Low-Rank Approximation of Structural Redundancy for Self-Supervised Learning

Kang Du   KANG.DU@UTAH.EDU
University of Utah

Yu Xiang   YU.XIANG@UTAH.EDU
University of Utah

Editors: Francesco Locatello and Vanessa Didelez

Abstract

We study the data-generating mechanism for reconstructive SSL to shed light on its effectiveness. With an infinite amount of labeled samples, we provide a sufficient and necessary condition for perfect linear approximation. The condition reveals a full-rank component that preserves the label classes of Y, along with a redundant component. Motivated by the condition, we propose to approximate the redundant component by a low-rank factorization and measure the approximation quality by introducing a new quantity ε_s, parameterized by the rank of factorization s. We incorporate ε_s into the excess risk analysis under both linear regression and ridge regression settings, where the latter regularization approach is to handle scenarios when the dimension of the learned features is much larger than the number of labeled samples n for downstream tasks. We design three stylized experiments to compare SSL with supervised learning under different settings to support our theoretical findings.

Keywords: Self-supervised learning, redundancy, low-rank approximation, ridge regression.

1. Introduction

Reconstructive self-supervised learning (SSL) has been highly successful in various fields (Pathak et al., 2016; Vincent et al., 2010; Radford et al., 2018; Devlin et al., 2018), where the theme is to extract representations from unlabeled data that are potentially useful for downstream tasks. One of the major advantages of SSL is its significantly reduced dependency on labeled data. Despite abundant empirical evidence, the theoretical understanding of the performance of SSL under limited labeled data is still insufficient.

In reconstructive SSL, a *pretext task* is designed to predict a target X_2 with input features X_1, which yields the learned representation $\psi(X_1)$. Then, the downstream task is to predict the target Y using $\psi(X_1)$. Whether the learned representation is useful for the downstream task relies on the connections between the pretext and downstream tasks. To bridge the pretext and downstream tasks, the conditional independence (CI) assumption, namely $X_1 \perp\!\!\!\perp X_2 \,|\, Y$, has been studied in the seminal work (Lee et al., 2021). For the classification setting, they show that CI is a *sufficient condition* for a linear predictor to be optimal for the downstream task, that is, $\psi(X_1)$ *can linearly predict Y perfectly with an infinite number of samples available for the task*. Motivated by this key observation, they provide theoretical guarantees showing the superior sample complexity of SSL under general *approximate* conditional independence settings. However, a fundamental theoretical question for understanding reconstructive SSL still remains:

What is the sufficient and necessary condition on (X_1, X_2, Y), in the classification setting, for a linear predictor to be optimal for the downstream task?

To address this question, it is helpful to express X_2 as $X_2 = h(X_1, Y) + N$, where (X_1, Y) is for an arbitrary supervised learning task and $N := X_2 - \mathsf{E}[X_2|X_1, Y]$. With this expression, roughly speaking, the target Y can be decoded from X_2 if h is invertible in some sense. We formalize this notion of invertibility by focusing on classification problems. Our formulation allows for general dependency between X_2 and X_1 when conditioning on Y. Thus there are features in the learned representation $\psi(X_1)$ that are redundant for the prediction of Y. For instance, for image classification problems, the redundant features may come from the background in the image; if the object of interest is surrounded by other objects in the background, a pretext task of predicting blocked patches of the image may mistakenly extract too many features from the background (Pathak et al., 2016). Without any constraints, a large percentage of redundant features can potentially make SSL fail. To this end, we introduce a quantity ε_s, indexed by rank s, for a low-rank approximation of the redundancy in the learned representation. We show how ε_s affects the performance of SSL through both of our theoretical analysis and experiments.

Our main contributions are summarized below.

1. Under the classification setting, we characterize in Section 3 a sufficient and necessary condition for a linear predictor to be optimal in the downstream task.
2. In Section 4, we introduce a low-rank approximation quantity to characterize the redundancy in the learned representation.
3. Based on the low-rank approximation quantity, we derive finite sample bounds on the excess risk and the corresponding sample complexity for both ordinary least squares and ridge regression estimators in Section 5.
4. In Section 6.1, we design a simulation setting to demonstrate the effectiveness of the low-rank approximation. Our sufficient and necessary condition is partially verified through two computer vision tasks in Section 6.2.

1.1. Related work

Reconstructive SSL is focused on recovering deliberately concealed information in the data. In computer vision, examples include the prediction of blocked patches (Pathak et al., 2016), recovering the color (Zhang et al., 2016), denoising (Vincent et al., 2010), and identifying the rotated angle (Gidaris et al., 2018), while the simple scheme of next word prediction is widely adopted in NLP (Radford et al., 2018; Devlin et al., 2018). From the theoretical perspective, (Saunshi et al., 2020) and (Wei et al., 2021) study how pre-trained language models yield useful representation for downstream tasks. For computer vision tasks, (Pathak et al., 2016) provides a theoretical understanding of features learned by auto-encoders under a multi-view data assumption. Under a general formulation of reconstructive SSL, (Lee et al., 2021) shows that CI is sufficient for a linear predictor to be optimal in the downstream task and provides finite sample analysis. Since CI often fails to hold in practical settings, (Teng et al., 2022) proposes to modify the unlabeled data to make CI hold. Their theoretical analysis suggests that the modification is hurtful rather than helpful for the performance of SSL. The other popular type of SSL is called contrastive SSL, where the goal is to learn representations that make different views of the same data point closer. The CI assumption has been adopted in (Arora et al., 2019; Tosh et al., 2021) to provide theoretical guarantees for contrastive learning. In the context of contrastive SSL, CI is a natural assumption since, ideally, two views are

expected to share less information given the label. The literature on SSL is vast and we refer the readers to (Gui et al., 2023; Ozbulak et al., 2023) for detailed reviews.

1.2. Notation

Throughout the paper, $\|\cdot\|$ denotes the l_2 norm for vectors or Frobenius norm for matrices. We use $\mathbf{0}$ and $\mathbf{1}$ to denote vectors or matrices of zeros and ones, respectively. For a full (column) rank matrix $A \in \mathbb{R}^{m \times n}$ with $n < m$, we use $A^\dagger$ to denote its (left) pseudoinverse. Let $\mathrm{Cov}(X)$ denote the covariance matrix of a random vector X and $\mathrm{Cov}(X_1, X_2)$ denote that of two random vectors X_1 and X_2. We use $\widetilde{\mathcal{O}}$ to hide log factors and $\lesssim$ to hide constants in inequalities. We use $\mathbf{I}_d$ to denote the identity matrix of size $d \times d$. A random vector $X \in \mathbb{R}^d$ is said to be σ^2-sub-Gaussian if $\mathsf{E}[X] = \mathbf{0}$ and $\mathsf{E}[e^{tu^\top X}] \le e^{\frac{t^2 \sigma^2}{2}}$ for any $t \in \mathbb{R}$ and $u \in \mathbb{R}^d$ such that $\|u\| = 1$.

2. Problem Formulation: Reconstructive SSL

Consider $(X_1, X_2, Y) \in \mathcal{X}_1 \times \mathcal{X}_2 \times \mathcal{Y}$ where $X_2 \in \mathbb{R}^{d_2}$ and Y are the target variables for the pretext and downstream tasks, respectively, and $X_1 \in \mathbb{R}^{d_1}$ is a vector of features shared by the two prediction tasks. We focus on the classification setting for the downstream task, i.e., Y is categorical. For regression problems, one can consider the continuous target variable being discretized to a set of values. We use $\bar{Y} \in \bar{\mathcal{Y}} = \{y_1, \dots, y_{p+1}\}$ to denote the original label variable and $Y = (\mathbb{1}_{\bar{Y} = y_1}, \dots, \mathbb{1}_{\bar{Y} = y_p})^\top$ to denote its one-hot encoding with one class excluded to avoid multicolinearity as $\sum_{i=1}^{p+1} \mathbb{1}_{\bar{Y} = y_i} = 1$, and we will simply refer to Y as the one-hot encoding of $\bar{Y}$ throughout this work. We assume $p < d_2$ throughout the work. For simplicity, we assume that the optimal predictors for different classes of Y are not linearly dependent, i.e., $\mathrm{Cov}(\mathsf{E}[Y|X_1])$ has full rank; otherwise, certain classes of Y can be hidden to make it hold.

Concretely, we consider the following reconstructive SSL procedure.

1. *Pretext task*: Given unlabeled data, predict X_2 using X_1 under some function class Ψ, i.e., estimate $\psi^* := \arg\min_{\psi \in \Psi} \mathsf{E}[\|X_2 - \psi(X_1)\|^2]$.

2. *Downstream task*: Given n labeled data, regress Y on the learned representation $\psi^*(X_1)$ using *simple* regression functions such as linear or ridge regression.

Since there is often a large amount of unlabeled data and one can adopt deep neural networks to achieve universal approximation, we fix $\psi^*(x) := \mathsf{E}[X_2|X_1 = x]$ and focus on analyzing the downstream task. Due to the nature of the small (labeled) sample size of SSL, the function class for the downstream task is often assumed to have lower complexity compared to Ψ (e.g., smaller parameter space). For theoretical analysis, we consider the class of all linear functions for the downstream task similarly as in (Lee et al., 2021). In practice, the advantage of SSL over supervised learning (SL) is more significant when the labeled sample size n is relatively small, in which case the dimension of ψ^* can be larger than n. To avoid the downstream task being ill-posed, we adopt the ridge estimator. To measure the gap between the SSL prediction and the optimal predictor $\mathsf{E}[Y|X_1]$ in infinite and finite samples, respectively, we define the approximation error and excess risk.

Definition 1 *Define the approximation error of SSL as* $\mathrm{error}^*_{apx} := \min_\beta \mathrm{error}_{apx}(\beta)$, *where*

$$\mathrm{error}_{apx}(\beta) := \mathsf{E}\left[\|\mathsf{E}[Y|X_1] - \beta\psi^*(X_1)\|^2\right] \tag{1}$$

with $\psi^*(x) = \mathsf{E}[X_2|X_1 = x]$, the optimal predictor of X_2 given X_1.

Note that $\psi^*(x) = \mathsf{E}[X_2|X_1 = x]$ can be ensured by a function class with universal approximation power such as deep neural networks.

Definition 2 *We say there is an exact matching between Y and X_2 given X_1 if* $\mathsf{error}^*_{apx} = 0$.

For simplicity, we will omit the intercept $b(\beta) := \mathsf{E}[Y] - \beta\,\mathsf{E}[X_2]$ throughout the work. The performance of the downstream task is usually quantified through the so-called *excess risk* defined with respect to the finite sample analysis. Denote $X := \psi^*(X_1) = \mathsf{E}[X_2|X_1]$. Let $\boldsymbol{X}_1 \in \mathbb{R}^{n\times d_1}$ and $\boldsymbol{Y} \in \mathcal{Y}^{n\times p}$ be the labeled data, and $\boldsymbol{X} := \psi^*(\boldsymbol{X}_1) \in \mathbb{R}^{n\times d_2}$ denote the learned representation from pretraining. For the downstream task and $\lambda \geq 0$, let

$$\hat{\beta}_\lambda := \arg\min_\beta \frac{1}{n}\|\boldsymbol{Y} - \boldsymbol{X}\beta^\top\|^2 + \lambda\|\beta\|^2 = \boldsymbol{Y}^\top\boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X} + \lambda n\mathbf{I}_{d_2})^{-1}.$$

Definition 3 *The excess risk induced by the estimator $\hat{\beta}_\lambda$ is defined as $\mathcal{R}(\hat{\beta}_\lambda) := \mathsf{error}_{apx}(\hat{\beta}_\lambda)$.*

The term "matching" can be viewed in the following sense: (1) the form of nonlinearity in $\mathsf{E}[Y|X_1]$ should be captured by $\mathsf{E}[X_2|X_1]$; (2) the "redundant" nonlinearity in $\mathsf{E}[X_2|X_1]$ should be linearly dependent so that they can be removed through a linear transform. As a toy example, consider $\mathsf{E}[Y|X_1] = X_1^2$ and $\mathsf{E}[X_2|X_1] = (-X_1^2 + \sin(X_1),\, 0.5\sin(X_1))^\top$ and note they share the same quadratic term X_1^2, while the sine functions in $\mathsf{E}[X_2|X_1]$ are redundant for predicting Y. Observe that SSL with $\beta = (-1, 2)^\top$ extracts the quadratic term while eliminating the sine functions. In contrast, $\mathsf{E}[X_2|X_1] = (X_1,\, 0.5\cos(X_1))^\top$ will not lead to an exact matching.

Remark 4 *This notion of predicting a subset of X can be helpful for predicting Y is not limited to reconstructive SSL. For instance, in a series of recent papers Du and Xiang (2022, 2023a,b), the authors have explored a similar direction from an invariance perspective for multi-environment domain adaption, which has partially motivated this study.*

3. Necessary and Sufficient Condition for Exact Matching

In an attempt to demystify the matching between the pretext and downstream tasks, we propose to identify the conditions on the generating mechanism of (X_1, X_2, Y) that enable an exact matching. The generating mechanism of (X_1, Y) in a supervised learning task is often complicated, and thus we make no assumptions on how (X_1, Y) is generated and focus on the interactions between (X_1, Y) and X_2. Without loss of generality, we can write X_2 in the following form

$$X_2 = h(X_1, Y) + N, \tag{2}$$

where $h(X_1, Y) := \mathsf{E}[X_2|X_1, Y]$ is the regression function of X_2 on (X_1, Y) and therefore the residual variable $N := X_2 - h(X_1, Y)$ satisfies $\mathsf{E}[N|X_1, Y] = 0$. The function h captures how the label Y and feature X_1 are encoded into X_2.

Equation (2) can be viewed from a causal perspective through a general structural causal model (SCM) (Pearl, 2009), $X_2 = f(X_1, Y, \varepsilon)$, where ε is a vector of exogenous variables independent of (X_1, Y). Since this general SCM suffers from identifiability issues, we focus on (2), observing that

$h(X_1, Y) := \mathsf{E}[X_2|X_1, Y] = \mathsf{E}[f(X_1, Y, \varepsilon)|X_1, Y]$. It is important to note that (2) is valid even when there is no underlying causal graph over (X_1, X_2, Y).

Recall that Y is the one-hot encoding of $\bar{Y}$. Observe that an arbitrary function $h : (\mathcal{X}, \mathcal{Y}) \to \mathbb{R}^d$ can be equivalently written as

$$h(X, Y) = \sum_{j=1}^{p} h(X, e_j)\mathbb{1}_{\bar{Y}=y_j} = \sum_{j=1}^{p} h(X, e_j)e_j^\top Y := C^h(X)Y, \tag{3}$$

where we use the fact that $\mathbb{1}_{\bar{Y}=y_j} = e_j^\top Y$. This simple derivation implies a one-to-one correspondence between h and C^h, meaning that the general function model (2) can be expressed as

$$X_2 = h(X_1, Y) + N = C^h(X_1)Y + N, \tag{4}$$

with $C^h : \mathcal{X}_1 \to \mathbb{R}^{d_2 \times p}$. The role of the latent random matrix $C^h(X_1)$ is to encode the label variable Y into X_2, thus we call C^h the *encoding function*. The expression in (4) implies the identity $\mathsf{E}[X_2|X_1] = C^h(X_1)\mathsf{E}[Y|X_1]$, which is equivalent to

$$\mathsf{E}[X_2|X_1] = (C^h(X_1) + O(X_1))\mathsf{E}[Y|X_1] := \bar{C}^h(X_1)\mathsf{E}[Y|X_1], \tag{5}$$

for any $O : \mathcal{X}_1 \to \mathbb{R}^{d_2 \times p}$ such that $O(X_1)\mathsf{E}[Y|X_1] = \mathbf{0}$. In words, the rows of $O(x)$ are orthogonal to $\mathsf{E}[Y|X_1 = x]$ for $\forall x \in \mathcal{X}_1$. We call such $O(x)$ an orthogonal term. For instance, the orthogonality holds when $\mathsf{E}[Y|X_1] = (X_1, X_1^2)^\top$ and each row of $O(X_1)$ is $(-X_1, 1)$. Equation (5) defines an equivalent class of encoding functions $\mathcal{C} = \{\bar{C}^h\}$ that results in the same pretext representation $\psi^*(X_1) = \mathsf{E}[X_2|X_1]$. This shows that such orthogonal terms do not affect the analysis of SSL, and thus we use $\dot{=}$ to hide the orthogonal term (added to C^h) in equations throughout the paper.

Proposition 5 *The exact matching in Definition 2 holds if and only if $\beta C^h(x) \dot{=} \mathbf{I}_p$, for $\forall x \in \mathcal{X}_1$ and some $\beta \in \mathbb{R}^{p \times d_2}$.*

Therefore, in this formulation, finding an exact matching is equivalent to inverting the encoding function C^h. Proposition 5 implies that the full rank of $C^h(x)$ for every $x \in \mathcal{X}_1$ is a necessary condition for the exact matching. In the following lemma, we provide *a sufficient and necessary condition for the exact matching* through a full characterization of the invertibility of C^h.

Lemma 6 (sufficient and necessary condition for exact matching) *There is an exact matching between Y and X_2 given X_1 if and only if*

$$C^h(x) \dot{=} A \begin{bmatrix} \mathbf{I}_p \\ R(x) \end{bmatrix} \quad \text{for } \forall x \in \mathcal{X}_1, \tag{6}$$

for some invertible matrix $A \in \mathbb{R}^{d_2 \times d_2}$, an arbitrary matrix function $R : \mathcal{X}_1 \to \mathbb{R}^{(d_2-p) \times p}$.

The identity mapping $\mathbf{I}_p$ fully preserves each class of Y, and $R(x)$ represents the redundancy encoded into X_2. It is worth noting that redundancy refers to the features extracted from X_1 that are predictive for X_2, but redundant for the prediction of Y (given the optimal predictor $\mathsf{E}[Y|X_1]$). In our stylized MNIST experiment in Section 6.2.2, the dash pattern in the background is redundancy since it is useful for predicting the image orientation (i.e., X_2), but it contains no information about

the label. In contrast, the dot pattern is not redundant, since it is independent of both the image orientation and the label. The lemma above reveals that the label Y should be encoded into X_2 through an invertible linear mixture (i.e., A) of the full label information and some redundancy. When A is an identity matrix, the first p rows of $\psi^*(X_1) = \mathsf{E}[X_2|X_1]$ capture the full label information, thus the downstream task has a sparse solution $\beta^* = [\mathbf{I}_p, \mathbf{0}]$. However, the solutions to the downstream task may not be sparse in general, and we handle this challenge in Section 4. Below are two examples with explicit forms of C^h.

Example 1 *An important special case of model (4) is $X_2 = \widetilde{C}Y + N$, where $C^h \equiv \widetilde{C}$ is a constant function. In this case, the necessary and sufficient condition simplifies to the condition that $\widetilde{C}$ has full rank. Observe that $\mathsf{E}[X_2|X_1] = \widetilde{C}\,\mathsf{E}[Y|X_1]$ implies $\mathsf{error}_{apx}(\widetilde{C}^\dagger) = 0$.*

In Appendix A, we show that model $X_2 = \widetilde{C}Y + N$ is equivalent to $\mathsf{E}[X_2|X_1, Y] = \mathsf{E}[X_2|Y]$, which we call *conditional mean independence*, which is weaker than CI, i.e., $X_1 \perp\!\!\!\perp X_2 \,|\, Y$. The setting in Example 1 has been studied in (Lee et al., 2021) under CI. Despite the simplicity of conditional mean independence, it can be unrealistic in practical settings, since it requires that the label Y is fully encoded into X_2 with no redundant information (depending on X_1) as if the pretext and downstream tasks are two equivalent prediction tasks. Even though approximate conditional independence has been studied in (Lee et al., 2021), it is unclear if the approximation provides sufficient insights into explaining why and when SSL works (or fails), since conditional independence (or constant C^h) is not a necessary condition for exact matching.

Example 2 (partially linear model) *Define an invertible matrix $A \in \mathbb{R}^{d_2 \times d_2}$ with a column partition as $A = [A_1|A_2|A_3]$, where A_1 has p columns, A_2 has k columns such that $1 \le k \le d_2 - p$, and A_3 has the rest of the columns. Let $X_2 = A_1Y + A_2a(X_1) + N$, where $a : \mathcal{X}_1 \to \mathbb{R}^k$ satisfies $\mathsf{E}[a(X_1)] = \mathbf{0}$. Its encoding function is $C^h(X_1) = A_1 + A_2a(X_1)\mathbf{1}^\top$ as derived below. The sufficient and necessary condition is immediately satisfied with A and $R(x) = [\mathbf{1}a^\top(x), \mathbf{0}]^\top$, where $R(x)$ has $d_2 - p - k$ all zero rows.*

$$h(X_1, Y) = \sum_{j=1}^{p}(A_2a(X_1) + A_1e_j)e_j^\top Y = \big(A_2a(X_1)\sum_j e_j^\top + A_1\sum_j e_je_j^\top\big)Y := C^h(X_1)Y.$$

Even though conditional independence fails to hold since C^h is not constant, according to Lemma 6, there is still an exact matching.

4. Structural Redundancy

The pretext representation $\psi^*(X_1)$ is typically high-dimensional, designed to capture abundant information for various downstream tasks. Given limited labeled samples ($n \ll d_2$ in our notation), the downstream task is a high-dimensional linear regression problem. Without any assumptions such as sparsity of the true coefficients (Tibshirani, 1996; Candes and Tao, 2007) or low effective dimension of the features (Zhang, 2005; Hsu et al., 2012), SSL may not perform well even if the exact matching holds. Since the true coefficients are not necessarily sparse as discussed in Section 3, we explore how low-rank structures in the redundancy (i.e., $R(X_1)$) naturally lead to a low effective dimension. In particular, we adopt the definition of effective dimension from (Hsu et al., 2012) in the context of ridge regression (see details below); a closely related notion is called effective degrees of freedom (Efron, 1986; Hastie et al., 2009). Roughly speaking, the effective dimension measures

the number of features that are not linearly correlated; when it is low, a small number of labeled samples can be sufficient for reliable estimation

To simplify the notation, denote $X := \psi^*(X_1)$ with covariance matrix $\Sigma := \mathrm{Cov}(X)$. Let $\{\lambda_j\}_{j=1}^{d_2}$ denote the eigenvalues of Σ. The population ridge estimator is given by

$$\beta_\lambda = \arg\min_\beta \mathsf{E}\left[\|Y - X\beta\|^2\right] + \lambda\|\beta\|^2 = (\Sigma + \lambda\mathbf{I}_{d_2})^{-1}\mathsf{E}[XY^\top].$$

Implicitly dimension reduction is performed in ridge regression when some appropriate shrinkage parameter λ is chosen. The reduced dimension for a chosen λ can be quantified by the *effective dimension*, defined as $d_\lambda = \sum_{j=1}^d \frac{\lambda_j}{\lambda_j+\lambda}$ for $X \in \mathbb{R}^d$. Note that the bias of the ridge estimator increases monotonically as λ increases. When Σ has exactly s nonzero eigenvalues, d_λ is upper bounded by s for any $\lambda \geq 0$. Besides this special case, a low effective dimension can be achieved under a weak penalty (i.e., small λ) when there is a large percentage of small eigenvalues.

In the next subsection, we demonstrate that a low effective dimension is naturally attained when the redundancy $R(X_1)$ can be approximated by a low-rank decomposition. Our finite sample analysis on the high-dimensional setting, presented in Section 5, relies on an upper bound of the low effective dimension, utilizing a measure of the low-rank approximation introduced in the following subsection (Lemma 16). When $d_2 < n$, d_λ with $\lambda = 0$ offers an interpretation of our upper bound on the excess risk and sample complexity (see details in Theorem 9 and Remark 10).

4.1. Low-rank Approximation of Redundancy

Recall, redundancy refers to information in X_1 useful for predicting X_2 but not for the label Y. For instance, in computer vision tasks, consider a label Y determined by the object of interest within a surrounding background. Redundancy arises when the pretext task captures background information unrelated to the label. If the background features *simple* patterns, such as sky, grassland, or beach, this redundancy can be considered low-rank. Consequently, it is relatively easy to eliminate such redundancy in downstream tasks (recall the cancellation of sine functions below Definition 3). In the following, we present the technical details of the low-rank approximation of redundancy.

Denote $\widetilde{C} := \mathsf{E}[C^h(X_1)]$ and recall that $C^h(X_1)$ reduces to $\widetilde{C}$ under conditional mean independence. Assume that the necessary and sufficient condition in Lemma 6 is satisfied,

$$X = C^h(X_1)\,\mathsf{E}[Y|X_1] = \left(\widetilde{C} + A\left[\mathbf{0}\ \ R(X_1) - \widetilde{R}\right]\right)\mathsf{E}[Y|X_1]$$
$$= (\widetilde{C} + A_{p+1:d_2}(R(X_1) - \widetilde{R}))\,\mathsf{E}[Y|X_1],$$

where $\widetilde{R} := \mathsf{E}[R(X_1)]$ and $A_{p+1:d_2} \in \mathbb{R}^{d_2 \times (d_2-p)}$ denotes the last $d_2 - p$ columns of A. If the (centered) redundancy $R(X_1) - \widetilde{R}$ admits a low-rank decomposition, i.e., $R(X_1) - \widetilde{R} = Bg(X_1)$ for some $B \in \mathbb{R}^{(d_2-p)\times s}$ and $g : \mathcal{X}_1 \to \mathbb{R}^{s\times p}$, where $s \ll d_2$, we get

$$X = (\widetilde{C} + A_{p+1:d_2}Bg(X_1))\,\mathsf{E}[Y|X_1], \tag{7}$$

which has at most $p+s$ linearly independent components, as $\mathrm{rank}(\widetilde{C}) \leq p$ and $\mathrm{rank}(A_{p+1:d_2}B) \leq s$. This shows that the effective dimension of X is bounded by $p + s$ for any $\lambda \geq 0$.

Since the low-rank decomposition may not hold exactly for a chosen s, we identify $\hat{X}$ of the form $(\widetilde{C} + Bg(X))\,\mathsf{E}[Y|X_1]$ that best approximates X. Specifically, for any fixed s s.t. $1 \leq s \leq$

$d_2 - p$, we consider $B \in \mathbb{R}^{d_2 \times s}$, $g : \mathcal{X}_1 \to \mathbb{R}^{s \times p}$, and define

$$\varepsilon_s := \min_{\hat{X}} \frac{1}{d_2} \mathsf{E}\left[\left\|X - \hat{X}\right\|^2\right] = \min_{B,g} \frac{1}{d_2} \mathsf{E}\left[\left\|\left(C^h(X_1) - \widetilde{C} - Bg(X_1)\right)\mathsf{E}[Y|X_1]\right\|^2\right], \quad (8)$$

with minimizers $\{(B^*, g^*)\}$ and we use the shorthand $\widetilde{X} := (\widetilde{C} + B^* g^*(X_1))\mathsf{E}[Y|X_1]$ for any pair of optimizer (B^*, g^*). Without loss of generality, we normalize $g(X_1)$ and assume $\mathsf{E}[\|g(X_1)\|] = 1$. The low-rank approximation error is averaged so that ε_s does not grow with d_2. A challenge for analyzing ε_s is that the minimizers do not have closed-form expressions. When (X_1, X_2, Y) follows a Gaussian distribution, we show (8) reduces to a weighted low-rank approximation problem (with no randomness) in Appendix E. However, these weighted problems do not have closed-form expressions in general (Srebro and Jaakkola, 2003; Dutta and Li, 2017). Therefore, we leave the further investigations of ε_s for future work. For $s = 0$, we simply define

$$\varepsilon_0 := \frac{1}{d_2} \mathsf{E}\left[\left\|\left(C^h(X_1) - \widetilde{C}\right)\mathsf{E}[Y|X_1]\right\|^2\right],$$

which measures how approximately conditional mean independence (i.e., $C^h = \widetilde{C}$) holds. There is a tradeoff between the effective dimension of $\widetilde{X}$ (which are no greater than s) and the approximation quality, as the approximation level ε_s is non-increasing as s increases. An important special case of the low-rank approximation is when the encoding functions are smooth.

Example 3 (smooth encoding function) *Consider a binary classification problem with a scalar predictor X_1, i.e., $p = d_1 = 1$, assume that the encoding function $C^h : \mathbb{R} \to \mathbb{R}^{d_2}$ is twice continuously differentiable, then its second-order Taylor expansion at $a \in \mathbb{R}$ is given by*

$$C^h(x) = C^h(a) + \left[\frac{\mathrm{d}C^h}{\mathrm{d}x}\big|_{x=a} \quad \frac{\mathrm{d}^2 C^h}{\mathrm{d}x^2}\big|_{x=a}\right]\left[x - a \quad (x-a)^2\right]^\top + \mathcal{O}((x-a)^3),$$

where we can choose a so that $C^h(a) = \widetilde{C}$. This provides a rank-two approximation for $C^h(x) - \widetilde{C}$ such that $\varepsilon_s = \mathcal{O}((x-a)^3)$, where $s = 2$. This example can be generalized to high-order, multiclass, and multivariate cases, and we provide the details in Appendix D.

To understand the impact of the size of ε_s on how approximately the matching holds (or how small the approximation error is), we derive the following upper bound via a ridge-type estimator. Unlike ridge-type estimators used in practice, the parameter ε_s that restricts the size of the coefficients is determined by the generating mechanism of (X_1, X_2, Y).

Lemma 7 *Consider B^* and g^* corresponding to ε_s, we have*

$$\min_\beta \mathsf{error}_{apx}(\beta) \le 2(p + \mathsf{E}[\|N\|^2]) \min_\beta \left(\|\mathbf{I}_p - \beta\widetilde{C}\|^2 + \|\beta B^*\|^2 + \varepsilon_s \|\beta\|^2\right),$$

where the minimum of the RHS is attained at $\beta_s := (\widetilde{C}^\top \widetilde{C} + (B^*)^\top B^* + \varepsilon_s I)^{-1}\widetilde{C}$. The equality holds with the RHS being zero when $\varepsilon_s = 0$.

5. Finite Sample Analysis

Let $\beta^* \in \arg\min_\beta \text{error}_{\text{apx}}(\beta)$ be a fixed true parameter for the downstream task. Recall that the excess risk is defined as $\mathcal{R}(\hat{\beta}_\lambda) = \mathsf{E}[\|\mathsf{E}[Y|X_1] - \hat{\beta}_\lambda X)\|^2]$. Under conditional mean independence, observe that $X = \mathsf{E}[X_2|X_1] = \widetilde{C}\,\mathsf{E}[Y|X_1]$ is a feature vector with at most p (out of d_2) features that are linearly independent. Since the number of classes p is often much smaller than the dimension of the learned representation d_2, the design matrix $\boldsymbol{X}$ for the downstream task is of low rank. This enables a finite-sample bound on the excessive risk $\widetilde{\mathcal{O}}(\frac{p}{n}\sigma^2)$ with sample complexity $\widetilde{\mathcal{O}}(p)$ (Lee et al., 2021), where the bound is independent of the dimension d_2. In the following, we provide a finite-sample analysis of SSL in the general setting when conditional independence can be violated, based on the low-rank approximation defined in (8).

First, we introduce a few technical assumptions. Let Σ, $\widetilde{\Sigma}$, and $\bar{\Sigma}$ denote the covariance matrix of X, $\widetilde{X}$, and $\bar{X} := \widetilde{X} - X$, respectively.

Assumption 1 *We assume $N := Y - \mathsf{E}[Y|X_1]$ is σ^2-sub-Gaussian, and the whitened feature vectors $\widetilde{\Sigma}^{-1/2}\widetilde{X}$ and $\bar{\Sigma}^{-1/2}\bar{X}$ are ρ^2-sub-Gaussian.* [1]

Assumption 2 *There exists $\tilde{b}, \bar{b} \geq 0$ s.t. the following holds almost surely,*

- $\|\widetilde{\Sigma}^{-1/2}\widetilde{X}(\mathsf{E}[Y|X_1] - \beta^* X])^\top\| \leq \tilde{b}\sqrt{p+s}$;

- $\|\bar{\Sigma}^{-1/2}\bar{X}(\mathsf{E}[Y|X_1] - \beta^* X])^\top\| \leq \bar{b}\sqrt{d_2}$.

Remark 8 *A similar assumption has been made in (Lee et al., 2021, Assumption 3.3), which is motivated by (Hsu et al., 2012, Condition 4).*

Let $\lambda_{\max}(A)$ denote the largest eigenvalue of a symmetric real matrix A such that $A \neq \boldsymbol{0}$, $\lambda_{\min\neq 0}(A)$ denote its smallest nonzero eigenvalue, and $\{\lambda_i(A)\}$ denote the set of all its eigenvalues. When $\widetilde{X}$ is good approximates of X, we expect $\widetilde{\Sigma} - \Sigma$ and $\bar{\Sigma}$ to be close to zero matrices. Therefore, we consider restricting the largest eigenvalues of the two matrices, respectively. A generic bound is provided in (Wolkowicz and Styan, 1980), that is $\lambda_{\max}(A) \leq \frac{\text{tr}(A)}{d} + \sqrt{d-1} \cdot s(A)$, where $s(A) := \frac{\text{tr}(A^2)}{d} - \frac{\text{tr}^2(A)}{d^2}$ is the variance of $\{\lambda_i(A)\}$. The equality holds when the $d-1$ smallest eigenvalues are equal. However, this bound can be quite loose when d is large. Instead, we make the following assumption.

Assumption 3 *For some universal constants $c_1 \geq 0$ and $c_2 \geq 0$,*

- $\lambda_{max}(\widetilde{\Sigma} - \Sigma) \leq c_1 \frac{1}{d_2} \cdot |tr(\widetilde{\Sigma} - \Sigma)|$;

- $\lambda_{max}(\bar{\Sigma}) \leq c_2 \frac{1}{d_2} \cdot tr(\bar{\Sigma})$.

Both inequalities require that the average eigenvalue is comparable to the largest eigenvalue. The assumption can be unrealistic when $\widetilde{\Sigma} - \Sigma$ or $\bar{\Sigma}$ has mostly zero eigenvalues but a few large positive eigenvalues. We explain why such a case will not happen when $s \ll d_2$. Case I: When $\text{rank}(\Sigma) \ll d_2$, there exists $s = \text{rank}(\Sigma)$ such that $\widetilde{X} = X$ and $\bar{X} = \boldsymbol{0}$, in which case the inequalities are satisfied with $c_1 = c_2 = 0$. Case II: In settings when $\text{rank}(\Sigma)$ is comparable to d_2

---

1. When $\widetilde{\Sigma}$ or $\bar{\Sigma}$ is not invertible, the whitened feature vector is defined through the generalized inverse.

(i.e., rank(Σ) is a fraction of d_2), $\widetilde{X}$ satisfies rank$(\widetilde{\Sigma}) \leq p + s \ll d_2$, and thus the rank of $\widetilde{\Sigma} - \Sigma$ should be greater than $d_2 - p - s$, meaning that most of eigenvalues are nonzero. Similarly, $\bar{X} = X - \widetilde{X}$ should have at least $d_2 - p - s$ linearly independent components, i.e., rank$(\bar{\Sigma}) \geq d_2 - p - s$. Given that $\widetilde{X}$ serves as an approximation for X with a lower effective dimension, we make the following technical assumption on the rank.

Assumption 4 $rank(\widetilde{\Sigma}) \leq rank(\Sigma)$ and $rank(\bar{\Sigma}) \leq rank(\Sigma)$.

Since X has rank(Σ) linearly independent components while $\widetilde{X}$ is introduced to approximate rank$(\widetilde{\Sigma})$ independent components out of them, $\bar{X} = X - \widetilde{X}$ is expected to have less independent components than X.

Theorem 9   *Under Assumptions 1— 4, for and any $\delta \in (0, 1)$, if $n \gg \rho^4(d_2 + \log \frac{1}{\delta})$, the excess risk of the downstream task induced by $\hat{\beta}_0$ is upper bounded, with probability at least $1 - \delta$, by*

$$\mathcal{R}(\hat{\beta}_0) \leq \mathsf{error}^*_{apx} + \widetilde{\mathcal{O}}\left((1 + \varepsilon_s)\frac{p + s}{n}\sigma^2 + \varepsilon_s\frac{d_2}{n}\sigma^2\right).$$

Remark 10   *When $\varepsilon_s = 0$, if $n \gg \rho^4(p + s + \log \frac{1}{\delta})$, we have $\mathcal{R}(\hat{\beta}_0) \leq \widetilde{\mathcal{O}}\left(\frac{p+s}{n}\sigma^2\right)$.*

The proof of Theorem 9 follows a similar idea to that of (Lee et al., 2021, Theorem 3.5); a subtle yet important difference is that we consider approximation errors due to the violations of the exact matching while they consider approximation errors due to choices of the function class Ψ. When $\varepsilon_s = 0$, the dominating rate of $\mathcal{R}(\hat{\beta}_0)$ is $\frac{p+s}{n}\sigma^2$, which shows that SSL enjoys a similar sample complexity as shown in (Lee et al., 2021) even when conditional independence is violated. We have demonstrated in Lemma 7 how the approximation error $\mathsf{error}_{apx}(\beta^*)$ depends on the approximation level ε_s. We also provide the bound with respect to $\hat{\beta}_\lambda$, stated below. The proof is largely followed from (Hsu et al., 2012, Theorem 16) and we only outline the main steps in Appendix H.

Corollary 11 (Informal)   *Under suitable assumptions, the excess risk of the downstream task induced by $\hat{\beta}_\lambda$ can be upper bounded by*

$$\mathcal{R}(\hat{\beta}_\lambda) \leq \mathsf{error}^*_{apx} + \mathsf{E}[\|(\beta_\lambda - \beta^*)X\|^2] + \mathcal{O}\left(\frac{p + s}{n}\left(1 + \frac{\varepsilon_s}{\lambda}\right)\tilde{\sigma}^2\right),$$

with high probability, where $\tilde{\sigma}^2 := \mathsf{error}_{apx}(\beta_\lambda) + \mathsf{E}[\|(\beta_\lambda - \beta^*)X\|^2] + \sigma^2$.

For simplicity, the parameters that depend on the choice of λ are omitted. The bound requires $p + s \ll n$ even though $n < d_2$, thus an approximation (8) with lower rank is expected in this more challenging setting. The term $\mathsf{E}[\|(\beta_\lambda - \beta^*)X\|^2]$ relies on the difference between β_λ and β^*, as well as the choice of λ. When $\beta_\lambda = \beta^*$ and $\varepsilon_s = 0$, the dominating rate $\frac{p+s}{n}\sigma^2$ is the same as that in Remark 10. This shows that low-rank structures enable SSL to share a similar excess risk upper bound and sample complexity in low- and high-dimensional settings.

6. Experiments

We propose a synthetic dataset and two computer vision tasks to examine the importance of the full rank condition on $C^h(x)$ and the low-rank approximation quality. Recall that a necessary condition for the exact matching is that $C^h(x)$ has full rank for every $x \in \mathcal{X}_1$. For the synthetic dataset,

we ensure that $C^h(x)$ is of full rank and focus on the low-rank approximation. For image data, since C^h is a latent matrix function, it is not straightforward to test whether $C^h(x)$ has full rank in general. To this end, we design images of simple geometric shapes, that can be seen as abstractions of real images and show how some geometric properties make the rank condition hold or fail. To further understand the importance of the low-rank approximation, we add background patterns to the MNIST dataset and show that certain patterns can lead to poor low-rank approximation. See more details of the experiments in Appendixes I, J, and K. SSL approaches have achieved superior performance on large benchmark datasets, while the function class for downstream tasks is often much larger than linear models (e.g., MLPs), which is beyond the scope of our theoretical analysis. The implementation of our experiments is provided at https://github.com/dukang4655/reconstructive_ssl.

6.1. Synthetic Data

We use a synthetic dataset to verify our theoretical results when $n > d_2$. First, we generate (X_1, X_2, Y) with $d_1 = 10$, $d_2 = 20$, and $p = 2$, where $X_2 = (A + Bg(X_1))Y + N$, where $B \in \mathbb{R}^{d_2 \times s}$. See details of the model parameters in Appendix I. We compare SSL with two supervised learning (SL) procedures in two settings: I. Fix $s = 5$ and vary the labeled sample size $n \in \{100, 200, 400, 800, 1600\}$; II. Fix $n = 300$ and vary the low-rank approximation by considering $s \in \{1, 2, \ldots, 5, 10, 15, 20\}$. We consider two supervised learning procedures. SL_2: Predicting Y by X_1, and SL_2: Predicting Y by (X_1, X_2). We use MLPs for the pretext task and the two supervised learning procedures. In Fig. 1, the performance of SL_1 is roughly invariant with respect to s since it does not use X_2 for the prediction, making it more robust than SL_2. This verifies that predictions using the parents of Y as predictors (which we call causal predictions) are more robust than non-causal predictions under a small sample size. The superior performance of SSL degrades as s increases. When $s < d_2 - p = 18$, we have $\varepsilon_s = 0$ according to the factorization in (7). In Fig. 1, when $s = 20$, a low-rank approximation (8) could lead to a large approximation error $\varepsilon_{\bar{s}}$ for $\bar{s} \leq 18$. As a consequence, the advantage of SSL over SL_1 is much smaller compared with the case with $s = 0$. This indicates that a good low-rank approximation is not only sufficient but also necessary for SSL. In the other
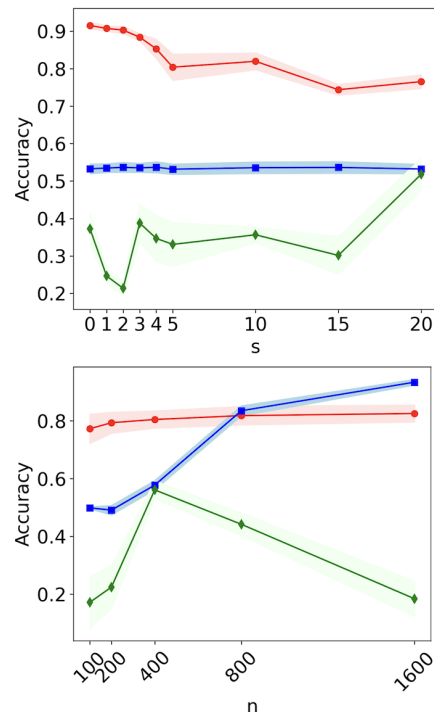

Figure 1: Setting I (top row): $n = 300$ and vary s. SSL (red), SL_1 (blue), and SL_2 (green). 50 repeated experiments. Solid lines: mean; shaded region: standard derivation. Setting II (bottom row): $s = 5$ and n varies.

setting when s is fixed to 5 but the sample size n varies, as shown in Fig.1, the performance of SSL improves slowly with the increasing sample size, since it already achieves performance that to close to the optimal (i.e., the performance of SL_1 with a large n) under a small n. SL_1 starts to catch up with SSL when $n \geq 800$, while the accuracy of SL_2 does not consistently improve as n increases.

6.2. Computer Vision Tasks

6.2.1. GEOMETRIC SHAPES (ON THE RANK CONDITION)

In computer vision applications, it is common that the dimension of the learned representation is much larger than the labeled sample size (i.e., $d_2 \gg n$) for SSL. We design a simple task to help understand how the patterns in an image make SSL work or fail; the task is inspired by (Gidaris et al., 2018), where the pretext task is to predict the rotated angles of images. We consider X_1 as a random image of some objects, where the objects have random sizes and random locations. The goal is to classify the shape of the object $\bar{Y}$. We created X_2 by randomly rotating X_1 by 0 or 90 degrees. Observe that the location and the size of an object are redundant features for predicting its shape and orientation. This stylized setting, even though much simpler than real-world images, is designed this way on purpose in order for the redundancy variable to have low-rank approximations. According to (4), the jth column of $C^h(x)$ can be viewed as a feature vector for the rotation angle corresponding to the jth class of objects. Since we consider the classification of two classes (i.e., $p = 2$), the condition requires that the two feature vectors should not be similar. To verify this necessary condition, we consider two pairs of objects:
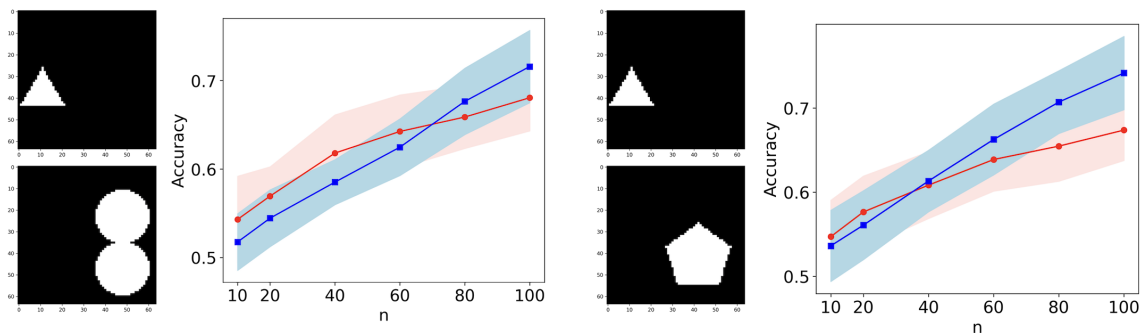

Figure 2: (a) Left: Triangle vs. Tangent Circles. (b) Right: Triangle vs. Pentagon. 50 repeated experiments. SSL (red) and SL (blue).

Triangle vs. Tangent Circles (Fig. 2(a)). In this case, the identification of orientation is based on characteristics specific to the two shapes. For triangles, it is natural to focus on the edges and vertices, while those characteristics are not even defined for circles. Thus, we think the rank condition is approximately satisfied. We examine this observation using Grad-CAM (Selvaraju et al., 2017) that visualizes the contributing features that the model extracted from the image (see Fig. 4 from Appendix J). Similarly for the pair of objects below.

Triangle vs. Pentagon (Fig. 2(b)) In this case, the orientation of the two objects can be identified in similar ways, mainly based on the edges and vertices. In this case, the columns of $C^h(x)$ are close to linearly dependent for different $x \in \mathcal{X}_1$. As a consequence, the necessary condition for the exact matching is violated.

We compare SSL with SL under different labeled sample sizes $n \in \{10, 20, 40, 60, 80, 100\}$. For Triangle vs. Tangent Circles, as shown in Fig. 2, SSL consistently outperforms SL for small sample sizes (i.e., $n \leq 60$). The performance of SSL improves slower compared with SL for sufficiently large n, since the prediction error of SSL will be dominated by the population error instead of the estimation error. Recall that our finite-sample bound on the excess risk in Corollary 11 converges to the sum of the approximation error and an error term depending on λ as $n \to \infty$. In

contrast, SSL behaves very differently for Triangle vs. Pentagon. From Fig. 2(b), the accuracy of SSL improves slower as n increases, potentially due to the large approximation error. Overall, SSL has almost no advantage over SL. This experiment shows that the rank condition is crucial for SSL.

6.2.2. STYLIZED MNIST (ON THE LOW-RANK APPROXIMATION)

To examine how the redundancy affects the performance of SSL, we consider the same rotation prediction task for a stylized MNIST dataset illustrated in Fig. 3, where the density of the background pattern varies randomly. A key observation is that the dot pattern does not help identify the image orientation, so it is not encoded into the orientation variable as redundancy. On the contrary, one can tell the orientation of the image simply by the orientation of the dash pattern, thus the pretext task will extract features from the dash pattern as redundant information. Again, we use Grad-CAM to visualize our observation in Fig. 7 from Appendix K. Consequently, a dense dash pattern can lead to poor low-rank approximation. In Fig. 6 from Appendix K, the performance of SSL is almost invariant to the density of the dot pattern while the performance of SL drops as the density increases. In contrast, SSL is quite sensitive to a sparse dash pattern and the performance gets worse as the density increases (see Fig. 3). We have tested the dot vs. dash patterns for the geometric shape images, and similar results are observed as shown in Fig. 5 from Appendix J.
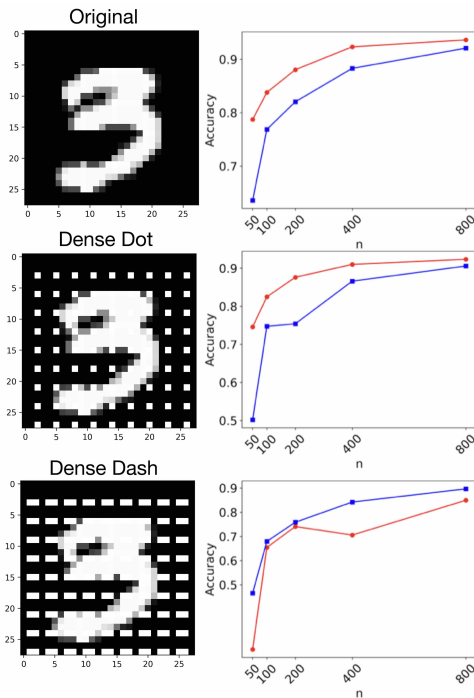

Figure 3: SSL(red) and SL(blue).

7. Discussion

Many important questions remain to be studied and we list a few in this section. One natural next step is to study nonlinear function classes for the downstream task and characterize the corresponding sufficient and necessary conditions. Since our theoretical results can potentially provide guidance for developing SSL procedures, especially for designing pretext tasks, it would be worthwhile to design systematic and extensive experiments to better bridge the theories and practical designs. Besides the superior performance under limited labeled samples, the other major advantage of SSL is that the learned representation can be useful for a diverse class of downstream tasks; a theoretical understanding of its ability to generalize to new tasks or unseen environments (e.g., by exploiting invariance Du and Xiang (2023b)) is of great importance.

References

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. 2007.

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.

Kang Du and Yu Xiang. An invariant matching property for distribution generalization under intervened response. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1387–1391. IEEE, 2022.

Kang Du and Yu Xiang. Generalized invariant matching property via lasso. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023a.

Kang Du and Yu Xiang. Learning invariant representations under general interventions on the response. *IEEE Journal on Selected Areas in Information Theory*, 2023b.

Aritra Dutta and Xin Li. On a problem of weighted low-rank approximation of matrices. *SIAM Journal on Matrix Analysis and Applications*, 38(2):530–553, 2017.

Bradley Efron. How biased is the apparent error rate of a prediction rule? *Journal of the American statistical Association*, 81(394):461–470, 1986.

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

Jie Gui, Tuo Chen, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey of self-supervised learning from multiple perspectives: Algorithms, theory, applications and future trends. *arXiv preprint arXiv:2301.05712*, 2023.

Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

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

Jason D Lee, Qi Lei, Nikunj Saunshi, and Jiacheng Zhuo. Predicting what you already know helps: Provable self-supervised learning. *Advances in Neural Information Processing Systems*, 34:309–323, 2021.

Utku Ozbulak, Hyun Jung Lee, Beril Boga, Esla Timothy Anzaku, Homin Park, Arnout Van Messem, Wesley De Neve, and Joris Vankerschaver. Know your self-supervised learning: A survey on image-based generative and discriminative training. *arXiv preprint arXiv:2305.13689*, 2023.

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.

Judea Pearl. *Causality*. Cambridge University Press, 2009.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

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

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 720–727, 2003.

Jiaye Teng, Weiran Huang, and Haowei He. Can pretext-based self-supervised learning be boosted by downstream data? a theoretical analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 4198–4216. PMLR, 2022.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*, pages 1179–1206. PMLR, 2021.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

Colin Wei, Sang Michael Xie, and Tengyu Ma. Why do pretrained language models help in downstream tasks? an analysis of head and prompt tuning. *Advances in Neural Information Processing Systems*, 34:16158–16170, 2021.

Henry Wolkowicz and George PH Styan. Bounds for eigenvalues using traces. *Linear algebra and its applications*, 29:471–506, 1980.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *Computer Vision– ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 649–666. Springer, 2016.

Tong Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural computation*, 17(9):2077–2098, 2005.

Appendix A. Conditional Mean Independence

Recall that when C^h is a constant function, we have $X_2 = \widetilde{C}Y + N$ and we require $\mathsf{E}[N|X_1, Y] = 0$ rather than $\mathsf{E}[N|Y] = 0$.

Proposition 12 *Model (4) holds with C^h being a constant function if and only if $\mathsf{E}[X_2|X_1, Y] = \mathsf{E}[X_2|Y]$.*

Proof First, assume that $X_2 = \widetilde{C}Y + N$ holds with $\mathsf{E}[N|X_1, Y] = 0$. It follows immediately that

$$\mathsf{E}[X_2|X_1, Y] = \mathsf{E}[\widetilde{C}Y + N|X_1, Y] = \widetilde{C}Y$$

and $\mathsf{E}[X_2|Y] = \mathsf{E}[\widetilde{C}Y + N|Y] = \widetilde{C}Y$, where we use the fact that $\mathsf{E}[N|Y] = \mathsf{E}[\mathsf{E}[N|X_1, Y]|Y] = 0$. Thus we have $\mathsf{E}[X_2|X_1, Y] = \mathsf{E}[X_2|Y]$. Now we prove the other direction. Assume that $\mathsf{E}[X_2|X_1, Y] = \mathsf{E}[X_2|Y]$, then

$$\mathsf{E}[X_2|X_1, Y] = \mathsf{E}[X_2|Y] = \sum_{i=1}^{p} \mathsf{E}[X_2|Y = e_i]\mathbb{1}_{\bar{Y}=y_i} := \widetilde{C}Y,$$

where $\widetilde{C}$ has columns $\mathsf{E}[X_2|Y = e_i]$'s, which implies model (12) in the form of $X_2 = \widetilde{C}Y + N$. ∎

Appendix B. Proof of Proposition 5

Proof First, according to Definition 2, the exacting matching is equivalent to

$$\mathsf{E}[Y|X_1] = \beta\,\mathsf{E}[X_1|X_2] \qquad (9)$$

for some $\beta \in \mathbb{R}^{p \times d_2}$. By our assumption that $\mathrm{Cov}(\mathsf{E}[Y|X_1])$ has full rank, β has to have full rank when the exact matching holds. Plugging $\mathsf{E}[X_2|X_1] = C^h(X_1)\,\mathsf{E}[Y|X_1]$ into (9), we have

$$(\mathbf{I}_p - \beta C^h(X_1))\,\mathsf{E}[Y|X_1] = \mathbf{0},$$

for some $\beta \in \mathbb{R}^{p \times d_2}$. Equivalently,

$$\mathbf{I}_p - \beta C^h(X_1) = \bar{O}(X_1) \qquad (10)$$

for some $\bar{O} : \mathcal{X}_1 \to \mathbb{R}^{p \times p}$ such that $\bar{O}(X_1)\,\mathsf{E}[Y|X_1] = \mathbf{0}$. Since β has full rank, (10) is equivalent to

$$\beta(C^h(X_1) - \beta^{-1}\bar{O}(X_1)) = \mathbf{I}_p, \qquad (11)$$

where β^{-1} denotes the right inverse of β and $O(X_1) := \beta^{-1}\bar{O}(X_1)$ satisfies $O(X_1)\,\mathsf{E}[Y|X_1] = \mathbf{0}$.
∎

Appendix C. Proof of Lemma 6

Proof For simplicity of notation, we prove the lemma without considering the orthogonal term, while the orthogonal term can be directly added to the final expression of C^h. Recall the dimension of β is p by d_2. According to Proposition 5, it is sufficient to show that $\beta C^h(x) = \mathbf{I}_p$ is equivalent to (6). The "if" part is immediate since $\beta = [C^{-1}, \mathbf{0}]A^{-1}$ will lead to $\beta C^h(x) = \mathbf{I}_p, \forall x \in \mathcal{X}_1$. In the following, we prove the other direction. When the exact matching holds, recall that the full rank of $\mathrm{Cov}(\mathsf{E}[Y|X_1])$ implies that β has full row rank since $p < d_2$. The QR decomposition of $\beta^\top$ gives $\beta = [\bar{C}, \mathbf{0}]\bar{A}$, where $\bar{C} \in \mathbb{R}^{p \times p}$ is an invertible lower-triangular matrix and $\bar{A} \in \mathbb{R}^{d_2 \times d_2}$ is an orthonormal matrix. Then, $\beta C^h(x) = \mathbf{I}_p$ implies $[\bar{C}, \mathbf{0}]B(x) = \mathbf{I}_p$ with $B(x) := \bar{A}C^h(x)$. Due to

the zero columns in $[\bar{C}, \mathbf{0}]$, the first p rows of $B(x)$ has to be $\bar{C}^{-1}$, while the other rows, denoted by $R(x)$, can be arbitrary. Therefore, we obtain

$$C^h(x) = \bar{A}^{-1}B(x) = \bar{A}^{-1}\begin{bmatrix} \bar{C}^{-1} \\ R(x) \end{bmatrix} = \widetilde{A}^{-1}\begin{bmatrix} \mathbf{I}_p \\ R(x) \end{bmatrix}, \quad \forall x \in \mathcal{X}_1,$$

where $\widetilde{A}^{-1}$ is the product of $\bar{A}^{-1}$ and some elementary matrices introduced to transform $\bar{C}^{-1}$ to an identity matrix $\mathbf{I}_p$. ∎

Appendix D. Low-rank approximation of Smooth Encoding Functions

In this section, we study how the smoothness of the encoding function enables the low-rank approximation. Specifically, we will construct the approximation in (8) explicitly with polynomial functions. For simplicity, we present the idea for second-order approximation, while the higher-order cases can be derived in a similar manner.

Let $C^h : \mathcal{X}_1 \to \mathbb{R}^{d_2 \times p}$ be a twice continuously differentiable matrix function, the Taylor expansion of its jth column C^h_j at $a \in \mathcal{X}_1$ is given by

$$C^h_j(x) = C^h_j(a) + (C^h_j)'(a)(x - a) + \begin{bmatrix} \frac{1}{2}(x-a)^\top (C^h_{1j})''(a)(x-a) \\ \cdots \\ \frac{1}{2}(x-a)^\top (C^h_{d_2j})''(a)(x-a) \end{bmatrix} + \mathcal{O}(||x - a||^3)$$

$$:= C^h_j(a) + A_j(a)\phi(x) + \mathcal{O}(||x - a||^3), \tag{12}$$

where the ith row of $(C^h_j)'(a) \in \mathbb{R}^{d_2 \times d_1}$ is the derivative of C^h_{ij} evaluated at $x = a$ and $(C^h_{ij})''(a)$ is the Hessian matrix of C^h_{ij} evaluated at $x = a$. We represent $C^h_j(x)$ in a matrix form in (12) by introducing $\phi(x) = (x_1 - a_1, x_2 - a_2, \ldots, x_{d_1} - a_{d_1}, (x_1 - a_1)^2, (x_1 - a_1)(x_2 - a_2), (x_1 - a_1)(x_3 - a_3), \ldots, (x_{d_1-1} - a_{d_1-1})(x_{d_1} - a_{d_1}) + (x_{d_1} - a_{d_1})^2)^\top \in \mathbb{R}^{d_1 + d_1^2}$ and the coefficient matrix $A_j(a) \in \mathbb{R}^{d_2 \times (d_1 + d_1^2)}$ consisting of the (scaled) first two order derivatives. This allows us to approximate C^h by

$$C^h(x) = C^h(a) + \left[A_1(a)\phi(x), A_2(a)\phi(x), \ldots, A_p(a)\phi(x) \right] + \mathcal{O}(||x - a||^3).$$

Let the maximum rank of the matrices $\{A_i(a) : i = 1, \ldots, p\}$ be s, then there exists $B \in \mathbb{R}^{d_2 \times s}$ and $\{D_i(a) \in \mathbb{R}^{s \times (d_1 + d_1^2)} : i = 1, \ldots, p\}$ such that

$$C^h(x) = C^h(a) + B \left[D_1(a)\phi(x), D_2(a)\phi(x), \ldots, D_p(a)\phi(x) \right] + \mathcal{O}(||x - a||^3), \tag{13}$$

which is enabled by the decomposition $A_i(a) = BD_i(a)$ for each i. With the continuity of $C^h(x)$, we can fix a so that $C^h(a) = \tilde{C}$ by the mean value theorem. The high-order reminder term can be ignored when the third derivatives of C^h are all zeros. If this is not the case, we can include high-order terms in ϕ until the reminder term is small enough. If the maximum rank s is not small, one can still consider the low-rank approximation of $\{A_i(a)\}$, and there will be an additional approximation error term in (13).

Appendix E. SSL under Gaussian Distribution

Even though our formulation focuses on the classification setting, the extension of our results to the Gaussian case is straightforward. Assume $\{X_1, X_2, Y\}$ are jointly Gaussian, where Y is a scalar Gaussian variable. Let $\Sigma_{XZ} := \mathrm{Cov}(X, Z)$ for any random vectors X and Z. Then, $X := \mathsf{E}[X_2|X_1] = \Sigma_{X_2 X_1} \Sigma_{X_1 X_1}^{-1} X_1$ and $\mathsf{E}[Y|X_1] = \Sigma_{Y X_1} \Sigma_{X_1 X_1}^{-1} X_1$. If $d_2 \geq d_1$ and $\Sigma_{X_2 X_1}$ has a full rank, it is straightforward to see that an exact matching holds with $\beta = \Sigma_{Y X_1} \Sigma_{X_2 X_1}^{\dagger}$. Concretely, there exists $A \in \mathbb{R}^{d_1 \times d_2}$ and $b \in \mathbb{R}^{d_2}$ such that

$$X_2 = h(X_1, Y) + N = AX_1 + bY + N, \tag{14}$$

where $\mathsf{E}[N|X_1, Y] = \mathbf{0}$.

Since the encoding function is not well-defined when Y is continuous, we formulate the low-rank approximation by

$$\varepsilon_s := \min_{\hat{X}} \mathsf{E}\left[\left\|X - \hat{X}\right\|^2\right] = \min_{B \in \mathbb{R}^{d_2 \times d_1} : \mathrm{rank}(B) = s} \mathsf{E}\left[\left\|\Sigma_{X_2 X_1} \Sigma_{X_1 X_1}^{-1} X_1 - BX_1\right\|^2\right]$$

$$= \min_{B \in \mathbb{R}^{d_2 \times d_1} : \mathrm{rank}(B) = s} \left\|(\Sigma_{X_2 X_1} \Sigma_{X_1 X_1}^{-1} - B)\Sigma_{X_1 X_1}^{1/2}\right\|^2,$$

which aims to find a weighted low-rank approximation for $\Sigma_{X_2 X_1} \Sigma_{X_1 X_1}^{-1}$.

Appendix F. Proof of Lemma 7

Proof We have

$$\min_{\beta} \mathsf{error}_{\mathrm{apx}}(\beta) = \min_{\beta} \mathsf{E}[\|\mathsf{E}[Y|X_1] - \beta C^h(X_1) \mathsf{E}[Y|X_1]\|^2]$$

$$\leq \min_{\beta} \mathsf{E}[\|\mathsf{E}[Y|X_1]\|^2] \mathsf{E}[\|\mathbf{I}_p - \beta(\widetilde{C} + B^* g^*(X_1))\|^2] + \varepsilon_s \|\beta\|^2$$

$$\leq \min_{\beta} 2(p + \mathsf{E}[\|N\|^2]) \left(\|\mathbf{I}_p - \beta\widetilde{C}\|^2 + \|\beta B^*\|^2\right) + \varepsilon_s \|\beta\|^2$$

$$\leq 2(p + \mathsf{E}[\|N\|^2]) \min_{\beta} \left(\|\mathbf{I}_p - \beta\widetilde{C}\|^2 + \|\beta B^*\|^2 + \varepsilon_s \|\beta\|^2\right),$$

where the first inequality follows from the sub-multiplicativity of the matrix norm and the triangle inequality and the last two inequalities are due to the triangle inequality, and the fact that $\mathsf{E}[\|\mathsf{E}[Y|X_1]\|^2] = \mathsf{E}[\|Y - N\|^2] \leq 2\mathsf{E}[\|Y\|^2] + 2\mathsf{E}[\|N\|^2] \leq 2(p + \mathsf{E}[\|N\|^2])$. When $\varepsilon_s = 0$, note that $[\widetilde{C}, B^*]$ has rank at most $p + s \leq d_2$, thus there exists at least one solution $\beta \in \mathbb{R}^{p \times d_2}$ for the equation $\beta[\widetilde{C}, B^*] = [\beta\widetilde{C}, \beta B^*] = [\mathbf{I}_p, \mathbf{0}]$. The expression of β_s is a standard expression of ridge-type estimators. ∎

Appendix G. Proof of Theorem 9

Lemma 13 (Concentration on the covariance matrix (Lee et al., 2021)) *For $\mathbf{X} \in \mathbb{R}^{n \times d}$ with i.i.d. rows, where each row is ρ^2-sub-Gaussian with covariance Σ. For any $B \in \mathbb{R}^{d \times m}$ with rank*

k that is independent of $\boldsymbol{X}$. For any $\delta \in (0,1)$, if $n \gg \rho^4(k + \log(\frac{1}{\delta}))$, with probability at least $1 - \frac{\delta}{10}$, we have

$$0.9B^\top \Sigma B \preceq \frac{1}{n} B^\top \boldsymbol{X}^\top \boldsymbol{X} B \preceq 1.1 B^\top \Sigma B.$$

Lemma 14 ((Lee et al., 2021)) *Let $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ be a projection matrix and let $\boldsymbol{Z} \in \mathbb{R}^{n \times d}$ be a matrix with i.i.d. rows, where each row of $\boldsymbol{Z}$ is mean zero (conditioning on $\boldsymbol{P}$ being rank k) σ^2-sub-Gaussian. For any $\delta \in (0,1)$, with probability at least $1 - \delta$,*

$$\|\boldsymbol{P}\boldsymbol{Z}\| \lesssim \sigma\sqrt{k(1 + \log(k/\delta))}.$$

G.1. Technical Lemmas

Lemma 15 *Under Assumptions 3 and 4, for any s such that $1 \leq s \leq d_2 - p$,*

- *$\widetilde{\Sigma}$ satisfies $\widetilde{\Sigma} \preceq \tilde{a}(1 + \varepsilon_s)\Sigma$ for some $\tilde{a} \geq 0$;*

- *$\bar{\Sigma}$ satisfies $\bar{\Sigma} \preceq \bar{a}\varepsilon_s\Sigma$ for some $\bar{a} \geq 0$.*

Proof To prove $A \preceq aB$ for symmetrical matrices $A, B \in \mathbb{R}^{d \times d}$ and $a \geq 0$, one can simply prove $\lambda_j(A) \leq a\lambda_j(B), \forall j$. This holds immediately for zero eigenvalues $\lambda_j(B)$'s if $\text{rank}(A) \leq \text{rank}(B)$. Therefore, by Assumption 4, we will focus on the case when Σ has all positive eigenvalues. First,

$$\varepsilon_s = \frac{1}{d_2}\mathsf{E}[\|\widetilde{X} - X\|^2] \geq \frac{1}{d_2}\left|\mathsf{E}[\|\widetilde{X}\|^2] - \mathsf{E}[\|X\|^2]\right| = \frac{1}{d_2}\left|\text{tr}(\widetilde{\Sigma} - \Sigma)\right|.$$

In the following, we will use the fact that $\lambda_{\min \neq 0}(A)\mathbf{I}_d \preceq A \preceq \lambda_{\max}(A)\mathbf{I}_d$ for any symmetric matrix $A \in \mathbb{R}^{d \times d}$. Using Assumption 3,

$$\widetilde{\Sigma} - \Sigma \preceq \lambda_{\max}(\widetilde{\Sigma} - \Sigma)\mathbf{I}_{d_2} \leq \frac{c_1}{d_2}|\text{tr}(\widetilde{\Sigma} - \Sigma)|\mathbf{I}_{d_2} \leq c_1\varepsilon_s\mathbf{I}_{d_2},$$

which implies,

$$\lambda_i(\widetilde{\Sigma}) \leq \lambda_i(\Sigma) + c_1\varepsilon_s \leq \lambda_i(\Sigma) + \frac{\lambda_i(\Sigma)}{\lambda_{\min \neq 0}(\Sigma)}c_1\varepsilon_s = \left(1 + \frac{c_1\varepsilon_s}{\lambda_{\min \neq 0}(\Sigma)}\right)\lambda_i(\Sigma) \leq \tilde{a}(1 + \varepsilon_s)\lambda_i(\Sigma),$$

for every i and $\tilde{a} \geq \max(1, \frac{c_1}{\lambda_{\min \neq 0}(\Sigma)})$. This immediately leads to $\widetilde{\Sigma} \preceq \tilde{a}(1 + \varepsilon_s)\Sigma$.

Finally, recall the fact that $\varepsilon_s = \frac{1}{d_2} \cdot \text{tr}(\bar{\Sigma})$. By Assumption 3, we have

$$\bar{\Sigma} \preceq \lambda_{\max}(\bar{\Sigma})\mathbf{I}_{d_2} \preceq \frac{c_2}{d_2}\text{tr}(\bar{\Sigma})\mathbf{I}_{d_2} = c_2\varepsilon_s\mathbf{I}_{d_2} \preceq \frac{c_2}{\lambda_{\min \neq 0}(\Sigma)}\varepsilon_s\Sigma := \bar{a}\varepsilon_s\Sigma.$$

∎

G.2. Proof of Theorem 9

Proof First, recall that $\beta^* \in \arg\min_\beta \text{error}_{\text{apx}}(\beta)$ and the shorthand $X := \mathsf{E}[X_2|X_1]$. By the triangle inequality, we have

$$\mathcal{R}(\hat{\beta}_0) = \mathsf{E}[\|\mathsf{E}[Y|X_1] - \hat{\beta}_0 X)\|^2] \leq \text{error}_{\text{apx}}^* + \mathsf{E}[\|(\beta^* - \hat{\beta}_0)X\|^2].$$

Denote $a(X_1) := \mathsf{E}[Y|X_1] - \beta^* X$.

Recall that $X = \widetilde{X} + \bar{X}$. Then, we can write $Y = \beta^* \widetilde{X} + \beta^* \bar{X} + a(X_1) + N$, with N satisfying $\mathsf{E}[N|X_1] = 0$ by the tower property. The definition of $\hat{\beta}_0$ implies

$$\|\boldsymbol{Y} - \boldsymbol{X}\hat{\beta}_0^\top\|^2 \leq \|\boldsymbol{Y} - \boldsymbol{X}(\beta^*)^\top\|^2 = \|a(\boldsymbol{X}_1) + \boldsymbol{N}\|^2.$$

By rearranging the terms, we get

$$\|\boldsymbol{X}(\beta^* - \hat{\beta}_0)^\top)\|^2 \leq \langle a(\boldsymbol{X}_1), \tilde{\boldsymbol{X}}(\beta^* - \hat{\beta}_0)^\top\rangle - \langle \boldsymbol{N}, \tilde{\boldsymbol{X}}(\beta^* - \hat{\beta}_0)^\top\rangle$$
$$+ \langle a(\boldsymbol{X}_1), \bar{\boldsymbol{X}}(\beta^* - \hat{\beta}_0)^\top\rangle - \langle \boldsymbol{N}, \bar{\boldsymbol{X}}(\beta^* - \hat{\beta}_0)^\top\rangle.$$

We bound the first two inner products in the following, and the other two follow similarly. First,

$$\langle a(\boldsymbol{X}_1), \tilde{\boldsymbol{X}}(\beta^* - \hat{\beta}_0)^\top\rangle = \langle \widetilde{\Sigma}^{-1/2}\tilde{\boldsymbol{X}}^\top a(\boldsymbol{X}_1), \widetilde{\Sigma}^{1/2}(\beta^* - \hat{\beta}_0)^\top\rangle$$
$$\leq \|\widetilde{\Sigma}^{-1/2}\tilde{\boldsymbol{X}}^\top a(\boldsymbol{X}_1)\|\|\widetilde{\Sigma}^{1/2}(\beta^* - \hat{\beta}_0)^\top\|$$
$$\leq 1.1\tilde{a}\tilde{b}\sqrt{1 + \varepsilon_s}\sqrt{n(p + s)}\|\Sigma^{1/2}(\beta^* - \hat{\beta}_0)^\top\|$$
$$\lesssim \sqrt{1 + \varepsilon_s}\sqrt{p + s}\|\boldsymbol{X}(\beta^* - \hat{\beta}_0)^\top\|, \tag{15}$$

where the inequality $\|\widetilde{\Sigma}^{-1/2}\tilde{\boldsymbol{X}}^\top a(\boldsymbol{X}_1)\| \leq 1.1\tilde{b}\sqrt{n(p + s)}$ is due to Assumption 2 and the covariance concentration in Lemma 13, and the last inequality is simply due to the covariance concentration. The replacement of $\widetilde{\Sigma}$ by Σ is by Lemma 15. Let $\boldsymbol{P}_{\tilde{\boldsymbol{X}}}$ denote the projection matrix defined with respect to $\tilde{\boldsymbol{X}}$, we have

$$\langle \boldsymbol{N}, \tilde{\boldsymbol{X}}(\beta^* - \hat{\beta}_0)^\top\rangle = \langle \boldsymbol{P}_{\tilde{\boldsymbol{X}}}\boldsymbol{N}, \tilde{\boldsymbol{X}}(\beta^* - \hat{\beta}_0)^\top\rangle$$
$$\leq \|\boldsymbol{P}_{\tilde{\boldsymbol{X}}}\boldsymbol{N}\|\|\tilde{\boldsymbol{X}}(\beta^* - \hat{\beta}_0)^\top\|.$$
$$\lesssim \sigma\sqrt{1 + \varepsilon_s}\sqrt{(p + s)\left(1 + \log\frac{p + s}{\delta}\right)}\|\boldsymbol{X}(\beta^* - \hat{\beta}_0)^\top\|, \tag{16}$$

where the last bound is due to Lemma 14 and the replacement of $\tilde{\boldsymbol{X}}$ by $\boldsymbol{X}$ follows from the covariance concentration and Lemma 15. Since we make no assumptions on the rank of $\bar{\Sigma}$, it has at most rank d_2. Similarly, we get

$$\langle a(\boldsymbol{X}_1), \bar{\boldsymbol{X}}(\beta^* - \hat{\beta}_0)^\top\rangle \lesssim \sqrt{\varepsilon_s}\sqrt{d_2}\|\boldsymbol{X}(\beta^* - \hat{\beta}_0)^\top\| \tag{17}$$

$$\langle \boldsymbol{N}, \bar{\boldsymbol{X}}(\beta^* - \hat{\beta}_0)^\top\rangle \lesssim \sigma\sqrt{\varepsilon_s}\sqrt{d_2\left(1 + \log\frac{d_2}{\delta}\right)}\|\boldsymbol{X}(\beta^* - \hat{\beta}_0)^\top\|. \tag{18}$$

Combining (15), (16), (17), and (18) yields

$$\|\boldsymbol{X}(\beta^* - \hat{\beta}_0)^\top\| \lesssim \sigma\sqrt{1 + \varepsilon_s}\sqrt{(p + s)\left(1 + \log\frac{p + s}{\delta}\right)} + \sigma\sqrt{\varepsilon_s}\sqrt{d_2\left(1 + \log\frac{d_2}{\delta}\right)}.$$

Finally, the covariance concentration implies

$$\mathsf{E}[\|(\beta^* - \hat{\beta}_0)X\|^2] \lesssim (1 + \varepsilon_s)\frac{(p + s)\left(1 + \log \frac{p+s}{\delta}\right)}{n}\sigma^2 + \varepsilon_s \frac{d_2 \left(1 + \log \frac{d_2}{\delta}\right)}{n}\sigma^2.$$

■

Appendix H. Proof of Corollary 11

Lemma 16 *Under Assumptions 3 and 4, for $\lambda > 0$, there exists c_2 such that*

$$d_\lambda \leq c_2 \left(1 + \frac{\varepsilon_s}{\lambda}\right)(p + s).$$

Remark 17 *It is known that $d_\lambda = \mathsf{E}[\|(\Sigma + \lambda\mathbf{I})^{-1/2}X\|^2]$ (Hsu et al., 2012). When $\varepsilon_s = 0$, we have $\mathsf{E}[\|(\Sigma + \lambda\mathbf{I})^{-1/2}X\|^2] \leq rank(\Sigma) = p + s$, where the equality holds when $\lambda = 0$.*

Proof Let $\{\tilde{\lambda}_i\}$ and $\{\lambda_i\}$ denote the eigenvalues of Σ and $\widetilde{\Sigma}$, respectively. Recall that Assumptions 3 and 4 imply $\tilde{\lambda}_i - \lambda_i \leq c_1\varepsilon_s$ as shown in the proof of Lemma 15, then we have

$$\begin{aligned}
d_\lambda = \sum_{i=1}^{d} \frac{\lambda_i}{\lambda + \lambda_i} &= \sum_{i=1}^{d} \frac{\tilde{\lambda}_i}{\lambda + \tilde{\lambda}_i} + \lambda \sum_{i=1}^{d} \frac{\lambda_i - \tilde{\lambda}_i}{(\lambda + \lambda_i)(\lambda + \tilde{\lambda}_i)} \\
&\leq \sum_{i=1}^{d} \frac{\tilde{\lambda}_i}{\lambda + \tilde{\lambda}_i} + c_1\varepsilon_s \sum_{i=1}^{d} \frac{1}{(\lambda + \lambda_i)(\lambda + \tilde{\lambda}_i)} \\
&= \sum_{i=1}^{d} \frac{\tilde{\lambda}_i}{\lambda + \tilde{\lambda}_i} + \frac{c_1\varepsilon_s}{\lambda \cdot \lambda_{\min\neq 0}(\widetilde{\Sigma})} \sum_{i=1}^{d} \frac{\lambda \cdot \lambda_{\min\neq 0}(\widetilde{\Sigma})}{(\lambda + \lambda_i)(\lambda + \tilde{\lambda}_i)},
\end{aligned}$$

where $\frac{\lambda}{\lambda + \lambda_i} \leq 1$ for $\forall i$ and $\sum_{i=1}^{d} \frac{\tilde{\lambda}_i}{\lambda + \tilde{\lambda}_i} \leq p + s$ since $\text{rank}(\widetilde{\Sigma}) \leq p + s$. Finally, we get

$$d_\lambda \leq p + s + \frac{c_1\varepsilon_s(p + s)}{\lambda \cdot \lambda_{\min\neq 0}(\widetilde{\Sigma})} \leq c_2 \left(1 + \frac{\varepsilon_s}{\lambda}\right)(p + s),$$

where $c_2 \geq \max\left\{1, \frac{c_1}{\lambda_{\min\neq 0}(\widetilde{\Sigma})}\right\}$. ■

Corollary 11 *Under (Hsu et al., 2012, Condition 2 and 4), and the assumptions in Lemma 16, the excess risk of the downstream task can be upper bounded by*

$$\mathcal{R}(\hat{\beta}_\lambda) \leq \text{error}^*_{apx} + \mathsf{E}[\|(\beta_\lambda - \beta^*)X\|^2] + \mathcal{O}\left(\frac{p + s}{n}\left(1 + \frac{\varepsilon_s}{\lambda}\right)\tilde{\sigma}^2\right),$$

with high probability, where $\tilde{\sigma}^2 := \left(\mathsf{E}[|\mathsf{E}[Y|X] - \beta_\lambda X\|^2] + \mathsf{E}[\|(\beta_\lambda - \beta^*)X\|^2] + \sigma^2\right)$.

We only outline the main steps and refer the readers to (Hsu et al., 2012, Theorem 16) for details. First, by the triangular inequality, we have $\mathcal{R}(\hat{\beta}_\lambda) \leq \text{error}^*_{apx} + \mathsf{E}[\|(\hat{\beta}_\lambda - \beta^*)X\|^2]$. The bound on the second term can be obtained as discussed below. The last term is simply due to

$d_{2,\lambda} \le d_{1,\lambda} \le c_\lambda(1+\frac{\varepsilon_s}{\lambda})(p+s)$ by Lemma 16, where $d_{l,\lambda} = \sum_{j=1}^{d} \left(\frac{\lambda_j}{\lambda_j+\lambda}\right)^l$ for $l \in \{1,2\}$. Observe that the ridge estimator with $Y_j = \mathbb{1}_{\bar{Y}=y_j}$ as the target variable is equivalent to the jth row of the ridge estimator with Y as the target variable. Thus we can provide a bound on $\mathsf{E}[\|(\hat{\beta}_{\lambda,j} - \beta_j^*)X\|^2]$ for each $j \in \{1,\ldots,p\}$ according to (Hsu et al., 2012, Theorem 16). Summing up the inequalities gives $\mathsf{E}[\|(\hat{\beta}_\lambda - \beta^*)X\|^2] \le \mathsf{E}[\|(\beta_\lambda - \beta^*)X\|^2] + \mathsf{E}[\|(\hat{\beta}_\lambda - \beta_\lambda)X\|^2]$, where $\mathsf{E}[\|(\hat{\beta}_\lambda - \beta_\lambda)X\|^2]$ is upper bounded in (Hsu et al., 2012, Theorem 16).

Appendix I. Synthetic Data: Details of the Data Generation

Let $X_1 \sim \mathcal{N}(\mathbf{0}, I_{d_1})$. Note that $\mathsf{E}[\|X_1\|] = \sqrt{2}\frac{\Gamma(5.5)}{\Gamma(5)} \approx 3.08$, where $\Gamma(\cdot)$ is the Gamma function. The label $\bar{Y}$ is determined by $\|X_1\|$ as follows: $\bar{Y} = 0$ when $\|X_1\| < 2.5$; $\bar{Y} = 1$ when $2.5 \le \|X_1\| < 3.5$; $\bar{Y} = 2$ when $\|X_1\| \ge 3.5$. Then, let $X_2 = (A + Bg(X_1))Y + N$ where A and B are matrices with i.i.d. entries from $\mathrm{Unif}[-2,2]$ and $N \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d_2})$. The $(i,j)^{th}$ element of $g(X_1)$ is given by $(\max_k(X_{1,k})) \cdot \sin\left(\frac{2\pi i}{s}\min_k(X_{1,k}) + \frac{2\pi j}{p}\right)$. The sample sizes of the pretext training data and testing data are 10^4 and 10^3, respectively. The MLPs used for the pretext task and two SL procedures all have two fully connected hidden layers with ReLU activation. The batch size is 32, the number of epochs is 10, and the learning rate is 0.001.

Appendix J. Further Details of the Computer Vision Task

The sample sizes of the pretext task and testing are fixed to be 20000 and 1000, respectively. The edge of the triangle is sampled from $\mathrm{Unif}[8,32]$, the radius of the circles is sampled from $\mathrm{Unif}[5,10]$, and the pentagon is drawn within a circle with radius samples from $\mathrm{Unif}[8,64/3]$. The sizes are chosen to ensure that the average areas of the objects are similar. The pretext task and SL both use convolution neural networks (CNNs) consisting of two convolution layers and two fully connected layers with ReLU activation. The learned representation is obtained from the second convolution layer of the CNN, which has a dimension $d_2 = 12544$. Since the rotated image X_2 has the same label as X_1, we also use the rotated images as additional labeled data for the downstream task and SL. In the pretext task, the batch size is 64, the number of epochs for training is 15, and the learning rate is 0.001. For SL, the batch size the modified to be 32 since the sample size is much smaller. For ridge regression, we choose the shrinkage parameter λ from 200 numbers evenly spaced on a log scale over $[0.001, 100]$ with 5-fold cross-validation.

In Section 6.2, we explain that the unsatisfactory performance of SSL for Triangle vs. Pentagon is potentially due to the analogous characteristics (i.e., the edges and vertices) that determine the orientation of the object. From a theoretical perspective, the full rank condition on $C^h(x)$ (that is a necessary condition for the exact matching) is violated since the columns of $C^h(x)$ are approximately linearly dependent. This does not happen for Triangle vs. Tangent Circles since the orientation of the circles is determined by curved edges. To further support our observation, we use Grad-CAM (Selvaraju et al., 2017) to visualize the contributing features that the pretext model used for rotation prediction. To highlight the major components, we only show pixels of the heatmap with top-20% intensity in Fig. 4.
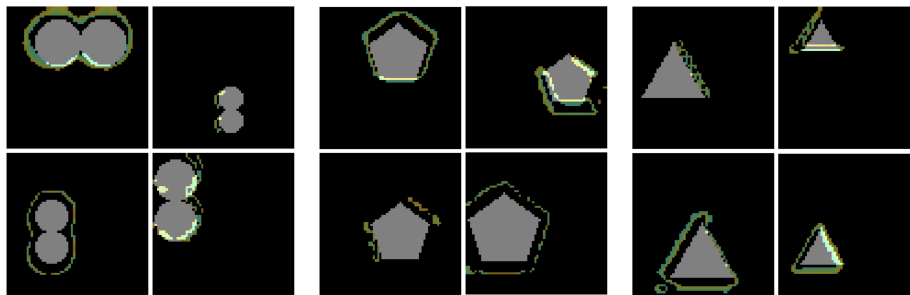
Figure 4: Visualization of the contributing features for rotation prediction in the computer vision task.

Similarly to the stylized MNIST dataset, we add dot and dash patterns to the image background. The space in patterns is sampled from $\mathrm{Unif}[8, 32]$.
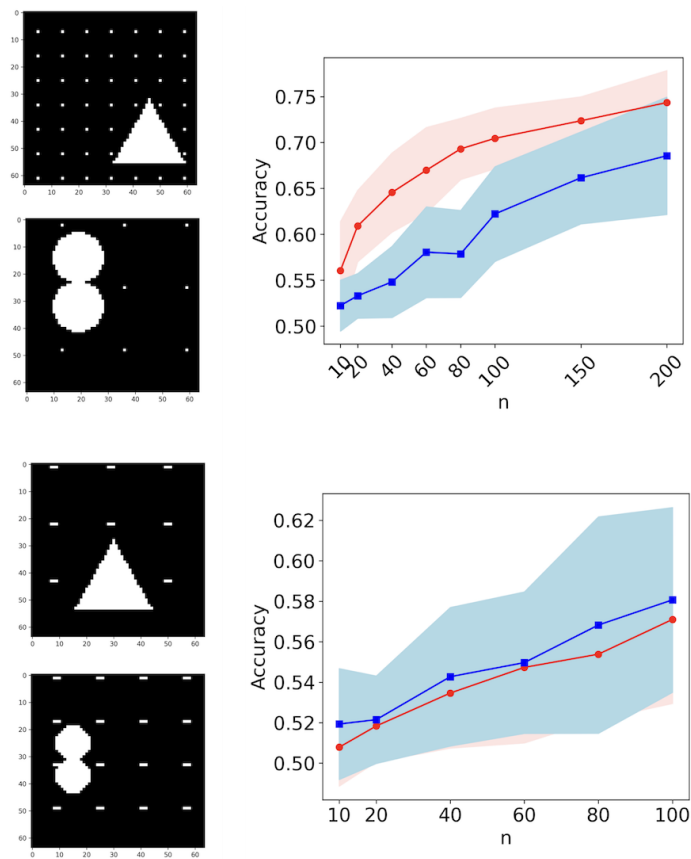

Figure 5: Geometric shape images with dot vs. dash background. SSL (red) and SL (blue)

Appendix K. Experiments on the MNIST Dataset

The sample sizes for the pretext task and testing is 45000 and 10000, respectively. We compare the performance of SSL and SL under different labeled sample sizes $\{50, 100, 200, 400\}$. The space d in the sparse and dense patterns is sampled from $\mathrm{Unif}[3, 8]$ and $\mathrm{Unif}[8, 15]$, respectively. We randomly shift each pattern by $\mathrm{Unif}[0, d]$ to avoid the position of the pattern being correlated with the image orientation. We use the same CNN configuration as the geometric shape task.
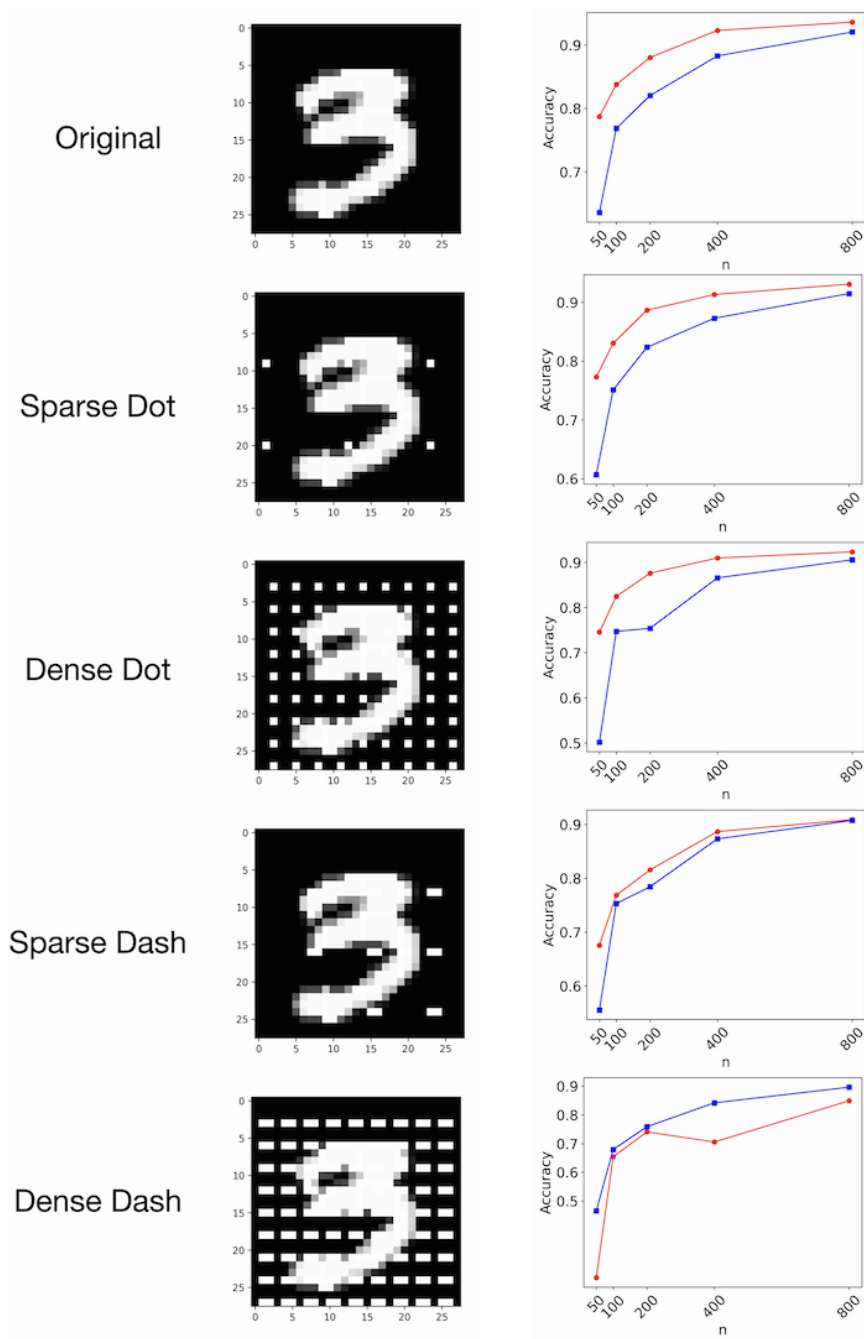
Figure 6: The original MNIST dataset and four stylized versions of MNIST. SSL (red) and SL (blue).
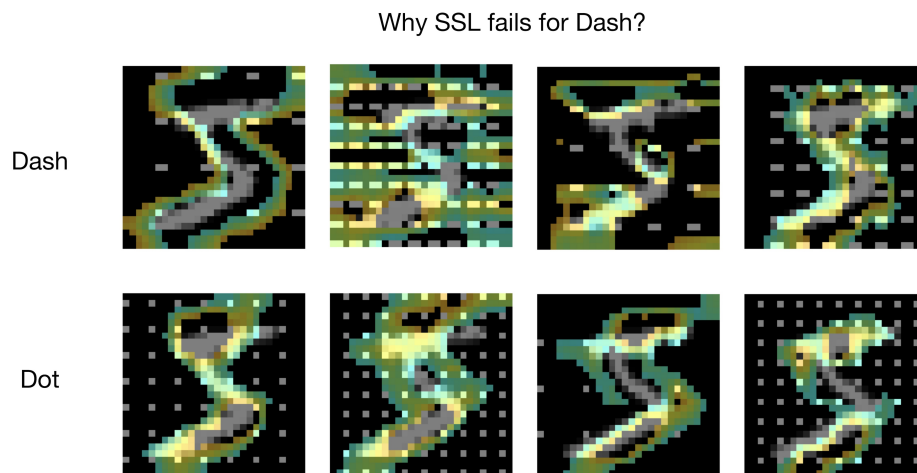

Figure 7: A visualization of the contributing features for MNIST dataset with dash vs. dot background.