Predicting What You Already Know Helps: Provable Self-Supervised Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Self-supervised representation learning solves auxiliary prediction tasks (known as pretext tasks) without requiring labeled data to learn useful semantic representations. These pretext tasks are created solely using the input features, such as predicting a missing image patch, recovering the color channels of an image from context, or predicting missing words in text; yet predicting this *known* information helps in learning representations effective for downstream prediction tasks.
We posit a mechanism exploiting the statistical connections between certain *reconstruction-based* pretext tasks that guarantee to learn a good representation. Formally, we quantify how the approximate independence between the components of the pretext task (conditional on the label and latent variables) allows us to learn representations that can solve the downstream task by just training a linear layer on top of the learned representation. We prove the linear layer yields small approximation error even for complex ground truth function class and will drastically reduce labeled sample complexity.

1 Introduction

Self-supervised learning revitalizes machine learning models in computer vision, NLP, and control problems (see reference therein [32, 34, 12, 58, 31]). Training a model with auxiliary tasks based only on input features reduces the extensive costs of data collection and semantic annotations for downstream tasks. It is also known to improve the adversarial robustness of models [26, 8, 9]. Self-supervised learning creates pseudo labels solely based on input features, and solves auxiliary prediction tasks (or pretext tasks) in a supervised manner. However, the underlying principles of self-supervised learning are mysterious since it is a-priori unclear why predicting what we already know should help. We thus raise the following question:

What conceptual connection between pretext and downstream tasks ensures good representations? What is a good way to quantify this?

As a thought experiment, consider a simple downstream task of classifying desert, forest, and sea images. A meaningful pretext task is to predict the background color of images (known as image colorization [61]). Denote X_1, X_2, Y to be the input image, color channel, and the downstream label respectively. Given knowledge of the label Y, one can possibly predict the background X_2 without knowing much about X_1. In other words, X_2 is approximately independent of X_1 conditional on the label Y. Consider another task of inpainting [43] the front of a building (X_2) from the rest (X_1). While knowing the label "building" (Y) is not sufficient for successful inpainting, adding additional latent variables Z such as architectural style, location, window positions, etc. will ensure that variation in X_2 given Y, Z is small. We can mathematically interpret this as X_1 being approximate conditionally independent of X_2 given Y, Z.

The main insight that we exploit in this work is that with approximate conditional independence (as in the above examples), a method that predicts X_2 from X_1 will inadvertently implicitly encode and

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

learn to predict Y (and Z) from X_1 as an intermediate step, and then predict X_2 from Y[1]. Building upon this insight, we make the following contributions.

Contributions. The goal of this paper, as in statistical learning theory, is to investigate the *statistical connections* between the random variables of input features (in this paper (X_1, X_2)) and downstream labels Y, and show how specific connections can guarantee a successful learning procedure. For self-supervised learning (SSL), success is measured using the following 2 notions, 1) expressivity, i.e. does the learned representation from SSL have the ability to express the ground truth prediction function for labels Y, and 2) sample complexity, i.e. can it do so with way fewer labeled samples than what would be required without SSL.

In this work, we show such guarantees for a class of *reconstruction-based* SSL methods under a statistical assumption of *approximate conditional independence (ACI)*. In particular we show that under such an assumption, the learned representation from SSL will end up having the following properties, 1) it can express the ground truth label as a *linear function*, thus guaranteeing expressivity, and 2) will also end up being low-rank (or low-dimensional), thus guaranteeing smaller labeled sample complexity. Note that such an expressive and sample efficient (summarized as *good*) representation is often not a-priori available. For instance, the original input features themselves may not be able to express the ground truth function linearly, while kernel methods with a fixed kernel, while expressive, may not be sample efficient for many problems of interest. The strategy in modern machine learning is to find such a *good* representation as the output of a complicated neural network. The benefit of SSL, as we formally show here, is that the complicated but *good* representation function can be learned using just *unlabeled data*, so that labeled data is just needed to learn a linear function.

The *reconstruction-based* SSL method (differentiated from other SSL methods in Section 1.1) we consider is strongly motivated by empirical works [61, 43, 12, 22], but is a simplification that captures the essence of the problem and is amenable to a precise theoretical analysis. We consider a two-staged pipeline, where we first learn a representation function ψ (e.g. output of a neural network) from input X_1 and pretext target X_2 using *unlabeled data* by minimizing $\mathbb{E}_{(X_1,X_2)}[\|X_2 - \psi(X_1)\|^2]$. In the second stage of downstream task, we learn a *linear layer* on top of representation ψ using labeled samples (X_1, Y), thus restricting to learning from a significantly smaller hypothesis class of $\mathcal{H}_\psi = \{f : X_1 \to Y | f \text{ is linear in } \psi\}$. The key non-trivial question of expressivity is now whether the ground truth predictor $f^* \equiv \mathbb{E}[Y|X_1]$ can be approximated well by this class $\mathcal{H}_\psi$, and the question of sample complexity reduces to the understanding the sample complexity of learning $\mathcal{H}_\psi$. Under appropriate statistical connections[2] between input data X_1, X_2 and target Y, we prove both the desired properties, expressivity and low sample complexity, for the aforementioned SSL method.

Our statistical assumption based on approximate conditional independence (ACI) helps us demonstrate how solving pretext tasks created from *known information* can learn useful representations. Specifically, we show that once the complicated representation function ψ is learned using an abundance of *unlabeled data* in the SSL stage, not only is ψ expressive enough, but it will also require only $\tilde{\mathcal{O}}(k)$ *labeled samples* to solve a k-way supervised learning task under exact conditional independence (CI). In contrast, solving the downstream task without any pretraining will require a lot of labeled data to learn the representation function from scratch. Since the strong exact conditional independence assumption will likely not be satisfied in practice, our main contribution is to derive similar risk bounds when only approximate CI (ACI) is satisfied. We quantify the notion of ACI using the norm of a certain partial covariance matrix (Definition 4.1) and our risk bound scales linearly with it. We verify this and other aspects of our main Theorem 4.2 using simulations and also find that pretext task helps when CI is approximately enforced in text domain. We further demonstrate on a real-world image dataset that a pretext task-based linear model performs at least as well as many baselines.

1.1 Related work

Self-supervised learning (SSL) methods in practice: There has been a flurry of self-supervised methods lately. One class of methods reconstruct images from corrupted or incomplete versions of it, like denoising auto-encoders [56], image inpainting [43], and split-brain autoencoder [62]. Pretext

[1]This is formally demonstrated in the proof sketch of Lemma 3.1.

[2]We note that since the representation ψ is the result of the SSL method and not something we have access to a-priory, we cannot make any direct assumptions on it.

tasks are also created using visual common sense, including predicting rotation angle [19], relative patch position [13], recovering color channels [61], solving jigsaw puzzle games [41], and discriminating images created from distortion [14]. We refer to the above procedures as reconstruction-based SSL. Another popular paradigm is contrastive learning [10, 11]. The idea is to learn representations that bring similar data points closer while pushing randomly selected points further away [58, 35, 3] or to maximize a contrastive-based mutual information lower bound between different views [27, 42, 49]. A popular approach for text domain is based on language modeling where models like BERT and GPT create auxiliary tasks for next word predictions [12, 44]. The natural ordering or topology of data is also exploited in video-based [59, 39, 16], graph-based [60, 29] or map-based [63] SSL. For instance, the pretext task is to determine the correct temporal order for video frames as in [39].

Theory for SSL: While we theoretically study reconstruction-based SSL, prior work has different flavors of theoretical results for different kinds of SSL methods. Most relevant are the guarantees for representation learning using SSL methods on downstream tasks that just learn a linear classifier on top of the learned representations. [3] shows guarantees for representations from a contrastive learning objective: $L_1^{cont}(\psi) = \mathbb{E}_{(X_1,X_2),X_2'}[\log(1 + e^{-\psi(X_1)^\top \psi(X_2) + \psi(X_1)^\top \psi(X_2')})]$. Under a class conditional independence assumption, i.e. $X_1 \perp X_2 \mid Y$, they show that representation ψ that does well on contrastive objective, i.e. $L_1^{cont}(\psi) \leq \epsilon$, will have $\mathcal{O}(\epsilon)$ linear classification loss on the average binary task involving pairs of classes (y_1, y_2). However, their analysis cannot handle the general case of approximate conditional independence. Recently, Tosh *et al.* [51] show that contrastive learning representations can *linearly* recover continuous functions of the underlying topic posterior under a topic modeling assumption for text. While their assumption bears similarity to ours, the assumption of independent sampling of words is strong and does not generalizable to other domains like images. Most relevant is a concurrent work [52] that shows guarantees for a contrastive learning objective that looks like $L_2^{cont}(\psi, \eta) = \mathbb{E}_{(X_1,X_2),X_2'}\left[\log(1 + e^{-\psi(X_1)^\top \eta(X_2)}) + \log(1 + e^{\psi(X_1)^\top \eta(X_2')})\right]$, with a multi-view redundancy assumptions that is very similar to our ACI assumption. We take a closer look at their assumption in Section E.2. All the above objectives are different from the simple reconstruction-based objective we consider: $L(\psi) = \mathbb{E}_{(X_1,X_2)}\left[\|X_2 - \psi(X_1)\|^2\right]$. Saunshi *et al.* [46] show guarantees for representations learned using language modeling on sentence classification tasks. Some more recent work [53, 40, 50, 57] provide theoretical understanding on SSL respectively based on causality, mutual information, gradient-descent dynamics, and alignment/uniformity of representations, without explicit risk bounds for downstream tasks. There is a mutual information maximization view of contrastive learning, but [54] points out issues with it. Previous attempts to explain negative sampling [38] based methods use the theory of noise contrastive estimation [24, 36] to show asymptotic guarantees, without explicit connections to downstream tasks. CI is also used in sufficient dimension reduction [18, 17], while CI and redundancy assumptions on multiple views [33, 2] are used to analyze a canonical-correlation based dimension reduction algorithm and also for self-supervised learning algorithms like co-training [7]. Finally, [1, 55] provide a theoretical analysis for denoising auto-encoder.

1.2 Overview of results:

Section 2 introduces notation, setup, and the self-supervised learning procedure considered in this work. In Section 3, we analyze downstream sample complexity under exact CI and unlimited labeled data to highlight the key ideas. Section 4 presents our main result with relaxed conditions: under ACI with latent variables, and assuming finite samples in both pretext and downstream tasks, for various function classes, and both regression and classification tasks. Experiments verifying our theoretical findings are in Section 5. Proofs of most results are in the Appendix.

2 Preliminary

2.1 Notation

We use lower case symbols (x) to denote scalar quantities, bold lower case symbols ($\boldsymbol{x}$) for vector values, capital letters (X) for random variables, and capital and bold letters $\boldsymbol{X}$ for matrices. P_X denotes the probability law of random variable X, and the space of square-integrable functions with probability P is denoted by $L^2(P)$. We use standard $\mathcal{O}$ notation to hide universal factors and $\tilde{\mathcal{O}}$ to hide log factors. $\|\cdot\|$ stands for ℓ_2-norm for vectors or Frobenius norm for matrices.

Linear conditional expectation. $\mathbb{E}^L[Y|X]$ denotes the prediction of Y with linear regression:

$$\mathbb{E}^L[Y|X=\boldsymbol{x}] := \boldsymbol{W}^*\boldsymbol{x}+\boldsymbol{b}^*, \quad \text{where } \boldsymbol{W}^*, \boldsymbol{b}^* := \arg\min_{\boldsymbol{W},\boldsymbol{b}} \mathbb{E}[\|Y-\boldsymbol{W}X-\boldsymbol{b}\|^2].$$

In other words, $\mathbb{E}^L[Y|X]$ denotes the best linear predictor of Y given X. We also note that $\mathbb{E}[Y|X] \equiv \arg\min_f \mathbb{E}[\|Y-f(X)\|^2]$ is the best predictor of Y given X.

(Partial) covariance matrix. For random variables X, Y, we denote $\boldsymbol{\Sigma}_{XY}$ to be covariance matrix of X and Y. For simplicity in most cases, we assume $\mathbb{E}[X]=0$ and $\mathbb{E}[Y]=0$; thus we do not distinguish $\mathbb{E}[XY]$ and $\boldsymbol{\Sigma}_{XY}$. The partial covariance matrix between X and Y given Z is:

$$\boldsymbol{\Sigma}_{XY|Z} := \text{cov}\{X-\mathbb{E}^L[X|Z], Y-\mathbb{E}^L[Y|Z]\} \equiv \boldsymbol{\Sigma}_{XY}-\boldsymbol{\Sigma}_{XZ}\boldsymbol{\Sigma}_{ZZ}^{-1}\boldsymbol{\Sigma}_{ZY}, \tag{1}$$

which captures the correlation between X and Y setting aside the effect of Z.

Sub-gaussian random vectors. $X \in \mathbb{R}^d$ is ρ^2-sub-gaussian if for every fixed unit vector $\boldsymbol{v} \in \mathbb{R}^d$, the variable $\boldsymbol{v}^\top X$ is ρ^2-sub-gaussian, i.e., $\mathbb{E}[e^{s\cdot\boldsymbol{v}^\top(X-\mathbb{E}[X])}] \le e^{s^2\rho^2/2}$ ($\forall s \in \mathbb{R}$).

2.2 Setup and methodology

We denote by X_1 the input variable, X_2 the target random variable for the pretext task, and Y the label for the downstream task, with $X_1 \in \mathcal{X}_1 \subset \mathbb{R}^{d_1}, X_2 \in \mathcal{X}_2 \subset \mathbb{R}^{d_2}$ and $Y \in \mathcal{Y} \subset \mathbb{R}^k$. If $\mathcal{Y}$ is finite with $|\mathcal{Y}|=k$, we assume $\mathcal{Y} \subset \mathbb{R}^k$ is the one-hot encoding of the labels. $P_{X_1X_2Y}$ denotes the joint distribution over $\mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y}$. P_{X_1Y}, P_{X_1} denote the corresponding marginal distributions. Our proposed self-supervised learning aims to fulfill the following two steps:

Step 1 (pretext task): Learn a representation $\psi(\boldsymbol{x}_1)$ close to $\psi^* := \arg\min_{g\in\mathcal{H}} \mathbb{E}\|X_2-g(X_1)\|^2$, where $\mathcal{H}$ can vary for different settings that we will specify and discuss later.

Step 2 (downstream task): Perform linear regression on Y with $\psi(X_1)$, i.e. $f(\boldsymbol{x}_1) := (\boldsymbol{W}^*)^\top\psi(\boldsymbol{x}_1)$, where $\boldsymbol{W}^* \leftarrow \arg\min_{\boldsymbol{W}} \mathbb{E}_{X_1,Y}[\|Y-\boldsymbol{W}^\top\psi(X_1)\|^2]$. Namely we learn $f(\cdot)=\mathbb{E}^L[Y|\psi(\cdot)]$.

We study this simplified version in the main text, where in practice, the SSL procedure may utilize an encoder-decoder structure, while the downstream task uses both X_1 and X_2 to predict Y. We incorporate these extensions in Appendix C.3 and F.

With finite samples, performance of a learned representation ψ on the downstream task depends on the following quantities that capture expressivity and sample complexity respectively:

Approximation error indicates whether Y is *linearly separable* by the learned representation ψ, thus measuring expressivity. We measure this by comparing $\boldsymbol{W}\psi(X_1)$ to the optimal predictor $f^* := \mathbb{E}[Y|X_1=\boldsymbol{x}_1]$. Denote $e_{\text{apx}}(\psi) = \min_{\boldsymbol{W}} \mathbb{E}[\|f^*(X_1)-\boldsymbol{W}\psi(X_1)\|^2]$. This gives a measure of how well ψ can linearly predict Y when given infinite samples for the task.

Estimation error measure sample complexity of ψ on the downstream task and assume access to n_2 i.i.d. samples $(\boldsymbol{x}_1^{(1)}, \boldsymbol{y}^{(1)}), \cdots, (\boldsymbol{x}_1^{(n_2)}, \boldsymbol{y}^{(n_2)})$ drawn from P_{X_1Y}. We express the n_2 samples collectively as $\boldsymbol{X}_1^{\text{down}} \in \mathbb{R}^{n_2\times d_1}$, $\boldsymbol{Y} \in \mathbb{R}^{n_2\times k}$ and overload notation to say $\psi(\boldsymbol{X}_1^{\text{down}}) = \left[\psi(\boldsymbol{x}_1^{(1)})|\psi(\boldsymbol{x}_1^{(2)})\cdots|\psi(\boldsymbol{x}_1^{(n_2)})\right]^\top \in \mathbb{R}^{n_2\times d_2}$. We perform linear regression on the learned representation ψ and measure excess risk, that incorporates both approximation and estimation errors.

$$\hat{\boldsymbol{W}} \leftarrow \arg\min_{\boldsymbol{W}} \frac{1}{2n_2}\|\boldsymbol{Y}-\psi(\boldsymbol{X}_1)\boldsymbol{W}\|_F^2; \;\; \text{ER}_\psi(\hat{\boldsymbol{W}}) := \frac{1}{2}\mathbb{E}\|f^*(X_1)-\hat{\boldsymbol{W}}^\top\psi(X_1)\|_2^2.$$

3 Guaranteed recovery with conditional independence

In this section, we focus on the case where the input X_1 and pretext target X_2 are conditionally independent (CI) given the downstream label Y. While this is a strong assumption that is rarely satisfied in practice, it helps us understand the role of CI with clean results and builds up to our main results with ACI with latent variables in Section 4. As a warm-up, we show how CI helps when (X_1, X_2, Y) are jointly Gaussian to give us a flavor for the results to follow in Appendix B. We then analyze it for general random variables under two settings: (a) when the function class used for ψ is universal, (b) when ψ is restricted to be a linear function of given features. For now we assume

access to a large amount of unlabeled data so as to learn the optimal ψ^* perfectly and this will be relaxed later in Section 4. The general recipe for the results is as follows:

1. Find a closed-form expression for the optimal solution ψ^* for the pretext task.
2. Use conditional independence to show that optimal f^* is linear in ψ^*, i.e., $e_{\text{apx}}(\psi^*)$ is small.
3. Exploit the low rank structure of ψ^* to show small estimation error on downstream tasks.

Data assumption. Suppose $Y = f^*(X_1) + N$, where $f^* = \mathbb{E}[Y|X_1]$ and $\mathbb{E}[N] = 0$. We assume N is σ^2-subgaussian. For simplicity, we assume non-degeneracy: $\mathbf{\Sigma}_{X_iX_i}, \mathbf{\Sigma}_{YY}$ are full rank.

Assumption 3.1. *Let $X_1 \in \mathbb{R}^{d_1}, X_2 \in \mathbb{R}^{d_2}$ be random variables from some unknown distribution. Let label $Y \in \mathcal{Y}$ be a discrete random variable with $k = |\mathcal{Y}| < d_2$. We assume conditional independence: $X_1 \perp X_2 | Y$.*

Here Y can be interpreted as the multi-class labels where k is the number of classes. For regression problems, one can think about Y as the discretized values of continuous labels. We do not specify the dimension for Y since Y could be arbitrarily encoded but the results only depend on k and the variance of Y (conditional on the input X_1).

3.1 Universal function class.

Suppose we learn the optimal ψ^* among all measurable functions The optimal function ψ^* in this case is naturally given by conditional expectation: $\psi^*(\boldsymbol{x}_1) = \mathbb{E}[X_2|X_1 = \boldsymbol{x}_1]$. We show that CI implies that ψ^* is good for downstream tasks, which is not apriori clear.

Lemma 3.1 (Approximation error)**.** *If random variables X_1, X_2, Y satisfy Assumption 3.1, and $\boldsymbol{A} \in \mathbb{R}^{\mathcal{Y}\times d_2}$ with $\boldsymbol{A}_{y,:} := \mathbb{E}[X_2|Y = \boldsymbol{y}]$ has rank $k = |\mathcal{Y}|$. Then $f^* \equiv \boldsymbol{W}^*\psi^*$, i.e., $e_{apx}(\psi^*) = 0$.*

This tells us that although f^* could be nonlinear in $\boldsymbol{x}_1$, it is guaranteed to be linear in $\psi^*(\boldsymbol{x}_1)$.

Proof Sketch of Lemma 3.1. Lemma is proved by law of total expectation:

$$\begin{aligned}\psi^*(\cdot) := \mathbb{E}[X_2|X_1] =& \mathbb{E}[\mathbb{E}[X_2|X_1, Y]|X_1] = \mathbb{E}[\mathbb{E}[X_2|Y]|X_1] \qquad \text{(uses CI)}\\ =& \sum_y P(Y = y|X_1)\mathbb{E}[X_2|Y = y] =: f(X_1)^\top \boldsymbol{A},\end{aligned}$$

where $f(x_1)_y = P(Y = y|X_1 = x_1)$, and $\boldsymbol{A} \in \mathbb{R}^{\mathcal{Y}\times d_2}$ satisfies $\boldsymbol{A}_{y,:} = \mathbb{E}[X_2|Y = y]$. One could see that through predicting X_2, due to the CI assumption, ψ^* has implicitly encoded the information of $Y|X_1$. Finally due to the fact that matrix $\boldsymbol{A}$ is full rank, we get that f^* is linear in ψ^* as well. □

We see that besides CI, another important property is $\mathbb{E}[X_2|Y]$ being rank k. This means X_2 is correlated with every instance of Y, and thus captures information of every prediction class. This is naturally a necessary assumption for X_2 to be a reasonable pretext task for predicting Y. Note that this assumption does not trivialize the problem and that even though ψ is designed to predict X_2, it can still be a better representation than X_2 for downstream tasks. Note that Y does not have to be linear in X_2 but is proven to be linear in ψ, since ψ learns to ignore some information in X_2 that is irrelevant to Y. We provide this simple example for better understanding:

Example 3.1. *Let $Y \in \{-1, 1\}$ be binary labels, and X_1, X_2 be $2-$mixture Gaussian random variables with $X_1 \sim \mathcal{N}(Y\boldsymbol{\mu}_1, \mathbf{I}), X_2 \sim \mathcal{N}(Y\boldsymbol{\mu}_2, \mathbf{I})$. In this example, $X_1 \perp X_2|Y$. Although $\mathbb{E}[Y|X_2]$ and $\mathbb{E}[Y|X_1]$ are not linear, $\mathbb{E}[Y|\psi]$ is linear: $\psi(\boldsymbol{x}_1) = P(Y = 1|X_1 = \boldsymbol{x}_1)\boldsymbol{\mu}_2 - P(Y = -1|X_1 = \boldsymbol{x}_1)\boldsymbol{\mu}_2$ and $f^*(\boldsymbol{x}_1) = P(Y = 1|X_1 = \boldsymbol{x}_1) - P(Y = -1|X_1 = \boldsymbol{x}_1) \equiv \boldsymbol{\mu}_2^T\psi(\boldsymbol{x}_1)/\|\boldsymbol{\mu}_2\|^2$.*

Given that ψ^* is good for downstream, we now care about the sample complexity. We will need to assume that the representation has some nice concentration properties. We make an assumption about the whitened data $\psi^*(X_1)$ to ignore scaling factors.

Assumption 3.2. *We assume the whitened feature variable $U := \mathbf{\Sigma}_\psi^{-1/2}\psi(X_1)$ is a ρ^2-subgaussian random variable, where $\mathbf{\Sigma}_\psi = \mathbb{E}[\psi(X_1)\psi(X_1)^\top]$.*

We note that all bounded random variables satisfy sub-gaussian property.

Theorem 3.2 (General conditional independence)**.** *Fix a failure probability $\delta \in (0, 1)$, under the same assumption as Lemma 3.1 and Assumption 3.2 for ψ^*, if additionally $n \gg \rho^4(k + \log(1/\delta))$, then the excess risk of the learned predictor $\boldsymbol{x}_1 \to \hat{\boldsymbol{W}}^\top\psi^*(\boldsymbol{x}_1)$ on the downstream task satsifies*

$$\mathrm{ER}_{\psi^*}[\hat{\boldsymbol{W}}] \leq \tilde{\mathcal{O}}\left(\frac{k}{n_2}\sigma^2\right)^{3}$$

Remark 3.1. *This analysis assumes we could perfectly learn $\psi^* = \mathbb{E}[X_2|X_1]$ disregarding the number of samples in the SSL phase (unlabeled data is cheap to obtain). Here by sample complexity we refer to the labeled data (X_1, Y). We defer the effect of imprecise representation ψ in Section 4.*

Remark 3.2. *We note that since $X_1 \perp X_2 | Y$ ensures $X_1 \perp h(X_2)|Y$ for any deterministic function h, we could replace X_2 by $h(X_2)$ and all results hold. Therefore we could replace X_2 with $h(X_2)$ in our algorithm especially when $d_2 < km$.*

3.2 Function class induced by feature maps.

Given feature map $\phi_1 : \mathcal{X}_1 \to \mathbb{R}^{D_1}$, we consider the function class $\mathcal{H}_1 = \{\psi : \mathcal{X}_1 \to \mathbb{R}^{d_2} | \exists \boldsymbol{B} \in \mathbb{R}^{d_2 \times D_1}, \psi(\boldsymbol{x}_1) = \boldsymbol{B}\phi_1(\boldsymbol{x}_1)\}$.

Claim 3.3 (Closed form solution)**.** *The optimal function in $\mathcal{H}$ is $\psi^*(\boldsymbol{x}_1) = \boldsymbol{\Sigma}_{X_2\phi_1}\boldsymbol{\Sigma}_{\phi_1\phi_1}^{-1}\boldsymbol{x}_1$, where $\boldsymbol{\Sigma}_{X_2\phi_1} := \boldsymbol{\Sigma}_{X_2\phi_1(X_1)}$ and $\boldsymbol{\Sigma}_{\phi_1\phi_1} := \boldsymbol{\Sigma}_{\phi_1(X_1)\phi_1(X_1)}$.*

We again show the benefit of CI, but only comparing the performance of ψ^* to the original features ϕ_1. Since ψ^* is linear in ϕ_1, it cannot have smaller approximation error than ϕ_1. However CI will ensure that ψ^* has the same approximation error as ϕ_1 and enjoys better sample complexity.

Lemma 3.4 (Approximation error)**.** *If Assumption 3.1 is satisfied, and if the matrix $\boldsymbol{A} \in \mathbb{R}^{\mathcal{Y} \times d_2}$ with $\boldsymbol{A}_{y,:} := \mathbb{E}[X_2|Y = \boldsymbol{y}]$ is of rank $k = |\mathcal{Y}|$. Then $e_{apx}(\psi^*) = e_{apx}(\phi_1)$.*

We additionally need an assumption on the residual $a(\boldsymbol{x}_1) := \mathbb{E}[Y|X_1 = \boldsymbol{x}_1] - \mathbb{E}^L[Y|\phi_1(\boldsymbol{x}_1)]$.

Assumption 3.3. *(Bounded approx. error; Condition 3 in [28]) We have almost surely*

$$\|\boldsymbol{\Sigma}_{\phi_1\phi_1}^{-1/2}\phi_1(X_1)a(X_1)^\top\|_F \leq b_0\sqrt{k}$$

Theorem 3.5. *(CI with approximation error) Fix a failure probability $\delta \in (0, 1)$, under the same assumption as Lemma 3.4, Assumption 3.2 for ψ^* and Assumption 3.3, if $n_2 \gg \rho^4(k + \log(1/\delta))$, then the excess risk of the learned predictor $\boldsymbol{x}_1 \to \hat{\boldsymbol{W}}^\top\psi^*(\boldsymbol{x}_1)$ on the downstream task satisfies:*

$$\mathrm{ER}_{\psi^*}[\hat{\boldsymbol{W}}] \leq e_{apx}(\phi_1) + \tilde{\mathcal{O}}\left(\frac{k}{n_2}\sigma^2\right).$$

Thus with SSL, the requirement of labels is reduced from complexity for D_1 to $\mathcal{O}(k)$.

4 Beyond conditional independence

In the previous section, we focused on the case where we have exact CI. A weaker but more realistic assumption is that Y captures some portion of the dependence between X_1 and X_2 but not all. We quantify this notion of approximate ACI through a quantity ϵ_{CI}^2 (Definition 4.1), and show excess risk bounds for the representation learned from SSL[4]. In particular, the excess risk will have the form $\tilde{\mathcal{O}}\left(\frac{d_2}{n_2} + \epsilon_{\text{CI}}^2 + \epsilon_{\text{pre}}^2\right)$, which suggests that only $n_2 = \mathcal{O}(d_2)$ labeled samples will be required to get small error on downstream task, as long as approximate CI is satisfied (ϵ_{CI}^2 is small) and the pretext task is solved well enough (ϵ_{pre}^2 is small). This is in contrast to not doing SSL, where many more labeled samples will be required to learn a solve the downstream task that learns a complicated representation function from scratch. We now describe the SSL method on finite samples, followed by the definition of ACI which we use to discuss the main excess risk bound and its consequences.

SSL with finite samples and general function space: Let $\boldsymbol{X}_1^{\text{pre}} = [\boldsymbol{x}_1^{(1,\text{pre})}, \cdots, \boldsymbol{x}_1^{(n_1,\text{pre})}]^\top \in \mathbb{R}^{n_1 \times d_1}$ and $\boldsymbol{X}_2 = [\boldsymbol{x}_2^{(1)}, \cdots, \boldsymbol{x}_2^{(n_1)}]^\top \in \mathbb{R}^{n_1 \times d_2}$ be n_1 training samples for pretext task, where $(\boldsymbol{x}_1^{(i,\text{pre})}, \boldsymbol{x}_2^{(i)})$ is sampled from $P_{X_1X_2}$. The n_2 labeled samples for the downstream task are defined as $\boldsymbol{X}_1^{\text{down}} \in \mathbb{R}^{n_2 \times d_1}$, $\boldsymbol{Y} \in \mathbb{R}^{n_2 \times d_3}$[5]. Given a representation function space $\mathcal{H} : \mathcal{X}_1 \to \mathbb{R}^{d_2}$, we

[3] We will use $\tilde{O}$ to hide log factor $\log(k/\delta)$ or $\log(d_2/\delta)$.

[4] Results for jointly-Gaussian variables is in Appendix D.1; ACI is quantified by the partial covariance matrix.

[5] $d_3 = k$ and $Y \equiv \phi_y(Y)$ (one-hot encoding) refers multi-class classification task, $d_3 = 1$ refers to regression.

learn $\tilde{\psi}$ from $\mathcal{H}$ using the n_1 unlabeled samples and then use the n_2 labeled samples to learn a linear classifier on the learned representation $\tilde{\psi}(\boldsymbol{X}_1^{\text{down}})$ to fit $\boldsymbol{Y}$. This process is summarized below.

$$1)\ \ \tilde{\psi} := \arg\min_{f\in\mathcal{H}} \frac{1}{n_1}\|\boldsymbol{X}_2 - f(\boldsymbol{X}_1^{\text{pre}})\|_F^2,\ \ 2)\ \ \hat{\boldsymbol{W}} \leftarrow \arg\min_{\boldsymbol{W}} \frac{1}{2n_2}\|\boldsymbol{Y} - \tilde{\psi}(\boldsymbol{X}_1^{\text{down}})\boldsymbol{W}\|_F^2. \tag{2}$$

In our main results, we consider two types of function spaces: $\mathcal{H} \in \{\mathcal{H}_1, \mathcal{H}_u\}$. Recall that $\mathcal{H}_1 = \{\psi(\cdot) = \boldsymbol{B}\phi_1(\cdot); \boldsymbol{B} \in \mathbb{R}^{d_2\times D_1}\}$ is a class of *linear representations* induced by feature map $\phi_1 : \mathcal{X}_1 \to \mathbb{R}^{D_1}$. We use $\mathcal{H}_u$ to denote a function space with universal approximation power (e.g. deep networks) that ensures $\psi^* = \mathbb{E}[X_2|X_1] \in \mathcal{H}_u$. We define the optimal predictor in each case as $f^*_{\mathcal{H}}(X_1) = \mathbb{E}^L[Y|\phi_1(X_1)]$ when $\mathcal{H} = \mathcal{H}_1$, $f^*_{\mathcal{H}} = f^*$ for $\mathcal{H} = \mathcal{H}_u$, we define excess risk as

$$\text{ER}_{\tilde{\psi}}(\hat{\boldsymbol{W}}) := \mathbb{E}_{X_1}\left[\|f^*_{\mathcal{H}}(X_1) - \hat{\boldsymbol{W}}^\top\tilde{\psi}(X_1)\|_2^2\right].$$

Approximate conditional independence: Our new assumption will generalize Assumption 3.1 in two ways, 1) we allow for additional latent variables Z that together with Y could potentially make X_1 and X_2 independent, and 2) we allow this conditional independence to be approximate. Note that allowing for extra latent variable can trivially make X_1 and X_2 to be conditionally independent by picking a large enough Z (e.g. $Z = (X_1, X_2)$). However the following assumption, that needs the pretext target X_2 to correlate with all instances of variable $\bar{Y} = [Y, Z]$ (analogous to Lemma 3.1), will impose this restriction on how large Z can be.

Assumption 4.1 (Correlation between X_2 and Y, Z)**.** *Suppose there exists latent variable $Z \in \mathcal{Z}, |\mathcal{Z}| = m$ that ensures $\boldsymbol{\Sigma}_{\phi_{\bar{y}}X_2}$ is full column rank and $\|\boldsymbol{\Sigma}_{Y\phi_{\bar{y}}}\boldsymbol{\Sigma}^\dagger_{X_2\phi_{\bar{y}}}\|_2 = 1/\beta$, where $A^\dagger$ is pseudo-inverse, and $\phi_{\bar{y}}$ is the one-hot embedding for $\bar{Y} = [Y, Z]$.*

Just as in Section 3, this assumption will not assume away the problem (Example 3.1 can be suitably extended). The additional term $1/\beta$ here captures both the "scale" of X_2 and also the strength of correlation between X_2 and $[Y, Z]$ that was discussed after Lemma 3.1. For $\boldsymbol{\Sigma}_{\phi_{\bar{y}}X_2}$ to be full column rank, it is essential that $d_2 \geq km$, and this already gives an upper bound on the size of Z. Given this restriction on Z (and thus $\bar{Y}$), we define the notion of approximate conditional independence.

Definition 4.1 (Approximate conditional independence with function space $\mathcal{H}$)**.** *For $\bar{Y} = [Y, Z]$,*

1. For $\mathcal{H} = \mathcal{H}_1$, define $\epsilon_{CI} := \|\boldsymbol{\Sigma}^{-1/2}_{\phi_1\phi_1}\boldsymbol{\Sigma}_{\phi_1X_2|\phi_{\bar{y}}}\|_F$.

2. For $\mathcal{H} = \mathcal{H}_u$, define $\epsilon^2_{CI} := \mathbb{E}_{X_1}[\|\mathbb{E}[X_2|X_1] - \mathbb{E}_{\bar{Y}}[\mathbb{E}[X_2|\bar{Y}]|X_1]\|^2]$.

Firstly we note that this is indeed an extension of exact CI, since exact CI in both cases will imply that $\epsilon_{\text{CI}} = 0$. We present a unified analysis in the appendix that shows the ϵ_{CI} for the second case is same as the first case, with covariance operators instead of matrices (A direct derivation is in Claim D.8). We also present more relaxed and general form of the above assumptions in Appendix E.1. With this assumption, we are ready to present our main bound.

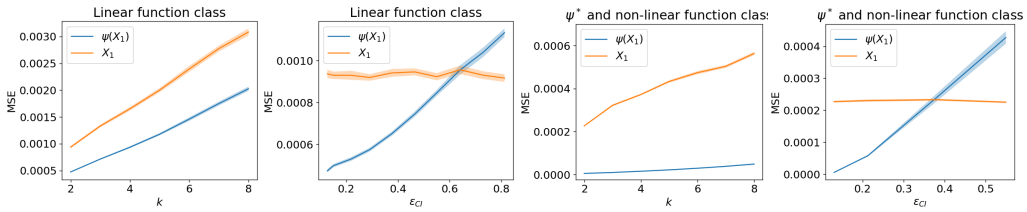
Bound on excess risk: Recall that we assume that the residual term $N := Y - \mathbb{E}[Y|X_1]$ is mean zero and σ^2-subgaussian. Before showing our main result, analogous to Assumption 3.3, for the class $\mathcal{H}_1$ with non-universal features ϕ_1, we will need an assumption[6] on the residual $a := f^* - f^*_{\mathcal{H}_1} = \mathbb{E}[Y|X_1] - \mathbb{E}^L[Y|\phi_1(X_1)]$:

Assumption 4.2. *(Bounded approximation error on pretext phase [28]) There exists a universal constant b_0, such that $\|\boldsymbol{\Sigma}^{-1/2}_{\phi_1\phi_1}\phi_1(X_1)a(X_1)^\top\|_F \leq b_0\sqrt{d_2}$ almost surely.*

Theorem 4.2. *For a fixed $\delta \in (0, 1)$, under Assumptions 4.1, 4.2 for $\tilde{\psi}$ and ψ^* and 3.2 for non-universal feature maps, if $n_1, n_2 \gg \rho^4(d_2 + \log 1/\delta)$, and we learn the pretext tasks such that: $\mathbb{E}\|\tilde{\psi}(X_1) - \psi^*(X_1)\|_F^2 \leq \epsilon^2_{pre}$. Then the generalization error for downstream task w.p. $1 - \delta$ is:*

$$\text{ER}_{\tilde{\psi}}(\hat{\boldsymbol{W}}) \leq \tilde{\mathcal{O}}\left(\underbrace{\sigma^2\frac{d_2}{n_2}}_{\textit{estimation error}} + \underbrace{\frac{\epsilon^2_{CI}}{\beta^2} + \frac{\epsilon^2_{pre}}{\beta^2}}_{\textit{approximation error}}\right) \tag{3}$$

[6]This rules out the failure if one chooses a very simple function class to learn $\mathbb{E}[X_2|X_1]$. In practice we usually use neural networks (with universal approximation power) and this bound should be very small.



(a) (b) (c) (d)

Figure 1: **Left two**: how MSE scales with k (the dimension of Y) and ϵ_{CI} (Definition 4.1) with the linear function class. **Right two**: how MSE scales with k and ϵ with ψ^* and non-linear function class. Mean of 30 trials are shown in solid line and one standard error is shown by shadow.

We defer the proof to the appendix. The proof technique is similar to that of Section 3. The difference is that now $\tilde{\psi}(\boldsymbol{X}^{(\text{down})}) \in \mathbb{R}^{n_2 \times d_2}$ will be an approximately low rank matrix, where the low rank part is the high-signal features that implicitly comes from Y, Z that can linearly learn downstream task. The remaining part comes from ϵ_{CI} and ϵ_{pre} and causes the approximation error. Again by selecting the top km (dimension of $\phi_{\bar{y}}$) features we could further improve the bound:

Remark 4.1. *By applying PCA on $\tilde{\psi}(\boldsymbol{X}_1^{down})$ and keeping the top km principal components only, we can improve the bound in Theorem 4.2 to* $\text{ER}_{\tilde{\psi}}(\hat{\boldsymbol{W}}) \leq \tilde{\mathcal{O}}\left(\sigma^2 \frac{km}{n_2} + \frac{\epsilon_{CI}^2}{\beta^2} + \frac{\epsilon_{pre}^2}{\beta^2}\right)$.

We take a closer look at the different sources of errors in Lemma 4.1: 1) The first term is estimation error on learning with finite samples n_2 with noise level σ^2 in $Y - f^*(X_1)$; 2) ϵ_{CI} measures the approximate CI; and 3) ϵ_{pre} is the error from not learning the pretext task exactly. The first term is optimal ignoring log factors as we do linear regression on mk-dimensional features. The second and third term together form approximation error. They are non-reducible due to the fact that f^* is not exactly linear in ψ and we use it as a fixed representation. Fine-tuning the representations might be necessary to get rid of these terms when we have sufficient downstream labeled data. We leave this exploring this as future work. Compared to traditional supervised learning, learning $f^*_{\mathcal{H}}$ requires sample complexity scaling with the (Rademacher/Gaussian) complexity of $\mathcal{H}$ (see e.g. [6, 47]), which is very large for complicated models such as deep networks. Thus SSL can significantly reduce the labeled sample complexity down from this complexity measure of $\mathcal{H}$ to $\tilde{\mathcal{O}}(km)$, demonstrating the power of predicting what you already know using unlabeled data. In Section G, we consider a similar result for classification.

5 Experiments

In this section, we empirically verify our claim that SSL performs well when ACI is satisfied. More details for experiments can be found in Section I, including experiments in the text domain.

Simulations. With synthetic data, we verify how excess risk (ER) scales with the cardinality/feature dimension of $\mathcal{Y}$ (k), and ACI (ϵ_{CI} in Definition 4.1). We consider a mixture of Gaussian data and conduct experiments with both linear function space ($\mathcal{H}_1$ with ϕ_1 as identity map) and universal function space $\mathcal{H}_u$. We sample the label Y uniformly from $\{1, ..., k\}$. For i-th class, the centers $\mu_{1i} \in \mathbb{R}^{d_1}$ and $\mu_{2i} \in \mathbb{R}^{d_2}$ are uniformly sampled from $[0, 10)$. Given $Y = i$, $\alpha \in [0, 1]$, let $X_1 \sim \mathcal{N}(\mu_{1i}, \mathbf{I})$, $\hat{X}_2 \sim \mathcal{N}(\mu_{2i}, \mathbf{I})$, and $X_2 = (1 - \alpha)\hat{X}_2 + \alpha X_1$. Therefore α is a correlation coefficient: $\alpha = 0$ ensures X_2 being CI with X_1 given Y and when $\alpha = 1$, X_2 fully depends on X_1. (if $d_1 \neq d_2$, we append zeros or truncate to fit accordingly).

We first conduct experiments with linear function class. We learn a linear representation ψ with n_1 samples and the linear prediction of Y from ψ with n_2 samples. We set $d_1 = 50$, $d_2 = 40$, $n_1 = 4000$, $n_2 = 1000$ and ER is measured with Mean Squared Error (MSE). As shown in Figure 1(a)(b), the MSE of learning with $\psi(X_1)$ scales linearly with k as indicated in Theorem 3.5, and scales linearly with ϵ_{CI} associated with linear function class as indicated in Theorem 4.2. Next we move on to general function class, i.e., $\psi^* = \mathbb{E}[Y|X_1]$ with a closed form solution (see example 3.1). We use the same parameter settings as above. For baseline method, we use kernel linear regression to predict Y using X_1 (we use RBF kernel which also has universal approximation power). As shown

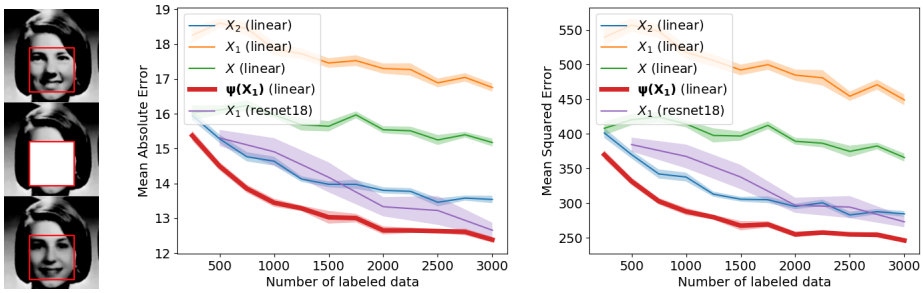


Figure 2: **Left**: Example of the X_2 (in the red box of the 1st row), the X_1 (out of the red box of the 1st row), the input to the inpainting task (the second row), $\psi(X_1)$ (the 3 row in the red box), and in this example $Y = 1967$. **Middle**: Mean Squared Error comparison of yearbook regression predicting dates. **Right**: Mean Absolute Error comparison of yearbook regression predicting dates. Experiments are repeated 10 times, with mean shown in solid line and one standard deviation in shadow.

in Figure 1(c)(d), the phenomenon is the same as what we observe in the linear function class setting, and hence they respectively verify Theorem 3.2 and Theorem 4.2 with $\mathcal{H}_u$.

Computer Vision Task. We verify if learning from ψ is more effective than learning directly from X_1, in a realistic setting (without enforcing conditional independence). Specifically, we test on the Yearbook dataset [20], and try to predict the date when the portraits are taken (denoted as Y_D), which ranges from 1905 to 2013. We resize all the portraits to be 128 by 128. We crop out the center 64 by 64 pixels (the face), and treat it as X_2, and treat the outer rim as X_1 as shown in Figure 2. Our task is to predict Y_D, which is the year when the portraits are taken, and the year ranges from 1905 to 2013. For ψ, we learn X_2 from X_1 with standard image inpainting techniques [43], and full set of training data (without labels). After that we fix the learned ψ and learn a linear model to predict Y_D from ψ using a smaller set of data (with labels). Besides linear model on X_1, another strong baseline that we compare with is using ResNet18 [25] to predict Y_D from X_1. With the full set of training data, this model is able to achieve a Mean Absolute Difference of 6.89, close to what state-of-the-art can achieve [20]. ResNet18 has similar amount of parameters as our generator, and hence roughly in the same function class. We show the MSE result as in Figure 2. Learning from ψ is more effective than learning from X_1 or X_2 directly, with linear model as well as with ResNet18. Practitioner usually fine-tune ψ with the downstream task, which leads to more competitive performance [43].

6 Conclusion

In this work we theoretically quantify how an approximate conditional independence assumption that connects pretext and downstream task data distributions can give sample complexity benefits of self-supervised learning on downstream tasks. Our theoretical findings are also supported by experiments on simulated data and also on real CV and NLP tasks. We would like to note that approximate CI is only a sufficient condition for a useful pretext task. We leave it for future work to investigate other mechanisms by which pretext tasks help with downstream tasks.

References

[1] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

[2] Rie Kubota Ando and Tong Zhang. Two-view feature generation model for semi-supervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 25–32, 2007.

[3] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[4] Charles R Baker. Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, 186:273–289, 1973.

[5] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[6] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[7] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 1998.

[8] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 11190–11201, 2019.

[9] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. *arXiv preprint arXiv:2003.12862*, 2020.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[11] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

[14] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

[15] Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

[16] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017.

[17] Kenji Fukumizu, Francis R Bach, and Michael I Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5(Jan):73–99, 2004.

[18] Kenji Fukumizu, Francis R Bach, Michael I Jordan, et al. Kernel dimension reduction in regression. *The Annals of Statistics*, 37(4):1871–1905, 2009.

[19] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[20] Shiry Ginosar, Kate Rakelly, Sarah Sachs, Brian Yin, and Alexei A Efros. A century of portraits: A visual historical record of american high school yearbooks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1–7, 2015.

[21] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[23] David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

[24] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

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

[26] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, pages 15637–15648, 2019.

[27] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[28] Daniel Hsu, Sham M Kakade, and Tong Zhang. Random design analysis of ridge regression. In *Conference on learning theory*, pages 9–1, 2012.

[29] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[30] Tzee-Ming Huang. Testing conditional independence using maximal nonlinear conditional correlation. *The Annals of Statistics*, 38(4):2047–2091, 2010.

[31] Eric Jang, Coline Devin, Vincent Vanhoucke, and Sergey Levine. Grasp2vec: Learning object representations from self-supervised grasping. *arXiv preprint arXiv:1811.06964*, 2018.

[32] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[33] Sham M Kakade and Dean P Foster. Multi-view regression via canonical correlation analysis. In *International Conference on Computational Learning Theory*, pages 82–96. Springer, 2007.

[34] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019.

[35] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. In *Proceedings of the International Conference on Learning Representations*, 2018.

[36] Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[37] Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667, 2006.

[38] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 2013.

[39] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[40] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*, 2020.

[41] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

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

[43] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[44] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[45] Michael Reed. *Methods of modern mathematical physics: Functional analysis*. Elsevier, 2012.

[46] Nikunj Saunshi, Sadhika Malladi, and Sanjeev Arora. A mathematical exploration of why language models help solve downstream tasks. *arXiv preprint arXiv:2010.03648*, 2020.

[47] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[48] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013.

[49] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[50] Yuandong Tian, Lantao Yu, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning with dual deep networks. *arXiv preprint arXiv:2010.00578*, 2020.

[51] Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive estimation reveals topic posterior information to linear models. *arXiv preprint arXiv:2003.02234*, 2020.

[52] Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. *arXiv preprint arXiv:2008.10150*, 2020.

[53] Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Demystifying self-supervised learning: An information-theoretical framework. *arXiv preprint arXiv:2006.05576*, 2020.

[54] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

[55] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[56] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[57] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *arXiv preprint arXiv:2005.10242*, 2020.

[58] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[59] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018.

[60] Han Yang, Xiao Yan, Xinyan Dai, and James Cheng. Self-enhanced gnn: Improving graph neural networks using model outputs. *arXiv preprint arXiv:2002.07518*, 2020.

[61] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[62] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017.

[63] Zaiwei Zhang, Zhenxiao Liang, Lemeng Wu, Xiaowei Zhou, and Qixing Huang. Path-invariant map networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11084–11094, 2019.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 1, especially the contribution paragraph.
   (b) Did you describe the limitations of your work? [Yes] See the Conclusion section and line 314-326.
   (c) Did you discuss any potential negative societal impacts of your work? [No] This is mainly a theoretical work with simulations and should not cause much social impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Every assumption is listed as a standalone block. Every theorem or lemma is self-contained.
   (b) Did you include complete proofs of all theoretical results? [Yes] For every theorem, lemma, claim, and some technical remarks, we indicate where their proof is located in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The codes will be made public after this work is accepted for publish.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The details are provided in the sections 5 and I

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 1, Figure 5 and Figure 2.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] The details are provided in the Appendix

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section 5.

(b) Did you mention the license of the assets? [Yes] The details are provided in the Appendix

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] All the portrait in the Yearbook dataset are in public domain. Please find the detailed discussion in Section 3 of the Yearbook dataset paper [20]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]