An Empirical Study of The Power of Contrast in Feature Learning

Chew Kin Whye [*] **Kshitij Singh** [*] **Prakhar Ganesh** [*]

Abstract

Feature learning, or representation learning, aims to learn a compact representation of the input signal by extracting underlying latent patterns, usually transferable across various downstream tasks. Feature learning has pushed the state-of-the-art in various learning domains, and at the forefront of this progress is contrastive learning. The tremendous success of contrastive learning has motivated researchers to explore the theoretical foundation and power of contrast in feature learning. In this work, we continue the discussion started by Ji et al. [2021] and attempt to provide an empirical companion to the original paper. We start by investigating various assumptions made by the authors. We then provide experimental results and insights on the tightness of proposed theoretical bounds. Finally, we generalize their work to stochastic augmentations, varying noise generation and deep non-linear settings to observe changes in empirical trends.

1 Introduction

Self-supervised learning methods are the backbone of large pre-trained models, providing a framework to utilize vast amount of unlabeled data. Self-supervised representation learning exploits supervision signals present in dataset structure to learn intermediate representations that can carry semantic and structural information beneficial to various downstream tasks [Weng, 2019]. Commonly used self-supervised representation learning methods include principal component analysis [Abdi and Williams, 2010], generative adversarial networks [Goodfellow et al., 2014], auto encoders [Rumelhart et al., 1988], and contrastive learning [Jaiswal et al., 2020].

Contrastive learning has seen a recent resurgence in representation learning [Chen et al., 2020], igniting renewed interest in the theoretical foundations underneath the success of contrast-based feature learning [Arora et al., 2019, HaoChen et al., 2021, Bao et al., 2021, Ji et al., 2021]. We focus on the work by Ji et al. [2021], which provides theoretical bounds on contrastive learning and auto-encoders in linear representation setting. We examine the practical implications of their assumptions, test the tightness of provided theoretical bounds, and generalize the discussion to stochastic augmentations, varying noise generation and non-linear settings. We conclude our work with a brief comment on existing caveats in their work and future directions of research.

1.1 Background

We first introduce the experiment setting proposed by Ji et al. [2021] for their analysis. Next, we highlight the theoretical bounds derived by the authors. Finally, we discuss the lack of open source code or hyperparameter list by the authors and our attempt to replicate their results.

Data Generation Input data x is assumed to be a linear representation of a low-dimensional latent feature z and is formally generated using a *spiked covariance model* as follows,

[*] Equal contribution

$$x = U^* z + \xi, \text{where } z \in \mathbb{R}^r \text{and } \xi \in \mathbb{R}^d \tag{1}$$

The input vector x is created by spanning the latent representation z by the columns of U^* and adding a dense noise term ξ, where $U^* \in \mathbb{O}_{d,r}$ is an orthonormal matrix in dimensions $d \times r$. Latent representation z and noise ξ are both generated as zero-mean sub-Gaussian independent random variables with covariance matrix $\text{Cov}(z) = v^2 I_r$ and $\text{Cov}(\xi) = \text{diag}(\sigma_1^2, ..., \sigma_d^2)$, respectively. The noise generation follows regular covariance condition, i.e. it satisfies $\sigma_{(1)}^2/\sigma_{(d)}^2 < C$, where $\sigma_{(j)}^2$ represents the jth largest variance among $\sigma_1^2, .., \sigma_d^2$ and $C > 0$ is a universal constant. The ratio between latent feature generation and noise generation variance is bounded as $\rho = \frac{v}{\sigma_{(1)}} = \Theta(1)$.

While representation learning is performed in a self-supervised setting, the authors also generate downstream labels for regression using original latent feature z as follows,

$$\hat{y} = \langle z, w^* \rangle / v + \epsilon, \tag{2}$$

where w^* is a fixed unit vector and ϵ is gaussian noise independent of z. The authors also propose label generation for classification, which we found were improper (see discussion in Appendix A.3).

Performance Metric A projection matrix $W \in \mathbb{R}^{r \times d}$ is learned from generated observations x to extract the original information in latent features z, which is eventually used to predict downstream labels. To compare the quality of projection W, the authors rely on sine distance defined as,

$$||\text{sin}\Theta(U, U^*)||_F = \sqrt{||U_\perp^{*T} U||_F^2} = \sqrt{\frac{1}{2}||UU^T - U^* U^{*T}||_F^2}, \text{where } svd(W) = (U \Sigma V^T)^T \tag{3}$$

To compare the downstream performance, the authors train a linear regression model on top of extracted latent representations from the input observations. They report downstream risk as the difference between downstream performance for representations extracted using U^{*T} and W.

Theoretical Bounds Ji et al. [2021] derived a lower bound for auto encoders (AE) and an upper bound for contrastive learning (CL) for the expected value of sine distance, presented as follows,

$$\mathbb{E}||\text{sin}\Theta(U^*, U_{AE})||_F \geq c\sqrt{r} \tag{4}$$

$$\mathbb{E}||\text{sin}\Theta(U^*, U_{CL})||_F \lesssim \frac{r^{3/2} \log d}{d} + \sqrt{\frac{dr}{n}} \tag{5}$$

where U_{CL} and U_{AE} are derived from singular value decomposition of learned projection matrix $W_{CL} = (U_{CL} \Sigma_{CL} V_{CL}^T)^T$ and $W_{AE} = (U_{AE} \Sigma_{AE} V_{AE}^T)^T$ respectively, and $c \in (0, 1)$ is a universal constant. The authors also show that similar bounds are applicable to downstream risk.

Reproducibility Research community in machine learning is going through a reproducibility crisis, with top tier conferences adopting mandatory checklists and competitions to motivate authors and promote open exchange of resources [Pineau et al., 2019]. While the work done by Ji et al. [2021] is inspiring and exceptional, the lack of open source code or even an appropriate list of hyperparameter settings used in their experiments hinders further exploration. We succeed in reproducing the reported trends in Appendix A.2. We make our code publicly available [2] and also provide a detailed response to the NeurIPS reproducibility checklist in Appendix A.1.

2 Experiments

In this section we discuss the motivation and provide detailed results of various experiments that we have conducted. We start by introducing the evaluation setup, based on data generation and metrics

[2] `github.com/prakharg24/contrastive-learning`

proposed by Ji et al. [2021]. We then question the practical implications of various assumptions made by the authors and investigate the tightness of the proposed theoretical bounds. Finally, we generalize the discussion to stochastic augmentations, noise generation, and non-linear settings.

Evaluation Setup We conduct all our experiments with a contrastive learning model (CL) and an auto-encoder (AE). Contrastive learning can be defined as an encoder learning to create similar representations of a data point passed through multiple augmentations, while pushing representations of other data points away. Auto-encoders can be defined as an encoder-decoder pair, learning to replicate the original input. We train contrastive learning with NT-XENT loss [Sohn, 2016] and auto-encoders with MSE loss. We use regularization to promote orthogonality [Ji et al., 2021].

We use the *spiked covariance model* for data generation and *sine distance, downstream risk* for evaluation to replicate the setting proposed by Ji et al. [2021]. Unless specified otherwise, we use latent feature size $r = 10$; generated observation size $d = 40$; training data size $n = 20,000$; latent feature variance $v = 1$; signal to noise ratio $\rho = 1$; regular covariance bound for noise generation $C = 2$; and standard deviation of downstream label generation noise $std(\epsilon) = 0.1$. We use an effective batch size of 64 (formed by applying two augmentations g_1, g_2 on an input batch of size 32), and random mask augmentation with Bernoulli mean 0.5. Refer to Appendix A.1 for details. All experiments are repeated for 10 random seeds and both the average as well as the range of the results are reported. The original numerical experiments are replicated in Appendix A.2.

2.1 Practical Constraints of Underlying Assumptions

We first discuss the underlying assumptions in Ji et al. [2021] which might not transfer in a real-world setting, and use empirical results to study if the trends observed follow the proposed behavior.

Assumption 1 *Training data size (n) needs to be larger than input signal dimension (d).*

The authors assume $n > d \gg r$ throughout the paper. Here we question the applicability of this assumption, especially in low-data settings (eg., few-shot learning) or large input and latent dimension domains (eg., computer vision). To create a realistic setup, we increase the feature and observation dimensions to $r = 40, d = 4000$; and vary the training data size n as shown in Figure 1. We only repeat the experiment once due to the computational burden for large r, d.

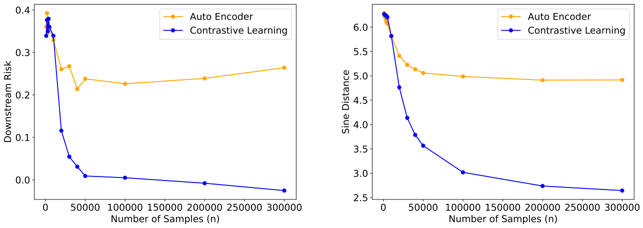


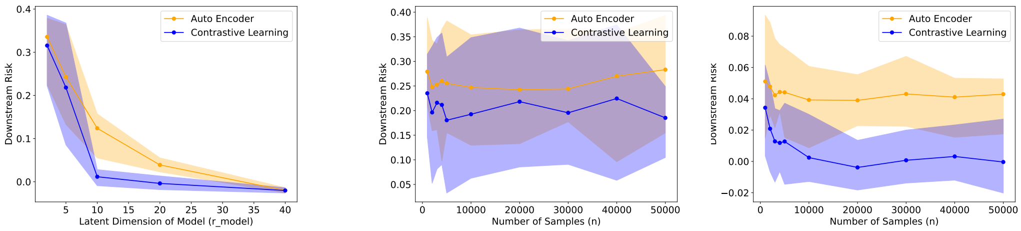
Figure 1: Impact of training size n for large values of $r = 40; d = 4000$

For $n < d$, both models perform significantly worse and the improvement beyond $n > d$ is quite steep, highlighting the limitations of such analysis in low-data or large input dimension settings. However, the performance for contrastive learning keeps improving well beyond $n > d$ and stabilizes around $n \approx dr$. This relates to the term $\sqrt{dr/n}$ in theoretical bound proposed by the authors (see eq 5), and shows that we need $n = O(dr)$ (not just $n > d$) for reasonable performance.

Assumption 2 *Latent dimension of the feature (r) is known during model selection.*

The theoretical analysis and experiments performed by the authors assume the learning model is created with knowledge of true latent feature dimension r. In reality however, we do not have access to such information and instead rely on domain knowledge to estimate r. Thus, instead of using the same value of r, we experiment with using r_{latent} to generate data and r_{model} for the learning model.

We first experiment with varying values of r_{model} while using default values of $r_{latent} (= 10)$, d and n to understand the impact of under- or over-estimating the latent feature size. The results are collected in Figure 2a [3]. As expected, contrastive learning performance stabilizes when $r_{model} = r_{latent} = 10$. While there is a slight improvement in performance for $r_{model} > r_{latent}$, it is not noticeable compared to the steep change for $r_{model} < r_{latent}$. This points to the importance of over-estimating latent feature dimension in real-world scenarios. Surprisingly, additional trends are seen in auto-encoders, whose performance improves significantly for even large values of r_{model}. As we discuss later, this could be attributed to auto-encoders learning noise more aggressively than contrastive learning.



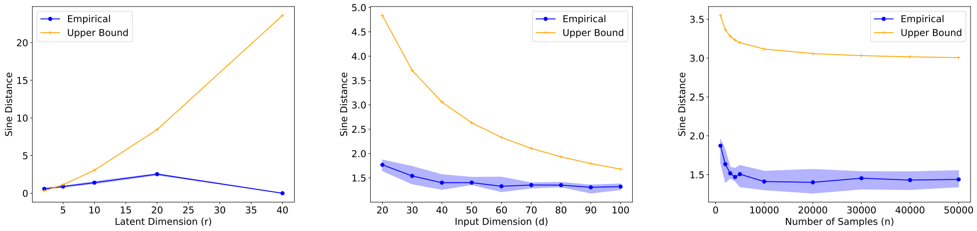
(a) $r_{model} = \{2, 5, 10, 20, 40\}$ (b) $r_{model} = 5$ (c) $r_{model} = 20$

Figure 2: Investigating latent dimension of the learning model r_{model}

To further inspect the model's response to changing values of r_{model}, we choose two fixed settings $r_{model} = 5 < r_{latent}$ and $r_{model} = 20 > r_{latent}$, while varying training size n (and observation dimension d in Appendix A.4) as shown in Figure 2b,2c. For $r_{model} = 20$, as expected, contrastive learning outperforms auto-encoders and both models perform worse for smaller n while improving asymptotically. However, for $r_{model} = 5$, both models have significantly higher downstream risk and are indistinguishable from one another with no significant impact of training size. These results emphasize the negative impact of under-estimating r despite the availability of huge training datasets.

2.2 Tightness of Performance Bounds

The upper bound proposed by Ji et al. [2021] for the performance of contrastive learning (see eq 5) showcases a reliance on the input dimension d, latent dimension r and training size n. We investigate the tightness of the theoretical bound and empirical trends for each variable in Figure 3. We found that while the bounds hold in practice (see Appendix A.5 for exception), there are certain limits to their tightness specially when the assumption $d \gg r$ is challenged. Increasing the value of r (Figure 3a) or decreasing the value of d (Figure 3b) both reduce the ratio d/r and thus in effect create weaker performance bounds, highlighting the limitations of the analysis by Ji et al. [2021].



(a) (b) (c)

Figure 3: Empirical investigation of contrastive learning performance and theoretical upper bound

Interestingly, the theoretical bound on training size n (Figure 3c) never approaches the empirical results asymptotically. This can be attributed to the following lemma [Zhang et al., 2022],

Lemma 2.1 ***(Lemma B.8 in Ji et al. [2021]; Lemma 4 in Zhang et al. [2022])*** *If $M \in \mathbb{R}^{p \times p}$ is any square matrix and $\Delta(M)$ is the matrix M with diagonal entries set to* 0*, then*

$$||\Delta(M)||_2 \leq 2||M||_2 \tag{6}$$

Zhang et al. [2022] points out that the factor of "2" in the above lemma cannot be improved. Thus, it can be seen that bounds calculated by Ji et al. [2021] using this lemma will also suffer from same limits, which explains distance between theoretical bound and empirical performance in Figure 3c.

[3] Note that measuring sine distance is not appropriate when $r_{model} \neq r_{latent}$.

2.3 Generalization Beyond Ji et al. [2021]

We will now generalize our work into settings beyond the environment proposed by Ji et al. [2021]. To keep the discussion simple, we focus mostly on contrastive learning for the rest of this work.

Stochastic random masking augmentation Ji et al. [2021] adopt random masking augmentation in their work, that can be defined as $g_1(x_i) = Ax_i$, $g_2(x_i) = (I - A)x_i$; where A is a randomly generated diagonal masking matrix for 50% masking, fixed throughout training. However, state-of-the-art methods in contrastive learning [Chen et al., 2020] instead use stochastic modules and generate random augmentations for every input batch, which contradicts the setup used by the authors.

We generalize random masking augmentation to, (i) no flipping, i.e. $g_1(x_i) = A^1 x_i$, $g_2(x_i) = A^2 x_i$, (ii) no fixing, i.e. $g(x_i) = A_i x_i$, $g_2(x_i) = (I - A_i)x_i$, and (iii) both, i.e. $g_1(x_i) = A_i^1 x_i$, $g_2(x_i) = A_i^2 x_i$, and collect results in Figure 4. The learning model performs the worst when there is no flipping but the masks are fixed, and can be explained as this setting is capable of causing overlap between the two augmentations which will never be corrected due to fixed masks, causing the model to never see certain dimensions of the input. While the flip and fix mask setting originally used by the authors performs reasonably, the best performance by a large margin is without flipping or fixing the masks, similar to stochastic augmentation settings proposed in literature. This implies a significant drop in performance for discussions and empirical results present in Ji et al. [2021].

Masking		Sine	Downstream
Flip	Fix	Distance	Risk
✓	✓	1.3907	0.02813
✓	✗	2.0430	0.10549
✗	✓	2.6922	0.22263
✗	✗	**0.7875**	**0.01687**

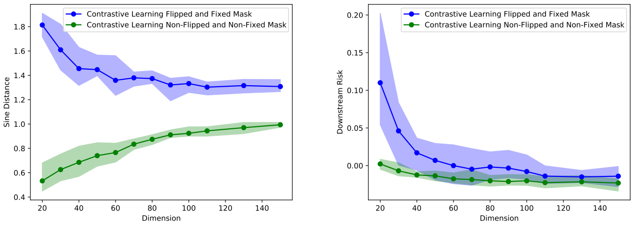


Figure 4: **Left:** Contrastive learning under different augmentation settings. **Right:** Sine distance and downstream risk against d for contrastive learning under different augmentation settings.

We further extend our experiments and re-evaluate the trends shown by the authors (specifically, varying the value of observation dimension d) in light of improved stochastic random masking augmentation. The results are collected in Figure 4. We observe that stochastic augmentation outperforms its counterpart throughout the experiment, consistent with our discussion above. We thus believe further work is required in situating the bounds [Ji et al., 2021] in context of practical stochastic augmentation functions. We also note that behavior for sine distance and downstream risk is opposite of each other, a trend we will discuss in detail later in context of Figure 5.

Measurement noise Ji et al. [2021] employs a zero-mean sub-gaussian heteroskedastic (i.e., different variance values) noise generation with ratio $\sigma_{(1)}/\sigma_{(d)}$ bounded by C. As detailed in Section 2, we use $C = 2$ to reproduce original results. Here, we further investigate the ratio of heteroskedasticity, including special case $C = 1$ (homogenous noise), and collect results in Figure 5.

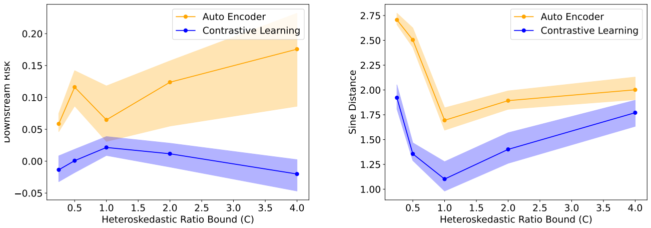
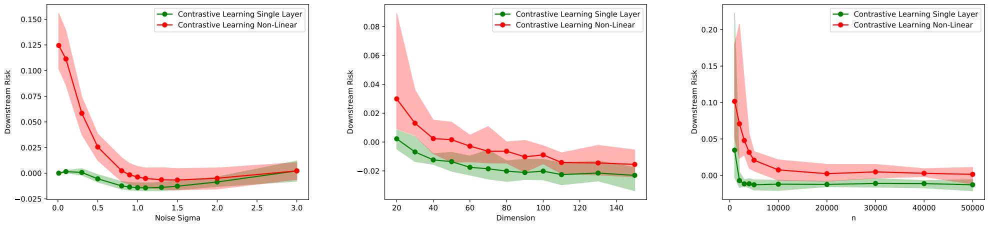


Figure 5: Experiments with various values of heteroskedastic ratio bound C

Interestingly, it seems that trends of sine distance and downstream risk are opposite, most notably the peak downstream risk and low sine distance for contrastive learning at $C = 1$. While understanding this anomaly would require further experiments beyond the scope of this work, we would like to use these results to emphasize the potentially opposing nature of the two metrics. Ji et al. [2021] assumes that both sine distance and downstream risk are straightforward indicators of the model's performance, however we note that extracting back the original mapping (i.e. sine distance) or reproducing the latent feature information for downstream tasks (i.e. downstream risk) are two separate objectives which might provide separate trends under certain setups (as we also spotted earlier in Figure 4).

We also notice significant negative downstream risk for higher values of C. While negative downstream risk is mildly present in Figure 2c,4c (and will be noticed later in Figure 6), its significantly visible and consistent here. As discussed in Section 1.1, downstream risk is the difference between downstream performance of representations extracted by ${U^*}^T$ and W. Since the downstream labels are created using original latent features z and representations extracted by ${U^*}^T$, i.e. $z^* = {U^*}^T x = z + {U^*}^T \xi$, still contain the noise term, negative downstream risk shows learning model's ability to remove noise from extracted representations and outperform ${U^*}^T$. This supports the current understanding of contrastive learning's robustness in literature [Emami et al., 2021].

Non-linear models The original work by Ji et al. [2021] as well as our discussion until now has focused exclusively on linear representation setting. We now generalize our work beyond linear models and study the trends for deep non-linear learning, in particular a two-layer dense network with rectified linear units (ReLU) as activation. Note that we do not change the original data generation module to create an overparameterized non-linear setting. Additionally, based on our observations from Figure 4, we use stochastic augmentation (no flipping, no fixing) for these experiments.



(a) (b) (c)

Figure 6: Comparing single-layer (linear) and two-layer (non-linear) models

We test the downstream risk with changing values of noise sigma $\sigma_{(1)}$, observation dimension d, and train size n, and collect the results in Figure 6. We first point out the existence of consistently negative downstream risk, which indicates the robust nature of contrastive learning as discussed above. Next, we look at the comparison between single-layer (linear) and two-layer (non-linear) models. Surprisingly, non-linear models seem to perform noticeably worse than linear models across all variations of hyperparameters. While we speculate the possibility of overfitting, or linear models exploiting the advantage that underlying data generation is linear, no concrete conclusions can be made based on these results alone. A thorough extension of this work focused on non-linear models in the future can possibly uncover the reasoning behind this counter-intuitive behavior.

3 Future Directions of Research

In this work, we present detailed experiments and explore various assumptions, theoretical bounds, and generalization beyond the original paper by Ji et al. [2021]. While we found extensions of their work in which the models follow the proposed behavior, we point out several issues with the underlying assumptions used by the authors, emphasize the distance between theoretical bounds and empirical performance, and provide conclusive evidence of the expected trends breaking down when generalized to real-world settings. We believe that the work of Ji et al. [2021] is an important stepping stone in understanding the superiority of contrast-based representation learning, but is still quite far away from explaining real-world behavior. We expect future research in this field to rise above the constrained view of linear setting, adapt to the ever-growing improvements in augmentation setups, and potentially embrace atypical applications like few-shot learning, meta-learning, etc.

References

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

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.

Han Bao, Yoshihiro Nagano, and Kento Nozawa. Sharp learning bounds for contrastive unsupervised representation learning. *arXiv preprint arXiv:2110.02501*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Melikasadat Emami, Dung Tran, and Kazuhito Koishida. Augmented contrastive self-supervised learning for audio invariant representations. *arXiv preprint arXiv:2112.10950*, 2021.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34, 2021.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

Wenlong Ji, Zhun Deng, Ryumei Nakada, James Zou, and Linjun Zhang. The power of contrast for feature learning: A theoretical analysis. *arXiv preprint arXiv:2110.02473*, 2021.

Joelle Pineau, Koustuv Sinha, Genevieve Fried, Rosemary Nan Ke, and Hugo Larochelle. Iclr reproducibility challenge 2019. 2019.

DE Rumelhart, GE Hinton, and RJ Williams. Learning internal representations by error propagation. In *Neurocomputing: foundations of research*, pages 673–695. 1988.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

Lilian Weng. Self-supervised representation learning, Nov 2019. URL `https://lilianweng.github.io/posts/2019-11-10-self-supervised/`.

Anru R Zhang, T Tony Cai, and Yihong Wu. Heteroskedastic pca: Algorithm, optimality, and applications. *The Annals of Statistics*, 50(1):53–80, 2022.

A Appendix

A.1 Reproducibility Checklist

Data Generation We use the *spiked covariance model* defined by Ji et al. [2021] for data generation. We employ singular value decomposition of a randomly created matrix to generate the orthonormal mapping U^*. The formulae are directly implemented as defined in the original paper, with the default values provided below. For all experiments reported in our work, we use 10 different generated datasets, corresponding to 10 random seeds, to compare the range and average performance. While we do not provide the code for generated datasets (as they vary based on the chosen settings), we provide our code to replicate data creation for same random seeds that we used.

Train and Test Split Training dataset size n is a hyperparameter which is changed based on the experiment (more details below). We used a fixed test dataset size $m = 1000$.

Default Hyperparameter Values We use default values of latent feature size $r = 10$; generated observation size $d = 40$; training data size $n = 20,000$; latent feature variance $v = 1$; signal to noise ratio $\rho = 1$; regular covariance bound for noise generation $C = 2$; and standard deviation of downstream label generation noise $std(\epsilon) = 0.1$. We use an effective batch size of 64 (formed by applying two augmentations g_1, g_2 on an input batch of size 32), and random mask augmentation with bernoulli mean 0.5. All experiments are repeated for 10 random seeds by default.

We use *PyTorch* library for representation model training. All models are trained for a total of 50 epochs, with learning rate $3e-4$ and regularization constant $1e-3$. We use *Scikit-learn* library for learning downstream regression, and use linear regression model for the same.

Non-default Values In cases where values other than the default are used, we appropriately reference them in the main text. Most notably, we use stochastic augmentations for results in Figure 4, 6.

A.2 Reproducing Original Empirical Results

We reproduce the original empirical results (see Figure 7) as a sanity check for our implementation. These experiments were performed under default settings as described in Appendix A.1.

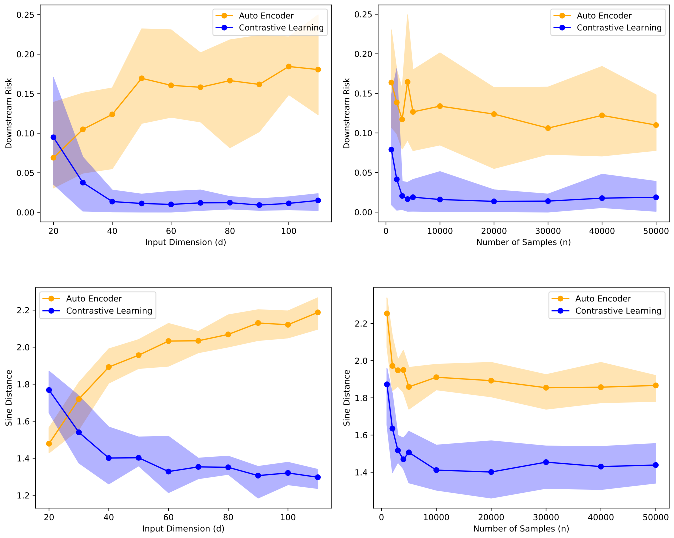


Figure 7: Reproducing empirical results for variation across input dimension (d) and train size (n)

A.3 Classification Labels

Continuing our discussion in Section 1.1, the authors generate downstream classification labels as,

$$y|z \sim Ber(F(\langle z, w^* \rangle / v)), \text{where } F(u) = 1/(1 + e^{-u}) \tag{7}$$

In other words, the angle between coefficient vector w^* and latent feature z, normalised by signal standard deviation v and sigmoid function F, provides the probability of the classification label being chosen. If the input to $Ber()$ is close to 1, the output label is highly likely to be 1, and similarly if the input to $Ber()$ is close to 0, the output label is highly likely to be 0. However, it is important to note that when the input to $Ber()$ is close to 0.5, the output label is chosen randomly with equal probabilities of being 0 or 1. Such a label generation contains no relevant generalizable signal and will require explicit memorization if a large portion of inputs have label selection probability close to 0.5. Since data generation follows a zero-mean distribution; and the coefficient vector w^* is created randomly with no influence from data generation, we believe that it is likely for a significant portion of the data to have close to randomly generated labels.

To emphasize this, we plot a histogram of selection probabilities for a dataset of size $n = 20,000$, and collect results in Figure 8. A large portion of data points ($\sim$ 6000, or 30%) have bernoulli mean between 0.4 and 0.6 when generating classification labels, implying that the labels for these data points will be generated almost randomly. On the other hand, only a small percentage of total examples (~ 3000 or 15%) have labels generated with high probability in any one direction. This shows the negative impact of using a distribution centered on 0.5 as input for classification label generation. To finally verify our claim of the absence of learning signal in the dataset, we train a classifier on top of the original latent features z and the generated classification labels. Note that this is different from the downstream score of U^* discussed in the main text; as here we even bypass the noise in data generation and directly use the gold latent features to emphasize our claim. The linear classifier even in this gold setting achieves an accuracy of only 65.2% (a random classifier can achieve 50% accuracy as it is binary classification with balanced classes). Due to these issues with the classification label generation, we decide to use only regression labels for downstream comparison.

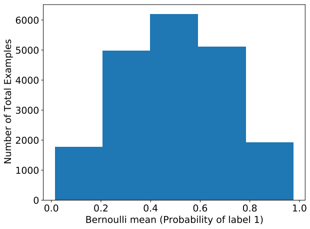
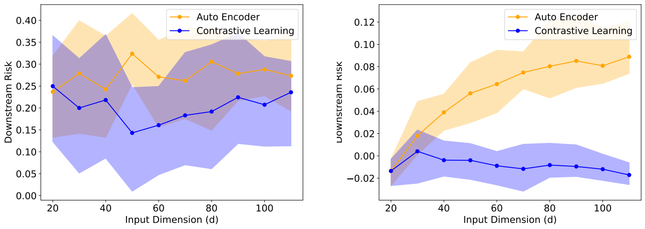


Figure 8: Histogram of generated bernoulli mean values for label creation

A.4 Additional results for $r_{latent} \neq r_{model}$

Additional results for $r_{model} = 5$ and $r_{model} = 20$ while varying the observation dimension d are collected in Figure 9. The trends are in line with the results in Figure 2 and discussion in Section 2.1.



(a) $r_{model} = 5$

(b) $r_{model} = 20$

Figure 9: Additional results : Investigating latent dimension of the learning model r_{model}

A.5 Additional results for theoretical bounds against high values of d

Against a setting of fixed values of r and n, we investigated how very high values of input dimension d compare against the upper bound on performance. Our observation indicates that there is a more nuanced relation between the input feature dimension d and its latent dimension r than the one encapsulated in the assumption $d \gg r$ for the bounds. In practice, inputs with large number of features would require a certain number of latent dimensions for the contrastive learner to achieve performance as good as indicated by Ji et al. [2021].

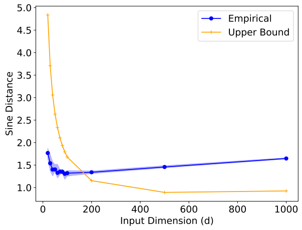


Figure 10: Empirical contrastive learning compared to upper bound against feature dimension d