Unsupervised Visual Representation Learning via Mutual Information Regularized Assignment

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper proposes Mutual Information Regularized Assignment (MIRA), a pseudo-labeling algorithm for unsupervised representation learning inspired by information maximization. We formulate online pseudo-labeling as an optimization problem to find pseudo-labels that maximize the mutual information between the label and data while being close to a given model probability. We derive a fixed-point iteration method and prove its convergence to the optimal solution. In contrast to baselines, MIRA combined with pseudo-label prediction enables a simple yet effective clustering-based representation learning without incorporating extra training techniques or artificial constraints such as sampling strategy, equipartition constraints, etc. With relatively small training epochs, representation learned by MIRA achieves state-of-the-art performance on various downstream tasks, including the linear/*k*-NN evaluation and transfer learning. Especially, with only 400 epochs, our method applied to ImageNet dataset with ResNet-50 architecture achieves 75.5% linear evaluation accuracy.

1 Introduction

There has been a growing interest in using a large-scale dataset to build powerful machine learning models [42]. Self-supervised learning (SSL), which aims to learn a useful representation without labels, is suitable for this trend; is actively studied in the fields of natural language processing [19, 20] and computer vision [10, 29]. In the vision domain, recent SSL methods are commonly designed to use augmented views and train visual representation to be augmentation-invariant. They have achieved state-of-the-art performance surpassing supervised representation in a variety of visual tasks, including semi-supervised learning [8, 49], transfer learning [21], and object detection [13].

Meanwhile, a line of works use clustering for un-/self-supervised representation learning. They explicitly assign pseudo-labels to embedded representation via clustering, and the model is thereby trained to predict such labels. These clustering-based methods can account for inter-data similarity; representations are encouraged to encode the semantic structure of data. Prior works [47, 45, 4, 31] have shown encouraging results in small-scaled settings; Caron et al. [6] show that it can be also applied to the large-scaled dataset or even to a non-curated dataset [7]. Recently, several works [2, 8, 37] have adapted the philosophy of augmentation invariance and achieved strong empirical results. They typically assign pseudo-labels using augmented views while predicting the labels looking at other differently augmented views.

Despite its conceptual simplicity, a naive application of clustering to representation learning is hard to achieve, especially in large-scale dataset. This is because clustering-based methods are prone to collapse, i.e., all samples are assigned to a single cluster. To address this, recent methods heavily rely on extra training techniques or artificial constraints, such as pre-training [46], sampling strategy [6],

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

equipartition constraints [2, 8], etc. However, it is unclear if these additions are appropriate or how such components will affect the representation quality.

In this paper, we propose Mutual Information Regularized Assignment (MIRA), a pseudo-labeling algorithm that enables clustering-based SSL without any artificial constraints or extra training techniques. MIRA is designed to follow the infomax principle [38] and the intuition that *good labels are something that can reduce most of the uncertainty about the data*. Our method assigns a pseudo-label in a principled way by constructing an optimization problem. For a given training model that predicts pseudo-labels, the optimization problem seeks a solution that maximizes the mutual information (MI) between the pseudo-labels and data while taking the model probability into account. We explicitly derive a solution to this optimization problem via a fixed point iteration and prove its convergence. We remark that MIRA does not require any form of extra training techniques or artificial constraints, e.g., equipartition constraints.

We apply MIRA to clustering-based representation learning and verify the representation quality on several standard self-supervised learning benchmarks. We demonstrate its state-of-the-art performance on linear/k-NN evaluation, semi-supervised learning, and transfer learning benchmark. We further experiment with convergence speed, scalability, and different components of our method.

Our contributions are summarized as follows:

- We propose MIRA, a simple and principled pseudo-label assignment strategy based on mutual information. Our method does not require extra training techniques or artificial constraints.
- We apply MIRA to clustering-based representation learning and it shows comparable performance against the state-of-the-art methods with half of the training epochs. Especially it achieves 75.5% top-1 accuracy on ImageNet linear evaluation with only 400 epochs of training and best performance in 9 out of 11 datasets in transfer learning.
- Representation by MIRA also consistently improves over other information-based SSL methods [22, 49]. Especially, our method without a multi-crop augmentation strategy achieves 73.8% top-1 accuracy and outperforms BarlowTwins [49], an information maximization-based self-supervised method.

2 Related works

Self-supervised learning SSL methods are designed to learn the representation by solving pretext tasks. Recent state-of-the-art SSL methods train their representation to be augmentation invariant. They are based on various pretext tasks: instance discrimination [10, 11, 13, 14], metric learning [27, 12], self-training [50, 9], and clustering [2, 6, 8]; our method belongs to the clustering-based SSL method. Meanwhile, these methods are prone to collapsing into a trivial solution where every representation is map into a constant vector. To address this, a variety of schemes and mechanisms are suggested, e.g., the asymmetric structure, redundancy reduction, etc. We will review more relevant works in detail below.

Collapse preventing Many SSL approaches rely on extra training techniques and artificial assumptions to prevent collapsing. In clustering-based methods, DeepCluster [6] adapts a sample strategy to sample elements uniformly across pseudo-labels to deal with empty clusters; SeLa [2] and SwAV [8] impose equipartition constraints to balance the cluster distribution. Similarly, SelfClassifier [1] uses a uniform pseudo-label prior, and PCL [37] employs concentration scaling. DINO [9] and ReSSL [50] address collapsing by specific combinations of implementation details, i.e., centering and scaling with an exponential moving average network; while their mechanism for preventing collapse is unclear. In this work, we show our method can naturally avoid collapsing without any of these assumptions or training techniques. We achieve results better than baselines with a simple but novel information regularization algorithm. We take a more detailed comparison with SeLa and SwAV in Sec. 3.3.

Information maximization Information maximization is a principal approach to learn representation and to avoid collapse. DeepInfoMax [30] propose the MI maximization between the local and global views for representation learning; the existence of negative pairs prevents training toward the trivial solution. BarlowTwins [49] and W-MSE [22] addresses collapsing with redundancy reduction, indirectly maximizing the information content of the embedding vectors [3]. Among clustering-based

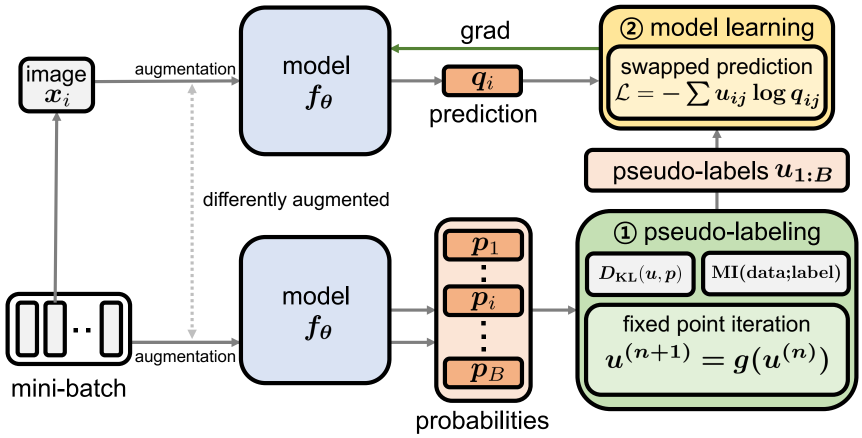

Figure 1: **Overview of representation learning via MIRA.** In our representation learning, MIRA provides pseudo-labels with model probabilities, and the model is learned by predicting the pseudo-labels. Our main contribution is in the ① **pseudo-labeling** process that accounts for mutual information between the pseudo-label and data. In MIRA, optimal pseudo-labels are computed through the fixed-point iteration (Eq. 5). Given such pseudo-labels, ② **model updates** its parameters by gradient update on swapped prediction loss.

approaches, IIC [33] maximizes the MI between the embedding codes for representation learning; most similarly to ours, TWIST [25] proposes to combine the mutual information between the data and class prediction as a negative loss term with a consistency loss. Both IIC and TWIST use the MI as a loss function and directly optimize their model parameters with gradient descent of the loss. However, direct optimization of MI terms by updating model parameters often leads to a sub-optimal solution [25]; TWIST copes with this issue by appending the normalization layer before softmax and introducing an additional self-labeling stage. In contrast, MIRA addresses the difficulty of MI maximization in a principled way via explicit optimization.

3 Method

In this section, we explain our pseudo-labeling algorithm–MIRA. When applying MIRA to representation learning, we follow the basic framework of clustering-based representation learning that alternates between *pseudo-labeling*, i.e., cluster assignments, and *model training* to predict such labels. Figure 1 illustrates our representation training cycle. We will first explain our main contribution, MIRA (*pseudo-labeling*) and then explain how it applies to *model training*.

Our idea is to employ the information maximization principle into pseudo-labeling. We formulate an optimization problem for online clustering that assigns soft pseudo-labels to mini-batch samples (Sec. 3.1). The problem accounts for the model probability and mutual information between the pseudo-labels and data. We propose an iterative method to solve the optimization problem (Sec. 3.2). For the model training, we use the swapped prediction loss as in [8] (Sec. 3.3).

3.1 MI regularized cluster assignment

We have a model[1] f_θ parametrized by θ that outputs K-dimensional logit $f_\theta(\boldsymbol{x}) \in \mathbb{R}^K$ for an image $\boldsymbol{x}$, where K is a predefined number of clusters. The model probability $\boldsymbol{p}$ of an image $\boldsymbol{x}$ is then given by the temperature τ_t scaled output of the model—$\boldsymbol{p} \coloneqq \mathrm{softmax}(f_\theta(\boldsymbol{x})/\tau_t)$—as in [8, 9]. For a mini-batch of input images $\boldsymbol{X} = \{\boldsymbol{x}\}_{i=1}^B$, we denote the model probability $\boldsymbol{P} = \{\boldsymbol{p}_i\}_{i=1}^B \subset \mathbb{R}^K$. In our pseudo-labeling, for the given model probability $\boldsymbol{P}$, we want to assign pseudo-labels $\boldsymbol{W}^* = \{\boldsymbol{w}^*\}_{i=1}^B$ that will be used for training the model by predicting them.

We argue that such pseudo-labels should maximize the mutual information between themselves and data while accounting for the model probability $\boldsymbol{P}$. Let $\mathcal{B} \in \{1, ..., B\}$ and $\mathcal{Y}_{\boldsymbol{W}} \in \{1, ..., K\}$ be the random variables associated with the data index in mini-batch and label by probability distributions

[1]In our setting, the model consists of an encoder, projection head, and classification (prototype) head as in [8, 9]; the encoder output will be used as a representation.

$\boldsymbol{W} = \{\boldsymbol{w}\}_{i=1}^{B}$, respectively. Our online pseudo-label (cluster) assignment is determined by solving the following optimization problem:

$$\boldsymbol{W}^{*} = \underset{\boldsymbol{W} \subset \Delta_K}{\arg\min} \frac{1}{B} \sum_{i=1}^{B} D_{\mathrm{KL}}(\boldsymbol{w}_i, \boldsymbol{p}_i) - \beta \hat{I}(\mathcal{Y}_{\boldsymbol{W}}; \mathcal{B}), \tag{1}$$

where $\Delta_K \coloneqq \{x \in \mathbb{R}_{+}^{K} \mid x^{\intercal} \mathbf{1}_K = 1\}$, $\hat{I}$ indicates an empirical (Monte Carlo) estimates of MI, and β is a trade-off parameter. The problem consists of the (1) KL divergence term that makes pseudo-labels to be based on the model probability $\boldsymbol{p}$ and (2) MI term between the pseudo-labels and data to induce more information into the pseudo-labels. By combining these two terms, we provide a refined pseudo-label that take account of both the model probability and MI.

To make the optimization problem tractable, we substitute the MI term $\hat{I}$ with the mini-batch estimates of the entropy $\hat{H}(\mathcal{Y}_{\boldsymbol{W}}|\mathcal{B})$ and marginal entropy $\hat{H}(\mathcal{Y}_{\boldsymbol{W}})$ in Eq. 2. We get:

$$\hat{I}(\mathcal{Y}_{\boldsymbol{W}}; \mathcal{B}) = \hat{H}(\mathcal{Y}_{\boldsymbol{W}}) - \hat{H}(\mathcal{Y}_{\boldsymbol{W}}|\mathcal{B}) = -\sum_{j=1}^{K} \bar{w}_j \log \bar{w}_j + \frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{K} w_{ij} \log w_{ij}, \tag{2}$$

$$\boldsymbol{W}^{*} = \underset{\boldsymbol{W} \subset \Delta_K}{\arg\min} -\frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{K} w_{ij} \log p_{ij} + \frac{1-\beta}{B} \sum_{i=1}^{B} \sum_{j=1}^{K} w_{ij} \log w_{ij} + \beta \sum_{j=1}^{k} \overline{w}_j \log \overline{w}_j, \tag{3}$$

where $\overline{w}_j = \frac{1}{B} \sum_{i=1}^{B} w_{ij}$ is the marginal probability of a cluster j with $\boldsymbol{W}$. In practice, we find the optimal point $\boldsymbol{W}^{*}$ of the optimization problem Eq. 3.

3.2 Solving strategy

To solve efficiently, we propose a fixed-point iteration that guarantees convergence to the unique optimal solution $\boldsymbol{W}^{*}$ of our optimization problem. The method is based on the following proposition.

Proposition 1. *For $\beta \in [0, 1)$, the problem Eq. 3 is a strongly convex optimization problem; has a unique optimal point $\boldsymbol{W}^{*}$ that satisfies the following necessary and sufficient condition.*

$$\forall (i, j) \in \{1, ..., B\} \times \{1, ..., K\}, \quad {w^{*}}_{ij} = \frac{\overline{w^{*}}_j^{-\frac{\beta}{1-\beta}} p_{ij}^{\frac{1}{1-\beta}}}{\sum_{k=1}^{K} \overline{w^{*}}_k^{-\frac{\beta}{1-\beta}} p_{ik}^{\frac{1}{1-\beta}}}. \tag{4}$$

Based on the necessary and sufficient condition by proposition 1, we propose the following update rule for $\{u_j^{(n)}\}_{j=1}^{K} \subset \mathbb{R}_{+}$:

$$\forall j \in \{1, ..., K\}, \quad u_j^{(n+1)} = \left[\frac{1}{B} \sum_{i=1}^{B} \frac{p_{ij}^{\frac{1}{1-\beta}}}{\sum_{k=1}^{K} (u_k^{(n)})^{-\frac{\beta}{1-\beta}} p_{ik}^{\frac{1}{1-\beta}}} \right]^{1-\beta}, \tag{5}$$

where $u_j^{(n)}$ converges to $\overline{w^{*}}_j$ as $n \to \infty$. We can easily get ${w^{*}}_{ij}$ by Eq. 4 when the marginal probability $\overline{w^{*}}_j$ is given. The proof of the proposition and convergence is in the Appendix.

By using the iterative updates of Eq. 5, we get our desirable pseudo-labels. This requires a few lines of code that are simple to implement. We find that a few steps of iterations are enough for training. This is supported by the convergence analysis in Sec. 4.3. We use this fixed point iteration for pseudo-labeling and name the method–**M**utual **I**nformation **R**egularized **A**ssignment (**MIRA**) since it finds the pseudo-labels that are regularized by the mutual information.

3.3 Representation learning with MIRA

We explain how our pseudo-labeling algorithm is applied to representation learning. We integrate the computed pseudo-labels with swapped prediction loss [8]. Specifically, given the two mini-batches of differently augmented views $\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}$, MIRA outputs the pseudo-labels $\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}$ for each mini-batch independently. In parallel, model f_θ provides the temperature τ_s scaled softmax predictions $\boldsymbol{Q}^{(1)}, \boldsymbol{Q}^{(2)}$ of each mini-batch. The swapped prediction loss is given as follows:

$$
\begin{aligned}
L(\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}) &= \ell(\boldsymbol{U}^{(1)}, \boldsymbol{Q}^{(2)}) + \ell(\boldsymbol{U}^{(2)}, \boldsymbol{Q}^{(1)}) \\
&= -\frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{K} u_{ij}^{(1)} \log q_{ij}^{(2)} - \frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{K} u_{ij}^{(2)} \log q_{ij}^{(1)}.
\end{aligned} \tag{6}
$$

This loss function (Eq. 6) is minimized with respect to the parameters θ of the model f_θ used to produce the predictions $\boldsymbol{Q}^{(1)}, \boldsymbol{Q}^{(2)}$. For more detailed information about swapped prediction loss, please refer to [8].

In this paper, we verify our pseudo-labeling algorithm MIRA for a representation learning purpose with Eq. 6. For convenience, in the rest of this paper, we call the *representation learning with MIRA* also as MIRA. We note that MIRA can integrate recently suggested SSL components such as exponential moving average (EMA) or multi-crop augmentation strategy following the baselines [14, 8, 9]. The pseudo-code for MIRA is provided in the Appendix. We discuss some further details as follows:

Preventing collapse The MI term in Eq. 3 takes a minimum value when collapsing happens. MIRA naturally avoids collapsed solution via penalizing assignment that exhibits low MI. To be more specific, unless starting from the collapsed state, MIRA finds MI-maximizing points around the model prediction; will not choose collapsed pseudo-labels. Hence, the iterative training to predict such labels will not lead to collapsing whenever the prediction of pseudo-labels is achievable. Our empirical results verify that MIRA doesn't require extra training techniques or artificial constraints to address collapsing.

Comparison to SwAV and SeLa SeLa [2] and SwAV [8] assume the equipartition of data into clusters. They formulate their pseudo-labeling process into optimal transport (OT) problem; solving it with the iterative Sinkhorn-Knopp (SK) algorithm [16]. Mathematically, the difference to MIRA is in *how to deal with the marginal entropy*. SeLa and SwAV constrain the marginal entropy to maximum value–equipartition while MIRA decides marginal entropy by MI regularization[2]. Asano et al. [2] argue that their pseudo-labels with OT problem maximize the information between labels and data indices under the equipartition constraints. However, it more resembles assuming MI maximization and finding the assignments that are OT to the model probability. In contrast, MIRA directly maximizes the MI without artificial constraints. While SwAV performs better than SeLa in most self-supervised benchmarks, we empirically verify that MIRA improves over SwAV in various downstream tasks.

4 Experiments

In this section, we evaluate the representation quality learned via MIRA. We first provide the implementation details of our representation learning with MIRA (Sec. 4.1). We present our main results on linear, *k*-NN, semi-supervised learning, and transfer learning benchmarks in comparison to other self-supervised baselines (Sec. 4.2). Finally, we conduct an analysis of MIRA (Sec. 4.3).

4.1 Implementation details

We mostly follow the implementation details from our baselines [8, 9, 49]. More training details about evaluation procedures and analysis are described in the Appendix.

Architecture The training model (network) consists of an encoder, projection head, and classification head. We use a widely used ResNet50 [28] as our base encoder and use the output of average-pooled 2048d embedding as our representation for both representation training and downstream evaluations. The projection head is a 3-layer fully connected MLP of sizes $[2048, 2048, d]$; hidden layers are followed by batch normalization [32] and ReLU. The classification head is used to predict the pseudo-labels; is composed of an L2-normalization layer and a weight-normalized layer of the size $d \times K$ as in [8, 9]. We use $d = 256$ and $K = 3000$.

[2] Adding the equipartition constraints, our optimization problem converts to the OT problem of SwAV.

Table 1: **Linear evaluation with respect to training epochs.** All models use a ResNet-50 encoder and trained on training set of ImageNet. † are results from [29]. Results style: **best**, <u>second best</u>

Method	Epochs 100	200	400	800
without multi-crop augmentations				
SimCLR† [10]	66.5	68.3	69.8	70.4
BYOL† [27]	66.5	70.6	**73.2**	**74.3**
SimSiam† [29]	68.1	70.0	70.8	71.3
MoCo-v3 [14]	68.9	-	-	<u>73.8</u>
DeepCluster-v2 [8]	-	-	70.2	-
SwAV† [8]	66.5	69.1	70.7	71.8
TWIST [25]	**70.4**	<u>70.9</u>	71.8	72.6
MIRA	<u>69.4</u>	**72.1**	<u>72.9</u>	<u>73.8</u>
with multi-crop augmentations				
DeepCluster-v2 [8]	-	-	-	75.2
SwAV [8]	72.1	<u>73.9</u>	<u>74.6</u>	<u>75.3</u>
TWIST [25]	<u>72.9</u>	73.7	74.4	74.1
MIRA	**73.5**	**74.8**	**75.5**	-

Table 2: **Linear evaluation on ImageNet.** Comparison with other self-supervised methods on ImageNet. *SL* denotes for *self-labeling* by [25]. Results style: **best**

Method	Arch.	Epochs	Top-1	Top-5
Supervised	R50	-	-	-
PCL [37]	R50	200	67.6	-
SimSiam [29]	R50	800	71.3	-
SimCLR-v2 [11]	R50	800	71.7	-
InfoMin [43]	R50	800	73	91.1
BarlowTwins [49]	R50	1000	73.2	91.0
VicReg [3]	R50	1000	73.2	91.1
SelfClassifier [1]	R50	800	74.1	-
TWIST w/o *SL* [25]	R50	800	74.1	-
BYOL [27]	R50	1000	74.3	91.6
MoCo-v3 [14]	R50	1000	74.6	-
DeepCluster-v2 [8]	R50	800	75.2	-
SwAV [8]	R50	800	75.3	-
DINO [8]	R50	800	75.3	-
TWIST w/ *SL* [25]	R50	450	**75.5**	-
MIRA	R50	400	**75.5**	**92.5**

Training details We train our model on the training set of the ImageNet-1k ILSVRC-2012 dataset [18] without using class labels. We use the same data augmentation scheme (color jittering, Gaussian blur, and solarization) and multi-crop strategy (two 224×224 and six 96×96) used in [9]. We use a batch size of 4096 and employ the LARS optimizer [48] with a weight decay of 10^{-6}. We use linearly scaled learning rate of $lr \times$ batch size$/256$ [26] with a base learning rate of 0.3. [3] We adjust the learning rate with 10 epochs of a linear warmup followed by cosine scheduling. We also use EMA network by default. When the EMA is used, we set the momentum update parameter to start from 0.99 and increase to 1 by cosine scheduling. We use temperature scales of $\tau_s = 0.1, \tau_t = 0.225$ with trade-off coefficient $\beta = 2/3$. We assign soft pseudo-labels after 30 steps of the fixed point iteration. We further discuss this choice in Sec. 4.3. Otherwise stated, we use the encoder model trained by MIRA with 400 epochs training and multi-crop augmentations for the evaluations in this section.

4.2 Main results

Linear evaluation Tables 1 and 2 report linear evaluation results. We follow the linear evaluation settings in [27, 10]. We train a linear classifier on the top of the frozen trained backbone with the labeled training set of ImageNet. We train for 100 epochs using a LARS optimizer with a batch size of 1024. We use a base learning rate of 0.3 and adjust the learning rate by cosine annealing schedule. We apply random-resized-crop and horizontal flip augmentations for training. We evaluate the representation quality by the linear classifier's performance on the validation set of ImageNet.

Table 1 shows linear evaluation performance in top-1 accuracy for different un-/self-supervised representation training epochs. We train and evaluate MIRA with and without multi-crop augmentations. With multi-crop augmentations, MIRA consistently outperforms baselines while achieving 75.5% top-1 accuracy with only 400 epochs of training. We also report that 200 epochs of training with MIRA can outperform the 800 epochs results of other baselines that don't use multi-crops. Without multi-crop augmentations, MIRA is comparable to MoCo-v3 [14] and performs slightly worse than BYOL [27]. However, MIRA performs the best among the clustering-based [6, 8] and information-driven [49, 25] methods.

In Table 2, we compare MIRA to other self-supervised methods with the final performance. MIRA achieves the state-of-the-art performance on linear evaluation of ImageNet with only 400 epochs of training. While TWIST can achieve similar performance to MIRA within 450 epochs, they require an extra training stage with *self-labeling*; without it, they achieve 74.1% accuracy with 800 epochs of training. In contrast, MIRA doesn't require additional training.

[3] Otherwise stated, we also use linearly scaled learning rate for evaluation training.

Table 3: ***k*-NN classification results on ImageNet with respect to subsets.** For 1% and 10% results, we evaluate the baselines by models of official codes. Other baseline results are from [9]. Results style: **best**

Method	ImageNet subset 100%	10%	1%
BYOL [27]	64.8	57.4	45.2
SwAV [8]	65.7	57.4	44.3
BarlowTwins [49]	66.0	59.0	47.7
DeepCluster-v2 [8]	67.1	59.2	46.5
DINO [9]	67.5	59.3	47.2
MIRA	**68.7**	**60.7**	**47.8**

Table 4: **Semi-supervised learning results on ImageNet.** The baselines results are from [49]. Results style: **best**, second best

	1%		10%	
	Top-1	Top-5	Top-1	Top-5
Supervised	25.4	48.4	56.4	80.4
SimCLR [10]	48.3	75.5	65.6	87.8
BYOL [27]	53.2	78.4	68.8	89
SwAV [8]	53.9	78.5	**70.2**	89.9
BarlowTwins [49]	55	79.2	69.7	89.3
MIRA	**55.5**	**80.3**	69.9	**90.0**

Table 5: **Linear evaluation results on the transfer learning datasets.** Following [21], we report top-1 accuracy on Food, CIFAR-10/100, SUN397, Cars, DTD; mean-per-class accuracy on Aircraft, Pets, Caltech-101, Flowers; 11-point mAP metric on VOC2007. Results style: **best**

	Aircraft	Caltech101	Cars	CIFAR10	CIFAR100	DTD	Flowers	Food	Pets	SUN397	VOC2007	avg.
Supervised	43.59	90.18	44.92	91.42	73.90	72.23	89.93	69.49	**91.45**	60.49	83.6	73.75
InfoMin [43]	38.58	87.84	41.04	91.49	73.43	74.73	87.18	69.53	86.24	61.00	83.24	72.21
MoCo-v2 [13]	41.79	87.92	39.31	92.28	74.90	73.88	90.07	68.95	83.3	60.32	82.69	72.31
SimCLR-v2 [11]	46.38	89.63	50.37	92.53	76.78	76.38	92.9	73.08	84.72	61.47	81.57	75.07
BYOL [27]	53.87	91.46	56.4	93.26	77.86	76.91	94.5	73.01	89.1	59.99	81.14	77.05
DeepCluster-v2 [8]	54.49	91.33	58.6	94.02	**79.61**	**78.62**	94.72	77.94	89.36	65.48	83.94	78.92
SwAV [8]	54.04	90.84	54.06	93.99	79.58	77.02	94.62	76.62	87.6	65.58	83.68	77.97
MIRA	**59.06**	**92.21**	**61.05**	**94.20**	79.51	77.66	**96.07**	**78.76**	**89.95**	**65.84**	**84.10**	**79.86**

Semi-supervised learning In Table 4, we evaluate the trained model on the semi-supervised learning benchmark of ImageNet. Following the evaluation protocol in [27, 10], we add a linear classifier on top of the trained backbone and fine-tune the model with ImageNet 1% and 10% subsets. We report top-1 and top-5 accuracies on the validation set of ImageNet. For the 1% subset, MIRA outperforms the baselines; both the top-1 and top-5 accuracies achieve the best. For the 10% subset, MIRA is comparable to SwAV [8].

***k*-NN evaluation** We further evaluate the quality of learned representation via the nearest neighbor classifier. We follow the procedures of [9]. First, representations of the labeled training data are stored. Then, the label of the new validation data is predicted with the majority vote of *k*-nearest stored representations. We use the same evaluation settings in [9] with 20 nearest neighbors, temperature scaling[4] of 0.07, and cosine distance metric.

Table 3 shows the k-NN classification accuracies on the validation set of ImageNet. We use 1/10/100% subsets of ImageNet training dataset to produce labeled representations. For ImageNet 1% and 10% subsets, we use the same subsets of semi-supervised learning evaluation. The results show that our method achieves state-of-the-art *k*-NN evaluation performance with ResNet50. To be more specific, our method outperforms the previous state-of-the-art DINO [9] on 100% and 10% subset evaluation by 1.2 $\sim$1.4%. We note that BarlowTwins [49], a method also motivated by information-maximization, shows a strong performance of 47.7% in the 1% subset evaluation.

Transfer learning We further evaluate the representation learned by MIRA on the transfer learning benchmark following [21] that includes FGVC aircraft [39], Caltech-101 [24], Standford Cars [34], CIFAR-10/100 [35], DTD [15], Oxford 102 Flowers [40], Food-101 [5], Oxford-IIIT Pets [41], SUN397 [44], and Pascal VOC2007 [23] datasets. We follow the linear evaluation procedure in [21] that fits a multinomial logistic regression model on the extracted representations of 2048d from the trained backbone. First, we perform a hyperparameter search on the L2-normalization coefficient of the logistic regression model; then the final performance is evaluated on the model that is retrained on all training and validation sets with the found coefficient.

[4]The temperature scaling τ is used to calculate contributions $\alpha_i \sim \exp(\text{distance}_i/\tau)$ and voting is weighted by the contributions of the nearest neighbors.

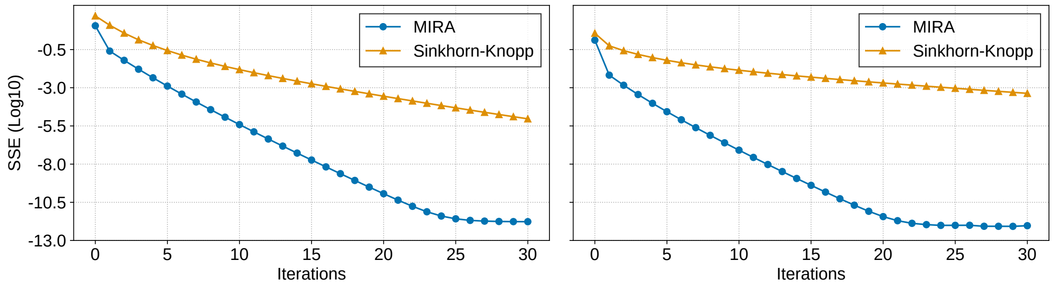

Figure 2: **Convergence analysis of MIRA and Sinkhorn-Knopp.** We observe the converging behavior of MIRA **(blue)** and Sinkhorn-Knopp **(yellow)**. We experiment with trained models of MIRA **(left)** and SwAV **(right)**. Since both methods are proven to converge, we iterate each method 1000 steps and regard the results as ground truth. We report the sum-squared error (SSE) with respect to the converging point in the log scale.

Table 5 shows the performance of our algorithm compared to other baselines in 11 datasets. MIRA outperforms supervised representation on 10 out of 11 datasets. Compared to the other self-supervised methods, representation learned by MIRA achieves the best performance in 9 out of 11 datasets and improves 0.9% over the second-best baseline method on average. The results confirm that the representation trained with MIRA has a strong generalization ability for classification.

4.3 Analysis

Convergence of pseudo-label assignment We study the speed of convergence of the proposed fixed-point iteration in MIRA. We also experiment with the Sinkhorn-Knopp (SK) algorithm [16] used in SwAV [8] as a baseline. We experiment with both methods on the ImageNet with a batch size of 512. We observe the converging behavior with the pre-trained models from MIRA and SwAV. Results are averaged over 1000 randomly sampled batches.

Figure 2 shows the result of the converging behavior of our method **(blue)** and SK algorithm **(yellow)** on trained models of MIRA **(left)** and SwAV **(right)**. Our fixed-point iteration converges faster than the SK algorithm in both pre-trained models. Especially our default setting of 30 steps of updates are sufficent for our fixed point iteration.

Multi-crop and EMA Table 6 reports an ablation study on how EMA and multi-crop augmentation affects our representation quality. We train a model for 200 epochs in the settings with and without EMA or Multi-crop. Both EMA and Multi-crop augmentations greatly improve the linear evaluation performance as in [8, 9]. We take a further comparison with baselines that are in the same setups. With the only difference in the pseudo-labeling algorithm, our method outperforms SwAV [8] by 1.3% in top-1 accuracy. While DINO [9] also uses both multi-crop and EMA, our method outperforms DINO with fewer training epochs. The results validate the effectiveness of our pseudo-labeling algorithm. These results validate the effectiveness of our pseudo-labeling algorithm.

Table 6: **Ablation study about EMA and multi-crop augmentation.** We report top-1 accuracy with linear evaluation on validation set of ImageNet. The results of SwAV is from [29].

Method	Multi-Crop	EMA	Epochs	Top-1
SwAV	✗	✗	200	69.1
DINO	✓	✓	300	74.5
	✗	✗	200	70.4
MIRA	✗	✓	200	72.1
	✓	✓	200	74.8

Scalability We further validate MIRA's scalability on the small-and medium-scaled datasets. ResNet-18 is used as a base encoder throughout the experiments. While changing the base encoder, other architectural details remain the same as in ImageNet-1k. We do not apply multi-crop augmentations while using the EMA. We use image sizes of 32×32 and 256×256 for small and medium datasets, respectively. Following the procedures in [17], we report the linear evaluation performance on the validation set. More experimental details about the optimizer, batch size, augmentations, etc., are provided in the Appendix.

Table 7: **Linear evaluation performance in small-and medium-scaled datasets.** We report top-1 and top-5 accuracies of linear evaluation on validation dataset. The training results are based on 1000 and 400 epochs of training on CIFAR-10/100 and ImageNet-100, respectively. Results style: **best**, <u>second best</u>

Method	Arch.	CIFAR-10		CIFAR-100		ImageNet-100	
		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
BarlowTwins [49]	R18	92.10	99.73	**70.90**	91.91	80.16	<u>95.14</u>
BYOL [27]	R18	<u>92.58</u>	<u>99.79</u>	70.46	<u>91.96</u>	<u>80.32</u>	94.94
DeepCluster-v2 [8]	R18	88.85	99.58	63.61	88.09	75.36	93.10
DINO [8]	R18	89.52	99.71	66.76	90.34	74.90	92.78
SwAV [8]	R18	89.17	99.68	64.88	88.78	77.83	95.06
MIRA	R18	**93.02**	**99.87**	<u>70.65</u>	**92.23**	**81.00**	**95.56**

The results are in Table 7. In CIFAR-10 and ImageNet-100, our method outperforms other self-supervised baselines by 0.4% and 0.7% in top-1 accuracy, respectively. For CIFAR-100, our method is comparable to the best performing baseline–BarlowTwins; MIRA performs better in top-5 accuracy.

Training with small batch Throughout the experiments in Sec. 4.2, we use a batch size of 4096. While such batch size is commonly used in self-supervised methods, large amounts of GPU memory are required; hence limiting the accessibility. In Table 8, we test our method with a smaller batch size of 512 that can be used in an 8 GPU machine with 96GB memory. In this setting, we use the SGD optimizer with a weight decay of 10^{-4}. We also test the robustness of pseudo-labeling with the Sinkhorn-Knopp algorithm in SwAV [8] reproduced by us and compare the results.

We report a top-1 linear evaluation performance of both methods after 100 epochs of training. In the result, the performance gap between our method and SwAV is amplified from 2.9% to 6% in the reduced batch size of 512. One possible explanation is that since SwAV is based on the equipartition constraint, the performance of SwAV harshly degrades when the batch size is not enough to match the number of clusters.

Table 8: **Linear evaluation performance with smaller batch size.** All results are based on ImageNet training. We also report the GPU memory usage and time spent for one epoch training. † is result by us.

Method	Batch size	Epochs	GPU	GPU memory	Time per Epoch	Top-1
SwAV†	512	100	8 × TITAN V	71 GB	23 min	62.3
MIRA w/o EMA	512	100	8 × TITAN V	71 GB	23 min	66.3
MIRA	512	100	8 × TITAN V	73 GB	29 min	68.3
SwAV [29]	4096	100	-	-	-	66.5
MIRA w/o EMA	4096	100	16 × A100	486 GB	9 min	68.7
MIRA	4096	100	16 × A100	504 GB	9 min	69.4

5 Discussion

Conclusion This paper proposes the mutual information maximization inspired pseudo-labeling algorithm MIRA. We formulate pseudo-labeling into an optimization problem and solve it in a principled way. We apply MIRA to representation learning and demonstrate its effectiveness in self-supervised learning benchmarks. We hope that our simple yet theoretically guaranteed approach to information maximization will guide many future applications.

Limitation and negative social impact Our information maximization perspective pseudo-labeling seems applicable to various tasks and domains, e.g., semi-supervised training [36]. We validate the effectiveness only in self-supervised visual representation learning. Furthermore, despite our improved training efficiency, the self-supervised learning methods still require a huge amount of computations compared to supervised learning. Such computational requirements may accelerate the environmental problems of global warming.

References

[1] E. Amrani, L. Karlinsky, and A. Bronstein. Self-supervised classification network. *arXiv preprint arXiv:2103.10994*, 2021.

[2] Y. M. Asano, C. Rupprecht, and A. Velaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

[3] A. Bardes, J. Ponce, and Y. LeCun. Vicreg: variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

[4] M. A. Bautista, A. Sanakoyeu, E. Sutter, and B. Ommer. Cliquecnn: Deep unsupervised exemplar learning. In *NIPS*, 2016.

[5] L. Bossard, M. Guilaumin, and L. V. Gool. Food-101 - mining discriminative components with random forests. In *ECCV*, 2014.

[6] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[7] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

[8] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

[9] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[11] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020.

[12] X. Chen and K. He. Exploring simple siamese representation learning. In *CVPR*, 2021.

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

[14] X. Chen, S. Xie, and K. He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021.

[15] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

[16] M. Cuturi. Lightspeed computation of optimal transport. In *NeurIPS*, 2013.

[17] V. G. T. da Costa, E. Fini, M. Nabi, N. Sebe, and E. Ricci. solo-learn: a library of self-supervised methods for visual representation learning. *Journal of machine learning research*, 2022.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: a large-scale hierarchical image database. In *CVPR*, 2009.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert:pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[20] J. Du, E. Grave, B. Gunel, V. Chaudhary, O. Celebi, M. Auli, V. Stoyanov, and A. Conneau. Self-training improves pre-training for natural language understanding. In *NAACL*, 2021.

[21] L. Ericsson, H. Gouk, and T. M. Hospedales. How well do self-supervised models transfer? In *CVPR*, 2021.

[22] A. Ermolov, A. Siarohin, E. Sangineto, and N. Sebe. Whitening for self-supervised representation learning. In *ICML*, 2021.

[23] M. Everingham, L. V. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 2010.

[24] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *CVPR workshops*, 2004.

[25] W. Feng, K. Tao, Z. Rufeng, L. Huaping, and L. Hang. Self-supervised learning by estimating twin class distributions. *arXiv preprint arXiv:2110.07402*, 2021.

[26] P. Goyal, P. Dollár, R. Girshick, P. Noord-huis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[27] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. Á. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent: a new approach to self-supervised learning. In *NeurIPS*, 2020.

[28] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[29] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[30] R. D. Hjelm, A. Fedorov, S. L-Marchildron, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

[31] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *ICML*, 2017.

[32] S. Ioeffe and C. Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[33] X. Ji, J. F. Henriques, and A. Veldaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019.

[34] J. Krause, J. Deng, M. Stark, and L. Fei-Fei. Collecting a large-scale dataset of fine-grained cars. In *Second workshop on fine-grained visual categorization*, 2013.

[35] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[36] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on challenges in representation learning*, 2013.

[37] J. Li, P. Zhou, C. Xiong, and S. Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021.

[38] R. Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21:105–117, 1988.

[39] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[40] M. E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Indian conference on computer vision, graphics and image processing (IVCGIP)*, 2008.

[41] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012.

[42] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Kruger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[43] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola. What makes for good views for contrastive learning? In *NeurIPS*, 2020.

[44] J. Xiao, J. Hays, S. X. Yu, and D. Lin. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[45] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016.

[46] X. Yan, I. Misra, A. Gupta, D. Ghadiyaram, and D. Mahajan. Clusterfit: Improving generalization of visual representations. In *CVPR*, 2020.

[47] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, 2016.

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

[49] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: self-supervised learning via redundancy reduction. In *ICML*, 2021.

[50] M. Zheng, S. You, F. Wang, C. Qian, C. Zhang, X. Wang, and C. Xu. Ressl: relational self-supervised learning with weak augmentation. In *NeurIPS*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5 Discussion - Limitation and negative social impact
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5 Discussion - Limitation and negative social impact
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] in the Appendix
   (b) Did you include complete proofs of all theoretical results? [Yes] in the Appendix
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] in supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] in Appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] in Appendix
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] in Appendix
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] code in supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]