# Can Pretext-Based Self-Supervised Learning Be Boosted by Downstream Data?

Weiran Huang
Huawei Noah's Ark Lab

Joint work with Jiaye Teng and Haowei He

2021 BAAI Conference

# What Is Self-Supervised Learning (SSL)?

Learning Paradigm Evolution:

- Only labeled data $\rightarrow$ Supervised Leaning
- Some labeled data + Large amounts of unlabeled data $\rightarrow$ Semi-supervised Leaning
- Only unlabeled data $\rightarrow$ Self-Supervised Learning (of unsupervised leaning)

Self-Supervised Learning (SSL) learns data representations through self-supervised tasks, and then use the learned representations for downstream prediction tasks. It has been used in computer vision [3, 9], language modeling [4, 13], graph learning [12], etc.

# SSL Approaches

There are three common approaches for SSL:

- Generative model: learning a bijective mapping between input and representation, e.g., BiGAN [6, 7], BigBiGAN [5].
- Contrastive learning: maximizing the mutual information between the features of positive samples, e.g., SimCLR [3], MoCo [9], Deep InfoMax [10].
- Pretext task: learning the representation via a handcrafted pretext task, i.e., image colorization [16], predicting image rotations [8], BERT [4].

# Theory for Self-Supervised Learning

The theoretical study of SSL is still at an early stage.

- General Analysis: Bansal et al. [2].
- Contrastive Learning Based Approach: Arora et al. [1], Tian et al. [14], and Tosh et al. [15].
- Pretext Task Based Approach: Lee et al. [11].

# Why Does Pretext-Based SSL Work?

Lee et al. [11] prove that pretext-base SSL can effectively reduce the sample complexity of downstream tasks under Conditional Independence between the components of the pretext task conditional on the downstream label.

For example, consider input variable $\boldsymbol{x}$, pretext label $\boldsymbol{z}$, and downstream label $\boldsymbol{y}$ are Gaussian variables.

- If $\boldsymbol{x} \perp \boldsymbol{z} \mid \boldsymbol{y}$, the downstream sample complexity can be reduced to $\tilde{\mathcal{O}}(\dim(\boldsymbol{y}))$.
- Otherwise, the downstream sample complexity gets worse to $\tilde{\mathcal{O}}(\dim(\boldsymbol{z}))$.

As a comparison, the sample complexity of directly using $\boldsymbol{x}$ to predict $\boldsymbol{y}$ is $\tilde{\mathcal{O}}(\dim(\boldsymbol{x}))$, where the dimension of $\boldsymbol{x}$ is supposed to be much larger than the dimension of $\boldsymbol{y}$.

# Can Pretext-Based SSL Be Boosted?

In practice, the CI condition rarely holds, and thus self-supervised learning cannot realize its full potential.

An interesting question raises:

*Can we make the CI condition hold with the help of downstream data to boost self-supervised learning?*

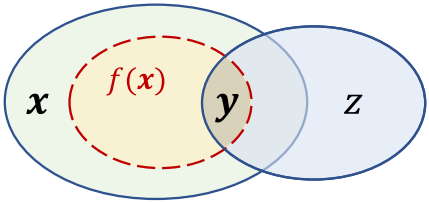



Figure: Applying a function $f$ such that $f(x) \perp z \mid y$.

# Can Pretext-Based SSL Be Boosted? (Cont.)

We introduce a function $f$ (called data processor) to refine the pretext data such that $f(\boldsymbol{x}) \perp \boldsymbol{z} \,|\, \boldsymbol{y}$ holds, and the downstream data (or extra data from downstream data distribution) are allowed to access to learn the processor.

- If such processor $f$ exists, according to the result of [11], the downstream sample complexity can be reduced by replacing all the unlabeled data $\boldsymbol{x}$ with $f(\boldsymbol{x})$.
- Otherwise, self-supervised learning cannot be boosted by the downstream data (w.r.t. the order of sample complexity).

# Intuition Is Not Alway True...

At first glance, one might think that seeing downstream data in advance would always boost downstream task performance.

- If the accessible downstream data are the extra data from downstream data distribution, then it is more likely to boost downstream task performance.

However, we show that the above intuition is NOT always true and point out that in some cases, the downstream performance will be hurt instead.

Our results validate self-supervised learning in some sense, since we prove that it is better not to use downstream data in such cases, as the standard self-supervised learning does.

# Pretext-Based SSL

Let $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^{d_x}$, $\boldsymbol{z} \in \mathcal{Z} \subset \mathbb{R}^{d_z}$, $\boldsymbol{y} \in \mathcal{Y} \subset \mathbb{R}^{d_y}$ denote the input variable, pretext label, and downstream label, respectively.

Following [11]'s formulation of SSL, a representation $\psi^* \colon \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ is first learned from the pretext task data by

$$\psi^* \triangleq \arg\min_{\psi} \mathbb{E} \left\| \psi(\boldsymbol{x}) - \boldsymbol{z} \right\|^2 .$$

Then a linear layer $W$ following $\psi^*(\boldsymbol{x})$ is learned via the downstream task, i.e.,

$$W^* \triangleq \arg\min_{W} \mathbb{E} \left\| W\psi^*(\boldsymbol{x}) - \boldsymbol{y} \right\|^2 .$$

Finally, we use $g(\cdot) = W^*\psi^*(\cdot)$ as the predictor for the downstream task.

# Pretext-Based SSL (Cont.)

The empirical representation can be written as

$$\widehat{\psi} = \arg\min_{\psi} \frac{1}{n_1} \|\psi(X_{pre}) - Z_{pre}\|^2 .$$

Similarly, the empirical linear layer learned can be written as

$$\widehat{W} = \arg\min_{W} \frac{1}{n_2} \left\| W\widehat{\psi}(X_{down}) - Y_{down} \right\|^2 .$$

The final predictor for the downstream task is $\hat{g}(\cdot) = \widehat{W}\widehat{\psi}(\cdot)$.

# The Conditions That Processor $f$ Needs to Satisfy

The following two criteria are essential for a meaningful processor $f$:

$$\mathrm{Cov}[f(\boldsymbol{x}), \boldsymbol{z} \,|\, \boldsymbol{y}] = 0, \tag{C1}$$

$$\mathbb{E}\left\|\boldsymbol{y} - W^*_{\boldsymbol{y},f(\boldsymbol{x})} f(\boldsymbol{x})\right\|^2 = \min_{f'} \mathbb{E}\left\|\boldsymbol{y} - W^*_{\boldsymbol{y},f'(\boldsymbol{x})} f'(\boldsymbol{x})\right\|^2. \tag{C2}$$

where $W^*_{\boldsymbol{y},f(\boldsymbol{x})} \triangleq \arg\min_{W \in \mathbb{R}^{d_y \times d_f}} \mathbb{E}\,\|\boldsymbol{y} - Wf(\boldsymbol{x})\|^2$ is defined as the best linear predictor of $\boldsymbol{y}$ on $f(\boldsymbol{x})$.

- (C1) is a conditional uncorrelatedness condition, which is a relaxation of the conditional independence condition.
- (C2) ensures that applying function $f$ to input variable $\boldsymbol{x}$ does not lose the information for predicting $\boldsymbol{y}$.

# The Conditions That Processor $f$ Needs to Satisfy (Cont.)

Consequences of violation:

- If processor $f$ fails to satisfy Criterion (C1), according to the result of [11], the downstream sample complexity cannot be improved.
- If processor $f$ fails to satisfy Criterion (C2), then $f(\boldsymbol{x})$ will not contain all the information for predicting $\boldsymbol{y}$, leading to a constant generalization error. Therefore, the performance of downstream task will be even worse.

Now, the question turns to be whether using extra downstream samples can make $f$ satisfy Criterion (C1) and (C2) simultaneously.

# How to Obtain $f$?

Criterion (C1) requires that $f(\boldsymbol{x})$ and $\boldsymbol{z}$ are uncorrelated conditional on $\boldsymbol{y}$. Thus, we try to minimize the following loss:

$$\mathcal{L}_1 = -\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{z}} \left\| \boldsymbol{z} - W^*_{\boldsymbol{z},f(\boldsymbol{x})} f(\boldsymbol{x}) \right\|^2.$$

For Criterion (C2), we need to guarantee that applying $f$ to $\boldsymbol{x}$ does not lose the information for predicting $\boldsymbol{y}$, indicating that $f(\boldsymbol{x})$ can still fit $\boldsymbol{y}$ well:

$$\mathcal{L}_2 = \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} \left\| \boldsymbol{y} - W^*_{\boldsymbol{y},f(\boldsymbol{x})} f(\boldsymbol{x}) \right\|^2.$$

# How to Obtain $f$? (Cont.)

Total population loss

$$\mathcal{L}(f;\mathcal{P}) \triangleq \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y})\sim\mathcal{P}} \left[ \left\| \boldsymbol{y} - W^*_{\boldsymbol{y},f(\boldsymbol{x})} f(\boldsymbol{x}) \right\|^2 - \lambda \left\| \boldsymbol{z} - W^*_{\boldsymbol{z},f(\boldsymbol{x})} f(\boldsymbol{x}) \right\|^2 \right].$$

The corresponding training loss of $\mathcal{L}(f;\mathcal{P})$ can be defined as

$$\mathcal{L}_{n_1,n_0}(f;\mathcal{P}) \triangleq \frac{1}{n_0} \left\| Y_{extra} - \widetilde{W}_2 f(X_{extra}) \right\|^2 - \frac{\lambda}{n_1} \left\| Z_{pre} - \widetilde{W}_1 f(X_{pre}) \right\|^2,$$

# Rationality of The Loss

**Theorem 1 (Loss Rationality, Informal)**

Define two sets of processors:

$$\mathcal{A}_{\mathcal{P}} = \left\{ f : f \in \arg\min_{f} \mathcal{L}(f; \mathcal{P}) \right\},$$

$$\mathcal{B}_{\mathcal{P}} = \{ f : f \text{ satisfies Criterion (C1) and Criterion (C2)} \} \neq \phi.$$

Under mild assumptions, there exist a number of population distributions $\{\mathcal{P}\}$'s such that every function in $\mathcal{A}_{\mathcal{P}}$ satisfies Criterion (C1) and Criterion (C2), by choosing a proper parameter $\lambda$, i.e.,

$$\mathbb{S} \triangleq \{ \mathcal{P} : \mathcal{A}_{\mathcal{P}} \subset \mathcal{B}_{\mathcal{P}} \} \neq \phi.$$

# Insufficient Downstream Samples Provably Fails

To better understand the role of the extra downstream samples, we consider the following loss

$$\mathcal{L}_{\infty,n_0}(f,\mathcal{P}) \triangleq \frac{1}{n_0}\left\|Y_{extra} - \widetilde{W}_2 f(X_{extra})\right\|^2 - \lambda \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{z}}\left\|\boldsymbol{z} - W^*_{\boldsymbol{z},f(\boldsymbol{x})} f(\boldsymbol{x})\right\|^2.$$

**Theorem 2 (Model-Free Lower Bound, Informal)**

Let

$$\mathcal{A}'_{\mathcal{P}} = \left\{ f : f \in \arg\min_f \mathcal{L}_{\infty,n_0}(f;\mathcal{P}) \right\},$$

$$\mathcal{B}_{\mathcal{P}} = \{f : f \text{ satisfies Criterion (C1) and Criterion (C2)}\} \neq \phi.$$

Under mild assumptions, if $n_0 = o(d_f)$, there exists a distribution $\mathcal{P}^0 \in \mathbb{S}$ (i.e., $\mathcal{A}_{\mathcal{P}^0} \subset \mathcal{B}_{\mathcal{P}^0}$), such that

$$\mathcal{A}'_{\mathcal{P}^0} \cap \mathcal{B}_{\mathcal{P}^0} = \phi.$$

# Remark of Theorem 2

- When $n_0 = o(d_f)$, even if we have infinite pretext data $(X_{pre}, Z_{pre})$, the criteria cannot be satisfied, leading to a constant generalization error. This means that the downstream performance will get worse, even if we use infinite data $(X_{down}, Y_{down})$ for downstream training. Therefore, one can conclude that the failure is due to the lack of extra downstream data $(X_{extra}, Y_{extra})$.
- Theorem 2 suggests that it is better NOT to use downstream samples when the downstream samples are insufficient, as the standard self-supervised learning usually does.
- Theorem 2 is derived based on the specific loss. What about its generality?

# More General Loss

$$\overline{\mathcal{L}}(f, \lambda;\ g_2, \rho_2, g_1, \rho_1) = \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{y}} g_2(\rho_2(\boldsymbol{y}, W^*_{\boldsymbol{y},f(\boldsymbol{x})} f(\boldsymbol{x}))) - \lambda \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{z}} g_1(\rho_1(\boldsymbol{z}, W^*_{\boldsymbol{z},f(\boldsymbol{x})} f(\boldsymbol{x}))).$$

Some constraints: $g_1$ and $g_2$ are strictly increasing functions over $[0, \infty)$, $\lambda$ is a positive penalty coefficient, $\rho_1$ and $\rho_2$ are distance metrics, and so on.

**Theorem 3 (Model-Free Lower Bound for General Loss, Informal)**

Let

$$\mathcal{A}^g_{\mathcal{P}} = \left\{ f : f \in \arg\min_f \overline{\mathcal{L}}_{\infty,n_0}(f; \mathcal{P}) \right\},$$

$$\mathcal{B}_{\mathcal{P}} = \{ f : f \text{ satisfies Criterion (C1) and Criterion (C2)} \} \neq \phi.$$

Under some assumptions, if $n_0 = o(d_f)$, there *exists* a distribution $\mathcal{P}^0$ such that

$$\mathcal{A}^g_{\mathcal{P}^0} \cap \mathcal{B}_{\mathcal{P}^0} = \phi.$$

# Model-Dependent Result

**Definition 1 (Model Capacity, Informal)**

Define the model capacity $\mathcal{M}(\mathcal{F}, \mathcal{L})$ of function class $\mathcal{F}$ with respect to loss function $\mathcal{L}$ as

$$\mathcal{M}(\mathcal{F}, \mathcal{L}) = \sup \left\{ n : \forall \mathcal{D}, \exists f \in \mathcal{F}, \sup_{(X,Y) \in \mathcal{D}^n} \mathcal{L}(f(X), Y) = 0 \right\},$$

where $\mathcal{D}$ is the data distribution, and $X, Y$ are data vectors.

- For linear functions $\mathcal{J} = \{g : g(\boldsymbol{x}) = w^\top \boldsymbol{x}, \boldsymbol{x} \in \mathbb{R}^d, w \in \mathbb{R}^{k \times d}\}$, $\mathcal{M}(\mathcal{J}, \mathcal{L}) \geq d$.
- For two-layer neural networks $\mathcal{N}$ with $k$ neurons, $\mathcal{M}(\mathcal{N}, \mathcal{L}) \geq k/4$.
- For multi-layer neural networks $\mathcal{N}_m$ with $L$ layers and $k_i$ neurons in $i$-th layer, $\mathcal{M}(\mathcal{N}_m, \mathcal{L}) \geq \min_{i \in [L]} k_i/4$.

# Model-Dependent Result (Cont.)

**Theorem 4 (Model-Dependent Lower Bound, Informal)**

Assume that the function class of the processor can be decomposed as $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2$, where there exists a function $f_2 \in \mathcal{F}_2$ such that $f_2(\boldsymbol{x}) \perp \boldsymbol{z}$. Let

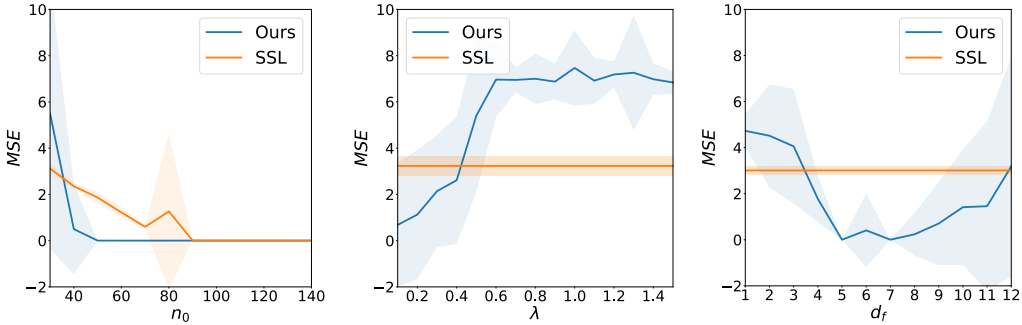
$$\mathcal{A}'_{\mathcal{P}} = \left\{ f : f \in \arg\min_{f} \mathcal{L}_{\infty, n_0}(f; \mathcal{P}) \right\},$$

$$\mathcal{B}_{\mathcal{P}} = \{ f : f \text{ satisfies Criterion (C1) and Criterion (C2)} \} \neq \phi.$$

Under some assumptions, if $n_0 = o(\mathcal{M}(\mathcal{F}_1, \mathcal{L}))$, there exists a distribution $\mathcal{P}^0 \in \mathbb{S}$ such that

$$\mathcal{A}'_{\mathcal{P}^0} \cap \mathcal{B}_{\mathcal{P}^0} = \phi.$$

# Experiments




(a) MSE under different $n_0$ (b) Large $\lambda$ hurts MSE (c) MSE under different $d_f$

Figure: Downstream performance under different hyperparameters on synthetic data.

# Experiments

| $n_0$ | 1k | 2k | 3k | 4k | 5k | SSL |
|---|---|---|---|---|---|---|
| Full | 9.95 (0.05) | 9.97 (0.07) | 9.99 (0.08) | 10.01 (0.06) | 9.96 (0.07) | 79.31 (0.09) |
| Half | 10.00 (0.07) | 10.01 (0.09) | 9.99 (0.08) | 9.99 (0.08) | 10.01 (0.09) | 75.25 (0.04) |

Table: Downstream performance on CIFAR-10. The training set is split into 5k (labeled, extra data for processor training), 30k (unlabeled), and 15k (labeled) subsets without overlap.

# Conclusion

We provably answer the question whether we can make the CI condition hold with the help of downstream data to boost self-supervised learning.

- It is better NOT to use downstream samples when they are insufficient, as the standard self-supervised learning does.
- We provide both model-free and model-dependent lower bounds of extra downstream sample size.
- Our theoretical results are verified by experiments.

# Thank you!

We are looking for research interns (Contact me for details).

# For Further Reading I


[1] Sanjeev Arora et al. "A theoretical analysis of contrastive unsupervised representation learning". 2019.

[2] Yamini Bansal et al. "For self-supervised learning, Rationality implies generalization, provably". 2020.

[3] Ting Chen et al. "A simple framework for contrastive learning of visual representations". 2020.

[4] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". 2018.

[5] Jeff Donahue and Karen Simonyan. "Large scale adversarial representation learning". 2019.

[6] Jeff Donahue et al. "Adversarial feature learning". 2017.

[7] Vincent Dumoulin et al. "Adversarially learned inference". 2017.

[8] Spyros Gidaris et al. "Unsupervised representation learning by predicting image rotations". 2018.

[9] Kaiming He et al. "Momentum contrast for unsupervised visual representation learning". 2020.

[10] R Devon Hjelm et al. "Learning deep representations by mutual information estimation and maximization". 2018.

# For Further Reading II


[11] Jason D Lee et al. "Predicting what you already know helps: Provable self-supervised learning". 2020.

[12] Zhen Peng et al. "Self-Supervised Graph Representation Learning via Global Context Prediction". 2020.

[13] Alec Radford et al. *Improving language understanding by generative pre-training*. 2018.

[14] Yuandong Tian et al. "Understanding self-supervised learning dynamics without contrastive pairs". 2021.

[15] Christopher Tosh et al. "Contrastive learning, multi-view redundancy, and linear models". 2020.

[16] Richard Zhang et al. "Colorful image colorization". 2016.