, 2018.

Jianfeng Wang, Thomas Lukasiewicz, Daniela Massiceti, Xiaolin Hu, Vladimir Pavlovic, and Alexandros Neophytou. Np-match: When neural processes meet semi-supervised learning. In *ICML*, pp. 22919–22934, 2022b.

Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *TPAMI*, 2022.

Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, pp. 2794–2802, 2015.

Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R Scott. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, pp. 5022–5030, 2019.

Yizhou Wang, Shixiang Tang, Feng Zhu, Lei Bai, Rui Zhao, Donglian Qi, and Wanli Ouyang. Revisiting the transferability of supervised pretraining: an mlp perspective. In *CVPR*, pp. 9183–9193, 2022c.

Longhui Wei, Lingxi Xie, Jianzhong He, Xiaopeng Zhang, and Qi Tian. Can semantic labels assist self-supervised visual representation learning? In *AAAI*, volume 36, pp. 2642–2650, 2022.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *ICCV*, pp. 8392–8401, 2021.

Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *ECCV*, pp. 574–591, 2020.

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, pp. 9653–9663, 2022.

Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. In *ICCV*, pp. 3060–3069, 2021.

Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *ICCV*, pp. 9657–9666, 2019.

Mang Ye and Jianbing Shen. Probabilistic structural latent representation for unsupervised embedding. In *CVPR*, pp. 5457–5466, 2020.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv*, abs/1708.03888, 2017.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pp. 6023–6032, 2019.

Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. *arXiv*, abs/1811.12649, 2018.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *NeurIPS*, 34, 2021.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pp. 649–666, 2016.

Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv*, abs/2006.06606, 2020.

Wenzhao Zheng, Borui Zhang, Jiwen Lu, and Jie Zhou. Deep relational metric learning. In *CVPR*, pp. 12065–12074, 2021.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127:302–321, 2019.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv*, abs/2111.07832, 2021.

A Proof of Proposition 1

Proof. We consider the overall supervision on a pair of samples $(\mathbf{y}, \mathbf{p})$ in Eq. (9), which is as follows:

$$
\begin{aligned}
J^O(\mathbf{y},\mathbf{p}) = -I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}) \cdot w_p^{self} \cdot s(\mathbf{y}^{self}, \mathbf{p}^{self}) + (1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot w_n^{self} \cdot s(\mathbf{y}^{self}, \mathbf{p}^{self}) \\
- I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) \cdot w_p^{full} \cdot s(\mathbf{y}^{full}, \mathbf{p}^{full}) + (1 - I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full})) \cdot w_n^{full} \cdot s(\mathbf{y}^{full}, \mathbf{p}^{full})
\end{aligned} \tag{13}
$$

We calculate the gradient of $J^O(\mathbf{y}, \mathbf{p})$ towards $\mathbf{y}$ as follows:

$$
\begin{aligned}
\frac{\partial J^O(\mathbf{y},\mathbf{p})}{\partial \mathbf{y}} = -I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}) \cdot w_p^{self} \cdot \boldsymbol{W}_g^T \gamma(\mathbf{y}^{self}, \mathbf{p}^{self}) + (1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot w_n^{self} \cdot \boldsymbol{W}_g^T \gamma(\mathbf{y}^{self}, \mathbf{p}^{self}) \\
-I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) \cdot w_p^{full} \cdot \boldsymbol{W}_g^T \boldsymbol{W}_h^T \gamma(\mathbf{y}^{full}, \mathbf{p}^{full}) + (1 - I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full})) \cdot w_n^{full} \cdot \boldsymbol{W}_g^T \boldsymbol{W}_h^T \gamma(\mathbf{y}^{full}, \mathbf{p}^{full})
\end{aligned} \tag{14}
$$

where $\gamma(\mathbf{y}, \mathbf{p}_p) = \frac{\partial s(\mathbf{y}, \mathbf{p}_p)}{\partial \mathbf{y}}$. For simplicity and clarity, we define $s(\mathbf{y}, \mathbf{p}) = \mathbf{y}^T \mathbf{p}$. Under such circumstances, Eq. (14) can be formulated as follows:

$$
\begin{aligned}
\frac{\partial J^O(\mathbf{y},\mathbf{p})}{\partial \mathbf{y}} = -I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}) \cdot w_p^{self} \cdot \boldsymbol{W}_g^T \boldsymbol{W}_g \mathbf{p} + (1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot w_n^{self} \cdot \boldsymbol{W}_g^T \boldsymbol{W}_g \mathbf{p} \\
-I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) \cdot w_p^{full} \cdot \boldsymbol{W}_g^T \boldsymbol{W}_h^T \boldsymbol{W}_h \boldsymbol{W}_g \mathbf{p} + (1 - I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full})) \cdot w_n^{full} \cdot \boldsymbol{W}_g^T \boldsymbol{W}_h^T \boldsymbol{W}_h \boldsymbol{W}_g \mathbf{p}
\end{aligned} \tag{15}
$$

Under such circumstances, the concrete form of Eq. (15) is determined by the label connection between $\mathbf{y}$ and $\mathbf{p}$. Specifically, when $I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}) \cdot I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) = 1$, denoting that $\mathbf{y}$ and $\mathbf{p}$ shares the same self-supervised and fully supervised label, Eq. (15) degenerates to:

$$
\frac{\partial J^O(\mathbf{y},\mathbf{p})}{\partial \mathbf{y}} = \boldsymbol{W}_g^T(-w_p^{self} \boldsymbol{I} - w_p^{full} \boldsymbol{W}_h^T \boldsymbol{W}_h) \boldsymbol{W}_g \mathbf{p} \tag{16}
$$

Similarly, when $(1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) = 1$, Eq. (15) degenerates to:

$$
\frac{\partial J^O(\mathbf{y},\mathbf{p})}{\partial \mathbf{y}} = \boldsymbol{W}_g^T(w_n^{self} \boldsymbol{I} - w_p^{full} \boldsymbol{W}_h^T \boldsymbol{W}_h) \boldsymbol{W}_g \mathbf{p} \tag{17}
$$

And when $(1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot (1 - I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full})) = 1$, Eq. (15) degenerates to:

$$
\frac{\partial J^O(\mathbf{y},\mathbf{p})}{\partial \mathbf{y}} = \boldsymbol{W}_g^T(w_n^{self} \boldsymbol{I} + w_n^{full} \boldsymbol{W}_h^T \boldsymbol{W}_h) \boldsymbol{W}_g \mathbf{p} \tag{18}
$$

Next, we consider that $\mathbf{p}$ is fixed during optimization (such as a prototype) and provide the change of $s(\mathbf{y}, \mathbf{p})$ based on Eq. (17) for example:

$$
\begin{aligned}
\Delta s^O(\mathbf{y},\mathbf{p}) \propto (\frac{\partial J^O(\mathbf{y},\mathbf{p})}{\partial \mathbf{y}})^T \cdot \mathbf{p} &= \mathbf{p}^T \boldsymbol{W}_g^T (w_n^{self} \boldsymbol{I} - w_p^{full} \boldsymbol{W}_h^T \boldsymbol{W}_h) \boldsymbol{W}_g \mathbf{p} \\
&= w_n^{self} (\mathbf{p}^{self})^T \mathbf{p}^{self} - w_p^{full} (\mathbf{p}^{full})^T \mathbf{p}^{full} \\
&= w_n^{self} \alpha(\boldsymbol{W}_g) - w_p^{full} \beta(\boldsymbol{W}_g, \boldsymbol{W}_h)
\end{aligned} \tag{19}
$$

Therefore, we formulate the above equation considering all the possible relations between the label of $\mathbf{y}$ and $\mathbf{p}$ as follows:

$$
\begin{aligned}
\Delta s^O(\mathbf{y},\mathbf{p}) \propto I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}) \cdot I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) \cdot (-w_p^{self} \alpha(\boldsymbol{W}_g) - w_p^{full} \beta(\boldsymbol{W}_g, \boldsymbol{W}_h)) \\
+(1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) \cdot (w_n^{self} \alpha(\boldsymbol{W}_g) - w_p^{full} \beta(\boldsymbol{W}_g, \boldsymbol{W}_h)) \\
+(1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot (1 - I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full})) \cdot (w_n^{self} \beta(\boldsymbol{W}_g) + w_n^{full} \alpha(\boldsymbol{W}_g, \boldsymbol{W}_h))
\end{aligned} \tag{20}
$$

For Eq. (10), we similarly consider a pair of samples $(\mathbf{y}, \mathbf{p})$ and we can obtain the gradient of $J(\mathbf{y}, \mathbf{p})$ towards $s(\mathbf{y}, \mathbf{p})$ as follows:

$$
\begin{aligned}
\frac{\partial J(\mathbf{y},\mathbf{p})}{\partial s(\mathbf{y},\mathbf{p})} = I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}) \cdot I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) \cdot (-w_p^{self} \alpha(\boldsymbol{W}_g) - w_p^{full} \beta(\boldsymbol{W}_g, \boldsymbol{W}_h)) \\
+ (1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) \cdot (w_n^{self} \alpha(\boldsymbol{W}_g) - w_p^{full} \beta(\boldsymbol{W}_g, \boldsymbol{W}_h)) \\
+ (1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot (1 - I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full})) \cdot (w_n^{self} \beta(\boldsymbol{W}_g) + w_n^{full} \alpha(\boldsymbol{W}_g, \boldsymbol{W}_h))
\end{aligned} \tag{21}
$$

The change of $s(\mathbf{y}, \mathbf{p})$ during optimization for Eq. (10) is proportional to to $\frac{\partial J(\mathbf{y},\mathbf{p})}{\partial s(\mathbf{y},\mathbf{p})}$:

$$\begin{aligned}\Delta s(\mathbf{y}, \mathbf{p}) \propto\, & I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}) \cdot I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) \cdot (-w_p^{self}\alpha(\boldsymbol{W}_g) - w_p^{full}\beta(\boldsymbol{W}_g, \boldsymbol{W}_h)) \\ & + (1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) \cdot (w_n^{self}\alpha(\boldsymbol{W}_g) - w_p^{full}\beta(\boldsymbol{W}_g, \boldsymbol{W}_h)) \\ & + (1 - I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self})) \cdot (1 - I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full})) \cdot (w_n^{self}\beta(\boldsymbol{W}_g) + w_n^{full}\alpha(\boldsymbol{W}_g, \boldsymbol{W}_h))\end{aligned} \tag{22}$$

Therefore, the optimization towards $s(\mathbf{y}, \mathbf{p})$ of Eq. (10) is equal to Eq. (9). In addition, this conclusion is also applicable to the summation form of Eq. (10) and Eq. (9), which means that Eq. (10) is an equivalent form of Eq. (9). □

B Proof of Corollary 1

Proof. With the gradient of Eq. (10) in Eq. (21), we provide the loss weight on $(\mathbf{y}, \mathbf{p})$ as follows:

$$w(l_{\mathbf{y}}^{self} = l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} = l_{\mathbf{p}}^{full}) = -w_p^{self}\alpha(\boldsymbol{W}_g) - w_p^{full}\beta(\boldsymbol{W}_g, \boldsymbol{W}_h) \tag{23}$$

$$w(l_{\mathbf{y}}^{self} \neq l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} = l_{\mathbf{p}}^{full}) = w_n^{self}\alpha(\boldsymbol{W}_g) - w_p^{full}\beta(\boldsymbol{W}_g, \boldsymbol{W}_h) \tag{24}$$

$$w(l_{\mathbf{y}}^{self} \neq l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} \neq l_{\mathbf{p}}^{full}) = w_n^{self}\alpha(\boldsymbol{W}_g) + w_n^{full}\beta(\boldsymbol{W}_g, \boldsymbol{W}_h) \tag{25}$$

Therefore, we can obtain the following two inequalities:

$$\begin{aligned}w(l_{\mathbf{y}}^{self} = l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} = l_{\mathbf{p}}^{full}) - w(l_{\mathbf{y}}^{self} \neq l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} = l_{\mathbf{p}}^{full}) \\ = -w_p^{self}\alpha(\boldsymbol{W}_g) - w_n^{self}\alpha(\boldsymbol{W}_g) \leq 0\end{aligned} \tag{26}$$

$$\begin{aligned}w(l_{\mathbf{y}}^{self} \neq l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} = l_{\mathbf{p}}^{full}) - w(l_{\mathbf{y}}^{self} \neq l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} \neq l_{\mathbf{p}}^{full}) \\ = w_p^{full}\beta(\boldsymbol{W}_g, \boldsymbol{W}_h) - w_n^{full}\beta(\boldsymbol{W}_g, \boldsymbol{W}_h) \leq 0\end{aligned} \tag{27}$$

We organize the above inequalities, which can be formulated as follows:

$$w(l_{\mathbf{y}}^{self} = l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} = l_{\mathbf{p}}^{full}) \leq w(l_{\mathbf{y}}^{self} \neq l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} = l_{\mathbf{p}}^{full}) \leq w(l_{\mathbf{y}}^{self} \neq l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} \neq l_{\mathbf{p}}^{full}). \tag{28}$$

□

C Proof of Corollary 2

Proof. For contradictory situation where $I(l_{\mathbf{y}}^{self}, l_{\mathbf{p}}^{self}) = 0$ and $I(l_{\mathbf{y}}^{full}, l_{\mathbf{p}}^{full}) = 1$, the loss weight is as follows:

$$w(l_{\mathbf{y}}^{self} \neq l_{\mathbf{p}}^{self}, l_{\mathbf{y}}^{full} = l_{\mathbf{p}}^{full}) = w_n^{self} \cdot \alpha(\boldsymbol{W}_g) - w_p^{full} \cdot \beta(\boldsymbol{W}_g, \boldsymbol{W}_h) \tag{29}$$

□

Therefore, the direction and intensity of optimization is determined by the values of $alpha(\boldsymbol{W}_g)$ and $\beta(\boldsymbol{W}_g, \boldsymbol{W}_h)$. For example, when $w_n^{self} \cdot \alpha(\boldsymbol{W}_g) - w_p^{full} \cdot \beta(\boldsymbol{W}_g, \boldsymbol{W}_h) < 0$, the model increases the similarity between $\mathbf{y}$ and $\mathbf{p}$ during optimization. Consequently, OPERA adaptively adjusts the loss weight between each pair of samples to resolve the contradiction in Eq. (5).

D Instantiation of OPERA

We present the instantiation of the proposed omni-supervised representation learning with hierarchical supervisions. In the pretraining procedure, we extract hierarchical proxy representations for each image $\mathbf{x}_i$ in our model, denoted as $\{\mathbf{y}_i^{self}, \mathbf{y}_i^{full}\}$. We conduct self-supervised learning with the instance-level label l_i^{self} on the instance-level representation $\mathbf{y}_i^{self}$ and the class-level label l_i^{full} is imposed on $\mathbf{y}_i^{full}$. The overall objective of our framework follows Eq. (9) and OPERA can be optimized in an end-to-end manner. During finetuning, the downstream task head is directly applied to the learned representations $\mathcal{Y}$. The transfer learning includes image classification and other dense prediction tasks such as semantic segmentation.

Our OPERA framework is compatible with a variety of existing contrastive learning methods. For example, we apply OPERA to MoCo-v3 (Chen et al., 2021b) by instantiating $\mathcal{Y}^{self}$ as the output of the online predictor and the target predictor denoted as $\mathcal{Y}_q^{self}$ and $\mathcal{Y}_k^{self}$, respectively. Additionally, $J(\mathcal{Y}^{self}, \mathcal{L}^{self})$ is the widely-used InfoNCE loss (Van den Oord et al., 2018). Furthermore, we employ an extra MLP block that explicitly connects to the online predictor to obtain $\mathcal{Y}^{full}$ and fix the output dimension to the class number of the pretrained dataset (*e.g.*, 1,000 for ImageNet). We then introduce full supervision on $\mathcal{Y}^{full}$ with the Softmax loss. The overall objective based on MoCo-v3 is as follows:

$$J_m(\mathcal{Y}, \mathcal{L}) = \frac{1}{N}\sum_{i=1}^{N}[-log\frac{exp(\mathbf{y}_{q,i}^{self} \cdot \mathbf{y}_{k,i}^{self}/\tau)}{exp(\mathbf{y}_{q,i} \cdot \mathbf{y}_{k,i}/\tau) + \sum_{j \neq i} exp(\mathbf{y}_{q,i} \cdot \mathbf{y}_{k,j}/\tau)} - log\frac{exp(\mathbf{y}_{i,l_i}^{full})}{\sum_{j \neq l_i} exp(\mathbf{y}_{i,j}^{full})}] \tag{30}$$

where $\mathbf{y}_{i,j}^{full}$ denotes the jth component of $\mathbf{y}_i^{full}$. In addition, we also adopt the stop-gradient operation and the momentum update to the target network following He et al. (2020). Therefore, the proposed OPERA framework preserves the instance-level information in MoCo-v3 to prevent damaging the transferability of the model. Furthermore, OPERA involves class-level knowledge with the class-level full supervision, which further boosts the performance of the learned representations.

E Implementation Details

We provide more implementation details of our experiments on linear evaluation, end-to-end finetuning, semantic segmentation, and object detection.

E.1 Linear Evaluation and End-to-End Finetuning

We evaluated our method on linear evaluation and end-to-end finetuning on the ImageNet (Russakovsky et al., 2015) dataset. For linear evaluation, we used the SGD optimizer and fixed the batch size to 1024. We set the learning rate to 0.1 for R50 (He et al., 2016) and 3.0 for DeiT-S (Touvron et al., 2021). The weight decay was 0 and the momentum of the optimizer was 0.9 for both architectures. Additionally, we conducted end-to-end finetuning with DeiTs and respectively set the batch size to 1024, 2048, and 4096. We used the AdamW (Loshchilov & Hutter, 2018) optimizer with an initial learning rate of 5e-4 and a weight decay of 0.05. We employed the cosine annealing (Loshchilov & Hutter, 2016) learning schedule during training.

E.2 Semantic Segmentation

We transferred the pretrained models to the semantic segmentation task with R50 and DeiTs on the ADE20K (Zhou et al., 2019) dataset. For R50, we used FCN (Shelhamer et al., 2017) as the basic segmentation head. We applied the SGD (Robbins & Monro, 1985) optimizer with a learning rate of 0.01, a momentum of 0.9, and a weight decay of 5e-4. For DeiTs, we adopted the UPerNet (Xiao et al., 2018) as the basic decoder and FCN (Shelhamer et al., 2017) as the auxiliary head. The optimizer, the momentum, and the weight decay are the same as R50. In addition, we trained the models for 160k for both architectures.

E.3 Object Detection

We conducted experiments on object detection and instance segmentation with R50 on the COCO (Lin et al., 2014) dataset. We employed Mask R-CNN (He et al., 2017) with R50-FPN as the backbone. We used the SGD (Robbins & Monro, 1985) optimizer with a learning rate of 0.02, a momentum of 0.9, and a weight decay of 1e-4 for both $1\times$ and $2\times$ schedules.

Table 7: Top-1 and top-5 accuracies (%) under the linear classification protocol on ImageNet.

Method	Batch Size	Pretraining	Finetuning	Backbone	Top-1 Acc	Top-5 Acc
MoCo-v1	256	200	100	R50	60.6	-
MoCo-v2	256	200	100	R50	67.5	-
MoCo-v2	256	800	100	R50	71.1	-
SimCLR	4096	100	1000	R50	69.3	89.0
SimSiam	256	800	100	R50	71.3	-
BYOL	4096	1000	80	R50	74.3	91.6
MoCo-v3†	1024	300	90	R50	70.5	90.0
OPERA	1024	150	90	R50	**73.7**	**91.2**
OPERA	1024	300	90	R50	**74.8**	**91.9**
MoCo-v3†	1024	300	90	DeiT-S	71.2	90.3
OPERA	1024	150	90	DeiT-S	**72.7**	**90.7**
OPERA	1024	300	90	DeiT-S	**73.7**	**91.3**

Table 8: Top-1 and top-5 accuracies (%) under the end-to-end finetuning protocol on ImageNet.

Method	Batch Size	Pretraining	Finetuning	Backbone	Top-1 Acc	Top-5 Acc
Supervised	1024	-	300	DeiT-S	79.8	95.0
Supervised	1024	-	300	DeiT-B	81.8	95.6
DINO†	1024	300	300	DeiT-B	82.8	96.3
MoCo-v3†	1024	300	100	DeiT-S	78.8	94.6
OPERA	1024	100	100	DeiT-S	**78.8**	**94.7**
OPERA	1024	150	100	DeiT-S	**79.1**	**94.7**
OPERA	1024	300	100	DeiT-S	**80.0**	**95.1**
MoCo-v3†	1024	300	150	DeiT-S	79.1	94.6
OPERA	1024	100	150	DeiT-S	**79.8**	**94.9**
OPERA	1024	150	150	DeiT-S	**79.9**	**95.1**
OPERA	1024	300	150	DeiT-S	**80.4**	**95.3**
MoCo-v3†	1024	300	200	DeiT-S	80.0	95.2
OPERA	1024	100	200	DeiT-S	**80.3**	**95.3**
OPERA	1024	300	200	DeiT-S	**80.8**	**95.5**
MoCo-v3†	1024	300	150	DeiT-B	82.1	95.9
OPERA	1024	150	150	DeiT-B	**82.4**	**96.0**
OPERA	1024	300	150	DeiT-B	**82.6**	**96.2**
MoCo-v3†	2048	300	150	DeiT-B	82.7	96.3
OPERA	2048	150	150	DeiT-B	**82.8**	**96.3**
OPERA	2048	300	150	DeiT-B	**83.1**	**96.4**
MoCo-v3†	4096	300	150	DeiT-B	83.0	96.3
OPERA	4096	150	150	DeiT-B	**83.2**	**96.4**
OPERA	4096	300	150	DeiT-B	**83.5**	**96.5**

F More Experimental Results

We present more experimental results of our OPERA framework in this section including comparison experiments with diverse pretraining epochs, as shown in Table 7, Table 8, Table 9, Table 10, Table 11, and Table 12. We observe that OPERA pretrained with fewer epochs (150 or 100) still obtained consistent performance boosts compared with the MoCo-v3 baseline. For example, in Table 8, OPERA based on DeiT-S pretrained for 100 epochs and finetuned on ImageNet for 150 epochs with the batch size of 1024 achieved 79.8% top-1 accuracy, which is 0.7% higher than the baseline. Additionally, for semantic segmentation in Table 10, we can see that OPERA based on R50 pretrained for 150 epochs achieved 37.7 mIoU, which surpassed both the MoCo-v3 baseline and the supervised counterpart. Therefore, the proposed OPERA framework improves the performances on these vision tasks in an efficient training process, which further demonstrates the superiority of our method.

Table 9: Top-1 accuracy (%) of the transfer learning on other classification datasets.

Method	Pretraining	Finetuning	Backbone	CIFAR-10	CIFAR-100	Flowers-102	Pets
Supervised†	300	100	R50	97.6	85.5	95.6	92.2
MoCo-v3†	300	100	R50	97.8	86.0	93.7	90.0
OPERA	150	100	R50	**97.9**	**86.3**	**93.9**	**91.1**
OPERA	300	100	R50	**98.2**	**86.8**	**95.6**	**92.7**
Supervised†	300	100	DeiT-S	98.4	86.9	95.4	93.0
MoCo-v3†	300	100	DeiT-S	97.9	86.6	90.3	90.1
OPERA	150	100	DeiT-S	**98.4**	**88.5**	**94.6**	**91.9**
OPERA	300	100	DeiT-S	**98.6**	**89.0**	**95.5**	**93.3**

Table 10: Experimental results of semantic segmentation on ADE20K (160k schedule).

Method	Pretraining	Backbone	Batch Size	mIoU	mAcc	aAcc
Supervised	300	R50	1024	36.1	45.4	77.5
MoCo-v3†	300	R50	1024	37.0	47.0	77.6
OPERA	100	R50	1024	**37.2**	**47.4**	**77.6**
OPERA	150	R50	1024	**37.7**	**47.9**	**77.7**
OPERA	300	R50	1024	**37.9**	**48.1**	**77.9**
OPERA	150	R50	4096	**38.1**	**47.9**	**78.0**
OPERA	300	R50	4096	**38.4**	**48.5**	**78.1**
Supervised	300	DeiT-S	1024	42.9	53.9	80.3
MoCo-v3†	300	DeiT-S	1024	42.3	53.5	80.6
OPERA	100	DeiT-S	1024	**42.4**	**53.0**	**80.4**
OPERA	150	DeiT-S	1024	**43.4**	**54.2**	**80.8**
OPERA	300	DeiT-S	1024	**43.6**	**54.4**	**80.9**
OPERA	150	DeiT-S	4096	**43.5**	**54.3**	**80.8**
OPERA	300	DeiT-S	4096	**43.8**	**54.6**	**80.9**
Supervised	300	DeiT-B	1024	45.4	56.5	81.4
MoCo-v3†	300	DeiT-B	1024	44.4	55.1	81.5
OPERA	150	DeiT-B	1024	**44.8**	**55.7**	**81.8**
OPERA	300	DeiT-B	1024	**45.2**	**55.9**	**81.9**
MoCo-v3†	300	DeiT-B	2048	45.2	55.5	81.9
OPERA	150	DeiT-B	2048	**45.6**	**56.4**	**82.0**
OPERA	300	DeiT-B	2048	**45.9**	**56.7**	**82.0**
MoCo-v3†	300	DeiT-B	4096	46.1	56.7	82.1
OPERA	150	DeiT-B	4096	**46.4**	**56.9**	**82.1**
OPERA	300	DeiT-B	4096	**46.6**	**57.2**	**82.1**

G GENERALIZING TO MIM METHODS

Masked image modeling (MIM) methods mask part of the input images and extract the representations based on the masked images. These methods then predict the missing portion using the obtained representations. For example, MAE (He et al., 2021) utilizes an autoencoder structure where an encoder extracts the latent representations and a decoder reconstructs the whole image with the representations. The experimental results between contrastive learning methods and MIM-based methods are listed in Table 13. The MIM-based methods include BEiT (Bao et al., 2021), MSN (Assran et al., 2022), MAE (He et al., 2021), iBOT (Zhou et al., 2021), and SimMIM (Xie et al., 2022). We can see that MIM-based methods tend to pretrain the models for more epochs and obtain better performances than contrastive learning approaches. However, our OPERA framework achieves 83.5% top-1 accuracy and is comparable with MIM-based methods (higher than BEiT (Bao et al., 2021) and MSN (Assran et al., 2022)), which demonstrates the effectiveness of the proposed method. In addition, as we have mentioned before, our framework can be extended to masked image modeling (MIM) methods by inserting a new task space in our hierarchy. Specifically, we maintain the image reconstruction supervision in MIM models and transform the representations to a new latent space where we adopt fully supervised learning with the ground truth labels. The practical implementation of OPERA to MIM models serves as our future work.

Table 11: Experimental results of object detection and instance segmentation on the COCO dataset. (Mask R-CNN, R50-FPN, 1 × schedule)

Method	Pretraining	Batch Size	$\mathbf{AP}^{bb}$	$\mathbf{AP}^{bb}_{50}$	$\mathbf{AP}^{bb}_{75}$	$\mathbf{AP}^{mk}$	$\mathbf{AP}^{mk}_{50}$	$\mathbf{AP}^{mk}_{75}$
Rand. Init.	-	1024	31.0	49.5	33.2	28.5	46.8	30.4
Supervised	300	1024	38.2	58.8	41.4	34.7	55.7	37.2
MoCo-v3†	300	1024	38.9	58.8	42.4	35.2	56.0	37.7
OPERA	150	1024	**38.9**	**58.9**	**42.1**	**35.3**	**55.8**	**37.8**
OPERA	300	1024	**39.2**	**59.2**	**42.6**	**35.9**	**56.2**	**38.1**
OPERA	150	4096	**39.1**	**59.1**	**42.7**	**35.6**	**56.2**	**38.0**
OPERA	300	4096	**39.3**	**59.3**	**42.9**	**36.0**	**56.4**	**38.1**

Table 12: Experimental results of object detection and instance segmentation on the COCO dataset (Mask R-CNN, R50-FPN, 2 × schedule).

Method	Pretraining	Backbone	$\mathbf{AP}^{bb}$	$\mathbf{AP}^{bb}_{50}$	$\mathbf{AP}^{bb}_{75}$	$\mathbf{AP}^{mk}$	$\mathbf{AP}^{mk}_{50}$	$\mathbf{AP}^{mk}_{75}$
Rand. Init.	-	1024	36.7	56.7	40.0	33.7	53.8	35.9
Supervised	300	1024	39.2	59.6	42.8	35.4	56.4	37.9
MoCo-v3†	300	1024	40.3	60.0	44.3	36.5	57.4	39.0
OPERA	150	1024	**40.5**	**60.0**	**44.6**	**36.4**	**57.3**	**39.0**
OPERA	300	1024	**41.2**	**60.7**	**45.0**	**36.9**	**57.7**	**39.5**
OPERA	150	4096	**41.2**	**60.9**	**45.1**	**37.0**	**58.0**	**39.6**
OPERA	300	4096	**41.5**	**61.2**	**45.5**	**37.3**	**58.2**	**39.9**

Table 13: Top-1 accuracy (%) under the end-to-end finetuning protocol on ImageNet based on MIM methods.

Method	Type	Pretraining	Backbone	Top-1 Acc
BEiT	Masked Image Modeling	800	ViT-B	83.2
MSN	Masked Image Modeling	600	ViT-B	83.4
MAE	Masked Image Modeling	1600	ViT-B	83.6
iBOT	Masked Image Modeling	1600	ViT-B	83.8
SimMIM	Masked Image Modeling	800	ViT-B	83.8
DINO†	Contrastive Learning	300	ViT-B	82.8
MoCo-v3†	Contrastive Learning	300	ViT-B	83.0
OPERA	Contrastive Learning	300	ViT-B	83.5