Identity-Disentangled Adversarial Augmentation for Self-Supervised Learning

Kaiwen Yang [1] **Tianyi Zhou** [2 3] **Xinmei Tian** [1 4] **Dacheng Tao** [5]

Abstract

Data augmentation is critical to contrastive self-supervised learning, whose goal is to distinguish a sample's augmentations (positives) from other samples (negatives). However, strong augmentations may change the sample-identity of the positives, while weak augmentation produces easy positives/negatives leading to nearly-zero loss and ineffective learning. In this paper, we study a simple adversarial augmentation method that can modify training data to be hard positives/negatives without distorting the key information about their original identities. In particular, we decompose a sample x to be its variational auto-encoder (VAE) reconstruction $G(x)$ plus the residual $R(x) = x - G(x)$, where $R(x)$ retains most identity-distinctive information due to an information-theoretic interpretation of the VAE objective. We then adversarially perturb $G(x)$ in the VAE's bottleneck space and adds it back to the original $R(x)$ as an augmentation, which is therefore sufficiently challenging for contrastive learning and meanwhile preserves the sample identity intact. We apply this "identity-disentangled adversarial augmentation (IDAA)" to different self-supervised learning methods. On multiple benchmark datasets, IDAA consistently improves both their efficiency and generalization performance. We further show that IDAA learned on a dataset can be transferred to other datasets. Code is available at https://github.com/kai-wen-yang/IDAA.

[1]University of Science and Technology of China, Hefei, China [2]University of Washington, Seattle, USA [3]University of Maryland, College Park, USA [4]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China [5]JD Explore Academy, Beijing, China. Correspondence to: Tianyi Zhou <tianyizh@uw.edu>, Xinmei Tian <xinmei@ustc.edu.cn>, Dacheng Tao <dacheng.tao@gmail.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

1. Introduction

Empowered by deep neural networks and the computational capability of recent hardware, machine learning has achieved breakthroughs on some challenging problems when sufficient labeled data is available. However, deep learning is known to be data-hungry and annotating data in many domains, e.g., medical care or predictions of protein structures, are either difficult or expensive. To overcome this limitation, self-supervised learning (SSL) methods train a model on unlabeled data in a supervised learning manner using self-generated labels by manipulating the data, e.g., rotation degrees (Gidaris et al., 2018), Jigzaw puzzle solutions (Noroozi & Favaro, 2016), clustering (Caron et al., 2018), back-translation (Zhu et al., 2017), etc. These methods recently start to perform on par with supervised learning and exhibit potential to even surpass it (Chen et al., 2020a;b). Moreover, their learned representations can be generally applied to different downstream tasks.

Many widely-used SSL methods are built upon sample-identity preservation tasks, e.g., contrastive learning (Oord et al., 2018; Tian et al., 2020a; Chen et al., 2020a) and consistency regularization (Chen & He, 2021; Grill et al., 2020; Caron et al., 2020), which aim at learning representations that can preserve the identity of the original sample after applying data augmentations and thus distinguish the augmentations of different samples. For example, contrastive learning targets on an representation space in which a sample (anchor) is closer to its own augmentations (positives) than other samples or their augmentations (negatives). The effectiveness of contrastive learning therefore heavily depends on the quality of data augmentations.

Most SSL methods (Ye et al., 2019; Chen et al., 2020a; He et al., 2020; Chen & He, 2021) utilize pre-defined data augmentations to generate positives and negatives. However, as shown in the example (green point) of Fig. 1, they are not adaptive to the data manifold in the embedding space during training and the generated augmentations can be too easy for the sample identification task. In practice, these SSL methods need tens to hundreds times of epochs required by supervised learning to reach comparable performance (Chen et al., 2020a; He et al., 2020; Chen & He, 2021). For the same reason, large batch-size is common and necessary for contrastive learning in order to involve sufficient hard negatives. Therefore, how to modify the positives/negatives

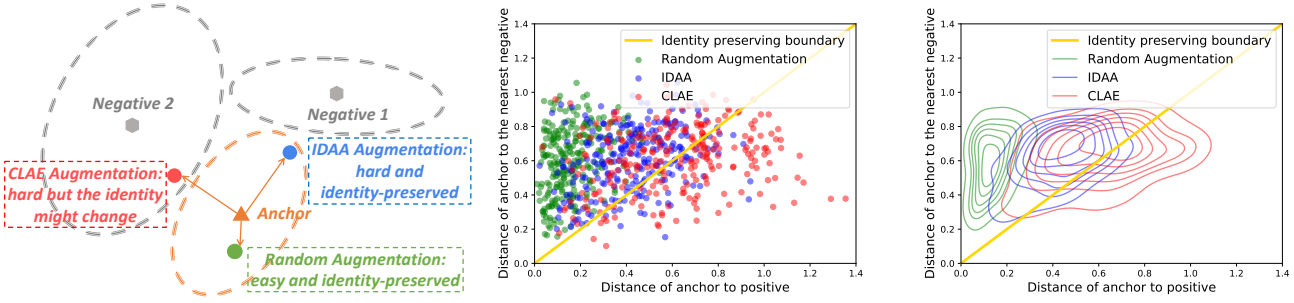


Figure 1. Data augmentations for contrastive Learning: IDAA, CLAE and random augmentation. (a) Identity-preserving and hardness of data augmentations: random augmentation generates identity-preserved but easy samples, CLAE can adversarially generates hard but identity-distorted samples, while IDAA (ours) can generate hard and identity-preserved samples. (b) Identity-preserving and hardness of augmentations by different methods: Points below the boundary line have the identity changed, while points close to the boundary are hard samples. (c) Kernel density plot for (b): IDAA generates hard samples without changing the identities.

for more informative contrastive loss is a critical yet open challenge towards more efficient SSL. Learnable and adaptive data augmentations have been explored for supervised learning (Cubuk et al., 2018; 2020; Liu et al., 2021) but have not been widely studied in SSL. Probably the most related augmentation method for SSL is CLAE (Ho & Vasconcelos, 2020), which generates hard positives/negatives by adversarially attacking the input. However, as shown in Fig. 1, adversarial augmentations may change the original sample identity and it is infeasible to tune the attack strength for every sample to preserve the identity.

Hence, another challenge for data augmentation in SSL is how to preserve sample identity. Although hard positives/negatives might be achievable by stronger or adversarial augmentations listed above, they can also distort the true identities of the original samples as the red point in Fig. 1, so the model may erroneously identify other different samples or their augmentations as the anchor sample. Training with those identity-changed augmentations might lead to trivial solutions for SSL and poor representations.

To address the aforementioned two primary challenges of contrastive learning, we have to consider how they interfere with each other. For better efficiency, the data augmentation needs to generate positives and negatives as challenging as possible for the model to distinguish the sample identity, but it should not remove or distort the minimum necessary information retaining the true identity. Thereby, the sample identification task is neither too trivial nor infeasible to learn. In this paper, we study how to automatically generate data augmentations that fulfill the above conditions and improve both the efficiency and effectiveness of current self-supervised learning. We relate the objective of variational antoencoder (VAE) with the sample identification task in contrastive learning from an information-theoretical perspective, which inspires us to disentangle the identity-essential information of an input x as the residual of VAE reconstruction $G(x)$, i.e., $R(x) = x - G(x)$. As illustrated in Fig. 2, in order to modify x to be more challenging in terms of sample identification, we propose to apply adversarial perturbations to the bottleneck features of VAE, which maximize the contrastive loss in an ϵ-ball and results in a modified $G'(x)$. We then utilize $x' = G'(x) + R(x)$ as a data augmentation of x so the identity information captured by $R(x)$ remains intact in x'.

Our method, called "identity-disentangled adversarial augmentation (IDAA)", only needs a VAE model pre-trained on unlabeled data. In the experiments on multiple benchmarks, when applied to different SSL methods, this simple yet principal data augmentation approach consistently brings improvements on both the efficiency and downstream task performance. Although our theory is mainly based on contrastive learning, IDAA consistently brings empirical improvement to other SSL methods such as SimSiam (Chen et al., 2020a) because the key idea of "identity-preserving" helps general SSL methods to avoid representational collapse. In addition, we present a thorough ablation study to analyze the influence of hyperparameters (e.g., VAE hyperparameters, bottleneck dimensions, attack strength ϵ) and experimental settings (e.g., batch size, training epochs, model architecture) on the contrastive learning's performance.

2. Background

Contrastive learning (CL) (Wu et al., 2018; Zhuang et al., 2019; Chen et al., 2020a) aims at learning representations that can distinguish different samples and their augmentations. Specifically, it formulates this sample-identification task as a classification problem on each sample (anchor), where the positives are its own augmentations and the negatives are other samples and/or their augmentations.

A widely-used loss for CL is "InfoNCE" (Tian et al., 2020a) built upon the positives and negatives created by data aug-

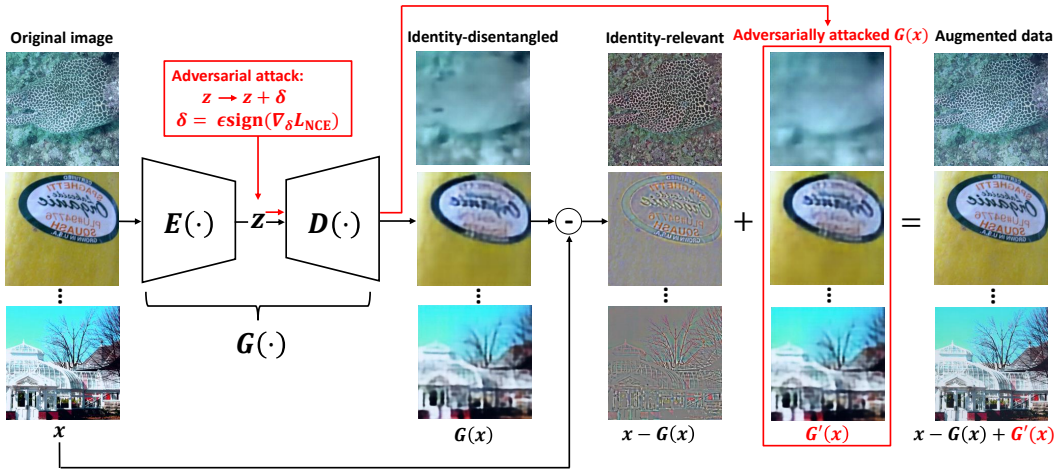


Figure 2. Architecture and pipeline of Identity-Disentangled Adversarial Augmentation (IDAA).

mentations. For each mini-batch $\vec{x} = \{x_1, x_2, \cdots, x_N\}$ of size N, InfoNCE loss is computed as:

$$L_{\text{NCE}}(\vec{x}) = -\frac{1}{N}\sum_{i=1}^{N} \log q_{\text{NCE}}(i|x = x_i),$$
$$q_{\text{NCE}}(i|x = x_i) \triangleq \frac{\exp\langle f(A(x_i)), h(B(x_i))\rangle}{\sum_{j=1}^{N} \exp\langle f(A(x_i)), h(B(x_j))\rangle}, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes inner product, $A(\cdot)$ and $B(\cdot)$ denote two different transformations for data augmentation, $f(\cdot)$ and $h(\cdot)$ are the embedding networks that can be either identical (Chen et al., 2020a; Ye et al., 2019) or different (Misra & Maaten, 2020; He et al., 2020), and $q_{\text{NCE}}(i|x = x_i)$ is the softmax probability of identifying the augmentation $A(x_i)$ is transformed from the original sample x_i. The choices of data augmentation are critical to the performance and efficiency of CL: weak augmentations might already result in fully distinguishable representations over different samples and nearly zero CL loss, while strong augmentations may overly distort the identity of a sample and make CL too challenging, infeasible, or inefficient. In previous works such as SimCLR (Chen et al., 2020a), different (compositions of) augmentations can result in large gaps on CL's performance but finding the best one is difficult and time-consuming.

Another challenge in CL is to keep a large memory bank or mini-batch to cover sufficient amount of "hard negatives" because a sample might be distant from most other samples (and their augmentations) and hard negatives close to it are sparse in the training set. (Wu et al., 2018) trains a non-parametric classifier to maximally scatter the representations of all samples over a unit sphere. However, it needs to build a memory bank to store all these representations, which is infeasible for large-scale datasets. (He et al., 2020) instead maintain a fixed-sized dynamic dictionary based on a FIFO queue of representations of incoming mini-bathes during training. It finds that a relatively large dictionary is critical to the success of CL. The lack of hard negatives forms a bottleneck to the sample efficiency of CL. Therefore, how to select or modify the data to increase the chance of including more hard negatives is a significant open problem for CL.

Consistency regularization (CR) has been studied in several recent works for self-supervised (Chen & He, 2021; Caron et al., 2020) or semi-supervised learning (Zhou et al., 2020; Sohn et al., 2020), which achieves comparable or even better performance than CL. Compared to CL, CR removes the comparison to negatives and only focus on maximizing the similarity between model outputs for two augmentations of the same sample, e.g.,

$$L_{\text{CS}}(\vec{x}) = -\sum_{i=1}^{N} \frac{\langle f(A(x_i)), h(B(x_i))\rangle}{\|f(A(x_i))\| \cdot \|h(B(x_i))\|}, \tag{2}$$

Similar to CL, CR also aims at preserving the sample identity on the learned representations and it heavily relies on the choice of data augmentations. For example, FixMatch (Sohn et al., 2020) chooses to apply multiple weak augmentation for generating the pseudo-labels (e.g., $f(A(\cdot))$) and strong augmentations to the other branch $h(B(\cdot))$.

3. Identity-Disentangled Adversarial Augmentation

In this section, we will firstly relate the sample-identification task broadly used in designs of self-supervised learning with the training objective of variational auto-encoder (VAE). Specifically, we will show that VAE's training tries to remove the sample-identity related information from its bottleneck features. Hence, the residual of VAE reconstruction may retain the sample-identity information that we wish to preserve in the data augmentation for self-supervised learning. We then propose a data generation and augmentation model based on VAE and analyze the lower bound for identity preservation in its augmented data. In the end of this section, we apply adversarial attack methods to this data augmentation model, which modify samples to be hard pos-

itives and negatives for more efficient contrastive learning without changing their sample-identities.

3.1. Identity-Disentanglement in Contrastive Learning

The goal of sample-identification can be formulated as maximizing $q(y=i|x=x_i)$, i.e., the likelihood estimation of identifying x_i or its augmentations as sample-i, where x and y are two random variables for a sample and an identity label, respectively. The following proposition shows that $q(y=i|x=x_i)$ provides a lower bound for the mutual information $I(x;y)$ between x and y.

Proposition 3.1. (Sample-identification likelihood as a lower bound of $I(x;y)$). *If $\vec{x}$ is a random mini-batch of size N and the sample-identification likelihood estimation of x_i on its correct identification label $y=i$ to be $q(y=i|x=x_i)$, the mutual information $I(x;y)$ can be lower bounded by:*

$$\begin{aligned} I(x;y) &= \log N + \mathbb{E}_{\vec{x}}\left[\frac{1}{N}\sum_{i=1}^{N}\log p(y=i|x=x_i)\right] \\ &\geq \log N + \mathbb{E}_{\vec{x}}\left[\frac{1}{N}\sum_{i=1}^{N}\log q(y=i|x=x_i)\right]. \end{aligned} \tag{3}$$

where $p(\cdot)$ is the true probability and $q(\cdot)$ denotes an estimation of $p(\cdot)$. A detailed proof is given in the Appendix. In contrastive learning, $p(y=i|x=x_i)$ can be modeled and estimated by $q_{\mathrm{NCE}}(i|x=x_i)$ defined in Eq. (1) using neural network embedding $f(\cdot)$ and $h(\cdot)$ of data augmentations $A(\cdot)$ and $B(\cdot)$. Hence, InfoNCE loss can provide a lower-bound of $I(x;y)$, i.e.,

$$\begin{aligned} I(x;y) &\geq \log N + \mathbb{E}_{\vec{x}}\left[\frac{1}{N}\sum_{i=1}^{N}\log q_{\mathrm{NCE}}(i|x=x_i)\right] \\ &= \log N - \mathbb{E}_{\vec{x}}\left[L_{\mathrm{NCE}}(\vec{x})\right]. \end{aligned} \tag{4}$$

The above lower bound relates the mutual information $I(x;y)$ and CL: increasing the batch size N and/or minimizing the InfoNCE loss can improve the tightness of the lower bound and results in representations with better capability on sample identification. Due to the natural sparsity of hard examples in the original data $\vec{x}$, data augmentations of $\vec{x}$ are necessary to improve the sample efficiency when minimizing $\mathbb{E}_{\vec{x}}\left[L_{\mathrm{NCE}}(\vec{x})\right]$. However, without any constraints, strong data augmentations may remove the identity-related information of some data and change their identities, which results in $q_{\mathrm{NCE}}(i|x=x_i) \ll p(y=i|x=x_i)$, a loose lower bound in Eq. (4), and poor representations via CL.

3.2. Identity-Disentanglement via VAE

Next, we will show that the training objective of VAE (Kingma & Welling, 2013) is also related to $I(x,y)$ and sample identification. In particular, VAE aims at removing identity-specific information from the bottleneck features z, i.e., minimizing $I(z,y)$. With an output $G(x)$ decoded from z, VAE naturally disentangles identity-relevant information from x. If we can remain such information, i.e., $x-G(x)$, intact in data augmentations and only perturb the rest part $G(x)$, the above problem of CL can be resolved.

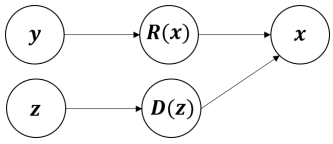


Figure 3. Data generative model.

Lemma 3.2. (VAE objective and $I(z;y)$ from Eq. (29) in (Alemi et al., 2016)). *Assume that the bottleneck features of VAE are denoted by z, the encoder is $E(\cdot)$ and produces distribution $p_E(z|x)$, the decoder is $D(\cdot)$ and produces distribution $q_D(x|z)$, the prior for z is $p(z)$, and the KL-divergence regularization in the VAE objective L_{VAE} has a weight β, we have:*

$$-I(z;x) + \beta I(z;y) \leq L_{\mathrm{VAE}}, \tag{5}$$

$$\begin{aligned} L_{\mathrm{VAE}} \triangleq &-\int \mathrm{d}x p(x)\int \mathrm{d}z p_E(z|x)\log q_D(x|z) \\ &+\beta\frac{1}{N}\sum_{i=1}^{N}\mathrm{D}_{\mathrm{KL}}\left(p_E\left(z|x=x_i\right)||p(z)\right), \end{aligned} \tag{6}$$

In L_{VAE}, β (Higgins et al., 2016) controls how close the distribution of bottleneck features $p_E(z|x=x_i)$ is to the prior $p(z)$, e.g., a standard Gaussian distribution independent of the sample identity $y=i$. So β controls the strength of identity disentanglement on z. Lemma 3.2 shows that VAE is trained to minimize an upper bound of $-I(z,x)+\beta I(z,y)$, i.e., preserving most information of input x in the bottleneck feature z but removing from z the critical information about sample identity $y=i$. Hence, the most identity-relevant information is disentangled from the VAE output $G(x)\sim p_D(x|z)=p(x|D(z))$ and preserved in the residual $R(x)\triangleq x-G(x)$. We discuss the choices of other generative models in Sec. D.2 of the Appendix.

3.3. Identity-Disentangled Data Augmentation

We can study a data generative model based on identity-disentanglement of VAE, which generates $D(z)$ and $R(x)$ from z and y respectively and combine them to generate $x=G(x)+R(x)$, as shown in Fig. 3. The following lemma compares $I(R(x);y)$ with $I(x;y)$ and analyzes how much identity information can be preserved in $R(x)$.

Lemma 3.3. (Identity-disentangled data generation). *For a data generative model described above,*

$$I(R(x);y) \geq I(x;y) - I(z;y). \tag{7}$$

The detailed proof of Lemma 3.3 is given in the appendix.

Assumption 3.4. (Identifiability extended from Theorem 1

of Robust PCA (Candès et al., 2011)). There exists a small $\epsilon > 0$, when perturbing z within the ϵ-ball, the identity-disentangled part $D(z)$ and identity-relevant part $R(x)$ are still separable using VAE (more details in the Appendix).

Theorem 3.5. (Identity-disentangled data augmentation). *If we use a VAE in the identity-disentangled data generative model for Lemma 3.3, and if we define an augmentation $x' = R(x) + G'(x)$ with $G'(x) \sim q_D(x|z')$ and $z' = z + \delta$ (a δ-perturbed z)), there exists a small $\epsilon > 0$ such that for any $\|\delta\|_p \leq \epsilon$, we can lower bound $I(x'; y)$ as*

$$I(x'; y) \geq I(x; y) - \frac{1}{\beta}(L_{\text{VAE}} + I(z; x)). \tag{8}$$

Theorem 3.5 is proved by combing Lemma 3.2, 3.3, and Assumption 3.4 which is illustrated in the Appendix.

Therefore, the augmentation x' preserves most of the identity information in x if $L_{\text{VAE}} + I(z; x)$ is small. VAE training aims at minimizing L_{VAE} so the first term can be kept small. There is a trade-off between the second term and identity preservation $I(x'; y)$ since a sufficiently large $I(z; x)$ is necessary to produce augmentation x' approximately drawn from the true data distribution. A key observation of the above theorem is that we can perturb z arbitrarily within the ϵ-ball (as long as it fulfills Assumption 3.4) to generate x' without hurting the lower-bound of $I(x'; y)$. This implies that we can adversarially perturb z to produce hard negatives and positives for more efficient CL without heavily distorting the original identity information. In contrast, most data augmentation techniques used in SSL have not taken this into account so they may change the sample-identity and result in poor representations. A formal definition of "identity-preserving" is given in Sec. D.1 of the Appendix.

3.4. Identity-Disentangled Adversarial Augmentation

As discussed in Sec. 2 and Sec. 3.1, keeping a sufficient amount of hard positives and negatives in each batch $\vec{x}$ is critical to the effectiveness and sample efficiency of contrastive learning. Although any ϵ-bounded perturbation on z can produce an identity-preserved data augmentation x' according to Theorem 3.5, adversarially perturbing the VAE-bottleneck feature z of the original samples can produce hard positives/negatives with sample-identities preserved. Fortunately, we can use off-the-shelf adversarial attack algorithms for this purpose. The main difference here is: (1) we apply them to perturb the bottleneck features z instead of x; and (2) the objective is to maximize the InfoNCE loss instead of a classification loss such as cross entropy. We call this method "identity-disentangled adversarial augmentation (IDAA)", as illustrated in Fig. 2. The architecture is similar to Class-Disentangled VAE (Yang et al., 2021).

In particular, the goal of IDAA is to generate an augmentation of x in the form of x' in Theorem 3.5 such that the InfoNCE loss is maximized using an ϵ-bounded perturbation δ to $z = E(x)$ when generating x'. For a batch of data $\vec{x}$, this problem can be formulated as optimizing $\{\delta_i\}_{i=1}^N$ by

$$\max_{\|\delta_i\|_p \leq \epsilon,\ \forall i \in [N]} L_{\text{NCE}}(\vec{x}'), \quad x'_i = R(x_i) + D(E(x_i) + \delta_i). \tag{9}$$

To only perturb the positives and negatives to be harder for contrastive learning while keeping the anchors intact, we slightly modify $L_{\text{NCE}}(\vec{x}')$ to be

$$L_{\text{NCE}}(\vec{x}') = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\langle f(x_i), h(x'_i)\rangle}{\sum_{j=1}^{N} \exp\langle f(x_i), h(x'_j)\rangle},$$

Since the augmentation x' can be a positive for its own identification (i.e., when x is the anchor) and a negative for other samples' identification, the objective in Eq. (9) modifies a sample x to be both a hard positive and a hard negative. In other words, IDAA perturbs the bottleneck features to move x' away from x to other difference samples in the embedding space. Besides InfoNCE loss, IDAA can be applied to other self-supervised losses, e.g., replacing $L_{\text{NCE}}(\cdot)$ in Eq. (9) with the consistency regularization $L_{\text{CS}}(\cdot)$ in Eq (2).

IDAA is complementary to and can be applied to existing data augmentation techniques by replacing x_i in Eq. (9) with a pre-defined data augmentation $A(x_i)$. It can also be applied together with other data augmentations, e.g., by combining multiple InfoNCE loss terms computed on different augmentations. A primary advantage of IDAA is that the generated augmentations are adaptive to the training models $f(\cdot)$ and $h(\cdot)$. It aims at finding the weaknesses of the SSL models on the sample-identification task and improve them without heavily distorting the original sample-identities, which is underexplored in previous literature.

There exists various off-the-shelf adversarial attack methods that can be directly applied to solve the problem in Eq. (9). In this paper, we adopt fast gradient sign method (FGSM) (Goodfellow et al., 2014b) for its computational efficiency. FGSM perturbs $z = E(x)$ for one step by adding noises along the gradient sign's direction of the loss w.r.t. δ and the augmentation x' is generated based on the perturbed bottleneck features $z + \delta$, i.e.,

$$x' = R(x) + D\left(E(x) + \delta^*\right), \quad \delta^* = \epsilon\,\text{sign}(\nabla_\delta L_{\text{NCE}}(\vec{x}')).$$

4. Experiments

In this section, we evaluate the improvements that IDAA as a data augmentation method brings to several popular methods in (1) self-supervised learning and (2) semi-supervised learning on standard benchmarks such as CIFAR (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009). We compare IDAA with (1) default random augmentations used in the original methods; (2) other data augmentation methods for SSL (Ho & Vasconcelos, 2020; Tian et al., 2020a; Hu et al., 2021); and (3) view generation

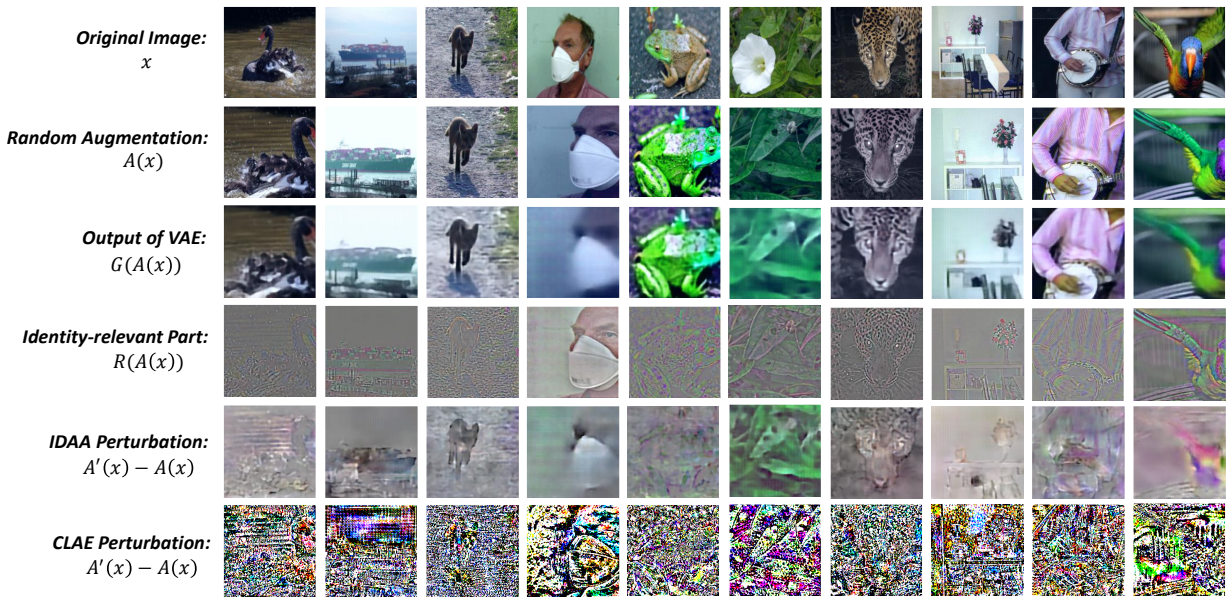


Figure 4. Comparing IDAA and CLAE on their adversarial augmentations for ImageNet samples.

methods (Kalantidis et al., 2020; Chuang et al., 2020). In all experiments, IDAA consistently improves both the training efficiency and the test accuracy of all the methods and significantly outperforms other data augmentation methods. Moreover, IDAA trained on one dataset can be transferred to other datasets and improve downstream tasks on them, demonstrating the generalization of the learned augmentation model. A walk-clock time comparison is given in Sec. C.3 of the Appendix, in which IDAA effectively reduces the computational cost. In addition, we conduct a thorough sensitivity study of IDAA by changing (1) batch sizes; (2) network architectures; (3) training epochs; (4) regularization weight β in the VAE objective; (5) dimensions of VAE's bottleneck; and (6) adversarial attack strength ϵ. It's worth noting that by following most data augmentation and view generation methods for SSL (Ho & Vasconcelos, 2020; Tian et al., 2020a; Hu et al., 2021; Kalantidis et al., 2020; Chuang et al., 2020), IDAA is also applied on top of each SSL method's default random augmentations for fair comparisons. More implementation details can be found in Sec. B of the Appendix.

4.1. Case Study of Augmentations by IDAA vs. CLAE

We firstly present a case study of augmentations generated by IDAA, CLAE, and widely used random augmentations. We compare their identity preserving, hardness to contrastive learning, and their perturbation patterns. In Fig. 1 (b)-(c), we compare the distance of an anchor to a positive with its distance to the nearest negative in the embedding space we apply the contrastive learning. If the latter is smaller than the former, i.e., the point is located below the "identity-preserving boundary" in the plots, its original identity cannot be preserved in the augmentation. On the other hand, if the latter is much larger than the former, i.e., the point is located on the left region of the plots, the sample identification task is too trivial and the contrastive learning is not efficient. From the plots, we can see that random augmentations are too easy while the CLAE augmentations are much harder but cannot preserve the original identity for some samples. In contrast, IDAA (ours) produces sufficiently hard augmentations within the boundary of identity preserving, which is ideal for contrastive learning.

We visualize the augmentations produced by the three methods in Fig. 4 for natural images from ImageNet. Both IDAA and CLAE are applied to random augmentations $A(x)$ and introduce further perturbations $A'(x) - A(x)$. IDAA generates more semantic perturbations to important regions of the images because its adversarial attack is conducted in the VAE bottleneck space, which is supposed to capture semantic attributes of the images. On the contrary, CLAE generates the perturbations on the input pixels and produces unnatural artifacts, which may distort the original sample-identity. In addition, we show the identity-relevant part $R(A(x))$ and the identity-disentangled part $G(A(x))$ produced by VAE. The former well preserves the most important patterns to identity the original sample, e.g., the edge of the steamship (column 2) and spots of the frog (column 5). Since IDAA keep it intact in its augmentation model, the identity is well preserved in the augmentations.

4.2. Self-Supervised Learning

We evaluate IDAA and compare it with default random augmentations, data augmentations designed for SSL such as

Table 1. Top-1 accuracy of linear and KNN evaluation on four self-supervised learning methods using different data augmentations, i.e., their default ones, CLAE, and IDAA (ours). All experiments train a ResNet-18 for 300 (100) epochs on CIFAR (miniImageNet).

Method	kNN			Linear Evaluation		
	CIFAR10	CIFAR100	miniImageNet	CIFAR10	CIFAR100	miniImageNet
Plain	82.78±0.20	54.73±0.20	46.96±0.32	79.65±0.43	51.82±0.46	44.90±0.29
Plain+CLAE	83.09±0.19	55.28±0.12	47.01±0.28	79.94±0.28	52.14±0.21	45.43±0.15
Plain+IDAA	**86.00±0.16**	**58.64±0.15**	**47.83±0.29**	**82.83±0.10**	**56.12±0.16**	**46.81±0.16**
UEL	83.63±0.14	55.23±0.28	40.71±0.73	80.63±0.18	52.99±0.25	43.08±0.35
UEL+CLAE	84.00±0.15	55.96±0.06	41.75±0.39	80.94±0.13	54.27±0.40	44.32±0.24
UEL+IDAA	**86.69±0.13**	**59.04±0.18**	**43.24±0.32**	**83.65±0.17**	**57.25±0.19**	**45.74±0.30**
SimSiam	88.22±0.10	57.13±0.20	31.68±0.28	89.84±0.15	62.76±0.13	40.62±0.48
SimSiam+CLAE	85.59±0.21	53.88±0.08	27.77±3.47	87.77±0.08	60.89±0.22	37.32±0.47
SimSiam+IDAA	**89.08±0.12**	**58.19±0.19**	**32.14±0.58**	**90.99±0.18**	**65.21±0.37**	**41.24±0.51**
SimCLR	80.79±0.10	41.11±0.28	30.13±0.28	86.40±0.18	57.81±0.10	46.13±0.23
SimCLR+CLAE	80.27±0.18	43.57±0.17	32.23±0.08	85.25±0.07	57.69±0.25	46.76±0.16
SimCLR+IDAA	**83.41±0.22**	**46.78±0.22**	**33.66±0.16**	**88.07±0.22**	**60.90±0.08**	**48.23±0.23**

Table 2. Top-1 accuracy of linear and KNN evaluation on CIFAR. All the methods train a ResNet-50 for 400 epochs.

Method	kNN		Linear Evaluation	
	CIFAR10	CIFAR100	CIFAR10	CIFAR100
Debiased	88.98	57.57	91.55	68.16
Debiased+IDAA	**90.75**	**63.00**	**93.19**	**72.67**
HCL	89.03	60.46	91.48	68.83
HCL+IDAA	**90.88**	**64.21**	**93.10**	**71.81**

CLAE (Ho & Vasconcelos, 2020), InfoMin (Tian et al., 2020b) and AdCo (Hu et al., 2021), and effective/hard view generation like HCL (Robinson et al., 2020) and Debiased (Chuang et al., 2020), on several benchmarks by following each method's original setting.

Comparison with Random and Adversarial Augmentations: Since CLAE (Ho & Vasconcelos, 2020) also studies adversarial augmentation as IDAA, it serves as an important baseline. Following CLAE's original setting, we compare IDAA with random augmentation and CLAE by applying them to four SSL methods: Plain (InfoNCE in Eq. (1), UEL (Ye et al., 2019), SimSiam (Chen & He, 2021) and SimCLR (Chen et al., 2020a), on three datasets. As reported in Table 1, IDAA consistently improves the original SSL methods (with their default random augmentations) and substantially outperforms CLAE on the improvement. In contrast, though CLAE performs better or comparable than the default augmentations in contrastive learning methods, it results in significant degradation when applied to SimSiam. Since SimSiam optimizes the anchor-positive consistency without considering the negatives, it is more prone to possible identity distortion in augmentations introduced by CLAE. Instead, IDAA consistently improves different types of SSL methods due to the identity preservation.

Table 3. Linear evaluation (top-1 and top-5) accuracy of ResNet-50 on ImageNet. § denotes reproduced results using the official code.

Method	Epoch	Batch Size	ImageNet Top-1	ImageNet Top-5
MoCo (He et al., 2020)	200	256	60.6	-
MoCo v2 (Chen et al., 2020b)	200	256	67.5	88.2
MoCHi (Kalantidis et al., 2020)	800	512	68.7	-
SimCLR (Chen et al., 2020a)	1000	4096	69.3	89.0
SwAV (Caron et al., 2020)	400	4096	70.1	-
AdCo (Hu et al., 2021)	200	256	68.6	-
InfoMin (Tian et al., 2020a)	200	256	70.1	89.4
SimSiam (Chen et al., 2020a)	100	256	68.1	-
SimSiam (Chen et al., 2020a)	200	256	70.0	-
SimSiam§	100	256	68.1	88.2
SimSiam§+IDAA	100	256	69.0	88.8
SimSiam§	200	256	69.8	89.2
SimSiam§+IDAA	200	256	**70.6**	**89.7**

Comparison with methods of Effective Views or Hard Negatives: hard sample mining (Robinson et al., 2020) or effective view generation (Chuang et al., 2020) aims to improve SSL efficiency through hard or effective data. Hence, we study whether IDAA can improve HCL (Robinson et al., 2020) and Debiased (Chuang et al., 2020) by applying IDAA to them. We follow their original setting to train a ResNet-50 for 400 epochs on CIFAR and report the results in Table 2, which shows that IDAA improves HCL and Debiased by a large margin, e.g., $> 3\%$ kNN accuracy on CIFAR100. Therefore, IDAA is complementary to this category of data selection methods, e.g., one can apply IDAA for data augmentation and then apply HCL for data selection.

ImageNet Experiments: To evaluate IDAA on high-resolution large-scale dataset, we apply IDAA to SimSiam (Chen & He, 2021) on ImageNet (Deng et al., 2009) since SimSiam performs better than many SSL methods

Table 4. Transfer learning performance (test accuracy) of a ResNet-18 (trained on ImageNet100) on other datasets.

	CIFAR10	CIFAR100	Birdsnap	Aircraft	DTD	Pets	Flower	CUB-200
SimCLR	61.83	36.55	12.68	24.19	54.35	46.46	75.00	16.73
SimCLR+CLAE	61.59	37.13	13.61	25.87	52.12	43.55	76.82	17.58
SimCLR+IDAA	**64.49**	**38.82**	**13.89**	**26.02**	**54.97**	**46.76**	**77.99**	**18.15**

Table 5. Transfer learning performance on object detection and instance segmentation. SimSiam and "SimSiam+IDAA" are pre-trained for 200 epochs on ImageNet, fine-tuned using Mask R-CNN(He et al., 2017) in COCO 2017 train, and evaluated in COCO 2017 val. Scratch denotes training a model with the same structure from scratch. "ImageNet supervised" is the supervised ImageNet pre-training counterpart.

Method	COCO detection			COCO instance seg.		
	AP_{50}	AP	AP_{75}	AP^{mask}_{50}	AP^{mask}	AP^{mask}_{75}
scratch	44.0	26.4	27.8	46.9	29.3	30.8
ImageNet supervised	58.2	38.2	41.2	54.7	33.3	35.2
SimSiam (Chen et al., 2020a)	57.5	37.9	40.9	54.2	33.2	35.2
SimSiam+IDAA	**58.2**	**38.7**	**42.0**	**55.1**	**33.9**	**35.9**

Table 6. Semi-supervised learning performance of a WideResNet-28 trained on CIFAR100 with different amounts of data labeled.

Method	CIFAR100		
	400 labels	2500 labels	10000 labels
Fixmatch	47.76	66.30	74.13
Fixmatch+CLAE	50.34	68.58	74.54
Fixmatch+IDAA	**52.88**	**68.96**	**75.28**

in this setting. As shown in Table 3, IDAA consistently improves SimSiam by a large margin, e.g., 0.8% on top-1 accuracy after 200 epochs. Moreover, IDAA also outperforms data augmentation methods specifically designed for SSL, e.g., AdCo (Hu et al., 2021) and InfoMin (Tian et al., 2020a), where it outperforms Adco 2% on top-1 accuracy. IDAA also outperforms a hard sample mining method MoCHi (Kalantidis et al., 2020) while requiring smaller batch size and fewer epochs.

Transfer Learning Performance: To evaluate whether IDAA can be transferred to other unseen datasets to improve downstream tasks. We apply IDAA, CLAE and random augmentation to SimCLR respectively to train a model on ImageNet100 and evaluate the trained model on 8 other datasets. The linear evaluation accuracy are reported in Table 4. The improvement of IDAA can be transferred to the 8 datasets. As shown in the Table, IDAA consistently outperforms CLAE and improves the original SimCLR, while CLAE brings degeneration on one dataset, i.e., Pets. We also compare their transfer learning results when applied to SimSiam in Sec. C.1 of the Appendix. In Table 5, we compare the representation quality by transferring them to other tasks, i.e., COCO (Lin et al., 2014) object detection and instance segmentation. We can clearly see that IDAA can also improve detection and segmentation performance. Equipped with IDAA, SimSiam can surpass ImageNet supervised pre-training counterparts in all tasks, indicating that the key idea of "identity-preserving" can be applied to different tasks.

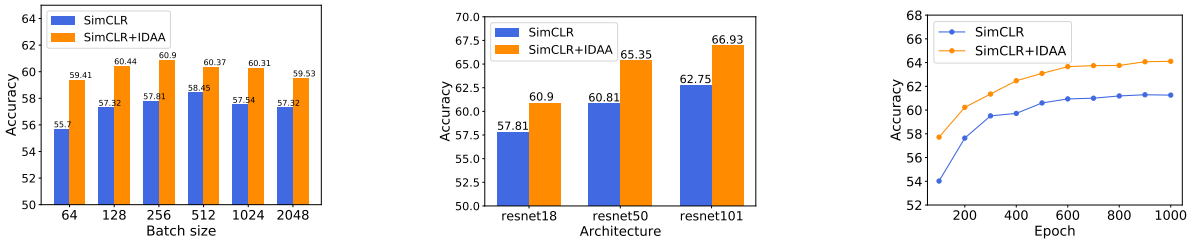
4.3. Semi-Supervised Learning

Data augmentation is also critical to state-of-the-art semi-supervised learning algorithms such as FixMatch (Sohn et al., 2020), which relies on accurate pseudo labeling and confidence-based data selection and their quality heavily depends on data augmentations. The test accuracy of the trained models are reported in Table 6, where IDAA consistently improves FixMatch's accuracy and the improvement is more significant with fewer labeled data available. Moreover, in Fig. 7 of the Appendix, IDAA significantly improves the efficiency and convergence of FixMatch and can save a great amount of computation to reach a reasonable accuracy.

4.4. Sensitivity Analysis of Hyperparameters

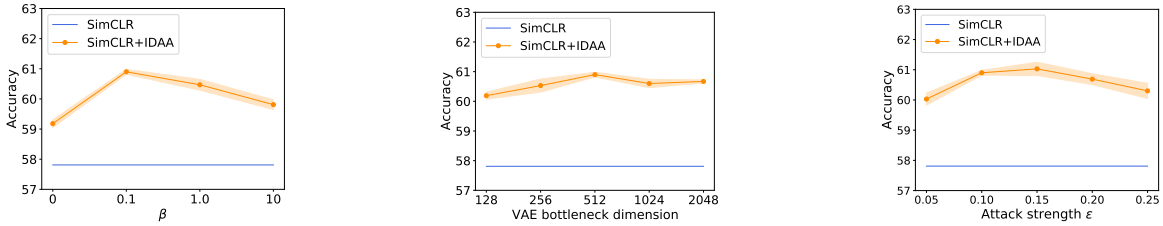
Batch Size: We evaluate how SimCLR and SimCLR+IDAA perform using six different batch sizes and the results are shown in Fig. 5(a). As verified by previous works (Chen et al., 2020a; He et al., 2020), increasing the batch size can improve SimCLR and other contrastive learning methods' performance, which is also reflected in our results. However, IDAA can significantly improve SimCLR's performance under small batch size, e.g., 64, because it effectively modifies various samples to be hard positives/negatives without distorting their original identities. Due to the same reason, SimCLR+IDAA is less sensitive to the change of batch size. This demonstrates the advantage of IDAA on improving the data/memory efficiency of SSL.

Model Architecture: We evaluate the linear evaluation performance of SSL when training different ResNet architectures in Fig. 5(b). Increasing the model size improves the



(a) (b) (c)

Figure 5. SSL performance under different (a) batch sizes, (b) ResNet architectures, and (c) training epochs.



(a) (b) (c)

Figure 6. SSL performance using different (a) β, (b) VAE bottleneck dimensions, and (c) Attack strength ϵ.

performance of both methods but SimCLR+IDAA always outperforms SimCLR by a large margin. Hence, IDAA can improve SSL of different models and it can train a smaller model with less costs to match the performance of training a larger model by random augmentations.

Training epochs: As shown in Fig. 5(a), the performance of both methods improves when investing more training epochs but the SimCLR+IDAA saturates much earlier and only spends 300 epochs to achieve comparable performance as SimCLR trained using 1000 epochs. Hence, IDAA can greatly improve the training efficiency of SSL.

Regularization weight β in VAE: As revealed by Theorem 3.5, β in the VAE objective controls the lower bound of the identity information preserved in IDAA augmentations: larger β enforces more identity preservation in x' and stronger identity-disentanglement on z. However, it also controls $I(z;x)$ which reflects the proximity of z's distribution to the true data manifold of x: large β leads to small $I(z;x)$ and less information of x preserved in z that can be leveraged to produce stronger adversarial attacks for hard positives/negatives. In Fig. 6(a), we observe that the trade-off reaches a sweet spot at $\beta = 0.1$ among all the four β values between 0 and 10. Sensitivity analysis of β on different datasets is given in Sec. C.2 of the Appendix.

VAE Bottleneck Dimension: As shown in Fig. 6(b), SSL performance with IDAA is not sensitive to the change of bottleneck dimension of VAE, though 512-dimension performs slightly better than other choices in the experiment.

Attack Strength ϵ: Stronger attacks may produce more hard positives/negatives but also increases the risk of identity distortion and unrealistic augmentations biased from the true data distribution/manifold. Results in Fig. 6(c) shows this trade-off and its effects on the SSL performance. Nevertheless, the performance of SimCLR+IDAA is still quite stable and only varies in a small range when changing ϵ since the identity-relevant information in $R(x)$ stays intact.

5. Conclusion

We propose a simple automatic data-augmentation method IDAA, which can generate more informative but identity-preserved augmentations to improve the efficiency and generalization of SSL. Motivated by an information theoretical analysis of VAE and identity preserving, IDAA adds adversarial noise to an identity-disentangled space learned by VAE and combines the perturbed VAE outputs with an intact identity-relevant part to produce augmentations. IDAA merely relies on a pre-trained VAE without requiring any labeled data but consistently improves a diverse set of popular self-supervised/semi-supervised learning methods across multiple benchmarks. It also enhances the transfer learning performance and improves the learning efficiency.

Acknowledgements

This work was supported in part by NSFC No. 61872329 and the Fundamental Research Funds for the Central Universitie sunder contract WK3490000005. We would like to thank ICML area chairs and reviewers for their efforts in reviewing this paper and their constructive comments! We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Berg, T., Liu, J., Lee, S. W., Alexander, M. L., Jacobs, D. W., and Belhumeur, P. N. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3): 1–37, 2011.

Cao, Y.-T., Wang, J., and Tao, D. Symbiotic adversarial learning for attribute-based person search. In *European Conference on Computer Vision*, pp. 230–247. Springer, 2020.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149, 2018.

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

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

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

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

Chuang, C.-Y., Robinson, J., Yen-Chen, L., Torralba, A., and Jegelka, S. Debiased contrastive learning. *arXiv preprint arXiv:2007.00224*, 2020.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , and Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Dai, B., Wang, Y., Aston, J., Hua, G., and Wipf, D. Connections with robust pca and the role of emergent sparsity in variational autoencoder models. *The Journal of Machine Learning Research*, 19(1):1573–1614, 2018.

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

Ebrahimi, S., Meier, F., Calandra, R., Darrell, T., and Rohrbach, M. Adversarial continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 386–402. Springer, 2020.

Fu, H., Gong, M., Wang, C., Batmanghelich, K., Zhang, K., and Tao, D. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2427–2436, 2019.

Fu, S., He, F., Liu, Y., Shen, L., and Tao, D. Robust unlearnable examples: Protecting data against adversarial learning. *arXiv preprint arXiv:2203.14533*, 2022.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014a.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014b.

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

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, pp. 9729–9738, 2020.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Ho, C.-H. and Vasconcelos, N. Contrastive learning with adversarial examples. *arXiv preprint arXiv:2010.12050*, 2020.

Hu, Q., Wang, X., Hu, W., and Qi, G.-J. Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1074–1083, 2021.

Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 172–189, 2018.

Ilyas, A., Santurkar, S., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Kalantidis, Y., Sariyildiz, M. B., Pion, N., Weinzaepfel, P., and Larlus, D. Hard negative mixing for contrastive learning. *arXiv preprint arXiv:2010.01028*, 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Liu, T., Fan, J., Luo, Y., Tang, N., Li, G., and Du, X. Adaptive data augmentation for supervised learning over missing data. *Proceedings of the VLDB Endowment*, 14(7): 1202–1214, 2021.

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

Misra, I. and Maaten, L. v. d. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020.

Nilsback, M.-E. and Zisserman, A. A visual vocabulary for flower classification. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1447–1454. IEEE, 2006.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pp. 69–84. Springer, 2016.

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

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

Robinson, J., Chuang, C.-Y., Sra, S., and Jegelka, S. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 776–794. Springer, 2020a.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020b.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

Tu, Z., Zhang, J., and Tao, D. Theoretical analysis of adversarial learning: A minimax approach. *Advances in Neural Information Processing Systems*, 32, 2019.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29:3630–3638, 2016.

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

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A. L., and Le, Q. V. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 819–828, 2020.

Yang, K., Zhou, T., Zhang, Y., Tian, X., and Tao, D. Class-disentanglement and applications in adversarial detection and defense. *Advances in Neural Information Processing Systems*, 34:16051–16063, 2021.

Ye, M., Zhang, X., Yuen, P. C., and Chang, S.-F. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6210–6219, 2019.

Zhang, Y., Tian, X., Li, Y., Wang, X., and Tao, D. Principal component adversarial example. *IEEE Transactions on Image Processing*, 29:4804–4815, 2020.

Zhou, T. and Tao, D. Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 2011.

Zhou, T., Wang, S., and Bilmes, J. Time-consistent self-supervision for semi-supervised learning. In *International Conference on Machine Learning*, pp. 11523–11533. PMLR, 2020.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

Zhuang, C., Zhai, A. L., and Yamins, D. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6002–6012, 2019.

A. Proof

A.1. Proof for Proposition 3.1

Proposition A.1. (Sample-identification likelihood as a lower bound of $I(x;y)$). *If $\vec{x}$ is a random mini-batch of size N and the sample-identification likelihood of x_i on its correct identification label $y=i$ to be $p(y=i|x=x_i)$, the mutual information $I(x;y)$ can be lower bounded by*

$$I(x;y) \geq \log N + \mathbb{E}_{\vec{x}}\left[\frac{1}{N}\sum_{i=1}^{N}\log p(y=i|x=x_i)\right]. \tag{10}$$

Proof. Here we consider such data distribution $p(x,y) = E_{\vec{x}}[p(x,y|\vec{x})]$ where $\vec{x} = \{x_1, x_2, \cdots, x_N\}$ denotes a batch of size N. Then we have:

$$\begin{aligned}
I(x,y) &= H(y) - H(y|x) \\
&= \log N + \sum_{y\in\mathcal{Y}}\int_{x\in\mathcal{X}} p(x,y)\log p(y|x)\mathrm{dx} \\
&= \log N + \sum_{y\in\mathcal{Y}}\int_{x\in\mathcal{X}} \mathbb{E}_{\vec{x}}\left[p(x,y|\vec{x})\right]\log p(y|x)\mathrm{dx} \\
&= \log N + \mathbb{E}_{\vec{x}}\left[\sum_{y=1}^{N}\int_{x\in\vec{x}} p(x,y)\log p(y|x)\mathrm{dx}\right] \\
&= \log N + \mathbb{E}_{\vec{x}}\left[\sum_{y=1}^{N}\int_{x\in\vec{x}} p(x)p(y|x)\log p(y|x)\mathrm{dx}\right] \\
&\geq \log N + \mathbb{E}_{\vec{x}}\left[\sum_{y=1}^{N}\int_{x\in\vec{x}} p(x)p(y|x)\log q(y|x)\mathrm{dx}\right] \\
&= \log N + \mathbb{E}_{\vec{x}}\left[\sum_{y=1}^{N}\int_{x\in\vec{x}} p(x,y)\log q(y|x)\mathrm{dx}\right] \\
&= \log N + \mathbb{E}_{\vec{x}}\left[\sum_{y=1}^{N}\int_{x\in\vec{x}} p(x|y)p(y)\log q(y|x)\mathrm{dx}\right] \\
&= \log N + \mathbb{E}_{\vec{x}}\left[\sum_{i=1}^{N}\int_{x\in\vec{x}} p(x|y=i)p(y=i)\log q(y=i|x)\mathrm{dx}\right] \\
&= \log N + \mathbb{E}_{\vec{x}}\left[\sum_{i=1}^{N}\int_{x\in\vec{x}} \delta(x-x_i)\frac{1}{N}\log q(y=i|x)\mathrm{dx}\right] \\
&= \log N + \mathbb{E}_{\vec{x}}\left[\sum_{i=1}^{N}\frac{1}{N}\log q(y=i|x=x_i)\right]
\end{aligned} \tag{11}$$

where $p(\cdot)$ denotes the true probability and $q(\cdot)$ denotes arbitrary estimation of $p(\cdot)$. The fourth equality in Equation (11) holds true using Fubini's theorem (switch the order of integration). The inequality in Equation (11) comes from $\mathrm{D}_{\mathrm{KL}}(p(y|x), q(y|x)) \geq 0$. □

A.2. Proof for Theorem 3.5

We start by proving Lemma 3.2, Lemma 3.3 and then we can prove Theorem 3.5 by combing Lemma 3.2, Lemma 3.3 and Assumption 3.4.

Lemma A.2. (VAE objective and $I(z;y)$ from Eq. (29) in Appendix B of (Alemi et al., 2016)). *Assume that the bottleneck features of VAE are denoted by z, the encoder is $E(\cdot)$ and produces distribution $p_E(z|x)$, the decoder is $D(\cdot)$ and produces distribution $q_D(x|z)$, the prior for z is $p(z)$, and the KL-divergence regularization in the VAE objective L_{VAE} has a weight*

β, we have:

$$-I(z;x)+\beta I(z;y)\leq L_{\mathrm{VAE}}, \tag{12}$$

$$L_{\mathrm{VAE}}\triangleq -\int \mathrm{d}x p(x)\int \mathrm{d}z p_E(z|x)\log q_D(x|z)+\beta\frac{1}{N}\sum_{i=1}^{N}\mathrm{D}_{\mathrm{KL}}\left(p_E\left(z|x=x_i\right)||p(z)\right), \tag{13}$$

The proof can be found in Appendix B of (Alemi et al., 2016).

Lemma A.3. (Identity-disentangled data generation). *For a data generative model described above,*

$$I(R(x);y)\geq I(x;y)-I(z;y). \tag{14}$$

Proof. Due to the Markov chain $y\to(R(x),z)\to x$ in the data generative model in Fig. 3, we have

$$I(x;y)\leq I(R(x),z;y)=I(R(x);y)+I(z;y|R(x)). \tag{15}$$

By the definition of conditional mutual information, we have

$$I(z;y|R(x))=I(z;y)+\left[H(R(x)|z)-H(R(x))\right]+\left[H(R(x)|y)-H(R(x)|y,z)\right]. \tag{16}$$

Since $z\perp\!\!\!\perp R(x)$ in the generative model, the last two terms in the above equation are zeros and $I(z;y|R(x))=I(z;y)$. Substituting it to Eq. (15) completes the proof. □

Assumption A.4. (Identifiability extended from Theorem 1 of Robust PCA (Candès et al., 2011; Zhou & Tao, 2011)). There exists a small $\epsilon>0$, when perturbing z within the ϵ-ball, the identity-disentangled part $D(z)$ and identity-relevant part $R(x)$ are still separable using VAE.

The identifiability assumption is a mild and reasonable assumption because VAE is known as an extension of Robust PCA (Candès et al., 2011), as pointed out by (Dai et al., 2018). Similar assumption can be found in Proposition 1 of (Huang et al., 2018), which assumes the reversibility of Autoencoders.

Theorem A.5. (Identity-disentangled data augmentation). *If we use a VAE in the identity-disentangled data generative model for Lemma 3.3, and if we define an augmentation $x'=R(x)+G'(x)$ with $G'(x)\sim q_D(x|z')$ and $z'=z+\delta$ (a δ-perturbed z)), there exists a small $\epsilon>0$ such that for any $\|\delta\|_p\leq\epsilon$ we can lower bound $I(x';y)$ as*

$$I(x';y)\geq I(x;y)-\frac{1}{\beta}(L_{\mathrm{VAE}}+I(z;x)). \tag{17}$$

Proof. Applying the results from Assumption 3.4, Lemma 3.3, and Lemma 3.2, for any $\|\delta\|_p\leq\epsilon$, we have

$$\begin{aligned} I(R(x)+D'(z+\delta);y) &= I(R(x)+G'(x);y) && (18)\\ &\geq I(R(x);y) && (19)\\ &\geq I(x;y)-I(z;y) && (20)\\ &\geq I(x;y)-\frac{1}{\beta}(L_{\mathrm{VAE}}+I(z;x)). && (21)\end{aligned}$$

The first inequality comes from Assumption 3.4: we have the conditional entropy $H(R(x)|R(x)+G'(x))=0$ because $R(x)+G'(x)$ can be separated into $R(x)$ and $G'(x)$ again in an unique and exact way using VAE and thus $I(R(x)+G'(x);y)\geq I(R(x);y)$. □

B. Experimental Implementation and Reproduction Details

All code are implemented with Pytorch (Paszke et al., 2019). All CIFAR (ImageNet) experiments are conducted on NVIDIA V100 (A100) GPU. The pre-trained VAE uses a standard VAE architecture (Kingma & Welling, 2013) with 512 (3072) bottleneck dimension for CIFAR (ImageNet). Default β in Eq. (6) is set to be 0.1 and default ϵ in Eq. (9) is set to be 0.15. When applying IDAA to a SSL method, IDAA uses that SSL method's default random augmentation, which is a standard setting used by all current data augmentation methods for SSL (Ho & Vasconcelos, 2020; Tian et al., 2020a; Hu et al., 2021; Kalantidis et al., 2020; Chuang et al., 2020). To train the backbone with IDAA augmentation, we apply seperate batch norm layer (BN) on ResNet architecture, i.e., adversarial data and normal data use different BN. Please refer to AdvProp (Xie et al., 2020) for more implementation details. In the main results, a VAE is trained from scratch on the same dataset as self-supervised learning.

Table 7. Comparison of CLAE's and our reproduction results of SimCLR. All methods train a ResNet-18 for 300 epochs.

Method	Linear Evaluation	
	CIFAR10	CIFAR100
SimCLR (CLAE (Chen et al., 2020a)'s results)	83.27±0.17	53.79±0.21
SimCLR+CLAE (CLAE (Chen et al., 2020a)'s results)	83.32±0.26	55.52±0.30
SimCLR (Our reproduction)	86.40±0.18	57.81±0.23
SimCLR+CLAE (Our reproduction)	85.25±0.07	57.69±0.25

Table 8. Comparison of the original reported and our reproduction results of SimSiam. All methods train a ResNet-50 with batch size 256.

Method	Epoch	Linear Evaluation
SimSiam (Reported results (Chen et al., 2020a))	100	68.1
SimSiam (Our reproduction)	100	68.1
SimSiam (Reported results (Chen et al., 2020a))	200	70.0
SimSiam (Our reproduction)	200	69.8

To evaluate self-supervised learning methods, k nearest neighbor (kNN) and linear evaluation are considered. For kNN, the evaluation is identical to the protocal used in (Ho & Vasconcelos, 2020), where k is set to be 200. For linear evaluation, we train a single linear layer on the embedding extracted from the fixed backbone, as in (Chen et al., 2020a).

Case Study of Augmentations generated by IDAA and CLAE: The experiments in Fig. 1 are conducted on CIFAR10 and the model used to compute the distance is trained using the original SimCLR before convergence because we aim at simulating the intermediate stage of self-supervised learning when the model does not fully converge and when high-quality data augmentations with identity preserved are critical to the future training (while poor augmentations with identity distorted are detrimental to the future training). For random augmentations, we choose *RandomFlip, ColorJitter and GreyScale* for their popularity in recent SSL methods (Ye et al., 2019; Chen et al., 2020a; Chen & He, 2021). The distance is measured in feature space where contrastive loss is applied. The "identity-preserving boundary" in Fig. 1 is defined by those points whose distances to positive equals to that to their nearest negative.

Comparison with Random and Adversarial Augmentation: Here we mainly follow the setting of CLAE (Ho & Vasconcelos, 2020) to evaluate data augmentations on four self-supervised learning methods: Plain (InfoNCE in Eq. (1)), UEL (Ye et al., 2019), SimSiam (Chen & He, 2021) and SimCLR (Chen et al., 2020a) on three datasets, i.e., CIFAR10 (Krizhevsky et al., 2009), CIFAR100 (Krizhevsky et al., 2009), and miniImageNet (Vinyals et al., 2016). The four SSL methods cover different contrastive learning methods (Plain, UEL, SimCLR) and a consistency regularization based method (SimSiam). The training/test splitting of miniImageNet follows (Ebrahimi et al., 2020). Follow CLAE's setting, We train a ResNet-18 for 300 (100) epochs and linear evaluation model for (1000) 200 epochs with batch size 256 (128) for CIFAR (miniImageNet). We use CLAE's official code[1] to implement plain, UEL, SimCLR and CLAE. We further implement SimSiam as an additional baseline using their official code[2]. It worth noting that CLAE reported a much lower accuracy on SimCLR than the one reported in the original SimCLR (Chen et al., 2020a) paper and the one we achieved on SimCLR. This is shown in the Table 7, in which we provide a side-by-side comparison with the results from CLAE's paper and we can clearly see that our reproduction of SimCLR is much higher than that of CLAE's. We posit the reason is that CLAE did not use a decaying learning rate as instructed in SimCLR paper, which uses a decaying cosine learning rate (and we use it too).

Comparison with Hard/Effective View Generation Methods: Here we compare IDAA with a hard sample mining method, i.e., HCL (Robinson et al., 2020) and a effective view generation method, i.e., Debiased (Chuang et al., 2020). We implemented these two methods using HCL's official code[3] and apply IDAA to them. We follow their original setting to train a ResNet-50 for 400 epochs and the linear evaluation layer for 100 epochs with batch size 256 on CIFAR10/CIFAR100.

ImageNet Experiments: Here we evaluate IDAA's performance on ImageNet, which contains 1.28M images in the training set and 50K images in the validation set from 1000 classes. We select SimSiam (Chen et al., 2020a) as a baseline due to its

[1]https://github.com/chihhuiho/CLAE
[2]https://github.com/facebookresearch/simsiam
[3]https://github.com/joshr17/HCL

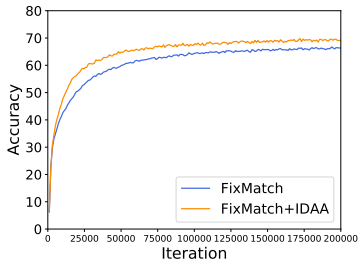


Figure 7. Convergence curve under 2500 labels on CIFAR100 for semi-supervised learning.

Table 9. Transfer learning performance (in test accuracy %) on 8 datasets for CLAE and IDAA applied to SimSiam (pre-trained on ImageNet100).

	CIFAR10	CIFAR100	Birdsnap	Aircraft	DTD	Pets	Flower	CUB-200
SimSiam	49.75	25.59	7.32	16.29	40.07	31.33	58.20	9.96
SimSiam+CLAE	53.05	26.78	8.80	20.01	43.97	33.62	63.54	12.39
SimSiam+IDAA	**55.74**	**29.63**	**9.16**	**20.16**	**47.32**	**36.22**	**65.63**	**12.84**

effectiveness and efficiency. We reproduce SimSiam using their official code[2] and apply IDAA to it. We follow its original setting to train a ResNet-50 for 100 and 200 epochs with batch size 256 and linear evaluation layer for 90 epochs. We compare our reproduction results and the original reported results in Table 8, where we can see that our reproduction is quite close to the original reported results.

Transfer Learning: Here we follow the setting of CLAE (Ho & Vasconcelos, 2020) to train a ResNet-18 for 100 epochs on ImageNet-100 by applying default random augmentation, IDAA and CLAE to SimCLR respectively, and then train a linear layer using the embedding outputed by the backbone network on other 8 datasets (Krizhevsky et al., 2009; Berg et al., 2014; Maji et al., 2013; Cimpoi et al., 2014; Parkhi et al., 2012; Nilsback & Zisserman, 2006; Welinder et al., 2010). For the detection and instance segmentation experiments in Table 5, we follow the setting of (Chen et al., 2020a) to use Mask R-CNN (He et al., 2017) (1× schedule) with the C4-backbone.

Semi-Supervised Learning: Here we reproduce one state-of-the-art semi-supervised learning method, i.e., FixMatch (Sohn et al., 2020), then we apply IDAA and CLAE to FixMatch to train a a WideResNet-28-8 model on CIFAR100 with only $\{400, 2500, 10000\}$ labeled samples and the rest unlabeled. We train all the methods for 2×10^5 steps with batch size 512.

Sensitivity Analysis of Hyperparameters: The experiments are conducted on CIFAR100 with SimCLR.

C. Additional Experiments

C.1. Transfer learning performance using SimSiam as baseline

We further add new experiments of SimSiam, SimSiam+CLAE, and SimSiam+IDAA, and report their results in Table 9 as an extension of Table 4. Unlike SimCLR, SimSiam (Chen & He, 2021) adopts another popular idea of self-supervised learning based on the consistency regularization and Siamese network.

We evaluate the three methods on ImageNet-100 as we did for SimCLR, i.e., by running each method for 100 epochs and evaluating their transfer learning performance on the 8 datasets as in Table 4. Similar to SimCLR, IDAA consistenly improves the transfer learning performance of SimSiam and outperform CLAE on all the datasets.

We select SimCLR and SimSiam for this study because they represent the two most widely studied self-supervised learning strategies, i.e., contrastive learning and consistency learning respectively.

C.2. Sensitivity Analysis of VAE hyperparameter β on Multiple Datasets

Below we provide a thorough ablation study regarding β on all the three datasets. Specifically, we tried four values of β, i.e., $0, 0.1, 1, 10$, in SimCLR+IDAA on three datasets and report the results in the Table 10. The performance of our method is robust to the change of β and keeps surpassing all the baselines.

Table 10. Linear Evaluation (top-1) accuracy regarding different values of β on three datasets. All the methods train a ResNet-18 for 300 (100) epochs for CIFAR (miniImageNet).

β	CIFAR10	CIFAR100	miniImageNet
0	86.79	59.18	48.19
0.1	88.07	60.90	48.23
1	87.75	60.47	48.44
10	87.64	59.81	48.51

Table 11. Computation cost comparison of SimCLR and "SimCLR+IDAA" by training a ResNet-18 on CIFAR100.

	SimCLR		SimCLR+IDAA	
Epoch	Training time (s)	Linear Evaluation (%)	Training time (s)	Linear Evaluation (%)
100	3.54×10^3	54.02	6.76×10^3	57.72
200	7.08×10^3	57.64	1.35×10^4	60.23
300	3.54×10^4	59.51	2.02×10^4	61.35
400	1.42×10^4	59.72	2.70×10^4	62.47
500	1.77×10^4	60.60	3.38×10^4	63.09
600	2.12×10^4	60.94	4.06×10^4	63.66
700	2.48×10^4	61.61	4.73×10^4	63.74
800	2.83×10^4	61.19	5.41×10^4	63.76
900	3.19×10^4	61.29	6.10×10^4	64.07
1000	3.54×10^4	61.26	6.76×10^4	64.11

The best value of β for both CIFAR10 and CIFAR100 is 0.1, while that for miniImageNet is 10. This is because the images in miniImageNet are of higher resolution and contain richer identity-information than CIFAR, hence a larger β is needed to enforce a stronger identity-disentanglement.

The performance of IDAA is robust to the choice of β, e.g., the maximal difference on accuracy is merely $< 0.5\%$ between two choices of β in the miniImageNet experiments. Therefore, SimCLR+IDAA consistently outperforms SimCLR for all the evaluated values of β.

In the main paper, we did not tune the value of β for each dataset separately but instead use the same $\beta = 0.1$ for all the datasets. As shown in Table 10, we can achieve better results than the previous results in our paper if we carefully tune β for every dataset.

C.3. Computational Cost

Although IDAA requires more computation per epoch caused by the extra inference and adversarial perturbation on VAE, it produces more informative augmentations (more challenging but identity-preserved) that can significantly reduce the number of training epochs needed to reach the same accuracy. In practice, it can effectively reduce the overall training time.

For example, in our experiments on CIFAR100 with ResNet-18, the (averaged) training time per epoch on CIFAR100 (batch size=256) is 67.6s (seconds) for "SimCLR+IDAA" and 35.4s for vanilla SimCLR. However, as reported in the Table 11, IDAA greatly saves the total training time to reach a similar performance. For example, to reach $> 60\%$ accuracy, SimCLR takes 1.77×10^4s while IDAA taks only 1.35×10^4s and thus saves over 4×10^3s training time.

C.4. New Baseline: Data Augmentation without Decomposition ($G(z')$ only)

Here we compare with an additional baseline: simply attacking the z of a standard VAE (with beta=1) to produce an $x' = G(z')$ as augmentations for contrastive learning. We report its results on CIFAR in Table 12, denoted as "SimCLR+IDAA(w/o decomposition)". It shows that the performance significantly declines once the decomposition removed and it performs even worse than the original SimCLR.

Therefore, identity-disentanglement is critical and preserving the identity-relevant part $R(x)$ in augmentations is essential to self-supervised learning. This is illustrated in Figure 1. On the contrary, if we simply use the identity-distorted part $G(z')$

Table 12. Test accuracy on downstream classification tasks: comparing representations learned with ($x' = G(z') + R(x)$) and without identity-disentanglement decomposition ($x' = G(z')$).

Method	kNN		Linear Evaluation	
	CIFAR10	CIFAR100	CIFAR10	CIFAR100
SimCLR	80.79	41.11	86.40	57.81
SimCLR+IDAA(w/o decomposition)	67.25	32.77	75.82	47.41
SimCLR+IDAA	**83.41**	**46.78**	**88.07**	**60.90**

for augmentation with $R(x)$ removed, the positive and negative assignments in contrastive learning can be wrong. With many wrong identity labels, the identification task of contrastive learning can easily fail, resulting in poor representations and performance degradation on downstream tasks.

D. Additional Discussion

D.1. On the Definition of "Identity-Preserving"

Theoretically, an augmentation x' of a sample x preserves x's identity y if $I(x'; y) \geq I(x; y) - \epsilon$ holds with a small ϵ, e.g., Eq. (8) for IDAA. Empirically, "identity-preserving" refers to that a sample is closer to positive(s) than to its nearest negative in contrastive learning.

"Identity-preserving" is defined from the perspective of neural nets. Although augmentations including CLAE and IDAA are limited to make too much change to the original samples (ϵ-ball constraint for adversarial perturbations in CLAE and IDAA), it is hard for human to notice whether the sample identity is changed or not for the neural nets. This is similar to adversarial attacks (Fu et al., 2022; Zhang et al., 2020; Cao et al., 2020; Fu et al., 2019; Tu et al., 2019): they change a neural network's predictions of images using perturbations that are too small to be noticed by human eyes, because neural nets usually rely on very sparse patterns to make predictions, as discussed in (Ilyas et al., 2019). Thus, as shown in Fig. 1 (b)-(c), many CLAE augmentations cannot preserve the original identity since they produce negatives closer to the anchor than the positives.

D.2. Using Other Generative Model

From both theoretical and empirical intuitions, we believe that VAE is a simpler but better choice than other deep generative models. The main reason is that our objective here is not reconstruction/generation but identity-disentanglement and VAE serves best for this purpose:

- VAE provides tighter bound for identity-disentanglement in Lemma 3.2. For example, Wassertein Autoencoder (WAE) (Tolstikhin et al., 2018) matches the marginal distribution of latent factor $p_E(z)$ to the prior $p(z)$, while VAE matches the conditional distribution $p_E(z|x)$ to prior $p(z)$. By the convexity of KL divergence, we have $I(z; y) \leq \mathrm{E}_x[\mathrm{D}_{\mathrm{KL}}(p_E(z|x)\|p(z))] \leq \mathrm{D}_{\mathrm{KL}}(\mathrm{E}_x[p_E(z|x)]\|p(z)) = \mathrm{D}_{\mathrm{KL}}(p_E(z)\|p(z))$. Hence, comparing to WAE, VAE optimizes a tighter bound for the identity-disentanglement measured by $I(z; y)$.
- Empirically, other deep generative model like WAE or GAN (Goodfellow et al., 2014a) may generate higher-quality reconstruction $G(x)$ than VAE so their residual $R(x) = x - G(x)$ tends to preserve less identity information, which may lead to more identity distortion in generating the augmentations and hence performance degradation.

D.3. Extending IDAA to other Data Modality

The proposed scheme and its theoretical insights of IDAA are principal and can be extended to other data domains: the VAE based identity-disentanglement and the identity-preserved adversarial augmentation can be directly applied to other domains such as time series/texts. One of the most challenging parts when extending to other modalities is choosing the proper space to conduct identity-disentanglement. For example, on NLP data, we may perform identity-disentanglement in an embedding space instead of the raw discrete input space and choose transformer as the architecture for VAE's encoder and decoder.