Can Pretext-Based Self-Supervised Learning Be Boosted by Downstream Data? A Theoretical Analysis

Jiaye Teng*[1], Weiran Huang*[2], Haowei He*[1]   [1]Institute for Interdisciplinary Information Sciences, Tsinghua University   [2]Huawei Noah's Ark Lab

Background: Self-Supervised Learning

Pretext-based Self-Supervised Learning

- Data format
  Pretext data (x, z): unlabeled data x and its transformation z
  Downstream data (x, y): labeled data pair with feature x and response y
- Goal: predict response y from feature x
- Procedure
  Step 1 (pretext): Learn representation ψ from pretext task samples (x, z)
  Step 2 (downstream): Perform linear regression on the pair of the learned representation and output $(\psi(x), y)$ which returns W
  The final predictor is $\hat{y} = W\psi(x)$
- Example for pretext task: colorization, inpainting, GPTs…

Conditional Independence Matters in SSL

- Conditional Independence (CI): $x \perp z \mid y$
  which means that x and z have NO common information except y.
- Theorem [Lee et al.]: Under mild assumptions and the linear regimes,
  with CI conditions, the sample complexity is $O(\dim(y))$
  without CI conditions, the sample complexity is $O(\dim(x))$
  $$O(\dim(y)) \quad \text{v.s.} \quad O(\dim(x))$$
- Intuitively, at the first step, z helps eliminate the redundant information of x, and therefore, the sample complexity required at the downstream part can be significantly reduced.

Introduce a Processor?

- Can we introduce a processor f such that $\boldsymbol{f(x) \perp z}$?
- The new procedure:
  Step 1 (processor training): use (x, z) and (x, y) to train a processor f
  Step 2 (pretext): Learn ψ from pretext task samples $(f(x), z)$
  Step 3 (downstream): Perform linear regression on the pair of the learned representation and output $(\psi(f(x)), y)$ which returns W
  The final predictor is $\hat{y} = W\psi(f(x))$
- Does the processor training work?

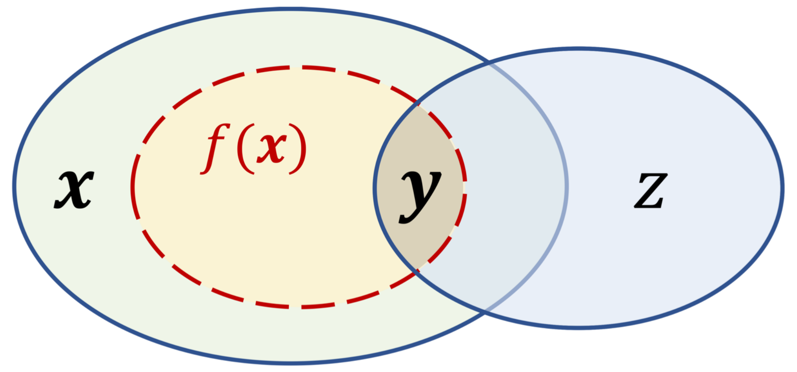

Figure 1: The common information between x and z can be redundant (the overlap part). Therefore, we introduce f such that the information between $f(x)$ and z is dense (which means that the overlap only includes y).

Main Results

Processor (f) Training

- Two criterion
  C1: $\mathrm{Cov}[f(x),\ z \mid y] = 0$ $\rightarrow f(x)$ and z have no redundant information
  C2: $f \in \arg\min \mathbb{E}\|y - W^* f(x)\|^2$ $\rightarrow f(x)$ has enough ability to predict y
  where W^* denotes the best linear predictor of y on $f(x)$.
- Training loss
  $$L(f) = \mathrm{dist}\big(y, f(x)\big) - \lambda\, \mathrm{dist}\big(z, f(x)\big)$$
  We want $f(x)$ to have enough information to predict y (minimize $\mathrm{dist}\big(y, f(x)\big)$) → C2.
  We want $f(x)$ not to have redundant information in z (maximize $\mathrm{dist}\big(z, f(x)\big)$) → C1.
- Rationality
  When we have enough downstream samples, namely, minimizing the population loss, there exist cases such that the training processor f can satisfy both C1 and C2.
- However, with limited downstream samples…

Criterion 1 & Criterion 2 cannot satisfy simultaneously with limited downstream samples

Model-free failure:

If $n = o(\dim(f))$, with mild assumptions, there exist cases such that the trained processor can only satisfy C1 **or** C2.
Notation $\dim(f)$ denotes the dimension of $f(x)$.

Model-dependent failure:

If $n = o(\mathcal{M}(\mathcal{F}))$, with some mild assumptions, there exist cases such that the trained processor can only satisfy C1 **or** C2.
Notation $\mathcal{M}(\mathcal{F})$ denotes the model capacity of the hypothesis class, which is defined as the maximal number of data points such that the function class $\mathcal{F}$ can be completely interpolated. Generally, a complex hypothesis class results in large model capacity.

Therefore, in theory, with unlimited downstream samples, the processor training works.
However, in practice, with limited downstream sample, the process training fails!

The processor Training easily fails…

- With large dimension of $f(x)$, namely $\dim(f)$, the processor training fails.
- With large model complexity, namely $\mathcal{M}(\mathcal{F})$, the processor training fails.
- With limited downstream samples, namely n_0, the processor training fails.
- With large penalty, namely λ, the processor training fails.

Experiment

Experiment results on both synthetic dataset and real-world dataset (CIFAR-10).

- **Large dimension / model capacity hurts performance.**

In the synthetic dataset (Figure 2 (a)), when $\dim(f)$ is larger, the model performance is worse. We additionally note that when $\dim(f)$ is too small, the model is underfitting.
In CIFAR-10 (Table 1), if we double the model size which indicates a larger model capacity, the model performance decreases. However, standard self-supervised leaning does not have this phenomenon.

- **Limited samples size in process-training hurts performance.**

In the synthetic dataset (Figure 2 (a)) and CIFAR-10 (Table 2), with limited downstream samples, the model performance get worse. In contrast, with enough labeled data, the performance indeed boosts.

- **Large penalty λ hurts model performance.**

When using large penalty λ, the trained processor f may eliminate useful information of y since z also contains the information of y. See Figure 2 (c) and Table 1 for more details.

- Experiments on Synthetic dataset.

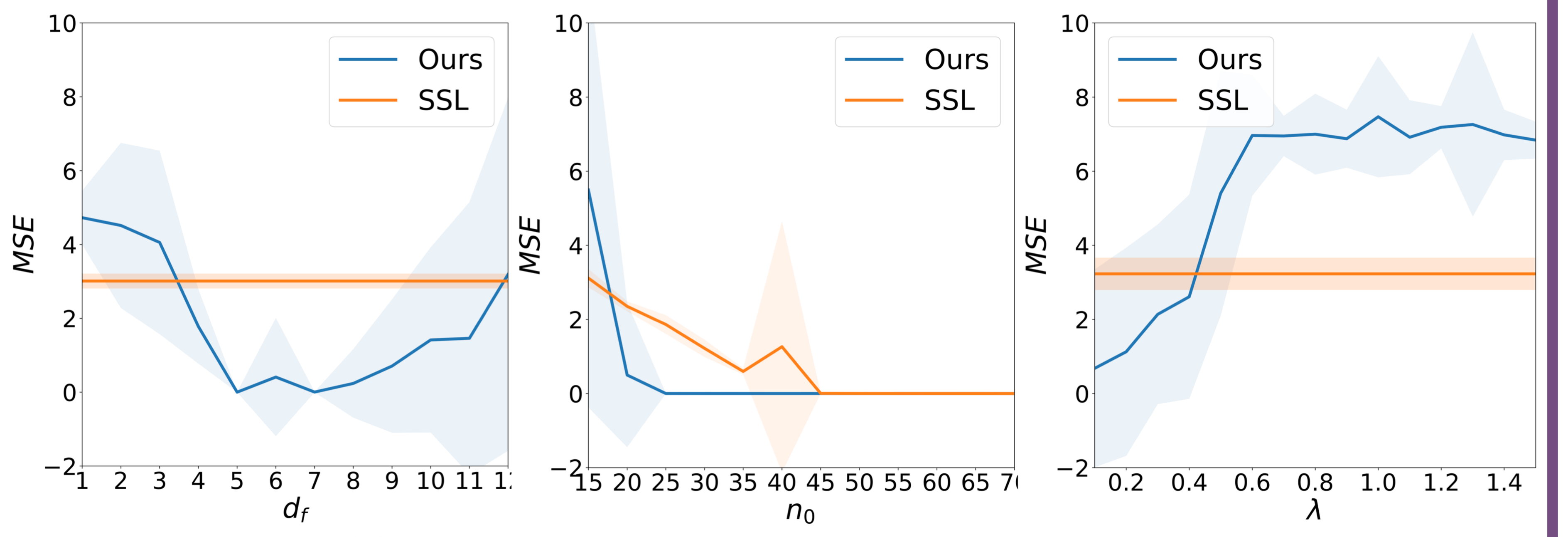

Figure 2 (a)   Figure 2 (b)   Figure 2 (c)

- Experiments on Real-world dataset (CIFAR-10).

Table 1

λ	0.001	1	10	SSL
Full	44.48 (0.84)	23.85 (6.17)	22.29 (6.21)	**74.28 (0.06)**
Double	38.72 (0.78)	26.89 (5.44)	16.86 (5.78)	**77.88 (0.10)**

Table 2

n_0	1k	5k	10k	15k	SSL
acc	42.49 (1.30)	43.38 (0.80)	44.76 (0.65)	44.48 (0.84)	**74.28 (0.06)**

Reference

[1] Lee, J. D., Lei, Q., Saunshi, N., & Zhuo, J. (2021). Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems, 34*.