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

Jiaye Teng*
IIIS, Tsinghua University

Weiran Huang*
Huawei Noah's Ark Lab

Haowei He*
IIIS, Tsinghua University

Abstract

Pretext-based self-supervised learning learns the semantic representation via a hand-crafted pretext task over unlabeled data and then uses the learned representation for downstream tasks, which effectively reduces the sample complexity of downstream tasks under Conditional Independence (CI) condition (Lee et al., 2020). However, the downstream sample complexity gets much worse if the CI condition does not hold. One interesting question is whether we can make the CI condition hold by using downstream data to refine the unlabeled data to boost self-supervised learning. At first glance, one might think that seeing downstream data in advance would always boost the downstream performance. However, we show that it is not intuitively true and point out that in some cases, it hurts the final performance instead. In particular, we prove both model-free and model-dependent lower bounds of the number of downstream samples used for data refinement. Moreover, we conduct various experiments on both synthetic and real-world datasets to verify our theoretical results.

1 INTRODUCTION

Data representations used to be learned in a supervised or semi-supervised learning way, e.g., He et al. (2016); Lee et al. (2013). Recently, self-supervised learning has drawn massive attention for its fantastic data efficiency and generalization ability, with many state-of-the-art models following this paradigm in computer vision (Chen et al., 2020; He et al., 2020), language modeling (Devlin et al., 2018; Radford et al., 2018), graph learning (Peng et al., 2020), etc. It learns data representations through self-supervised tasks and then uses the learned representations for downstream prediction tasks. Such a paradigm involves many unlabeled data that are easier to access to reduce the sample complexity of labeled data.

The renaissance of self-supervised learning began with artificially designed *pretext tasks* as self-supervised tasks, such as image colorization (Zhang et al., 2016), inpainting (Pathak et al., 2016), solving jigsaw puzzles (Noroozi and Favaro, 2016), and predicting image rotations (Gidaris et al., 2018a) in computer vision, and predicting next word (Radford et al., 2018) in natural language processing. Meanwhile, empirical evidence reveals that the representations learned from inappropriate pretext tasks may fail to transfer to downstream tasks (Yamaguchi et al., 2019; Zoph et al., 2020). Therefore, how pretext tasks affect the performance of downstream tasks is crucial to pretext-based self-supervised learning.

There are some empirical studies of the above question (e.g., Metzger et al. (2020)), however, the theoretical study is still at an early stage. A recent work (Lee et al., 2020) proves that, under conditional independence between the components of the pretext task conditional on the downstream label, the downstream task achieves minimal sample complexity. In particular, it shows that for Gaussian variables, the sample complexity of downstream tasks is reduced to $\tilde{\mathcal{O}}(\dim(\boldsymbol{y}))$[1] when Conditional Independence (CI) $\boldsymbol{x} \perp \boldsymbol{z} \mid \boldsymbol{y}$ holds, where $\boldsymbol{x}$, $\boldsymbol{z}$, $\boldsymbol{y}$ denotes the input variable, pretext label, and downstream label, respectively. 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 $\boldsymbol{y}$. However, if the CI condition $\boldsymbol{x} \perp \boldsymbol{z} \mid \boldsymbol{y}$ does not hold, the downstream sample complexity increases to

---
*Equal contribution. Correspondence to Weiran Huang <weiran.huang@outlook.com>.

---
[1]We use notation $\tilde{\mathcal{O}}$ to hide log factors in this paper, and abbreviation "dim" stands for dimension.

$\tilde{\mathcal{O}}(\dim(\boldsymbol{z}))$. This is because there is redundant mutual information between $\boldsymbol{x}$ and $\boldsymbol{z}$ which is unrelated to $\boldsymbol{y}$.

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?

To formulate the above question, we introduce a learnable function f (called data processor) to refine the pretext data such that $f(\boldsymbol{x}) \perp \boldsymbol{z} \mid \boldsymbol{y}$ holds, and the downstream data (or extra data from downstream data distribution) are allowed to be accessed to learn the processor. If such processor f exists, according to the result of Lee et al. (2020), the downstream sample complexity is 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).

At first glance, one might think that seeing downstream data in advance would always boost downstream task performance. However, we show that the above intuition is not always true and point out that the downstream performance will be hurt in some cases. Therefore, 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 standard self-supervised learning does.

Specifically, we first rigorously formulate the necessary conditions that data processor f needs to satisfy in Section 2. One is a relaxation of condition $f(\boldsymbol{x}) \perp \boldsymbol{z} \mid \boldsymbol{y}$, and the other condition ensures that $f(\boldsymbol{x})$ has enough ability to predict $\boldsymbol{y}$. We then design an ingenious loss function to learn such function f in Section 3. We demonstrate the rationality of the loss by proving that any function f minimizing the proposed loss satisfies the above criteria. Based on the proposed loss, we theoretically study the number of downstream data required. We derive a model-free lower bound in Section 4.1 and show that involving limited downstream data provably hurts model performance in the representation learning process. Furthermore, we consider the model capacity of function class and provide a model-dependent lower bound in Section 4.3. The lower bound gets higher when the model capacity gets larger, indicating that the number of downstream samples needs to match the model capacity. To verify our theoretical results, we conduct experiments on both synthetic and real-world datasets in Section 5. Due to space limitations, all proofs are deferred to the supplementary material.

In summary, our contributions are listed as follows:

- *Novelty*: We rigorously prove that learning the processor f with insufficient downstream data fails to boost self-supervised learning, which contradicts the intuitive wisdom and validates the standard self-supervised learning framework.

- *Technical Contributions*: We provide both model-free and model-dependent lower bounds for the sample size required in the training processor. Moreover, we show that more complex model classes are more likely to lead to failure.

- *Experiments*: We conduct several experiments on both synthetic datasets and real-world datasets to verify our theoretical results. We observe that training the processor with insufficient downstream data significantly decreases downstream task performance.

2 NOTATIONS AND PROBLEM FORMULATION

Data Distribution. 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, where the data are sampled from a joint distribution $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \sim \mathcal{P}$ and $d_x \geq d_z \geq d_y$. The pretext dataset with n_1 samples $(X_{pre}, Z_{pre}) \in \mathbb{R}^{d_x \times n_1} \times \mathbb{R}^{d_z \times n_1}$ and the downstream dataset with n_2 samples $(X_{down}, Y_{down}) \in \mathbb{R}^{d_x \times n_2} \times \mathbb{R}^{d_y \times n_2}$ are i.i.d. drawn from the joint distribution of $(\boldsymbol{x}, \boldsymbol{z})$ and $(\boldsymbol{x}, \boldsymbol{y})$, respectively. We assume $\mathbb{E}[\boldsymbol{x}] = \mathbb{E}[\boldsymbol{z}] = 0$ without loss of generality and denote $\mathrm{Cov}[\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{y}] \triangleq \mathbb{E}[\boldsymbol{x}\boldsymbol{z}^\top] - \mathbb{E}[\boldsymbol{x}\boldsymbol{y}^\top](\mathbb{E}[\boldsymbol{y}\boldsymbol{y}^\top])^{-1}\mathbb{E}[\boldsymbol{y}\boldsymbol{z}^\top]$ the partial covariance matrix. Unless stated otherwise, we consider the L_2-loss function in the following content[2].

Pretext-based Self-Supervised Learning. In this paper, we follow the formulation of pretext-based self-supervised learning proposed in Lee et al. (2020). The algorithm is roughly split into two steps:
- *Step 1 (pretext):* Learn representation $\widehat{\psi}$ from pretext task samples (X_{pre}, Z_{pre}).
- *Step 2 (downstream):* Perform linear regression of Y_{down} on $\widehat{\psi}(X_{down})$ and get $\widehat{W}$.

We next introduce the two steps in detail. We first learn a representation $\widehat{\psi} \colon \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ by minimizing the empirical risk[3]:

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

---

[2]When subscript is omitted, notation $\|\cdot\|$ stands for L_2-norm or Frobenius norm for vectors and matrices.

[3]When the context is clear, we abuse the notation $\hat{\psi} \colon \mathbb{R}^{d_x \times n} \to \mathbb{R}^{d_z \times n}$ over a dataset to denote $\hat{\psi} \colon \mathbb{R}^{d_x} \to \mathbb{R}^{d_z}$ over each data point in the dataset.

where $\mathcal{H}$ denotes the candidate function class (e.g., the class of multi-layer neural networks). Then a linear layer $\widehat{W} \in \mathbb{R}^{d_y \times d_z}$ following the representation $\widehat{\psi}(\cdot)$ is learned via the downstream task, i.e.,

$$\widehat{W} = \underset{W \in \mathbb{R}^{d_y \times d_z}}{\arg\min} \frac{1}{n_2} \left\| W\widehat{\psi}(X_{down}) - Y_{down} \right\|^2,$$

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

Conditional Independence. Lee et al. (2020) point out that Conditional Independence (CI) condition $\boldsymbol{x} \perp \boldsymbol{z} \mid \boldsymbol{y}$ plays crucial roles in self-supervised learning, which largely reduce the downstream sample complexity. However, without the guarantee of CI condition, the downstream sample complexity gets worse because there is much redundant information (irrelevant to the downstream task) involved in the representation during the learning via the pretext task. Therefore, for the downstream task, the mapping from the redundant features to the downstream label also needs to be learned, leading to a larger sample complexity.

The above discussion inspires us to introduce a learnable data processor f, which refines the unlabeled data such that CI condition $f(\boldsymbol{x}) \perp \boldsymbol{z} \mid \boldsymbol{y}$ holds. The processor f can help reduce the downstream sample complexity by simply replacing all the unlabeled data X with $f(X)$ in self-supervised learning. Formally, the following two criteria are essential for a meaningful processor[4] f:

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

$$f \in \arg\min_f \mathbb{E} \left\| \boldsymbol{y} - W^*_{\boldsymbol{y}, f(\boldsymbol{x})} f(\boldsymbol{x}) \right\|^2, \tag{C2}$$

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 $f(\boldsymbol{x})$.

Criterion (C1) guarantees that the processor $f(\boldsymbol{x})$ removes the redundant information which is useless for predicting $\boldsymbol{y}$. In fact, Criterion (C1) is a Conditional Uncorrelatedness (CU) condition, which is a relaxation of the CI condition. We use CU instead of CI because CU is weaker than CI when proving a negative result. In addition, setting $\mathrm{Cov}[f(\boldsymbol{x}), \boldsymbol{z} \mid \boldsymbol{y}] = 0$ as the only objective may lead to non-informative processors. For example, if $f(\boldsymbol{x})$ is an independently random noise, Criterion (C1) naturally holds, but such $f(\boldsymbol{x})$ does not carry any information of $\boldsymbol{y}$ and thus cannot be used to predict $\boldsymbol{y}$. Therefore, we need Criterion (C2) to ensure that applying function f to input variable $\boldsymbol{x}$ maintains the information for predicting $\boldsymbol{y}$.

To summarize, if processor f fails to satisfy Criterion (C1), the downstream sample complexity can-

not be improved according to the results in Lee et al. (2020). If processor f fails to satisfy Criterion (C2), $f(\boldsymbol{x})$ will not contain all the information for predicting $\boldsymbol{y}$, leading to a at least constant generalization error.

Can we learn a processor f which satisfies both Criterion (C1) and (C2)? To answer it, we first propose a provably rational loss for learning f in Section 3. Then we show that a training processor with insufficient downstream data provably fails to get a processor satisfying the above criteria simultaneously in some cases in Section 4. Moreover, we conduct experiments and observe the existence of the above phenomenon shown in Section 5.

3 LEARNING PROCESSOR f

In this section, we focus on learning the processor f satisfying both Criterion (C1) and (C2). We first propose a carefully designed loss in Section 3.1 and then give proof of its rationality in Section 3.2.

3.1 Loss Design

To learn the processor f, one needs to obtain a portion of downstream data before the pretext tasks[5] since Criterion (C1) is related to the downstream label $\boldsymbol{y}$. We split the downstream dataset (X_{down}, Y_{down}) to two folds, one with n_0 samples for processor training (X_{down1}, Y_{down1}) and the other with $n_2 - n_0$ samples for the original downstream tasks (X_{down2}, Y_{down2}).

Criterion (C1). Recall that Criterion (C1) requires that $f(\boldsymbol{x})$ and $\boldsymbol{z}$ are uncorrelated conditional on $\boldsymbol{y}$. This indicates that $f(\boldsymbol{x})$ should have less ability to predict $\boldsymbol{z}$ under linear regimes, which can be measured by loss[6]

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

where W^* is defined as the best linear projection from the second subscript to the first subscript, i.e.,

$$W^*_{\boldsymbol{b}, \boldsymbol{a}} \triangleq \arg\min_W \mathop{\mathbb{E}}_{\boldsymbol{b}, \boldsymbol{a}} \|\boldsymbol{b} - W\boldsymbol{a}\|^2.$$

If $f(\boldsymbol{x})$ can fit $\boldsymbol{z}$ well, then $\mathcal{L}_1$ will be large. Thus, minimizing $\mathcal{L}_1$ tends to pull $\boldsymbol{z}$ and $f(\boldsymbol{x})$ apart.

Criterion (C2). 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

---

[4]Assume that $\mathbb{E}[f(\boldsymbol{x})] = 0$ for simplicity.

$\boldsymbol{y}$ well. It can be formulated as

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

The smaller $\mathcal{L}_2$ indicates that $f(\boldsymbol{x})$ contains more information related to $\boldsymbol{y}$.

To minimize $\mathcal{L}_1$ and $\mathcal{L}_2$ simultaneously, we define the total population loss as $\lambda \mathcal{L}_1 + \mathcal{L}_2$, where $\lambda > 0$ is the penalty coefficient, i.e., for any data distribution $\mathcal{P}$,

$$\mathcal{L}(f;\mathcal{P}) \triangleq \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}) \sim \mathcal{P}} \left[\left\| \boldsymbol{y} - W^*_{\boldsymbol{y},f(\boldsymbol{x})} f(\boldsymbol{x}) \right\|^2 \right. \\ \left. - \lambda \left\| \boldsymbol{z} - W^*_{\boldsymbol{z},f(\boldsymbol{x})} f(\boldsymbol{x}) \right\|^2 \right]. \quad (1)$$

One might wonder that forcing $f(\boldsymbol{x})$ to fit $\boldsymbol{y}$ but not to fit $\boldsymbol{z}$ simultaneously seems to be in conflict with each other since $\boldsymbol{z}$ contains all the information of $\boldsymbol{y}$. In Section 3.2, we will give strict proof of its rationality. The intuition of such loss design is to keep the useful information for predicting $\boldsymbol{y}$, as well as remove the redundancy information unrelated to $\boldsymbol{y}$.

The corresponding training loss of $\mathcal{L}(f;\mathcal{P})$ can be defined as[7]:

$$\mathcal{L}_{n_1,n_0}(f;\mathcal{P}) \triangleq \frac{1}{n_0} \left\| Y_{down1} - \widetilde{W}_2 f(X_{down1}) \right\|^2 \\ - \frac{\lambda}{n_1} \left\| Z_{pre} - \widetilde{W}_1 f(X_{pre}) \right\|^2, \quad (2)$$

where $\widetilde{W}_1$, $\widetilde{W}_2$ represent respectively the best empirical linear predictors for the pretext data and the downstream data, and n_1, n_0 are respectively the sample sizes of pretext data (X_{pre}, Z_{pre}) and the downstream data used in process training (X_{down1}, Y_{down1}).

3.2 Rationality of Loss

We provide the following theorem to show the rationality of loss $\mathcal{L}(f;\mathcal{P})$, which states that by choosing a proper penalty coefficient λ, there exist numerous distributions such that the processor f which minimizes the population loss $\mathcal{L}(f;\mathcal{P})$ satisfies Criterion (C1) and (C2).

Theorem 1 (Rationality of Loss). *Assume that there exists a ground truth $f^* \in \mathcal{F}$ satisfying both Criterion (C1) and (C2), where $\mathcal{F}$ is the candidate function class. Assume that for each $f \in \mathcal{F}$, the matrix $\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]$ is nonsingular. Let $\mathcal{A}_\mathcal{P}$ be the set of processors that minimize the population loss under data*

distribution $\mathcal{P}$:

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

Let $\mathcal{B}_\mathcal{P}$ be the set of processor that satisfy the criteria, namely:

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

Then by choosing a proper parameter λ, 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), namely, the following defined set $\mathbb{S}$ is not empty:

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

In Theorem 1, we show that there *exist* numerous data distributions such that under those distributions, Criterion (C1) and (C2) hold when the population loss is minimized. However, we will show in the next section that for those data distributions, minimizing the corresponding empirical loss fails to satisfy the criteria when training with insufficient downstream samples. Therefore, when we do not have much knowledge about the data distribution, it is better not to use downstream data, as standard self-supervised learning does. Otherwise, we may get a worse downstream task performance.

4 INSUFFICIENT DOWNSTREAM SAMPLES PROVABLY FAILS

Theorem 1 has shown that we can learn the processor f by minimizing the population loss, such that f satisfies both Criterion (C1) and (C2). However, we can only access a finite number of downstream samples in practice. In this section, we will prove that even under the rational loss, one fails to learn a processor that satisfies both Criterion (C1) and (C2) with limited downstream data when training the processor. Specifically, we first provide a model-free lower bound in Section 4.1 as a warm-up and extend the loss to a general loss in Section 4.2. We further take the structure of function class into account and give a model-dependent lower bound in Section 4.3.

4.1 Warm-Up: Model-Free Results

To better understand the role of the downstream samples, we consider the following loss (See Eq. (3)), which differs from the population loss only in the downstream part (i.e., infinite pretext samples). Concretely, we replace the population loss of the downstream part with its empirical version.

---

[7]We again abuse the notation $f \colon \mathbb{R}^{d_x \times n} \to \mathbb{R}^{d_f \times n}$ over a dataset to denote $f \colon \mathbb{R}^{d_x} \to \mathbb{R}^{d_f}$ over each data point in the dataset.

Jiaye Teng*, Weiran Huang*, Haowei He*

$$\mathcal{L}_{\infty,n_0}(f,\mathcal{P}) \triangleq \frac{1}{n_0} \left\| Y_{down1} - \widetilde{W}_2 f(X_{down1}) \right\|^2 \tag{3}$$
$$- \lambda \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{z}} \left\| \boldsymbol{z} - W^*_{\boldsymbol{z},f(\boldsymbol{x})} f(\boldsymbol{x}) \right\|^2 .$$

We show that minimizing the above empirical loss fails to satisfy Criterion (C1) and (C2) simultaneously when $n_0 = o(d_f)$, where d_f is the dimension of $f(\boldsymbol{x})$.

Theorem 2 (Model-Free Failure). *Assume that there exists a ground truth $f^* \in \mathcal{F}$ satisfying both Criterion (C1) and (C2), where $\mathcal{F}$ is the candidate function class. And assume that there exist function $f_1 \in \mathcal{F}$ and $f_2 \in \mathcal{F}$ such that covariance $\mathrm{Cov}[f_1(\boldsymbol{x}), \boldsymbol{y}] \neq 0$ and $\mathrm{Cov}[f_2(\boldsymbol{x}), \boldsymbol{z}] = 0$. Assume that for any $f_2 \in \mathcal{F}$, the matrix $\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]$ is nonsingular. Let $\mathcal{A}'_{\mathcal{P}}$ be the set of processor that minimizes the training loss with $n_0 = o(d_f)$ downstream samples and infinite pretext samples, where d_f is the dimension of $f(\boldsymbol{x})$,*

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

And let $\mathcal{B}_{\mathcal{P}}, \mathbb{S}$ be defined as in Theorem 1. Then there exists a distribution $\mathcal{P}^0 \in \mathbb{S}$ (which means $\mathcal{A}_{\mathcal{P}^0} \subset \mathcal{B}_{\mathcal{P}^0}$), such that all the function that minimizes the loss $\mathcal{L}_{\infty,n_0}$ cannot meet Criterion (C1) and (C2) simultaneously, namely:

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

In Theorem 2, the condition $\mathrm{Cov}[f_1(\boldsymbol{x}), \boldsymbol{y}] \neq 0$ assumes that the function class of f is meaningful such that f has ability to predict y, while $\mathrm{Cov}[f_2(\boldsymbol{x}), \boldsymbol{z}] = 0$ could be easily satisfied when the function class is chosen independently with the pretext task. Theorem 2 indicates that under mild assumptions, to train a function f satisfying both Criterion (C1) and (C2), we need at least $\Omega(d_f)$ labeled downstream samples, although unlimited unlabeled data can be accessed. One can expect the larger downstream sample size requirement with finite unlabeled data.

Remark 1. 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 gets worse, even if we use infinite data (X_{down2}, Y_{down2}) for downstream training. Therefore, one can conclude that the failure is due to the lack of downstream data used in processor training (X_{down1}, Y_{down1}).

4.2 Model-Free Results for General Loss

Theorem 2 is derived based on the L_2 loss that we design in Section 3.1. One may concern the generality. In fact, we can extend the loss in Theorem 2 to a more

general one to strengthen our result and show that the failure of limited downstream samples is not caused by a specific loss. Inspired by the previously designed L_2 loss, we define the *general loss* as

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

where g_1 and g_2 are strictly increasing functions over $[0,\infty)$, and $\lambda > 0$ denotes a positive penalty coefficient. Mappings $\rho_1 \colon \mathbb{R}^{d_z} \times \mathbb{R}^{d_z} \to \mathbb{R}_{\geq 0}$ and $\rho_2 \colon \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \to \mathbb{R}_{\geq 0}$ are distance metrics, and ρ_1 is consistent with ρ_2 in subspace $\mathbb{R}^{d_y} \times \mathbb{R}^{d_y}$, i.e., for all $a, b \in \mathbb{R}^{d_y}$, $\rho_1((a, \vec{0}_{d_z - d_y}), (b, \vec{0}_{d_z - d_y})) = \rho_2(a, b)$ holds. We abuse $W^*_{\boldsymbol{y},f(\boldsymbol{x})}$ and $W^*_{\boldsymbol{z},f(\boldsymbol{x})}$ as the best linear predictors which minimize the corresponding loss $\mathbb{E} g_2(\rho_2(\boldsymbol{y}, Wf(\boldsymbol{x})))$ and $\mathbb{E} g_1(\rho_1(\boldsymbol{z}, Wf(\boldsymbol{x})))$, respectively.

It is not hard to verify that the loss $\mathcal{L}(f,\mathcal{P})$ defined in Equation (1) is a special case of the above general loss, by setting ρ_2, ρ_1 to be Euclidean distances and g_2, g_1 to be the square functions. The corresponding training loss with infinite pretext data is

$$\overline{\mathcal{L}}_{\infty,n_0}(f;\mathcal{P}) = \frac{1}{n_0} g_2 \left(\rho_2 \left(Y_{down1}, \widetilde{W}_1 f(X_{down1}) \right) \right)$$
$$- \lambda \mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{z}} g_1 \left(\rho_1 \left(\boldsymbol{z}, W^*_{\boldsymbol{z},f(\boldsymbol{x})} f(\boldsymbol{x}) \right) \right),$$

where $\widetilde{W}_1$ is the best empirical linear predictor for the downstream data[8]. We next prove that the procedure provably fails in Theorem 3.

Theorem 3 (Lower Bound for General Loss). *Assume that there exists a function f such that $f(\boldsymbol{x}) \perp \boldsymbol{z}$ holds, and $f(\boldsymbol{x}) \perp \boldsymbol{z}$ holds if and only if $\mathbb{E} g_1(\rho_1(\boldsymbol{z}, W^*_{\boldsymbol{z},f(\boldsymbol{x})} f(\boldsymbol{x})))$ reaches the maximum. For the function g_2 and g_1, assume that $g_2 g_1^{-1}$ is convex and non-decreasing. Besides, we assume that there exists a linear transformation $V \in \mathbb{R}^{d_z \times d_z}$ such that:*

$$\mathbb{E}[\rho_1(\bar{\boldsymbol{y}}, V\boldsymbol{z})] < \frac{1}{M} g_2 g_1^{-1} \max_f \mathbb{E} \left[g_1 \rho_1 \left(W^*_{V\boldsymbol{z},f(\boldsymbol{x})} f(\boldsymbol{x}), V\boldsymbol{z} \right) \right]$$
$$- \frac{1}{M} \min_f \mathbb{E} \left[g_2 \rho_2 \left(W^*_{\boldsymbol{y},f(\boldsymbol{x})} f(\boldsymbol{x}), \boldsymbol{y} \right) \right],$$

where $\bar{\boldsymbol{y}} = (\boldsymbol{y}, \vec{0}_{d_z - d_y})$ denotes the augmented vector of $\boldsymbol{y}$ with zero padding, and M denotes the upper bound of the derivative of g_2, namely, $M \geq \mathrm{d}g_2(x)/\mathrm{d}x, \forall x \in [0, \sup_{\boldsymbol{x},\boldsymbol{y}} \rho_2(W^*_{\boldsymbol{y},\hat{f}(\boldsymbol{x})} \hat{f}(\boldsymbol{x}), \boldsymbol{y}) + \sup_{\boldsymbol{y},\boldsymbol{z}} \rho_1(\bar{\boldsymbol{y}}, V\boldsymbol{z})]$.

Let $\mathcal{A}^g_{\mathcal{P}}$ denote the set of processor that minimize the general training loss with $n_0 = o(d_f)$ downstream sam-

---

[8]We again abuse the notation $g(\rho_2) \colon \mathbb{R}^{d_y \times n} \times \mathbb{R}^{d_y \times n} \to \mathbb{R}$ on a dataset to denote the summation over $g(\rho_2) \colon \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \to \mathbb{R}$ on each point in the dataset.

ples and infinite pretext samples, where d_f is the dimension of $f(\boldsymbol{x})$,

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

Let $\mathcal{B}_{\mathcal{P}}$ be defined as in Theorem 1. Then there exists a distribution $\mathcal{P}^0$ such that all the function that minimizes the loss $\overline{\mathcal{L}}_{\infty,n_0}(f;\mathcal{P})$ cannot meet Criterion (C1) and (C2) simultaneously, namely:

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

Although we use linear transformation V in Theorem 3, it can be replaced with any other forms as long as it defines a proper distance between $\boldsymbol{z}$ and $\boldsymbol{y}$.

Remark 2. The main purpose of Theorem 3 is to show that the failure in Theorem 2 does not come from the loss form. Instead, the failure indeed comes from insufficient downstream data used in processor training since various loss forms all lead to failure. Therefore, we do not provide the rationality of Theorem 3.

4.3 Model-Dependent Results

The results stated in Section 4.1 hold without any assumption of model structures (i.e., the function class of f). This section considers the model structures and provides fine-grained analysis. Before diving into the main theorem, we introduce a measure of *model capacity* at first, which is defined as follows.

Definition 1 (Model Capacity)**.** *Define the model capacity $\mathcal{M}(\mathcal{F},\mathcal{L})$ of function class $\mathcal{F}$ with respect to loss function $\mathcal{L}$ as*

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

where $\mathcal{D}$ is the data distribution, and X, Y are data vectors, each of which consists of n samples.

Intuitively, model capacity measures how well the function class fits the noise under finite samples. It is similar to the VC dimension under regression settings. In the following Lemma 4.1, we provide the lower bounds of the model capacity for linear models and neural networks.

Lemma 4.1 (Model Capacity)**.** *Let $\mathcal{J} = \{ g : g(\boldsymbol{x}) = w^\top \boldsymbol{x}, \boldsymbol{x} \in \mathbb{R}^d, w \in \mathbb{R}^{k \times d} \}$ be the class of linear functions, its model capacity satisfies*

$$\mathcal{M}(\mathcal{J},\mathcal{L}) \geq d.$$

Let $\mathcal{K}$ be the class of two-layer neural networks with k neurons, its model capacity satisfies

$$\mathcal{M}(\mathcal{K},\mathcal{L}) \geq k/4.$$

Let $\mathcal{K}_m$ be the class of multi-layer neural networks with L layers and k_i neurons in each layer i, its model capacity satisfies

$$\mathcal{M}(\mathcal{K}_m,\mathcal{L}) \geq \min_{i \in [L]} k_i/4.$$

We next provide the model-dependent lower bound in Theorem 4:

Theorem 4 (Model-Dependent Lower Bound)**.** *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 $f_0 \in \mathcal{F}$ such that $\text{Cov}[f_0(\boldsymbol{x}), \boldsymbol{y}] \neq 0$ and a function $f_2 \in \mathcal{F}_2$ such that $f_2(\boldsymbol{x}) \perp \boldsymbol{z}$. Assume matrix $\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]$ is nonsingular for all $f \in \mathcal{F}$. Let $\mathcal{A}'_{\mathcal{P}}$ be the set of processor that minimize the loss with $n_0 = o(\mathcal{M}(\mathcal{F}_1, \mathcal{L}))$ downstream samples and infinite pretext samples, where d_f is the dimension of f,*

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

And let $\mathcal{B}_{\mathcal{P}}, \mathbb{S}$ be defined as in Theorem 1. Then there exists a distribution $\mathcal{P}^0 \in \mathbb{S}$ (which means $\mathcal{A}_{\mathcal{P}^0} \subset \mathcal{B}_{\mathcal{P}^0}$), such that all the function that minimizes the loss $\mathcal{L}_{\infty,n_0}$ cannot meet Criterion (C1) and (C2) simultaneously, namely:

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

The function class $\mathcal{F}$ is decomposed into two parts: The inner one $\mathcal{F}_2$ is the minimal class of functions in which there exists a function that can remove all the information of $\boldsymbol{z}$, and the outer one $\mathcal{F}_1$'s model capacity decides the lower bound of the downstream sample size. Usually, the information of $\boldsymbol{x}$ is much more than the information of $\boldsymbol{z}$, and therefore, $\mathcal{F}_2$ is expected to be not complex.

Intuitively, Theorem 4 shows that with a more complex function class to train the processor, the lower bound for downstream samples is higher. Note that the model capacity is infinity for the neural network with infinite width. Therefore, the procedure provably fails due to the large model capacity.

Remark 3. The model-dependent bound (Theorem 4) and the model-free bound (Theorem 2) focus on the processor training and the pretext step, respectively. Therefore, one cannot reach the conclusion that Theorem 4 dominate Theorem 2. However, since we usually use neural networks to train processor f, which usually suffers from a large model capacity, Theorem 4 is better than Theorem 2 with large model capacity.

5 EXPERIMENTS

In this section, we conduct experiments on synthetic data and real-world data, mainly to show that provid-

ing few downstream samples hurts the model performance. In each experiment, we run each experiment 5 times repeatedly and calculate its mean and 95% confidence bands. We defer the experiment details to the supplementary materials.

5.1 Synthetic Data

We validate the statements proved before in this section, showing that the processor-based learning fails when (a) training processor with insufficient downstream samples or (b) the penalty parameter is large.

Setup. We consider a linear regime, where x is generated from Gaussian distribution, $z = x[0 : d_z - 1] + \epsilon_z$ and $y = x[0 : d_y - 1] + \epsilon_y$, where ϵ_z, ϵ_y are generated from Gaussian distributions. Figure 1 summarizes the experiments (blue and yellow) with their 95% confidence bands (light blue and light yellow). We refer to supplementary materials for more details.

Algorithm and Metrics. We run our newly proposed algorithm (labeled as Ours), which first trains the processor and then does pretext and downstream tasks. We denote d_f as the coarse representation dimension and λ as the penalty coefficient in the loss (See Equation (1)). We also run the standard self-supervised learning (labeled as SSL). In terms of metrics, we calculate the MSE on test downstream samples as the model performance. Intuitively, MSE is small when the features are learned well.

Analysis. We plot the results in Figure 1 and defer the specific statistics in the supplementary materials due to space limitations. Firstly, we plot the MSE as the number of samples used in processor training n_0 varies from 30 to 140 in Figure 1(a). With large n_2, the downstream tasks benefit from small MSE, showing that the algorithm indeed finds the proper coarse representations (demonstrated in Theorem 1). Secondly, we test the case λ varies from 0.1 to 1.5, as plotted in Figure 1(b). When λ is large, the algorithm suffers from unsatisfying performances as demonstrated in Theorem 1. Thirdly, we test the case d_f varies from 1 to 12, and plot them in Figure 1(c).

5.2 Real-World Data (CIFAR-10)

In this section, we conduct experiments on CIFAR-10 (Krizhevsky et al., 2009) and show that the newly proposed processor-based method indeed fails in real-world tasks. We demonstrate that the performance gets worse with (a) larger penalty parameter λ, (b) larger model capacity, and (c) training processors with insufficient downstream samples.

Setup. We conduct experiments on CIFAR-10 and choose rotation prediction (Gidaris et al., 2018b) as the

Table 1: **Large lambda, large model capacity hurt the performance** with 15k downstream data used in processor training. The newly proposed processor-training method gets worse with larger model, while SSL becomes better.

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: **Training processors with insufficient downstream data hurts the performance.** With more samples used in processor training (n_0), the performance becomes better. However, they do not exceed standard SSL in CIFAR-10.

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)**

baseline SSL framework. As previously described, the training set is split into pretext samples (unlabeled) and downstream samples (labeled) without overlap to mimic the SSL training and a downstream classification task.

The baseline follows Gidaris et al. (2018b), where we train a plain rotation prediction task using 30k unlabeled training samples on a four-block NIN model (Lin et al., 2014). Then we treat the 15k labeled training samples and 10k test samples as the training and test sets of the downstream classification task. A linear classifier is learned to conduct the task[9]. For the processor-based learning, the only difference is that we now specify f in Equation (1) as the first two blocks in a NIN model. When optimizing f, the 5k labeled and 30k unlabeled training samples are used to minimize Loss in Equation (1). After obtaining f, we fix the weights of the first two blocks in the NIN model and go through the SSL training pipeline again, same as above.

Algorithm and Metrics. We use the test accuracy (acc) to evaluate the model performance. We run the newly proposed processor training techniques with different λ (see Equation (1)) and different model capacity (see Definition 1) in Table 1. We label the original model as *Full* and label the double-size model as *Double*. Obviously, Double has a larger model capacity compared to Full. Besides, we run the newly proposed

---

[9]To fully take advantage of the downstream samples, we require that the sum of downstream samples used in processor training and the downstream samples used in the downstream task are 20k.

(a) MSE under different n_0 (b) MSE under different λ (c) MSE under different d_f
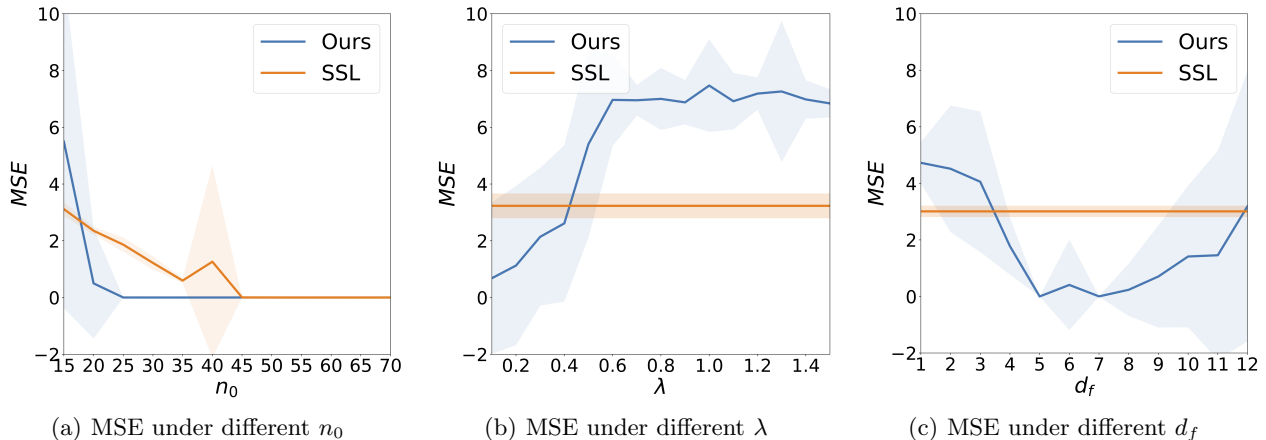
Figure 1: **Algorithm performance under different hyperparameters on synthetic data.** (a) Our algorithm learns better with sufficient downstream samples (large n_0). MSE decreases to zero as downstream samples increase. (b) A large penalty forces the coarse representation to abandon the $\boldsymbol{y}$ information, leading to a large MSE. (c) When d_f is small, the model underfits; when d_f is large, the model suffers from a limited number of downstream samples.

method under different downstream samples used in processor training in Table 2.

Analysis. We list the results in Table 1 and Table 2 and defer the specific statistics in the supplementary materials. Firstly, we test the role of λ in Table 1, demonstrating that large λ indeed hurts the model performance (as shown in Theorem 1). Secondly, we validate Theorem 2 in Table 2, showing that a training processor with fewer downstream samples X_{down1} results in worse performance. We finally test the statement in Theorem 4 by Table 1, showing that when double the model size, the performance gets much worse.

6 RELATED WORKS

Self-Supervised Methods in Practice. There are three common approaches for Self-Supervised Learning (SSL): generative model based, contrastive learning based, and pretext based. Generative model based SSL (Donahue et al., 2017; Donahue and Simonyan, 2019; Dumoulin et al., 2017) learns a bijective mapping between input and representation. Contrastive learning based SSL learns representations by maximizing the mutual information between the global and local features (Bachman et al., 2019; Hjelm et al., 2018; Oord et al., 2018) or between the features of positive samples (Chen et al., 2020; He et al., 2020; Tian et al., 2019). Pretext based SSL learns representations via handcrafted pretext tasks (Gidaris et al., 2018a; Noroozi and Favaro, 2016; Pathak et al., 2016; Zhang et al., 2016). These three approaches are quite different in technique. For example, contrastive learning

based approaches are trained by comparing positive (and negative) pairs, while pretext-based approaches assign a label for each unlabeled sample. Such difference leads to each sample with a pretext label loss in pretext-based SSL, while in contrastive learning-based SSL, a batch of unlabeled samples produces a loss. In this paper, we focus on studying the pretext-based SSL.

Theory for Self-Supervised Learning. Although there are a number of great empirical works for SSL, the theoretical study of SSL is still at an early stage. The most related work to ours is given by Lee et al. (2020). They are the first to formulate pretext-based SSL, and show that it can reduce the sample complexity of downstream tasks compared with supervised learning. They also point out that when the CI condition holds, the downstream sample complexity achieves the optimal, and it gets worse when the CI condition does not hold. In this paper, we further study the situation that the CI condition does not hold and explore the idea of applying a learnable function to the input to make the CI condition hold. Furthermore, Saunshi et al. (2021) study the specific pretext task of next word prediction and the downstream task of text classification. Other works (Huang et al., 2021; Saunshi et al., 2019; Tian et al., 2021; Wang et al., 2022) study the generalization error of contrastive learning based SSL, whose setting is different from our paper. Moreover, Bansal et al. (2020) analyze the generalization gap for most SSL methods. However, the rationality gap, which is a part of the generalization gap, cannot be theoretically bounded.

7 FUTURE WORK

In this work, we explore the idea of using part of downstream data to boost the pretext-based self-supervised learning by making the conditional independence $f(\boldsymbol{x}) \perp \boldsymbol{z} \mid \boldsymbol{y}$ hold. We show that taking limited downstream data provably hurts the performance and give both model-free and model-dependent lower bounds of sample size. One possible future work is to rigorously prove that the pretext based self-supervised learning can be boosted with sufficient downstream data, suggested by the experiments (Figure 1). Moreover, one can consider generalizing this paper to the contrastive learning regime to see if involving part of downstream information in the pre-training can boost the downstream performance.

Acknowledgements

References

Bachman, P., Hjelm, R. D., and Buchwalter, W. (2019). Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15535–15545.

Bansal, Y., Kaplun, G., and Barak, B. (2020). For self-supervised learning, rationality implies generalization, provably. *arXiv preprint arXiv:2010.08508*.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Donahue, J., Krähenbühl, P., and Darrell, T. (2017). Adversarial feature learning. In *International Conference on Learning Representations (ICLR)*.

Donahue, J. and Simonyan, K. (2019). Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, pages 10542–10552.

Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., and Courville, A. (2017). Adversarially learned inference. In *International Conference on Learning Representations (ICLR)*.

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

Gidaris, S., Singh, P., and Komodakis, N. (2018b). Unsupervised representation learning by predicting image rotations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2018). Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Huang, W., Yi, M., and Zhao, X. (2021). Towards the generalization of contrastive self-supervised learning. *arXiv preprint arXiv:2111.00743*.

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

Lee, D.-H. et al. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3.

Lee, J. D., Lei, Q., Saunshi, N., and Zhuo, J. (2020). Predicting what you already know helps: Provable self-supervised learning. *arXiv preprint arXiv:2008.01064*.

Lin, M., Chen, Q., and Yan, S. (2014). Network in network. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Metzger, S., Srinivas, A., Darrell, T., and Keutzer, K. (2020). Evaluating self-supervised pretraining without using labels. *arXiv preprint arXiv:2009.07724*.

Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer.

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

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544.

Peng, Z., Dong, Y., Luo, M., Wu, X.-M., and Zheng, Q. (2020). Self-supervised graph representation learning via global context prediction. *arXiv preprint arXiv:2003.01604*.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Saunshi, N., Malladi, S., and Arora, S. (2021). A mathematical exploration of why language models help solve downstream tasks. In *International Conference on Learning Representations*.

Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., and Khandeparkar, H. (2019). A theoretical analysis of contrastive unsupervised representation learning. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5628–5637. PMLR.

Tian, Y., Chen, X., and Ganguli, S. (2021). Understanding self-supervised learning dynamics without contrastive pairs. *arXiv preprint arXiv:2102.06810*.

Tian, Y., Krishnan, D., and Isola, P. (2019). Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*.

Wang, Y., Zhang, Q., Wang, Y., Yang, J., and Lin, Z. (2022). Chaos is a ladder: A new understanding of contrastive learning. In *International Conference on Learning Representations*.

Yamaguchi, S., Kanai, S., Shioda, T., and Takeda, S. (2019). Multiple pretext-task for self-supervised learning via mixing multiple image transformations. *arXiv preprint arXiv:1912.11603*.

Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer.

Zoph, B., Ghiasi, G., Lin, T.-Y., Cui, Y., Liu, H., Cubuk, E. D., and Le, Q. (2020). Rethinking pre-training and self-training. *Advances in Neural Information Processing Systems*, 33.

Jiaye Teng*, Weiran Huang*, Haowei He*

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

We give all the proofs of lemmas and theorems here organized by theorems, i.e., Section A for Theorem 1 and related lemmas, Section B for Theorem 2, Section C for Theorem 3, and Section D for Theorem 4.

A   Proof of Theorem 1

Theorem 1 (Rationality of Loss). *Assume that there exists a ground truth $f^* \in \mathcal{F}$ satisfying both Criterion* (C1) *and* (C2)*, where $\mathcal{F}$ is the candidate function class. Assume that for each $f \in \mathcal{F}$, the matrix $\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]$ is nonsingular. Let $\mathcal{A}_\mathcal{P}$ be the set of processors that minimize the population loss under data distribution $\mathcal{P}$:*

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

Let $\mathcal{B}_\mathcal{P}$ be the set of processor that satisfy the criteria, namely:

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

Then by choosing a proper parameter λ, 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)*, namely, the following defined set $\mathbb{S}$ is not empty:*

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

To prove Theorem 1, we need to construct a data distribution $\mathcal{P}_0$ such that $\mathcal{P}_0 \in \mathbb{S}$. We construct such distribution as follows: Let $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ be the input variables, pretext label, and downstream label, respectively. Firstly, let $\mathbb{E}[\boldsymbol{x}] = \mathbb{E}[\boldsymbol{y}] = \mathbb{E}[\boldsymbol{z}] = 0$ and $\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top] = I$. Given a marginal distribution of $(\boldsymbol{x}, \boldsymbol{y})$, construct a linear relationship $\boldsymbol{y} = B\boldsymbol{z}$ where the singular values of matrix B be $\sigma_1 = \cdots = \sigma_{d_y} = \sigma$. And choose the penalty $\lambda < \sigma^2$.

We first take a closer look at the loss function $\mathcal{L}(f) = \mathcal{L}_2 + \lambda \mathcal{L}_1$. We can rewrite it as $\mathcal{L}(f) = (1 - \frac{\lambda}{\sigma^2})\mathcal{L}_2 + \lambda(\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2)$. The first term is $\mathcal{L}_2$ multiplied by a coefficient, which still captures the information of $\boldsymbol{y}$ as $\mathcal{L}_2$ does. The second term intuitively captures the redundant information of $\boldsymbol{z}$.

We first provide Lemma A.1 and Lemma A.2 used during the proof of Theorem 1. Lemma A.1 shows that $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$ is minimized if and only if all the redundant information of $\boldsymbol{z}$ is eliminated.

Lemma A.1. *Under the assumptions of Theorem 1 and data distribution $\mathcal{P}_0$ with penalty $\lambda < \sigma^2$, then $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$ is minimized if and only if the conditional independence criterion* (C1) *holds, i.e., $\mathrm{Cov}[f(\boldsymbol{x}), \boldsymbol{z} \,|\, \boldsymbol{y}] = 0$.*

The advantage of decomposing $\mathcal{L}(f)$ into $(1 - \frac{\lambda}{\sigma^2})\mathcal{L}_2$ and $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$ is that these two terms can be optimized individually. In other words, when $\mathcal{L}(f)$ is minimized, the above two terms are also minimized. We state it formally as the following Lemma A.2.

Lemma A.2. *Under the assumptions of Theorem 1 and data distribution $\mathcal{P}_0$ with penalty $\lambda < \sigma^2$, if $\mathcal{L}(f)$ is minimized, then $(1 - \frac{\lambda}{\sigma^2})\mathcal{L}_2$ and $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$ are both minimized.*

According to Lemma A.2, for each function f that minimizes the loss, it also minimizes $(1 - \frac{\lambda}{\sigma^2})\mathcal{L}_2$ and $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$ at the same time. For the first term, since $1 - \frac{\lambda}{\sigma^2}$ is a positive coefficient, $\mathcal{L}_2$ is minimized. Therefore, Criterion (C2) holds. Since the second term is minimized, Criterion (C1) holds by Lemma A.1. Thus, such f satisfies both Criterion (C1) and (C2). We next prove Lemma A.1 and Lemma A.2 in Section A.1 and Section A.2, respectively,

A.1   Proof of Lemma A.1

Lemma A.1. *Under the assumptions of Theorem 1 and data distribution $\mathcal{P}_0$ with penalty $\lambda < \sigma^2$, then $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$ is minimized if and only if the conditional independence criterion (C1) holds, i.e., $\mathrm{Cov}[f(\boldsymbol{x}), \boldsymbol{z} \mid \boldsymbol{y}] = 0$.*

Proof. Criterion (C1) is equivalent to

$$\Sigma_{f(\boldsymbol{x}),\boldsymbol{z}|\boldsymbol{y}} = \Sigma_{\boldsymbol{z}f(\boldsymbol{x})} - \Sigma_{\boldsymbol{y}f(\boldsymbol{x})}\Sigma_{\boldsymbol{yy}}^{-1}\Sigma_{\boldsymbol{yz}} = 0.$$

Notice that $\mathbb{E}[\boldsymbol{y}] = \mathbb{E}[\boldsymbol{z}] = \mathbb{E}[f(\boldsymbol{x})] = 0$, thus the above equation can be rewritten as

$$\mathbb{E}[f(\boldsymbol{x})\boldsymbol{z}^\top] = \mathbb{E}[f(\boldsymbol{x})\boldsymbol{y}^\top]\left(\mathbb{E}[\boldsymbol{yy}^\top]\right)^{-1}\mathbb{E}[\boldsymbol{yz}^\top]. \tag{4}$$

On the other hand, we express the term $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$ as

$$
\begin{aligned}
&\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2 \\
&= \frac{1}{\sigma^2}\mathbb{E}\left\|\boldsymbol{y} - W_{\boldsymbol{y},f(\boldsymbol{x})}^* f(\boldsymbol{x})\right\|^2 - \mathbb{E}\left\|\boldsymbol{z} - W_{\boldsymbol{z},f(\boldsymbol{x})}^* f(\boldsymbol{x})\right\|^2 \\
&= \mathrm{tr}[\frac{1}{\sigma^2}\mathbb{E}\left[\left(\boldsymbol{y} - W_{\boldsymbol{y},f(\boldsymbol{x})}^* f(\boldsymbol{x})\right)\left(\boldsymbol{y} - W_{\boldsymbol{y},f(\boldsymbol{x})}^* f(\boldsymbol{x})\right)^\top\right] \\
&\qquad - \mathbb{E}\left[\left(\boldsymbol{z} - W_{\boldsymbol{z},f(\boldsymbol{x})}^* f(\boldsymbol{x})\right)\left(\boldsymbol{z} - W_{\boldsymbol{z},f(\boldsymbol{x})}^* f(\boldsymbol{x})\right)^\top\right]] \\
&= \mathrm{tr}[\frac{1}{\sigma^2}\left(\mathbb{E}[\boldsymbol{yy}^\top] - \mathbb{E}\left[\boldsymbol{y}f^\top(\boldsymbol{x})\right]\left(\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]\right)^{-1}\mathbb{E}\left[f(\boldsymbol{x})\boldsymbol{y}^\top\right]\right) \\
&\qquad - \left(\mathbb{E}[\boldsymbol{zz}^\top] - \mathbb{E}\left[\boldsymbol{z}f^\top(\boldsymbol{x})\right]\left(\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]\right)^{-1}\mathbb{E}\left[f(\boldsymbol{x})\boldsymbol{z}^\top\right]\right)] \\
&= \mathrm{tr}[\mathbb{E}\left[\boldsymbol{z}f^\top(\boldsymbol{x})\right]\left(\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]\right)^{-1}\mathbb{E}\left[f(\boldsymbol{x})\boldsymbol{z}^\top\right] \\
&\qquad - \frac{1}{\sigma^2}\mathbb{E}\left[\boldsymbol{y}f^\top(\boldsymbol{x})\right]\left(\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]\right)^{-1}\mathbb{E}\left[f(\boldsymbol{x})\boldsymbol{y}^\top\right]] + \left(\frac{1}{\sigma^2}\mathbb{E}[\boldsymbol{y}^\top\boldsymbol{y}] - \mathbb{E}[\boldsymbol{z}^\top\boldsymbol{z}]\right) \\
&= \mathrm{tr}\left[\left(I - \frac{1}{\sigma^2}B^\top B\right)\mathbb{E}\left[\boldsymbol{z}f^\top(\boldsymbol{x})\right]\left(\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]\right)^{-1}\mathbb{E}\left[f(\boldsymbol{x})\boldsymbol{z}^\top\right]\right] + \left(\frac{1}{\sigma^2}\mathbb{E}[\boldsymbol{y}^\top\boldsymbol{y}] - \mathbb{E}[\boldsymbol{z}^\top\boldsymbol{z}]\right),
\end{aligned}
$$

where the third equation holds because $W_{\boldsymbol{y},f(\boldsymbol{x})}^* = \mathbb{E}[\boldsymbol{y}f^\top(\boldsymbol{x})]\left(\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]\right)^{-1}$ and the last equation follows the assumption that $\boldsymbol{y} = B\boldsymbol{z}$. Notice that the second term $\frac{1}{\sigma^2}\mathbb{E}\left[\boldsymbol{y}^\top\boldsymbol{y}\right] - \mathbb{E}\left[\boldsymbol{z}^\top\boldsymbol{z}\right]$ is unrelated to f. Therefore, minimizing $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$ is equivalent to minimizing

$$T := \mathrm{tr}\left[\left(I - \frac{1}{\sigma^2}B^\top B\right)\mathbb{E}\left[\boldsymbol{z}f^\top(\boldsymbol{x})\right]\left(\mathbb{E}\left[f(\boldsymbol{x})f^\top(\boldsymbol{x})\right]\right)^{-1}\mathbb{E}\left[f(\boldsymbol{x})\boldsymbol{z}^\top\right]\right]. \tag{5}$$

(a) We first prove the **sufficient condition**, namely, the conditional independence criterion (C1) leads to minimizing $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$.

Plugging Equation (4) to Equation (5), we have

$$
\begin{aligned}
T &= \mathrm{tr}\left[\left(I - \frac{1}{\sigma^2}B^\top B\right)\mathbb{E}\left[\boldsymbol{z}f^\top(\boldsymbol{x})\right]\left(\mathbb{E}\left[f(\boldsymbol{x})f^\top(\boldsymbol{x})\right]\right)^{-1}\mathbb{E}\left[f(\boldsymbol{x})\boldsymbol{z}^\top\right]\right] \\
&= \mathrm{tr}\left[\mathbb{E}\left[f(\boldsymbol{x})\boldsymbol{z}^\top\right]\left(I - \frac{1}{\sigma^2}B^\top B\right)\mathbb{E}\left[\boldsymbol{z}f^\top(\boldsymbol{x})\right]\left(\mathbb{E}\left[f(\boldsymbol{x})f^\top(\boldsymbol{x})\right]\right)^{-1}\right] \\
&= \mathrm{tr}\left[\mathbb{E}\left[f(\boldsymbol{x})\boldsymbol{y}^\top\right]\left(\mathbb{E}[\boldsymbol{yy}^\top]\right)^{-1}\mathbb{E}[\boldsymbol{yz}^\top]\left(I - \frac{1}{\sigma^2}B^\top B\right)\left(\mathbb{E}\left[f(\boldsymbol{x})\boldsymbol{y}^\top\right]\left(\mathbb{E}[\boldsymbol{yy}^\top]\right)^{-1}\mathbb{E}[\boldsymbol{yz}^\top]\right)^\top\left(\mathbb{E}\left[f(\boldsymbol{x})f^\top(\boldsymbol{x})\right]\right)^{-1}\right].
\end{aligned}
$$

By assumption that $\boldsymbol{y} = B\boldsymbol{z}$ and $\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top] = I$, we have

$$\mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{y}^\top\right]\left(\mathbb{E}[\boldsymbol{y}\boldsymbol{y}^\top]\right)^{-1}\mathbb{E}[\boldsymbol{y}\boldsymbol{z}^\top]\left(I - \frac{1}{\sigma^2}B^\top B\right)\left(\mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{y}^\top\right]\left(\mathbb{E}[\boldsymbol{y}\boldsymbol{y}^\top]\right)^{-1}\mathbb{E}[\boldsymbol{y}\boldsymbol{z}^\top]\right)^\top$$

$$= \mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]B^\top\left(\mathbb{E}[B\boldsymbol{z}\boldsymbol{z}^\top B^\top]\right)^{-1}B\,\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top]\left(I - \frac{1}{\sigma^2}B^\top B\right)\left(\mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]B^\top\left(\mathbb{E}[B\boldsymbol{z}\boldsymbol{z}^\top B^\top]\right)^{-1}B\,\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top]\right)^\top$$

$$= \mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]B^\top[BB^\top]^{-1}B\left(I - \frac{1}{\sigma^2}B^\top B\right)\left(\mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]B^\top[BB^\top]^{-1}B\right)^\top$$

$$= \mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]B^\top[BB^\top]^{-1}\left[BB^\top - \frac{1}{\sigma^2}BB^\top BB^\top\right][BB^\top]^{-1}B\,\mathbb{E}\left[\boldsymbol{z}f\left(\boldsymbol{x}\right)^\top\right]$$

$$= \mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]B^\top[BB^\top]^{-1}\cdot 0_{d_y \times d_y}\cdot[BB^\top]^{-1}B\,\mathbb{E}\left[\boldsymbol{z}f\left(\boldsymbol{x}\right)^\top\right]$$

$$= 0.$$

Furthermore, notice that the eigenvalues of $I - \frac{1}{\sigma^2}B^\top B$ is no less than zero by the definition of σ. And notice that the eigenvalues of $\mathbb{E}\left[\boldsymbol{z}f^\top\left(\boldsymbol{x}\right)\right]\left(\mathbb{E}\left[f\left(\boldsymbol{x}\right)f^\top\left(\boldsymbol{x}\right)\right]\right)^{-1}\mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]$ is also no less than zero. Therefore, T reaches the minimum when it reaches zero. To conclude, the sufficient condition holds.

(b) We next prove the **necessary condition**, namely, minimizing $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$ leads to the conditional independence.

Under the assumption that there exists a ground truth f^* satisfying Criterion (C1) and (C2), we see from the sufficient condition that when T reaches the minimum, T must be equal to zero:

$$T = \mathrm{tr}\left[\left(I - \frac{1}{\sigma^2}B^\top B\right)\mathbb{E}\left[\boldsymbol{z}f^\top\left(\boldsymbol{x}\right)\right]\left(\mathbb{E}\left[f\left(\boldsymbol{x}\right)f^\top\left(\boldsymbol{x}\right)\right]\right)^{-1}\mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]\right] = 0.$$

Since the matrix is semi-definite, we can omit the trace term and rewrite it as:

$$\mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]\left(I - \frac{1}{\sigma^2}B^\top B\right)\mathbb{E}\left[\boldsymbol{z}f^\top\left(\boldsymbol{x}\right)\right]\left(\mathbb{E}\left[f\left(\boldsymbol{x}\right)f^\top\left(\boldsymbol{x}\right)\right]\right)^{-1} = 0.$$

Besides, since $\mathbb{E}\left[f\left(\boldsymbol{x}\right)f^\top\left(\boldsymbol{x}\right)\right]$ is non-singular, we have

$$\mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]\left(I - \frac{1}{\sigma^2}B^\top B\right)\mathbb{E}\left[\boldsymbol{z}f^\top\left(\boldsymbol{x}\right)\right] = 0.$$

On the other hand, we represent the covariance as the follows based on the assumption $\mathbb{E}\left[\boldsymbol{z}\boldsymbol{z}^\top\right] = I$,

$$\Sigma_{f(\boldsymbol{x}),\boldsymbol{z}|\boldsymbol{y}} = \mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right] - \mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{y}^\top\right]\left(\mathbb{E}\left[\boldsymbol{y}\boldsymbol{y}^\top\right]\right)^{-1}\mathbb{E}\left[\boldsymbol{y}\boldsymbol{z}^\top\right] = \mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]\left[I - \frac{1}{\sigma^2}B^\top B\right].$$

Finally, the covariance meets

$$\Sigma_{f(\boldsymbol{x}),\boldsymbol{z}|\boldsymbol{y}}\Sigma_{f(\boldsymbol{x}),\boldsymbol{z}|\boldsymbol{y}}^\top = \mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]\left[I - \frac{1}{\sigma^2}B^\top B\right]\mathbb{E}\left[\boldsymbol{z}f\left(\boldsymbol{x}\right)^\top\right] = 0.$$

This leads to the conclusion that $\Sigma_{f(\boldsymbol{x}),\boldsymbol{z}|\boldsymbol{y}} = 0$.

Combining (a) and (b), we finish the proof. $\qquad\square$

A.2   Proof of Lemma A.2

Lemma A.2. *Under the assumptions of Theorem 1 and data distribution $\mathcal{P}_0$ with penalty $\lambda < \sigma^2$, if $\mathcal{L}(f)$ is minimized, then $(1 - \frac{\lambda}{\sigma^2})\mathcal{L}_2$ and $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$ are both minimized.*

Proof. Notice that there exists a ground truth f^* that satisfies both Criterion (C1) and Criterion (C2). By definition of Criterion (C2), the ground truth f^* should minimize $\mathcal{L}_2$. On the other hand, we have proved in Section A.1 that satisfying Criterion (C2) leads to minimizing $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$.

Notice that

$$\mathcal{L}(f) = \lambda\mathcal{L}_1 + \mathcal{L}_2 = \left(1 - \frac{\lambda}{\sigma^2}\right)\mathcal{L}_2 + \lambda\left(\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2\right),$$

with $\lambda > 0$. Therefore, the ground truth f^* minimizes $\left(1 - \frac{\lambda}{\sigma^2}\right)\mathcal{L}_2$ and $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$ at the same time. Thus, any function f minimizing the loss must minimize both $\left(1 - \frac{\lambda}{\sigma^2}\right)\mathcal{L}_2$ and $\mathcal{L}_1 + \frac{1}{\sigma^2}\mathcal{L}_2$, or it would be larger than $\mathcal{L}(f^*)$. $\qquad\square$

B Proof of Theorem 2

Theorem 2 (Model-Free Failure)**.** *Assume that there exists a ground truth $f^* \in \mathcal{F}$ satisfying both Criterion (C1) and (C2), where $\mathcal{F}$ is the candidate function class. And assume that there exist function $f_1 \in \mathcal{F}$ and $f_2 \in \mathcal{F}$ such that covariance $\mathrm{Cov}[f_1(\boldsymbol{x}), \boldsymbol{y}] \neq 0$ and $\mathrm{Cov}[f_2(\boldsymbol{x}), \boldsymbol{z}] = 0$. Assume that for any $f_2 \in \mathcal{F}$, the matrix $\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]$ is nonsingular. Let $\mathcal{A}'_{\mathcal{P}}$ be the set of processor that minimizes the training loss with $n_0 = o(d_f)$ downstream samples and infinite pretext samples, where d_f is the dimension of $f(\boldsymbol{x})$,*

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

And let $\mathcal{B}_{\mathcal{P}}, \mathbb{S}$ be defined as in Theorem 1. Then there exists a distribution $\mathcal{P}^0 \in \mathbb{S}$ (which means $\mathcal{A}_{\mathcal{P}^0} \subset \mathcal{B}_{\mathcal{P}^0}$), such that all the function that minimizes the loss $\mathcal{L}_{\infty, n_0}$ cannot meet Criterion (C1) and (C2) simultaneously, namely:

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

Proof. We consider the distribution $\mathcal{P}_0$ as in Section A, that is: Firstly, let $\mathbb{E}[\boldsymbol{x}] = \mathbb{E}[\boldsymbol{y}] = \mathbb{E}[\boldsymbol{z}] = 0$ and $\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top] = I$. Given a marginal distribution of $(\boldsymbol{x}, \boldsymbol{y})$, construct a linear relationship $\boldsymbol{y} = B\boldsymbol{z}$ where the singular values of matrix B be $\sigma_1 = \cdots = \sigma_{d_y} = \sigma$. And choose the penalty $\lambda < \sigma^2$.

Let us first consider the training process. We first claim that when $n_0 = o(d_f)$, for any f, the first term of $\mathcal{L}_{\infty, n_0}$ can be trained to zero, i.e.,

$$\frac{1}{n_0}\left\|Y_{down1} - \widetilde{W}_2 f\left(X_{down1}\right)\right\|^2 = 0,$$

by taking

$$\widetilde{W}_2 = Y_{down1} f^\top\left(X_{down1}\right)\left[f\left(X_{down1}\right)f^\top\left(X_{down1}\right)\right]^{-1},$$

where we abuse $f\left(X_{down1}\right) \in \mathbb{R}^{d_f \times n_0}$.

Therefore, to minimize the loss $\mathcal{L}_{\infty, n'_2}$ only needs to minimize the second term of $\mathcal{L}_{\infty, n_0}$:

$$\hat{f} \in \arg\min_f\left[-\min_W \mathbb{E}_{\boldsymbol{x}, \boldsymbol{z}}\|\boldsymbol{z} - Wf\left(\boldsymbol{x}\right)\|^2\right].$$

Note that for any f,

$$\min_W \mathbb{E}_{\boldsymbol{x}, \boldsymbol{z}}\|\boldsymbol{z} - Wf\left(\boldsymbol{x}\right)\|^2 = \mathbb{E}\|\boldsymbol{z}\|^2 - \mathrm{tr}\left[\mathbb{E}\left[\boldsymbol{z}f^\top\left(\boldsymbol{x}\right)\right]\left(\mathbb{E}\left[f\left(\boldsymbol{x}\right)f^\top\left(\boldsymbol{x}\right)\right]\right)^{-1}\mathbb{E}\left[f\left(\boldsymbol{x}\right)\boldsymbol{z}^\top\right]\right] \leq \mathbb{E}\|\boldsymbol{z}\|^2.$$

When $f\left(\boldsymbol{x}\right)$ is uncorrelated to $\boldsymbol{z}$ (i.e., $f = f_2$), the equality holds, i.e.,

$$\min_W \mathbb{E}_{\boldsymbol{x}, \boldsymbol{z}}\|\boldsymbol{z} - Wf_2\left(\boldsymbol{x}\right)\|^2 = \mathbb{E}\|\boldsymbol{z}\|^2.$$

Therefore, any function $\hat{f}$ that minimizes the loss satisfies:

$$\min_W \mathbb{E}_{\boldsymbol{x}, \boldsymbol{z}}\left\|\boldsymbol{z} - W\hat{f}\left(\boldsymbol{x}\right)\right\|^2 = \mathbb{E}\|\boldsymbol{z}\|^2.$$

We next prove that any $\hat{f}$ that minimizes training loss $\mathcal{L}_{\infty,n_2'}$ could not contain any information of $\boldsymbol{y}$. Under the condition that $\mathbb{E}\left[\hat{f}(\boldsymbol{x})\hat{f}^\top(\boldsymbol{x})\right]$ is invertible, we derive that

$$
\begin{aligned}
\min_W \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \left\|\boldsymbol{y} - W\hat{f}(\boldsymbol{x})\right\|^2 &= \mathbb{E}\|\boldsymbol{y}\|^2 - \mathrm{tr}\left[\mathbb{E}\left[\boldsymbol{y}\hat{f}^\top(\boldsymbol{x})\right]\left(\mathbb{E}\left[\hat{f}(\boldsymbol{x})\hat{f}^\top(\boldsymbol{x})\right]\right)^{-1}\mathbb{E}\left[\hat{f}(\boldsymbol{x})\boldsymbol{y}^\top\right]\right] \\
&= \mathbb{E}\|\boldsymbol{y}\|^2 - \mathrm{tr}\left[B\,\mathbb{E}\left[\boldsymbol{z}\hat{f}^\top(\boldsymbol{x})\right]\left(\mathbb{E}\left[\hat{f}(\boldsymbol{x})\hat{f}^\top(\boldsymbol{x})\right]\right)^{-1}\mathbb{E}\left[\hat{f}(\boldsymbol{x})\boldsymbol{z}^\top\right]B^\top\right] \\
&= \mathbb{E}\|\boldsymbol{y}\|^2 - \mathrm{tr}\left[B^\top B\,\mathbb{E}\left[\boldsymbol{z}\hat{f}^\top(\boldsymbol{x})\right]\left(\mathbb{E}\left[\hat{f}(\boldsymbol{x})\hat{f}^\top(\boldsymbol{x})\right]\right)^{-1}\mathbb{E}\left[\hat{f}(\boldsymbol{x})\boldsymbol{z}^\top\right]\right] \\
&= \mathbb{E}\|\boldsymbol{y}\|^2 - \mathrm{tr}\left[B^\top B\right]\left(\mathbb{E}\|\boldsymbol{z}\|^2 - \min_W \mathbb{E}\left\|\boldsymbol{z} - W\hat{f}(\boldsymbol{x})\right\|^2\right) \\
&= \mathbb{E}\|\boldsymbol{y}\|^2.
\end{aligned}
$$

However, since there exists f_1 such that $\mathrm{Cov}(f_1(\boldsymbol{x}),\boldsymbol{y}) \neq 0$, leading to

$$
\min_W \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \left\|\boldsymbol{y} - W\hat{f}(\boldsymbol{x})\right\|^2 < \min_W \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \left\|\boldsymbol{y} - Wf_1(\boldsymbol{x})\right\|^2
$$

Therefore, $\hat{f}$ violates Criterion (C2). The proof is done. □

C   Proof of Theorem 3

Theorem 3 (Lower Bound for General Loss)**.** *Assume that there exists a function f such that $f(\boldsymbol{x}) \perp \boldsymbol{z}$ holds, and $f(\boldsymbol{x}) \perp \boldsymbol{z}$ holds if and only if $\mathbb{E}\,g_1(\rho_1(\boldsymbol{z}, W^*_{\boldsymbol{z},f(\boldsymbol{x})}f(\boldsymbol{x})))$ reaches the maximum. For the function g_2 and g_1, assume that $g_2 g_1^{-1}$ is convex and non-decreasing. Besides, we assume that there exists a linear transformation $V \in \mathbb{R}^{d_z \times d_z}$ such that:*

$$
\begin{aligned}
\mathbb{E}[\rho_1(\bar{\boldsymbol{y}}, V\boldsymbol{z})] < &\frac{1}{M}g_2 g_1^{-1} \max_f \mathbb{E}\left[g_1\rho_1\left(W^*_{V\boldsymbol{z},f(\boldsymbol{x})}f(\boldsymbol{x}), V\boldsymbol{z}\right)\right] \\
&- \frac{1}{M}\min_f \mathbb{E}\left[g_2\rho_2\left(W^*_{\boldsymbol{y},f(\boldsymbol{x})}f(\boldsymbol{x}), \boldsymbol{y}\right)\right],
\end{aligned}
$$

where $\bar{\boldsymbol{y}} = (\boldsymbol{y}, \vec{0}_{d_z - d_y})$ denotes the augmented vector of $\boldsymbol{y}$ with zero padding, and M denotes the upper bound of the derivative of g_2, namely, $M \geq \mathrm{d}g_2(x)/\mathrm{d}x, \forall x \in [0, \sup_{\boldsymbol{x},\boldsymbol{y}} \rho_2(W^*_{\boldsymbol{y},\hat{f}(\boldsymbol{x})}\hat{f}(\boldsymbol{x}), \boldsymbol{y}) + \sup_{\boldsymbol{y},\boldsymbol{z}} \rho_1(\bar{\boldsymbol{y}}, V\boldsymbol{z})]$.

Let $\mathcal{A}^g_{\mathcal{P}}$ denote the set of processor that minimize the general training loss with $n_0 = o(d_f)$ downstream samples and infinite pretext samples, where d_f is the dimension of $f(\boldsymbol{x})$,

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

Let $\mathcal{B}_{\mathcal{P}}$ be defined as in Theorem 1. Then there exists a distribution $\mathcal{P}^0$ such that all the function that minimizes the loss $\overline{\mathcal{L}}_{\infty,n_0}(f;\mathcal{P})$ cannot meet Criterion (C1) and (C2) simultaneously, namely:

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

Proof. Denote the trained predictor as $\hat{f}$. Notice that with $n = o(d_f)$, $\boldsymbol{y}_i = \widehat{W}_2 f(\boldsymbol{x}_i)$ by setting

$$
\widehat{W}_2 = (Y_{down1})^\top \left[f(X_{down1})f(X_{down1})^\top\right]^{-1} f(X_{down1}),
$$

where we abuse the notation $f(X) \in \mathbb{R}^{n \times d_f}$. This leads to

$$
\min_W g_2\rho_2\left(Y_{down1}, W\hat{f}(X_{down1})\right) = 0,
$$

on the training set.

Therefore, $\hat{f}$ is trained to minimize the loss

$$\mathcal{L}_1 = -\min_W \mathbb{E}\, g_1 \rho_1 \left(\boldsymbol{z}, W f\left(\boldsymbol{x}\right)\right),$$

leading to the independence between $f(\boldsymbol{x})$ and $\boldsymbol{z}$.

By the definition of ρ_2 and ρ_1, we have for any $V > 0$,

$$\min_{W_1} \mathbb{E}\, g_2 \rho_2 \left(\boldsymbol{y}, W_1 f\left(\boldsymbol{x}\right)\right) \triangleq \mathbb{E}\, g_2 \rho_2 \left(\boldsymbol{y}, \widehat{W}_1 \hat{f}\left(\boldsymbol{x}\right)\right) = \mathbb{E}\, g_2 \rho_1 \left(\bar{\boldsymbol{y}}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right), \tag{6}$$

where $\bar{\boldsymbol{y}}, \widehat{\overline{W}}_1$ are the augmented version filled with zero. By the upper bound of $\frac{\mathrm{d}g_2(x)}{\mathrm{d}x}$, we have

$$
\begin{aligned}
&g_2 \rho_1 \left(\bar{\boldsymbol{y}}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right) - g_2 \rho_1 \left(V \boldsymbol{z}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right) \\
=& \frac{\mathrm{d}g_2\left(x\right)}{\mathrm{d}x}\Big|_{x=\xi} \left[\rho_1 \left(\bar{\boldsymbol{y}}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right) - \rho_1 \left(V \boldsymbol{z}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right)\right] \\
\geq& \frac{\mathrm{d}g_2\left(x\right)}{\mathrm{d}x}\Big|_{x=\xi} \left[-\rho_1 \left(\bar{\boldsymbol{y}}, V \boldsymbol{z}\right)\right] \\
\geq& -M \rho_1 \left(\bar{\boldsymbol{y}}, V \boldsymbol{z}\right).
\end{aligned}
\tag{7}
$$

where the first equation is due to mean value theorem of integrals with point ξ. The second equality is due to $\frac{\mathrm{d}g_2(x)}{\mathrm{d}x} \geq 0$ and triangle inequality. And the third inequality is due to the condition $\frac{\mathrm{d}g_2(x)}{\mathrm{d}x}\big|_{x=\xi} \leq M$, and $\xi \leq \max\{\rho_1 \left(\bar{\boldsymbol{y}}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right), \rho_1 \left(V \boldsymbol{z}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right)\} \leq Q_1 + Q_2$.

Besides, notice that

$$
\begin{aligned}
&\mathbb{E}\, g_2 \rho_1 \left(V \boldsymbol{z}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right) \\
=& \mathbb{E}\, g_2 g_1^{-1} g_1 \left(\rho_1 \left(V \boldsymbol{z}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right)\right) \\
\stackrel{(i)}{\geq}& g_2 g_1^{-1} \mathbb{E}\, g_1 \left(\rho_1 \left(V \boldsymbol{z}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right)\right) \\
\stackrel{(ii)}{\geq}& g_2 g_1^{-1} \min_W \mathbb{E}\, g_1 \left(\rho_1 \left(V \boldsymbol{z}, W \hat{f}\left(\boldsymbol{x}\right)\right)\right) \\
=& g_2 g_1^{-1} \max_f \min_W \mathbb{E}\, g_1 \left(\rho_1 \left(V \boldsymbol{z}, W f\left(\boldsymbol{x}\right)\right)\right),
\end{aligned}
\tag{8}
$$

where inequality (i) follows Jensen's inequality with condition $g_2 g_1^{-1}$ is convex, and inequality (ii) follows that $g_2 g_1^{-1}$ is increasing. The final equation is due to the independence between $f(\boldsymbol{x})$ and $\boldsymbol{z}$.

Combining Equation (6), Equation (7), and Equation (8), leads to

$$
\begin{aligned}
&\min_{W_1} \mathbb{E}\, g_2 \rho_2 \left(\boldsymbol{y}, W_1 f\left(\boldsymbol{x}\right)\right) \\
=& \mathbb{E}\, g_2 \rho_1 \left(\bar{\boldsymbol{y}}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right) \\
\geq& \mathbb{E}\, g_2 \left(\rho_1 \left(V \boldsymbol{z}, \widehat{\overline{W}}_1 \hat{f}\left(\boldsymbol{x}\right)\right)\right) - M \mathbb{E}\, \rho_1 \left(\bar{\boldsymbol{y}}, V \boldsymbol{z}\right) \\
\geq& g_2 g_1^{-1} \max_f \min_W \mathbb{E}\, g_1 \left(\rho_1 \left(V \boldsymbol{z}, W \hat{f}\left(\boldsymbol{x}\right)\right)\right) - M \mathbb{E}\, \rho_1 \left(\bar{\boldsymbol{y}}, V \boldsymbol{z}\right) \\
>& \min_f \min_W \mathbb{E}\, g_2 \rho_2 \left(\boldsymbol{y}, W_1 f\left(\boldsymbol{x}\right)\right).
\end{aligned}
$$

This contradicts with Criterion (C2). The proof is done. $\square$

D   Proof of Theorem 4

Theorem 4 (Model-Dependent Lower Bound). *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 $f_0 \in \mathcal{F}$ such that $\mathrm{Cov}[f_0(\boldsymbol{x}), \boldsymbol{y}] \neq 0$ and a function $f_2 \in \mathcal{F}_2$ such*

that $f_2(\boldsymbol{x}) \perp \boldsymbol{z}$. *Assume matrix* $\mathbb{E}[f(\boldsymbol{x})f^\top(\boldsymbol{x})]$ *is nonsingular for all* $f \in \mathcal{F}$. *Let* $\mathcal{A}'_{\mathcal{P}}$ *be the set of processor that minimize the loss with* $n_0 = o(\mathcal{M}(\mathcal{F}_1, \mathcal{L}))$ *downstream samples and infinite pretext samples, where* d_f *is the dimension of* f,

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

And let $\mathcal{B}_{\mathcal{P}}, \mathbb{S}$ *be defined as in Theorem* 1. *Then* there *exists a distribution* $\mathcal{P}^0 \in \mathbb{S}$ *(which means* $\mathcal{A}_{\mathcal{P}^0} \subset \mathcal{B}_{\mathcal{P}^0}$)*, such that all the function that minimizes the loss* $\mathcal{L}_{\infty, n_0}$ *cannot meet Criterion* (C1) *and* (C2) *simultaneously, namely:*

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

We consider the distribution $\mathcal{P}_0$ as in Section A, that is: Firstly, let $\mathbb{E}[\boldsymbol{x}] = \mathbb{E}[\boldsymbol{y}] = \mathbb{E}[\boldsymbol{z}] = 0$ and $\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^\top] = I$. Given a marginal distribution of $(\boldsymbol{x}, \boldsymbol{y})$, construct a linear relationship $\boldsymbol{y} = B\boldsymbol{z}$ where the singular values of matrix B be $\sigma_1 = \cdots = \sigma_{d_y} = \sigma$. And choose the penalty $\lambda < \sigma^2$.

Firstly, consider the training process. Under the assumptions on $\mathcal{F}_1 \times \mathcal{F}_2$, there exists a $f_q(f_2(\cdot))$ such that:

(a) on the one hand, $f_q(f_2(\cdot))$ minimize $\mathcal{L}_1$.

$$f_q(f_2(\cdot)) \in \arg\min_f \mathcal{L}_1(f).$$

Note that $f_q f_2(\boldsymbol{x}) \perp \boldsymbol{z}$ follows $f_2(\boldsymbol{x}) \perp \boldsymbol{z}$. As a result, due to the assumption $\mathbb{E}\,\boldsymbol{z} = 0$, we have

$$\arg\min_W \|\boldsymbol{z} - W f_q f_2(\boldsymbol{x})\| = 0, \ \min_W \|\boldsymbol{z} - W f_q f_2(\boldsymbol{x})\|^2 = \|\boldsymbol{z}\|^2.$$

And further notice that for any predictor f, we have

$$\min_W \|\boldsymbol{z} - W f(\boldsymbol{x})\|^2 \leq \|\boldsymbol{z} - 0 f(\boldsymbol{x})\|^2 = \|\boldsymbol{z}\|^2,$$

therefore, $f_q(f_2(\cdot))$ minimize $\mathcal{L}_1$.

(b) on the other hand, $f_q(f_2(\cdot))$ makes $\mathcal{L}_2$ equal to zero on the training set. Due to the definition of model capacity, when $n < o(\mathcal{M}(\mathcal{F}_1, \mathcal{L}))$, by fixing f_2, there always exists f_q such that it can fit any n samples:

$$\min_W \|Y_{down1} - W f_q(f_2(X_{down1}))\|^2 = 0.$$

As a result, $f_q(f_2(\cdot))$ can minimize the training loss since it minimize both $\mathcal{L}_1$ and $\mathcal{L}_2$. Therefore, any predictor $\hat{f}(\cdot)$ that minimize the training loss must minimize $\mathcal{L}_1$, leading to

$$\min_W \mathbb{E}\|\boldsymbol{z} - W \hat{f}(\boldsymbol{x})\|^2 = \mathbb{E}\|\boldsymbol{z}\|^2.$$

Next we consider the predictor $\hat{f}(\cdot)$.

$$
\begin{aligned}
&\min_W \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \left\| \boldsymbol{y} - W \hat{f}(\boldsymbol{x}) \right\|^2 \\
&= \mathbb{E}(\|\boldsymbol{y}\|^2) - \operatorname{tr}\left[\mathbb{E}\left(\boldsymbol{y}\hat{f}^\top(\boldsymbol{x})\right) \left(\mathbb{E}\left[\hat{f}(\boldsymbol{x})\hat{f}^\top(\boldsymbol{x})\right]\right)^{-1} \mathbb{E}\left(\hat{f}(\boldsymbol{x})\boldsymbol{y}^\top\right) \right] \\
&= \mathbb{E}(\|\boldsymbol{y}\|^2) - \operatorname{tr}\left[B\,\mathbb{E}\left(\boldsymbol{z}\hat{f}^\top(\boldsymbol{x})\right) \left(\mathbb{E}\left[\hat{f}(\boldsymbol{x})\hat{f}^\top(\boldsymbol{x})\right]\right)^{-1} \mathbb{E}\left(\hat{f}(\boldsymbol{x})\boldsymbol{z}^\top\right) B^\top \right] \\
&= \mathbb{E}(\|\boldsymbol{y}\|^2) - \operatorname{tr}\left[B^\top B\,\mathbb{E}\left(\boldsymbol{z}\hat{f}^\top(\boldsymbol{x})\right) \left(\mathbb{E}\left[\hat{f}(\boldsymbol{x})\hat{f}^\top(\boldsymbol{x})\right]\right)^{-1} \mathbb{E}\left(\hat{f}(\boldsymbol{x})\boldsymbol{z}^\top\right) \right] \\
&= \mathbb{E}(\|\boldsymbol{y}\|^2) - \operatorname{tr}\left[B^\top B \left(\min_W \|\boldsymbol{z} - W\hat{f}(\boldsymbol{x})\|^2 - \|\boldsymbol{z}\|^2\right) \right] \\
&= \mathbb{E}(\|\boldsymbol{y}\|^2).
\end{aligned}
$$
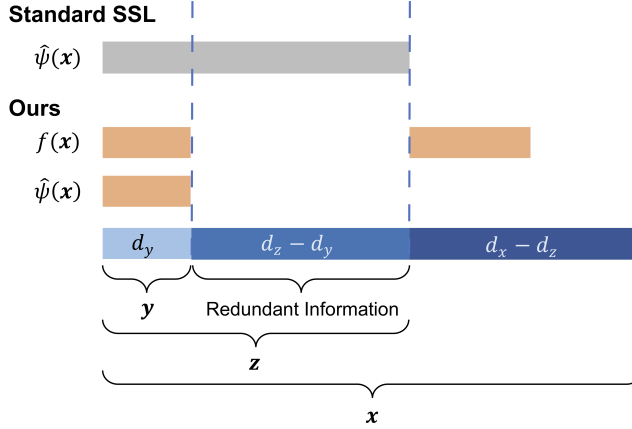
Figure 2: A simple case for understanding redundant information. Here $\boldsymbol{y}$ is the prefix of $\boldsymbol{z}$, and $\boldsymbol{z}$ is the prefix of $\boldsymbol{x}$. For the standard SSL, the learned representation $\widehat{\psi}(\boldsymbol{x})$ consists of not only the information of $\boldsymbol{y}$ but also the redundant information in $\boldsymbol{z}$. For the processor-based SSL, the learned coarse representation $f(\boldsymbol{x})$ contains the information of $\boldsymbol{y}$ and some information of $\boldsymbol{x}$, and does not include any redundant information in $\boldsymbol{z}$. Then the final learned representation $\widehat{\psi}(\boldsymbol{x})$ will only contain the information of $\boldsymbol{y}$, hopefully.

Notice that $\min_W \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \| \boldsymbol{y} - W f_1(\boldsymbol{x}) \|^2 < \mathbb{E} (\| \boldsymbol{y} \|^2)$ since $\mathrm{Cov}[f_1(\boldsymbol{x}), \boldsymbol{y}] \neq 0$. Therefore,

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \left\| \boldsymbol{y} - W^*_{\boldsymbol{y},\hat{f}(\boldsymbol{x})} \hat{f}(\boldsymbol{x}) \right\|^2 > \min_f \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}} \left\| \boldsymbol{y} - W^*_{\boldsymbol{y},f(\boldsymbol{x})} f(\boldsymbol{x}) \right\|^2 ,$$

which contradicts to the Criterion (C2).

E   Illustration of redundant information

In the main text, we state that the processor-based SSL aims at eliminating redundant information. For better understanding, let us consider a simple case. Suppose $\boldsymbol{x}$ is a d_x-dimensional random vector. Let $\boldsymbol{z}$ and $\boldsymbol{y}$ consist of the first d_z and d_y elements of $\boldsymbol{x}$, respectively, i.e., $\boldsymbol{z} = \boldsymbol{x}[0 : d_z - 1]$ and $\boldsymbol{y} = \boldsymbol{x}[0 : d_y - 1]$. When we execute the standard self-supervised learning (with linear representation $\hat{\psi}$) using infinite pretext samples, the learned representation has rank d_z which consists of d_y-dimensional useful information and $(d_z - d_y)$-dimensional redundancy information that we cannot eliminate (See Figure 2). Therefore, we need $\tilde{\mathcal{O}}(d_z)$ samples to train the liner layer $\widehat{W}$ in the downstream tasks. In comparison, the processor approach requires a function f that eliminates the redundancy information at the beginning, i.e., $f(\boldsymbol{x}) = \boldsymbol{x}[0 : d_y - 1, d_z : d_x]$. Hopefully, $f(\boldsymbol{x})$ will consist of all the useful information and some information of $\boldsymbol{x}[d_z : d_x - 1]$, without any information of $\boldsymbol{x}[d_y : d_z - 1]$. And then we apply pretext task on $(f(\boldsymbol{x}), \boldsymbol{z})$, which further removes the information of $\boldsymbol{x}[d_z : d_x - 1]$. Thus, the rank of representation $\widehat{\psi}(\boldsymbol{x})$ could be reduced to d_y. In this way, we will only need $\tilde{\mathcal{O}}(d_y)$ samples for the downstream task.

F   Experimental details

F.1   Experimental details for synthetic data

Datasets and environment. The dataset is simulated as follows. Firstly, we randomly generate n_1 pretext samples $\{(x^i_{pre}, z^i_{pre})\}_{i \in [n_1]}$, where $x^i_{pre} \in \mathbb{R}^{d_x}$ is generated from Gaussian distribution $\mathcal{N}(0, I_{d_x})$, and $z^i_{pre} \in \mathbb{R}^{d_z}$ is a perturbation of the first d_z elements of x^i_{pre}, namely, $z^i_{pre} = x^i_{pre}[0 : d_z] + 0.01 \cdot \varepsilon^i_1$, where $\varepsilon^i_1 \sim \mathcal{N}(0, I_{d_z})$. For the ease of notations, we present the data using sample form instead of the matrix form (X_{pre}, Y_{pre}).

Similarly, we generate n_2 downstream samples $\{(x^i_{down}, y^i_{down})\}_{i \in [n_2]}$, where $x^i_{down} \in \mathbb{R}^{d_x}$ is generated from Gaussian distribution $\mathcal{N}(0, I_{d_x})$ and $y^i_{down} \in \mathbb{R}^{d_y}$ is a perturbation of the first d_y elements of x^i_{down}, i.e., $y^i_{down} = x^i_{down}[0 : d_y] + 0.01 \cdot \varepsilon^i_2$, where $\varepsilon^i_2 \sim \mathcal{N}(0, I_{d_y})$. For the processor training procedure, we use n_0 of

downstream data as $\{(x_{down1}^i, y_{down1}^i)\}_{i \in [n_0]}$, and the rest $n_2 - n_0$ samples are used in downstream tasks (denoted by $\{(x_{down2}^i, y_{down2}^i)\}_{i \in [n_2 - n_0]}$). We require $n_0 = \alpha n_2$, and setting $\alpha = 0.5$ without loss of generality. We generate n_t test samples under the same procedure.

In each run of our proposed algorithm, we first use n_0 samples $\{(x_{down1}^i, y_{down1}^i)\}_{i \in [n_0]}$ and n_1 pretext samples to train the processor. Then we apply the processor and the n_1 pretext samples $n_2 - n_0$ downstream samples to finish the training. For a fair comparison, we use n_1 pretext samples to train the feature during the SSL procedure and then use n_2 downstream samples to learn the linear layer. We run each experiment 5 times repeatedly and calculate its mean and standard deviation. We plot their 95% confidence bands in all the figures.

The rationality of the loss $\mathcal{L}(f)$. Set $d_x = 100$, $d_z = 80$, $d_y = 5$, $d_f = 5$, $n_1 = 20000$ and $\lambda = 0.1$. We plot the MSE as n_0 varies from 15 to 70 in Figure 1(a). With large n_0, the downstream tasks benefit from small MSE, showing that the algorithm indeed finds the proper coarse representations. Theorem 1 demonstrates this phenomenon, which claims that with sufficient downstream task samples, the proposed algorithm finds the proper coarse representations.

We further remark that in the case $n_0 = 40$, the SSL suffers from a large MSE. It is related to the "double decent" phenomenon in the downstream tasks, and MSE reaches the maximum when $n_0 = d_z$. We finally remark that when $n_0 \in [20, 40]$, the proposed algorithm outperforms SSL in terms of MSE.

Large coefficient λ harms the performance. Set $d_x = 100$, $d_z = 80$, $d_y = 5$, $d_f = 5$, $n_1 = 20000$ and $n_2 = n_0 = 15$. We test the case λ varies from 0.1 to 1.5, as plotted in Figure 1(b). Figure 1(b) illustrates that when λ is large, the algorithm suffers from unsatisfying performances. Theorem 1 demonstrates the phenomenon. Intuitively, that is because z contains the information of y. With a large coefficient λ, the trained classifier tends to eliminate the information of y, leading to a lousy model performance (MSE).

Large dimension d_f harms the performance. Set $d_x = 100$, $d_z = 80$, $d_y = 5$, $n_1 = 20000$, $n_2 = n_0 = 15$, and $\lambda = 0.1$. We test the case d_f varies from 1 to 12, and plot them in Figure 1(c). Firstly, notice that MSE reaches the minimum when $d_f = d_y$, that is because we force the coarse representation to have no redundancy information by limiting its dimension. Secondly, notice that the figure shows a Basin-type phenomenon and reaches the minimum when $d_f = d_y$. On the one hand, when $d_f < d_y$, the coarse representation is forced to lose the information of y, leading to the underfitting phenomenon. On the other hand, when $d_f > d_y$, the performance of coarse representation is limited by the downstream sample numbers, as illustrated in Theorem 2 and Theorem 4.

F.2 Experimental details for real-world data

All experiments can be handled with an RTX 2080Ti graphic card. To simplify the comparison, we require that the processor-training method uses the same downstream samples in downstream tasks as standard SSL. Therefore, the processor training process is extra compared to standard SSL. We emphasize that the processor training method in extra-sample settings performs better than the settings in the main text (which split some downstream samples to train the processor), and the negative results in such extra-sample settings are stronger.

Data split strategy. We first classify the samples according to their categories and sort them according to their index in the CIFAR-10 dataset. And then pick the correct number of samples in each class one by one without overlapping. For each subset, we assure that there is the same number of samples for each class.

Model training. For standard SSL training, we directly follow all the hyperparameters and training strategies from Gidaris et al. (2018b) except for the training set size.

For processor-based learning, we also copy the hyperparameters from the standard setting, which is necessary. There are two additional SGD optimizers for training $W_{y,f(x)}^*$ and $W_{z,f(x)}^*$, which are two linear layers with the same input size and different output size in our implementation. We set the learning rate as 0.1 and train 20 times before calculating the loss function used to update f. When optimizing Equation (1), λ is set to 0.1 to balance the two loss terms in Equation (1).

For the half NIN model, we half the channel number in each block. Consequently, the size of all linear layers should also be changed.