Simple Disentanglement of Style and Content in Visual Representations

Lilian Ngweta [*1] Subha Maity [*2] Alex Gittens [1] Yuekai Sun [2] Mikhail Yurochkin [3 4]

Abstract

Learning visual representations with interpretable features, i.e., disentangled representations, remains a challenging problem. Existing methods demonstrate some success but are hard to apply to large-scale vision datasets like ImageNet. In this work, we propose a simple post-processing framework to disentangle content and style in learned representations from pre-trained vision models. We model the pre-trained features probabilistically as linearly entangled combinations of the latent content and style factors and develop a simple disentanglement algorithm based on the probabilistic model. We show that the method provably disentangles content and style features and verify its efficacy empirically. Our post-processed features yield significant domain generalization performance improvements when the distribution shift occurs due to style changes or style-related spurious correlations.

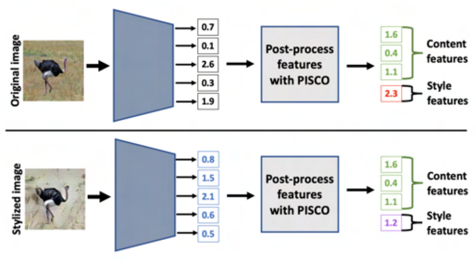


Figure 1. Illustration of the proposed method, PISCO. Features of an original image and features of a stylized image are different when extracted using a feature extractor such as ResNet-50 pre-trained in ImageNet. These features are entangled, thus changing the style affects all features. When PISCO is used to disentangle these features, it isolates style features and content features, thus the content features of the two images are the same and only the style features are different.

1. Introduction

Deep learning models produce data representations that are useful for many downstream tasks. Disentangled representations, i.e. representations where coordinates have meaningful interpretations, are harder to learn (Locatello et al., 2019b) but they come with many additional benefits, e.g., data-efficiency (Higgins et al., 2018) and use-cases in causality (Schölkopf et al., 2021), fairness (Locatello et al., 2019a), recommender systems (Ma et al., 2019), and image (Lee et al., 2018) and text (John et al., 2018) processing.

In this paper, we consider the problem of isolating content from style in visual representations (Wu et al., 2019; Nemeth, 2020; Ren et al., 2021; Kügelgen et al., 2021), a special case of learning disentangled representations. Here we use the term style to refer to features or factors that are not causally related to the outcome of interest. We also note prior works that, different from our work, study style in the context of image appearance (Garcia & Vogiatzis, 2018; Saleh & Elgammal, 2015; Ruta et al., 2022; 2021). Our goal is to obtain representations where a pre-specified set of factors (coordinates) encodes image "styles" (e.g., rotation, color scheme, or style transfer (Huang & Belongie, 2017)), while the remaining factors encode content and are invariant to style changes (see Figure 1).

An important application of such representations is out-of-distribution (OOD) generalization. Image recognition systems have been demonstrated to be susceptible to spurious correlations associated with style, e.g., due to background colors (Beery et al., 2018; Sagawa et al., 2019), and to various style-based distribution shifts, e.g., due to image corruptions (Hendrycks & Dietterich, 2018), illumination, or camera angle differences (Koh et al., 2020). Simply discarding style factors when training a prediction model on disentangled representations can aid OOD generalization. Disentangling content from style is also advantageous in many other applications, e.g., image retrieval (Wu et al., 2009), image-to-image translation (Ren et al., 2021), and

[*]Equal contribution [1]Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York, United States [2]Department of Statistics, University of Michigan, Ann Arbor, Michigan, United States [3]IBM Research, Cambridge, Massachusetts, United States [4]MIT-IBM Watson AI Lab, Cambridge, Massachusetts, United States. Correspondence to: Lilian Ngweta <ngwetl@rpi.edu>, Subha Maity <smaity@umich.edu>, Mikhail Yurochkin <mikhail.yurochkin@ibm.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

visually-aware recommender systems (Deldjoo et al., 2022).

While there is abundant literature on learning disentangled representations, most statistically principled methods fit sophisticated generative models (Bouchacourt et al., 2018; Hosoya, 2018; Shu et al., 2019; Wu et al., 2019; Locatello et al., 2020). These methods work well on synthetic and smaller datasets but are hard to train on larger datasets like ImageNet (Russakovsky et al., 2015). This is in stark contrast to representation learning practice; the most common representation learning methods only learn an encoder (*e.g.* SimCLR (Chen et al., 2020)). That said, there are some recent works that consider how to learn disentangled encoders (Zimmermann et al., 2021; Kügelgen et al., 2021; Wang et al., 2021).

Specific to style and content, Kügelgen et al. (2021) show that contrastive learning, e.g., SimCLR (Chen et al., 2020), *theoretically* can isolate style and content, i.e. learn representations that are invariant to style. Contrastive learning methods are gaining popularity due to their ability to learn high-quality representations from large image datasets without labels via self-supervision (Doersch et al., 2015; Chen et al., 2020; Chen & Batmanghelich, 2020; Grill et al., 2020; Chen & He, 2021). Unfortunately, style invariance of contrastive learning representations is rarely achieved in practice due to a variety of additional requirements that are hard to control for (see Section 5 and Appendix C.1 in Kügelgen et al. (2021)).

Most of the prior works are in-processing methods that train an end-to-end encoder from scratch. On the other hand, we focus on post-processing representations from a pre-trained deep model (which may not be disentangled) so that they become provably disentangled. The post-processing setup is appealing as it allows re-using large pre-trained models, thus reducing the carbon footprint of training new large models (Strubell et al., 2019) and making deep learning more accessible to practitioners with limited computing budgets. Post-processing problem setups are prominent in the algorithmic fairness literature (Wei et al., 2019; Petersen et al., 2021).

To develop our post-processing setup for learning disentangled representations, we assume that the pre-trained representations are simply an *invertible* linear transformation of the style and content factors (*cf.* Assumption 2.1). While the linear model assumption may appear too simple at a first glance, it is motivated by the result of Zimmermann et al. (2021) showing that contrastive learning recovers true data-generating factors up to an *orthogonal* transformation. The representation may not come from a contrastive learning model or assumptions of Zimmermann et al. (2021) might be violated in practice, thus we consider a more general class of linear invertible transformations in our model which we justify theoretically and verify empirically. Our contributions are summarized below:

- We formulate a simple linear model of entanglement in pre-trained visual representations and a corresponding method for Post-processing to Isolate Style and COntent (PISCO).
- We establish theoretical guarantees that PISCO learns disentangled style and content factors and recovers correlations among styles. Our theory is supported by a synthetic dataset study.
- We verify the ability of PISCO to disentangle style and content on three image datasets of varying size and complexity via post-processing of various pre-trained deep visual feature extractors. In our experiments, discarding the learned style factors yields significant out-of-distribution performance improvements while preserving the in-distribution accuracy.

2. Problem formulation

In light of the recent success of contrastive learning techniques for obtaining self-supervised embeddings, Zimmermann et al. (2021) have performed a theoretical investigation on the InfoNCE family (Gutmann & Hyvärinen, 2012; Oord et al., 2018; Chen et al., 2020) of contrastive losses. Under some distributional assumptions, their investigation reveals that InfoNCE loss can invert the underlying generative model of the observed data. More specifically, given an observed data $\mathbf{x} = g(\mathbf{z})$ where $\mathbf{z}$ and g are correspondingly the underlying latent factors and the generative model, Zimmermann et al. (2021) showed that InfoNCE loss finds a representation model f such that $f \circ g(\mathbf{z}) = \mathbf{R}\mathbf{z}$ for some orthogonal matrix $\mathbf{R}$. Though this is quite welcoming news in nonlinear independent component analysis (ICA) literature (Hyvärinen & Pajunen, 1999; Hyvärinen & Morioka, 2016; Jutten et al., 2010), the representation model may not be good enough for learning a disentangled representation. In fact, Zimmermann et al. (2021) show that only under a very specific generative modeling assumption a type of contrastive objective can achieve disentanglement, and that disentanglement is lost when the assumptions are violated.

At a high level, disentanglement in representation learning means the style and content factors are not affected by each other. Looking back at the result (Zimmermann et al., 2021) that contrastive loss can recover the generative latent factors up to an unknown rotation, an implication is that disentanglement in the learned representation may not be achieved. However, all is not lost; we suggest a simple post-processing method for the learned representations and show that it achieves the desired disentanglement.

We now formally describe our post-processing setup. We denote the space of the latent factor as $\mathcal{Z} \subset \mathbf{R}^d$ and assume that the latent factor is being generated from a probability distribution $\mathbb{P}_{\mathbf{z}}$ on $\mathcal{Z}$. Similar to Zimmermann et al. (2021) we assume that there exists a one-to-one generative map g such that the observed data is generated as $\mathcal{X} \ni \mathbf{x} = g(\mathbf{z}),\ \mathbf{z} \sim \mathbb{P}_{\mathbf{z}}$. The next assumption is crucial for our linear post-processing technique and is motivated by the finding in Zimmermann et al. (2021).

Assumption 2.1. There exists a representation map $f : \mathcal{X} \to \mathcal{Z}' \subset \mathbf{R}^{d'}$ such that $f \circ g(\mathbf{z}) = \mathbf{A}\mathbf{z}$ for some left invertible matrix $\mathbf{A} \in \mathbf{R}^{d' \times d}$.

An example of such f could be the representation model learned from InfoNCE loss minimization, where Zimmermann et al. (2021) showed that the assumption is true for $\mathcal{Z}' = \mathcal{Z}$ and $\mathbf{A}$ is an orthogonal matrix. A consequence of left invertibility for $\mathbf{A}$ is that $d' \geq d$, *i.e.*, the dimension of learned representation could be potentially higher than that of the generating latent factors, which is often natural to assume in many applications.

Throughout the paper, we denote $f \circ g(\mathbf{z})$ as $\mathbf{u}$ and call it *entangled representation*. With this setup, we're now ready to formally specify the disentanglement (also known as sparse recovery) in representation learning.

Definition 2.2 (Disentangled representation learning/sparse recovery)**.** Let us denote $\mathbf{F}_{\rm S} \subset [d] \triangleq \{1, 2, \ldots, d\}$ as the set of **style factors** and it's cardinality as $m \triangleq |\mathbf{F}_{\rm S}|$. We denote the remaining factors $\mathbf{F}_{\rm C} \triangleq [d] - \mathbf{F}_{\rm S}$ and call them **content factors**. For a matrix $\mathbf{P} \in \mathbf{R}^{d \times d'}$ we say **the linear post-processing $\mathbf{u} \mapsto \hat{\mathbf{z}} \triangleq \mathbf{P}\mathbf{u}$ disentangles or sparsely recovers the style and content factors** if the following hold for the matrix $\mathbf{PA}$:

1. $[\mathbf{PA}]_{\mathbf{F}_{\rm S}, \mathbf{F}_{\rm S}}$ is an $m \times m$ diagonal matrix.
2. $[\mathbf{PA}]_{\mathbf{F}_{\rm C}, \mathbf{F}_{\rm C}}$ is a $(d - m) \times (d - m)$ invertible matrix.
3. $[\mathbf{PA}]_{\mathbf{F}_{\rm S}, \mathbf{F}_{\rm C}}$ and $[\mathbf{PA}]_{\mathbf{F}_{\rm C}, \mathbf{F}_{\rm S}}$ are $m \times (d - m)$ and $(d - m) \times m$ null matrices.

In other words, $\hat{\mathbf{z}} = \mathbf{P}_{\mathcal{S}}\mathbf{u}$ is disentangled or sparsely recovered in $\mathbf{F}_{\rm S}$ if for any $j \in \mathbf{F}_{\rm S}$ the coordinate $[\hat{\mathbf{z}}]_j$ is a constant multiplication of $[\mathbf{z}]_j$ and $[\hat{\mathbf{z}}]_{\mathbf{F}_{\rm C}}$ is just a pre-multiplication of $[\mathbf{z}]_{\mathbf{F}_{\rm C}}$ by an invertible matrix.

Without loss of generality we assume that $\mathbf{F}_{\rm S} = [m]$. Next, we highlight a conclusion of the sparse recovery, which has a connection to independent component analysis (ICA).

Corollary 2.3 (Correlation recovery)**.** *One of the conclusions of sparse recovery is that the estimated style factors have the same correlation structure as the true style factors. Denoting* $\text{corr}(\mathbf{X})$ *as the correlation matrix for a generic random vector* $\mathbf{X}$ *the conclusion can be mathematically stated as*

$$\text{corr}([\hat{\mathbf{z}}]_{\mathbf{F}_S}) = \text{corr}([\mathbf{z}]_{\mathbf{F}_S})\,. \tag{2.1}$$

A proof of the statement is provided in §A.3. In a special case connected to ICA, where the true correlation distribution $\mathbb{P}_{\mathbf{z}}$ *has uncorrelated style factors, i.e.,* $\text{corr}([\mathbf{z}]_{\mathbf{F}_S}) = \mathbf{I}_m$*, then same is true for estimated style factors.*

The rest of the paper describes the estimation of $\mathbf{P}_{\mathcal{S}}$ and investigates its quality in achieving disentanglement.

3. PISCO

To achieve sparse recovery, we assume that we can manipulate the samples in some specific ways, which we describe below.

Assumption 3.1. We assume the following:

1. **Sample manipulations:** For each sample $\mathbf{x} = g(\mathbf{z})$ and style factor $j \in \mathbf{F}_{\rm S}$ we have access to the sample $\mathbf{x}^{(j)} \triangleq g(\mathbf{z}^{(j)})$ that has been created by modifying the j-th style factor of $\mathbf{x}$ while keeping content factors unchanged, *i.e.*,
$$[\mathbf{z}^{(j)}]_i = \begin{cases} \neq [\mathbf{z}]_i, & i = j, \\ [\mathbf{z}]_i, & i \in \mathbf{F}_{\rm C}\,. \end{cases} \tag{3.1}$$
2. **Sample annotations:** There exist two numbers $\alpha_j, \beta_j \in \mathbf{R},\ \beta_j \neq 0$ which are associated to each j-th style factor and independent of the latent factors $\mathbf{z}$ such that for each sample $\mathbf{x} = g(\mathbf{z})$ and it's modified version $\mathbf{x}^{(j)} = g(\mathbf{z}^{(j)})$ we observe the sample annotations $\mathbf{y}_j = \alpha_j + \beta_j[\mathbf{z}]_j + \epsilon^{(j)}$ and $\tilde{\mathbf{y}}_j = \alpha_j + \beta_j[\mathbf{z}^{(j)}]_j + \tilde{\epsilon}^{(j)}$, where $(\epsilon^{(j)}, \tilde{\epsilon}^{(j)})$ pair has zero mean and is uncorrelated with $(\mathbf{z}, \mathbf{z}^{(j)})$.

Sample annotations formalize the notion of concept from interpretable ML (Kim et al., 2018) and generalize the usual disentangled representation setting in which the latent factors are the concepts. By taking $\alpha_j = 0$ and $\beta_j = 1$, we have $\mathbf{y}_j = [\mathbf{z}]_j$ and $\tilde{\mathbf{y}}_j = [\mathbf{z}^{(j)}]_j$, which equates the annotations and the latent factors. We provide an illustration for a single style factor, *i.e.* $\mathbf{F}_{\rm S} = \{1\}$, in Figure 1. Here, $\mathbf{x}$ is the original image and we annotate it as $\alpha_1 + \beta_1[\mathbf{z}]_1 = +1$. We stylize the image to obtain $\mathbf{x}^{(1)}$ and assume that style transformation does not change any content factors of the image. We annotate the transformed image as $\alpha_1 + \beta_1[\mathbf{z}^{(1)}]_1 = -1$. Examples of such sample manipulations are easily available in vision problems, *e.g.*, image corruptions (Hendrycks & Dietterich, 2018) and style transfer (Huang & Belongie, 2017). Combining style transfer and prompt-based image generation systems like DALL·E 2 further enables using natural language to describe desired sample manipulations (Figure 1 illustrates such image manipulation - see Appendix C.1 for prompt and other details and Figure 5 for more examples). In §5 we use these examples for our experiments.

We denote the entangled representations (obtained from Assumption 2.1) corresponding to the images $\mathbf{x}$ and $\mathbf{x}^{(j)}$ as $\mathbf{u}$ and $\mathbf{u}^{(j)}$. With access to such sample manipulations, one can recover the j-th latent factor from a simple minimum norm least square regression problem:

$$\begin{aligned}[\hat{\mathbf{z}}]_j \triangleq \hat{\mathbf{p}}_j^\top \mathbf{u}, \ \text{where}\\ \hat{\mathbf{p}}_j \triangleq \lim_{\mu\to 0+} \operatorname*{arg\,min}_{a\in\mathbf{R},\mathbf{p}\in\mathbf{R}^{d'}} \frac{1}{2n}\sum_{i=1}^n \Big[(\mathbf{y}_i^{(j)} - a - \mathbf{p}^\top \mathbf{u}_i)^2\\ +(\tilde{\mathbf{y}}_i^{(j)} - a - \mathbf{p}^\top \mathbf{u}_i^{(j)})^2\Big] + \frac{\mu}{2}\|\mathbf{p}\|_2^2\end{aligned} \tag{3.2}$$

We resort to the minimum norm least square regression instead of the simple least square regression because the variance of the predictor $\text{var}(\mathbf{u}) = \text{var}(\mathbf{A}z) = \mathbf{A}\text{var}(\mathbf{z})\mathbf{A}^\top$ has rank $d \le d'$, which leads to non-invertible covariance for the design matrix whenever $d < d'$.

Intuitively, $\hat{\mathbf{p}}_j^\top \mathbf{u}$ is the one dimensional linear function of $\mathbf{u}$ which is most aligned to the coordinate $[\mathbf{z}]_j$. As we shall see later, under our setup $\hat{\mathbf{p}}_j^\top \mathbf{u}$ is just a scalar multiple of $[\mathbf{z}]_j$, and hence we successfully recover the j-th style factor. We stack $\hat{\mathbf{p}}_j$ into the j-th row of $\mathbf{P}$, *i.e.* $[\mathbf{P}]_{j,\cdot} = \hat{\mathbf{p}}_j^\top$.

To extract the content factors we first recall that they should exhibit minimal change corresponding to any changes in the style factors $[\mathbf{z}]_j,\ j \in \mathbf{F}_\text{S}$. We enforce this by leveraging our ability to manipulate styles of samples, as described in Assumption 3.1. We describe our method below:

Estimation of content factors: We recall $m = |\mathbf{F}_\text{S}|$ and let $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_n, \mathbf{u}_1^{(j)}, \dots, \mathbf{u}_n^{(j)}; j \in \mathbf{F}_\text{S}]^\top \in \mathbf{R}^{(m+1)n\times d'}$ be the matrix of entangled representations, and for each $j \in \mathcal{S}$ let $\Delta_j = [\mathbf{u}_1 - \mathbf{u}_1^{(j)}, \dots, \mathbf{u}_n - \mathbf{u}_n^{(j)}] \in \mathbf{R}^{n\times d'}$ be the matrix of representation differences. We estimate the content factors from the following optimization:

$$\begin{aligned}[\hat{\mathbf{z}}]_{\mathbf{F}_\text{C}} = \hat{\mathbf{Q}}(\lambda)\mathbf{u}, \ \text{where}\\ \hat{\mathbf{Q}}(\lambda) \triangleq \operatorname*{arg\,min}_{\substack{\mathbf{Q}\in\mathbf{R}^{(d-|\mathbf{F}_\text{S}|)\times d'}\\ \mathbf{Q}\mathbf{Q}^\top = \mathbf{I}}} \text{tr}\left[\left(\mathbf{I}_{d'} - \mathbf{Q}^\top\mathbf{Q}\right)\left(\tfrac{\mathbf{U}^\top\mathbf{U}}{(m+1)n}\right)\right]\\ + \tfrac{\lambda}{m}\textstyle\sum_{j\in\mathbf{F}_\text{S}} \text{tr}\left[\mathbf{Q}^\top\mathbf{Q}\left(\Delta_j^\top\Delta_j/n\right)\right].\end{aligned} \tag{3.3}$$

Our objective has two parts: the first part is easily recognized by noticing its similarity to a principle component analysis objective. To understand the second part, we fix a style factor $j \in \mathbf{F}_\text{S}$ and observe that,

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^n \|\mathbf{Q}(\mathbf{u}_i - \mathbf{u}_i^{(j)})\|_2^2\\ &= \frac{1}{n}\sum_{i=1}^n \text{tr}\left[\mathbf{Q}^\top\mathbf{Q}(\mathbf{u}_i - \mathbf{u}_i^{(j)})(\mathbf{u}_i - \mathbf{u}_i^{(j)})^\top\right]\\ &= \text{tr}\left[\mathbf{Q}^\top\mathbf{Q}\left(\Delta_j^\top\Delta_j/n\right)\right].\end{aligned} \tag{3.4}$$

Algorithm 1 PISCO

```
Input: Dataset and styles: (1) entangled representations $\{\mathbf{u}_i\}_{i=1}^n \subset \mathbf{R}^{d'}$ of $n$ images, and (2) $m$ styles. Hyperparameters: (1) regularization strength for disentanglement between style and content factors $\lambda > 0$, and (2) number of content factors $k$.
{representations of }
for $j = 1$ to $m$ do
  for $i = 1$ to $n$ do
    $\mathbf{u}_i^{(j)} \leftarrow$ entangled feature of $i$-th image after changing it's $j$-th style.
    $\delta_i^{(j)} \leftarrow \mathbf{u}_i^{(j)} - \mathbf{u}_i$.
  end for
  $\hat{\mathbf{p}}_j \leftarrow$ coefficient from regression (3.2) on $\{(\mathbf{u}_i, -1)\}_{i=1}^n \cup \{(\mathbf{u}_i^{(j)}, +1)\}_{i=1}^n$.
end for
$\mathbf{U} \leftarrow [\mathbf{u}_1, \mathbf{u}_1^{(1)}, \dots, \mathbf{u}_1^{(m)}, \dots, \mathbf{u}_n, \mathbf{u}_n^{(1)}, \dots, \mathbf{u}_n^{(m)}]^\top \in \mathbf{R}^{n(m+1)\times d'}$
$\Delta \leftarrow [\delta_1^{(1)}, \dots, \delta_1^{(m)}, \dots, \delta_n^{(1)}, \dots, \delta_n^{(m)}]^\top \in \mathbf{R}^{mn\times d'}$
$\hat{\mathbf{Q}}(\lambda) \leftarrow$ top $k$ eigenvectors of $\frac{\mathbf{U}^\top\mathbf{U}}{n(m+1)} - \lambda\frac{\Delta^\top\Delta}{mn}$
Return: Post-processing matrix $\mathbf{P}(\lambda)$ as in (3.5).
```

Following the above, one can easily realize that the second part of the objective enforces that the content factors $[\hat{\mathbf{z}}]_{\mathbf{F}_\text{C}}$ exhibit minimal change for any changes in the style factors $[\mathbf{z}]_j,\ j \in \mathbf{F}_\text{S}$. In a special case $\lambda = +\infty$, the $[\hat{\mathbf{z}}]_{\mathbf{F}_\text{S}}$ will be invariant to any changes in the style factors.

From (3.2) and (3.3) we obtain the linear post-processing matrix (as defined in 2.2) as

$$\mathbf{P} \equiv \mathbf{P}(\lambda) \triangleq \begin{cases}[\mathbf{P}]_{j,\cdot} = \hat{\mathbf{p}}_j^\top, & j \in \mathbf{F}_\text{S}\\ [\mathbf{P}]_{\mathbf{F}_\text{C},\cdot} = \hat{\mathbf{Q}}(\lambda).\end{cases} \tag{3.5}$$

We summarize our method in Algorithm 1 which is a combination of simple regressions (per style factors) and an eigen-decomposition. In the next section, we show that for large values of λ the post-processing matrix $\mathbf{P}(\lambda)$ achieves sparse recovery with high probability.

4. Theory

In this section, we theoretically establish that our post-processing approach PISCO guarantees sparse recovery. We divide the proof into two parts: the first part analyzes the asymptotic quality of the estimated style factors and the second part analyzes the quality of the estimated content factors. Our first result follows:

Theorem 4.1. *Let $\Sigma_\mathbf{z} \triangleq \text{var}(\mathbf{z})$ be invertible. Then for any $j \in \mathbf{F}_S$ it holds:*

$$\mathbf{A}^\top\hat{\mathbf{p}}_j \to \beta_j e_j \tag{4.1}$$

almost surely as $n \to \infty$, where $\{e_j\}_{j=1}^d$ is the canonical basis vector for $\mathbf{R}^d$. Subsequently, the following hold at *almost sure limit: (1)* $[\mathbf{PA}]_{\mathbf{F}_S,\mathbf{F}_S}$ *converges to a diagonal matrix, and (2)* $[\mathbf{PA}]_{\mathbf{F}_S,\mathbf{F}_C} \to \mathbf{0}$.

To establish theoretical guarantee for the content factors we require the following technical assumption.

Assumption 4.2 (Linear independence in style factors)**.** Over the distribution of latent factors, the style factors are not linearly dependent with each other. Mathematically speaking, the following event positive probability

$$\mathbf{E} = \left\{\mathbf{z} : [\mathbf{z}-\mathbf{z}^{(1)}, \dots, \mathbf{z}-\mathbf{z}^{(m)}]_{\mathbf{F}_S,\cdot} \text{ is invertible}\right\}. \quad (4.2)$$

The assumption is related to cases when the style factors are dependent with each other. One such example is the blurring and contrasting of images: we assume that one doesn't *completely* determine the other. To see how the assumption is violated under linear dependence of style factors let the first two of them completely determine one another, *i.e.*, one is just a constant multiplication of the other. In that case, we point out that for any $\mathbf{z}$ the first two rows of the matrix $[\mathbf{z}-\mathbf{z}^{(1)}, \dots, \mathbf{z}-\mathbf{z}^{(m)}]_{\mathbf{F}_S,\cdot}$ are just constant multiplication of one another and the matrix is singular with probability one.

With the setups provided by Assumptions 2.1, 3.1 and 4.2 we're now ready to state our result about sparse recovery for our post-processing technique.

Theorem 4.3 (Sparse recovery for PISCO)**.** *Let* $n \geq d+1$ *and that the Assumptions 2.1, 3.1 and 4.2 hold. Define* $\kappa \triangleq 1 - \mathbb{P}_{\mathbf{z}}(\mathbf{E}) < 1$ *where* $\mathbb{P}_{\mathbf{z}}$ *is the distribution of latent factors and* $\mathbf{E}$ *is defined in* (4.2)*. With probability at least* $1-\kappa^n$ *the post-processing matrix* $\mathbf{P} \equiv \mathbf{P}(+\infty) \triangleq \lim_{\lambda\to\infty} \mathbf{P}(\lambda)$ *satisfies the following: (1)* $[\mathbf{PA}]_{\mathbf{F}_S,\mathbf{F}_C} = \mathbf{0}$*, and (2)* $[\mathbf{PA}]_{\mathbf{F}_C,\mathbf{F}_C}$ *is invertible.*

Proofs of Theorems 4.1 and 4.3 are provided in §A.2 and §A.3. We combine the conclusions of the two theorems in the following corollary.

Corollary 4.4. *Let the Assumptions 2.1, 3.1 and 4.2 hold. Then at the limit* $n \to \infty$ *the the post-processing matrix* $\mathbf{P} \equiv \mathbf{P}(+\infty)$ *almost surely satisfies sparse recovery conditions in Definition 2.2.*

4.1. Synthetic data study

We complement our theoretical study with an experiment in a synthetic setup. Below we describe the data generation, sample manipulations, and their annotations, and provide their detailed descriptions in §B.

We generate **the latent variables** ($\mathbf{z}$) from a 10-dimensional centered normal random variable, where the coordinates have unit variance, the first two coordinates are correlated with correlation coefficient ρ and all the other cross-coordinate correlations are zero. We consider the first five

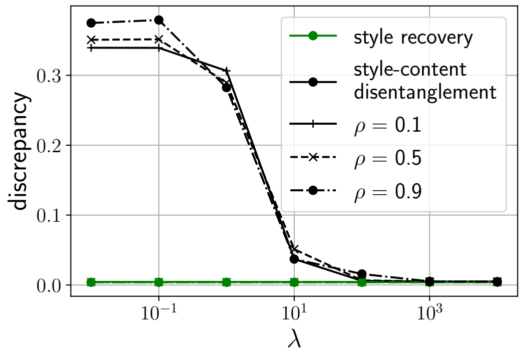


Figure 2. Plots (with error-bars over 50 repetitions) for discrepancies in style recovery ($\|\operatorname{corr}([\mathbf{z}]_{\mathbf{F}_S}, [\hat{\mathbf{z}}]_{\mathbf{F}_S}) - \operatorname{corr}([\mathbf{z}]_{\mathbf{F}_S})\|_F$) and style-content disentanglement ($\|\operatorname{corr}([\hat{\mathbf{z}}]_{\mathbf{F}_C}, [\mathbf{z}]_{\mathbf{F}_S})\|_F$) for estimated factors, where $\|\cdot\|_F$ is the normalized Frobenius norm (see Footnote 1). Here, ρ is the correlation between the first two coordinates in true factors.

coordinates of $\mathbf{z}$ as the style factors, *i.e.* $\mathbf{F}_S = \{1, 2, \dots, 5\}$, and the rest of them as content factors.

The entangled representations are $d' = 10$ dimensional vectors and which we obtain as $\mathbf{u} = \mathbf{Az} = \mathbf{LUz}$, where $\mathbf{L}$ is a 10×10 lower triangular matrix whose diagonal entries are one and off-diagonal entries are 0.9 and $\mathbf{U}$ is a randomly generated $d \times d$ orthogonal matrix.

Sample manipulations and annotations: For j-th style coordinates we obtain two manipulated samples per latent factor $\mathbf{z}$, which (denoted as $\mathbf{z}^{(j),+}$ and $\mathbf{z}^{(j),-}$) set the j-th coordinate to it's positive (resp. negative) absolute value, *i.e.* $[\mathbf{z}^{(j),+}]_j = |[\mathbf{z}]_j|$ (resp. $[\mathbf{z}^{(j),-}]_j = -|[\mathbf{z}]_j|$), and annotate it as $+1$ (resp. -1). Since the first two coordinates have correlation coefficient ρ, if either of them is changed by the value δ then the other one must be changed by $\rho\delta$. Note that one of $\mathbf{z}^{(j),+}$ and $\mathbf{z}^{(j),-}$ is exactly equal to $\mathbf{z}$. The corresponding entangled representations to the manipulated latent factors are used for recovering the style factors (as in (3.2) and (3.5)), and content factors (as in (3.3)).

Style recovery: In our synthetic experiments we validate the quality of sparse recovery for estimated latent factors on two fronts: (1) recovery in the style factors, and (2) disentanglement between style and content factors. To verify recovery in style factors we recall Theorem 4.1 that the estimated style factors ($[\hat{\mathbf{z}}]_{\mathbf{F}_S}$) approximate the true style factors ($[\mathbf{z}]_{\mathbf{F}_S}$) up to constant multiplications. This implies that the cross-correlation between estimated and true style factors ($\operatorname{corr}([\mathbf{z}]_{\mathbf{F}_S}, [\hat{\mathbf{z}}]_{\mathbf{F}_S})$) should be approximately identical to the correlation of the true style factors ($\operatorname{corr}([\mathbf{z}]_{\mathbf{F}_S})$). In Figure 2 we verify this by calculating $\|\operatorname{corr}([\mathbf{z}]_{\mathbf{F}_S}, [\hat{\mathbf{z}}]_{\mathbf{F}_S}) - \operatorname{corr}([\mathbf{z}]_{\mathbf{F}_S})\|_F$ where $\|\cdot\|_F$ is the normalized Frobenius norm of a matrix.[1] We refer to it as the *discrepancy in style recovery* and observe that it is small and not affected by ρ. Even for ρ as large as 0.9 the recovery of style factors has small discrepancy, which matches with the conclusion of Theorem 4.1. Additionally, we observe that the discrepancies are the same for different values of λ (that appears in (3.3)) since the estimation of the style factors doesn't involve λ.

Style and content disentanglement: Note that the style and content factors are uncorrelated with each other, *i.e.* $\mathrm{corr}([\mathbf{z}]_{\mathbf{F}_{\mathrm{S}}}, [\mathbf{z}]_{\mathbf{F}_{\mathrm{C}}}) = \mathbf{0}$. If the content factors and style factors are truly disentangled then the cross-correlation between estimated content factors $[\hat{\mathbf{z}}]_{\mathbf{F}_{\mathrm{C}}}$ and true style factors $[\mathbf{z}]_{\mathbf{F}_{\mathrm{S}}}$ should be approximately equal to zero. In Figure 2 we verify this by plotting $\|\,\mathrm{corr}([\hat{\mathbf{z}}]_{\mathbf{F}_{\mathrm{C}}}, [\mathbf{z}]_{\mathbf{F}_{\mathrm{S}}})\|_{\mathrm{F}}$, which we refer to as the *discrepancy in style-content disentanglement* (SCD) and notice that for large enough values of the parameter λ (*i.e.* $\lambda > 100$) the discrepancy is quite small. Though ρ has a mild effect on disentanglement between style and content factors for smaller values of λ, the effect is indistinguishable for large λ ($\lambda > 10^3$).

5. Experiments

We verify the ability of PISCO (Algorithm 1) to isolate content and style in pre-trained visual representations and the utility of the learned representations for OOD generalization when (i) train data is spuriously correlated with the style and the correlation is reversed in the test data; (ii) test data is modified with various image transformations, i.e., domain generalization with style-based distribution shifts. We consider nine transformations in our experiments: four types of image corruptions (rotation, contrast, blur, and saturation) on CIFAR-10 (Krizhevsky et al., 2009), similar to ImageNet-C (Hendrycks & Dietterich, 2018), four transformations based on style transfer (Huang & Belongie, 2017) on ImageNet (Russakovsky et al., 2015), similar to Stylized ImageNet (Geirhos et al., 2018), and a color transformation on MNIST, similar to Colored MNIST (Arjovsky et al., 2019) (see §D.1 for Colored MNIST experiment). The experiments code is available on GitHub.[2]

5.1. Transformed CIFAR

In this set of experiments, our goal is to disentangle four styles ($m = 4$) corresponding to image corruptions (rotation, contrast, blur, and saturation) from content. For feature extraction we consider a ResNet-18 (He et al., 2016) pre-trained on ImageNet (Russakovsky et al., 2015) (`Supervised`) and a SimCLR (Chen et al., 2020) trained on CIFAR-10 via self-supervision with the same architecture (`SimCLR`). For each feature extractor, we learn a *single* PISCO post-processing feature transformation matrix $\mathbf{P}(\lambda)$ as in Algorithm 1 to jointly disentangle all considered styles from content. We report results for $\lambda \in \{1, 10, 50\}$.[3]

Baselines Our main baseline is the vanilla `SimCLR` representations due to Kügelgen et al. (2021) who argued that it is sufficient for style and content disentanglement under some assumptions. Thus we study whether we can further improve style-content disentanglement in `SimCLR` in a real data setting in addition to experiments with features obtained via supervised pretraining on ImageNet. We also compare PISCO's style-content disentanglement with IP-IRM (Wang et al., 2021), which is an *in-processing* method combining self-supervised learning and invariant risk minimization (Arjovsky et al., 2019) to learn disentangled representations. We use IP-IRM model trained on CIFAR-100 provided by the authors.

We note that there are many other methods for learning disentangled representations (Wu et al. (2019); Nemeth (2020); Ren et al. (2021); Kügelgen et al. (2021), to name a few), however, they all require training an encoder-decoder model from scratch and can not take advantage of powerful feature extractors pre-trained on large datasets as in our setting. In comparison to these works, the simplicity and scalability of our method (as well as of using vanilla SimCLR features) come at a cost, i.e., we forego the ability to visualize disentanglement via controlled image generation due to the absence of a generator/decoder. Instead, we demonstrated disentanglement theoretically (§4) and verify it empirically via correlation analysis of learned style and content factors, similar to prior works that studied disentanglement in settings without a generator/decoder (Zimmermann et al., 2021; Kügelgen et al., 2021).

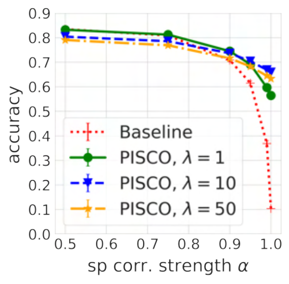
Disentanglement In Table 1 we summarize the disentanglement metrics for the smallest considered $\lambda = 1$. In the style correlation columns (Style Corr.), we report the correlation between the corresponding style value (encoded as -1 for the original images and $+1$ for the transformed ones) and the factor corresponding to style in the learned representations. None of the baselines explicitly identify style factors, thus we use the coordinate maximally correlated with the corresponding style as the style factor.

We notice that the blur style is the hardest to learn for both supervised and unsupervised representations. As we will see

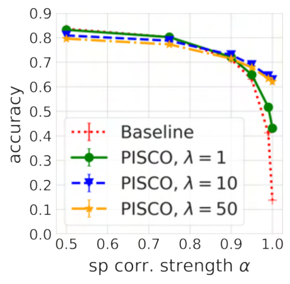
[1] The normalized Frobenius norm of a matrix $\mathbf{A} \in \mathbf{R}^{m\times n}$ is denoted as $\|\mathbf{A}\|_{\mathrm{F}}$ and defined as $\|\mathbf{A}\|_{\mathrm{F}} \triangleq \sqrt{\frac{\sum_{i,j}[\mathbf{A}]_{i,j}^2}{mn}}$.

[2] Code: github.com/liliangweta/PISCO.

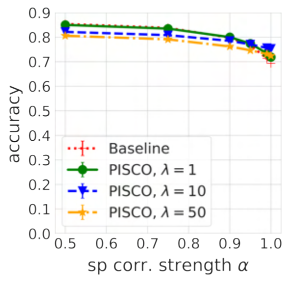
[3] In all experiments we set the number of content factors to $k = \eta d' - m$, where d' is the representation dimension. We set $\eta = 0.95$ for all experiments in the main paper and report results for other values of η in §D. As long as η is close to 1, baselines and PISCO in-distribution results are similar. For smaller values of η, PISCO in-distribution accuracy naturally deteriorates.



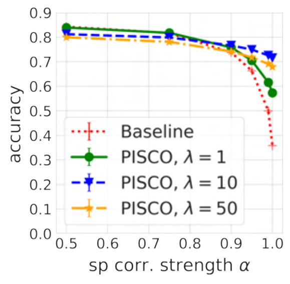
(a) Rotation - `Supervised`



(b) Contrast - `Supervised`

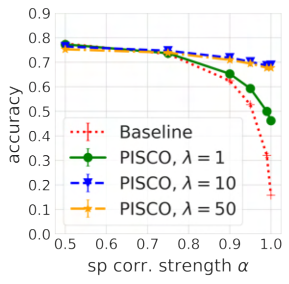


(c) Blur - `Supervised`

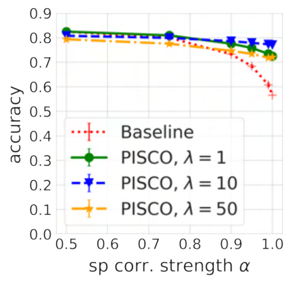


(d) Satur. - `Supervised`

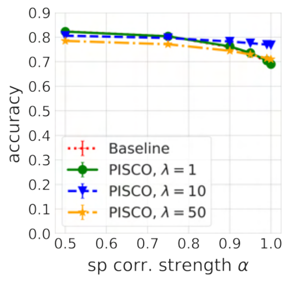
Figure 3. OOD accuracy of `Supervised` representations on CIFAR-10 where the label is spuriously correlated with the corresponding transformation. PISCO significantly improves OOD accuracy, especially in the case of rotation. Both $\lambda = 1$ and $\lambda = 10$ preserve in-distribution accuracy, while larger $\lambda = 50$ may degrade it as per (3.3).



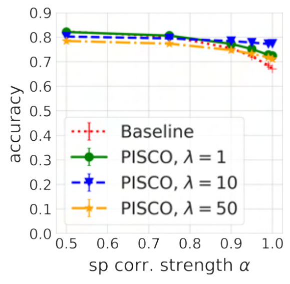
(a) Rotation - `SimCLR`



(b) Contrast - `SimCLR`



(c) Blur - `SimCLR`



(d) Saturation - `SimCLR`

Figure 4. OOD accuracy of `SimCLR` representations on CIFAR-10 where the label is spuriously correlated with the corresponding transformation. Results are analogous to Figure 3. The `SimCLR` baseline representations are less sensitive to contrast and saturation but remain sensitive to rotation.

later, both representations are fairly invariant to this style. Comparing PISCO on `Supervised` and `SimCLR`, the style recovery is better on `Supervised` since `SimCLR` representations are more robust to style changes (Kügelgen et al., 2021).

In the style-content disentanglement (SCD) columns, we report the disentanglement of style from content features as in the synthetic experiment in Figure 2. Here `SimCLR` representations appear slightly harder to disentangle using PISCO than `Supervised` representations. In the SCD of original representations for both `Supervised` and `SimCLR`, as expected, we observe that these representations are more entangled with the styles, especially the `Supervised` representations. Comparing the SCD for PISCO with that of IP-IRM, we see that PISCO can post-process popular pre-trained representations to achieve comparable or better disentanglement without re-training (i.e., in-processing). Overall we conclude that PISCO is successful in isolating style and content.

Spurious correlations Next, we create four variations of CIFAR-10 where labels are spuriously correlated with one of the four styles (image corruptions). Specifically, in the training dataset, we corrupt images from the first half of the classes with probability α and from the second half of the classes with probability $1 - \alpha$. In test data the correlation is reversed, i.e., images from the first half of the classes are corrupted with probability $1 - \alpha$ and images from the second half with probability α (see §C for details). Thus, for $\alpha = 0.5$ train and test data have the same distribution where each image is randomly transformed with the corresponding image corruption type, and $\alpha = 1$ corresponds to the extreme spurious correlation setting.

For each α we train and test a linear model on the original representations and on PISCO representations (in this and subsequent experiments all learned style factors are discarded for downstream tasks; see §C for additional details) for varying λ. Recall that here we use the same PISCO transformation matrices learned previously without knowledge of the specific corruption type and spurious correlation value α of a given dataset. We summarize results for `Supervised` features in Figure 3 and for `SimCLR` features in Figure 4. PISCO improves upon both original representations and across all transformations. For $\lambda = 1$, PISCO always preserves the in-distribution accuracy, i.e. when $\alpha = 0.5$, and improves upon the baselines in the presence of spurious correlations. Larger $\lambda = 50$ can degrade in-distribution accuracy in some cases (recall that λ controls the tradeoff between the reconstruction of the original features with the content factors and style-content disentanglement per (3.3)), while $\lambda = 10$ provides a favorable tradeoff

Table 1. Content and style disentanglement of `Supervised` and `SimCLR` representations on CIFAR-10 with PISCO. The style correlation columns (style Corr.) show correlations between styles in the data and representations corresponding to style. The isolation of style from content is measured with style-content disentanglement (SCD, see Figure 2). In the last column we compare to representations learned by IP-IRM. Bold denotes best results.

Style	Supervised				Unsupervised					
	`Supervised`		PISCO		`SimCLR`		PISCO		IP-IRM	
	Style Corr.	SCD	Style Corr.	SCD	Style Corr.	SCD	Style Corr.	SCD	Style Corr.	SCD
blur	0.319	0.090	**0.716**	**0.051**	0.304	0.096	**0.719**	0.060	0.032	**0.022**
contrast	0.490	0.243	**0.927**	**0.055**	0.094	0.076	**0.897**	**0.049**	0.419	0.188
rotation	0.746	0.212	**0.936**	**0.029**	0.368	0.182	**0.945**	**0.056**	0.617	0.114
saturation	0.641	0.204	**0.882**	**0.048**	0.120	0.071	**0.738**	0.060	0.219	**0.044**

Table 2. Accuracy of `Supervised` representations on CIFAR-10 test set in-distribution, i.e., no transformation (referred to as "none"; last row), and OOD when modified with the corresponding transformation. PISCO with $\lambda = 1$ provides significant improvements for rotation, contrast, and saturation while preserving in-distribution accuracy.

Style	Baseline (`Supervised`)	PISCO ($\lambda = 1$)	PISCO ($\lambda = 10$)	PISCO ($\lambda = 50$)
rotation	0.678	**0.737**	0.733	0.710
contrast	0.625	0.683	**0.744**	0.726
saturation	0.699	**0.758**	0.745	0.721
blur	**0.817**	**0.817**	0.793	0.775
none	**0.873**	0.870	0.844	0.826

Table 3. Accuracy of `SimCLR` representations on CIFAR-10 test set in-distribution, i.e., no transformation (referred to as "none"; last row), and OOD when modified with the corresponding transformation. `SimCLR` features are robust to the considered transformations and perform similarly to PISCO with $\lambda = 1$.

Style	Baseline (SimCLR)	PISCO ($\lambda = 1$)	PISCO ($\lambda = 10$)	PISCO ($\lambda = 50$)
rotation	0.620	0.625	**0.697**	0.696
contrast	**0.816**	0.814	0.806	0.794
saturation	**0.810**	0.806	0.789	0.774
blur	**0.808**	0.801	0.793	0.780
none	**0.828**	0.827	0.808	0.792

with a small reduction of in-distribution accuracy and large improvements when spurious correlations are present.

Comparing results across the representations, we notice that `SimCLR` features are less sensitive to image transformations as discussed previously. However, for both representations, spurious correlation with rotation causes a significant accuracy drop without PISCO post-processing.

Domain generalization To evaluate the domain generalization performance, we train a logistic regression classifier on the corresponding representation of the clean CIFAR-10 dataset and compute accuracy on the test set with every image transformed with one of the four corruptions, as well as the original test set to verify the in-distribution accuracy. Results are presented in Table 2 for `Supervised` features and in Table 3 for `SimCLR` features. We observe significant OOD accuracy gains when applying PISCO post-processing on the `Supervised` features while preserving the in-distribution accuracy for $\lambda = 1$. In this experiment, we see that `SimCLR` features are sufficiently robust and perform as well as PISCO post-processing with $\lambda = 1$. Overall we have observed that applying our method with smaller $\lambda = 1$ never hurts the performance, while it yields significant OOD accuracy gains in many settings.

5.2. Stylized ImageNet

In this experiment, we evaluate the domain generalization of PISCO on more sophisticated styles obtained via style transfer (Huang & Belongie, 2017), similar to the Stylized ImageNet (Geirhos et al., 2018) dataset. In addition, we evaluate the ability of PISCO to generalize to styles that are similar to but weren't used to fit PISCO. We used the "dog sketch" and "Picasso dog" styles to obtain PISCO transformation and evaluate on two additional similar but unseen styles, "woman sketch" and "Picasso self-portrait". See Figure 5 and §C for visualization and additional details.

As in the CIFAR-10 domain generalization experiment, the logistic regression classifier is trained on the original train images and tested on transformed test images. In Table 4 we report results for ResNet-50 features pre-trained on ImageNet (Baseline) and for the same features transformed with PISCO with $\lambda = 1$. PISCO improves OOD top-1 and top-5 accuracies across all four styles, including the unseen ones, while maintaining good in-distribution performance. We also report analogous results for another popular feature extractor, MAE-ViT-Base (He et al., 2022), in Table 5. We again observe that PISCO ($\lambda = 1$) improves top-1 and top-5 OOD performances with no degradation of the in-distribution performance. We present results for other values

Table 4. Top-1 and top-5 accuracies on 5 variations of the ImageNet test set for Baseline pre-trained ResNet-50 features and the corresponding post-processed PISCO ($\lambda = 1$) features.

Style	Baseline		PISCO	
	Top-1	Top-5	Top-1	Top-5
dog sketch	0.516	0.752	**0.546**	**0.777**
woman sketch	0.478	0.712	**0.518**	**0.752**
Picasso dog	0.445	0.686	**0.500**	**0.738**
Picasso s.-p.	0.474	0.706	**0.514**	**0.747**
none	**0.757**	**0.927**	0.749	0.921

Table 5. Top-1 and top-5 accuracies on 5 variations of the ImageNet test set for Baseline pre-trained MAE-ViT-Base features and the corresponding post-processed PISCO ($\lambda = 1$) features.

Style	Baseline		PISCO	
	Top-1	Top-5	Top-1	Top-5
dog sketch	0.530	0.749	**0.575**	**0.773**
Picasso dog	0.472	0.686	**0.519**	**0.716**
Picasso s.-p.	0.512	0.727	**0.558**	**0.752**
woman sketch	0.504	0.719	**0.550**	**0.746**
none	0.811	0.952	**0.818**	**0.953**

of λ in §D.

We note that in this experiment the *sample manipulations and annotations* required for our method (§3) were simple to obtain. We generated the styles for fitting PISCO with basic text prompts using DALL·E 2 and obtained pairs of original and transformed images using a style transfer method (Huang & Belongie, 2017). Thus, this experiment demonstrates how PISCO can be applied to improve robustness to a variety of distribution shifts in vision tasks where we have some amount of prior knowledge needed to formulate a relevant prompt to obtain a style image.

6. Conclusion

In this paper, we studied the problem of disentangling style and content of pre-trained visual representations. We presented PISCO, a simple post-processing algorithm with theoretical guarantees. In our experiments, we demonstrated that post-processing with PISCO can improve OOD performance of popular pre-trained deep models while preserving the in-distribution accuracy. Our method is computationally inexpensive and simple to implement.

In our experiments, we mainly were interested in discarding the style factors and keeping the style-invariant content factors for OOD generalization. However, we also demonstrated both theoretically and empirically that the learned style factors are representative of the presence or absence of the corresponding styles. Thus, the values of the style factors can be used to assist in outlier/OOD samples detection, or in some special cases of image retrieval, e.g., finding all images with a specific style.

One limitation of our method is the reliance on the availability of meaningful data transformations (or augmentations). While there are plenty of such transformations for images, they could be harder to identify for other data modalities. Natural language processing is one example where it is not as straightforward to define meaningful text augmentations. However, text data augmentations is also an active research area (Wei & Zou, 2019; Bayer et al., 2021; Shorten et al., 2021) which could enable applications of PISCO to NLP.

Another interesting direction to explore is extending our model to various weak supervision settings (Bouchacourt et al., 2018; Shu et al., 2019; Chen & Batmanghelich, 2020). In comparison to data augmentation functions, such forms of supervision are typically easier to obtain outside of the image domain. Thus, an extension of our model to weak supervision could enable disentanglement via post-processing for a broader class of data modalities.

Acknowledgements

This paper is based upon work supported by the National Science Foundation (NSF) under grants no. 2027737 and 2113373, and the Rensselaer-IBM AI Research Collaboration (`http://airc.rpi.edu`), part of the IBM AI Horizons Network (`http://ibm.biz/AIHorizons`).

References

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant Risk Minimization. *arXiv:1907.02893 [cs, stat]*, September 2019.

Bayer, M., Kaufhold, M.-A., and Reuter, C. A survey on data augmentation for text classification. *ACM Computing Surveys*, 2021.

Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 456–473, 2018.

Bouchacourt, D., Tomioka, R., and Nowozin, S. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32(1), 2018.

Chen, J. and Batmanghelich, K. Weakly supervised disentanglement by pairwise similarities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(04), pp. 3495–3502, 2020.

Chen, M., Wei, Z., Huang, Z., Ding, B., and Li,

Y. Simple and Deep Graph Convolutional Networks. *arXiv:2007.02133 [cs, stat]*, July 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Deldjoo, Y., Di Noia, T., Malitesta, D., and Merra, F. A. Leveraging content-style item representation for visual recommendation. In *European Conference on Information Retrieval*, pp. 84–92. Springer, 2022.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Garcia, N. and Vogiatzis, G. How to read paintings: semantic art understanding with multi-modal retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised Learning. *arXiv:2006.07733 [cs, stat]*, September 2020.

Gutmann, M. U. and Hyvärinen, A. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2), 2012.

He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.90.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Hendrycks, D. and Dietterich, T. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*, September 2018.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Hosoya, H. Group-based learning of disentangled representations with generalizability for novel contents. *arXiv preprint arXiv:1809.02383*, 2018.

Huang, X. and Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510, 2017.

Hyvärinen, A. and Morioka, H. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 3772–3780, Red Hook, NY, USA, December 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9.

Hyvärinen, A. and Pajunen, P. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

John, V., Mou, L., Bahuleyan, H., and Vechtomova, O. Disentangled representation learning for non-parallel text style transfer. *arXiv preprint arXiv:1808.04339*, 2018.

Jutten, C., Babaie-Zadeh, M., and Karhunen, J. Nonlinear mixtures. In *Handbook of Blind Source Separation*, pp. 549–592. Elsevier, 2010.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International Conference on Machine Learning*, pp. 2668–2677, July 2018.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Beery, S., Leskovec, J., Kundaje, A., Pierson, E., Levine, S., Finn, C., and Liang, P. WILDS: A Benchmark of in-the-Wild Distribution Shifts. *arXiv:2012.07421 [cs]*, December 2020.

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

Kügelgen, J., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., and Locatello, F. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021.

Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 35–51, 2018.

Locatello, F., Abbati, G., Rainforth, T., Bauer, S., Schölkopf, B., and Bachem, O. On the fairness of disentangled representations. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 1309, pp. 14611–14624. Curran Associates Inc., Red Hook, NY, USA, December 2019a.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 4114–4124. PMLR, May 2019b.

Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., and Tschannen, M. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pp. 6348–6359. PMLR, 2020.

Ma, J., Zhou, C., Cui, P., Yang, H., and Zhu, W. Learning disentangled representations for recommendation. *Advances in neural information processing systems*, 32, 2019.

Nemeth, J. Adversarial disentanglement with grouped observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10243–10250, 2020.

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

Petersen, F., Mukherjee, D., Sun, Y., and Yurochkin, M. Post-processing for individual fairness. *Advances in Neural Information Processing Systems*, 34:25944–25955, 2021.

Ren, X., Yang, T., Wang, Y., and Zeng, W. Rethinking content and style: exploring bias for unsupervised disentanglement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1823–1832, 2021.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Ruta, D., Motiian, S., Faieta, B., Lin, Z., Jin, H., Filipkowski, A., Gilbert, A., and Collomosse, J. Aladin: all layer adaptive instance normalization for fine-grained style similarity. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11926–11935, 2021.

Ruta, D., Gilbert, A., Aggarwal, P., Marri, N., Kale, A., Briggs, J., Speed, C., Jin, H., Faieta, B., Filipkowski, A., et al. Stylebabel: Artistic style tagging and captioning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, pp. 219–236. Springer, 2022.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization. *arXiv:1911.08731 [cs, stat]*, November 2019.

Saleh, B. and Elgammal, A. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *arXiv preprint arXiv:1505.00855*, 2015.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proceedings of the IEEE*, 109(5): 612–634, 2021.

Shorten, C., Khoshgoftaar, T. M., and Furht, B. Text data augmentation for deep learning. *Journal of big Data*, 8 (1):1–34, 2021.

Shu, R., Chen, Y., Kumar, A., Ermon, S., and Poole, B. Weakly supervised disentanglement with guarantees. *arXiv preprint arXiv:1910.09772*, 2019.

Strubell, E., Ganesh, A., and McCallum, A. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1355.

Wang, T., Yue, Z., Huang, J., Sun, Q., and Zhang, H. Self-supervised learning disentangled group representation as feature. *Advances in Neural Information Processing Systems*, 34:18225–18240, 2021.

Wei, D., Ramamurthy, K. N., and Calmon, F. d. P. Optimized Score Transformation for Fair Classification. *arXiv:1906.00066 [cs, math, stat]*, December 2019.

Wei, J. and Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

Wu, T. T., Chen, Y. F., Hastie, T., Sobel, E., and Lange, K. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, March 2009. ISSN 1460-2059, 1367-4803. doi: 10.1093/bioinformatics/btp041.

Wu, W., Cao, K., Li, C., Qian, C., and Loy, C. C. Disentangling content and style via unsupervised geometry distillation. *arXiv preprint arXiv:1905.04538*, 2019.

Zimmermann, R. S., Sharma, Y., Schneider, S., Bethge, M., and Brendel, W. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pp. 12979–12990. PMLR, 2021.

A. Supplementary proofs

A.1. Proof of Corollary 2.3

Proof. We denote $\hat{\mathbf{z}}_S \triangleq [\hat{\mathbf{z}}]_{\mathbf{F}_\mathrm{S}}$, $\mathbf{z}_S \triangleq [\mathbf{z}]_{\mathbf{F}_\mathrm{S}}$, $\Sigma \triangleq \mathrm{cov}(\mathbf{z})$ and Δ as the diagonal matrix of Σ. Notice that

$$\mathrm{corr}(\mathbf{z}) = \Delta^{-1/2}\Sigma\Delta^{-1/2}\,. \tag{A.1}$$

From Definition 2.2 $\hat{\mathbf{z}}_S = [\mathbf{PA}]_{\mathbf{F}_\mathrm{S},\mathbf{F}_\mathrm{S}}\mathbf{z}_S$ where $[\mathbf{PA}]_{\mathbf{F}_\mathrm{S},\mathbf{F}_\mathrm{S}}$ is a diagonal matrix. We denote $[\mathbf{PA}]_{\mathbf{F}_\mathrm{S},\mathbf{F}_\mathrm{S}}$ as $\mathbf{D}$. Then the covariance matrix of $\hat{\mathbf{z}}_S$ is

$$\mathrm{cov}(\hat{\mathbf{z}}_S) = \mathrm{cov}(\mathbf{D}\mathbf{z}_S) = \mathbf{D}\Sigma\mathbf{D} \tag{A.2}$$

and its diagonal matrix is

$$\begin{aligned}\mathrm{diag}(\mathbf{D}\Sigma\mathbf{D}) &= \mathbf{D}\,\mathrm{diag}(\Sigma)\mathbf{D}\\ &= \mathbf{D}\Delta\mathbf{D}\\ &= \Delta\mathbf{D}^2\,,\end{aligned} \tag{A.3}$$

where the last equality is obtained using the fact that the matrix multiplication of the diagonal matrices is commuting. Expressing $\mathrm{corr}(\hat{\mathbf{z}}_S)$ in terms of $\mathrm{cov}(\hat{\mathbf{z}}_S)$ and it's diagonal matrix we obtain

$$\begin{aligned}&\mathrm{corr}(\hat{\mathbf{z}}_S)\\ &= \mathrm{diag}\left\{\mathrm{corr}(\hat{\mathbf{z}}_S)\right\}^{-1/2}\mathrm{corr}(\hat{\mathbf{z}}_S)\,\mathrm{diag}\left\{\mathrm{corr}(\hat{\mathbf{z}}_S)\right\}^{-1/2}\\ &= \{\Delta\mathbf{D}^2\}^{-1/2}\mathbf{D}\Sigma\mathbf{D}\{\Delta\mathbf{D}^2\}^{-1/2}\\ &= \Delta^{-1/2}\mathbf{D}^{-1}\mathbf{D}\Sigma\mathbf{D}\mathbf{D}^{-1}\Delta^{-1/2}\\ &= \Delta^{-1/2}\Sigma\Delta^{-1/2} = \mathrm{corr}(\mathbf{z}_S)\end{aligned} \tag{A.4}$$

and we obtain (2.1). □

A.2. Proof of Theorem 4.1

Proof. The closed form of $\hat{\mathbf{p}}_j$ in (3.2) can be written as:

$$\hat{\mathbf{p}}_j = \hat{\Sigma}_\mathbf{u}^\dagger\frac{1}{2n}\sum_{i=1}^n\Big[(\mathbf{u}_i-\bar{\mathbf{u}})(\mathbf{y}_i^{(j)}-\bar{\mathbf{y}})+(\mathbf{u}_i^{(j)}-\bar{\mathbf{u}})(\tilde{\mathbf{y}}_i^{(j)}-\bar{\mathbf{y}})\Big] \tag{A.5}$$

where $\bar{\mathbf{u}} = \frac{1}{2n}\sum_{i=1}^n\big[\mathbf{u}_i+\mathbf{u}_i^{(j)}\big]$, $\hat{\Sigma}_\mathbf{u} = \frac{1}{2n}\sum_{i=1}^n\big[\mathbf{u}_i\mathbf{u}_i^\top + \mathbf{u}_i^{(j)}\{\mathbf{u}_i^{(j)}\}^\top\big] - \bar{\mathbf{u}}\bar{\mathbf{u}}^\top$, $\hat{\Sigma}_\mathbf{u}^\dagger$ is the Moore-Penrose inverse of $\hat{\Sigma}_\mathbf{u}$, $\bar{\mathbf{y}} = \frac{1}{2n}\sum_{i=1}^n\big[\mathbf{y}_i^{(j)}+\tilde{\mathbf{y}}_i^{(j)}\big]$, and $\bar{\epsilon}^{(j)} = \frac{1}{2n}\sum_{i=1}^n\big[\epsilon_i^{(j)}+\tilde{\epsilon}_i^{(j)}\big]$. Here, defining $\bar{\mathbf{z}} = \frac{1}{2n}\sum_{i=1}^n\big[\mathbf{z}_i+\mathbf{z}_i^{(j)}\big]$ and $\hat{\Sigma}_\mathbf{z} = \frac{1}{2n}\sum_{i=1}^n\big[\mathbf{z}_i\mathbf{z}_i^\top+\mathbf{z}_i^{(j)}\{\mathbf{z}_i^{(j)}\}^\top\big] - \bar{\mathbf{z}}\bar{\mathbf{z}}^\top$ we notice the following.

$$\mathbf{y}_i^{(j)} - \bar{\mathbf{y}} = \beta_j\big([\mathbf{z}_i]_j - [\bar{\mathbf{z}}]_j\big) + (\epsilon_i^{(j)} - \bar{\epsilon}^{(j)}) \tag{A.6}$$

$$\tilde{\mathbf{y}}_i^{(j)} - \bar{\mathbf{y}} = \beta_j\big([\mathbf{z}_i^{(j)}]_j - [\bar{\mathbf{z}}]_j\big) + (\tilde{\epsilon}_i^{(j)} - \bar{\epsilon}^{(j)}) \tag{A.7}$$

$$\hat{\Sigma}_\mathbf{u} = \mathbf{A}\hat{\Sigma}_\mathbf{z}\mathbf{A}^\top \tag{A.8}$$

and hence

$$\begin{aligned}&\frac{1}{2n}\sum_{i=1}^n\Big[(\mathbf{u}_i-\bar{\mathbf{u}})(\mathbf{y}_i^{(j)}-\bar{\mathbf{y}})+(\mathbf{u}_i^{(j)}-\bar{\mathbf{u}})(\tilde{\mathbf{y}}_i^{(j)}-\bar{\mathbf{y}})\Big]\\ &= \frac{1}{2n}\sum_{i=1}^n\Big[\mathbf{A}(\mathbf{z}_i-\bar{\mathbf{z}})\beta_j\big([\mathbf{z}_i]_j-[\bar{\mathbf{z}}]_j\big)\\ &\qquad + \mathbf{A}(\mathbf{z}_i^{(j)}-\bar{\mathbf{z}})\beta_j\big([\mathbf{z}_i^{(j)}]_j-[\bar{\mathbf{z}}]_j\big)\Big]\\ &\quad + \frac{1}{2n}\sum_{i=1}^n\Big[\mathbf{A}(\mathbf{z}_i-\bar{\mathbf{z}})(\epsilon_i^{(j)}-\bar{\epsilon}^{(j)})\\ &\qquad + \mathbf{A}(\mathbf{z}_i^{(j)}-\bar{\mathbf{z}})(\tilde{\epsilon}_i^{(j)}-\bar{\epsilon}^{(j)})\Big]\\ &\triangleq \mathrm{cov}_1 + \mathrm{cov}_2\,,\end{aligned} \tag{A.9}$$

where

$$\begin{aligned}\mathrm{cov}_1 &\triangleq \frac{1}{2n}\sum_{i=1}^n\Big[\mathbf{A}(\mathbf{z}_i-\bar{\mathbf{z}})\beta_j\big([\mathbf{z}_i]_j-[\bar{\mathbf{z}}]_j\big)\\ &\qquad + \mathbf{A}(\mathbf{z}_i^{(j)}-\bar{\mathbf{z}})\beta_j\big([\mathbf{z}_i^{(j)}]_j-[\bar{\mathbf{z}}]_j\big)\Big]\\ &= \beta_j\mathbf{A}\frac{1}{2n}\sum_{i=1}^n\Big[(\mathbf{z}_i-\bar{\mathbf{z}})(\mathbf{z}_i-\bar{\mathbf{z}})^\top e_j\\ &\qquad + (\mathbf{z}_i^{(j)}-\bar{\mathbf{z}})(\mathbf{z}_i^{(j)}-\bar{\mathbf{z}})^\top e_j\Big]\\ &= \beta_j\mathbf{A}\hat{\Sigma}_\mathbf{z}e_j\,.\end{aligned}$$

and with $\hat{\Sigma}_{\mathbf{z},\epsilon} \triangleq \frac{1}{2n}\sum_{i=1}^n\Big[(\mathbf{z}_i-\bar{\mathbf{z}})(\epsilon_i^{(j)}-\bar{\epsilon}^{(j)}) + (\mathbf{z}_i^{(j)}-\bar{\mathbf{z}})(\tilde{\epsilon}_i^{(j)}-\bar{\epsilon}^{(j)})\Big]$ we obtain

$$\begin{aligned}\mathrm{cov}_2 &\triangleq \frac{1}{2n}\sum_{i=1}^n\Big[\mathbf{A}(\mathbf{z}_i-\bar{\mathbf{z}})(\epsilon_i^{(j)}-\bar{\epsilon}^{(j)})\\ &\qquad + \mathbf{A}(\mathbf{z}_i^{(j)}-\bar{\mathbf{z}})(\tilde{\epsilon}_i^{(j)}-\bar{\epsilon}^{(j)})\Big]\\ &= \mathbf{A}\hat{\Sigma}_{\mathbf{z},\epsilon}\end{aligned}$$

Using the identities (A.8) and (A.9) we rewrite $\hat{\mathbf{p}}_j$ as

$$\begin{aligned}\hat{\mathbf{p}}_j &= (\mathbf{A}\hat{\Sigma}_\mathbf{z}\mathbf{A}^\top)^\dagger\{\beta_j\mathbf{A}\hat{\Sigma}_\mathbf{z}e_j + \mathbf{A}\hat{\Sigma}_{\mathbf{z},\epsilon}\}\\ &= (\mathbf{A}\hat{\Sigma}_\mathbf{z}\mathbf{A}^\top)^\dagger\beta_j\mathbf{A}\hat{\Sigma}_\mathbf{z}e_j + (\mathbf{A}\hat{\Sigma}_\mathbf{z}\mathbf{A}^\top)^\dagger\mathbf{A}\hat{\Sigma}_{\mathbf{z},\epsilon}\\ &\triangleq \hat{\mathbf{p}}_j^{(1)} + \hat{\mathbf{p}}_j^{(2)}\,,\end{aligned} \tag{A.10}$$

where $\hat{\mathbf{p}}_j^{(1)} \triangleq (\mathbf{A}\hat{\Sigma}_\mathbf{z}\mathbf{A}^\top)^\dagger\beta_j\mathbf{A}\hat{\Sigma}_\mathbf{z}e_j$ and $\hat{\mathbf{p}}_j^{(2)} \triangleq (\mathbf{A}\hat{\Sigma}_\mathbf{z}\mathbf{A}^\top)^\dagger\mathbf{A}\hat{\Sigma}_{\mathbf{z},\epsilon}$.

Since $n \geq d+1$ we notice that the covariance matrix $\hat{\Sigma}_\mathbf{z}$ is invertible. This fact combined with left invertibility of $\mathbf{A}$ implies that the matrix $\mathbf{A}\hat{\Sigma}_\mathbf{z}^{1/2}$ is also left invertible. We recall the property of Moore-Penrose inverse that for any left invertible matrix $\mathbf{G}$ it holds:

$$(\mathbf{G}\mathbf{G}^\top)^\dagger\mathbf{G} = \mathbf{G}(\mathbf{G}^\top\mathbf{G})^{-1}\,.$$

Letting $\mathbf{G} = \mathbf{A}\hat{\Sigma}_{\mathbf{z}}^{1/2}$ and using the property in (A.10) we obtain

$$\begin{aligned}\hat{\mathbf{p}}_j^{(1)} &= \beta_j(\mathbf{A}\hat{\Sigma}_{\mathbf{z}}\mathbf{A}^\top)^\dagger\mathbf{A}\hat{\Sigma}_{\mathbf{z}}e_j \\ &= \beta_j(\mathbf{G}\mathbf{G}^\top)^\dagger\mathbf{G}\hat{\Sigma}_{\mathbf{z}}^{1/2}e_j \\ &= \beta_j\mathbf{G}(\mathbf{G}^\top\mathbf{G})^{-1}\hat{\Sigma}_{\mathbf{z}}^{1/2}e_j \\ &= \beta_j\mathbf{A}\Sigma_{\mathbf{z}}^{1/2}\left(\Sigma_{\mathbf{z}}^{1/2}\mathbf{A}^\top\mathbf{A}\Sigma_{\mathbf{z}}^{1/2}\right)^{-1}\hat{\Sigma}_{\mathbf{z}}^{1/2}e_j \\ &= \beta_j\mathbf{A}\Sigma_{\mathbf{z}}^{1/2}\Sigma_{\mathbf{z}}^{-1/2}(\mathbf{A}^\top\mathbf{A})^{-1}\Sigma_{\mathbf{z}}^{-1/2}\hat{\Sigma}_{\mathbf{z}}^{1/2}e_j \\ &= \beta_j\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1}e_j\,.\end{aligned} \tag{A.11}$$

Hence, we notice that

$$\begin{aligned}\mathbf{A}^\top\hat{\mathbf{p}}_j^{(1)} &= \mathbf{A}^\top\hat{\beta}_j\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1}e_j \\ &= \beta_j e_j\,.\end{aligned}$$

Repeating same calculation as above we obtain

$$\begin{aligned}\mathbf{A}^\top\hat{\mathbf{p}}_j^{(2)} &= \mathbf{A}^\top(\mathbf{A}\hat{\Sigma}_{\mathbf{z}}\mathbf{A}^\top)^\dagger\mathbf{A}\hat{\Sigma}_{\mathbf{z},\epsilon} \\ &= \hat{\Sigma}_{\mathbf{z}}^{-1}\hat{\Sigma}_{\mathbf{z},\epsilon}\,.\end{aligned}$$

From Assumption 3.1 we recall that $(\epsilon^{(j)}, \tilde{\epsilon}^{(j)})$ and $(\mathbf{z}, \mathbf{z}^{(j)})$ are uncorrelated and hence

$$\hat{\Sigma}_{\mathbf{z},\epsilon} \xrightarrow{\text{a.s.}} \mathbf{0}\,.$$

Since $\Sigma_{\mathbf{z}}$ is invertible we obtain that

$$\mathbf{A}^\top\hat{\mathbf{p}}_j^{(2)} \xrightarrow{\text{a.s.}} \mathbf{0}\,,$$

and

$$\mathbf{A}^\top\hat{\mathbf{p}}_j \xrightarrow{\text{a.s.}} \beta_j e_j$$

almost surely. Noticing that $\hat{\mathbf{p}}_j$ is the j-th row of $\mathbf{P}$ we conclude that at almost sure limit it holds: (1) $[\mathbf{PA}]_{\mathbf{F}_\mathrm{S},\mathbf{F}_\mathrm{S}}$ converges to a diagonal matrix, and (2) $[\mathbf{PA}]_{\mathbf{F}_\mathrm{S},\mathbf{F}_\mathrm{C}} \xrightarrow{\text{a.s.}} \mathbf{0}$.

□

A.3. Proof of Theorem 4.3

We divide the proof in two steps which are stated as lemmas.

Lemma A.1. *With probability at least $1-\kappa^n$ (κ is defined in Theorem 4.3) the following holds:*

$$\hat{\mathbf{Q}}(+\infty)[\mathbf{A}]_{\cdot,\mathbf{F}_S} = \mathbf{0}\,.$$

Proof of Lemma A.1. At $\lambda \to \infty$ it necessarily holds:

$$\frac{1}{m}\sum_{j\in\mathcal{S}}\mathrm{tr}\left[\hat{\mathbf{Q}}(+\infty)^\top\hat{\mathbf{Q}}(+\infty)\Big(\frac{\Delta_j^\top\Delta_j}{n}\Big)\right] = 0$$

which equivalently means for every $j \in \mathcal{S}$:

$$\mathrm{tr}\left[\hat{\mathbf{Q}}(+\infty)^\top\hat{\mathbf{Q}}(+\infty)\Big(\frac{\Delta_j^\top\Delta_j}{n}\Big)\right] = 0\,. \tag{A.12}$$

Combining (3.4) and the above we obtain

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^n\|\hat{\mathbf{Q}}(+\infty)(\mathbf{u}_i - \mathbf{u}_i^{(j)})\|_2^2 \\ &= \mathrm{tr}\left[\hat{\mathbf{Q}}(+\infty)^\top\hat{\mathbf{Q}}(+\infty)\Big(\frac{\Delta_j^\top\Delta_j}{n}\Big)\right] = 0\end{aligned}$$

which implies that for each $i \in [n]$ and $j \in \mathbf{F}_\mathrm{S}$

$$\begin{aligned}\mathbf{0} &= \hat{\mathbf{Q}}(+\infty)(\mathbf{u}_i - \mathbf{u}_i^{(j)}) \\ &= \hat{\mathbf{Q}}(+\infty)\mathbf{A}(\mathbf{z}_i - \mathbf{z}_i^{(j)})\,,\end{aligned} \tag{A.13}$$

where the second equality follows from Assumption 2.1. Since the latent factors $\{\mathbf{z}_i\}_{i=1}^n$ were drawn independently from the distribution $\mathbb{P}_{\mathbf{z}}$, we conclude that one of the samples is in the event $\mathbf{E}$ with probability at least $1-\kappa^n$. Denote the sample as $\mathbf{z}_0$. Then defining $\mathbf{Z}_0 = [\mathbf{z}_0 - \mathbf{z}_0^{(1)}, \ldots, \mathbf{z}_0 - \mathbf{z}_0^{(m)}]$ we notice the following: (1) from (3.1) in Assumption 3.1 it follows $[\mathbf{Z}_0]_{\mathbf{F}_\mathrm{C},\cdot} = \mathbf{0}$, and (2) from the Assumption 4.2 we see that the matrix $[\mathbf{Z}_0]_{\mathbf{F}_\mathrm{S},\cdot}$ is invertible and hence we obtain

$$\begin{aligned}\mathbf{0} &= \hat{\mathbf{Q}}(+\infty)\mathbf{A}[\mathbf{z}_0 - \mathbf{z}_0^{(1)}, \ldots, \mathbf{z}_0 - \mathbf{z}_0^{(m)}] \\ &= \hat{\mathbf{Q}}(+\infty)\begin{bmatrix}[\mathbf{A}]_{\cdot,\mathbf{F}_\mathrm{S}} & [\mathbf{A}]_{\cdot,\mathbf{F}_\mathrm{C}}\end{bmatrix}\cdot\begin{bmatrix}[\mathbf{Z}_0]_{\mathbf{F}_\mathrm{S},\cdot} \\ [\mathbf{Z}_0]_{\mathbf{F}_\mathrm{C},\cdot}\end{bmatrix} \\ &= \hat{\mathbf{Q}}(+\infty)\begin{bmatrix}[\mathbf{A}]_{\cdot,\mathbf{F}_\mathrm{S}} & [\mathbf{A}]_{\cdot,\mathbf{F}_\mathrm{C}}\end{bmatrix}\cdot\begin{bmatrix}[\mathbf{Z}_0]_{\mathbf{F}_\mathrm{S},\cdot} \\ \mathbf{0}\end{bmatrix} \\ &= \hat{\mathbf{Q}}(+\infty)[\mathbf{A}]_{\cdot,\mathbf{F}_\mathrm{S}}[\mathbf{Z}_0]_{\mathbf{F}_\mathrm{S},\cdot}\,,\end{aligned}$$

where using invertibility of $[\mathbf{Z}_0]_{\mathbf{F}_\mathrm{S},\cdot}$ we conclude

$$\hat{\mathbf{Q}}(+\infty)[\mathbf{A}]_{\cdot,\mathbf{F}_\mathrm{S}} = \mathbf{0}$$

and the lemma.

□

Lemma A.2. *Let $\mathbf{H}^\perp \in \mathbf{R}^{d'\times d'}$ be the orthogonal projector onto $\mathrm{span}\{[\mathbf{A}]_{\cdot,j} : j \in \mathbf{F}_S\}^\perp$. Then the matrix $\mathbf{H}^\perp\frac{\mathbf{U}^\top\mathbf{U}}{(m+1)n}\mathbf{H}^\perp$ has exactly $(d-m)$ many positive eigenvalues and $\mathbf{Q}(+\infty)$ is the collection of the eigen-vectors corresponding to them. Furthermore, $[\mathbf{P}(+\infty)\mathbf{A}]_{\mathbf{F}_C,\mathbf{F}_C}$ is invertible.*

Proof. We start by noticing that $\frac{\mathbf{U}^\top\mathbf{U}}{(m+1)n} = \mathbf{A}\Sigma'_{\mathbf{z}}\mathbf{A}^\top$ where the matrix

$$\Sigma'_{\mathbf{z}} = \frac{1}{(m+1)n}\sum_{i=1}^n\left[\mathbf{z}_i\mathbf{z}_i^\top + \sum_{j\in\mathcal{S}}\mathbf{z}_i^{(j)}\{\mathbf{z}_i^{(j)}\}^\top\right]$$

is invertible since $n \geq d+1$. Without loss of generality we assume that $\mathbf{F}_\mathrm{S} = [m]$. Denoting $\mathcal{C}_{\mathbf{A}} \triangleq \text{col-space}(\mathbf{A})$ we notice that for any $\mathbf{x} \in \mathcal{C}_{\mathbf{A}}^{\perp}$

$$\begin{aligned} &\mathbf{H}^{\perp}\frac{\mathbf{U}^{\top}\mathbf{U}}{(m+1)n}\mathbf{H}^{\perp}\mathbf{x} \\ &= \mathbf{H}^{\perp}\mathbf{A}\Sigma_{\mathbf{z}}'\mathbf{A}^{\top}\mathbf{x}, \text{ since } \mathbf{x} \in \mathcal{C}_{\mathbf{A}}^{\perp} \subset \operatorname{span}\{[\mathbf{A}]_{\cdot,\mathbf{F}_S}\}^{\perp} \\ &= \mathbf{0}, \text{ since } \mathbf{x} \in \mathcal{C}_{\mathbf{A}}^{\perp}. \end{aligned} \tag{A.14}$$

and for any $\mathbf{x} \in \operatorname{span}\{[\mathbf{A}]_{\cdot,j} : j \in \mathbf{F}_S\}$ it holds

$$\mathbf{H}^{\perp}\frac{\mathbf{U}^{\top}\mathbf{U}}{(m+1)n}\mathbf{H}^{\perp}\mathbf{x} = \mathbf{0}. \tag{A.15}$$

Since $\dim\{\mathcal{C}_{\mathbf{A}}^{\perp}\} = d' - d$ and $\dim\{[\mathbf{A}]_{\cdot,j} : j \in \mathbf{F}_S\} = m$, counting the degrees of freedom we obtain that $\mathbf{H}^{\perp}\frac{\mathbf{U}^{\top}\mathbf{U}}{(m+1)n}\mathbf{H}^{\perp}$ doesn't have rank more than $d' - (d' - d) - m = d - m$. Expressing $\mathbf{A}$ as

$$\mathbf{A} = \tilde{\mathbf{A}}\mathbf{U}, \tag{A.16}$$

where $\tilde{\mathbf{A}} \in \mathbf{R}^{d' \times d}$ is an orthogonal matrix and $\mathbf{U} \in \mathbf{R}^{d \times d}$ is an upper triangular such that $\operatorname{span}\{[\mathbf{A}]_{\cdot,j} : j \in \mathbf{F}_S\} = \operatorname{span}\{[\tilde{\mathbf{A}}]_{\cdot,j} : j \in \mathbf{F}_S\}$. Such a decomposition can easily be obtained from Gram-Schmidt orthogonalization of the columns of $\mathbf{A}$. Since $[\tilde{\mathbf{A}}]_{\cdot,j} \in \operatorname{span}\{[\mathbf{A}]_{\cdot,j} : j \in \mathbf{F}_S\}$ we notice that

$$\mathbf{H}^{\perp}\tilde{\mathbf{A}} = \begin{bmatrix}\mathbf{0}_{d' \times m} & [\tilde{\mathbf{A}}]_{\cdot,\mathbf{F}_C}\end{bmatrix} \tag{A.17}$$

and defining $\mathbf{M} \triangleq \mathbf{U}\Sigma_{\mathbf{z}}'\mathbf{U}^{\top}$ which is an invertible matrix we notice that

$$\mathbf{H}^{\perp}\frac{\mathbf{U}^{\top}\mathbf{U}}{(m+1)n}\mathbf{H}^{\perp} = [\tilde{\mathbf{A}}]_{\cdot,\mathbf{F}_C}[\mathbf{M}]_{\mathbf{F}_C,\mathbf{F}_C}[\tilde{\mathbf{A}}]_{\cdot,\mathbf{F}_C}^{\top}. \tag{A.18}$$

and hence it has rank $|\mathbf{F}_C| = d - m$. This concludes a part of the lemma.

Here, $[\mathbf{M}]_{\mathbf{F}_C,\mathbf{F}_C}$ is a partition matrix of the non-negative definite matrix $\mathbf{M} \triangleq \mathbf{U}\Sigma_{\mathbf{z}}'\mathbf{U}^{\top}$. We consider it's spectral decomposition

$$[\mathbf{M}]_{\mathbf{F}_C,\mathbf{F}_C} = \mathbf{W}\mathbf{D}\mathbf{W}^{\top} \tag{A.19}$$

where $\mathbf{W} \in \mathbf{R}^{(d-m)\times(d-m)}$ is an orthogonal matrix and $\mathbf{D} \in \mathbf{R}^{(d-m)\times(d-m)}$ is diagonal with positive diagonal entries (since $[\mathbf{M}]_{\mathbf{F}_C,\mathbf{F}_C}$ is full rank). This follows,

$$\mathbf{H}^{\perp}\tilde{\mathbf{A}} = [\tilde{\mathbf{A}}]_{\cdot,\mathbf{F}_C}\mathbf{W}\mathbf{D}\mathbf{W}^{\top}[\tilde{\mathbf{A}}]_{\cdot,\mathbf{F}_C}^{\top} \tag{A.20}$$

where $[\tilde{\mathbf{A}}]_{\cdot,\mathbf{F}_C}\mathbf{W}$ is again a $\mathbf{R}^{d' \times (d-m)}$ orthogonal matrix whose columns are the only eigen-vectors of $\mathbf{H}^{\perp}\frac{\mathbf{U}^{\top}\mathbf{U}}{(m+1)n}\mathbf{H}^{\perp}$ with positive eigen-values. Hence,

$$\mathbf{Q}(+\infty) = [\tilde{\mathbf{A}}]_{\cdot,\mathbf{F}_C}\mathbf{W}. \tag{A.21}$$

Since $[\mathbf{P}(+\infty)]_{\mathbf{F}_C,\cdot} = \mathbf{Q}(+\infty)^{\top}$ we obtain

$$[\mathbf{P}(+\infty)\mathbf{A}]_{\mathbf{F}_C,\cdot} = \mathbf{Q}(+\infty)^{\top}\mathbf{A}$$

where using (A.16) and (A.21) we obtain

$$\begin{aligned} &[\mathbf{P}(+\infty)\mathbf{A}]_{\mathbf{F}_C,\cdot} \\ &= \mathbf{W}^{\top}\{[\tilde{\mathbf{A}}]_{\cdot,\mathbf{F}_C}\}^{\top}\tilde{\mathbf{A}}\mathbf{U} \\ &= \mathbf{W}^{\top}\begin{bmatrix}\mathbf{0}_{(d-m)\times d} & \mathbf{I}_{(d-m)\times(d-m)}\end{bmatrix}\mathbf{U} \\ &= \mathbf{W}^{\top}\mathbf{U}_{\mathbf{F}_C,\cdot}. \end{aligned}$$

Finally we obtain

$$[\mathbf{P}(+\infty)\mathbf{A}]_{\mathbf{F}_C,\mathbf{F}_C} = \mathbf{W}^{\top}\mathbf{U}_{\mathbf{F}_C,\mathbf{F}_C}.$$

where, following that $\mathbf{W}$ is an orthogonal matrix and $\mathbf{U}$ an invertible upper-triangular matrix, both $\mathbf{W}$ and $\mathbf{U}_{\mathbf{F}_C,\mathbf{F}_C}$ are invertible. This implies $[\mathbf{P}(+\infty)\mathbf{A}]_{\mathbf{F}_C,\mathbf{F}_C}$ is invertible, and we conclude the lemma. □

A.4. Proof of Corollary 4.4

Proof. Note that the convergences in Theorem 4.1 are almost sure convergences. For each $j \in \mathbf{F}_S$ we define $\mathbf{B}_j$ as the probability one event on which $\mathbf{A}^{\top}\hat{\mathbf{p}}_j \to \beta_j e_j$. We further define $\mathbf{B} \triangleq \cap_{j\in\mathbf{F}_S}\mathbf{B}_j$ which is again a probability one event (an intersection of finitely many probability one events), and on the event the convergences hold simultaneously over $j \in \mathbf{F}_S$.

Drawing our attention to the conclusions in Theorem 4.3, we define $\mathbf{C}_n$ as the event that

$$\begin{aligned} \mathbf{C}_n \triangleq \{&\text{The conclusions in Theorem 4.3 hold} \\ &\text{with sample size } n\} \end{aligned}$$

for each $n \geq d + 1$ and notice that $\mathbb{P}_{\mathbf{z}}(\mathbf{C}_n) \geq 1 - \kappa^n$. This implies

$$\sum_{n\geq d+1}\mathbb{P}_{\mathbf{z}}(\mathbf{C}_n^c) \leq \sum_{n\geq d+1}\kappa^n < \infty.$$

We define $\mathbf{C} = \{\mathbf{C}_n^c \text{ holds infinitely often}\}$ and use the first Borel-Cantelli lemma to conclude that

$$\mathbb{P}_{\mathbf{z}}(\mathbf{C}^c) = 0 \text{ or, } \mathbb{P}_{\mathbf{z}}(\mathbf{C}) = 1.$$

Note that the event $\mathbf{C}$ is the same as the event that $\{\mathbf{C}_n$ holds all but finitely often$\}$, or that $\{$The conclusions in Theorem 4.3 holds at $n \to \infty$ $\}$, which are probability one events. Thus it follows that $\mathbf{B} \cap \mathbf{C}$, an event on which the conclusions in both the theorems 4.1 and 4.3 simultaneously hold (for $n \to \infty$), is a probability one event. Hence, we conclude that with probability one the following hold at the limit $n \to \infty$: (1) $[\mathbf{PA}]_{\mathbf{F}_S,\mathbf{F}_S}$ is a diagonal matrix, (2) $[\mathbf{PA}]_{\mathbf{F}_S,\mathbf{F}_C} = \mathbf{0}$, (3) $[\mathbf{PA}]_{\mathbf{F}_S,\mathbf{F}_C} = \mathbf{0}$, and (4) $[\mathbf{PA}]_{\mathbf{F}_C,\mathbf{F}_C}$ is invertible. These are the exact conditions in Definition 2.2 that are required for sparse recovery. Hence, the corollary follows. □

B. Details for synthetic data study in §4.1

B.1. The latent factors

The latent factors are generated as

$$\mathbf{R}^{10} \ni \mathbf{z} \sim \mathbf{N}(\mathbf{0}, \Sigma), \tag{B.1}$$

where Σ is a 10×10 covariance matrix whose entries are described below. For $i, j \in \{1, \ldots, 10\}$

$$[\Sigma]_{i,j} = \begin{cases} 1 & i = j, \\ \rho \in [0, 1) & (i,j) = (1,2), \text{ or, } (i,j) = (2,1), \\ 0 & \text{otherwise}. \end{cases} \tag{B.2}$$

We fix the first five coordinates as style factors, *i.e.* $\mathbf{F}_\mathrm{S} = \{1, \ldots, 5\}$ and the rest of them as content factors, *i.e.* $\mathbf{F}_\mathrm{C} = \{6, \ldots, 10\}$. Note that style and content factors are independent, *i.e.*,

$$[\mathbf{z}]_{\mathbf{F}_\mathrm{S}} \perp [\mathbf{z}]_{\mathbf{F}_\mathrm{C}} .$$

We draw $\{\mathbf{z}_i\}_{i=1}^n \overset{\text{iid}}{\sim} \mathbf{N}(\mathbf{0}, \Sigma)$.

B.2. The entangled representations

We fix $d' = 10$ and obtain entangled representations as

$$\mathbf{u} = \mathbf{A}\mathbf{z} = \mathbf{L}\mathbf{U}\mathbf{z} \in \mathbf{R}^{10}, \tag{B.3}$$

where $\mathbf{A} = \mathbf{L}\mathbf{U}$, $\mathbf{U}$ is a randomly generated 10×10 orthogonal matrix and $\mathbf{L}$ is a 10×10 lower triangular matrix described below.

$$[\mathbf{L}]_{i,j} = \begin{cases} 1 & i = j, \\ 0.9 & i > j, \\ 0 & i < j. \end{cases} \tag{B.4}$$

We use the same orthogonal matrix throughout our experiment. Note that both $\mathbf{L}$ and $\mathbf{U}$ are invertible and hence $\mathbf{A} = \mathbf{L}\mathbf{U}$ is also invertible.

B.3. Sample manipulations and annotations

For each of the latent factors $\mathbf{z}$ and j-th style coordinates we obtain two manipulated latent factors which we denote as $\mathbf{z}^{(j),+}$ and $\mathbf{z}^{(j),-}$ and their description follow. $\mathbf{z}^{(j),+}$ (resp. $\mathbf{z}^{(j),-}$) sets the j-th coordinate to its positive (resp. negative) absolute value, *i.e.*

$$[\mathbf{z}^{(j),+}]_j = |[\mathbf{z}]_j| \text{ (resp. } [\mathbf{z}^{(j),-}]_j = -|[\mathbf{z}]_j|),$$

and annotate it as $+1$ (resp. -1). Since the first two coordinates are correlated with correlation coefficient ρ, if either of them changes by the value δ then the other one changes by $\rho\delta$. We provide a concrete example of change in the second coordinate for the change in the first coordinate, but a similar change happens vice-versa. Since

$$[\mathbf{z}^{(1),+}]_1 - [\mathbf{z}]_1 = |[\mathbf{z}]_1| - [\mathbf{z}]_1 ,$$

it must hold

$$[\mathbf{z}^{(1),+}]_2 - [\mathbf{z}]_2 = \rho\big(|[\mathbf{z}]_1| - [\mathbf{z}]_1\big) .$$

Note that one of $\mathbf{z}^{(j),+}$ and $\mathbf{z}^{(j),-}$ is exactly same as $\mathbf{z}$. We obtain the entangled representations as $\mathbf{u}^{(j),+} = \mathbf{A}\mathbf{z}^{(j),+}$ and $\mathbf{u}^{(j),-} = \mathbf{A}\mathbf{z}^{(j),-}$.

B.4. Style factor estimations

Note that $\mathbf{A}$ is invertible and hence the covariance matrix of $\mathbf{u} = \mathbf{A}\mathbf{z}$ is also invertible. In this case, the minimum norm least square problem in (3.2) is the simple least square problem. For $j \in \mathbf{F}_\mathrm{S}$ we describe the estimation of j-th style factor below.

$$[\hat{\mathbf{z}}]_j \triangleq \hat{\mathbf{p}}_j^\top \mathbf{u}, \text{ where}$$

$$\hat{\mathbf{p}}_j \triangleq \operatorname*{arg\,min}_{a \in \mathbf{R}, \mathbf{p} \in \mathbf{R}^{d'}} \frac{1}{2n} \sum_{i=1}^n \Big[\big(+1 - a - \mathbf{p}^\top \mathbf{u}_i^{(j),+}\big)^2 + \big(-1 - a - \mathbf{p}^\top \mathbf{u}_i^{(j),-}\big)^2\Big] \tag{B.5}$$

C. Experimental details

C.1. Feature extractors and image style generation (image transformations)

MNIST data For the colored MNIST experiment, we train a multilayer perceptron (MLP) feature extractor, a 3-layer neural network with ReLU activation function and a hidden layer of size 50. The dataset for training the feature extractor is obtained by randomly coloring some original MNIST images green and some of them red. We then train the feature extractor by making it predict both the color of the image and the digit label. During training, we use a batch size of 256 and a learning rate of 0.001.

After training the feature extractor, we use it to extract features from MNIST images that we use in the experiments. For the experiments, we use original MNIST images and MNIST images colored green.

CIFAR-10 data For experiments on CIFAR-10, we use two different feature extractors, `Supervised` and `SimCLR`. For `Supervised`, we use a `Supervised` model that was pre-trained on ImageNet (Russakovsky et al., 2015) from Pytorch's Torchvision package [4]. For `SimCLR`, we first train a `SimCLR` [5] model on the original CIFAR-10 dataset before using it to extract features.

[4]https://pytorch.org/vision/stable/index.html

[5]https://github.com/spijkervet/SimCLR

We transform the CIFAR-10 dataset four different ways to generate new sets of data that we use in our various experiment settings. The first set of data is generated by rotating the original CIFAR-10 data at angle 15 degrees, the second set is generated by applying contrast to the original CIFAR-10 data using a contrast factor of 0.3, the third set is generated by blurring the original CIFAR-10 data using a sigma value of 0.3, and the fourth set is generated by making the original CIFAR-10 images saturated using a saturation factor of 5. We selected transformation parameters that transformed the original data without changing it into something completely different and unrecognizable. For the experiments, we extract features from the original CIFAR-10, rotated CIFAR-10, contrasted CIFAR-10, blurred CIFAR-10, and saturated CIFAR-10 data. We then use the extracted features to perform experiments as described in §5 and §3.

ImageNet data For experiments on ImageNet, we use a pre-trained ResNet-50 model to extract features from the ImageNet (Russakovsky et al., 2015) dataset. The ImageNet data that we use contains 1,281,167 images for training, 50,000 images for validation, and 1000 classes.

The ImageNet dataset was used to demonstrate PISCO's ability to scale and generalize under distribution shifts. For generalization, we use the code published by Geirhos et al. (2018) to generate four stylized ImageNet datasets with covariate distribution shifts by applying four styles on original ImageNet images. The styles applied are "dog sketch", "woman sketch", "Picasso self-portrait", and "Picasso dog" (see Figure 5). Two of the styles, "dog sketch" and "Picasso dog", were generated by DALL·E 2. For the "Picasso dog" style, the prompt used to generate it from DALL·E 2 was, "portrait of a dog in Picasso's 1907 self-portrait style". For the "dog sketch" style, the prompt used to generate it from DALL·E 2 was, "artistic hand drawn sketch of a dog face". The other two styles, "woman sketch" and "Picasso self-portrait", were downloaded from a GitHub repository[6] of the style transfer project by (Huang & Belongie, 2017). The generated stylized ImageNet data is then used to test PISCO's out-of-distribution (OOD) generalization capability. Figure 5 shows example images from the ImageNet dataset and the four styles that we use to generate the stylized ImageNet sets.

C.2. Spurious correlations and error bars

Spurious correlations for the experiments in both MNIST and CIFAR-10 datasets are created by first dividing images in each dataset into two halves, the first half contains images with class label below 4 and the second half contains images with class 4 and above. We then create datasets, where the image label is spuriously correlated with the image style, *i.e.* color green, rotation, contrast, blur, or saturation, as follows: in the training dataset, images from the first half are transformed with probability α and images from the second half are transformed with probability $1-\alpha$. In the test dataset, we do the reverse of what we did in the training data; images from the first half are transformed with probability $1-\alpha$ and images from the second half are transformed with probability α. We perform experiments and report results for α values $0.5, 0.75, 0.90, 0.95, 0.99, 1.0$. At $\alpha = 0.5$, the train and test data have the same distribution, and at $\alpha = 1$, the spurious correlations between labels and styles is extreme.

Results on experiments where there are spurious correlations between label and transformations (styles) in the data are reported in plots eg. Figure 3, Figure 4, Figure 13, etc. The reported results are over 10 restarts. We include error bars in the plots, but the errors are small so the error bars are not very visible.

C.3. Training and testing logistic regression models for classification

MNIST Data For MNIST data, the digits labels are from 0 to 9. For baseline results, we train and test the logistic regression model using all the features extracted using the MLP feature extractor. For PISCO results, we discard the style feature corresponding to color green and train and test the logistic regression model using only the remaining content features.

CIFAR-10 Data For CIFAR-10 data, we use the original image labels to train and test the logistic regression model. For baseline results, we train and test the logistic regression model using features extracted using `SimCLR` or `Supervised`. For PISCO results, we first discard the four style features corresponding to rotation, contrast, blur, and saturation and then train and test the logistic regression model using only the remaining content features.

ImageNet Data For ImageNet data, similar to CIFAR-10 and MNIST settings, after fitting the logistic regression model on features extracted using ResNet-50 to obtain baseline results, ImageNet features are post-processed with PISCO to isolate content and style, and then style features get dropped when fitting the model for OOD generalization. The batch size used when training the logistic regression model on ImageNet was 32768, the learning rate was 0.0001, and the number of epochs was 50.

For more experimental details, check out our released code on GitHub[7].

[6]https://github.com/xunhuang1995/AdaIN-style/tree/master/input/style

[7]https://github.com/liliangweta/PISCO

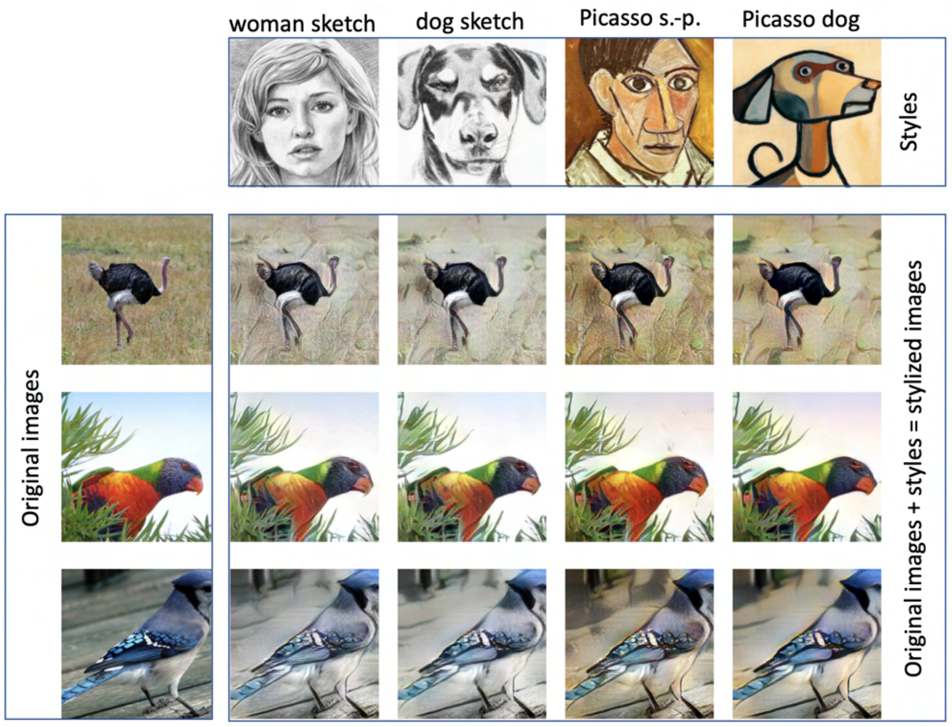


Figure 5. Example images from the ImageNet dataset with the styles applied to them.

Selecting the hyperparameter λ: λ trades-off disentanglement with the preservation of variance in the data (second and first terms in eq. (3.1), correspondingly). The easiest, no-harm, way to select λ is to increase it until the in-distribution performance starts to degrade. In our reported results, $\lambda = 1$ is the best λ value because it improves OOD performance without affecting the in-distribution accuracy. Further increasing λ can provide additional OOD gains at the cost of in-distribution performance. Our method works best when the styles are easy to predict from the original representations with a linear model (see first column in Tables **??** and **??**; note that this is also easy to evaluate at training time). For example, blur is hard to predict and PISCO with larger values of λ degrades the corresponding OOD performance in Tables 2 and 3, while rotation is easier to predict and PISCO with $\lambda = 10$ improves the performance in both tables. When a given style is hard to predict, it means that the representation is robust to it (as is the case with blur and some other styles for `SimCLR` representations) and it might make sense to exclude it when applying PISCO to avoid unnecessary trade-offs with the variance preservation. However, if strong spurious correlation is present, PISCO improves performance even for harder to predict styles (see Figures 3 and 4).

D. Additional results and selecting the hyperparameter η

In this section, we present results on the MNIST dataset (see §D.1). We also present additional ImageNet results (see §D.3) and additional CIFAR-10 results (see §D.2) on different values of the hyperparameter η, as well as ImageNet results for different values of λ. In experiments for all datasets (MNIST, CIFAR-10, and ImageNet), we have presented results for when $\eta = 0.95$. Here we present additional ImageNet and CIFAR-10 results when η is 0.90, 0.93, 0.95 (for ImageNet only), 0.98, and 1.0 to demonstrate its impact on performance. We also presented ImageNet results for when $\lambda = 1$ in the main paper; here we present additional results for when λ is 10 and 50 to demonstrate how varying λ affects performance on ImageNet data.

D.1. Colored MNIST experiment

In this experiment, our goal is to isolate color green from the digit class. First, to obtain representations with entangled color green and digit information, we train a neural network feature extractor to predict both color and digit label (see §C for details) and then use it to extract features from original and green MNIST images that we use in the experiment. In this experiment, we have a single style factor, i.e. color green, $m = 1$. We learn post-processing feature transformation matrices $\mathbf{P}(\lambda)$ with PISCO as in Algorithm 1 and report results for $\lambda \in \{1, 10, 50\}$.

Next, we create a dataset where the label is spuriously correlated with the color green, similar to Colored MNIST (Arjovsky et al., 2019). Specifically, in the training dataset, images from the first half of the classes are colored green with probability α and images from the second half of the classes are colored green with probability $1 - \alpha$. In test data the correlation is reversed, i.e., images from the first half of the classes are colored green with probability $1 - \alpha$ and images from the second half with probability α (see §C for additional details). Thus, for $\alpha = 0.5$ train and test data have the same distribution where each image is randomly colored green, and $\alpha = 1$ corresponds to the extreme spurious correlation setting.

For each α we train and test a linear model on the original representations and on PISCO representations (discarding the learned color green factor) for varying λ. We summarize the results in Figure 6. PISCO outperforms the baseline across all values of α and matches the baseline accuracy when there is no spurious correlation and train and test distributions are the same, i.e., $\alpha = 0.5$. Thus, our method provides a significant OOD accuracy boost while preserving the in-distribution accuracy.

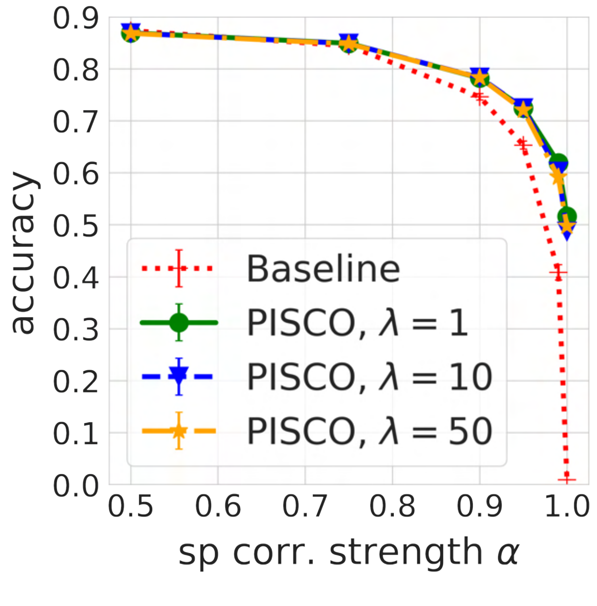
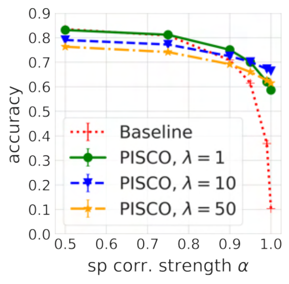
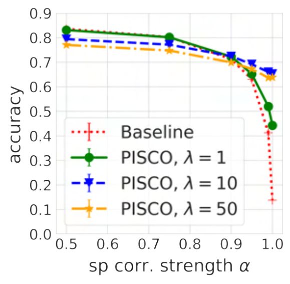


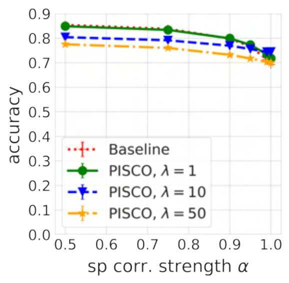
Figure 6. OOD accuracy on Colored MNIST dataset where the label is spuriously correlated with color green. The strength of the correlation is controlled by α. For each α, we train a logistic regression on the training data using the corresponding representations and report test accuracy. PISCO is robust to spurious correlations across all values of α with only a slight accuracy drop for extreme α values. The baseline is a multilayer perceptron (MLP).



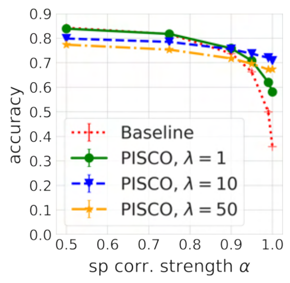
(a) Rotation - `Supervised`



(b) Contrast - `Supervised`



(c) Blur - `Supervised`



(d) Satur. - `Supervised`

Figure 7. $\eta = 0.90$, OOD performance of `Supervised` representations on CIFAR-10 where the label is spuriously correlated with the corresponding transformation. PISCO significantly improves OOD performance, especially in the case of rotation. Both $\lambda = 1$ and $\lambda = 10$ preserve in-distribution accuracy, while larger $\lambda = 50$ may degrade it as per (3.3).

D.2. Additional transformed CIFAR-10 experiment results

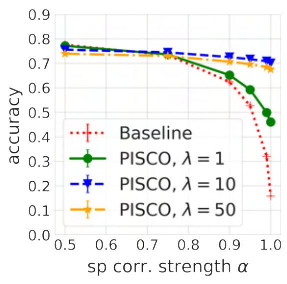
Results for $\eta = 0.90$: Results for when $\eta = 0.90$ can be found in Figure 7, Figure 8, Table 6, and Table 7.

Results for $\eta = 0.93$: Results for when $\eta = 0.93$ can be found in Figure 9, Figure 10, Table 8, and Table 9.

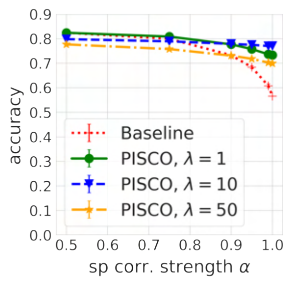
Results for $\eta = 0.98$: Results for when $\eta = 0.98$ can be found in Figure 11, Figure 12, Table 10, and Table 11.

Results for $\eta = 1.0$: Results for when $\eta = 1.0$ can be found in Figure 13, Figure 14, Table 12, and Table 13. When $\eta = 1.0$, it means the number of features in the baseline is the same as the number of features learned using PISCO and as a result, we observe the in-distribution performance of PISCO is almost the same as that of baseline methods even for higher values of λ.

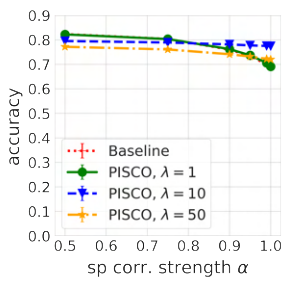
Overall CIFAR-10 results discussion. Even with different values of η, PISCO still outperforms the baselines in almost all cases. An expected observation from the results is as η increases, the in-distribution performance of PISCO goes up even for high values of λ and its OOD performance slightly goes down.



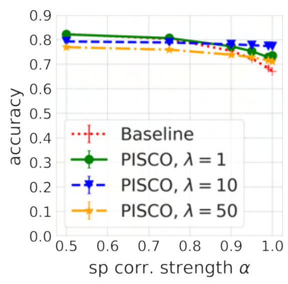
(a) Rotation - `SimCLR`



(b) Contrast - `SimCLR`



(c) Blur - `SimCLR`



(d) Saturation - `SimCLR`

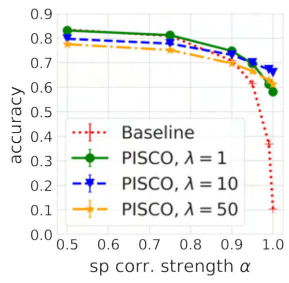
Figure 8. $\eta = 0.90$, OOD performance of `SimCLR` representations on CIFAR-10 where the label is spuriously correlated with the corresponding transformation. Results are analogous to Figure 7. The `SimCLR` baseline representations are less sensitive to contrast and saturation but remain sensitive to rotation.

Table 6. $\eta = 0.90$, Performance of `Supervised` representations on CIFAR-10 test set in-distribution, i.e., no transformation (referred to as "none"; last row), and OOD when modified with the corresponding transformation. PISCO with $\lambda = 1$ provides significant improvements for rotation, contrast, and saturation, while preserving in-distribution accuracy.

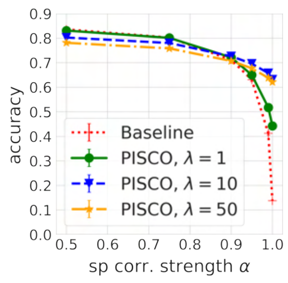
Style	Baseline (`Supervised`)	PISCO ($\lambda = 1$)	PISCO ($\lambda = 10$)	PISCO ($\lambda = 50$)
rotation	0.678	**0.741**	0.722	0.693
contrast	0.625	0.680	**0.741**	0.718
saturation	0.699	**0.759**	0.742	0.714
blur	**0.817**	**0.817**	0.777	0.750
none	**0.873**	0.869	0.823	0.791

Table 7. $\eta = 0.90$, Performance of `SimCLR` representations on CIFAR-10 test set in-distribution, i.e., no transformation (referred to as "none"; last row), and OOD when modified with the corresponding transformation. `SimCLR` features are robust to these transformations and perform similarly to PISCO with $\lambda = 1$.

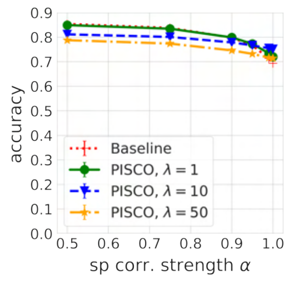
Style	Baseline (`SimCLR`)	PISCO ($\lambda = 1$)	PISCO ($\lambda = 10$)	PISCO ($\lambda = 50$)
rotation	0.620	0.632	**0.697**	0.689
contrast	**0.816**	0.815	0.795	0.775
saturation	**0.810**	0.805	0.782	0.763
blur	**0.808**	0.804	0.783	0.762
none	**0.828**	**0.828**	0.800	0.778



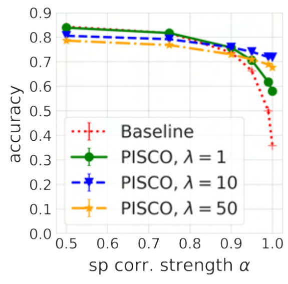
(a) Rotation - `Supervised`



(b) Contrast - `Supervised`

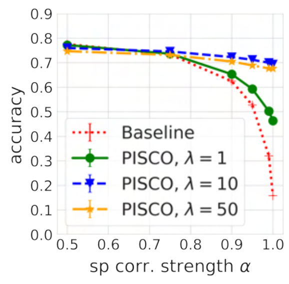


(c) Blur - `Supervised`

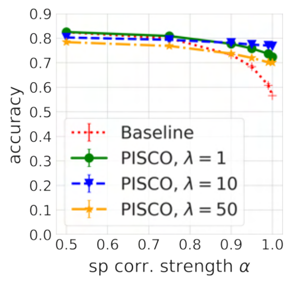


(d) Satur. - `Supervised`

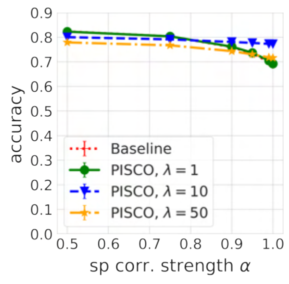
Figure 9. $\eta = 0.93$, OOD performance of `Supervised` representations on CIFAR-10 where the label is spuriously correlated with the corresponding transformation. PISCO significantly improves OOD performance, especially in the case of rotation. Both $\lambda = 1$ and $\lambda = 10$ preserve in-distribution accuracy, while larger $\lambda = 50$ may degrade it as per (3.3).



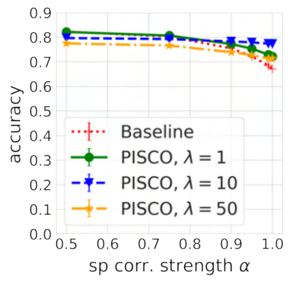
(a) Rotation - `SimCLR`



(b) Contrast - `SimCLR`



(c) Blur - `SimCLR`



(d) Saturation - `SimCLR`

Figure 10. $\eta = 0.93$, OOD performance of `SimCLR` representations on CIFAR-10 where the label is spuriously correlated with the corresponding transformation. Results are analogous to Figure 9. The `SimCLR` baseline representations are less sensitive to contrast and saturation but remain sensitive to rotation.

Table 8. $\eta = 0.93$, Performance of `Supervised` representations on CIFAR-10 test set in-distribution, i.e., no transformation (referred to as "none"; last row), and OOD when modified with the corresponding transformation. PISCO with $\lambda = 1$ provides significant improvements for rotation, contrast, and saturation while preserving in-distribution accuracy.

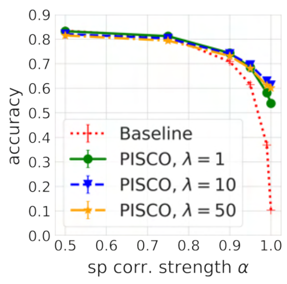
Style	Baseline (`Supervised`)	PISCO ($\lambda = 1$)	PISCO ($\lambda = 10$)	PISCO ($\lambda = 50$)
rotation	0.678	**0.741**	0.726	0.700
contrast	0.625	0.678	**0.744**	0.723
saturation	0.699	**0.759**	0.742	0.718
blur	**0.817**	**0.817**	0.788	0.761
none	**0.873**	0.871	0.827	0.805

Table 9. $\eta = 0.93$, Performance of `SimCLR` representations on CIFAR-10 test set in-distribution, i.e., no transformation (referred to as "none"; last row), and OOD when modified with the corresponding transformation. `SimCLR` features are robust to these transformations and perform similarly to PISCO with $\lambda = 1$.

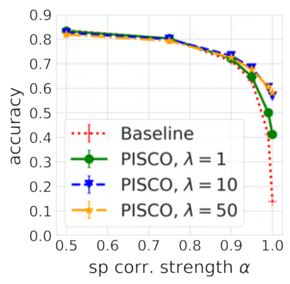
Style	Baseline (SimCLR)	PISCO ($\lambda = 1$)	PISCO ($\lambda = 10$)	PISCO ($\lambda = 50$)
rotation	0.620	0.632	**0.695**	0.692
contrast	0.816	**0.817**	0.797	0.786
saturation	**0.810**	0.809	0.786	0.765
blur	**0.808**	0.804	0.783	0.762
none	**0.828**	0.826	0.804	0.782

Table 10. $\eta = 0.98$, Performance of `Supervised` representations on CIFAR-10 test set in-distribution, i.e., no transformation (referred to as "none"; last row), and OOD when modified with the corresponding transformation. PISCO with $\lambda = 1$ provides significant improvements for rotation, contrast, and saturation, while preserving in-distribution accuracy.

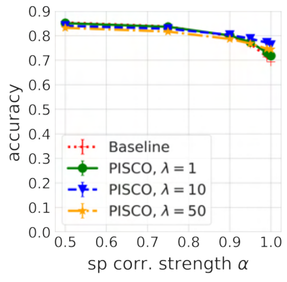
Style	Baseline (`Supervised`)	PISCO ($\lambda = 1$)	PISCO ($\lambda = 10$)	PISCO ($\lambda = 50$)
rotation	0.678	**0.739**	0.736	0.726
contrast	0.625	0.680	**0.740**	0.729
saturation	0.699	**0.757**	0.740	0.726
blur	**0.817**	**0.817**	0.807	0.797
none	**0.873**	0.871	0.861	0.851



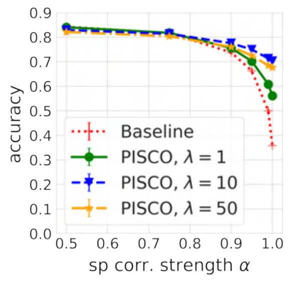
(a) Rotation - `Supervised`



(b) Contrast - `Supervised`

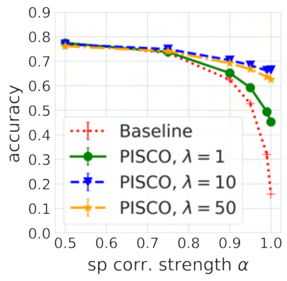


(c) Blur - `Supervised`

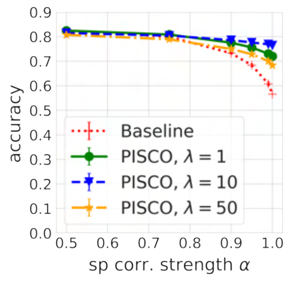


(d) Saturation - `Supervised`

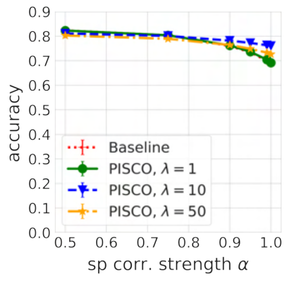
Figure 11. $\eta = 0.98$, OOD performance of `Supervised` representations on CIFAR-10 where the label is spuriously correlated with the corresponding transformation. PISCO significantly improves OOD performance, especially in the case of rotation. Both $\lambda = 1$ and $\lambda = 10$ preserve in-distribution accuracy, while larger $\lambda = 50$ may degrade it as per (3.3).



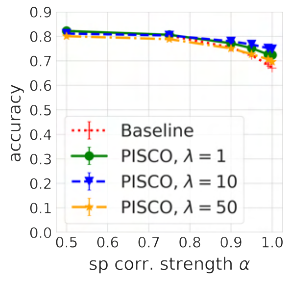
(a) Rotation - `SimCLR`



(b) Contrast - `SimCLR`



(c) Blur - `SimCLR`

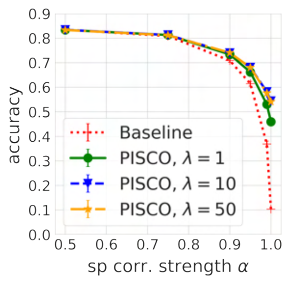


(d) Saturation - `SimCLR`

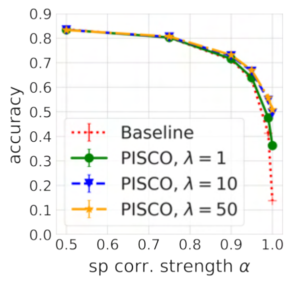
Figure 12. $\eta = 0.98$, OOD performance of `SimCLR` representations on CIFAR-10 where the label is spuriously correlated with the corresponding transformation. Results are analogous to Figure 11. The `SimCLR` baseline representations are less sensitive to contrast and saturation but remain sensitive to rotation.

Table 11. $\eta = 0.98$, Performance of `SimCLR` representations on CIFAR-10 test set in-distribution, i.e., no transformation (referred to as "none"; last row), and OOD when modified with the corresponding transformation. `SimCLR` features are robust to these transformations and perform similarly to PISCO with $\lambda = 1$.

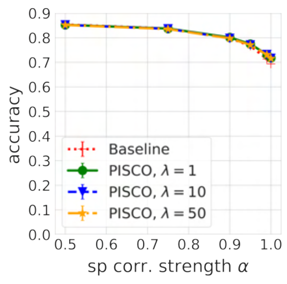
Style	Baseline (`SimCLR`)	PISCO ($\lambda = 1$)	PISCO ($\lambda = 10$)	PISCO ($\lambda = 50$)
rotation	0.620	0.633	**0.681**	0.677
contrast	0.816	**0.817**	0.808	0.799
saturation	**0.810**	0.809	0.796	0.778
blur	**0.808**	0.806	0.795	0.793
none	**0.828**	0.826	0.815	0.806



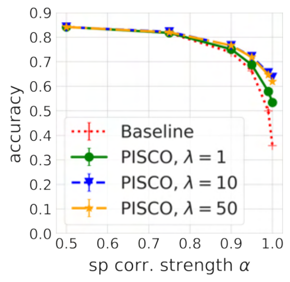
(a) Rotation - `Supervised`



(b) Contrast - `Supervised`

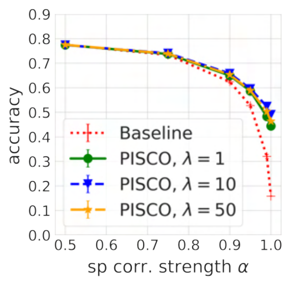


(c) Blur - `Supervised`

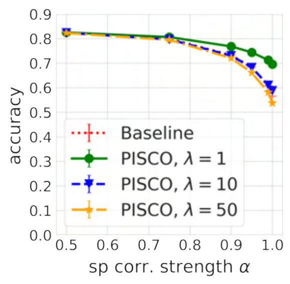


(d) Saturation - `Supervised`

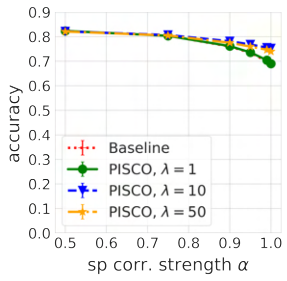
Figure 13. $\eta = 1.0$, OOD performance of `Supervised` representations on CIFAR-10 where the label is spuriously correlated with the corresponding transformation. PISCO significantly improves OOD performance, especially in the case of rotation. Both $\lambda = 1$ and $\lambda = 10$ preserve in-distribution accuracy, while larger $\lambda = 50$ may degrade it as per (3.3).



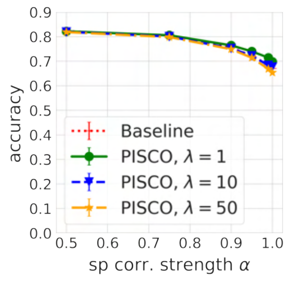
(a) Rotation - `SimCLR`



(b) Contrast - `SimCLR`



(c) Blur - `SimCLR`



(d) Saturation - `SimCLR`

Figure 14. $\eta = 1.0$, OOD performance of `SimCLR` representations on CIFAR-10 where the label is spuriously correlated with the corresponding transformation. Results are analogous to Figure 13. The `SimCLR` baseline representations are less sensitive to contrast and saturation but remain sensitive to rotation.

Table 12. $\eta = 1.0$, Performance of `Supervised` representations on CIFAR-10 test set in-distribution, i.e., no transformation (referred to as "none"; last row), and OOD when modified with the corresponding transformation. PISCO with $\lambda = 1$ provides significant improvements for rotation, contrast, and saturation, while preserving in-distribution accuracy.

Style	Baseline (`Supervised`)	PISCO ($\lambda = 1$)	PISCO ($\lambda = 10$)	PISCO ($\lambda = 50$)
rotation	0.678	0.736	**0.748**	0.747
contrast	0.625	0.669	**0.724**	0.721
saturation	0.699	**0.744**	0.736	0.729
blur	0.817	**0.820**	0.819	0.819
none	**0.873**	0.872	0.872	0.872

Table 13. $\eta = 1.0$, Performance of `SimCLR` representations on CIFAR-10 test set in-distribution, i.e., no transformation (referred to as "none"; last row), and OOD when modified with the corresponding transformation. `SimCLR` features are robust to these transformations and perform similarly to PISCO with $\lambda = 1$.

Style	Baseline (`SimCLR`)	PISCO ($\lambda = 1$)	PISCO ($\lambda = 10$)	PISCO ($\lambda = 50$)
rotation	0.620	0.629	**0.647**	0.624
contrast	**0.816**	**0.816**	0.814	0.811
saturation	**0.810**	0.808	0.806	0.805
blur	**0.808**	0.805	0.806	0.802
none	**0.828**	0.827	0.826	0.826

D.3. Additional stylized ImageNet experiment results

In this section for the ResNet-50 baseline, for each value of η, we report results for λ values 1, 10, and 50. For the MAE-ViT-Base (He et al., 2022) baseline, we report additional results for λ values 1, 10, and 50.

ResNet-50 results for $\eta = 0.90$: Results for when $\eta = 0.90$ can be found in Table 14.

ResNet-50 results for $\eta = 0.93$: Results for when $\eta = 0.93$ can be found in Table 15.

ResNet-50 results for $\eta = 0.95$: Results for when $\eta = 0.95$ can be found in Table 16. In the main paper we reported results for $\eta = 0.95$ when $\lambda = 1$. Table 16 contains results for λ value 1, and additional λ values 10 and 50.

ResNet-50 results for $\eta = 0.98$: Results for when $\eta = 0.98$ can be found in Table 17.

ResNet-50 results for $\eta = 1.0$: Results for when $\eta = 1.0$ can be found in Table 18. When $\eta = 1.0$, it means the number of features in the baseline is the same as the number of features learned using PISCO and as a result, similar for CIFAR-10 results in §D.2, we observe the in-distribution performance of PISCO in this case being almost the same as that of baseline methods even for higher values of λ.

MAE-ViT-Base results for λ values 1, 10, and 50: Results for additional values of λ when MAE-ViT-Base is the baseline are in Table 19.

Overall ImageNet results discussion. When we vary η in ImageNet experiments, we observe behavior similar to what we observed in the CIFAR-10 experiments in §D.2: PISCO outperforms the baseline across all values of η and $\lambda = 1$ provides the best PISCO results in all η values. An expected observation from the results is as η increases, the in-distribution performance of PISCO goes up even for high values of λ and its OOD performance slightly goes down.

Table 14. $\eta = 0.90$, Top-1 and top-5 accuracies on 5 variations of the ImageNet test set for Baseline pre-trained ResNet-50 features and the corresponding post-processed PISCO features on different values of λ.

Style	Baseline (ResNet-50)		PISCO ($\lambda = 1$)		PISCO ($\lambda = 10$)		PISCO ($\lambda = 50$)	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
dog sketch	0.514	0.752	**0.542**	**0.769**	0.522	0.723	0.520	0.717
woman sketch	0.477	0.711	**0.511**	**0.743**	0.490	0.693	0.488	0.686
Picasso dog	0.445	0.686	**0.495**	**0.730**	0.472	0.672	0.467	0.665
Picasso s.-p.	0.474	0.706	**0.508**	**0.737**	0.490	0.691	0.490	0.685
none	**0.758**	**0.927**	0.743	0.916	0.740	0.910	0.740	0.910

Table 15. $\eta = 0.93$, Top-1 and top-5 accuracies on 5 variations of the ImageNet test set for Baseline pre-trained ResNet-50 features and the corresponding post-processed PISCO features on different values of λ.

Style	Baseline (ResNet-50)		PISCO ($\lambda = 1$)		PISCO ($\lambda = 10$)		PISCO ($\lambda = 50$)	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
dog sketch	0.515	0.753	**0.545**	**0.775**	0.530	0.741	0.528	0.737
woman sketch	0.478	0.712	**0.514**	**0.748**	0.500	0.710	0.498	0.705
Picasso dog	0.446	0.686	**0.495**	**0.735**	0.480	0.692	0.479	0.688
Picasso s.-p.	0.474	0.706	**0.511**	**0.742**	0.501	0.709	0.499	0.706
none	**0.757**	**0.927**	0.746	0.918	0.742	0.914	0.743	0.913

Table 16. $\eta = 0.95$, Top-1 and top-5 accuracies on 5 variations of the ImageNet test set for Baseline pre-trained ResNet-50 features and the corresponding post-processed PISCO features on different values of λ.

Style	Baseline (ResNet-50)		PISCO ($\lambda = 1$)		PISCO ($\lambda = 10$)		PISCO ($\lambda = 50$)	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
dog sketch	0.516	0.752	**0.546**	**0.777**	0.534	0.751	0.532	0.750
woman sketch	0.478	0.712	**0.518**	**0.752**	0.506	0.723	0.504	0.719
Picasso dog	0.445	0.686	**0.500**	**0.738**	0.486	0.705	0.485	0.702
Picasso s.-p.	0.474	0.706	**0.514**	**0.747**	0.505	0.721	0.504	0.718
none	**0.757**	**0.927**	0.749	0.921	0.745	0.917	0.745	0.916

Table 17. $\eta = 0.98$, Top-1 and top-5 accuracies on 5 variations of the ImageNet test set for Baseline pre-trained ResNet-50 features and the corresponding post-processed PISCO features on different values of λ.

Style	Baseline (ResNet-50)		PISCO ($\lambda = 1$)		PISCO ($\lambda = 10$)		PISCO ($\lambda = 50$)	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
dog sketch	0.515	0.752	**0.553**	**0.785**	0.541	0.769	0.540	0.768
woman sketch	0.478	0.711	**0.520**	**0.757**	0.512	0.740	0.511	0.738
Picasso dog	0.446	0.686	**0.504**	**0.744**	0.492	0.725	0.492	0.722
Picasso s.-p.	0.474	0.706	**0.518**	**0.752**	0.512	0.737	0.512	0.737
none	**0.757**	**0.928**	0.754	0.926	0.751	0.922	0.751	0.921

Table 18. $\eta = 1.0$, Top-1 and top-5 accuracies on 5 variations of the ImageNet test set for Baseline pre-trained ResNet-50 features and the corresponding post-processed PISCO features on different values of λ.

Style	Baseline (ResNet-50)		PISCO ($\lambda = 1$)		PISCO ($\lambda = 10$)		PISCO ($\lambda = 50$)	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
dog sketch	0.516	0.752	**0.552**	**0.787**	0.548	**0.787**	0.548	0.786
woman sketch	0.478	0.712	0.517	**0.756**	0.517	0.755	**0.519**	**0.756**
Picasso dog	0.446	0.686	**0.501**	0.742	0.499	0.744	**0.501**	**0.745**
Picasso s.-p.	0.474	0.706	0.517	0.752	**0.518**	0.754	0.517	**0.755**
none	**0.758**	0.928	**0.758**	**0.929**	**0.758**	**0.929**	0.757	**0.929**

Table 19. $\eta = 0.95$, Top-1 and top-5 accuracies on 5 variations of the ImageNet test set for when MAE-ViT-Base features are the Baseline, and the corresponding accuracies for post-processed PISCO features on different values of λ.

Style	Baseline (MAE-ViT-Base)		PISCO ($\lambda = 1$)		PISCO ($\lambda = 10$)		PISCO ($\lambda = 50$)	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
dog sketch	0.530	0.749	0.575	**0.773**	**0.576**	0.770	0.576	0.770
Picasso dog	0.472	0.686	0.519	**0.716**	**0.520**	0.714	**0.520**	0.714
Picasso s.-p.	0.512	0.727	**0.558**	**0.752**	**0.558**	0.748	**0.558**	0.748
woman sketch	0.504	0.719	**0.550**	**0.746**	0.549	0.744	0.549	0.744
none	0.811	0.952	**0.818**	**0.953**	0.817	**0.953**	0.817	**0.953**