ADVERSARIAL DEPENDENCE MINIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Minimally redundant representations are typically learned by minimizing feature covariance. However, covariance-based methods fail to eliminate all dependencies/redundancies, as linearly uncorrelated variables can still exhibit nonlinear relationships. To address this, we introduce `ADM`, a differentiable algorithm that minimizes statistical dependence between feature dimensions through an adversarial game: auxiliary networks identify dependencies, while the encoder removes them. We prove that mutual independence is achieved at the global optimum, empirically verify convergence, and study three potential applications: extending PCA to nonlinear decorrelation, improving generalization in image classification, and preventing dimensional collapse in self-supervised learning. By promoting statistically independent representations, `ADM` paves the way for learning more robust, compressed, and generalizable representations across diverse applications.

1 INTRODUCTION

In representation learning (Rumelhart et al., 1986; Hinton et al., 2006; Bengio et al., 2013), algorithms learn to extract lower-dimensional encodings from input data, aiming for compact yet informative representations. A key strategy is to learn representations with minimally redundant dimensions, where each feature encodes a distinct concept. Such minimally redundant dimensions may offer several advantages, including efficient data compression, compositional generalization, and enhanced interpretability of the learned representations.

The strategy of minimizing redundancy has been applied in various methods, from classical techniques like Principal Component Analysis (Pearson, 1901; Hotelling, 1933) to recent self-supervised learning (SSL) approaches (Huang et al., 2018; Zbontar et al., 2021; Bardes et al., 2021). Both PCA and these SSL methods aim to extract representations with uncorrelated dimensions. However, although these methods minimize pairwise linear correlations (Cor) between embedding dimensions, non-linear dependencies may still be present. We illustrate this with Example 1 (Figure 1b).

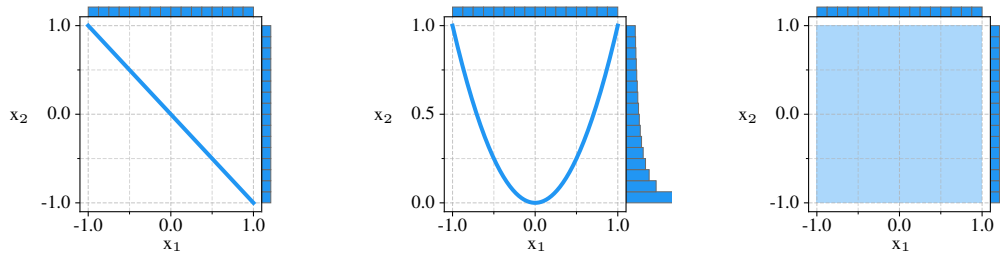
Example 1. Let a random variable x_1 be drawn from a uniform distribution over the interval $[-a, a]$ and $\mathrm{x}_2 = \mathrm{x}_1^2$. Since x_2 is a deterministic function of x_1, it is clear that the variables co-vary. However, we find: $\mathrm{Cor}(\mathrm{x}_1, \mathrm{x}_2) = 0$. The random variables are thus dependent despite a zero correlation.

This example highlights a limitation of algorithms that rely on Pearson correlation: they may not eliminate all forms of dependencies/redundancies, as linearly uncorrelated variables can still exhibit nonlinear relationships (Hyvärinen & Oja, 2000). Those methods may thus learn redundant embeddings, in which multiple dimensions encode the same concept. Still, developing a stable method for mutual and nonlinear decorrelation remains an unresolved problem. The challenge lies in designing a training objective that is simultaneously differentiable, scalable, and distribution-agnostic.

This paper presents a training algorithm for learning embeddings with minimal dependence among dimensions, achieved by removing linear, nonlinear, and higher-order relationships. The method is formulated as a minimax adversarial game between two types of players: (1) a series of small neural networks is trained to learn how embedding dimensions relate, and (2) an encoder is trained to counter them by updating the representations.

Our adversarial objective can be interpreted as a soft independence constraint in a task-specific optimization problem, enabling its incorporation into various learning algorithms. In this work, we explore three potential applications: extending PCA to non-linear decorrelation, learning features that generalize beyond label supervision in supervised learning, and preventing dimensional collapse in SSL by learning minimally redundant representations.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Lin. $\mathrm{x}_2 = -\mathrm{x}_1$: $\mathrm{Cor} = -1$ and $\mathrm{dCor} = 1$. (b) Nonlin. $\mathrm{x}_2 = \mathrm{x}_1^2$: $\mathrm{Cor} = 0$ and $\mathrm{dCor} = \sqrt{1/2}$. (c) Indep. $\mathrm{x}_2 \in \mathcal{U}(-1, 1)$: $\mathrm{Cor} = \mathrm{dCor} = 0$.

Figure 1: Illustration of the joint and marginal distributions for different types of dependencies between random variables $\mathrm{x}_1 \in \mathcal{U}(-1, 1)$ and x_2. The variables are linearly uncorrelated ($\mathrm{Cor} = 0$) in both (b) and (c) but are independent only in (c).

This paper's contributions are: (1) the introduction of an algorithm that minimizes all forms of dependencies between learned features, (2) proving that dimensions are statistically independent at the optimal solution, (3) showing empirically that the algorithm converges to this equilibrium, and (4) studying three potential applications.

2 Background

We begin by defining statistical independence and correlation measures. Let $\mathrm{x}_1, \ldots, \mathrm{x}_d$ be a finite set of random variables. These variables are mutually independent if and only if their joint cumulative distribution function (CDF) factorizes:

$$F_{\mathrm{x}_1,\ldots,\mathrm{x}_d}(x_1, \ldots, x_d) = \prod_{i=1}^{d} F_{\mathrm{x}_i}(x_i) \quad \forall x_1, \ldots, x_d \in \mathbb{R} \tag{1}$$

where F_{x_i} denotes the marginal CDF. Equivalently, mutual independence holds if and only if each variable x_i is statistically independent of the vector of all remaining variables $\mathbf{x}_{-i}$. Notably, pairwise independence between all x_i and x_j does not imply mutual independence (Driscoll, 1978). We refer to Appendix G.1 for an example.

Dependence is often quantified using Pearson's correlation coefficient (Cor), which measures linear relationships. However, $\mathrm{Cor}(\mathrm{x}_i, \mathrm{x}_j) = 0$ does not imply independence, as it fails to capture nonlinear dependencies. We can instead consider the *maximal correlation coefficient* (Gebelein, 1941; Rényi, 1959), defined as the supremum of Pearson correlations over all square-integrable non-constant transformations ϕ and ψ:

$$\mathrm{mCor}(\mathrm{x}_i, \mathrm{x}_j) \triangleq \sup_{\phi,\psi} \mathrm{Cor}(\psi(\mathrm{x}_i), \phi(\mathrm{x}_j)). \tag{2}$$

Crucially, mCor vanishes if and only if the variables are independent (Lancaster, 1960). We thus aim to design a method that minimizes the maximal correlation coefficient.

3 Related Work

Principal Component Analysis (PCA) was the first dimensionality reduction technique to focus on extracting uncorrelated features. This transformation is achieved while preserving the maximal variance in the original dataset.

Autoencoders (AEs). The PCA reduction is closely related to autoencoders (Kramer, 1991). An AE consists of two neural networks: an encoder that compresses the input into a lower-dimensional code, and a decoder that reconstructs the input from this representation. Both networks are trained jointly with a reconstruction error. Interestingly, the optimal solution to a linear AE is to perform PCA. However, unlike PCA, an AE can learn nonlinear dimensionality reductions, but its latent space

Table 1: Hierarchy of independence guarantees provided by $\mathrm{Cor}(\psi(\cdot), \phi(\cdot)) = 0$ across different method classes. While PCA and Predictability Minimization are limited to linear or mean independence, our ADM algorithm provably optimizes for mutual independence.

Method Class	Family of ψ	Family of ϕ	Input to ϕ	Implication of $\mathrm{Cor} = 0$
Decorrelation	Linear	Linear	$\mathrm{x}_j \in \mathbb{R}$	Linear independence
Predictability Min.	Linear	Square-integrable	$\mathbf{x}_{-i} \in \mathbb{R}^{d-1}$	Mean independence
ADM (ours)	Square-integrable	Square-integrable	$\mathbf{x}_{-i} \in \mathbb{R}^{d-1}$	Mutual independence

is not guaranteed to have uncorrelated dimensions. The variational autoencoder (VAE) from Kingma (2013) is an extension of AEs in which the encoder maps the input into a probabilistic representation and the decoder reconstructs data by sampling from this distribution. This extension of AEs enabled the generation of new samples similar to the training data. Multiple works (Higgins et al., 2017; Burgess et al., 2018; Kim & Mnih, 2018; Chen et al., 2018) proposed variations of the VAE objective to encourage disentangled representations. Intuitively, this implies learning representations where changes in one dimension correspond to changes in a single latent variable, but disentangled concepts may be dependent. From this perspective, the problem is ill-defined (Locatello et al., 2019) and differs from the independence objective of this work.

Predictability minimization (PM). Our framework generalizes the PM principle (Schmidhuber, 1992), which learns representations by making individual features unpredictable from the remaining ones. PM uses a minimax objective over reconstruction errors of raw features; at convergence (under standardization), this enforces $\mathbb{E}\left[\mathrm{z}_i \mid \mathbf{z}_{-i}\right] = \mathbb{E}\left[\mathrm{z}_i\right]$, a condition known as *mean independence*. While necessary, mean independence is strictly weaker than statistical independence and fails to capture higher-order dependencies (see App. G.2 for a simple example). Our method addresses this by transforming targets using *probe networks* and standardizing their outputs to ensure scale invariance; we prove that the resulting minimax equilibrium achieves mutual independence via *maximal correlation* minimization (see Table 1 for a comparison across method classes). A recent work (Zollikofer et al., 2024) applies PM to self-supervised learning by predicting randomly masked features via auxiliary networks, but inherits the same limitation as PM, leaving higher-order dependencies unaddressed.

Adversarial learning. We now discuss the core paradigm behind our algorithm. While adversarial principles date back to PM (Schmidhuber, 1992), Generative Adversarial Networks (GANs, Goodfellow et al. (2014)) popularized this framework for high-dimensional synthesis: a generator is trained to generate realistic synthetic data, while a discriminator is trained to predict whether a sample comes from the training dataset or the generator. InfoGANs (Chen et al., 2016) extends GANs by adding a criterion that maximizes a lower bound of the mutual information (MI) between the representations and the generated data. This helps learn disentangled representations. Our approach differs from InfoGANs in that it does not optimize for input-output MI maximization and is not bound to generative networks. Most similar to our work, Brakel & Bengio (2017) used adversarial networks to decrease dependence by training an encoder to produce samples from a joint distribution that are indistinguishable from samples of the product of its marginals. However, this training objective is unstable and requires careful tuning, whereas ours systematically converges.

Self-supervised learning (SSL). SSL is an active area of research in representation learning. Popularized by SimCLR (Chen et al., 2020), a leading paradigm in SSL is to train a network to align representations under carefully designed data augmentations. Intuitively, this enforces consistency between the input and the output by pushing two views of the same image to lie close in the embedding space, i.e., a slight change in the input should not lead to a completely different output. While effective, this idea comes with the risk of collapsed representations (Hua et al., 2021): dimensional collapse and collapse to a single value. The former occurs when information encoded across different dimensions of the representation is redundant (Jing et al., 2022), whereas the latter is generally due to a lack of a counterweight to the invariance term. Many SSL approaches relied on linear decorrelation of dimensions (Huang et al., 2018; Zbontar et al., 2021; Ermolov et al., 2021; Bardes et al., 2021) to avoid collapse. However, linearly uncorrelated dimensions can still co-vary and therefore suffer from dimensional collapse. Our algorithm can be used to extend this idea to minimize redundancy in general. We refer readers to App. E for an extended discussion of redundancy in SSL.

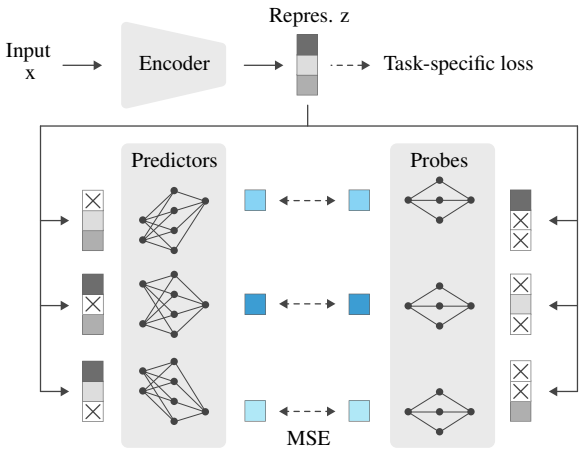


Figure 2: Overview of Adversarial Dependence Minimization. Probes expose dependencies among latent dimensions, while predictors reconstruct each transformed dimension from the others. The encoder adversarially maximizes the prediction error, driving the representation's dimensions toward statistical independence.

Independent Component Analysis (ICA). ICA (Jutten & Herault, 1991) is a method to separate mixed signals into independent ones, assuming that the observations are a linear mixture of mutually independent, non-Gaussian latent variables. Motivated by the central limit theorem, implementations typically involve the maximization of non-Gaussianity by relying on differential entropy (Hyvärinen & Oja, 2000) or mutual information (Bell & Sejnowski, 1995). When our method's encoder is a linear layer with equal input and output dimensions, it simplifies to an ICA implementation that does not explicitly optimize for non-Gaussianity. Also, when combined with an AE, our algorithm performs a joint PCA and ICA. However, the successive application of PCA and ICA (Back & Weigend, 1997; Yao et al., 2012) is not equivalent to our approach, since PCA may extract redundant features that ICA is then ill-posed on.

4 Adversarial Dependence Minimization

We present Adversarial Dependence Minimization (ADM), an algorithm for learning to extract features that are minimally interdependent. Consider the representations $\boldsymbol{z}^{(i)} = f_\theta(\boldsymbol{x}^{(i)})$ from input samples $\boldsymbol{x}^{(i)}$ drawn i.i.d. from a dataset $\mathbf{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^N$. Our algorithm involves three types of networks:

1. an encoder $f_\theta : \mathcal{X} \to \mathbb{R}^d$ that learns lower-dimensional representations of the training data.
2. a set of *probe networks* $\boldsymbol{\psi}$ that transform each feature z_i into a new variable $\tilde{z}_i = \psi_i(z_i)$ to maximize predictability. Intuitively, probes aim to expose dependencies that are impossible to predict with a simple regression on z_i.
3. a set of *dependency predictors* $\boldsymbol{\phi}$ with one predictor for every embedding dimension: $\phi_i : \mathbb{R}^{d-1} \to \mathbb{R}$. The predictors are trained to learn how dimensions relate by estimating the conditional expectation of a (transformed) dimension given the context of all others: $\hat{z}_i = \phi_i(\boldsymbol{z}_{-i}) = \phi_i(z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_d)$.

Taking inspiration from GANs (Goodfellow et al., 2014), we formulate training as a minimax game: the dependence branch $(\boldsymbol{\phi}, \boldsymbol{\psi})$ serves as a critic, minimizing the reconstruction error to expose dependencies. Simultaneously, the encoder (θ) maximizes the error by driving features toward statistical independence. The objective function is:

$$\max_{\theta} \min_{\{\boldsymbol{\phi},\boldsymbol{\psi}\}} \quad \mathbb{E}_{\boldsymbol{z}\sim P(\mathcal{X};\theta)} \frac{1}{d} \sum_{i=1}^{d} \left(\psi_i(z_i) - \phi_i(\boldsymbol{z}_{-i})\right)^2 \tag{3}$$

where $P(\mathcal{X}; \theta)$ denotes the encoder's learned representation distribution, parameterized by θ.

4.1 STANDARDIZATION AND PROBLEM SOLUTION

We may, without loss of generality, assume that probe outputs have zero mean and unit variance. This standardization makes the problem scale-invariant and ensures that the objective is well-defined, i.e., has finite and positive variance. Indeed, without constraints, the probe networks could trivially minimize Equation (3) by collapsing $\psi_i(z_i)$ to a constant (zero variance).

Practically, we standardize the output of the probe networks before computing the loss: $\tilde{z}_i \leftarrow (\tilde{z}_i - \mathbb{E}[\tilde{z}_i]) / \sqrt{\mathbb{V}[\tilde{z}_i]}$. Where the mean $\mathbb{E}[\tilde{z}_i]$ and variance $\mathbb{V}[\tilde{z}_i]$ statistics are estimated over the current mini-batch, similar to Batch Normalization (Ioffe & Szegedy, 2015).

We proceed by showing that, under standardization, the minimax objective has a global solution corresponding to mutual independence.

Theorem 4.1 (Global Optimality)**.** *Let* $\mathbf{z}$ *be the representation vector. Assume* $\mathbf{z}$ *induces a square-integrable distribution (all functions in* L^2*), and infinite capacity for the predictors* $\boldsymbol{\phi}$ *and probes* $\boldsymbol{\psi}$*, and that* ψ_i *are standardized to have zero mean and unit variance, the minimax solution to Equation* (3) *is achieved if and only if the components of* $\mathbf{z}$ *are mutually statistically independent.*

Proof Sketch. See Appendix C.1 for the full proof. Briefly, for fixed encoder θ and probes ψ, the optimal predictors ϕ^* are the conditional expectations $\mathbb{E}[\psi_j(z_i)|\mathbf{z}_{-i}]$. Substituting these into the loss yields the sum of conditional variances. Minimizing it is equivalent to maximizing the *Explained Variance* (fraction of variance of $\psi_j(z_i)$ explainable by $\mathbf{z}_{-i}$). By Rényi's theorem (Rényi, 1959), the supremum of this quantity is exactly the squared Maximal Correlation. Thus, the encoder maximizes the loss by forcing $\mathrm{mCor}(\mathrm{z}_i, \mathbf{z}_{-i}) = 0$ for all i, which implies mutual independence. □

Corollary 4.2. *At convergence, dependency predictors predict the constant function* $\phi_i(\mathbf{z}_{-i}) = \mathbb{E}[\tilde{\mathrm{z}}_i] = 0$ *for all* i*, and the expected average reconstruction error is equal to 1.*

4.2 IMPLEMENTATION

We parameterize both the predictors and probes using Multi-Layer Perceptrons (MLPs), as they are universal function approximators (Hornik et al., 1989; Cybenko, 1989; Leshno et al., 1993) and can, in theory, approximate arbitrarily well the relation among the variables if given enough capacity.

Input Standardization. We further standardize representations before feeding them into the dependence branch to simplify optimization. Although this step is not strictly required, it consistently yields a modest improvement in convergence speed.

Efficiency via Parallelization. Running d separate networks for the predictors and probes naively is computationally expensive. Since all predictors ϕ_i share the same architecture but take distinct masked inputs, we implement the full set $\boldsymbol{\phi}$ as a single batched network: a dense linear layer mapping $\mathbb{R}^d \to \mathbb{R}^{d \cdot h}$ with a precomputed static weight mask zeroing diagonal self-connections, which ensures the i-th group of hidden units receives only $\boldsymbol{z}_{-i}$ without materializing d separate inputs. Followed by per-feature hidden layers processed jointly via batched `einsum` operations over weight tensors. We apply the same strategy to the probe networks $\boldsymbol{\psi}$: rather than looping through d independent MLPs, we stack their parameters and evaluate all probes simultaneously via a single `einsum` call per layer. Together, these designs significantly reduce computational overhead by exploiting GPUs' parallelization capabilities.

Preventing Variance Collapse. Because standardization makes the objective scale-invariant, the pre-standardized representations are free to adopt arbitrary scales. In practice, this can result in variances approaching zero, which may cause numerical instabilities. We therefore apply a hinge loss (Cortes, 1995; Tsochantaridis et al., 2005) to the squared standard deviation of the pre-standardized representations, encouraging it to remain above 1.

5 APPLICATIONS

Having developed an algorithm for mutual and nonlinear decorrelation, we explore its applicability across diverse tasks. A key incentive for learning minimally dependent features is that independence

implies minimal redundancy (Shannon, 1948; Barlow et al., 1961), which may be beneficial in various contexts.

Consider a task-specific minimization problem with objective function $\mathcal{L}_{\text{task}}$ that we would like to restrict to solutions for which the output dimensions are mutually independent (denoted by $\perp\!\!\!\perp$):

$$\min_{\theta} \quad \mathcal{L}_{\text{task}}(\boldsymbol{X}, \cdot; \theta) \quad \text{s.t.} \quad \mathrm{z}_1 \perp\!\!\!\perp \mathrm{z}_2 \perp\!\!\!\perp \cdots \perp\!\!\!\perp \mathrm{z}_d \tag{4}$$

While solving this constrained problem directly is infeasible, the ADM algorithm can be seen as a Lagrangian relaxation of the independence constraint. Indeed, the encoder's objective can be written as a constraint of the form $\mathcal{L}_{\text{adm}}(\cdot) = 0$, which is satisfied when the dimensions are mutually independent. Therefore, the problem can be relaxed to:

$$\min_{\theta} \quad \mathcal{L}_{\text{task}}(\boldsymbol{X}, \cdot; \theta) + \lambda \mathcal{L}_{\text{adm}}(\boldsymbol{X}; \theta, \boldsymbol{\phi}, \boldsymbol{\psi}) \tag{5}$$

where $\lambda \in \mathbb{R}$ is a Lagrange multiplier and where the adversarial objective from the encoder is denoted by $\mathcal{L}_{\text{adm}}(\boldsymbol{z}; \boldsymbol{\phi}, \boldsymbol{\psi}) = \frac{1}{d} \sum_{i=1}^{d} 1 - (\psi_i(z_i) - \phi_i(\boldsymbol{z}_{-i}))^2$.

In the following, we investigate three potential applications: an extension of PCA to nonlinear decorrelation (Section 5.1), the minimization of redundancy in a supervised classification setting (Section 5.2), and the use of the algorithm to prevent dimensional collapse in SSL (Section 5.3).

5.1 Principal and Independent Component Analysis

PCA is a linear dimensionality reduction technique that transforms a set of variables $\mathbf{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$, $\boldsymbol{x}^{(i)} \in \mathbb{R}^l$ into a set of uncorrelated representations $\boldsymbol{z}^{(i)} = \boldsymbol{W}^T \boldsymbol{x}^{(i)}$, $\boldsymbol{z}^{(i)} \in \mathbb{R}^d$, where the columns of the matrix $\boldsymbol{W} \in \mathbb{R}^{l \times d}$ are called principal components. Intuitively, PCA identifies directions $\boldsymbol{W}_{:,i}$ in the input space along which the variance of the projected data is maximized. Assuming centered data, the PCA reduction solves the following problem:

$$\max_{\boldsymbol{W}} \quad \frac{1}{N-1} \sum_{i=1}^{N} \|\boldsymbol{W}^T \boldsymbol{x}^{(i)}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{W}^T \boldsymbol{W} = \boldsymbol{I}_d \tag{6}$$

We proceed to discuss two key properties of PCA. Firstly, the orthogonality constraint on the projection matrix $\boldsymbol{W}$ ensures that the dimensions of the projections are uncorrelated: $\mathrm{Cov}(\mathrm{z}_i, \mathrm{z}_j) = 0$ for all $i \neq j$. Secondly, PCA can equivalently be defined as finding the projection matrix minimizing the MSE for reconstructions $\hat{\boldsymbol{x}}^{(i)} = \boldsymbol{W}\boldsymbol{z}^{(i)} = \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{x}^{(i)}$. This dual form led to the use of autoencoders to solve PCA (Baldi & Hornik, 1989; Kramer, 1991; Hinton & Salakhutdinov, 2006). It is worth noting that the unit-norm constraint on the principal components is not needed in the dual formulation, as the reconstruction error inherently accounts for the scale of the projections.

Given these observations, we extend the PCA algorithm to non-linear decorrelation by replacing the covariance constraint with our more general dependence minimization objective:

$$\min_{\boldsymbol{W}} \quad \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{x}^{(i)} - \boldsymbol{W}\boldsymbol{z}^{(i)}\|_2^2 + \lambda \mathcal{L}_{\text{adm}}(\boldsymbol{z}^{(i)}; \boldsymbol{\phi}, \boldsymbol{\psi}) \tag{7}$$

This PCA extension also holds for non-linear AEs, leading to an NLPCA reduction (Kramer, 1991) with minimal redundancy. We refer interested readers to Appendix A.2. In the following, we illustrate the relevance of this PCA extension with a simple example and refer to it as *Principal and Independent Component Analysis* (PICA).

Example 2. let two independent latent factors v_1 and v_2 be uniformly distributed over the interval $[-\sqrt{3}, \sqrt{3}]$ (so that $\mathbb{V}[\mathrm{v}_1] = \mathbb{V}[\mathrm{v}_2] = 1$) and observations of a random vector defined by: $\mathbf{x} = \left[5\mathrm{v}_1, 3\cos(2\pi \mathrm{v}_1/\sqrt{3}), \mathrm{v}_2\right]^T$. The observations are linearly uncorrelated by construction: $\mathrm{Cov}[\mathrm{x}_i, \mathrm{x}_j] = 0$ for all $i \neq j$, and with descending variances: $\mathbb{V}[\mathrm{x}_1] = 25$, $\mathbb{V}[\mathrm{x}_2] = 9/2$, $\mathbb{V}[\mathrm{x}_3] = 1$. Since PCA extracts uncorrelated variables with maximal variance, its solution for $d = 2$ is: $\mathbf{z}_{\text{PCA}} = [\mathrm{x}_1, \mathrm{x}_2]^T$. Yet, this solution is redundant, since both x_1 and x_2 are functions of v_1. In contrast, the solution to PICA is the maximum variance solution with independent dimensions, namely: $\mathbf{z}_{\text{PICA}} = [\mathrm{x}_1, \mathrm{x}_3]^T = [5\mathrm{v}_1, \mathrm{v}_2]^T$. Unlike PCA, these learned features capture both true latent factors. We further discuss this example and empirically study its solutions in Appendix A.1.

5.2 Classifiers and Generalization

We observe that a key limitation of naive classifiers is their poor generalization to out-of-distribution (OOD) data. This limitation can be illustrated by the following example.

Example 3. Consider a dataset of colored geometric shapes with three training classes: red squares, green triangles, and blue triangles. Because color alone suffices to discriminate these classes, a classifier trained to minimize empirical risk can achieve zero training loss by relying solely on color, without learning shape-related features. Consequently, when evaluated on OOD samples such as red triangles, the classifier misclassifies them as red squares.

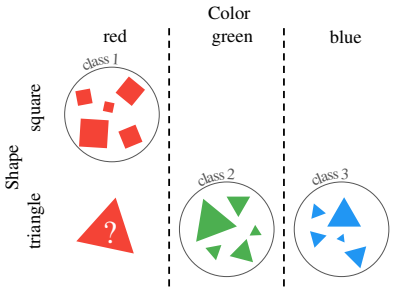


Figure 3: Illustration for Ex. 3: if a classifier only relied on *color* (dashed lines), it would predict that the red triangle is a red square.

The failure in Example 3 arises because the classifier ignores the shape. To mitigate this, we propose using ADM to minimize redundancy among embedding dimensions, encouraging the encoder to capture multiple factors of variation (e.g., shape and color) rather than a single dominant feature. This should help the model generalize to unseen combinations.

Formally, let $W \in \mathbb{R}^{n_c \times d}$ and $\boldsymbol{b} \in \mathbb{R}^{n_c}$ denote the linear head parameters for n_c classes and $\boldsymbol{z}^{(i)} \in \mathbb{R}^d$ the backbone's output. Then, we define the loss function as a weighted combination of a softmax cross-entropy and the ADM loss:

$$\min_{\theta, W, \boldsymbol{b}} \quad \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{CE}}(\text{softmax}(W\boldsymbol{z}^{(i)} + \boldsymbol{b}), y^{(i)}) + \lambda \mathcal{L}_{\text{adm}}(\boldsymbol{z}^{(i)}; \boldsymbol{\phi}, \boldsymbol{\psi}) \tag{8}$$

5.3 Self-Supervised Learning

SSL aims to learn representations without labels that transfer well to downstream tasks. Following common practice (Chen et al., 2020; Zbontar et al., 2021), we train a model by pushing two views of the same image to lie close in the embedding space. Specifically, given a mini-batch of n images, we duplicate each image, apply different augmentations to the two views, and minimize the MSE between the resulting embeddings $\boldsymbol{z}$ and $\boldsymbol{z}'$.

While this objective encourages invariance, embeddings can still suffer from full or dimensional collapse, limiting their usefulness for downstream tasks. Existing redundancy reduction techniques (Huang et al., 2018; Zbontar et al., 2021) mitigate collapse by decorrelating feature dimensions. However, linearly uncorrelated embeddings may retain significant redundancy. To address this, we use the ADM objective and generalize existing decorrelation techniques. The objective becomes:

$$\min_{\theta} \quad \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{z}^{(i)} - (\boldsymbol{z}')^{(i)}\|_2^2 + \frac{\lambda}{2} \left(\mathcal{L}_{\text{adm}}(\boldsymbol{z}^{(i)}; \boldsymbol{\phi}, \boldsymbol{\psi}) + \mathcal{L}_{\text{adm}}((\boldsymbol{z}')^{(i)}; \boldsymbol{\phi}, \boldsymbol{\psi}) \right) \tag{9}$$

6 Experiments

Our experimental evaluation aims to assess the algorithm's convergence and study the potential of the supervised and SSL applications. The PICA application is studied in App. A.

6.1 Convergence

This section analyzes the convergence of the adversarial game. Specifically, we trained a ResNet-18 (He et al., 2016) on the TinyImageNet dataset (Le & Yang, 2015) without data augmentations. We compared ADM to a linear decorrelation baseline. Experimental details are provided in App. I.1.

Metrics. We estimated nonlinear dependencies using the distance correlation (Székely et al., 2007) between one dimension and the random vector formed by the remaining $d - 1$ dimensions. We

reported the average of the estimates across the first 64 dimensions. Distance correlation is a non-negative coefficient that characterizes correlations between random vectors by measuring the distance between their joint characteristic function and the product of the marginal characteristic functions. For more details, see App. G.3. Of particular interest, the distance correlation is zero $\mathrm{dCor}\,(\mathrm{x}_1, \mathrm{x}_2) = 0$ if and only if x_1 and x_2 are independent.

Results. The MSE and distance correlation for the TinyImageNet experiment are reported in Figure 4. While both methods reduce correlation, adversarial dependence minimization reached a significantly lower final distance correlation than the linear decorrelation approach. Furthermore, ADM converged to an average reconstruction error of 1 as expected by Corollary 4.2. These results support the convergence hypothesis and empirically demonstrate the benefits of ADM over linear decorrelation.

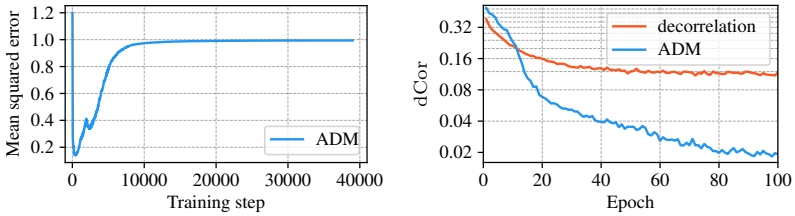


Figure 4: Convergence analysis on TinyImageNet. **Left:** the reconstruction error of ADM converges to 1. **Right:** logarithmic plot of the distance correlation on the validation set: dCor decreases over time, and ADM reaches much lower levels than the linear decorrelation loss.

6.2 Redundancy Minimization

This section investigates the impact of our algorithm on redundancy reduction in a classification setting. To facilitate the analysis of the representations, the classifiers were trained on a synthetic dataset with known latent factors.

Dataset. The Clevr-4 (Vaze et al., 2024) dataset is an extension of the CLEVR dataset (Johnson et al., 2017). It comprises 100,000 synthetic images of 3D objects of various shapes, colors, textures, and counts. Each taxonomy has 10 classes. The label for one taxonomy is sampled uniformly and independently from the others, implying that knowing the label for one taxonomy provides no information about the other taxonomies.

Evaluation protocol. We investigated generalization capabilities by training a classifier on one taxonomy (*shape*) and evaluating its accuracy on the remaining taxonomies to assess if representations encode features beyond those relevant to the training classes. The model is compared with a baseline classifier. We evaluate accuracy using a weighted nearest-neighbor (kNN) classifier trained on frozen features (Wu et al., 2018). The kNN algorithm classifies embeddings based on the majority class of their nearest neighbors, providing an easy way to assess clustering quality for each taxonomy.

Implementation details. The models are trained for 200 epochs with two data augmentations: random horizontal flipping and cropping. We train ResNet-18 encoders. For ADM, we used two-layer dependency predictors and probes. The networks are trained alternately, with one step for each network per iteration. More details are provided in App. I.2.

Results. We compared our regularized classifier to the baseline classifier. Table 2 reports the validation accuracy and feature correlation. The embedding dimensions of the baseline are highly correlated, with an average squared distance correlation of 0.409. Furthermore, performance on the taxonomies for which the model received no supervision is low, as expected, since it was not incentivized to retain information about them. When combining the cross-entropy loss with our adversarial objective, distance correlation drops to 0.038 and the accuracy on the *texture* and *color* taxonomies rises significantly. These results validate that ADM reduces redundancy, leading to representations that generalize better.

6.3 Self-Supervised Learning

This section examines whether our SSL approach prevents dimensional collapse and compares its performance to common approaches on ImageNet.

Table 2: kNN evaluation of the classification approaches on Clevr-4. ADM significantly reduces redundancy and helps the classifier generalize beyond the training taxonomy (shape).

method	kNN top-1 accuracy [%]				dCor^2
	shape	texture	color	count	
cross-entropy	**100.0**	25.0	16.4	**36.1**	0.409
cross-entropy + ADM	**99.9**	**81.8**	**98.9**	**36.0**	**0.038**

Dataset. ImageNet-1k (Deng et al., 2009) is a standard benchmark for SSL. It contains 1,281,167 training images across 1000 classes. This dataset provides a diverse collection of real-world images.

Implementation and evaluation. We trained a ResNet-50 backbone with a three-layer projection head with output dimension 512, and two-layer dependency predictors. At evaluation, the projection head is discarded, and the quality of the representations is evaluated by training a linear classifier on top of the backbone with its weights frozen. The detailed experimental setup is described in App. I.3.

Results and discussion. The main SSL methods are compared in Table 3. The average squared distance correlation achieved by our method is 0.022, indicating that the learned representation largely avoids dimensional collapse (Jing et al., 2022). This value is also substantially lower than those obtained by prior decorrelation-based approaches, such as Barlow Twins and VICReg, highlighting the benefit of accounting for nonlinear dependencies rather than only linear correlations. Nevertheless, the method does not match the performance of the strongest SSL baselines. A potential explanation is that, while eliminating inter-dimensional redundancy maximizes representational capacity for a fixed latent dimensionality, these representations may not be organized in a form readily exploitable by standard evaluation protocols: kNN and linear probing favor embeddings whose semantic structure is directly reflected in their geometry. Consequently, classification from such representations may require more expressive decoders. In summary, ADM is suboptimal for SSL as it may make features only nonlinearly extractable.

Table 3: Linear evaluation of SSL techniques trained with a ResNet-50 backbone on the ImageNet-1k dataset. The average squared distance correlation is estimated on the output representation.

method	acc. [%]	dCor^2
MoCo (He et al., 2020)	60.6	0.111
SimCLR (Chen et al., 2020)	69.3	—
Barlow Twins (Zbontar et al., 2021)	73.2	0.128
VICReg (Bardes et al., 2021)	73.2	0.130
BYOL (Grill et al., 2020)	74.3	0.145
DINO (Caron et al., 2021)	**75.3**	0.490
ours	65.3	**0.022**

7 Conclusion

This work studied the problem of learning representations with statistically independent dimensions and introduced the first general algorithm for obtaining such features. We showed that the minimax optimum of the proposed objective is attained if and only if the learned dimensions are mutually independent, providing a principled solution to the problem.

We further developed a computationally efficient implementation that scales to high-dimensional embeddings with thousands of dimensions. Using this implementation, we showed that ADM can recover principal and independent components, that reducing redundancy in supervised classification can improve generalization beyond the training classes, and that it prevents dimensional collapse in SSL, albeit at the cost of less accessible features.

While many questions remain open (see App. B), our results suggest that independence is a promising direction for representation learning: explicitly reducing redundancy may improve generalization, promote more compositional representations, and enable more effective compression.

AI USE STATEMENT

In this work, we used generative AI tools to refine writing at the paragraph level and to assist in writing intermediate proof steps for Theorem 4.1. We have not used generative AI tools to design methods, experiments, or figures. Related work was also identified and described without the aid of generative AI tools. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, or artifacts produced using generative AI.

REFERENCES

Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pp. 1247–1255. PMLR, 2013.

Andrew D Back and Andreas S Weigend. A first application of independent component analysis to extracting structure from stock returns. *International journal of neural systems*, 8(04):473–484, 1997.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2021.

Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01):217–233, 1961.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Philemon Brakel and Yoshua Bengio. Learning independent features with adversarial nets for non-linear ica. *arXiv preprint arXiv:1710.05050*, 2017.

Leo Breiman and Jerome H Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association*, 80(391):580–598, 1985.

Simon Buchholz, Michel Besserve, and Bernhard Schölkopf. Function classes for identifiable nonlinear independent component analysis. *Advances in Neural Information Processing Systems*, 35:16946–16961, 2022.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

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

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

Corinna Cortes. Support-vector networks. *Machine Learning*, 1995.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

George Darmois. Analyse des liaisons de probabilité. In *Proc. Int. Stat. Conferences 1947*, pp. 231, 1951.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE, 2009.

Michael F Driscoll. On pairwise and mutual independence: characterizations of rectangular distributions. *Journal of the American Statistical Association*, 73(362):432–433, 1978.

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *International conference on machine learning*, pp. 3015–3024. PMLR, 2021.

Hans Gebelein. Das statistische problem der korrelation als variations-und eigenwertproblem und sein zusammenhang mit der ausgleichsrechnung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 21(6):364–379, 1941.

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

P Goyal. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Vincent Grari, Oualid El Hajouji, Sylvain Lamprier, and Marcin Detyniecki. Learning unbiased representations via rényi minimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 749–764. Springer, 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

Harold Hotelling. Relations between two sets of variates. In *Biometrika*, pp. 321–377. 1936.

Hsiang Hsu, Salman Salamatian, and Flavio P Calmon. Correspondence analysis using neural networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2671–2680. PMLR, 2019.

Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9598–9608, 2021.

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 791–800, 2018.

Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in neural information processing systems*, 29, 2016.

Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022.

Daniel D Johnson, Ayoub El Hanchi, and Chris J Maddison. Contrastive learning can find an optimal basis for approximately view-invariant functions. *arXiv preprint arXiv:2210.01883*, 2022.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Christian Jutten and Jeanny Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal processing*, 24(1):1–10, 1991.

Christian Jutten and Juha Karhunen. Advances in blind source separation (bss) and independent component analysis (ica) for nonlinear mixtures. *International journal of neural systems*, 14: 267–92, 11 2004.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics*, pp. 2207–2217. PMLR, 2020.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International conference on machine learning*, pp. 2649–2658. PMLR, 2018.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

Pei Ling Lai and Colin Fyfe. Kernel and nonlinear canonical correlation analysis. *International journal of neural systems*, 10(05):365–377, 2000.

HO Lancaster. On statistical independence and zero correlation in several dimensions. *Journal of the Australian Mathematical Society*, 1(4):492–496, 1960.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

Yazhe Li, Roman Pogodin, Danica J Sutherland, and Arthur Gretton. Self-supervised learning with kernel dependence maximization. *Advances in Neural Information Processing Systems*, 34: 15543–15556, 2021.

Ralph Linsker. An application of the principle of maximum information preservation to linear systems. *Advances in neural information processing systems*, 1, 1988.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Grégoire Mialon, Randall Balestriero, and Yann LeCun. Variance covariance regularization enforces pairwise independence in self-supervised representations. *arXiv preprint arXiv:2209.14905*, 2022.

Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Carlos Ramos-Carreño and José L. Torrecilla. dcor: Distance correlation and energy statistics in Python. *SoftwareX*, 22, 2 2023. doi: 10.1016/j.softx.2023.101326. URL `https://www.sciencedirect.com/science/article/pii/S2352711023000225`.

Alfréd Rényi. On measures of dependence. *Acta mathematica hungarica*, 10(3):441–451, 1959.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7): 1757–1772, 2012.

Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural computation*, 4 (6):863–879, 1992.

Jürgen Schmidhuber, Martin Eldracher, and Bernhard Foltin. Semilinear predictability minimization produces well-known feature detectors. *Neural Computation*, 8(4):773–786, 1996.

Claude E Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27 (3):379–423, 1948.

AKAHO Shotaro. A kernel method for canonical correlation analysis. In *International Meeting of Psychometric Society*, 2001.

Ravid Shwartz Ziv and Yann LeCun. To compress or not to compress—self-supervised learning and information theory: A review. *Entropy*, 26(3):252, 2024.

Karthik Sridharan and Sham M Kakade. An information theoretic framework for multi-view learning. In *COLT*, number 114, pp. 403–414, 2008.

Gábor J Székely, Maria L Rizzo, and Nail K Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769, 2007.

Anisse Taleb and Christian Jutten. Source separation in post-nonlinear mixtures. *IEEE Transactions on signal Processing*, 47(10):2807–2820, 1999.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, Yasemin Altun, and Yoram Singer. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(9), 2005.

Sagar Vaze, Andrea Vedaldi, and Andrew Zisserman. No representation rules them all in category discovery. *Advances in Neural Information Processing Systems*, 36, 2024.

Alexei Vinokourov, Nello Cristianini, and John Shawe-Taylor. Inferring a semantic representation of text via cross-language correlation analysis. *Advances in neural information processing systems*, 15, 2002.

David Williams. *Probability with martingales*. Cambridge university press, 1991.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Xiangxiang Xu and Lizhong Zheng. Neural feature learning in function space. *Journal of Machine Learning Research*, 25(142):1–76, 2024.

Fangzhou Yao, Jeff Coquery, and Kim-Anh Lê Cao. Independent principal component analysis for biologically meaningful dimension reduction of large biological data sets. *BMC bioinformatics*, 13:1–15, 2012.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pp. 12310–12320. PMLR, 2021.

Runtian Zhai, Bingbin Liu, Andrej Risteski, Zico Kolter, and Pradeep Ravikumar. Understanding augmentation-based self-supervised representation learning via rkhs approximation and regression. *arXiv preprint arXiv:2306.00788*, 2023.

David Zollikofer, Béni Egressy, Frederik Benzing, Matthias Otth, and Roger Wattenhofer. Beyond pairwise correlations: Higher-order redundancies in self-supervised representation learning. *arXiv preprint arXiv:2412.01926*, 2024.

A Principal and Independent Component Analysis (PICA)

A.1 Empirical Study of Example 2

In Example 2, the solution that maximizes the explained variance under the zero correlation constraint is $\mathbf{z}_{\text{PCA}} = [\mathrm{x}_1, \mathrm{x}_2]^T$, with a total variance of $\mathbb{V}[\mathbf{z}_{\text{PCA}}] = 25 + 4.5 = 29.5$. However, this is not a solution to PICA since $\mathrm{x}_1 = 5\mathrm{v}_1$ and $\mathrm{x}_2 = 3\cos 2\pi \mathrm{v}_1/\sqrt{3}$ are both functions of the same latent factor v_1. The solution to PICA is thus $\mathbf{z}_{\text{PICA}} = [\mathrm{x}_1, \mathrm{x}_3]^T$, with a total explained variance of $\mathbb{V}[\mathbf{z}_{\text{PICA}}] = 25 + 1 = 26$. Attention should be drawn to the fact that the explained variance of PICA is always smaller than or equal to the PCA decomposition. Equality occurs only when the highest-variance uncorrelated combination of the inputs is mutually independent.

We empirically study this example by comparing the solutions to four different implementations:

1. the PCA decomposition solved with a Singular Value Decomposition. For this example, we rely on the *scikit-learn* (Pedregosa et al., 2011) implementation: `https://scikit-learn.org/1.6/modules/generated/sklearn.decomposition.PCA.html`.
2. the PCA decomposition implemented with a linear autoencoder and a covariance minimization objective. This approach is similar to (Mialon et al., 2022) but without their variance regularization term.
3. the PCA decomposition implemented with a linear autoencoder and our adversarial objective with linear dependency predictors.
4. the PICA decomposition implemented with a linear autoencoder and our adversarial objective with nonlinear dependency predictors and probe networks.

Implementation details. We train methods (2) to (4) for 10,000 steps. We generate 512 observations $\mathbf{X}$ by sampling from the uniform latent factors $\mathbf{v}$ at every iteration. The encoder of the linear autoencoder is implemented with a projection $\boldsymbol{W}^T \in \mathbb{R}^{3\times 2}$ and the decoder uses the same matrix $\boldsymbol{W}$. For the nonlinear autoencoder experiments introduced below, we use a three-layer residual autoencoder with a hidden dimension of 16. The autoencoders are optimized with Adam at a learning rate of 10^{-2}. For the decorrelation baseline, the covariance penalty is added directly to the encoder's loss, with no separate dependence networks. For `ADM`, the dependence branch is optimized at $4 \cdot 10^{-3}$ for linear dependency predictors or $4 \cdot 10^{-2}$ for nonlinear dependency predictors and probes, with $k = 16$ dependence branch updates per encoder update. The reconstruction term is weighted by 0.02 and the dependence term by $\lambda = 1$, which prevents the encoder from trading independence for a marginally lower reconstruction error. Reconstruction errors and dependence measures for the nonlinear autoencoder configurations are evaluated on a held-out set of $16,384$ observations.

The PCA implementations from (1), (2), and (3) find, respectively:

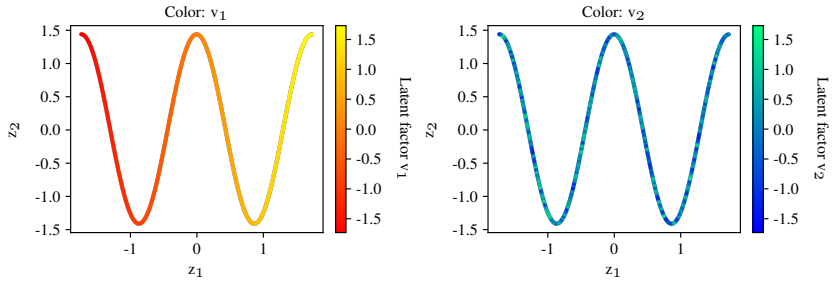
$$W_{\text{PCA},1} \approx \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix} \quad W_{\text{PCA},2} \approx \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \end{bmatrix} \quad W_{\text{PCA},3} \approx \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix} \tag{10}$$

In line with theory, the three PCA implementations extracted the first two observed variables x_1 and x_2. For implementations (1) to (3), the total explained variance for the representations z is approximately equal to 29.5, and the covariance matrices are close to diagonal. However, the learned latent dimensions are not independent, with a distance correlation of $\mathrm{dCor}(\mathrm{z}_1, \mathrm{z}_2) \approx 0.25$. The methods accurately reconstruct the first two observed variables x_1 and x_2, and achieve an average reconstruction error of 1 for x_3. It should be emphasized that this average error corresponds to the variance of x_3.

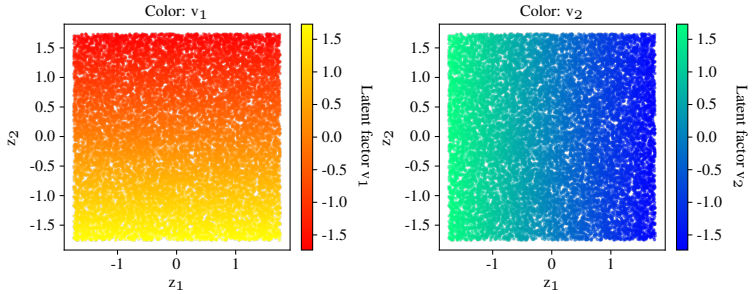
The representation $\mathbf{z}$ for experiment (3) is shown in Figure 5a. The abscissa and ordinate depict the learned latent variables z_1 and z_2, while the color of the predictions indicates the true value of the latent factors v_1 (left figure) and v_2 (right figure). The distributions align with our previous finding and illustrate that the PCA implementation did not encode the latent factor v_2.

On the contrary, the PICA implementation (4) finds:

$$W_{\text{PICA}} \approx \begin{bmatrix} 0.0 & 0.0 & -1.0 \\ -1.0 & 0.0 & 0.0 \end{bmatrix} \tag{11}$$



(a) Linear autoencoder and linear *dependency predictors* (PCA).



(b) Linear autoencoder, two-layer dependency predictors and probes (PICA).

Figure 5: Learned representations $\mathbf{z}$. The colors indicate the value of the original latent factors v_1 (left) and v_2 (right).

The average reconstruction error is 4.513, corresponding to the variance of the second dimension. Furthermore, the distance correlation between the two learned representations is $\mathrm{dCor}(\mathrm{z}_1, \mathrm{z}_2) \approx 0.009$. These empirical results validate that the PICA reduction captured the two independent latent factors v_1 and v_2. Furthermore, the method could not reconstruct x_2 because it uses a linear autoencoder. The distribution of the learned representations $\mathbf{z}$ is shown in Figure 5b.

A.2 Nonlinear Principal and Independent Component Analysis

The linear autoencoder results above illustrate that the PICA constraint can decrease the explained variance when the maximum-variance uncorrelated representation is not independent. We next study whether this reconstruction cost is intrinsic to PICA or a consequence of the linear autoencoder by extending to a nonlinear architecture.

The Non-Linear Principal Component Analysis (NLPCA) extends PCA to non-linear transformations. It was introduced by Kramer (1991) as an autoencoder that learns the identity mapping. The architecture included an intermediate "bottleneck" representation, forcing the network to learn low-dimensional data representations. The original architecture was a four-layer feed-forward neural network with sigmoid activation functions. Following this definition, one can learn an NLPCA reduction by training an encoder $f_{\boldsymbol{\theta}} : \mathbb{R}^l \to \mathbb{R}^d$ and a decoder $g_\xi : \mathbb{R}^d \to \mathbb{R}^l$ to minimize the reconstruction error:

$$\min_{\boldsymbol{\theta},\xi} \quad \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}^{(i)} - g_\xi(f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))\|_2^2 \tag{12}$$

One may assume that a solution with minimal reconstruction error should exhibit low redundancy in the bottleneck representation. Yet, Equation (12) does not explicitly push the bottleneck representation to have uncorrelated dimensions. Therefore, starting from the encoder-decoder architecture from the Equation (12), we add the `ADM` objective by following the development from Section 5.1:

$$\min_{\boldsymbol{\theta},\xi} \quad \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{x}^{(i)} - g_\xi(f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}))\|_2^2 + \lambda \mathcal{L}_{\mathrm{adm}}(\boldsymbol{z}^{(i)}; \boldsymbol{\phi}, \boldsymbol{\psi}) \tag{13}$$

where $\boldsymbol{z}^{(i)} = f_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)})$ is the bottleneck representation. We denote this extension as *Non-Linear Principal and Independent Component Analysis* (NLPICA). It extends the NLPCA reduction by adding an objective that pushes the representations to have minimally interdependent dimensions.

Note that the solution to the NLPICA reduction is not unique. In fact, research on nonlinear independent component analysis (NLICA) has demonstrated that there are numerous ways to transform independent variables while preserving statistical independence (Darmois, 1951; Jutten & Karhunen, 2004). These transformations can involve complex mixing functions that yield representations that may be challenging to interpret or exhibit undesirable properties in certain applications. For examples of mixing transformations, see (Taleb & Jutten, 1999). Despite this limitation, the problem can be made identifiable again with additional assumptions such as temporal non-stationarity (Hyvarinen & Morioka, 2016), a conditionally factorized prior distribution over the latent variables of a VAE (Khemakhem et al., 2020) or constraining the function class with constraints on their partial derivatives (Buchholz et al., 2022).

Table 4: Reconstruction and dependence on Example 2 for the six autoencoder/dependence configurations. Columns $\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3$ report the reconstruction error of each input. The nonlinear dependence branch drives dCor to approximately zero for both linear and nonlinear autoencoders.

Autoencoder	Dependence branch	Per-input recon. error			Cor	dCor
		x_1	x_2	x_3		
linear	decorrelation	0.000	0.000	1.001	0.01	0.25
	ADM linear	0.000	0.000	1.006	0.01	0.25
	ADM nonlinear	0.000	4.513	0.000	0.00	**0.01**
nonlinear	decorrelation	0.001	0.001	0.001	0.00	0.25
	ADM linear	0.001	0.001	0.001	0.01	0.26
	ADM nonlinear	0.000	0.000	0.000	0.00	**0.01**

We evaluate the NLPCA and NLPICA extensions on Example 2 using the nonlinear autoencoder described above. The results are summarized in Table 4 and visualized in Figure 6. With a nonlinear autoencoder, both the decorrelation objective and ADM with linear dependency predictors reconstruct all three observed variables almost perfectly, with per-input reconstruction errors of approximately 0.001. However, the learned bottleneck dimensions remain dependent, with distance correlations of approximately 0.25 and 0.26, respectively. Thus, accurate reconstruction does not imply independence, and these settings recover an NLPCA rather than an NLPICA solution.

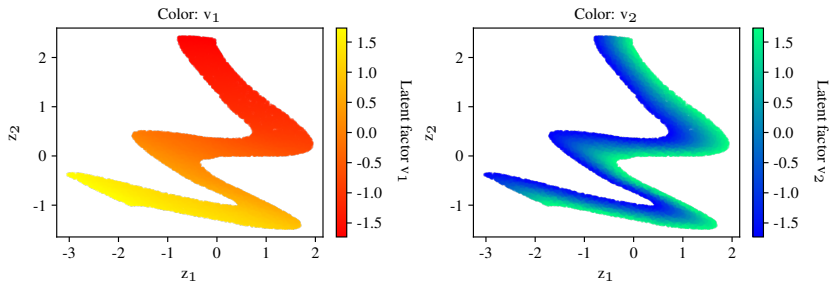
In contrast, combining the nonlinear autoencoder with the nonlinear dependence branch yields an NLPICA reduction. Indeed, the three observed variables are reconstructed almost perfectly while the distance correlation between the two learned dimensions decreases to approximately 0.01. In this setting, the nonlinear dependence branch prevents the bottleneck from encoding the same latent factor in multiple dimensions, and the nonlinear decoder can reconstruct both x_1 and its cosine x_2 from the same bottleneck dimension. The learned representations are shown in Figure 6.

Taken together, these results demonstrate that ADM with a nonlinear dependence branch allows the two latent factors to be separated, while decorrelation alone is insufficient. They also show that the reconstruction cost associated with enforcing independence depends on the autoencoder's expressiveness: with a linear autoencoder, enforcing independence reduces the explained variance when the maximum-variance uncorrelated projection is dependent, whereas a nonlinear autoencoder can recover the independent factors without increasing the reconstruction error.

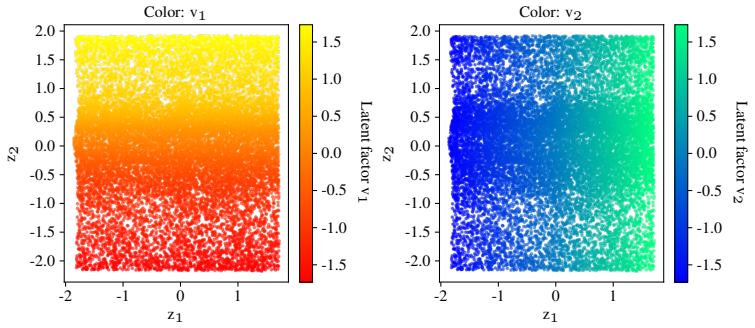
B Open Questions and Future Directions

This paper laid the foundations for learning mutually independent embedding dimensions, but several important questions and applications remain to be studied.

Compression-accessibility tradeoff. As illustrated with the self-supervised learning application, learning highly compressed, information-dense representations can limit the efficacy of simple downstream heads. Because eliminating all inter-dimensional dependencies scatters the semantic structure across a warped manifold, using these features for downstream tasks such as classification



(a) Nonlinear autoencoder with linear *dependency predictors* (NLPCA).



(b) Nonlinear autoencoder with nonlinear *dependency predictors* and probe networks (NLPICA).

Figure 6: Learned two-dimensional representations $\mathbf{z}$ for the nonlinear autoencoder configurations. Within each row, the two panels show the same points colored by the true latent factor v_1 (left) and v_2 (right). The NLPCA reduction reconstructs the observations accurately but retains dependence between the learned dimensions, whereas the NLPICA reduction separates the two latent factors.

will likely require deeper, non-linear decoders. Optimizing purely for compression may thus come at the cost of feature accessibility. Studying this tradeoff is an interesting open question.

Controllable capacity. ADM currently relies on a fixed, pre-defined number of embedding dimensions. This may, however, not always be ideal. Consider, for instance, the case where more embedding dimensions are available than latent factors; the method may split a single concept across several dimensions (for instance, encoding an integer in $\{1, \ldots, 16\}$ across four binary dimensions $\{0, 1\}^4$). One potential extension would be to adapt ADM to have a controllable capacity, for instance by optimizing for using a minimal number of dimensions, similar to the idea in Burgess et al. (2018).

Applications. The mechanism from ADM may benefit several application domains, including:

- *Open-set recognition:* This task requires a classifier to accurately categorize instances from known training classes while identifying novel, unseen classes during inference (Scheirer et al., 2012). While existing approaches primarily focus on defining compact decision boundaries, success also depends on representations retaining information beyond class-specific cues. Our classification results (Section 6.2) suggest that reducing redundancy promotes richer, more transferable representations, potentially supporting compositional generalization and improving the detection of unknown classes.
- *Multi-modal learning:* In vision-language models, ADM could be applied to separate shared and modality-specific components across modalities, improving alignment and fusion by explicitly structuring the latent space into complementary subspaces.
- *Domain adaptation:* Similarly, the independence objective provides a mechanism to separate domain-invariant from domain-specific information into two sets of features, with the potential to make domain shifts more explicit in the representation space.

Note that in multi-modal learning and domain adaptation, ADM would operate across sets of embeddings rather than across embedding dimensions.

C Proofs

C.1 Proof of Theorem 4.1

The proof proceeds by solving the optimization problem in stages: first for the predictors $\boldsymbol{\phi}$, second for the probes $\boldsymbol{\psi}$, and finally for the encoder θ.

Optimization of the Predictors. Fix the encoder θ (and thus the distribution of embeddings $\mathbf{z}$) and the probes $\boldsymbol{\psi}$. The value function is a sum of non-negative terms, so we can minimize each term independently with respect to the corresponding ϕ_i.

Consider the i-th MSE term: $\min_{\phi_i} \mathbb{E}\left[(\psi_i(\mathbf{z}_i) - \phi_i(\mathbf{z}_{-i}))^2\right]$. We seek for the min-cost function $\phi_i^*(\mathbf{z}_{-i})$. By the fundamental property of conditional expectation in L^2 spaces (Williams, 1991), the function that minimizes the L^2 distance to a target random variable $y = \psi_i(\mathbf{z}_i)$ is the orthogonal projection of y onto the subspace of functions measurable w.r.t. $\mathbf{z}_{-i}$. This projection is exactly the conditional expectation:

$$\phi_i^*(\mathbf{z}_{-i}) = \mathbb{E}[\psi_i(\mathbf{z}_i) \mid \mathbf{z}_{-i}] \tag{14}$$

Value at the optimum. By the definition of conditional variance:

$$\mathbb{V}[\psi_i(\mathbf{z}_i) \mid \mathbf{z}_{-i}] = \mathbb{E}\left[(\psi_i(\mathbf{z}_i) - \mathbb{E}[\psi_i(\mathbf{z}_i) \mid \mathbf{z}_{-i}])^2 \mid \mathbf{z}_{-i}\right] \tag{15}$$

If we take the expectation of the conditional variance over $\mathbf{z}_{-i}$ and use the Law of Total Expectation, we get:

$$\mathbb{E}[\mathbb{V}[\psi_i(\mathbf{z}_i) \mid \mathbf{z}_{-i}]] = \mathbb{E}\left[\mathbb{E}\left[(\psi_i(\mathbf{z}_i) - \mathbb{E}[\psi_i(\mathbf{z}_i) \mid \mathbf{z}_{-i}])^2 \mid \mathbf{z}_{-i}\right]\right] \tag{16}$$

$$= \mathbb{E}\left[(\psi_i(\mathbf{z}_i) - \mathbb{E}[\psi_i(\mathbf{z}_i) \mid \mathbf{z}_{-i}])^2\right] \tag{17}$$

We now substitute ϕ_i^* into the objective. Given that ϕ_i^* is the conditional mean, the MSE is thus the expected conditional variance, i.e., the *unexplained variance*:

$$\min_{\phi_i} \mathcal{L}_i = \mathbb{E}\left[(\psi_i(\mathbf{z}_i) - \mathbb{E}[\psi_i(\mathbf{z}_i) \mid \mathbf{z}_{-i}])^2\right] \tag{18}$$

$$= \mathbb{E}\left[\mathbb{V}\left[\psi_i\left(\mathbf{z}_i\right) \mid \mathbf{z}_{-i}\right]\right] \tag{19}$$

Finally, using the Law of Total Variance and imposing the standardization constraint $\mathbb{V}\left[\psi_i\left(\mathrm{z}_i\right)\right]=1$, we find:

$$\mathbb{E}\left[\mathbb{V}\left[\psi_i\left(\mathrm{z}_i\right) \mid \mathbf{z}_{-i}\right]\right]=\mathbb{V}\left[\psi_i\left(\mathrm{z}_i\right)\right]-\mathbb{V}\left[\mathbb{E}\left[\psi_i\left(\mathrm{z}_i\right) \mid \mathbf{z}_{-i}\right]\right] \tag{20}$$

$$=1-\mathbb{V}\left[\mathbb{E}\left[\psi_i\left(\mathrm{z}_i\right) \mid \mathbf{z}_{-i}\right]\right] \tag{21}$$

Optimization of the Probes. The adversaries $(\boldsymbol{\phi}, \boldsymbol{\psi})$ cooperate to minimize the value function. Minimizing the loss with respect to ψ_i is equivalent to maximizing the subtracted term:

$$\min_{\psi_i}\left(1-\mathbb{V}\left[(\mathbb{E}[\psi_i(\mathrm{z}_i) \mid \mathbf{z}_{-i}]]\right)\right. \iff \max_{\psi_i} \mathbb{V}\left[\mathbb{E}[\psi_i(\mathrm{z}_i) \mid \mathbf{z}_{-i}]\right] \tag{22}$$

subject to $\mathbb{E}[\psi_i(\mathrm{z}_i)]=0$ and $\mathbb{V}[\psi_i(\mathrm{z}_i)]=1$.

The square of the Maximal Correlation Coefficient (Gebelein, 1941; Rényi, 1959) between a variable u and a vector $\mathbf{v}$, denoted $\mathrm{mCor}^2(\mathrm{u}, \mathbf{v})$, is defined as:

$$\mathrm{mCor}^2(\mathrm{u}, \mathbf{v})=\sup_{\phi, \psi}(\mathrm{Cor}(\psi(\mathrm{u}), \phi(\mathbf{v})))^2=\sup_{\psi} \mathbb{V}[\mathbb{E}[\psi(\mathrm{u}) \mid \mathbf{v}]] \quad \text{s.t.} \quad \mathbb{V}[\psi(\mathrm{u})]=1 \tag{23}$$

where the supremum is taken over all zero-mean, unit-variance L^2 functions.

Therefore, the inner optimization for each dimension i computes exactly the maximal correlation between z_i and the rest of the vector $\mathbf{z}_{-i}$:

$$\max_{\psi_i} \mathbb{V}\left[\mathbb{E}\left[\psi_i\left(\mathrm{z}_i\right) \mid \mathbf{z}_{-i}\right]\right]=\mathrm{mCor}^2\left(\mathrm{z}_i, \mathbf{z}_{-i}\right) \tag{24}$$

The value of the game after the full inner optimization is:

$$\min_{\boldsymbol{\psi}, \boldsymbol{\phi}} \mathcal{L}=\frac{1}{d} \sum_{i=1}^{d}(1-\mathrm{mCor}_i^2(\mathrm{z}_i, \mathbf{z}_{-i})) \tag{25}$$

Optimization of the Encoder. The outer loop maximizes the encoder parameters θ:

$$\max_{\theta} \frac{1}{d} \sum_{i=1}^{d}\left(1-\mathrm{mCor}_i^2\left(\mathrm{z}_i, \mathbf{z}_{-i}\right)\right) \Longleftrightarrow \min_{\theta} \sum_{i=1}^{d} \mathrm{mCor}_i^2\left(\mathrm{z}_i, \mathbf{z}_{-i}\right) \tag{26}$$

Since mCor^2 is a squared correlation, we have $0 \leq \mathrm{mCor}^2 \leq 1$. The sum is minimized (and the original objective maximized to 1) if and only if:

$$\forall i \in\{1, \ldots, d\}, \quad \mathrm{mCor}_i^2\left(\mathrm{z}_i, \mathbf{z}_{-i}\right)=0 \tag{27}$$

A fundamental property of the Maximal Correlation Coefficient (Rényi, 1959) is that it vanishes if and only if the variables are independent.

$$\mathrm{mCor}_i^2\left(\mathrm{z}_i, \mathbf{z}_{-i}\right)=0 \Longleftrightarrow \mathrm{z}_i \perp \mathbf{z}_{-i} \tag{28}$$

Indeed, if $\mathrm{mCor}^2=0$, then $\mathbb{V}\left[\mathbb{E}\left[\psi_i\left(\mathrm{z}_i\right) \mid \mathbf{z}_{-i}\right]\right]=0$ for all valid ψ_i. Since expectations are zero, this implies $\mathbb{E}\left[\psi_i\left(\mathrm{z}_i\right) \mid \mathbf{z}_{-i}\right]=0$ almost surely for all square-integrable ψ_i. By the definition of conditional independence, this implies the distribution of z_i does not depend on $\mathbf{z}_{-i}$.

Since this holds for all i, each coordinate z_i is independent of the rest. Iteratively applying this factorization to the joint density yields $p(\boldsymbol{z})=\prod_i p(z_i)$, i.e., mutual independence of all dimensions.

□

C.2 Proof of Corollary 4.2

At the optimum, $\mathrm{z}_i \perp\!\!\!\perp \mathbf{z}_{-i}$ (Theorem 4.1), so $\psi_i(\mathrm{z}_i) \perp\!\!\!\perp \mathbf{z}_{-i}$ and the optimal predictor is constant:

$$\phi_i^*(\mathbf{z}_{-i})=\mathbb{E}[\psi_i(\mathrm{z}_i) \mid \mathbf{z}_{-i}]=\mathbb{E}[\psi_i(\mathrm{z}_i)]=0, \tag{29}$$

where the last equality follows from the zero-mean standardization of the probe outputs. Substituting into the objective and using the unit-variance constraint gives, for every dimension,

$$\mathbb{E}\left[(\psi_i(\mathrm{z}_i)-\phi_i^*(\mathbf{z}_{-i}))^2\right]=\mathbb{E}\left[\psi_i(\mathrm{z}_i)^2\right]=\mathbb{V}[\psi_i(\mathrm{z}_i)]=1, \tag{30}$$

so the average reconstruction error over the d dimensions equals 1.

□

D ALGORITHM

The complete training procedure for ADM is summarized in Algorithm 1.

Algorithm 1 Training Algorithm for Adversarial Dependence Minimization (ADM)

Require: Encoder f_θ; probe networks $\{\psi_j\}_{j=1}^d$ with parameters $\boldsymbol{\psi}$; dependency predictors $\{\phi_j\}_{j=1}^d$ with parameters $\boldsymbol{\phi}$; number of critic steps k; minibatch size n

for number of training iterations **do**

▷ *Update the dependence branch* $(\boldsymbol{\phi}, \boldsymbol{\psi})$*: minimize prediction error*

  for k steps **do**

    Sample a minibatch $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\}$ from the dataset

    Compute representations $\boldsymbol{z}^{(i)} \leftarrow f_\theta(\boldsymbol{x}^{(i)})$ for all i ▷ stop gradient w.r.t. θ

    Compute batch statistics: $\mu_j = \frac{1}{n}\sum_i z_j^{(i)}, \quad \sigma_j = \sqrt{\frac{1}{n-1}\sum_i (z_j^{(i)} - \mu_j)^2}$ for all j

    Standardize representations: $z_j^{(i)} \leftarrow (z_j^{(i)} - \mu_j) \,/\, \sigma_j$ for all i, j

    Apply probe networks: $\tilde{z}_j^{(i)} = \psi_j(z_j^{(i)})$ for all i, j

    Compute predictions: $\hat{z}_j^{(i)} = \phi_j\left(z_1^{(i)}, \dots, z_{j-1}^{(i)}, z_{j+1}^{(i)}, \dots, z_d^{(i)}\right)$ for all i, j

    Standardize probe outputs: $\tilde{z}_j^{(i)} \leftarrow \left(\tilde{z}_j^{(i)} - \bar{\tilde{z}}_j\right) / \tilde{\sigma}_j$ for all i, j

    Compute loss $\mathcal{L}_{\text{dep}} = \dfrac{1}{nd}\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{d}\left(\tilde{z}_j^{(i)} - \hat{z}_j^{(i)}\right)^2$

    Update $(\boldsymbol{\phi}, \boldsymbol{\psi})$ by gradient descent on $\mathcal{L}_{\text{dep}}$

  end for

▷ *Update the encoder* θ*: maximize prediction error*

  Sample a minibatch $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(n)}\}$ from the dataset

  Compute representations $\boldsymbol{z}^{(i)} \leftarrow f_\theta(\boldsymbol{x}^{(i)})$ for all i

  Compute batch statistics μ_j, σ_j as above for all j

  Standardize representations: $z_j^{(i)} \leftarrow (z_j^{(i)} - \mu_j) \,/\, \sigma_j$ for all i, j

  Apply probe networks: $\tilde{z}_j^{(i)} = \psi_j(z_j^{(i)})$ for all i, j

  Compute predictions: $\hat{z}_j^{(i)} = \phi_j\left(\boldsymbol{z}_{-j}^{(i)}\right)$ for all i, j ▷ stop gradient w.r.t. $\boldsymbol{\phi}, \boldsymbol{\psi}$

  Standardize probe outputs: $\tilde{z}_j^{(i)} \leftarrow \left(\tilde{z}_j^{(i)} - \bar{\tilde{z}}_j\right) / \tilde{\sigma}_j$ for all i, j

  Compute loss $\mathcal{L}_{\text{adm}} = \dfrac{1}{nd}\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{d} 1 - \left(\tilde{z}_j^{(i)} - \hat{z}_j^{(i)}\right)^2$

  Update θ by gradient descent on $\mathcal{L}_{\text{adm}}$

end for

E EXTENDED RELATED WORK

SSL and optimality of uncorrelatedness. Several studies in SSL highlighted the importance of uncorrelatedness. Specifically, Johnson et al. (2022) showed that contrastive learning approximates a positive-pair kernel, and applying Kernel PCA to this kernel yields orthogonal representations that are optimal for linear fine-tuning. More broadly, Zhai et al. (2023) demonstrated that the optimal d-dimensional encoder aligns with the top-d eigenspace of the augmentation-induced kernel operator. These findings suggest that optimal representations should have orthogonal dimensions — a property our approach explicitly optimizes for, as independence implies uncorrelatedness. Li et al. (2021) built upon this idea and extended BYOL (Grill et al., 2020) with the Hilbert-Schmidt Independence Criterion (HSIC), a kernel-based measure of dependence between probability distributions. However, unlike our method, this method's decorrelation effect depends on the choice of kernel. Finally, Shwartz Ziv & LeCun (2024) discussed SSL from an information-theoretic standpoint, highlighting the importance of both compression and invariance through the multi-view assumption (Sridharan & Kakade, 2008).

Input-output mutual information maximization. Another line of work on dimensionality reduction is based on the infomax principle (Linsker, 1988; Bell & Sejnowski, 1995), which aims to maximize the mutual information (MI) between the input data and the neural network's output to learn informative representations. Belghazi et al. (2018) introduced Mutual Information Neural Estimation (MINE), a first estimate of the MI between high-dimensional continuous random variables using neural networks. Based on MINE, DeepInfoMax (Hjelm et al., 2019) learns representations by maximizing the input-output MI, maximizing the MI between global and local representations, and matching the output to a uniform prior with adversarial learning. Similarly to our work's objective, MI is an information-theoretic measure of the information shared by random variables. However, unlike those methods, our algorithm minimizes dependencies among output dimensions rather than maximizing a proxy for the input-output MI. Furthermore, our algorithm is not limited to pairwise dependencies.

Dependence estimation. Measures of dependence or correlation have been extensively studied across foundational and modern contexts. Rényi (1959) introduced axioms for dependence measures, proposing maximal correlation as a key metric. Building on this, Hsu et al. (2019) used neural networks to model relationships among features via correspondence analysis, whereas our work minimizes rather than merely measures or analyzes dependencies. More recently, Xu & Zheng (2024) provided a framework for learning features by encoding statistical dependencies in a structured manner, but without solving a task-specific objective. Other notable areas of research that studied measures of correlation include Canonical Correlation Analysis (Hotelling, 1936) and its nonlinear extensions (Lai & Fyfe, 2000; Shotaro, 2001; Vinokourov et al., 2002; Andrew et al., 2013), Alternating Conditional Expectation (Breiman & Friedman, 1985), Deep Hirschfield-Gebelein-Rényi (Grari et al., 2021), etc.

F Ablation Studies

F.1 Dependency Predictors' Complexity

We investigate the impact of the architectural capacity of dependency predictors, specifically their width and depth, on the resulting decorrelation. We follow the same experimental setting as in the convergence analysis on TinyImageNet (Section 6.1).

Table 5 reports correlation metrics on the validation set. Increasing the hidden dimension consistently reduces both correlation measures. Notably, even relatively small hidden sizes (e.g., 4) already yield very low correlation levels, suggesting that strong decorrelation can be achieved with minimal predictor capacity.

In contrast, increasing depth does not lead to consistent improvements. One possible explanation is that deeper predictors may be more difficult to optimize under the same training regime, although we leave a more detailed investigation of optimization dynamics to future work.

Table 5: Impact of the width and depth of dependency predictors on correlation for TinyImageNet. Both correlation metrics are computed on the validation set. dCor is estimated on a subset of 5,000 samples using the first 64 dimensions.

depth	hidden	Cor	dCor
2	4	0.0095	0.0354
2	8	0.0092	0.0294
2	16	0.0091	0.0276
2	32	0.0085	0.0188
2	64	0.0083	0.0148
2	128	0.0082	0.0146
3	32	0.0090	0.0257

G Independence and Correlation

G.1 Pairwise Independence without Mutual Independence

Example 4. Let two random variables x_1 and x_2 be drawn from uniform distributions in the interval $[0, 1]$. Then, define the random variable $x_3 = x_1 + x_2 - \lfloor x_1 + x_2 \rfloor$ where $\lfloor \cdot \rfloor$ denotes the integral part of the value (e.g., $\lfloor 3.14 \rfloor = 3$).
The random variable x_3 is uniformly distributed on $[0, 1]$ (see Figure 7) and the pairs $\{x_1, x_2\}$, $\{x_1, x_3\}$ and $\{x_2, x_3\}$ are pairwise independent. Still, the variables are not mutually independent since the uncertainty of x_3 is zero when the random vector $\mathbf{x}_{-3} = [x_1, x_2]^T$ is known.

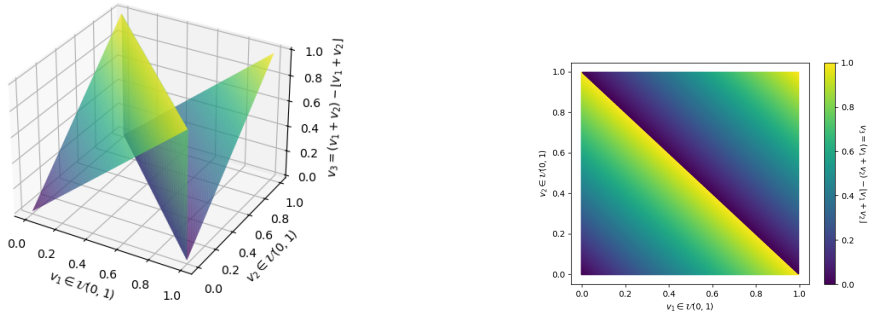


Figure 7: Illustration of Example 4: the random variables x_1, x_2 and x_3 are all mutually dependent despite being all pairwise independent.

G.2 Mean Independence without Full Independence

In this section, we provide a concrete example of a continuous bivariate distribution where x_2 is mean-independent of x_1 but not fully independent. This example illustrates a scenario where predictability minimization (Schmidhuber et al., 1996) would not succeed, even if our standardized formulation were adopted.

Example 5. Let x_1 be a continuous random variable uniformly distributed on the interval $[1, 2]$. We define the conditional distribution of x_2 given x_1 as a normal distribution centered at zero with a variance scaling quadratically with x_1:

$$x_2 \mid x_1 \sim \mathcal{N}(0, x_1^2). \tag{31}$$

To establish mean independence, we verify that the conditional expectation $\mathbb{E}[x_2 \mid x_1]$ equals the marginal expectation $\mathbb{E}[x_2]$. By definition, the conditional mean is invariant to x_1: $\mathbb{E}[x_2 \mid x_1] = 0$.

Applying the law of total expectation yields the marginal expectation:

$$\mathbb{E}[x_2] = \mathbb{E}_{x_1}[\mathbb{E}_{x_2|x_1}[x_2 \mid x_1]] = \mathbb{E}_{x_1}[0] = 0. \tag{32}$$

Since $\mathbb{E}[x_2 \mid x_1] = \mathbb{E}[x_2] = 0$, x_2 is mean-independent of x_1.

However, x_1 and x_2 are not fully independent. Full independence requires all conditional moments to be invariant to the conditioning variable. The conditional variance of x_2 given x_1 is:

$$\mathbb{V}[x_2 \mid x_1] = x_1^2. \tag{33}$$

Because the conditional variance depends on the value of x_1, the joint density function does not factorize into the product of the marginals. Consequently, full statistical independence does not hold.

G.3 Distance Correlation

Distance correlation (Székely et al., 2007) is a non-negative coefficient that characterizes both linear and nonlinear correlations between random vectors. Let x_1 and x_2 be two random vectors with

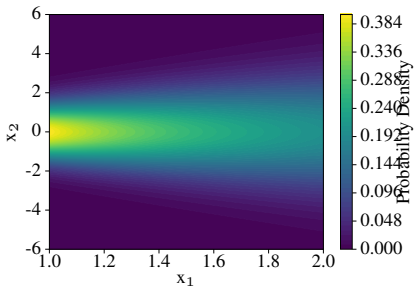


Figure 8: Illustration of Example 5: joint density $f(x_1, x_2)$ with mean independence without full statistical independence due to heteroscedastic variance.

finite first moments, their respective characteristic functions be denoted ψ_{x_1} and ψ_{x_2}, and their joint characteristic function be denoted $\psi_{\mathrm{x}_1,\mathrm{x}_2}$. Distance covariance measures the distance between their joint characteristic function and the product of the marginal characteristic functions:

$$\mathcal{V}^2(\mathrm{x}_1, \mathrm{x}_2) = \int_{\mathbb{R}^{p+q}} \left|\psi_{\mathrm{x}_1,\mathrm{x}_2}(t,s) - \psi_{\mathrm{x}_1}(t)\psi_{\mathrm{x}_2}(s)\right|^2 w(t,s)dtds \tag{34}$$

where $w(t,s)$ is a positive weight function and characteristic functions are $\psi_{\mathrm{x}}(t) = \mathbb{E}\left[e^{it\mathrm{x}}\right]$. Analogous to Pearson correlation, the squared distance correlation dCor^2 is defined by: $\mathrm{dCor}^2(\mathrm{x}_1, \mathrm{x}_2) = \mathcal{V}^2(\mathrm{x}_1, \mathrm{x}_2)/\sqrt{\mathcal{V}^2(\mathrm{x}_1, \mathrm{x}_1)\mathcal{V}^2(\mathrm{x}_2, \mathrm{x}_2)}$ if $\mathcal{V}^2(\mathrm{x}_1, \mathrm{x}_1)\mathcal{V}^2(\mathrm{x}_2, \mathrm{x}_2) > 0$ and 0 otherwise.

We refer to the original paper (Székely et al., 2007) for the empirical estimation of distance correlation and to the library (Ramos-Carreño & Torrecilla, 2023) for its implementation.

H Nearest Neighbors Visualization

We visualize the nearest neighbors for the self-supervised approach described in Section 5.3. Figure 9 shows the predicted nearest neighbors for 10 randomly sampled validation images from the ImageNet-1k dataset. The left-most image is the query image, and its nearest neighbors are the training samples whose representations have the highest cosine similarity to the query's representation. The figure demonstrates that the nearest neighbors visually resemble the query images.

I Detailed Experimental Setups

We provide here a detailed description of the training settings and hyperparameters to facilitate the reproducibility of our experimental results.

The encoder and dependence branch are trained alternately, following the algorithm presented in Appendix D. Epochs are counted relative to the encoder, so the dependence branch loops through the dataset k times per encoder epoch.

Dependence branch. The dependence branch is always trained with the same optimizer and schedulers as the encoder. The default dependency predictor is a two-layer fully connected network with a hidden dimension of 32 and an intermediate GELU (Hendrycks & Gimpel, 2016) activation function. There is no activation function at the network's output. Similarly, the default probe network is a two-layer fully connected network with a hidden dimension of 16 and an intermediate GELU activation.

Variance regularization. We set the standard deviation hinge loss coefficient to a value of 2 in all experiments, as we found the exact hyperparameter value to have minimal impact.

I.1 Experimental Setup: Convergence Analysis

The encoder is a ResNet-18 backbone with no projection head. We used the SGD optimizer with momentum 0.9, learning rate 0.8, batch size 256, and no weight decay. The learning rate follows a

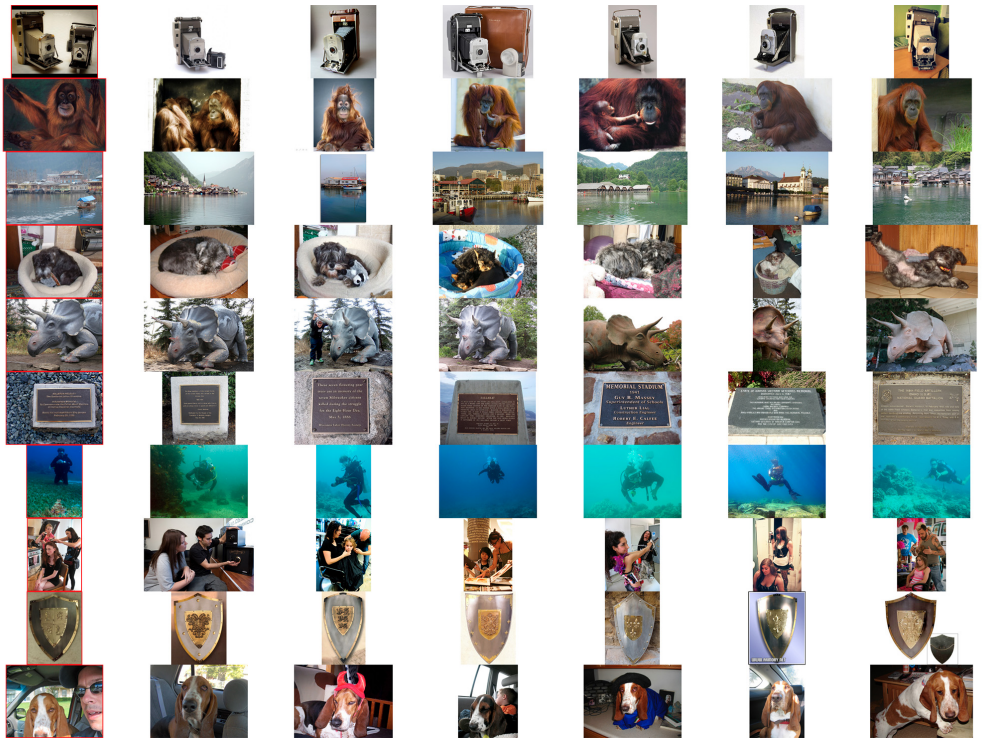


Figure 9: Nearest neighbors visualization for our SSL model trained on the ImageNet dataset. The nearest neighbors visually resemble the query images (highlighted in red).

cosine decay schedule (Loshchilov & Hutter, 2016) with 10 epochs of linear warmup. The dependence branch is trained with a ratio of $k = 16$ steps with a learning rate of 1.6. The model is trained on the TinyImageNet dataset for 100 epochs without data augmentation, and images are normalized using ImageNet mean and per-channel standard deviation.

Decorrelation baseline. The baseline is trained with the same strategy, except the `ADM` loss is replaced by a simple decorrelation loss that penalizes correlations between feature dimensions, similarly to the approach from BarlowTwins (Zbontar et al., 2021). More specifically, given a batch of features $\boldsymbol{z} \in \mathbb{R}^{b \times d}$, we first standardize each dimension, then compute the empirical covariance matrix $C = \boldsymbol{z}^\top \boldsymbol{z}/b$ and minimize the mean squared value of its off-diagonal entries:

$$\mathcal{L}_{\text{decorr}} = \frac{1}{d(d-1)} \sum_{i \neq j} C_{ij}^2.$$

This objective encourages the learned representation to have an approximately diagonal covariance structure.

I.2 Experimental Setup: CLEVR-4

Data augmentations. We apply two data augmentations during training: random horizontal flipping and random cropping. For random cropping, we keep at least 60% of the image area, and then resize the images to 224×224 pixels. We do not apply color alterations, as *color* is one of the taxonomies.

Training. The adversarial approach is trained with two-layer dependency predictors and probes. The encoder is a ResNet-18 backbone with no projection head. We used the SGD optimizer with momentum 0.9, learning rate 0.2, batch size 256, and weight decay $2 \cdot 10^{-5}$. The learning rate follows a cosine decay schedule (Loshchilov & Hutter, 2016) with 10 epochs of linear warmup. The dependence networks are trained with a ratio of $k = 1$ steps with a learning rate of 0.8. The baseline classifier follows the same strategy, except it only uses a cross-entropy loss.

I.3 Experimental Setup: ImageNet SSL

We trained two-layer dependency predictors with no probes. The encoder is a ResNet-50 backbone with a three-layer fully connected projection head with a hidden dimension of 4096, an output dimension of 512, ReLU activations, and intermediate BatchNorm layers. We used the LARS optimizer (You et al., 2017) with a momentum of 0.9, a base learning rate of 1.5 with linear scaling rule (Goyal, 2017), a batch size of 1024, and a weight decay of 10^{-4}. The learning rate follows a cosine decay schedule (Loshchilov & Hutter, 2016) with 10 epochs of linear warmup. The dependence branch is trained with a ratio of $k = 4$ steps with a base learning rate of 6. The model is trained on the ImageNet dataset for 300 epochs and follows the same data augmentations as in (Grill et al., 2020).

Linear evaluation. We followed standard procedure and trained a linear classifier on top of the frozen representations from the backbone. We used the SGD optimizer with a learning rate of 1.5, a weight decay of 10^{-6}, a batch size of 256, and trained for 100 epochs. The learning rate follows a cosine decay schedule (Loshchilov & Hutter, 2016). We applied two data augmentations during training: random horizontal flipping with $p = 0.5$ and random cropping by keeping at least 8% of the image area, followed by resizing to 224 $\times$ 224 pixels. During the evaluation, the images were resized so that the shorter side was 256 pixels, then center-cropped to 224 $\times$ 224 pixels.