Self-Distilled Self-supervised Representation Learning

Jiho Jang[1], Seonhoon Kim[2*], Kiyoon Yoo[1*], Chaerin Kong[1], Jangho Kim[3], Nojun Kwak[1]

[1]Seoul National University, [2]Coupang, [3]Kookmin University

{geographic,961230,rin4616,nojunk}@snu.ac.kr, sekim625@coupang.com, jangho.kim@kookmin.ac.kr

Abstract

State-of-the-art frameworks in self-supervised learning have recently shown that fully utilizing transformer-based models can lead to performance boost compared to conventional CNN models. Striving to maximize the mutual information of two views of an image, existing works apply a contrastive loss to the final representations. Motivated by self-distillation in the supervised regime, we further exploit this by allowing the intermediate representations to learn from the final layer via the contrastive loss. Through self-distillation, the intermediate layers are better suited for instance discrimination, making the performance of an early-exited sub-network not much degraded from that of the full network. This renders the pretext task easier also for the final layer, leading to better representations. Our method, Self-Distilled Self-Supervised Learning (SDSSL), outperforms competitive baselines (SimCLR, BYOL and MoCo v3) using ViT on various tasks and datasets. In the linear evaluation and k-NN protocol, SDSSL not only leads to superior performance in the final layers, but also in most of the lower layers. Furthermore, qualitative and quantitative analyses show how representations are formed more effectively along the transformer layers. Code is available at https://github.com/hagiss/SDSSL.

1. Introduction

GPT [49] and BERT [15] are two representative works in self-supervised learning (SSL) that use transformers [57] for natural language processing (NLP) tasks. Motivated by these successes, various efforts on self-supervised representation learning [41, 26, 3, 29, 18] have been made in the vision domain as well, many of which follow the recent paradigm of instance discrimination that matches the representations of different views of the same image generated by separate augmentations [9, 22, 21, 7, 10]. Recent self-supervised frameworks have focused on using transformer-based models such as ViT [16], which has demonstrated su-

perior performance over the conventional ResNet [23] architectures. MoCo v3 [11] and DINO [8] achieved state-of-the-art performances using ViT in self-supervised learning. MoCo v3 investigated the learning instability of ViT and tackled this to enhance performance, while DINO exploited the characteristics of ViT and proposed a unique MLP head to improve representation learning.

Meanwhile, in the supervised regime, knowledge distillation via self-distillation [61], which encourages the low layer outputs to follow the outputs of the final or higher layers, have shown to be effective with attempts to explain its performance boost by the mechanisms of ensemble [39] and regularization [1]. Despite its effectiveness, self-distillation (SD) has not been utilized in the self-supervised framework with works focusing only on distillation between the *final outputs* of a student and a teacher network (composed of exponential moving averages of students).[1] Motivated by this, we propose Self-Distilled Self-Supervised Learning (SDSSL), a natural application of SD to self-supervised learning. When applied to SSL methods such as SimCLR and MoCo, self-distillation has an intuitive explanation: aligning representations of the lower layers to the final one can enhance linear separability of the lower layer representations as illustrated in Fig. 1. This in turn renders the instance discrimination task easier for the subsequent layers.

Consequently, by solving the instance discrimination pretext task better than its counterpart, we empirically demonstrate SDSSL increases the performance for multiple downstream tasks. In addition, we quantitatively and qualitatively show that SDSSL generates better intermediate representations.

Because our method operates in an orthogonal manner to other SSL methods, we can simply apply our method on top of existing works. In this work, we apply our method to three representative SSL frameworks, namely SimCLR [9], BYOL [21], and MoCo v3 [11], using ViT [16] as the backbone and show that our method improves upon the already competitive baselines. We demonstrate the effectiveness of

---

*Equal Contribution

[1]DINO uses the term "self-distillation" to refer to distillation between a student network and a teacher network.
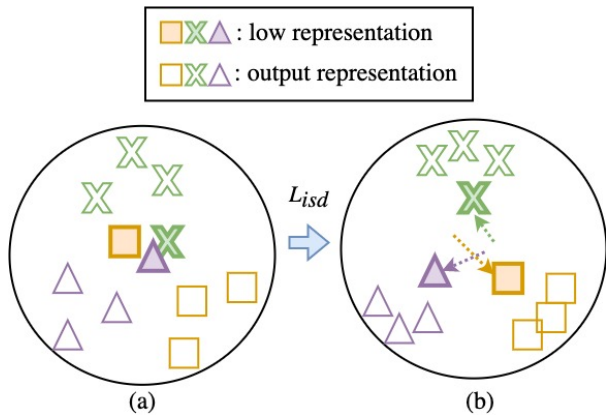
Figure 1: **Representations in hypersphere.** An illustration of representations of the student's low layer and the teacher's output on a hypersphere. Intermediate self-distillation loss explicitly shifts the representations of the low layer to the output representations.

SDSSL on ImageNet via k-nearest neighbor (k-NN) and linear evaluation. The superiority of SDSSL is also shown on various practical tasks such as copy detection, video segmentation and image retrieval. We further investigate the representations learned by SDSSL using recently proposed metrics [58] and discover that SDSSL makes a more linearly separable representation space compared to the baselines. Finally, similar to [44, 61], by encouraging the intermediate layers to explicitly learn the pretext task, we show that even the intermediate features can be successfully used for the downstream tasks, outperforming the baseline counterparts.

Overall, we propose a self-distillation method that lets the intermediate layers *explicitly* learn to discriminate instances. We show that our method improves upon the conventional SSL frameworks such as SimCLR, BYOL, and MoCo v3 on various benchmarks. Through thorough ablation studies, we demonstrate that naively applying our method leads to performance degradation and show how our approach overcomes these potential pitfalls.

2. Related Work

Self supervised learning Despite their impressive progress across multiple domains, deep neural networks (DNN) are extremely data-hungry, requiring a large scale dataset for training. As larger models demand even larger datasets, the annotation cost easily becomes unaffordable. For this reason, many works have explored the field of self-supervised learning, which is a family of unsupervised learning frameworks where the model is guided to learn representations useful for potential downstream tasks from a set of pretext tasks. DIM [26] maximizes the mutual information between input and output. AMDIM [3] makes multiple views of an input and tries to maximize the mutual information between input and output using different views. The key difference

between AMDIM and our SDSSL is that while AMDIM encourages the final representation to mimic intermediate features from a particular layer using convolutional neural networks, SDSSL enforces every intermediate representation to mimic the final feature, which is best suited for the pretext task, *i.e., instance discrimination.* CPC [41] trains the representation in a sequential model by using a contrastive method and shows that InfoNCE loss maximizes the lower bound of mutual information between the inputs and the representations. SimCLR [9] and MoCo [22] have shown impressive performance on various benchmarks by combining strong augmentations with contrastive learning objective, but they are yet limited in that they either require huge batch size or a separate memory bank. BYOL [21] has successfully overcome this limitation by only using positive samples while boosting the performance at the same time. Meanwhile, as transformers [57, 16] gain increasing popularity in the vision domain, self-supervised learning using transformer has been studied as well [11, 8, 19], pushing the previous state-of-the-art further and introducing several beneficial properties absent in traditional CNN models. Despite steady improvements in quantitative evaluations, little has been said about *how and why* these methods work. ReLIC [38] introduces the causal mechanism to explain SSL and [58] introduces alignment and uniformity for quantitative analysis. We provide alignment-uniformity analysis in Secs. 4.4 to shed light on the underlying factors that motivate the success of SDSSL.

Knowledge distillation Knowledge distillation (KD) is one of the regularization methods widely used to improve model performance [25, 31, 24]. The conventional offline KD framework utilizes a pre-trained teacher network to provide additional learning signals to the student network that is primarily trained with labels. In contrast, online KD methods adopt concurrent training scheme, where the teacher and the student are trained simultaneously, distilling information from each other. Recently, several works have explored the concept of self-distillation, where the knowledge from previous snapshot of the model is distilled to the current one [50, 52]. Multi-exit [44], among others, enforces the prediction from lower layers to match that of higher layers, which has similarities with our SDSSL in the high level idea. However, unlike the aforementioned methods, SDSSL is fully self-supervised, not requiring any labels in the training process. We also note that our method differs from others in the formulation of distillation, as ours is driven by instance-discrimination-based SSL objective.

3. Method

3.1. Baselines

SimCLR makes two views, X_1 and X_2, of an input image X (positive sample), by performing separate random aug-

mentations. The representations of X_1 and X_2 from the backbone network are first projected, and the contrastive objective enforces the cosine similarity between positive samples to be maximized while minimizing it between negative samples (other images in the batch) [6].

MoCo v3 learns from the contrastive loss like SimCLR, but instead of using an identical network to generate features for X_1 and X_2, a teacher network with exponential moving average (EMA) parameters is used. Randomly augmented views X_1 and X_2 are forwarded to the student network and the teacher network respectively, and then projected. The projected output of the student network is further processed through an additional MLP head to perform contrastive learning.

BYOL also has an EMA teacher and a predictor like MoCo v3, but learns simply by increasing the cosine similarity of positive samples without using the contrastive loss. Therefore, unlike the aforementioned SSL frameworks that utilize negative samples, the performance is robust to the choice of batch size.

SSL objective functions $\mathcal{L}_{ssl}$ differ for different baselines. Following common practice, let q denote the output of the student's last MLP head (projector or predictor) and z denote the output of the teacher's (student in SimCLR) projector. Then the objective for BYOL is

$$\mathcal{L}_{ssl}(q, z) = 2 - 2 \cdot \frac{\langle q, z \rangle}{\|q\|_2 \cdot \|z\|_2} \qquad (1)$$

while for the two contrasitive methods,

$$\mathcal{L}_{ssl}(q, z) = -\log \frac{e^{\langle q, z^+ \rangle / \tau}}{e^{\langle q, z^+ \rangle / \tau} + \sum_{z^-} e^{\langle q, z^- \rangle / \tau}} \qquad (2)$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product, τ is a temperature parameter and z^+/z^- are for the positive/negative samples. We note that the q and z are L2-normalized in Eq. (2)

3.2. SDSSL

We propose Self-Distilled Self-Supervised Representation Learning (SDSSL) which provides explicit signal to the intermediate representation by inducing intermediate representation to mimic the output representation as illustrated in Fig. 2. Our method can be applied to any existing SSL frameworks that aligns representations from multi-views.

Self-Distilled SSL We define our *intermediate self-distillation* loss $\mathcal{L}_{isd}$, which tries to maximize the mutual information of the output of an intermediate layer l and the final layer L ($I(f_l^1; f_L^2)$), as the following

$$\mathcal{L}_{isd} = \frac{1}{L-1} \sum_{l=1}^{L-1} \mathcal{L}_{ssl}(q_l, \text{sg}(z_L)), \qquad (3)$$
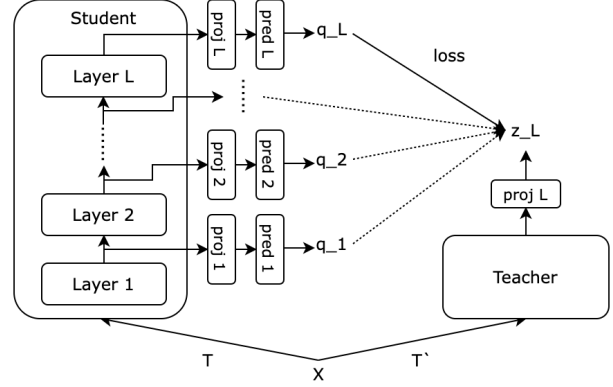

Figure 2: **Illustration of SDSSL** in MoCo v3 and BYOL. For SimCLR, predictors don't exist and teacher network is identical to student network. Solid line is updated by $\mathcal{L}_{ssl}$ while dotted line is for $\mathcal{L}_{isd}$ and $\mathcal{L}_{pred}$.

where q_l is the representation of the lth layer of the student encoder passed through the MLP heads corresponding to each layer, and z_L is the output of the teacher MLP head. The stop-gradient operator, $\text{sg}(z_L)$ implies that the gradient is not propagated through z_L so that only q_l learns to predict z_L without affecting z_L. The objective of SDSSL consists of $\mathcal{L}_{ssl}$ and $\mathcal{L}_{isd}$, resulting in $\mathcal{L}_{sdssl}$:

$$\mathcal{L}_{sdssl} = \mathcal{L}_{ssl}(q_L, z_L) + \alpha \mathcal{L}_{isd}, \qquad (4)$$

where the choice of α, which controls the weight of the self-distillation loss, is detailed in Secs. 4.1.

We observe that for frameworks where predictors exist, simply using Eq. (4) leads to some performance improvement, but can be further enhanced. This is because the predictors of the intermediate layers are only updated using gradients from $\mathcal{L}_{isd}$ as opposed to the encoder, which is able to utilize both $\mathcal{L}_{isd}$ and $\mathcal{L}_{ssl}$. Consequently, the optimality of the predictors at intermediate layers are not guaranteed, which is a key component of SSL training as discussed by [21]. Simply enlarging α causes the last layer of the encoder to be sub-optimal, because this updates the intermediate backbone layers as well. To alleviate this issue, we employ another loss $\mathcal{L}_{pred}$:

$$\mathcal{L}_{pred} = \sum_{l=1}^{L} \mathcal{L}_{ssl}(\text{pred}(\text{sg}(h_l)), \text{sg}(z_L)), \qquad (5)$$

where h_l is the representation of the lth layer of the student after passing through the projector. To only update the predictors, the $\text{sg}(\cdot)$ operator is used to h_l. By doing so, we attain better predictors for distillation and hence the final objective for SSL frameworks with predictors is

$$\tilde{\mathcal{L}}_{sdssl} = \mathcal{L}_{ssl}(q_L, z_L) + \alpha \mathcal{L}_{isd} + \beta \mathcal{L}_{pred}. \qquad (6)$$

We use $\beta = 1$ in Eq. (6). Alg. 1 provides the pseudocode for SD-MoCo v3 which applies SDSSL to MoCo v3.

Algorithm 1 SD-MoCo v3: PyTorch-like Pseudocode

```
# f_s: student: ViT + projectors
# f_t: momentum teacher: ViT + projector
# p: predictors
# alpha: intermediate self-distillation ratio
# tau: temperature
# L: number of layers in ViT

for x in loader: # load a minibatch x with N samples
    x1, x2 = aug(x), aug(x) # random augmentation
    q1, q2 = f_s(x1), f_s(x2) # shape: [L*N, dim]
    z1, z2 = f_t(x1), f_t(x2) # shape: [N, dim]

    loss_pred = ctr(p(q1.detach()), z2, L)
    loss_pred += ctr(p(q2.detach()), z1, L)

    q1, q2 = p(q1), p(q2)

    q1_isd, q1 = split(q1, [(L-1)*N, N])
    q2_isd, q2 = split(q2, [(L-1)*N, N])

    loss_isd = ctr(q1_isd, z2, L-1)
    loss_isd += ctr(q2_isd, z1, L-1)

    loss = ctr(q1, z2) + ctr(q2, z1)
    loss += alpha * loss_isd + L * loss_pred

    loss.backward()
    optimizer.update(f_s, p)
    momentum_update()

# contrastive loss
def ctr(q, z, num_layers=1):
    logits = mm(q, z.t()) # [num_layers*N, N] pairs
    labels = repeat(arange(N), num_layers)
    loss = CrossEntropyLoss(logits/tau, labels)
    return 2 * tau * loss
```

Framework	ViT-S/32		ViT-S/16		ViT-B/16	
	k-NN	Linear	k-NN	Linear	k-NN	Linear
SimCLR	51.5	52.8	57.8	62.1	62.1	70.5
SD-SimCLR	**53.4**	**55.3**	**59.1**	**65.0**	**64.4**	**72.1**
BYOL	56.4	59.8	66.0	70.3	68.1	73.7
SD-BYOL	**57.9**	**61.8**	**67.2**	**71.5**	**70.3**	**74.5**
MoCo v3	57.1	60.7	66.5	70.0	69.7	75.1
SD-MoCo v3	**59.0**	**63.7**	**68.0**	**71.5**	**72.0**	**76.0**

Table 1: **ImageNet Evaluation.** Comparison with three competitive baselines and SDSSL. ViT-S/32, ViT-S/16 and ViT-B/16 are trained on ImageNet for 300 epochs. For each framework, ViT-S/32, ViT-S/16 and ViT-B/16 share the same set of hyper-parameters except batch size.

Batch normalization [27] is applied to all output layers except BYOL and the hidden layers for all methods. The dimension of the hidden layer is 4096 for the last projector and all predictors, but 2048 for intermediate projectors. All outputs have 256 dimension. For frameworks using an exponential moving average (EMA) teacher, the teacher's projector is updated using the student's projector via EMA. This is done in SDSSL as well using only the last projector. In ablation study, we show that the effects of intermediate projector according to the number of layers.

Hyper-parameter We use AdamW [36] as the optimizer and batch size of 4096 for ViT-S/32 and 1024 for ViT-S/16 [54] and ViT-B/16. Learning rate is 1.5e-4 for MoCo v3 and BYOL, 1.3e-4 for SimCLR. We also adopt learning rate warmup for 40 epochs and cosine decay after warmup [20]. Weight decay is 0.1. For α, cosine scheduling [35] is performed from 0~1.0.

4.2. Main Results

ImageNet Pretraining We experiment with ViT-S/32, ViT-S/16 and ViT-B/16 on three self-supervised learning frameworks. In Table 1 we validate the representations found in ImageNet [14] pretrained encoder through using k-NN [60] and linear evaluation. We follow the protocol of MoCo v3 for the linear evaluation and DINO for k-NN. Across all frameworks, models, and evaluations, applying SDSSL increases performance. The baseline accuracies are lower than those reported in MoCo v3 paper [11], because of using 1024 batch size instead of 4096 due to computation constraint. Contrastive frameworks are particularly affected by the batch size. Nevertheless, our method significantly improves upon our reproduced baselines. In ViT-S models, linear evaluation performance improved more than or equal to it did on k-NN, whereas in ViT-B/16 model, k-NN performances improved more. We used 8 and 4 NVIDIA A100 for five days to train our ViT-B/16 and ViT-S/16 models respectively, and 4 NVIDIA A6000 for three days to train ViT-S/32 models.

Multi-exit Since self-distillation enables lower layers to learn from the higher layers, we expect that the lower layers of SDSSL learned more meaningful representations than

4. Experiments

In this section we describe details of our implementation. We follow the convention of MoCo v3 [11] unless otherwise noted. We show that SDSSL outperforms the baselines in various downstream tasks including ImageNet. Moreover, we employ t-SNE [55], Centered Kernel Alignment (CKA) [13, 32] and uniformity-alignment framework [58] to analyze the representations of SDSSL in an attempt to demystify its success. As we later show, the key is learning better representations in the lower layers, pushing the overall performance curve upwards. Thorough ablations show the effectiveness of individual components of SDSSL and the comparison of SDSSL with ResNet backbones reconfirms the effectiveness of our method.

4.1. Implementation Details

ViT Architecture We adopt the sine-cosine variant [57] in 2-D for positional embedding and freeze the random initialized patch projector. We concatenate patch embedding with a learnable [CLS] token and add its positional embedding. The representations are outputs of [CLS] token after passing through each transformer block and the layer normalization layer [2].

MLP Heads Following [9, 21], projectors and predictors are set as 3-layer MLPs and 2-layer MLPs, respectively.
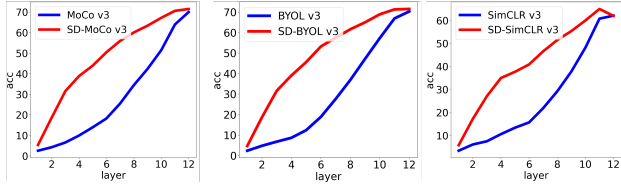
Figure 3: **Multi-exit.** Linear evaluations on Imagenet for baselines and SDSSL on each layer using pretrained ViT-S/16 on ImageNet for 300 epochs. SDSSL methods outperform the corresponding baselines at all layers and shows less degradation for earlier layers.
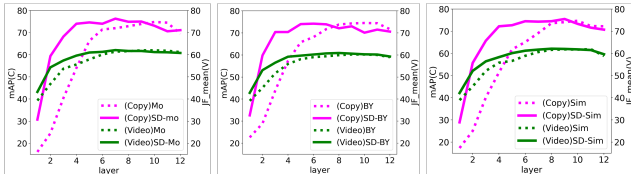

Figure 4: **Copy Detection and Video Segmentation.** (Left) Results of copy detection and video segmentation tasks on MoCo v3 and SD-MoCo v3 for each layer. With the exception of the some layers, SD-MoCo v3 outperforms MoCo v3. The best performing layers are 7th for SD-MoCo v3 and 10th for MoCo v3 in copy detection and video segmentation. The trends are almost the same for BYOL/SD-BYOL (middle) and SimCLR/SD-SimCLR (right).

those of the baselines. This is verified in Figure 3, which shows that the representations of lower layers for SDSSL are much more suitable as features than the counterparts. We performed linear evaluation on ImageNet using the frozen representations of each layer. In the last layer, the accuracy increased by 1.2%p, and the 6th layer showed the largest performance gap of 34.3%p in MoCo v3. Similar phenomena occur in BYOL and SimCLR.

4.3. Transferability

In this subsection, we evaluate the transferability of our method on various downstream tasks. Following DINO [8], we evaluate on the image retrieval task. In addition, we also evaluate on the copy-detection task and the video segmentation task, which uses features of patches rather than the [CLS] token. The three evaluation protocol do not require additional training of the encoder. Then, we evaluate on other image classification datasets such as CIFAR-10, CIFAR-100 [34], Oxford Flowers-102 [40], Oxford-IIT-Pets [42], CUB [59], AirCraft [37], Cars [33], Dogs [30], NABirds [56] and ImageNet by k-NN, linear evaluation and end-to-end fine-tuning [16]. Experiments are performed using all the three frameworks with ViT-S/16.

Image Retrieval Revisited [46] Oxford and Paris image retrieval datasets [43] contain 3 splits of various difficulty with query and database pairs. We evaluate all baselines and SDSSL on the Medium and Hard splits. We directly apply k-NN for image retrieval. As shown in Table 2, SDSSL outperform baselines.

Framework	ROx		RPar	
	M	H	M	H
SimCLR	20.1	3.9	**42.8**	**15.3**
SD-SimCLR	**22.0**	**5.0**	42.3	15.1
BYOL	27.7	6.9	51.7	22.2
SD-BYOL	**28.5**	**7.7**	**52.0**	**22.5**
MoCo v3	26.3	6.4	51.0	21.9
SD-MoCo v3	**26.7**	6.4	**52.4**	**22.7**

Table 2: **Image Retrieval.** Comparison of performance between baseline and SDSSL on image retrieval task. ViT-S/16 are pre-trained using each framework on ImageNet for 300 epochs. We evaluate image retrieval task using k-NN.

Framework	Copy D. mAP	Video S. $(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
SimCLR	74.7	61.8	59.9	63.6
SD-SimCLR	**75.5**	**62.1**	**60.3**	**64.0**
BYOL	**74.5**	60.2	58.1	62.3
SD-BYOL	74.2	**60.9**	**59.0**	**62.7**
MoCo v3	74.8	62.0	60.2	63.8
SD-MoCo v3	**76.3**	**62.1**	**60.4**	**63.9**

Table 3: **Copy detection and video segmentation.** For all scores, higher means better. The reported scores are the performance of the best layer in each method. ImageNet pretrained ViT-S/16 models are used to evaluate.

Copy-detection We report the mean average precision (mAP) of copy-detection on the *strong* subset of INRIA Copydays dataset [17]. The goal of Copy-detection is to recognize the original image when given a distorted (e.g. blur, insertion, print, scan) version of it. Following [4], we use the 10K samples of the YFCC100M dataset [53] as distractors, while 20K samples are used for whitening [4] the features. The features of [CLS] token and patch token are pooled using GeM [47] and concatenated. We use features of all layers to verify whether similar trend occurs as in the multi-exit experiment. We observe in Figure 4 that most SD-MoCo v3 intermediate features surpass those of MoCo v3 and have better performance on the respective best-performing features. For SD-MoCo v3 and MoCo v3, this is the 7th and 10th layer, respectively. We believe that the best-performing features are not formed in the final layer for some tasks that utilizes the features of the patch rather than using only the [CLS] token. Moreover, for SDSSL the best-performing layer is formed in the lower layers than the baseline. For SD-SimCLR and SimCLR, the best-performing layers are 9th and 10th, respectively; for SD-BYOL and BYOL, 6th and 10th layer, respectively. This may be explained by our knowledge distilling method which intends to extract more information in the lower layers. By providing an explicit loss, our method forms a suitable feature for copy detection earlier in the lower layers than the baseline.

Video segmentation We perform video instance segmentation on the DAVIS-2017 dataset [45]. We follow the experimental protocol in Jabri *et al.* [28] and segment scenes with

Framework	CI-10	CI-100	Flower	Pets	CUB	ACraft	Cars	Dogs	NABirds	INet	Avg.
SimCLR-knn	**87.2**	65.7	77.8	71.8	28.5	**22.9**	15.6	50.1	17.3	57.8	49.5
	86.9	**66.0**	**79.1**	**73.9**	29.2	22.8	**15.7**	**52.8**	**18.2**	**59.1**	**50.4**
linear	81.9	58.4	80.4	71.4	39.8	**24.3**	**17.5**	58.9	31.4	62.1	52.6
	82.8	**59.0**	**81.6**	**74.2**	**41.9**	22.6	16.9	**62.2**	**32.5**	**65.0**	**53.9**
fine	97.8	86.1	95.8	88.0	73.6	75.9	85.1	72.5	68.7	78.6	82.2
	98.5	**88.6**	**96.8**	**89.9**	**74.2**	**81.2**	**87.0**	**80.0**	**69.3**	**78.7**	**84.4**
BYOL-knn	90.0	70.6	**85.2**	83.4	52.8	31.1	19.9	67.0	38.2	66.5	60.4
	91.5	**72.5**	85.0	**85.3**	**54.5**	**32.4**	**21.5**	**69.0**	**40.7**	**68.0**	**62.0**
linear	89.9	73.4	**92.7**	87.5	70.7	46.4	42.3	76.8	61.6	70.3	71.2
	92.4	**75.2**	92.6	**88.0**	**73.4**	**46.6**	**43.8**	**78.5**	**63.2**	**71.5**	**72.5**
fine	98.6	89.3	**97.4**	91.1	78.9	80.4	88.7	80.9	74.9	79.3	86.0
	98.9	**89.4**	97.2	**91.4**	**79.9**	80.4	**88.9**	**81.1**	**75.7**	**79.5**	**86.2**
MoCo v3-knn	**91.8**	**73.8**	85.4	83.6	51.2	30.4	21.3	67.7	36.0	66.0	60.7
	91.2	73.4	**85.5**	**84.4**	**53.1**	**32.5**	**22.6**	**69.4**	**38.5**	**67.2**	**61.8**
linear	90.1	73.9	92.6	**87.6**	70.6	**47.3**	41.2	77.7	59.6	70.0	71.1
	90.2	**74.4**	92.6	87.5	**71.7**	47.2	**43.8**	**78.3**	**61.6**	**71.5**	**71.9**
fine	98.7	**89.5**	97.2	90.9	78.5	**81.6**	86.8	78.8	74.1	79.4	85.6
	98.7	89.2	**97.3**	**91.4**	**79.3**	81.3	**87.8**	**79.7**	**75.8**	**79.6**	**86.0**

Table 4: **Classification.** We report k-NN, linear and fine-tuning performances for 10 classification datasets. The upper row is the baseline accuracy, and the lower row is the SDSSL accuracy.

a nearest neighbor between consecutive frames in DINO. When all representations of all layers are tested as done in copy detection, a similar trend is observed in the video segmentation task as well. The best performing layer is 7th and 10th for SD-MoCo v3 and MoCo v3, respectively, and SD-MoCo v3 outperforms MoCo v3 as shown in Table 3. For SD-SimCLR and SimCLR, the best-performing layers are 8th and 9th, respectively; for SD-BYOL and BYOL, 8th and 10th layer, respectively.

Classification In this section, we demonstrate the results for image classification on CIFAR-10, CIFAR-100, Oxford Flowers-102, Oxford-IIT-Pets, CUB, AirCraft, Cars, Dogs, NABirds and ImageNet. Since end-to-end fine-tuning could lead to over-fitting on the specific dataset, this may obscure the evaluation of representation quality of the pre-trained encoder [48]. To address this issue, we also report scores from k-NN and linear evaluation. In Tab. 4, we witness improvements from SDSSL on most of the datasets for all three baselines. Performance gains in k-NN and linear evaluation are especially noticeable, which implies that representations from SDSSL are more separable in the feature space. Further analysis on the representations will be presented in Secs. 4.4. Results from end-to-end fine-tuning displays the typical *diminishing gains* as the overall baseline performance has been greatly boosted.

4.4. Analysis

Qualitative analysis Fig. 5 displays t-SNE visualization of representations from 10 randomly selected classes of ImageNet validation set, where SDSSL on MoCo v3 *(right)* shows clearly more separable representations compared to the MoCo v3 baseline *(left)*. As one of the basic assumptions of contrastive learning framework is to obtain representations useful for downstream tasks through instance discrimination, this implies that self-distillation encourages the model to form better representations from lower layers, making the overall self-supervised learning task easier. Results in Fig. 3 and Fig. 4 support this claim.

CKA [13, 32] in Fig. 6 illustrates the representation similarities between layers. For the MoCo v3 baseline, representations from neighboring layers display high similarity *(black box, left)* while the scores are largely reduced for distant layers *(red box, left)*. SDSSL, on the other hand, shows more uniform similarity structure *(right)* with lower local similarity and higher global similarity. As previous work [51] points at the uniform representation similarities across layers as a key property that distinguishes vision transformers from convolutional networks, we conjecture this can be one of the drivers for the success of SDSSL reinforcing the characteristics of ViTs.

Quantative analysis Wang *et al.* [58] demonstrated that contrastive learning optimizes two distinct metrics: (1) *alignment*, which quantifies compactness of representations of *positive* samples

$$\mathcal{L}_{ali}(f;\gamma) \triangleq \mathop{\mathbb{E}}_{(x,y)\sim p_{pos}} [\|f(x) - f(y)\|_2^\gamma], \ \gamma > 0, \quad (7)$$

for some γ, and (2) *uniformity*, which measures how dispersed the entire representations are in a hypersphere using the Gaussian potential kernel [12, 5]

$$\mathcal{L}_{uni}(f;t) \triangleq \log \mathop{\mathbb{E}}_{(x,y)\sim p_{data}} [e^{-t\|f(x)-f(y)\|_2^2}], \ t > 0. \quad (8)$$

Here, p_{pos} is the distribution of positive pairs generated by random augmentation from the input data, and p_{data} is the overall data distribution. They asserted that low alignment signifies that the positive samples are close to each other while low uniformity signifies that the negative sam-
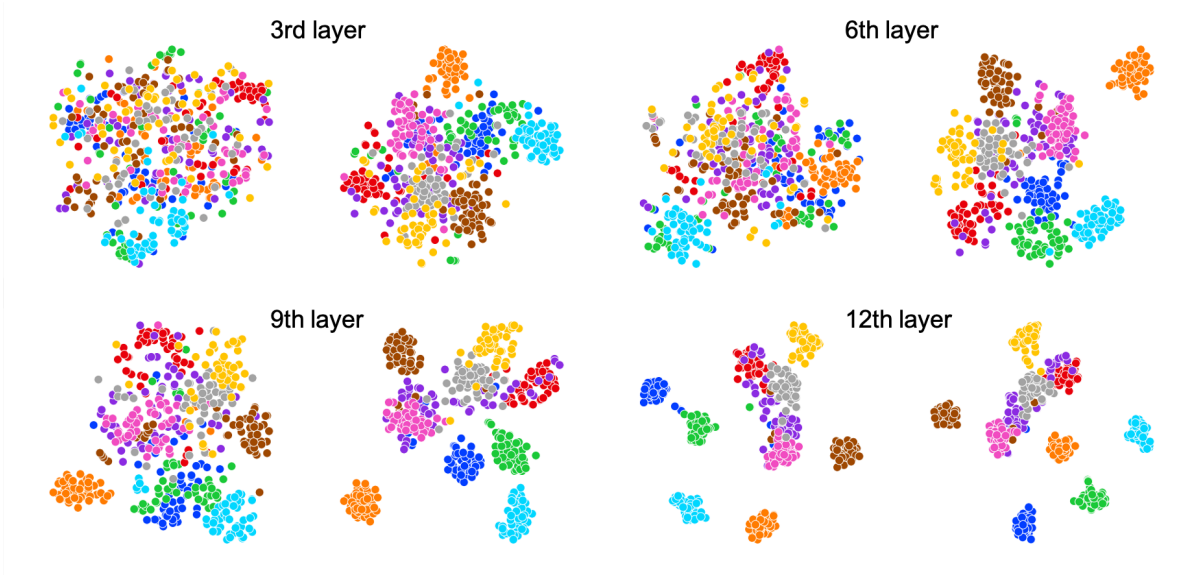
Figure 5: **Representation Visualization.** The representations of each layer of MoCo v3 and SD-MoCo v3 are visualized using t-SNE. The left side of each layer is MoCo v3, and the right side is SD-MoCo v3. 10 random classes of ImageNet validation set are drawn. We observe that the representations of the lower layers are aggregated better by class when SDSSL is applied.
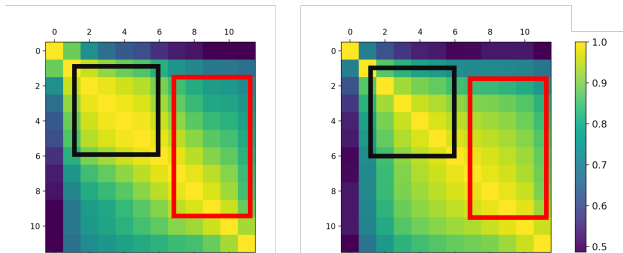

Figure 6: **Representation Similarity.** We compute CKA heatmap for all pairs of layers for MoCo v3 *(left)* and SD-MoCo v3 *(right)*. Black box demonstrates SDSSL's reduced similarity between neighboring layers and red box shows that lower layers faithfully mirrors upper layers.
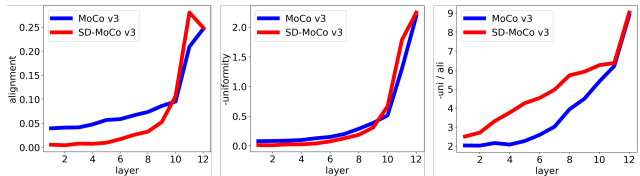

Figure 7: **Alignment and Uniformity** measured at each layer of MoCo v3 and SD-MoCo v3 on ImageNet validation set. Because uniformity and alignment have different signs due to the logarithm of uniformity, we report $-\mathcal{L}_{uni}$ for consistency. Additionally, we compute $-\mathcal{L}_{uni} / \mathcal{L}_{ali}$ to estimate efficiency of representation space.
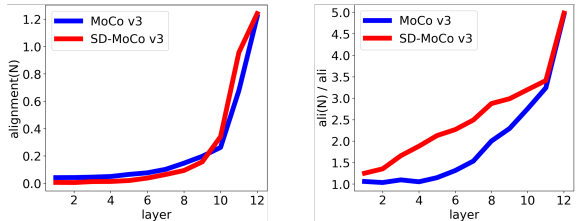

Figure 8: **Negative Alignment.** We plot negative alignment and alignment ratio R using Imagenet pretrained MoCo v3 and SD-MoCo v3. Negative alignment and alignment ratio show similar pattern with uniformity and uniformity alignment ratio, respectively.

ples are further apart. Thus, low alignment and low uniformity lead to a better representation with high linearly separability, although the two metrics are inherently in a trade-off relationship.

Empirically, we observed that SD-MoCo v3 has lower alignment in lower layer, but higher uniformity than vanilla MoCo v3. In the higher layer, the pattern is reversed. In other words, while distances of the positive and negative samples are both close in lower layer, in the higher layer, both are further away. Considering their conflicting characteristics, it is difficult to ascertain which representation is better. To answer this question, we propose a new metric, uniformity-alignment ratio, that divides the uniformity by alignment. For formulational simplicity, we compute negative alignment, *i.e.,* the distance between negative samples, as following and divide it by the original alignment, *i.e.,* the distance between positive samples. This successfully removes potential scale ambiguities and indicates the relative distance between negative samples compared to positive samples in an intuitive manner.

$$\mathcal{L}_{ali}^{n}(f; \gamma) \triangleq \mathop{\mathbb{E}}_{(x,y) \sim p_{data}} \left[\|f(x) - f(y)\|_2^{\gamma} \right], \ \gamma > 0. \quad (9)$$

Higher $\mathcal{L}_{ali}^{n}$ means that the negative samples are further apart from each other similar to uniformity.

The ratio of negative alignment divided by positive alignment $\mathcal{R} \triangleq \mathcal{L}_{ali}^{n}/\mathcal{L}_{ali}$ then quantifies how far apart the mean distance between positive and negative samples are. As shown in Figure 8, SD-MoCo v3 has higher alignment ratio

than MoCo v3 in all layers like uniformity alignment ratio shown in the third column of Fig. 7. Intuitively, learning a representation space where negative samples are placed further apart from each other compared to positive samples is one of the key desiderata of contrastive learning, and we believe this explains the effectiveness of SDSSL at least partially. Qualitative analyses such as Fig. 5 also support this view, delivering a consistent message as to how self-distillation helps self-supervised learning.

4.5. Ablation Study

In this subsection, we show the efficacy of ratio scheduling and the predictor loss through ablation and verify these are necessary factors for optimal performance. Furthermore, we experiment how the performance of SDSSL changes according to the values of α and β.

Tab. 5 shows that the performance of ablating the predictor loss results in a performance degradation. As discussed, this is consistent with the results in [21], which states that optimality of the predictor is crucial. Additionally, when only the predictor loss is used without the intermediate distillation loss $\mathcal{L}_{pred}$, the performance gain with respect to MoCo v3 is minimal (+0.3%p). This verifies that the intermediate distillation loss is a key component. During training, we used ratio annealing in Eq. (4) and Eq. (6), *i.e.*, α is set very low at initial iterations and gradually increased afterward rather than using a fixed α for the entire training. Without ratio annealing the performance decreases significantly, which shows that self-distilling once some training has been done is important.

Fig. 9 shows the performance change across ranges of α and β. For both parameters, the performances generally increase until reaching 1. As discussed, α, which controls L_{isd}, has a larger impact on the performance than β. In Tab. 6 we discuss the effects of some of the design choices of the projector and $\mathcal{L}_{isd}$. Distilling the [CLS] token directly without using a projector (w/o MLP) does not lead to any performance gain, while using a smaller MLP (2-layer) leads to a smaller performance gain. In addition, when $\mathcal{L}_{isd}$ is applied every n layer (MLP/n) the performance gain is smaller than applying it every layer as done in SDSSL (MLP/1).

4.6. SDSSL on ResNet

We further apply SD-Moco v3 framework on top of ResNet-50 backbone and present evaluations for 200 epochs of training. For our proposed self-distillation, we apply the average pooling to the output activation of each residual block and forward it through an MLP head to compute $\mathcal{L}_{sdssl}$. Visible from Tab. 7, we observe significant performance gains on ResNet backbone as well, which is consistent with findings of prior works that incorporate self-distillation in supervised learning [61, 44]. Also, we note that for simplicity, we optimize the self-distillation objec-
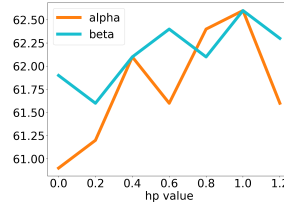

Figure 9: **Hyperparameter sweeping.** We vary α and β to see their effect on the ImageNet k-NN accuracy on MoCo v3. When sweeping for a parameter, the other parameter is fixed to 1.0.

	k-NN
MoCo v3	60.5
+pred. loss	60.8 (+0.3)
SD-MoCo v3	62.6
-ratio anneal	60.2 (-2.4)
-pred. loss	61.9 (-0.7)

Table 5: **Ablation.** This shows that ratio scheduling is indispensable for SDSSL and pred. loss is also beneficial to the final performance.

	k-NN
MoCo v3	60.5
W/o MLP	60.3 (-2.3)
2L-MLP	61.3 (-1.3)
MLP/4	61.3 (-1.3)
MLP/3	61.5 (-1.1)
MLP/2	61.5 (-1.1)
MLP/1	62.6

Table 6: **Ablation on design choices.** We explore the effect of MLP design.

	k-NN	Linear			
		1st	2nd	3rd	last
MoCo v3	60.5	10.5	23.7	45.3	60.3
SD-MoCo v3	61.2	24.7	43.5	59.2	62.2

Table 7: **SDSSL on ResNet.** We train ResNet-50 on ImageNet for 200 epochs using MoCo v3.

tive only for the output activations of each residual block, resulting in fewer distillation terms per batch. We expect further performance gains from applying self-distillation to more intermediate activations as in Tab. 6, but leave it for future work.

5. Discussion

As SDSSL computes the self-distillation loss through additional MLP heads, there are increase in the memory footprint and computational cost for model training. Upon naive implementations with ViT-B/16, the memory footprint rises by less than 6% and training time by about 16.9%. However with parallel computing, the increase in wall clock time can be reduced to 8.6%, which we believe to be moderate. As this additional cost is independent of the model size, the relative burden becomes smaller as we use larger backbone models with smaller patch size, which is often the case in recent practices.

6. Conclusion

In this work, we proposed a self-distillation method generally applicable to existing self-supervised learning frameworks. Our method is motivated by the hypothesis that self-distillation between lower representations and output representation may be favorable for separable representation learning, and through experiments, we empirically validated the effectiveness of our method. We showed SDSSL leads to superior performance not only in the final layers, but also in lower layers through the multi-exit experiment.

References

[1] Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

[4] Maxim Berman, Hervé Jégou, Andrea Vedaldi, Iasonas Kokkinos, and Matthijs Douze. Multigrain: a unified image embedding for classes and instances. *arXiv preprint arXiv:1902.05509*, 2019.

[5] Sergiy V Borodachov, Douglas P Hardin, and Edward B Saff. *Discrete energy on rectifiable sets*. Springer, 2019.

[6] John S Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In *Advances in neural information processing systems*, pages 211–217, 1990.

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[11] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised visual transformers. *arXiv e-prints*, pages arXiv–2104, 2021.

[12] Henry Cohn and Abhinav Kumar. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*, 20(1):99–148, 2007.

[13] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13:795–828, 2012.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[16] 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. *arXiv preprint arXiv:2010.11929*, 2020.

[17] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, pages 1–8, 2009.

[18] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021.

[19] Chongjian Ge, Youwei Liang, Yibing Song, Jianbo Jiao, Jue Wang, and Ping Luo. Revitalizing cnn attention via transformers in self-supervised visual representation learning. *Advances in Neural Information Processing Systems*, 34:4193–4206, 2021.

[20] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[21] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun, and Youliang Yan. Knowledge adaptation for efficient semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 578–587, 2019.

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[26] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[27] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[28] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*, 2020.

[29] Aakash Kaku, Sahana Upadhya, and Narges Razavian. Intermediate layers matter in momentum contrastive self supervised learning. *Advances in Neural Information Processing Systems*, 34:24063–24074, 2021.

[30] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[31] Jangho Kim, Seonguk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[32] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[33] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

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

[35] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[37] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

[38] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms, 2020.

[39] Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361, 2020.

[40] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[41] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[42] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[43] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[44] Mary Phuong and Christoph H Lampert. Distillation-based training for multi-exit architectures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1355–1364, 2019.

[45] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[46] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5706–5715, 2018.

[47] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1655–1668, 2018.

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[49] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[50] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4119–4128, 2018.

[51] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34, 2021.

[52] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[53] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

[54] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[55] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[56] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 595–604, 2015.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko-reit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[58] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[59] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[60] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[61] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 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*, pages 3713–3722, 2019.