Adversarial Auto-Augment with Label Preservation: A Representation Learning Principle Guided Approach

Kaiwen Yang[1] **Yanchao Sun**[2] **Jiahao Su**[2] **Fengxiang He**[3]
Xinmei Tian[1] **Furong Huang**[2] **Tianyi Zhou**[2] **Dacheng Tao**[3]
University of Science and Technology of China[1]
University of Maryland, College Park[2]; JD Explore Academy[3]
kwyang@mail.ustc.edu.cn, xinmei@ustc.edu.cn,
{ycs, jiahaosu, furongh}@umd.edu
{fengxiang.f.he, tianyi.david.zhou, dacheng.tao}@gmail.com

Abstract

Data augmentation is a critical contributing factor to the success of deep learning but heavily relies on prior domain knowledge which is not always available. Recent works on automatic data augmentation learn a policy to form a sequence of augmentation operations, which are still pre-defined and restricted to limited options. In this paper, we show that a prior-free autonomous data augmentation's objective can be derived from a representation learning principle that aims to preserve the minimum sufficient information of the labels. Given an example, the objective aims at creating a distant "hard positive example" as the augmentation, while still preserving the original label. We then propose a practical surrogate to the objective that can be optimized efficiently and integrated seamlessly into existing methods for a broad class of machine learning tasks, e.g., supervised, semi-supervised, and noisy-label learning. Unlike previous works, our method does not require training an extra generative model but instead leverages the intermediate layer representations of the end-task model for generating data augmentations. In experiments, we show that our method consistently brings non-trivial improvements to the three aforementioned learning tasks from both efficiency and final performance, either or not combined with pre-defined augmentations, e.g., on medical images when domain knowledge is unavailable and the existing augmentation techniques perform poorly. Code is available at: https://github.com/kai-wen-yang/LPA3.

1 Introduction

Data augmentation has emerged as an effective data pre-processing or data transformation step to mitigate overfitting [31], to encourage local smoothness [57], and to improve generalization [6] in machine learning pipelines such as deep neural networks. Notably, effective data augmentation, which incorporates class-related data invariance and enriches the in-class sample, is one of the key contributing factors for representation learning with weak or self supervision [9, 23].

Given a task, we aim to generate "good" augmentations efficiently. As part of the machine learning model pipeline, an *autonomous domain-agnostic* but *task-informed* data augmentation mechanism is desirable. However, a number of challenges exist. **(1)** Existing augmentation operators are usually hand-crafted based on domain expert knowledge, which is not always available in some domain [49]. For example, widely used augmentations for natural images are not effective on medical images. Moreover, the performance of those machine learning pipelines drastically varies with different choices of data augmentations. **(2)** Existing few autonomous augmentation approaches developed lately are neither fully autonomous nor universally applicable to varying domains. Although a

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

few autonomous data augmentation approaches have been developed in recent years [14, 12], they train policies to produce a sequence of pre-defined augmentation operations and thus are not fully automated and are limited to a few domains. **(3)** Existing augmentations usually do not fully utilize the task feedback (i.e., task-agnostic) and may be sub-optimal for the targeted task. A class of automated data augmentation methods train an extra data generative model to generate new augmentations from scratch given a real-world example [2]. However, they require training a generative model, which is a non-trivial task in practice that may either relies on strong prior knowledge or a substantial increased number of training examples.

In this paper, we first investigate the conditions required to generate domain-agnostic but task-informed data augmentations. Consider a representation learning pipeline, we started from a probabilistic graphical model that describes the relations among the label $\mathbf{Y}$, the nuisance $\mathbf{N}$, the example $\mathbf{X}$, its augmentation $\mathbf{X}'$, and the latent representations $\mathbf{Z}$. We argue that a minimum-sufficient representation for the task preserves the label information but excludes other distractive information from the nuisance. We then investigate the conditions for an augmentation $\mathbf{X}'$ that results in learning such preferred representations. These conditions motivate an optimization objective that can be used to produce automated domain-agnostic but task-informed data augmentations for each example, without replying on pre-defined augmentation operators or specific domain knowledge. Consequently, our proposed optimization objective addresses all aforementioned challenges.

For practicality, we further propose a surrogate of the derived objective that can be efficiently computed from the intermediate-layer representations of the model-in-training. The surrogate is built upon the data likelihood estimation through perceptual distance [24] defined on the intermediate layers' representations. Specifically, our proposed surrogate objective maximizes the perceptual distance between $\mathbf{X}$ and $\mathbf{X}'$, under a label preserving constraint on the model prediction of $\mathbf{X}'$. This problem can be efficiently solved by optimizing its Lagaragian relaxation. Thereby, given $\mathbf{X}$ and its label $\mathbf{Y}$, the solution to our surrogate objective generates "hard positive examples" for $\mathbf{X}$ without loosing its label information. Once generated, $\mathbf{X}'$ is used to train the model towards producing the minimum-sufficient representation $\mathbf{Z}$ for the targeted task. Our proposed method, named *Label-Preserving Adversarial Auto-Augment (LP-A3)*, does not require any extra generative models such as Generative Adversarial Networks, unlike previous automated augmentation methods [40]. We further propose a sharpness-aware criterion selecting only the most informative examples to apply our auto-augmentation on so it does not cause expensive extra computation.

Our proposed LP-A3 is a general and autonomous data augmentation technique applicable to a variety of machine learning tasks, such as supervised, semi-supervised and noisy-label learning. Moreover, we demonstrate that it can be seamlessly integrated with existing algorithms for these tasks and consistently improve their performance. In experiments on the three learning tasks, we equip LP-A3 with existing methods and obtain significant improvement on both the learning efficiency and the final performance. The generated augmentations are optimized for the model-in-training in a target-task-aware manner and thus notably accelerate the slow convergence in computationally intensive tasks such as semi-supervised learning. It is worth noting that our augmentation can consistently bring improvement to tasks without domain knowledge or pre-defined augmentations such as medical image classification, on which the traditional image augmentations lead to performance degeneration.

2 Related work

Hand-crafted vs. Autonomous Data Augmentations. Most of the existing widely used data augmentations are hand-crafted based on domain expert knowledge [30, 39, 29, 9, 13]. For example, MoCo [16] and InstDis [47] create augmentations by applying a stochastic but pre-defined data augmentation function to the input. CMC [39] splits images across color channels. PIRL [29] generates data augmentations through random JigSaw Shuffling. CPC [30] renders strong data augmentations by utilizing RandAugment [12], which learns a policy producing a sequence of pre-defined augmentation operations selected from a pool [14]. AdvAA [60] designs a adversarial objective to learn the augmentation policy. [12, 14, 60] are all based on pre-defined operations which is not available in certain domains, and their objective cannot guarantee the label-preserving of the generated data which may lead to suboptimal performance. "InfoMin" principle of data augmentation is proposed [40] to minimize the mutual information between different views (equivalent to $\min I(\mathbf{X}, \mathbf{X}')$). However, their theory depends on access to a minimal sufficient encoder which may be difficult to obtain. In contrast, we not only consider how to generate optimal views or augmentations, but also consider

generating the minimal sufficient representation. The algorithm [40] deploys a generator to render augmentation (which may be costly to train especially on non-natural image domains), while we directly learn the augmentation through gradient descent w.r.t. the input.

Information Theory for Representation Learning. Information theory is introduced in deep learning to measure the quality of representations [42, 1]. The key idea is to use information bottleneck methods [41, 42] to encourage the learned representation being minimal sufficient. Mutual information objectives are commonly used in self-supervised learning. For example, InfoMax principle [27] used by many works aims to maximize the mutual information between the representation and the input [39, 4, 46]. But simply maximizing the mutual information does not always lead to a better representation in practice [43]. In contrast, InfoMin principle [40] minimizes the mutual information between different views. Both InfoMax and the InfoMin principles can be associated with our proposed representation learning criteria in Section 4, as they lead to sufficiency and minimality of the learned representation, respectively.

Augmentation in Self-supervised Contrastive Learning. Self-supervised Contrastive Representation Learning [30, 18, 47, 39, 35, 9] learn representation through optimization of a contrastive loss which pulls similar pairs of examples closer while pushing dissimilar example pairs apart. Creating multiple views of each example is crucial for the success of self-supervised contrastive learning. However, most of the data augmentation methods used in generating views, although sophisticated, are hand-crafted or not learning-based. Some use luminance and chrominance decomposition [39], while others use random augmentation from a pool of augmentation operators [47, 9, 4, 16, 53, 37, 62, 65]. Recently, adversarial perturbation based augmentation has been proposed to generate more challenging positives/negatives for contrastive learning [51, 19].

Augmentation in Semi-supervised Learning. Data augmentation plays an important role in semi-supervised learning, e.g., (1) consistency regularization [33, 34] enforces the model to produce similar outputs for a sample and its augmentations; (2) pseudo labeling [25] trains a model using confident predictions produced by itself [38] for unlabeled data. Data augmentations are critical [23, 7] because they determine both the output targets and input signals: (1) accurate pseudo labels are achieved by averaging the predictions over multiple augmentations; (2) weak augmentations (e.g., flip-and-shift) are important to produce confident pseudo labels, while strong augmentations [14, 12]) are used to train the model and expand the confidence regions (so more confident pseudo labels can be collected later). Data selection [55, 63] for high-quality pseudo labels is also critical and its criterion is estimated on augmentations, e.g., the confidence [8] or time-consistency [63] of each sample.

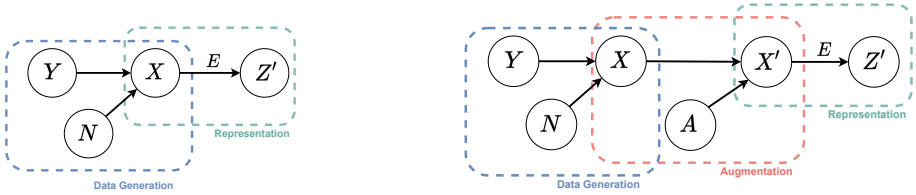
Augmentation in Noisy-label Learning Two primary challenges in noisy-label learning is clean label detection [28, 15, 20] and noisy label correction by pseudo labels [32, 3, 26]. Both significantly depend on the choices of data augmentations since the former usually relies on confidence thresholding and augmentations help rule out the overconfident samples, while the latter relies on the quality of semi-supervised learning. Moreover, as shown in previous works [26, 64], removing strong augmentations such as RandAugment can considerably degenerate the noisy label learning performance.

3 Preliminaries

Basics of Information Theory Our analyses make frequent use of information theoretical quantities [11]. Given a joint distribution $P_{\mathbf{X},\mathbf{Y}}$ and its marginal distributions $P_{\mathbf{X}}$, $P_{\mathbf{Y}}$, we define their *entropy* as $H(\mathbf{X},\mathbf{Y}) = \mathrm{E}_{\mathbf{X},\mathbf{Y}}[-\log P(x,y)]$, $H(\mathbf{X}) = \mathrm{E}_{\mathbf{X}}[-\log P(x)]$, and $H(\mathbf{Y}) = \mathrm{E}_{\mathbf{Y}}[-\log P(y)]$. Furthermore, we define the conditional entropy of $\mathbf{X}$ given $\mathbf{Y}$ as $H(\mathbf{X}|\mathbf{Y}) = \mathrm{E}_{\mathbf{Y}}[-\log H(\mathbf{X}|y)] = H(\mathbf{X},\mathbf{Y}) - H(\mathbf{Y})$. Finally, we define the mutual information between $\mathbf{X}$ and $\mathbf{Y}$ as $I(\mathbf{X} \wedge \mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}) = H(\mathbf{X}) + H(\mathbf{Y}) - H(\mathbf{X},\mathbf{Y})$.

Notations and Problem Setup. In this paper, we use bold capital letters (e.g., $\mathbf{X}, \mathbf{Y}$) to denote random variables, lowercase letters (e.g., x, y) to denote their realizations, and curly capital letters (e.g., $\mathcal{X}, \mathcal{Y}$) to denote the corresponding sample spaces.

Since we mainly consider supervised and semi-supervised problems, we define Let $P_{\mathbf{X},\mathbf{Y}}$ be the joint distribution of data *observation* $\mathbf{X}$ and *label* $\mathbf{Y}$, where $\mathbf{X}$ is a random vector taking values on a finite observation space $\mathcal{X}$ (e.g., images) and $\mathbf{Y}$ is a discrete random variable taking values on the label space $\mathcal{Y}$ (e.g., classes). Our goal is to learn a classifier to predict $y \in \mathcal{Y}$ from an observation $x \in \mathcal{X}$.



(a) Without Augmentation (b) With Augmentation

Figure 1: **Probabilistic graphical models** of representation learning.

Task-nuisance Decomposition. To advance the analysis, we decouple the randomness in $\mathbf{X}$ into two parts, one pertaining to the label and another independent to the label. Concretely, we define a random variable *nuisance* $\mathbf{N}$ such that **1)** the nuisance $\mathbf{N}$ is independent to the label $\mathbf{Y}$, i.e., $\mathbf{N} \perp\!\!\!\perp \mathbf{Y}$; and **2)** the observation $\mathbf{X}$ is a deterministic function of the nuisance $\mathbf{N}$ and the label $\mathbf{Y}$, i.e., $\mathbf{X} = g(\mathbf{Y}, \mathbf{N})$ for some g. Lemma 3.1 demonstrates that such a random variable always exists.

Lemma 3.1 (Task-nuisance Decomposition [1])**.** *Given a joint distribution $P_{\mathbf{X},\mathbf{Y}}$, where $\mathbf{Y}$ is a discrete random variable, we can always find a random variable $\mathbf{N}$ independent of $\mathbf{Y}$ such that $\mathbf{X} = d(\mathbf{Y}, \mathbf{N})$, for some deterministic function d.*

Remarks. We can rewrite the conditions of task-nuisance decomposition in terms of information theory. **1)** Since the nuisance $\mathbf{N}$ is independent to the label $\mathbf{Y}$, we have $I(\mathbf{Y} \wedge \mathbf{N}) = 0$; and **2)** Since the nuisance $\mathbf{N}$ and the label $\mathbf{Y}$ determines the observation $\mathbf{X}$, we have $H(\mathbf{X}|\mathbf{Y}, \mathbf{N}) = 0$.

4 Principles of Representation Learning: Theoretical Interpretation

4.1 What Is A Good Representation?

In real-world applications, the observation $\mathbf{X}$ is usually complex in a high-dimensional space $\mathcal{X}$, making it hard to directly learn a good classifier for $\mathbf{Y}$. To remedy this curse of dimensionality, it is important to learn a good representation of $\mathbf{X}$, i.e., learn an encoder $E(\cdot)$ that maps the high-dimensional observation $\mathbf{X}$ into a low-dimensional representation $\mathbf{Z}$. We illustrate the process of data generation and representation learning by a probabilistic graphical model as shown in Figure 1a.

An ideal encoder should keep the important information from $\mathbf{X}$ (e.g. label-relevant information) and maximally discard the noise or nuisance of $\mathbf{X}$, such that it is much easier to learn a classifier from $\mathbf{Z}$ than from $\mathbf{X}$. Based on the above intuition, we define an ϵ-optimal representation of $\mathbf{X}$, which has sufficient information for classifying w.r.t. $\mathbf{Y}$, while remaining little information of the nuisance.

Definition 4.0.1 (ϵ-Minimal Sufficient Representation (ϵ-Optimal Representation))**.** *For a Markov chain $\mathbf{Y} \to \mathbf{X} \to \mathbf{Z}$, we say thata representation $\mathbf{Z}$ of $\mathbf{X}$ is sufficient for $\mathbf{Y}$ if $I(\mathbf{Z} \wedge \mathbf{Y}) = I(\mathbf{X} \wedge \mathbf{Y})$, and $\mathbf{Z}$ is ϵ-minimal sufficient for $\mathbf{Y}$ if $\mathbf{Z}$ is sufficient and $I(\mathbf{Z} \wedge \mathbf{X}) \leq I(\tilde{\mathbf{Z}} \wedge \mathbf{X}) + \epsilon$ for all $\tilde{\mathbf{Z}}$ satisfying $I(\tilde{\mathbf{Z}} \wedge \mathbf{Y}) = I(\mathbf{X} \wedge \mathbf{Y})$.*

Remark. Due to the property of mutual information, we have $0 \leq \epsilon \leq H(\mathbf{X})$. The lower ϵ is, the more "minimal" the representation is. When $\epsilon = 0$, the representation is minimal sufficient, which is a desirable property as characterized by many prior works [42, 1].

Definition 4.0.1 characterizes how good a sufficient representation is, based on how much redundant information is remained. Recall that $\mathbf{X}$ comes from a deterministic function of label $\mathbf{Y}$ and nuisance $\mathbf{N}$. The redundancy of $\mathbf{Z}$ can also be measured by the mutual information between $\mathbf{Z}$ and $\mathbf{N}$. Achille et al. [1] prove that if a representation $\mathbf{Z}$ is sufficient and is invariant to nuisance $\mathbf{N}$, i.e., $I(\mathbf{Z} \wedge \mathbf{N}) = 0$, then $\mathbf{Z}$ is also minimal. However, since $\mathbf{N}$ is not known, it is hard to directly encourage the representation to be invariant to $\mathbf{N}$.

Can we learn an ϵ-minimal sufficient representation in a principled way? Inspired by the recent success of data augmentation techniques in self-supervised learning and semi-supervised learning, we find that data augmentation can implicitly encourage the representation to be invariant to the nuisance $\mathbf{N}$. However, most augmentation methods are driven by pre-defined transformations, which do not necessarily render a minimal sufficient representation. In the next section, we will analyze the effects of data augmentation in representation learning in details.

4.2 Proper Data Augmentation Leads to (Near-)Optimal Representation

In this section, we investigate the role of data augmentation for learning good representations. We first make the following mild assumption on the underlying relationship between $\mathbf{X}$ and $\mathbf{Y}$.

Assumption 4.1. *There exists a deterministic function $\pi : \mathcal{X} \to \mathcal{Y}$, i.e., $H(\mathbf{Y}|\mathbf{X}) = 0$.*

Assumption 4.1 requires that there exists a "perfect classifier" that identifies the label y of the observation x with no error, which is common in practice. Note that for data with ambiguity, a tie breaker can be used to map each observation to a unique label. Therefore, Assumption 4.1 is realistic.

Let g be a deterministic augmentation function such that $\mathbf{X}' := g(\mathbf{X}, \mathbf{A})$ is the augmented data, where $\mathbf{A}$ is a random variable denoting the augmentation selection. For example, if $\mathbf{X} = x$ is an image sample, $\mathbf{A} = a$ is the augmentation "rotate by 90 degree", then $\mathbf{X}' = x'$ is the corresponding rotated image sample. We learn an encoder $E(\cdot)$ that maps the augmented data $\mathbf{X}'$ to a representation $\mathbf{Z}'$. With this augmentation processes, the graphical model in Figure 1a is updated to Figure 1b.

We show in the theorem below that if the augmentation process preserves the information of $\mathbf{Y}$, $\mathbf{Z}'$ can be sufficient for $\mathbf{Y}$. Furthermore, if the augmented data $\mathbf{X}'$ contains no information of the original nuisance $\mathbf{N}$, $\mathbf{Z}'$ will be invariant to $\mathbf{N}$ and thus will become a minimal sufficient representation.

Theorem 4.2. *Consider label variable $\mathbf{Y}$, observation variable $\mathbf{X}$ and nuisance variable $\mathbf{N}$ satisfying Assumption 4.1. Let $\mathbf{A}$ be the augmentation variable, $\mathbf{X}'$ be the augmented data, and $\mathbf{Z}^*$ be the solution to*

$$\begin{aligned} \operatorname{argmax}_{\mathbf{Z}'} \quad & I(\mathbf{Z}' \wedge \mathbf{X}') \text{ or } I(\mathbf{Z}' \wedge \mathbf{Y}) \\ \text{subject to} \quad & I(\mathbf{Z}' \wedge \mathbf{A}) = 0. \end{aligned} \tag{1}$$

Then, $\mathbf{Z}^*$ is a ϵ-minimal sufficient representation of $\mathbf{X}$ for label $\mathbf{Y}$ if the following conditions hold:
***Condition (a)**: $I(\mathbf{X}' \wedge \mathbf{Y}) = I(\mathbf{X} \wedge \mathbf{Y})$ ($\mathbf{X}'$ is an in-class augmentation) and*
***Condition (b)**: $I(\mathbf{X}' \wedge \mathbf{N}) \leq \epsilon$ ($\mathbf{X}'$ does not remain much information about $\mathbf{N}$).*

Remarks. (1) The objective of learning $\mathbf{Z}^*$ can be either task-independent (maximizing $I(\mathbf{Z}' \wedge \mathbf{X}')$), or task-dependent (maximizing $I(\mathbf{Z}' \wedge \mathbf{Y})$). The former matches the "InfoMax" principle commonly used in self-supervised learning works [27, 18], while the latter can be achieved by supervised training (e.g., learning a classifier of $\mathbf{Z}$ for $\mathbf{Y}$ with cross-entropy loss).
(2) When Condition (b) holds for $\epsilon = 0$, representation $\mathbf{Z}'$ is optimal (minimal sufficient).

Theorem 4.2, proved in Appendix B.1, shows that if we have a good augmentation that maximally perturbs the label-irrelevant information while keeps the label-relevant information, then the representation learned on the augmented data can be minimal sufficient. Theorem 4.2 serves as a principle of constructing augmentation. Based on this principle, we propose an auto-augment algorithm in Section 5, and verify the algorithm in a wide range tasks in Section 6.

5 Proposed Methods

In this section, we introduce our data augmentation and how to obtain the augmentation using the representation learning network $F(\cdot)$. Then we show how to plug our augmentation into the representation learning procedure of $F(\cdot)$.

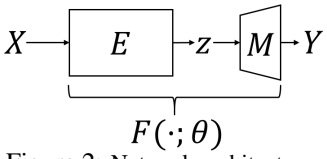


Figure 2: Network architecture.

5.1 Label-Preserving Adversarial Auto-Augment (LP-A3)

As illustrated in the previous section, an ideal data augmentation $\mathbf{X}'$ for representation learning should contain as little information about nuisance $\mathbf{N}$ as possible while still keeping all the information about class $\mathbf{Y}$. Since $\mathbf{N}$ is not observed, we transfer the objective $\min_{\mathbf{X}'} I(\mathbf{X}' \wedge \mathbf{N})$ into $\min_{\mathbf{X}'} I(\mathbf{X}' \wedge \mathbf{X})$ since $I(\mathbf{X}' \wedge \mathbf{X}) = I(\mathbf{X}' \wedge \mathbf{N}) + I(\mathbf{X}' \wedge \mathbf{Y})$ and $I(\mathbf{X}' \wedge \mathbf{Y})$ is a constant under the constraint $I(\mathbf{X}' \wedge \mathbf{Y}) = I(\mathbf{X} \wedge \mathbf{Y})$. Thus the optimization problem is:

$$\min_{\mathbf{X}'} I(\mathbf{X}' \wedge \mathbf{X}) \quad \text{s.t. } I(\mathbf{X}' \wedge \mathbf{Y}) = I(\mathbf{X} \wedge \mathbf{Y}). \tag{2}$$

Implementation of Mutual Information. To solve Equation (2), computing the mutual information terms $I(\mathbf{X}' \wedge \mathbf{X})$, $I(\mathbf{X}' \wedge \mathbf{Y})$ and $I(\mathbf{X} \wedge \mathbf{Y})$ is required. Next, we will show how to compute these terms using a neural net classifier $F(\cdot; \theta)$, parameterized by θ, that consists of two components:

a representation encoder $E(\cdot)$ and a predictor $M(\cdot)$. Specifically, $F(\cdot;\theta) = M(E(\cdot))$, where the representation encoder $E(\cdot)$ maps input $\mathbf{X}$ into representation $\mathbf{Z}$, and the predictor $M(\cdot)$ predicts the class of $\mathbf{Z}$. This is demonstrated in Figure 2.

Constraint implementation. Since $I(\mathbf{X}' \wedge \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}')$ and $I(\mathbf{X} \wedge \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X})$, we can remove the $H(\mathbf{Y})$ term in both sides and turn the constraint into $H(\mathbf{Y}|\mathbf{X}) = H(\mathbf{Y}|\mathbf{X}')$. Thus we only need to compute the conditional entropy of $\mathbf{Y}$ given $\mathbf{X}$ or $\mathbf{X}'$, which can be approximated through the neural net classifier: $H(\mathbf{Y}|\mathbf{X}) = \mathrm{E}_{\mathbf{X},\mathbf{Y}}\left[-\log P(y|x)\right] \approx \mathrm{E}_{\mathbf{X},\mathbf{Y}}\left[-\log(F(x;\theta)[y])\right]$, where we use softmax class probability $F(x;\theta)[y]$ to approximate the likelihood $P(y|x)$. And $H(\mathbf{Y}|\mathbf{X}')$ can be computed similarly.

Objective implementation. Then we show how to compute the objective $I(\mathbf{X}' \wedge \mathbf{X})$. Since $I(\mathbf{X}' \wedge \mathbf{X}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{X}')$ where $H(\mathbf{X})$ is not related to $\mathbf{X}'$ and thus can be neglected, we only need to compute $H(\mathbf{X}|\mathbf{X}') = \mathrm{E}_{\mathbf{X},\mathbf{X}'}\left[-\log P(x|x')\right]$. We use the Learned Perceptual Image Patch Similarity (LPIPS) [59] between x and x' to compute the data likelihood $P(x|x')$ since LPIPS distance is a widely used metric to measure the data similarity in data generative model field [21, 58] and many previous work has shown that LPIPS distance is the best surrogate for human comparisons of similarity [59, 24], compared with any other distance including ℓ_2 and ℓ_∞ distance. Although such surrogate may have error, it worth noting that Theorem 4.2 allows the surrogate to have ϵ error. The LPIPS distance is defined by the ℓ_2 distance of stacked feature maps from a neural network. Here we use $F(\cdot;\theta)$ to compute the LPIPS distance. Let $F(\cdot;\theta)$ has L layers and $\widehat{F}_l(\cdot;\theta)$ denotes these channel-normalized activations at the l-th layer of the network. Next, the activations are normalized again by layer size and flattened into a single vector $\phi(x) \triangleq (\frac{\widehat{F}_1(x;\theta)}{\sqrt{w_1 h_1}}, ..., \frac{\widehat{F}_L(x;\theta)}{\sqrt{w_L h_L}})$, where w_l and h_l are the width and height of the activations in layer l, respectively. The LPIPS distance between input x and the augmentation x' is then defined as:

$$\mathrm{LPIPS}(x, x') \triangleq \|\phi(x) - \phi(x')\|_2. \tag{3}$$

Constraint Relaxation for Efficiency. Now, given an input x, its data augmentation x' can be computed by solving the following optimization problem using the neural network $F(\cdot;\theta)$ in practice:

$$\min_{x'} -\|\phi(x) - \phi(x')\|_2 \quad \text{s.t. } \log F(x';\theta)[y] = \log F(x;\theta)[y]. \tag{4}$$

The equality constraint in Equation (4) is too strict to solve since it may be inefficient to search for an x' that exactly satisfies $\log F(x';\theta)[y] = \log F(x;\theta)[y]$. Thus we relax the constraint with a small σ and change the constaint into: $\log F(x';\theta)[y] \geq \log F(x;\theta)[y] - \sigma$. It's worth noting that if σ is sufficiently small, the label is still well preserved. There is a trade-off to the value of σ, we search σ to find a sweet spot where the problem is practical to solve and meanwhile the label is well preserved.

There are many off-the-shelf methods that solve Equation (4), and here we apply the Fast Lagrangian Attack Method [24] as a demonstration. We initialize x' by x plus a uniform noise. And we find the optimal x' by solving the following the Lagrangian multiplier function and gradually scheduling the value of the multiplier λ:

$$\min_{x'} -\|\phi(x) - \phi(x')\|_2 + \lambda \max(0, \log F(x;\theta)[y] - \log F(x';\theta)[y] - \sigma) \tag{5}$$

The detailed procedure of the algorithm can be found in Appendix 2. The algorithm has a similar form as adversarial attack [61, 52] in that they both find an optimal augmentation x' by adding perturbations to the original image x. However, the difference is that we aim to generate hard augmentation that preserves the label, while adversarial attack aims to change the class label.

5.2 Plugging LP-A3 into a Representation Learning Task

One primary advantage of LP-A3 is that it only requires a neural net $F(\cdot;\theta)$ to produce the augmentation and $F(\cdot;\theta)$ can be the current representation learning model, so we can plug LP-A3 into any representation learning procedure requiring no additional parameters, which is plug-and-play and parameter-free. At each step, we first fix $F(\cdot;\theta)$ to generate the augmentation x' by solving Equation (5) using Algorithm 2. And then we train $F(\cdot;\theta)$ by running the original representation learning algorithm using our augmentation x'.

Data selection. It is not necessary to find hard positives for every sample. To save more computation, we can apply a sharpness-aware criterion, i.e., time-consistency (TCS) [63], to select the most

informative data ($\tau\%$ data with the lowest TCS in Algorithm 1) that have sharp loss landscapes, which indicate the existence of nearby hard positives, and we only apply LP-A3 to them. It reduces the computational cost without degrading the performance because (1) the improvement brought by augmentations is limited for examples whose loss already reaches a flat minimum, while the model does not generalize well near examples with a sharp loss landscapes; and (2) the hard positives for examples with flat loss landscape are distant from the original ones and might introduce extra bias to the training.

LP-A3 is compatible with any representation learning task minimizing a loss $L : \mathcal{X} \times \mathcal{Y} \times \mathcal{W} \to \mathcal{R}_+$, which takes in a data batch and a model to output a loss value. $\mathcal{Y}$ here denotes the groundtruth label for labeled data and pseudo label for unlabeled data. The pseudo-code of plugging LP-A3 into the representation learning procedure with TCS-based data selection is provided in Algorithm 1.

Algorithm 1 Plug LP-A3 into any representation learning procedure

Input: Loss for the targeted task $L : \mathcal{X} \times \mathcal{Y} \times \mathcal{W} \to \mathcal{R}_+$; training data $(\mathcal{X}, \mathcal{Y})$; neural network $F(\cdot; \theta)$; class preserving margin ϵ; data selection ratio τ; learning rate η;
Output: Model parameter θ trained with LP-A3

1: **while** *not converged* **do**
2: Sample batch $\mathcal{B} = \{(x_1, y_1), ..., (x_b, y_b)\} \sim (\mathcal{X}, \mathcal{Y})$;
3: Data selection: $\mathcal{S} \leftarrow \tau\%$ data with the lowest TCS in $\mathcal{B}$;
4: LP-A3: Freeze θ and solve Equation (5) using Algorithm 2 for every sample in $\mathcal{S}$, resulting in an augmented set $\mathcal{A} = \{(x'_1, y_1), ..., (x'_m, y_m)\}$ of size $m = |\mathcal{S}|$;
5: Learning with LP-A3 augmented data and original data: $\theta \leftarrow \theta - \eta[\nabla_\theta L(\mathcal{B}; \theta) + \nabla_\theta L(\mathcal{A}; \theta)]$;
6: **end while**

6 Experiments

In this section, we apply LP-A3 as a data augmentation method to several popular methods for three different learning tasks, i.e., (1) semi-supervised classification; (2) noisy-label learning and (3) medical image classification. In all the experiments, LP-A3 can (1) consistently improve the convergence and test accuracy of existing methods and (2) autonomously produce augmentations that bring non-trivial improvement even without any domain knowledge available. A walk-clock time comparison is given in the Appendix, showing LP-A3 effectively reduces the computational cost. In addition, we conduct a thorough sensitivity study of LP-A3 by changing (1) label-preserving margin and (2) data selection ratio on the three tasks. More experimental details can be found in the Appendix.

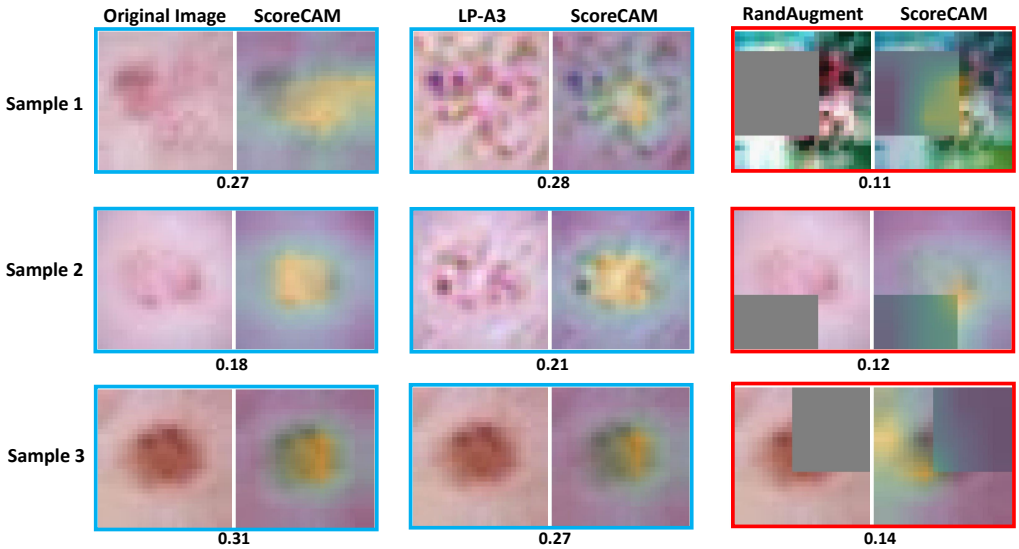
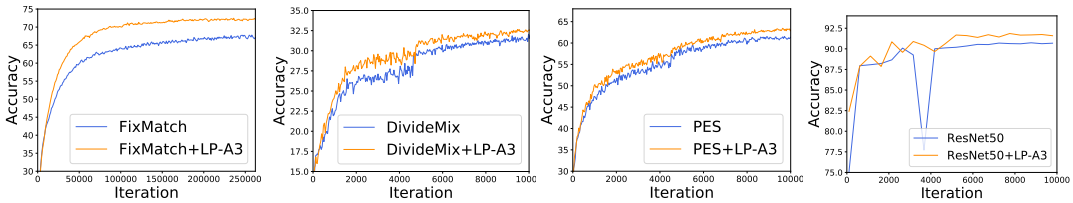


Figure 3: **Visualization of medical image augmentations** on the test set of DermaMnist. Blue (red) bounding box marks the correct (wrong) prediction of a ResNet18 classifier and its confidence on the groundtruth class is reported beneath the box. ScoreCAM [44] produces a heatmap highlighting important areas (by yellow color) of an image that a neural net mainly relies on to make the prediction.



(a) FixMatch (b) DivideMix (c) PES (d) MedMnist

Figure 4: **Convergence Curve** when applying LP-A3 to different tasks and baselines.

6.1 Medical Image Augmentations produced by LP-A3 vs. RandAugment

We visualize data augmentations generated by LP-A3 and RandAugment [12] on the testset of DermaMnist [50] with a ResNet-18 classifier and its confidence on the groundtruth class in Fig. 3. We also use ScoreCAM [44] as an interpretation method to highlight the area in each image that the classifier relies on to make the prediction. we find that LP-A3 preserves relevant derma areas highlighted by ScoreCAM and they are consistent with those in the original image. On the contrary, RandAugment changes the color or occludes those derma areas, resulting in highly different ScoreCAM heatmaps and hence wrong predictions (red bounding box in Fig. 3). Instead, LP-A3 can preserve the class information and mainly perturb the class-unrelated area in the original image.

6.2 Applying LP-A3 to Three Different Representation Learning Tasks

Here we apply LP-A3 to three different tasks by pluging LP-A3 to existing baselines of each task. Fig. 4 shows that LP-A3 greatly speeds up the convergence of each baseline.

Semi-supervised learning To evaluate how LP-A3 improves the learning without sufficient labeled data, we conduct experiments on semi-supervised classification on standard benchmarks including CIFAR [22] and STL-10 [10] where only a very small amount of labels are revealed. We apply LP-A3 in FixMatch [36] and compare it with the original FixMatch and InfoMin [40], a learnable augmentation method for semi-supervised learning. Their results are reported in Table 1, where LP-A3 consistently improves FixMatch and the improvement becomes more significant if reducing the labeled data. It's worth noting that the original FixMatch already employs a carefully designed set of pre-defined augmentations [13] that have been tuned to achieve the best performance, indicating that LP-A3 is complementary to existing data augmentations. Moreover, LP-A3 also outperforms InfoMin by a large margin ($> 5\%$), which indicates that LP-A3 is also superior to existing learnable augmentations.

Table 1: **Semi-supervised Learning** performance on CIFAR with different amounts of labeled data. § denotes results reproduced using the official code. FixMatch and LP-A3 are trained for 2^{18} SGD steps. InfoMin's results on CIFAR are missing since their paper only reports the result on STL-10. Error bars (mean and std) are computed over three random trails.

Dataset	CIFAR10			CIFAR100			STL-10
# Label	40	250	4000	400	2500	10000	1000
InfoMin (RGB) [40]	-	-	-	-	-	-	86.0
InfoMin (YDbDr) [40]	-	-	-	-	-	-	87.0
FixMatch [36]§	89.51±3.14	93.81±0.29	94.66±0.13	49.30±2.45	67.21±0.94	74.31±0.35	91.59±0.16
FixMatch [36] + LP-A3	**92.39±1.21**	**94.03±0.31**	**95.11±0.17**	**56.16±1.82**	**72.23±0.57**	**77.11±0.16**	**92.63±0.14**

Noisy-label Learning Data augmentation is critical to noisy-label learning by providing different views of data to prevent neural nets from overfitting to noisy labels. We apply LP-A3 to two state-of-the-art methods DivideMix [26] and PES [5] on CIFAR with different ratios of noise labels. LP-A3 can consistently improve the performance of these two SoTA methods and the improvement is more significant in more challenging cases with higher noise ratios, e.g., on CIFAR100 with 90% of labels to be noisy, LP-A3 improves PES by $\geq 15\%$ (Table 2).

Medical Image Classification To evaluate the performance in specific areas without domain knowledge, we compare LP-A3 with existing data augmentations on medical image classification tasks from MedMNIST [50], which is composed of several sub-dataset with various styles of medical images. We compare our LP-A3 with RandAugment [13] on training ResNet-18 and ResNet-50 [17]. We report the results in Table 3, where RandAugment designed for natural images fails to improve

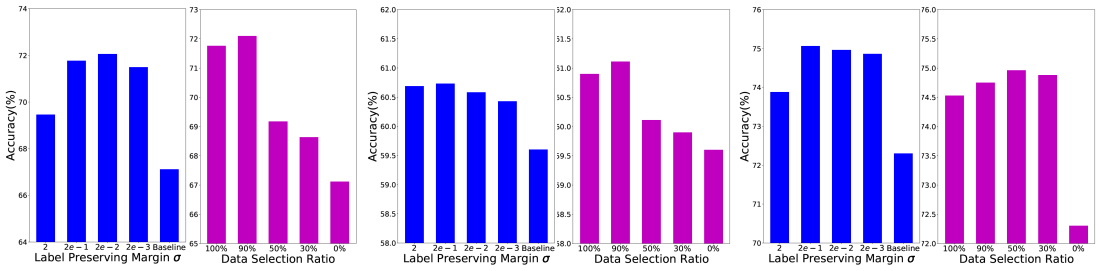
Table 2: **Noisy-label learning** performance on CIFAR with different ratios of symmetric label noises. § denotes the results reproduced by the official code. Error bars (mean and std) are computed over three random trails.

Dataset	CIFAR10			CIFAR100		
Noise Ratio	50%	80%	90%	50%	80%	90%
Mixup [56]	87.1	71.6	52.2	57.3	30.8	14.6
P-correction [54]	88.7	76.5	58.2	56.4	20.7	8.8
M-correlation [3]	88.8	76.1	58.3	58.0	40.1	14.3
DivideMix [26]	94.4	92.9	75.4	74.2	59.6	31.0
DivideMix+LP-A3	94.89±0.05	**93.70±0.19**	79.35±1.33	74.12±0.23	61.00±0.34	32.55±0.25
PES§ [5]	94.89±0.12	92.15±0.23	84.98±0.36	74.19±0.23	61.47±0.38	21.15±3.15
PES+LP-A3	**95.10±0.14**	93.26±0.21	**87.71±0.36**	**74.57±0.25**	**62.98±0.49**	**40.61±1.10**

Table 3: **Medical Image Classification** on MedMNIST [50]. All the models are trained for 100 epochs. Error bars (mean and std) are computed over three random trails.

Method	PathMNIST	DermaMNIST	TissueMNIST	BloodMNIST
ResNet-18	94.34±0.18	76.14±0.09	68.28±0.17	96.81±0.19
ResNet-18+RandAugment	93.52±0.09	73.71±0.33	62.03±0.14	95.00±0.21
ResNet-18+LP-A3	**94.42±0.24**	**76.22±0.27**	**68.63±0.14**	**96.97±0.06**
ResNet-50	94.47±0.38	75.24±0.27	69.69±0.23	96.91±0.06
ResNet-50+RandAugment	94.02±0.37	71.65±0.30	65.13±0.33	95.14±0.06
ResNet-50+LP-A3	**94.57±0.07**	**75.71±0.22**	**69.89±0.08**	**97.01±0.32**
	OctMNIST	OrganAMNIST	OrganCMNIST	OrganSMNIST
ResNet-18	78.67±0.26	94.21±0.09	91.81±0.12	81.57±0.07
ResNet-18+RandAugment	76.00±0.24	94.18±0.20	91.38±0.14	80.52±0.32
ResNet-18+LP-A3	**80.27±0.54**	**94.73±0.21**	**92.41±0.22**	**82.28±0.38**
ResNet-50	78.37±0.52	94.31±0.14	91.80±0.14	81.11±0.21
ResNet-50+RandAugment	76.63±0.58	94.59±0.17	91.10±0.12	80.47±0.37
ResNet-50+LP-A3	**79.40±0.36**	**94.95±0.19**	**92.16±0.23**	**82.15±0.08**

the performance in this scenario. In contrast, LP-A3 does not rely on any domain knowledge brings improvement to all the datasets, especially for OctMNIST where the improvement is over 1%. The results indicate that hand-crafted data augmentations do not generalize to all domains but LP-A3 can autonomously produce augmentations guided by our representation learning principle without relying on any domain knowledge.



(a) Semi-supervised classification (b) Noisy-label Classification (c) MedMnist

Figure 5: **Sensitivity Analysis** of label preserving margin σ and data selection ratio.

6.3 Sensitivity Analysis of Hyperparameters

Label preserving margin σ: We evaluate how LP-A3 performs with different label preserving margin σ on the three tasks. The results are presented in Fig. 5, where a reverse U-shape is observed. And LP-A3 using all the evaluated σ outperforms baselines, which indicates LP-A3 is robust to σ.

Data selection ratio: We evaluate the performance of LP-A3 with different amount of data selected on the three tasks. As shown in Fig. 5, selecting all the data does not perform the best since some data' augmentations are useless to apply data augmentation. Moreover, selecting only 30% data to apply LP-A3 can outperform all baselines by a large margin, especially on MedMNIST where the improvement is $\geq 2\%$, which verifies the effectiveness of LP-A3 and our data selection method.

7 Conclusion

In this paper, we study how to automatically generate domain-agnostic but task-informed data augmentations. We first investigate the conditions required for augmentations leading to representations that preserves the task (label) information and then derive an optimization objective for the augmentations. For practicality, we further propose a surrogate of the derived objective that can be efficiently computed from the intermediate-layer representations of the model-in-training. The surrogate is built upon the data likelihood estimation through perceptual distance. This leads to LP-A3, a general and autonomous data augmentation technique applicable to a variety of machine learning tasks, such as supervised, semi-supervised and noisy-label learning. In experiments, we demonstrate that LP-A3 can consistently bring improvement to SoTA methods for different tasks even without domain knowledge. In future work, we will extend LP-A3 to more learning tasks and further improve its efficiency.

Acknowledgements

This work was supported by, the Major Science and Technology Innovation 2030 "New Generation Artificial Intelligence" key project under Grant 2021ZD0111700, NSFC No. 61872329, No. 62222117, and the Fundamental Research Funds for the Central Universities under contract WK3490000005. Huang, Sun and Su are supported by NSF-IIS-FAI program, DOD-ONR-Office of Naval Research, DOD-DARPA-Defense Advanced Research Projects Agency Guaranteeing AI Robustness against Deception (GARD). Huang is also supported by Adobe, Capital One and JP Morgan faculty fellowships.

References

[1] Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

[2] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks, 2017.

[3] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International conference on machine learning*, pages 312–321. PMLR, 2019.

[4] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

[5] Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu. Understanding and improving early stopping for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34, 2021.

[6] Randall Balestriero, Leon Bottou, and Yann LeCun. The effects of regularization and data augmentation are class dependent, 2022.

[7] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations (ICLR)*, 2020.

[8] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 5050–5060. 2019.

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

[10] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[11] Thomas M Cover and Joy A Thomas. Information theory and statistics. *Elements of information theory*, 1(1):279–335, 1991.

[12] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

[13] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[14] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[15] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537, 2018.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

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

[18] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018.

[19] Chih-Hui Ho and Nuno Nvasconcelos. Contrastive learning with adversarial examples. *Advances in Neural Information Processing Systems*, 33:17081–17093, 2020.

[20] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313, 2018.

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

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

[23] Alex Kurakin, Chun-Liang Li, Colin Raffel, David Berthelot, Ekin Dogus Cubuk, Han Zhang, Kihyuk Sohn, Nicholas Carlini, and Zizhao Zhang. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

[24] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. *arXiv preprint arXiv:2006.12655*, 2020.

[25] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, 2013.

[26] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[27] Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

[28] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:447–461, 2016.

[29] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

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

[31] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *ArXiv*, abs/1712.04621, 2017.

[32] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[33] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, pages 1163–1171. 2016.

[34] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, pages 1163–1171. 2016.

[35] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

[36] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33:596–608, 2020.

[37] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

[38] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 1195–1204. 2017.

[39] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020.

[40] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

[41] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[42] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

[43] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations*, 2019.

[44] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020.

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

[46] Mike Wu, Chengxu Zhuang, Daniel Yamins, and Noah Goodman. On the importance of views in unsupervised representation learning. *preprint*, 3, 2020.

[47] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[48] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020.

[49] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification, 2021.

[50] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification. *arXiv preprint arXiv:2110.14795*, 2021.

[51] Kaiwen Yang, Tianyi Zhou, Xinmei Tian, and Dacheng Tao. Identity-disentangled adversarial augmentation for self-supervised learning. In *International Conference on Machine Learning*, pages 25364–25381. PMLR, 2022.

[52] Kaiwen Yang, Tianyi Zhou, Yonggang Zhang, Xinmei Tian, and Dacheng Tao. Class-disentanglement and applications in adversarial detection and defense. *Advances in Neural Information Processing Systems*, 34:16051–16063, 2021.

[53] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019.

[54] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7017–7025, 2019.

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

[56] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[57] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[58] Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. Cross-domain correspondence learning for exemplar-based image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5143–5153, 2020.

[59] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[60] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. *arXiv preprint arXiv:1912.11188*, 2019.

[61] Yonggang Zhang, Xinmei Tian, Ya Li, Xinchao Wang, and Dacheng Tao. Principal component adversarial example. *IEEE Transactions on Image Processing*, 29:4804–4815, 2020.

[62] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. Distilling localization for self-supervised representation learning. In *35th AAAI Conference on Artificial Intelligence (AAAI-21)*, pages 10990–10998. AAAI Press, 2021.

[63] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Time-consistent self-supervision for semi-supervised learning. In *International Conference on Machine Learning*, pages 11523–11533. PMLR, 2020.

[64] Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Robust curriculum learning: from clean label detection to noisy label self-correction. In *International Conference on Learning Representations*, 2021.

[65] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6002–6012, 2019.

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? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (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] See Sec.4
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix B
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)? [No] Will open source later
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Aappendix C
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (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)? [No]
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? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (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]

Supplementary Material

A Algorithmic Details

A.1 Data Selection Via Time-Consistency

We use time-consistency (TCS) [63] to select informative sample to apply our augmentation, which computes the consistency of the output distribution for each sample along the training procedure. Specifically, TCS metric $c^t(x)$ for an individual sample is an negative exponential moving average of $a^t(x)$ over training history before t:

$$c^t(x) = \gamma_c \left(-a^t(x)\right) + (1-\gamma_c)\, c^{t-1}(x) \tag{6}$$

$$a^t(x) \triangleq D_{KL}\left(F^{t-1}(x) \| F^t(x)\right) + \left|\log \frac{F^{t-1}(x)[y^{t-1}(x)]}{F^t(x)[y^{t-1}(x)]}\right| \tag{7}$$

where $D_{KL}(\cdot\|\cdot)$ is Kullback–Leibler divergence, $y^{t-1}(x)$ is pesudo label (for unlabeled data) or real label (for labeled data) for x at step t and $\gamma_c \in [0,1]$ is a discount factor. Intuitively, the KL-divergence between output distributions measures how consistent the output is between two consecutive steps, and a moving average of $a^t(x)$ naturally captures inconsistency of x over time quantify. And larger $c^t(x)$ means better time-consistency. We select top $\tau\%$ sample with the lowest TCS to apply our data augmentation because samples with small TCS tend to have sharp loss landscapes. These samples provide more informative gradients than others and applying our model-adaptive data augmentations can bring more improvement to their representation invariance and loss smoothness. In Fig. 5, we conduct a thorough sensistivity analysis on $\tau\%$ over three tasks and find that sample selection with TCS can effectively improve the performance. Moreover, in this way, we do not need to apply our augmentation to every training samples and thus save the training cost.

A.2 Fast Lagaragian Attack Method

We use the fast lagaragian perceptual attack method (Algorithm 3 in [24]) to solve the Lagragian multiplier function in Equation.(5), which finds the optimal x' through gradient descent over x', starting at x with a small amount of noise added. During the T gradient descent steps, λ is increased exponentially form 1 to 10 and the step size is decreased. T is set to be 5 for all the experiments.

Algorithm 2 Fast Lagarangian Attack Method

Input:
Training data (x,y); The class preserving margin σ; Neural Network $F(\cdot)$
Output:
$x' = x + 0.01 * \mathcal{N}(0,1)$
for $t = 1, ..., T$ **do**
  $\lambda \leftarrow 10^{t/T}$
  $\triangle = -\nabla_{x'}[\|\phi(x) - \phi(x')\|_2 - \lambda \max(0, \log F(x;\theta)[y] - \log F(x';\theta)[y] - \sigma)]$
  $\hat{\triangle} = \triangle / \|\triangle\|_2$
  $\gamma = \epsilon * (0.1)^t / T$
  $m \leftarrow (F(x;\theta)[y] - F(x' + h\hat{\triangle};\theta)[y])/h$
  $x' \leftarrow x + (\gamma/m)\hat{\triangle}$
end for

B Additional Theoretical Results and Proofs

B.1 Proof of Theorem 4.2

Proof of Theorem 4.2. Problem (1) contains two versions of objectives for $\mathbf{Z}'$:

$$\text{argmax}_{\mathbf{Z}'} I(\mathbf{Z}' \wedge \mathbf{X}') \text{ subject to } I(\mathbf{Z}' \wedge \mathbf{A}) = 0, \tag{8}$$

or

$$\text{argmax}_{\mathbf{Z}'} I(\mathbf{Z}' \wedge \mathbf{Y}) \text{ subject to } I(\mathbf{Z}' \wedge \mathbf{A}) = 0. \tag{9}$$

Both Problem (8) and Problem (9) lead to the ϵ-minimal sufficient representation $\mathbf{Z}^*$. We first prove the more challenging Problem (8) objective.

I. For Problem (8): $\text{argmax}_{\mathbf{Z}'} I(\mathbf{Z}' \wedge \mathbf{X}')$ subject to $I(\mathbf{Z}' \wedge \mathbf{A}) = 0$.

We first prove the sufficiency of $\mathbf{Z}^*$, then prove the ϵ-minimality of $\mathbf{Z}^*$.

1) Proof of sufficiency

Since $I(\mathbf{Z}' \wedge \mathbf{X}') = H(\mathbf{X}') - H(\mathbf{X}'|\mathbf{Z}')$, and $H(\mathbf{X}')$ does not depend on $\mathbf{Z}'$, we have that the solution to Problem (8), $\mathbf{Z}^*$, minimizes $H(\mathbf{X}'|\mathbf{Z}')$ under constraint $I(\mathbf{Z}' \wedge \mathbf{A}) = 0$.

Then, we show that $\mathbf{Z}^*$ also minimizes $H(\mathbf{X}|\mathbf{Z}')$.

We know that $I(\mathbf{X}, \mathbf{A} \wedge \mathbf{Z}'|\mathbf{X}') = 0$ because of the Markovian property. Since $I(\mathbf{X}, \mathbf{A} \wedge \mathbf{Z}'|\mathbf{X}') = H(\mathbf{X}, \mathbf{A}|\mathbf{X}') - H(\mathbf{X}, \mathbf{A}|\mathbf{X}', \mathbf{Z}')$, we have

$$H(\mathbf{X}, \mathbf{A}|\mathbf{X}') = H(\mathbf{X}, \mathbf{A}|\mathbf{X}', \mathbf{Z}'). \tag{10}$$

Then we can derive

$$\begin{aligned}
H(\mathbf{X}, \mathbf{A}|\mathbf{Z}') - H(\mathbf{X}, \mathbf{A}|\mathbf{X}') &= H(\mathbf{X}, \mathbf{A}|\mathbf{Z}') - H(\mathbf{X}, \mathbf{A}|\mathbf{X}', \mathbf{Z}') && (11)\\
&= I(\mathbf{X}, \mathbf{A} \wedge \mathbf{X}'|\mathbf{Z}') && (12)\\
&= H(\mathbf{X}'|\mathbf{Z}') - H(\mathbf{X}'|\mathbf{Z}', \mathbf{X}, \mathbf{A}) && (13)\\
&= H(\mathbf{X}'|\mathbf{Z}') && (14)
\end{aligned}$$

Equality (13) holds because $\mathbf{X}'$ comes from a deterministic function of $\mathbf{X}$ and $\mathbf{A}$. Since $H(\mathbf{X}, \mathbf{A}|\mathbf{X}')$ does not depend on $\mathbf{Z}'$, $\mathbf{Z}^*$ minimizes $H(\mathbf{X}, \mathbf{A}|\mathbf{Z}')$ as it minimizes $H(\mathbf{X}'|\mathbf{Z}')$.

Also, we known that $I(\mathbf{Z}^* \wedge \mathbf{A}) = 0$, so we can further obtain

$$\begin{aligned}
H(\mathbf{X}'|\mathbf{Z}') &= H(\mathbf{X}, \mathbf{A}|\mathbf{Z}') && (15)\\
&= H(\mathbf{X}|\mathbf{Z}') + H(\mathbf{A}|\mathbf{Z}') - I(\mathbf{X} \wedge \mathbf{A}|\mathbf{Z}') && (16)\\
&= H(\mathbf{X}|\mathbf{Z}') + H(\mathbf{A}|\mathbf{Z}') - H(\mathbf{A}|\mathbf{Z}') + H(\mathbf{A}|\mathbf{X}, \mathbf{Z}') && (17)\\
&= H(\mathbf{X}|\mathbf{Z}') + H(\mathbf{A}|\mathbf{X}, \mathbf{Z}') && (18)\\
&= H(\mathbf{X}|\mathbf{Z}'), && (19)
\end{aligned}$$

where Equation (19) holds because $H(\mathbf{A}|\mathbf{X}, \mathbf{Z}') \leq H(\mathbf{A}|\mathbf{Z}') = 0$.

Therefore, $\mathbf{Z}^*$ minimizes $H(\mathbf{X}|\mathbf{Z}')$.

Following the similar procedure as above (Equation (10) to Equation (19)), we are able to show that

$$H(\mathbf{X}|\mathbf{Z}') = H(\mathbf{Y}, \mathbf{N}|\mathbf{Z}') - H(\mathbf{Y}, \mathbf{N}|\mathbf{X}) \tag{20}$$

So $\mathbf{Z}^*$ also minimizes $H(\mathbf{Y}, \mathbf{N}|\mathbf{Z}')$, which can be further decomposed into $H(\mathbf{Y}|\mathbf{Z}') + H(\mathbf{N}|\mathbf{Z}', \mathbf{Y})$. Next we show by contradiction that $H(\mathbf{Y}|\mathbf{Z}^*)$ equals to $H(\mathbf{Y}|\mathbf{X})$ and thus $I(\mathbf{Y} \wedge \mathbf{Z}^*) = I(\mathbf{Y} \wedge \mathbf{X})$.

Define $L(\mathbf{Z}') := H(\mathbf{Y}|\mathbf{Z}') + H(\mathbf{N}|\mathbf{Z}', \mathbf{Y})$ Assume that the optimizer $\mathbf{Z}^*$ minimizes L, but does not satisfy sufficiency, i.e., $H(\mathbf{Y}|\mathbf{Z}^*) > H(\mathbf{Y}|\mathbf{X}) = 0$. We will then show that one can construct another representation $\hat{\mathbf{Z}}$ such that $L(\hat{\mathbf{Z}}) < L(\mathbf{Z}^*)$, conflicting with the assumption that $\mathbf{Z}^*$ minimizes L. The construction of $\hat{\mathbf{Z}}$ works as follows. Since the augmented data $\mathbf{X}'$ satisfies $I(\mathbf{X}' \wedge \mathbf{Y}) = I(\mathbf{X} \wedge \mathbf{Y})$ (Condition (a) of Theorem 4.2), we have $H(\mathbf{Y}|\mathbf{X}') = H(\mathbf{Y}|\mathbf{X}) = 0$. Hence, there exists a function π' such that $\pi'(\mathbf{X}') = \mathbf{Y}$. Define $\hat{\mathbf{Z}} := (\mathbf{Z}^*, \pi(\mathbf{X}'))$, then we have

$$
\begin{aligned}
L(\hat{\mathbf{Z}}) &= H(\mathbf{Y}|\hat{\mathbf{Z}}) + H(\mathbf{N}|\hat{\mathbf{Z}}, \mathbf{Y}) && (21)\\
&= H(\mathbf{Y}|\mathbf{Z}^*, \pi(\mathbf{X}')) + H(\mathbf{N}|\mathbf{Z}^*, \pi(\mathbf{X}'), \mathbf{Y}) && (22)\\
&= H(\mathbf{Y}|\mathbf{Z}^*, \mathbf{Y}) + H(\mathbf{N}|\mathbf{Z}^*, \mathbf{Y}, \mathbf{Y}) && (23)\\
&= 0 + H(\mathbf{N}|\mathbf{Z}^*, \mathbf{Y}) && (24)\\
&< H(\mathbf{Y}|\mathbf{Z}^*) + H(\mathbf{N}|\mathbf{Z}^*, \mathbf{Y}) && (25)\\
&= L(\mathbf{Z}^*) && (26)
\end{aligned}
$$

Therefore, the constructed $\hat{\mathbf{Z}}$ conflicts with the assumption. We can conclude that any optimizer $\mathbf{Z}^* \in \operatorname{argmin}_{\mathbf{Z}'} H(\mathbf{Y}|\mathbf{Z}') + H(\mathbf{N}|\mathbf{Z}', \mathbf{Y})$ has to satisfy $H(\mathbf{Y}|\mathbf{Z}^*) = H(\mathbf{Y}|\mathbf{X})$, which is equivalent to $I(\mathbf{Y} \wedge \mathbf{Z}^*) = I(\mathbf{Y} \wedge \mathbf{X})$. The sufficiency of $\mathbf{Z}^*$ is thus proven.

As a result, the maximizer to Problem (8), $\mathbf{Z}^*$, satisfies $I(\mathbf{Z}^* \wedge \mathbf{Y}) = I(\mathbf{X}' \wedge \mathbf{Y}) = I(\mathbf{X} \wedge \mathbf{Y})$.

2) Proof of ϵ-Minimality

Since $\mathbf{X}$ is a deterministic function of $\mathbf{Y}$ and $\mathbf{N}$, we have

$$
\begin{aligned}
I(\mathbf{X}' \wedge \mathbf{X}) &= I(\mathbf{X}' \wedge \mathbf{Y}, \mathbf{N}) && (27)\\
&= I(\mathbf{X}' \wedge \mathbf{N}) + I(\mathbf{X}' \wedge \mathbf{Y}|\mathbf{N}) && (28)\\
&\leq I(\mathbf{X}' \wedge \mathbf{Y}|\mathbf{N}) + \epsilon && (29)
\end{aligned}
$$

where the equality in Equation (27) holds because $I(\mathbf{X}' \wedge \mathbf{X}) \geq I(\mathbf{X}' \wedge \mathbf{Y}, \mathbf{N})$ and $I(\mathbf{X}' \wedge \mathbf{X}) \leq I(\mathbf{X}' \wedge \mathbf{Y}, \mathbf{N})$ both hold.

And we can derive

$$
\begin{aligned}
I(\mathbf{X}' \wedge \mathbf{Y}|\mathbf{N}) - I(\mathbf{X} \wedge \mathbf{Y}) &= H(\mathbf{Y}|\mathbf{N}) - H(\mathbf{Y}|\mathbf{X}', \mathbf{N}) - H(\mathbf{Y}) + H(\mathbf{Y}|\mathbf{X})\\
&= H(\mathbf{Y}|\mathbf{X}) - H(\mathbf{Y}|\mathbf{X}', \mathbf{N})\\
&\leq H(\mathbf{Y}|\mathbf{X})\\
&= 0
\end{aligned} \tag{30}
$$

Moreover, we know that

$$
I(\mathbf{Y} \wedge \mathbf{N}) + I(\mathbf{X}' \wedge \mathbf{Y}|\mathbf{N}) = I(\mathbf{X}' \wedge \mathbf{Y}) + I(\mathbf{Y} \wedge \mathbf{N}|\mathbf{X}') \tag{31}
$$

And we have $I(\mathbf{Y} \wedge \mathbf{N}) = 0$, so

$$
\begin{aligned}
I(\mathbf{X}' \wedge \mathbf{Y}|\mathbf{N}) - I(\mathbf{X} \wedge \mathbf{Y}) &= I(\mathbf{X}' \wedge \mathbf{Y}|\mathbf{N}) - I(\mathbf{X}' \wedge \mathbf{Y})\\
&= I(\mathbf{Y} \wedge \mathbf{N}|\mathbf{X}') \geq 0
\end{aligned} \tag{32}
$$

Combining (30) and (32), we have

$$
I(\mathbf{X}' \wedge \mathbf{Y}|\mathbf{N}) = I(\mathbf{X} \wedge \mathbf{Y}) \tag{33}
$$

Note that (33) holds for all sufficient statistics of $\mathbf{X}$ w.r.t. $\mathbf{Y}$.

Then we first show that $\mathbf{X}'$ is ϵ-minimal of $\mathbf{X}$ w.r.t. $\mathbf{Y}$ by contradiction.

Assume there exists a random variable $\tilde{\mathbf{X}}$ satisfying $I(\tilde{\mathbf{X}} \wedge \mathbf{Y}) = I(\mathbf{X} \wedge \mathbf{Y})$, such that $I(\tilde{\mathbf{X}} \wedge \mathbf{X}) < I(\mathbf{X}' \wedge \mathbf{X}) - \epsilon$.

Then we have

$$
\begin{aligned}
I(\tilde{\mathbf{X}} \wedge \mathbf{Y}|\mathbf{N}) &= I(\tilde{\mathbf{X}} \wedge \mathbf{X}) - I(\tilde{\mathbf{X}} \wedge \mathbf{N}) && (34)\\
&\leq I(\tilde{\mathbf{X}} \wedge \mathbf{X}) && (35)\\
&< I(\mathbf{X}' \wedge \mathbf{X}) - \epsilon && (36)\\
&= I(\mathbf{X}' \wedge \mathbf{Y}|\mathbf{N}) + \epsilon - \epsilon && (37)\\
&= I(\mathbf{X} \wedge \mathbf{Y}) && (38)\\
&= I(\tilde{\mathbf{X}} \wedge \mathbf{Y}|\mathbf{N}) && (39)
\end{aligned}
$$

where (34) holds by replacing $\mathbf{X}'$ with $\tilde{\mathbf{X}}$ in (28).

Hence, we get $I(\tilde{\mathbf{X}} \wedge \mathbf{Y}|\mathbf{N}) < I(\tilde{\mathbf{X}} \wedge \mathbf{Y}|\mathbf{N})$ which is impossible. So we have that there does not exist such an $\tilde{\mathbf{X}}$, and $\mathbf{X}'$ is ϵ-minimal representation of $\mathbf{X}$ w.r.t. $\mathbf{Y}$.

Then, since we have $I(\mathbf{Z}' \wedge \mathbf{N}) \leq I(\mathbf{X}' \wedge \mathbf{N}) \leq \epsilon$ (thanks to Data Processing Inequality), $\mathbf{Z}'$ is also a ϵ-minimal sufficient statistic of $\mathbf{X}$ w.r.t. $\mathbf{Y}$.

II. For Problem (9): $\operatorname{argmax}_{\mathbf{Z}'} I(\mathbf{Z}' \wedge \mathbf{Y})$ subject to $I(\mathbf{Z}' \wedge \mathbf{A}) = 0$.

Since the objective is to maximize $I(\mathbf{Z}' \wedge \mathbf{Y})$, we only need to show that $\mathbf{Z}^*$ achieves the maximum mutual information with $\mathbf{Y}$. According to the above proof for Problem (8), we know that there exist $\mathbf{Z}'$ such that $I(\mathbf{Z}' \wedge \mathbf{A}) = 0$ and $I(\mathbf{Z}' \wedge \mathbf{Y}) = I(\mathbf{X} \wedge \mathbf{Y})$. Hence, the optimizer to Problem (9) must satisfy sufficiency.

The proof of ϵ-minimality is identical to the one under Problem (8).

□

B.2 Additional Theoretical Results on Augmentation Properties

The two conditions in Theorem 4.2, Condition (a) or Condition (b), requires that the augmentation $\mathbf{X}'$ is (a) sufficient and (b) (ϵ)-minimal. These two conditions are closely related to some augmentation rationales in prior papers. For example, Wang et al. [45] propose a **symmetric augmentation**, which can result in Condition (a), as formalized in Lemma B.1 below. Furthermore, Tian et al. [40] propose an "InfoMin" principle of data augmentation, that minimizes the mutual information between different views (equivalent to $\min I(\mathbf{X}, \mathbf{X}')$). We show by Lemma B.2 that this InfoMin principle leads to the above Condition (b). In contrast, our Theorem 4.2 characterizes two key conditions of augmentation and directly relate them to the optimality of the learned representation.

Lemma B.1 (Sufficiency of Augmentation)**.** *Suppose the original and augmented observations* $\mathbf{X}$ *and* $\mathbf{X}'$ *satisfy the following properties:*

$$P(\mathbf{X} = u, \mathbf{X}' = v|\mathbf{Y} = y) = P(\mathbf{X} = v, \mathbf{X}' = u|\mathbf{Y} = y), \ \forall u, v, y \tag{40a}$$

$$P(\mathbf{X} = u|\mathbf{Y} = y) = P(\mathbf{X}' = u|\mathbf{Y} = y) \tag{40b}$$

Then the augmented observation $\mathbf{X}'$ *is sufficient for the label* $\mathbf{Y}$*, i.e.,* $I(\mathbf{X}' \wedge \mathbf{Y}) = I(\mathbf{X} \wedge \mathbf{Y})$.

Proof of Lemma B.1.

$$\begin{aligned}
I(\mathbf{X}' \wedge \mathbf{Y}) &= \sum_{x,y} P(\mathbf{X}' = x, \mathbf{Y} = y) \log \frac{P(\mathbf{X}') = x, \mathbf{Y} = y)}{P(\mathbf{X}' = x)P(\mathbf{Y} = y)} && (41)\\
&= \sum_{x,y} P(\mathbf{X}' = x|\mathbf{Y} = y)P(\mathbf{Y} = y) \log \frac{P(\mathbf{X}' = x|\mathbf{Y} = y)}{\sum_{\bar{y}} P(\mathbf{X}' = x|\mathbf{Y} = \bar{y})P(\mathbf{Y} = \bar{y})} && (42)\\
&= \sum_{x,y} P(\mathbf{X} = x|\mathbf{Y} = y)P(\mathbf{Y} = y) \log \frac{P(\mathbf{X} = x|\mathbf{Y} = y)}{\sum_{\bar{y}} P(\mathbf{X} = x|\mathbf{Y} = \bar{y})P(\mathbf{Y} = \bar{y})} && (43)\\
&= \sum_{x,y} P(\mathbf{X} = x, \mathbf{Y} = y) \log \frac{P(\mathbf{X}) = x, \mathbf{Y} = y)}{P(\mathbf{X} = x)P(\mathbf{Y} = y)} && (44)\\
&= I(\mathbf{X} \wedge \mathbf{Y}) && (45)
\end{aligned}$$

where the third equation utilizes the property of symmetric augmentation. □

Lemma B.2 (Maximal Insensitivity to Nuisance)**.** *If Assumption 4.1 holds, i.e.,* $H(\mathbf{Y}|\mathbf{X}) = 0$*, the mutual information* $I(\mathbf{X}' \wedge \mathbf{X})$ *can be decomposed as*

$$I(\mathbf{X}' \wedge \mathbf{X}) = I(\mathbf{X}' \wedge \mathbf{N}) + I(\mathbf{X}' \wedge \mathbf{Y}) \tag{46}$$

Since $\mathbf{X}'$ *is sufficient, i.e.,* $I(\mathbf{X}' \wedge \mathbf{X}) = I(\mathbf{X} \wedge \mathbf{Y})$ *is a constant, minimizing* $I(\mathbf{X}' \wedge \mathbf{X})$ *is equivalent to minimizing* $I(\mathbf{X}' \wedge \mathbf{N})$.

Lemma B.2 can be obtained by a simple adaptation from Proposition 3.1 by Achille and Soatto [1].

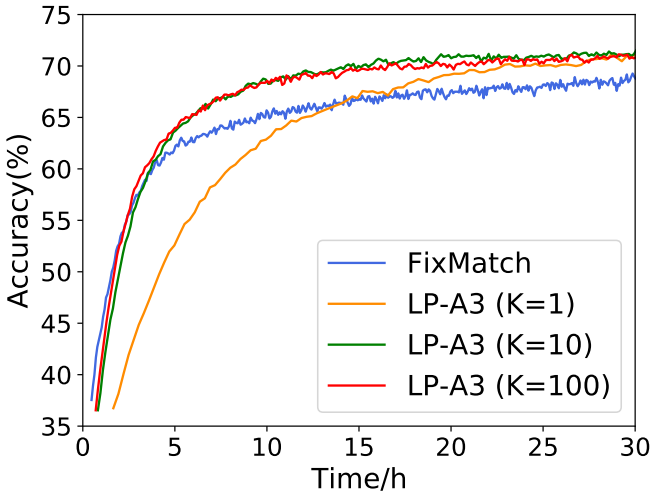


Figure 6: **Walk-clock time comparison** on CIFAR100 with 2500 labeled data.

C Experiments

C.1 Implementation Details

All codes are implemented with Pytorch[1]. To train the neural net with LP-A3 augmentation, we apply seperate batch norm layer (BN), i.e., agumented data and normal data use different BN, which is a common strategy used by previous adversarial augmentations [19, 48, 51]. The only hyperparameters for LP-A3 are label preserving margin σ and data selection ratio τ, which are tuned for each task according to the results in Sec.6.3.

Semi-supervised learning We reproduce Fixmatch [36] based on public code[2] and apply LP-A3 to it. Following [36], we used a Wide-ResNet-28-2 with 1.5M parameters for CIFAR10, WRN-28-8 for CIFAR100, and WRN-37-2 for STL-10. All the models are trained for 2^{18} iterations. σ is set to 0.002 for CIFAR10 and STL-10, and 0.02 for CIFAR100 and τ is set to be 90. Since FixMatch only apply data augmentation to those unlabeled data, here LP-A3 is also applied to those unlabeled data as data augmentation. For unlabeled data, label $\mathbf{Y}$ used in LP-A3 is the pesudo label generated by FixMatch algorithm.

Noisy-label learning We reproduce DivideMix [26] and PES [5] based on their official code[3] and apply LP-A3 to them as data augmentation. Following [26, 5], we used a ResNet-18 for CIFAR10 and CIFAR100. All the models are trained for 300 epochs. σ is set to 0.002 for CIFAR10 and 0.02 for CIFAR100, and τ is set to be 90. All the noise are symmetric noise. For noisy labeled data, label $\mathbf{Y}$ used in LP-A3 is the pesudo label generated by DivideMix or PES algorithm respectively.

Medical Image Classification Here we follow the original training and evaluation protocol of MedMNIST [4] and apply LP-A3 to the training procedure as data augmentation. ResNet-18 and ResNet-50 are trained for 100 epochs with cross-entropy loss on all the multi-class classfication subset of MedMNIST. σ is set to 0.02 and τ is tuned from $\{20, 50, 90\}$ for each dataset. The hyperparameters of RandAugment [13] is set to $N = 3, M = 5$ by following their original paper.

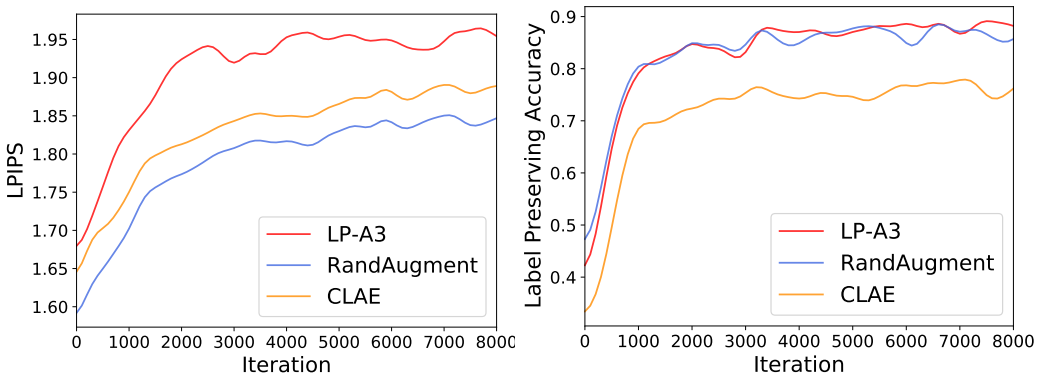
Sensitivity Analysis of Hyperparameters In Figure. 5, the experiments for semi-supervised learning are conducted on CIFAR100 with 2500 labeled data, the experiments for noisy-label learning are conducted on CIFAR100 with 80% noisy label, and the experiments for medical image classification are conducted on DermaMNIST with ResNet50.

[1] https://pytorch.org/

[2] https://github.com/kekmodel/FixMatch-pytorch

[3] https://github.com/LiJunnan1992/DivideMix, https://github.com/tmllab/PES

[4] https://github.com/MedMNIST/experiments



(a) $-I(\mathbf{X}' \wedge \mathbf{X})$ meadured by LPIPS distance (b) $I(\mathbf{X}' \wedge \mathbf{Y})$ measured by classification accuracy on x'

Figure 7: **Mutual information terms** of LP-A3, RandAugment and CLAE during training.

Table 4: Semi-supervised Learning performance on ImageNet. We eualate the performance of FixMatch and "FixMatch+LP-A3" at different training stages. Iter denotes training iterations.

	100,000 Iter	250,000 Iter	400,000 Iter
FixMatch	44.45	58.79	62.87
FixMatch+LP-A3	51.22	60.15	63.67

C.2 Computational Cost

Although Algorithm 2 is a pretty fast algorithm to solve Lagragian multiplier function, it still requires several gradient descent steps which is computationally expensive. One way to reduce computational cost is to generate LP-A3 for every few epochs. To be specific, once LP-A3 is generated, it will be saved and used to train the network for the next K epochs. When $K = 1$, it degenerates to the original Algorithm 1. The walk-clock time comparison on CIFAR100 with 2500 labeled data is give in Fig. 6, where we can see that LP-A3 ($K = 10$) and LP-A3 ($K = 100$) achieves much better accuracy than baseline within the same training time. Moreover, LP-A3 ($K = 10$) and LP-A3 ($K = 100$) achieves comparable accuracy as the original LP-A3 ($K = 1$) after convergence, indicating that LP-A3 is quite informative and it takes several epochs for the neural net to learn from it.

C.3 Mutual Information Terms of Different Data Augmentation

In order to further analyze the properties of different data augmentations, here we report the value of mutual information terms $I(\mathbf{X}' \wedge \mathbf{X})$ and $I(\mathbf{X}' \wedge \mathbf{Y})$ of training data generated by LP-A3, RandAugment and CLAE [19] (an adversarial augmentation method) during the training procedure. The results on CIFAR10 are given in Table. 7, where $-I(\mathbf{X}' \wedge \mathbf{X})$ is measured by the LPIPS distance between x and x' and $I(\mathbf{X}' \wedge \mathbf{Y})$ is measured by the label preserving accuracy, i.e., the classification accuracy of the current model on x'. We can clearly see that LP-A3 is the most different from the original data (largest LPIPS distance) and at the same time preserves the label well. Although RandAugment can also preserve label, as a pre-defined augmentation, it is the closest to the original data. Another adaptive augmentation CLAE has larger LPIPS ditance than RandAugment but cannot preserve the label well, achieving the lowest label preserving accuracy.

C.4 ImageNet Experiments

In order to validate the performance of LP-A3 on large-scale datasets, we compare FixMatch and "FixMatch+LP-A3" on ImageNet the semi-supervised learning. ResNet-50 is used as the backbone network. We randomly choose 10% data and set them as labeled data, while the remaining 90% are unlabeled data. The batch size for labeled (unlabeled) data is 64 (320). The results are reported in the Table 4, in which LP-A3 can improve FixMatch by a large margin, especially during the early stage (>6% at 100,000 iterations).

Table 5: Comparison with Mixup and Adversarial AutoAugment on the full dataset of CIFAR10.

CIFAR10	50 epochs	100 epochs	150 epochs
Mixup	87.39	93.74	96.05
Mixup+LPA3	86.82	94.73	96.12
Adversarial autoaugment	89.70	95.05	97.09
Adversarial autoaugment+LPA3	90.45	95.33	97.20

C.5 Comparison with Mixup and Adversarial AutoAugment

In this section, we compare LP-A3 with state-of-the-art data augmentation on the full dataset in a supervised manner. We select Mixup [56] and Adversarial AutoAugment [60] as the representative of sample-mixing based augmentaiton and automated data augmentation methods respectively, and we empirically compare LP-A3 with them by applying LP-A3 on top of them. The backbone model is WideResNet-28-10 and all the model are trained for 150 epochs. The results are given in Table 5. It shows that Mixup and Adversarial Autoaugment achieves very high accuracy (>96%) on CIFAR10 as they are designed for this task, but LP-A3 can further improve them, which indicates that LP-A3 is complementaray to these previous data augmentation using domain knowledge. Note in early stage the advantage of LP-A3 over Adversarial Autoaugment is especially significant ($> 0.7\%$).