Post-Training Augmentation Invariance

Anonymous authors
Paper under double-blind review

Abstract

This work develops a framework for post-training augmentation invariance, in which our goal is to add invariance properties to a pretrained network without altering its behavior on the original, non-augmented input distribution. We define this notion precisely and additionally introduce augmented encoders, which are probabilistic encoders that formalize augmentation-based encoding processes and that serve as our fundamental object of study. We introduce two optimal transport-based losses for augmented encoders, namely, Markov-Wasserstein minimization and Wasserstein correlation maximization, and we demonstrate empirically that both losses can be used to train lightweight, one-hidden-layer MLP adapter networks E_θ that, when appended to the latent space of a pretrained network F, do indeed lead to (approximate) post-training augmentation invariance. For example, on STL10 with F = DINOv2 features, the composite network $C \circ E_\theta \circ F$, where C is a linear classifier, achieves 90% classification accuracy on arbitrarily rotated images, whereas a network of the form $C \circ F$ without the adapter E_θ drops to 71% accuracy. Similarly, we can boost noise-invariant classification results from 62% up to nearly 80%. Significantly, we obtain these results with no fine-tuning (the weights of F remain frozen throughout), and our methods introduce little corruption to the original features, since E_θ acts nearly isometrically on the non-augmented latent distribution. In contrast, we show that adapter networks trained with alternative candidate losses, specifically SimCLR and HSIC maximization, produce uncompetitive classification results and fundamentally corrupt the original latent space.

1 Introduction

Large-scale pretrained models have become the foundation of modern machine learning pipelines. For vision, models such as vision transformers (Dosovitskiy, 2020), CLIP (Radford et al., 2021), DINO (Caron et al., 2021), and masked autoencoders (He et al., 2022) are trained on massive datasets and provide general-purpose features that tend to transfer well to a variety of downstream tasks, including classification, object detection, feature matching, segmentation, and more. However, pretrained models may not be invariant to augmentations that could be critical for various applications. For example, motion estimation or feature matching for satellite or medical images may require features that are invariant to rotations, or perhaps to affine or perspective transformations, more generally. We may also need strong robustness to noise or adversarial perturbations, both of which may be regarded as non-invertible augmentations that introduce information loss. However, any given model is not necessarily guaranteed to satisfy these requirements.

The motivating question of this work, then, is as follows: **Can we make a pretrained model invariant to specific augmentations post-training without corrupting its existing capabilities?** This problem, which we call **post-training augmentation invariance**, is distinct from general representation learning and poses unique challenges. Retraining foundation-scale models to be invariant to various augmentations is computationally prohibitive and risks degrading the learned representations, which may depend on particular augmentation pipelines for success, such as the central use of crops in many self-supervised learning frameworks. Additionally, Kumar et al. (2022) show that standard fine-tuning methods can distort pretrained features, sometimes significantly. They develop an effective method for combining linear probe methods with full fine-tuning, but we will show in this work that, at least for the goal of augmentation invariance, even simpler methods can dramatically improve performance without any fine-tuning at all. That is, we leave the weights of the pretrained network completely frozen and unchanged and instead only append lightweight adapter networks to the latent space.

1.1 Contributions

In this work, we provide a comprehensive framework and initial solution to the problem of post-augmentation invariance. Though our formal definitions are math-heavy and require machinery from measure theory and optimal transport, we highlight the following experimental results to emphasize the practical value of our work: On STL10 with $F = \text{DINOv2}$ features, the composite network $C \circ E_\theta \circ F$, where C is a linear classifier and where E_θ is one of our proposed, lightweight adapter networks, achieves 90% classification accuracy on arbitrarily rotated images, whereas a network of the form $C \circ F$ without the adapter E_θ drops to 71% accuracy. Similarly, we can boost noise-invariant classification results from 62% up to nearly 80%. We obtain these results with no fine-tuning (the weights of F remain frozen throughout), and, perhaps most importantly, we achieve these significant increases in accuracy without corrupting the original feature space: E_θ acts nearly isometrically on the non-augmented latent distribution.

Our exact contributions are as follows:

1. We formalize the problem of post-training augmentation invariance through the notion of (t, μ_X, F, V)-invariance (Definition 3.3), which requires that an adapter E_θ makes the composite network $E_\theta \circ F$ invariant to an augmentation t while preserving the structure of the pretrained latent distribution $F_\sharp \mu_X$ up to a map V in some admissible class of maps $\mathcal{V}$.
2. We introduce augmented encoders (Section 3.1), which are probabilistic encoders that formalize the process of randomly augmenting and then encoding an input, as a fundamental object of study and as a general framework for augmentation-based learning, independent of any specific training objective.
3. We propose two optimal transport-based training objectives for augmented encoders that can be used for post-training augmentation invariance, namely, Markov-Wasserstein minimization (Section 3.3.1), which directly enforces equality in a generalized L^p metric between the augmented encoder and the pretrained model, and Wasserstein correlation maximization (Section 3.3.2), which maximizes the Wasserstein correlation of the joint distribution induced by the augmented encoder.
4. We demonstrate empirically (Section 4) on STL10 with pretrained ResNet50 (SwAV) and vision transformer (DINOv2) networks that simple, one-hidden-layer MLPs E_θ trained on our proposed losses make the composite network $E_\theta \circ F$ approximately (t, μ_X, F, V)-invariant for an approximate, local isometry V. By contrast, we show that two other candidate losses, namely, SimCLR and a Hilbert-Schmidt Independence Criterion (HSIC) maximization loss, produce uncompetitive invariance results and significantly corrupt the pretrained latent space.

1.2 Related Work

1.2.1 (Self-Supervised) Representation Learning

Many self-supervised learning objectives, such as SimCLR (Chen et al., 2020), MoCo (He et al., 2020), SwAV (Caron et al., 2020), BYOL (Grill et al., 2020), and DINO (Caron et al., 2021), learn representations by maximizing agreement between differently augmented views of the same input. These approaches naturally induce (approximate) invariance to their chosen augmentations and have achieved remarkable success in pretraining. However, as we demonstrate empirically, applying a SimCLR loss (which we take to be a suitable stand-in for contrastive losses, more generally) in a post-training setting fundamentally alters the geometry of the original latent space, making the pretrained features subsequently unusable. This is unsurprising: Contrastive methods cluster semantically similar examples, which is ideal for learning representations from scratch, but inappropriate when the goal is to preserve existing structure.

Other work in representation learning more directly employs variants of the InfoMax principle first introduced by Linsker (1988) to maximize mutual information (MI) between input and latent distributions; see, for example, Bachman et al. (2019); Hjelm et al. (2018); Hu et al. (2017); Oord et al. (2018); Rezaabad & Vishwanath (2020); Zhao et al. (2019), and, in particular, see Tschannen et al. (2019) for a unified perspective on many of these works. MI estimation, of course, is computationally difficult in high-dimensions, and, in practice, neural estimators, as defined in Belghazi et al. (2018) or Poole et al. (2019), for example, are used to estimate lower bounds on the quantity. However, as shown in McAllester & Stratos (2020), high-confidence, distribution-free lower bounds of any type have exponential sample complexity in the size of the bound. More strikingly, Tschannen et al. (2019) show that maximizing MI does not necessarily lead to effective representations for various downstream tasks and that the success of MI-based objectives involves a subtle interplay between encoder architectures and MI estimators.

One of our initial motivations for undertaking this work was to explore the potential suitability of Wasserstein dependence, or its normalized version, Wasserstein correlation, as a general drop-in replacement for mutual information in MI-based representation learning methods. While Ozair et al. (2019) showed that Wasserstein dependence maximization can be used for general representation learning when combined with additional regularization terms, we found in our own investigations that Wasserstein dependence alone, or, more specifically, Wasserstein correlation maximization, cannot, in fact, be used as generic replacement for MI-maximization, as explained further in Remark 3.4. Briefly, Wasserstein correlation maximization inverts the dichotomy introduced before: It (approximately) preserves the metric structure of the input distribution, which is therefore fatal when the goal is to cluster data according to semantic similarity, but ideal when trying to leave a pretrained latent space intact.

To test whether other measures of statistical dependence might perform similarly to our Wasserstein correlation maximization objective, we also consider maximizing the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2005), which has been shown to be effective for self-supervised learning (Li et al., 2021). As will be shown experimentally, though, the HSIC loss, like the SimCLR loss, is also uncompetitive and corrupts the pretrained latent space even more dramatically. This suggests that the geometric properties of the optimal transport-based objectives are playing a non-trivial role in achieving post-training augmentation invariance.

1.2.2 Fine-Tuning and Model Adaptation

The overall aim of our work is related to fine-tuning (Kumar et al., 2022; Goyal et al., 2023; Wortsman et al., 2022; Xin et al., 2024), but our methods leave the pretrained network frozen and unaltered. We seek instead to develop adapter networks, which have been explored for various use-cases, such as improved semantic segmentation or domain generalization (Dukler et al., 2023; Ji et al., 2025; Kim et al., 2021; Rebuffi et al., 2017; Xu et al., 2023). To our knowledge, no one has developed a general framework comparable to ours for adapter networks that can achieve (approximate) post-training augmentation invariance for arbitrary augmentation types.

1.2.3 Optimal Transport-Based Methods

Our work is related, more broadly, to the many uses of optimal transport in machine learning. Transport-based methods have been influential in a number of areas, including generative modeling (Arjovsky et al., 2017; Bousquet et al., 2017; Kolouri et al., 2019; Kunkel & Trabs, 2024; Liutkus et al., 2019; Nadjahi et al., 2019; Nguyen et al., 2022; Tolstikhin et al., 2017; Wu et al., 2019), domain adaptation (Courty et al., 2016; 2017; Lee et al., 2019; Shen et al., 2018), and adversarial robustness (Bhagoji et al., 2019; Bai et al., 2023; Levine & Feizi, 2020; Nguyen et al., 2023; Wong et al., 2019; Wu et al., 2020). Our use of Wasserstein correlation, in particular, builds on a growing body of work on transport-based statistical dependency measures (Liu et al., 2022; Móri & Székely, 2020; Mordant & Segers, 2022; Nies et al., 2021; Wiesel, 2022). These ideas have been applied to feature selection (Li et al., 2023) and representation learning (Ozair et al., 2019; Xiao & Wang, 2019), but to our knowledge, no one has used Wasserstein correlation, nor our other transport-based loss, for invariance learning, particularly in a post-training setting where additional structure-preservation constraints are needed.

1.3 Paper Organization

The rest of the paper is organized as follows: In Section 2, we present all background material needed for the rest of the paper. The material on optimal transport is standard, but the material on Wasserstein correlation (Section 2.2) and Markov-Wasserstein kernels (Section 2.3) is likely less well-known. Our main contributions are presented in Section 3, and our experimental results are presented in Section 4. Finally, limitations of the current framework and suggestions for future work are given in Section 5.

2 Background

2.1 Wasserstein Distance

We first review the definition of the Wasserstein distance in the general setting of complete and separable metric spaces, which we call Polish metric spaces for short. This material should be standard for anyone familiar with Villani's introductory book Villani (2003). We also recall the definition of the sliced Wasserstein distance, which we use for our computational implementation of the Wasserstein correlation maximization objective discussed in Sections 2.2 and 3.3.2. .

Formally, we need the following regularity assumption to ensure that the Wasserstein distance (of order p) is indeed an actual metric, as opposed to a generalized metric. We state the condition for completeness, though we gloss over these kinds of nuances in practice.

Definition 2.1 *Let (X, d_X) be Polish metric space, let Σ_X denote the Borel σ-algebra on X induced by d_X, and let $1 \leq p < \infty$. A Borel probability measure μ on (X, Σ_X) (also called a distribution) has* ***finite* pth *moment*** *if*

$$\int_X d_X^p(x_0, x)\mu(dx) < \infty \tag{1}$$

for some (and hence, by the triangle inequality, any) $x_0 \in X$. We denote the set of all such probability measures by $\mathcal{P}_p(X)$, and we denote the collection of all Borel probability measures by $\mathcal{P}(X)$.

Definition 2.2 *Let $\mu \in \mathcal{P}_p(X)$, and let $\nu \in \mathcal{P}_p(Y)$. A* ***coupling*** *of μ and ν is a probability measure $\pi \in \mathcal{P}(X \times Y)$ with first and second marginals equal to μ and ν, respectively. That is, for all $A \in \Sigma_X$ and all $B \in \Sigma_Y$,*

$$\pi(A \times Y) = \mu(A), \quad \textit{and} \quad \pi(X \times B) = \nu(B).$$

The collection of all couplings between μ and ν is denoted by $\Pi(\mu, \nu)$. We will also denote the first and second marginals of π by π^1 and π^2.

A coupling can be seen as a relaxation of a transport map, which is a measurable map $T : X \to Y$ satisfying $T_\sharp\mu = \nu$ where $T_\sharp\mu$ is the **pushforward** of μ defined by

$$(T_\sharp\mu)(B) = \mu(T^{-1}(B)) \tag{2}$$

for all $B \in \Sigma_Y$. Note also that the space of couplings is always non-empty, since, at the least, it always contains the product measure.

Definition 2.3 *Let $\mu \in \mathcal{P}_p(X)$ and $\nu \in \mathcal{P}_p(Y)$. The* ***product measure****, or tensor product, of μ and ν is the joint probability measure $\mu \otimes \nu$ in $\mathcal{P}(X \times Y)$ defined on test functions $\phi \in C_b(X \times Y)$ by*

$$\int_{X\times Y} \phi(x, y)(\mu \otimes \nu)(dx, dy) = \int_Y \left(\int_X \phi(x, y)\mu(dx)\right) \nu(dy). \tag{3}$$

Alternatively, we can set

$$(\mu \otimes \nu)(A \times B) = \mu(A)\nu(B) \tag{4}$$

for $A \times B \in \Sigma_X \times \Sigma_Y$. Since we always assume that $X \times Y$ is standard Borel, the above equality determines $\mu \otimes \nu$ uniquely on all of $\Sigma_{X\times Y}$.

Definition 2.4 *Let (X, d_X) be a Polish metric space, and let $1 \leq p < \infty$. The **Wasserstein metric of order** $\mathbf{p}$*

$$W_{p,d_X} : \mathcal{P}_p(X) \times \mathcal{P}_p(X) \to \mathbb{R}_{\geq 0} \tag{5}$$

is defined by

$$W_{p,d_X}(\mu, \nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{X \times X} d_X^p(x_1, x_2) \pi(dx_1, dx_2) \right)^{1/p}. \tag{6}$$

*We call the pair $(\mathcal{P}_p(X), W_{p,d_X})$ the **Wasserstein space of order** $\mathbf{p}$ associated with (X, d_X).*

When the context is clear, we denote the Wasserstein metric by W_p alone. Likewise, we will sometimes denote the so-called ground metric d_X by d alone. Remarkably, the Wasserstein space is itself a Polish metric space whenever (X, d_X) is.

Proposition 2.5 *Let (X, d_X) be a complete and separable metric space. Then, the Wasserstein space $(\mathcal{P}_p(X), W_{p,d_X})$ is also complete and separable.*

Proof 1 *See Proposition 7.1.5 in Ambrosio et al. (2008).*

Informally, the Wasserstein construction is said to lift any ground metric d_X to a metric W_{p,d_X} on the space of (sufficiently regular) Borel probability measures over X. Further, the Wasserstein metric encodes the ground metric, in the sense that $W_{p,d_X}(\delta_x, \delta_y) = d_X(x, y)$, where δ_x is the usual Dirac mass defined by

$$\delta_x(E) = \begin{cases} 1 & \text{if } x \in E \\ 0 & \text{if } x \notin E \end{cases} \tag{7}$$

for $E \in \Sigma_X$; see Villani (2003) for more details. In passing, we observe that Definition 2.4 is still valid with a lower semicontinuous cost function c on X in place of d_X. Custom cost functions are indeed part of the appeal of optimal transport-based distances and are worth further study, though in the present work we restrict our attention to metrics.

To define Wasserstein correlation in Subsection 2.2, we need to equip the product space $X \times Y$ with a metric so that joint distributions can be viewed as elements of a Wasserstein space. We choose to work with the product metric for this.

Definition 2.6 *Let (X, d_X) and (Y, d_Y) be Polish metric spaces, and let $1 \leq p < \infty$. Then, the **product metric of order** $\mathbf{p}$*

$$d_{p,X \times Y} : (X \times Y) \times (X \times Y) \to \mathbb{R}_{\geq 0} \tag{8}$$

is defined by

$$d_{p,X \times Y}\big((x_1, y_1), (x_2, y_2)\big) = \left(d_X^p(x_1, x_2) + d_Y^p(y_1, y_2)\right)^{1/p}. \tag{9}$$

We'll denote this metric by d_{XY}, and we assume that p is always chosen to match the order of the Wasserstein metric.

The Wasserstein space $(\mathcal{P}_p(X \times Y), W_{p,d_{XY}})$ is the usual Wasserstein space where the ground metric is now d_{XY}. Further, it is elementary to check that any coupling π of $\mu \in \mathcal{P}_p(X)$ and $\nu \in \mathcal{P}_p(Y)$ has finite pth moment with respect to d_{XY}. That is, $\Pi(\mu, \nu) \subseteq \mathcal{P}_p(X \times Y)$.

2.1.1 Sliced Wasserstein Distance

Extensive work has gone into finding efficient algorithms for computing Wasserstein distances. Entropic regularization and slicing techniques are two of the most common approaches (Bonneel et al., 2015; Cuturi, 2013), and in our experiments, we use the latter.

Definition 2.7 *Let ω denote uniform distribution on the unit sphere $\mathbb{S}^{d-1} \subseteq \mathbb{R}^d$. Then, the **sliced Wasserstein distance of order** $\mathbf{p}$*

$$SW_p : \mathcal{P}_p(\mathbb{R}^d) \times \mathcal{P}_p(\mathbb{R}^d) \to \mathbb{R}_{\geq 0} \tag{10}$$

is defined by

$$SW_p(\mu,\nu) = \left(\int_{\theta:\mathbb{S}^{d-1}} W_p^p\big((P_\theta)_\sharp\mu, (P_\theta)_\sharp\nu\big)\omega(d\theta) \right)^{1/p}, \tag{11}$$

where P_θ is the linear form given by $P_\theta(x) = \theta^T x$.

See Bonnotte (2013) for a proof that the sliced Wasserstein distance is an actual metric. To compute sliced distances, we use the closed-form solution to the Wasserstein distance in the case of 1-dimensional distributions on $\mathbb{R}$. For $\mu \in \mathcal{P}_p(\mathbb{R})$, define the cumulative distribution function (CDF) $F_\mu : \mathbb{R} \to [0,1]$ by $F_\mu(x) = \mu\big((-\infty, x]\big)$, and define the quantile function $F_\mu^{-1} : [0,1] \to \mathbb{R}$ by

$$F_\mu^{-1}(t) = \inf\{x : F_\mu(x) \geq t\}. \tag{12}$$

Then, as shown in Rachev & Rüschendorf (2006),

$$W_p(\mu,\nu) = \left(\int_0^1 \big|F_\mu^{-1}(t) - F_\nu^{-1}(t)\big|^p dt \right)^{1/p}. \tag{13}$$

Letting $F_{\theta,\mu}^{-1}$ denote the quantile function of $(P_\theta)_\sharp\mu$, we have

$$SW_p(\mu,\nu) = \left(\int_{\mathbb{S}^{d-1}} \left(\int_0^1 \big|F_{\theta,\mu}^{-1}(t) - F_{\theta,\nu}^{-1}(t)\big|^p dt \right)^{1/p} \omega(d\theta) \right)^{1/p}. \tag{14}$$

In practice, given empirical distributions, we compute the one-dimensional Wasserstein distance by sorting the points of those distributions and then taking their average Euclidean distance. Our exact computational implementation follows the work of Nguyen et al. (2022).

2.2 Wasserstein Correlation

Next we introduce one of our fundamental quantities of interest, namely, Wasserstein correlation. We define the Wasserstein correlation maximization loss in Section 3.3.2. First, recall that the mutual information of random variables X and Y is defined as $I(X;Y) = D_{KL}(P_{XY} \| P_X \otimes P_Y)$, where D_{KL} is KL-divergence, and where P_{XY} and $P_X \otimes P_Y$ are, respectively, the joint and product distributions of the pair (X,Y). Wasserstein dependence is the natural optimal transport-based analog of this quantity.

Definition 2.8 *Let (X, d_X) and (Y, d_Y) be Polish metric spaces, and let $(X \times Y, d_{XY})$ be the standard product space equipped with the product metric of order p for some $1 \leq p < \infty$. Then, the* ***Wasserstein dependence of order*** **p** *on $\mathcal{P}_p(X \times Y)$*

$$WD_p : \mathcal{P}_p(X \times Y) \to \mathbb{R}_{\geq 0} \tag{15}$$

is defined by

$$WD_p(\pi) = W_{p,d_{XY}}(\pi, \pi^1 \otimes \pi^2), \tag{16}$$

where $\pi^1 \in \mathcal{P}_p(X)$ and $\pi^2 \in \mathcal{P}_p(Y)$ are the first and second marginals of π.

We cannot train models directly on Wasserstein dependence maximization, since the encoder can trivially hack the objective by spreading the latent codes arbitrarily far apart. One possibility is to add a prior-matching term in the latent space. This is the approach taken by information-maximizing variational autoencoders (Rezaabad & Vishwanath, 2020; Zhao et al., 2019). In these and similar models, the support of the latent distribution remains bounded by being constrained to remain close, in some divergence or metric, to a prior distribution, typically a Gaussian. In our case, we work instead with a normalized version of dependence, namely, Wasserstein correlation, which constrains the variance of the latent codes.

Definition 2.9 *Take the same setup as in Definition 2.8, and let $1 \leq p < \infty$ be an integer. Then, the* ***Wasserstein correlation of order*** $\mathbf{p}$ *on $\mathcal{P}_p(X \times Y)$*

$$WC_p : \mathcal{P}_p(X \times Y) \to \mathbb{R}_{\geq 0} \tag{17}$$

is defined by

$$WC_p(\pi) = \frac{WD_p(\pi)}{\left(WD_p(\pi_D^1) \cdot WD_p(\pi_D^2)\right)^{1/p}}, \tag{18}$$

where π_D^1 is the diagonal distribution on $\mathcal{P}_p(X \times X)$ given by $\pi_D^1(A \times B) = \pi^1(A \cap B)$ for $A \times B \in \Sigma_{X \times X}$, and similarly for $\pi_D^2 \in \mathcal{P}_p(Y \times Y)$.

In the above, the terms in the denominator are self-variance terms that measure how spread out the marginal distributions are, and the dependence scores are computed in the Wasserstein spaces $(\mathcal{P}_p(X \times X), W_{p,d_{XX}})$ and $(\mathcal{P}_p(Y \times Y), W_{p,d_{YY}})$, respectively. We leave this implicit to reduce notational clutter. These normalization terms guarantee that Wasserstein correlation is bounded between zero and one.

For computational implementations, we simply substitute the sliced Wasserstein distance for the ordinary Wasserstein distance in the definitions above. Naturally, we call the resulting quantities **sliced Wasserstein dependence** and **sliced Wasserstein correlation**, and we use SD_p and SC_p for the corresponding notation. In practice, we work (in the non-augmented case) with empirical joint distributions corresponding to pairs $(x_i, E_\theta(x_i))$, and we approximate product distributions by shuffling coordinates, as in Li et al. (2023). Altogether then, we have

$$\begin{aligned} &\mathrm{SC}_p\left(\frac{1}{N}\sum_{i=1}^{N} \delta_{\left(x_i, E_\theta(x_i)\right)}\right) \\ &\quad = \frac{SW_p\Big(P^N(X,Z), P^N_{\sigma_{XZ}}(X,Z)\Big)}{\Big(SW_p\big(P^N(X,X), P^N_{\sigma_{XX}}(X,X)\big) \cdot SW_p\big(P^N(Z,Z), P^N_{\sigma_{ZZ}}(Z,Z)\big)\Big)^{1/p}}, \end{aligned} \tag{19}$$

where

$$P^N(X,Z) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(x_i, E_\theta(x_i)\right)} \;\text{ and }\; P^N_{\sigma_{XZ}}(X,Z) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(x_{\sigma_1(i)}, E_\theta(x_{\sigma_2(i)})\right)} \tag{20}$$

for random permutations σ_k on $\{1, \ldots, N\}$ and for $Z = E_\theta(X)$. The other quantities are defined similarly. Also, note that it is sufficient to only shuffle one coordinate to approximate the product distribution, but the estimate is generally better when shuffling both.

2.3 Markov-Wasserstein Kernels

Abstractly, we model encoders as Markov-Wasserstein kernels, which we introduce next. Ordinary Markov kernels are defined as follows.

Definition 2.10 *Let (X, Σ_X) and (Y, Σ_Y) be measurable spaces. A* ***Markov kernel*** *from (X, Σ_X) to (Y, Σ_Y) is a map $F : \Sigma_Y \times X \to [0,1]$ satisfying*

1. *$F(-|x) : \Sigma_Y \to [0,1]$ is a probability measure for any fixed $x \in X$.*
2. *The map $x \mapsto F(E|x)$ is Σ_X-measurable for any fixed $E \in \Sigma_Y$.*

Note that the conditional notation $F(-|x)$ above is synonymous with $F(-, x)$.

Equivalently, a Markov kernel can be viewed as a measurable map $F : X \to \mathcal{P}(Y)$, where $\mathcal{P}(Y)$ is equipped with the initial σ-algebra with respect to all evaluation maps $\mathrm{ev}_E : \mathcal{P}(Y) \to [0,1]$, $\mathrm{ev}_E(\mu) = \mu(E)$, for $E \in \Sigma_Y$; see Lemma 3.1 in Kallenberg (2021).

Both of these definitions are purely measure-theoretic, whereas, for our purposes, we would like to work in a metric-enhanced setting. To do this, we impose in Definition 2.10 the additional constraint that each distribution $F(-|x)$ should have finite pth moment. Then, we additionally impose the following regularity condition to equip the resulting collection of Markov kernels with a metric.

Definition 2.11 *Let (X, d_X, μ_X) be a metric measure space (i.e., a Polish metric space equipped with a probability measure μ_X on the Borel σ-algebra induced by d_X), and let (Y, d_Y) be a Polish metric space. A Markov kernel $F : X \to \mathcal{P}_p(Y)$ has* ***finite* p*th moment*** *if*

$$\int_X \left(\int_Y d_Y^p(y_0, y) F(dy|x) \right) \mu_X(dx) < \infty \tag{21}$$

for some (and hence any) $y_0 \in Y$. We let $\mathcal{K}^p_{\mu_X}(X, Y)$ denote the collection of all such Markov kernels.

In Patterson (2021), Patterson shows how to equip $\mathcal{K}^p_{\mu_X}(X, Y)$ with a metric that closely resembles the ordinary Wasserstein metric using a generalized notion of coupling between two Markov kernels, and he additionally shows (in Proposition 5.5) that the metric can be computed as a generalized L^p metric.

Definition 2.12 *Let (X, d_X, μ_X) be a metric measure space, and let (Y, d_Y) be a Polish metric space. The* **$\mathbf{L^p}$ *space*** *$L^p_{\mu_X}(X, Y)$ is the set of μ_X-a.e. equal equivalence classes of measurable functions $f : X \to Y$ satisfying*

$$\int_X d_Y^p\big(y_0, f(x)\big)\mu(dx) < \infty \tag{22}$$

for some (and hence any) $y_0 \in Y$. We say that f is **$\mathbf{L^p}$ *integrable*** *if this condition is met. Further, $L^p_{\mu_X}(X, Y)$ is a metric space when equipped with the* **$\mathbf{L^p}$ *metric***

$$d_{L^p} : L^p_{\mu_X}(X, Y) \times L^p_{\mu_X}(X, Y) \to \mathbb{R}_{\geq 0} \tag{23}$$

defined by

$$d_{L^p}(f, g) = \left(\int_X d_Y^p\big(f(x), g(x)\big)\mu(dx) \right)^{1/p} \tag{24}$$

for $1 \leq p < \infty$.

For the case of $\mathcal{K}^p_{\mu_X}(X, Y)$, in particular, d_Y in the above is taken to be the Wasserstein metric W_{p,d_Y}. This discussion so far applies to Markov kernels as defined in Definition 2.10 with the extra regularity conditions noted before. Similar to the purely measure-theoretic case, though, we can equivalently view a Markov kernel F taking values in a Wasserstein space $\mathcal{P}_p(Y)$ as a measurable map $F : X \to \mathcal{P}_p(Y)$, where $\mathcal{P}_p(Y)$ is equipped with the Borel σ-algebra induced by the Wasserstein metric; see Eikenberry (2023) for details. This will be the perspective we adopt in the present work.

Remark 2.13 *That is, we work with the generalized L^p space $L^p_{\mu_X}(X, \mathcal{P}_p(Y))$ of Wasserstein-valued Markov kernels $F : X \to \mathcal{P}_p(Y)$, which we call the space of* ***Markov-Wasserstein kernels****. Further, we will denote the generalized L^p metric from Definition 2.12 by MW_p, and we call it the* ***Markov-Wasserstein metric****. We define the Markov-Wasserstein minimization loss in Section 3.3.1. Finally, we note that the two approaches to defining Markov-Wasserstein kernels presented here lead to isometrically isomorphic metric spaces; see Theorem 2.25 in Eikenberry (2023).*

Markov-Wasserstein kernels (and also general Markov kernels) are closely connected to joint distributions via disintegrations. The remaining definitions and theorems in this subsection are usually only defined for the measure-theoretic case, but, in fact, everything carries over seamlessly to the metric-enhanced setting.

Definition 2.14 *Let $\gamma \in \mathcal{P}_p(X \times Y)$ with marginals $\gamma^1 = \mu$ and $\gamma^2 = \nu$. Then, a Markov-Wasserstein kernel $F \in L^p_\mu(X, \mathcal{P}_p(Y))$ is a called a* ***disintegration*** *of γ if for all $\phi \in C_b(X \times Y)$,*

$$\int_{X \times Y} \phi(x, y)\gamma(dx, dy) = \int_X \left(\int_Y \phi(x, y) F_x(dy) \right) \mu(dx). \tag{25}$$

In Villani (2003), Villani writes $\int_X \big(\delta_x \otimes F_x\big)\mu(dx)$ for the joint distribution defined by the iterated integral above. As shown in Eikenberry (2023), it is also possible to define this measure as a literal Lebesgue-type integral for Markov-Wasserstein kernels. We will not review the details, since we primarily use this machinery to give a precise mathematical definition for the Wasserstein correlation maximization objective.

The well-known Disintegration Theorem states that disintegrations of (sufficiently nice) joint distributions always exist.

Theorem 2.15 (L$^{\mathbf{p}}$ Disintegration Theorem) *Let $\gamma \in (\mathcal{P}_p(X \times Y), W_{p,d_{XY}})$ with $\gamma^1 = \mu$ and $\gamma^2 = \nu$. Then, there exist almost surely unique Markov-Wasserstein kernels $F \in L^p_\mu(X, \mathcal{P}_p(Y))$ and $G \in L^p_\nu(Y, \mathcal{P}_p(X))$ with*

$$\int_X \big(\delta_x \otimes F_x\big)\mu(dx) = \gamma = \int_Y \big(G_y \otimes \delta_y\big)\nu(dy). \tag{26}$$

Proof 2 *See Theorem 10.4.5 of Bogachev & Ruas (2007) for the classic Disintegration Theorem. Also, see Theorem 3.16 in Eikenberry (2023) for the extension to the L^p case and for an interpretation of this result as an equality of actual integrals.*

We can use disintegrations to give a precise definition of Bayesian inverse maps. Markov-kernel formulations of Bayesian inference are well-developed in probabilistic programming semantics and in category-theoretic approaches to probability; see, for example, Cho & Jacobs (2019); Clerc et al. (2017); Culbertson & Sturtz (2014); Fritz (2020). We again state things for the metric-enhanced case. First, we need to define how to push a measure forward through a Markov, or Markov-Wasserstein, kernel, rather than a deterministic map.

Definition 2.16 *Let $F : X \to \mathcal{P}_p(Y)$ be a Markov-Wassertein kernel, and let $\mu \in \mathcal{P}_p(X)$. The **generalized pushforward** of μ under F, or, alternatively, the **Kleisli composition**[1] of F and μ, is the probability measure $F \odot \mu$ in $\mathcal{P}_p(Y)$ defined by*

$$(F \odot \mu)(B) = \int_X F(B|x)\mu(dx) \tag{27}$$

*for $B \in \Sigma_Y$. Note that if $\gamma = \int_X \big(\delta_x \otimes F_x\big)\mu(dx)$, then $\gamma^2 = F \odot \mu$, and we call this the **marginal likelihood** in Bayesian contexts.*

Definition 2.17 *Let $F \in L^p_\mu(X, \mathcal{P}_p(Y))$, and set $\nu := F \odot \mu$. Then, the **Bayesian inverse** $F^\dagger_\mu$ of F with respect to μ is the ν-almost surely unique Markov-Wasserstein kernel in $L^p_\nu(Y, \mathcal{P}_p(X))$ satisfying*

$$\int_X \big(\delta_x \otimes F(x)\big)\mu(dx) = \int_Y \big(F^\dagger_\mu(y) \otimes \delta_y\big)(F \odot \mu)(dy). \tag{28}$$

Note that $F^\dagger_\mu$ is guaranteed to exist by Theorem 2.15.

Finally, we submit the following as a highly general definition of an autoencoder that captures the statistical essence of many models without any secondary constraints, such as smoothness, determinism, or stricter inversion requirements.

Definition 2.18 *Let $\mu_X \in \mathcal{P}_p(X)$ be a (data) distribution. Then, a **probabilistic autoencoder**, or simply an **autoencoder**, for μ_X is a pair (E, D) of Markov-Wasserstein kernels $E : X \to \mathcal{P}_p(Z)$ and $D : Z \to \mathcal{P}_p(X)$ such that D is a Bayesian inverse of E with respect to μ_X. That is,*

$$\int_X \big(\delta_x \otimes E(x)\big)\mu_X(dx) = \int_Z \big(D(z) \otimes \delta_z\big)(E \odot \mu_X)(dz). \tag{29}$$

(We switch to Z here instead of Y to match the convention for latent spaces.)

[1] The Kleisli terminology comes from the fact that this operation can be viewed as an instance of composition in a Kleisli category. That is, it can be seen as an ordinary pushforward operation into an iterated probability space (i.e., a space of distributions of distributions), followed by an averaging operation. Details on Kleisli categories can be found in Fritz (2020), among other sources, but we will only need the more straightforward definition given here.

In practice, we work with deterministic (auto)encoders (specifically simple MLPs for our tests), which can still be seen as Markov-Wasserstein kernels (or simply Markov kernels) via Dirac embeddings. That is, given a measurable function $f : X \to Z$, we obtain a Markov kernel $F : X \to \mathcal{P}(Z)$ by setting $F(x) = \delta_{f(x)}$. Further, we can place additional integrability or smoothness constraints on f, which, in turn, induce constraints on the corresponding kernel F. Below, we show that even when starting with a deterministic encoder, data augmentation naturally leads to more general, probabilistic encoders as the main object of study.

3 Methods

3.1 Augmented Encoders

We now introduce our fundamental object of study, namely augmented encoders, which are probabilistic encoders that formalize the process of randomly augmenting and then encoding an input. Then, in Section 3.3, we define two loss functions for invariance learning with augmented encoders.

Define an **augmentation** to be a (measurable) map of the form $t : A \times X \to X$, where A is the parameter space, typically taken to be (a subset of) some (potentially factored) Euclidean space $\mathbb{R}^{p_1} \times \cdots \times \mathbb{R}^{p_\ell}$. Note that this includes composite augmentations since we can consider, for example, $t\big((a_1, a_2), x\big) = t_2\big(a_2, t_1(a_1, x)\big)$, and likewise for more general compositions $t_k \circ t_{k-1} \circ \cdots \circ t_1$. Now, let $T = \{t_1, \ldots, t_m\}$ be a collection of m augmentations, let $w = (w_1, \ldots, w_m)$ be a collection of positive weights with $\sum_{j=1}^m w_j = 1$, let $\nu = (\nu_1, \ldots, \nu_m)$ be a collection of distributions $\nu_j \in \mathcal{P}_p(A_j)$ on the parameter spaces A_j, and let $E : X \to Z$ be a deterministic encoder, which we simply take to be a measurable map, though, in practice, it will have more or less smoothness properties. Then, we define the **augmented encoder with respect to** (T, w, ν), denoted simply by $E^T : X \to \mathcal{P}_p(Z)$, by

$$E^T(x) = \sum_{j=1}^m w_j \int_{A_j} \delta_{E(t_j(a,x))} \; \nu_j(da), \tag{30}$$

where, for $x_0 \in X$, the distribution $E^T_\theta(x_0) \in \mathcal{P}_p(Z)$ is defined on test functions $\phi \in C_b(Z)$ by

$$\begin{aligned}\int_Z \phi(z) E^T(x_0)(dz) &= \int_Z \phi(z) \left(\sum_{j=1}^m w_j \int_{A_j} \delta_{E(t(a,x_0))} \nu_j(da) \right)(dz) \\ &= \sum_{j=1}^m w_j \int_{A_j} (\phi \circ E \circ t)(a, x_0) \nu_j(da).\end{aligned} \tag{31}$$

Empirically, if ν_j is approximated with n_j samples as $\frac{1}{n_j}\sum_{k=1}^{n_j} \delta_{a_j^k}$, then $E^T(x_0)$ is approximated as

$$E^T(x_0) \approx \sum_{j=1}^m \sum_{k=1}^{n_j} \frac{w_j}{n_j} \delta_{E(t_j(a_j^k, x_0))}. \tag{32}$$

Additionally, if μ_X is approximated empirically as $\frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, then the generalized pushforward distribution, or latent distribution, $E^T \odot \mu_X$ is approximated by

$$E^T \odot \mu_X \approx \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^{n_j} \frac{w_j}{n n_j} \delta_{E(t_j(a_j^k, x_i))}. \tag{33}$$

In practice, given a batch of data $\{x_i\}_{i=1}^N$, we only sample s times from (32) for each x_i (typically with $s = 3$), and we also allow the augmentation parameters to vary across the batch, giving us

$$E^T \odot \left(\frac{1}{N} \sum_{i=1}^N \delta_{x_i} \right) \approx \frac{1}{sN} \sum_{i=1}^N \sum_{k=1}^s \delta_{z_i^k}, \tag{34}$$

where $\{z_i^1, \dots, z_i^s\}$ is a collection of s independent samples from $E^T(x_i)$.

Strictly speaking, we have not yet guaranteed that E^T is measurable under definition (30). We will bypass a full consideration of this issue here, since, in practice, we only work with empirical distributions. Formally, though, this can be done by applying Fubini-Tonelli analogs in the Markov-Wasserstein setting.

3.2 Augmentation Invariance

Let $\mu_X \in \mathcal{P}_p(X)$ be a distribution, and let $t : A \times X \to X$ be an augmentation. We want encoders that are invariant to t, but we must first define invariance carefully, since literal mathematical invariance may immediately imply collapse of (parts of) the representation space. In particular, if there are **collisions**, namely distinct points x_1 and x_2 in $\text{supp}(\mu_X)$ with either $t(a_1, x_1) = x_2$ for some augmentation parameter a_1, or else $t(a_1, x_1) = t(a_2, x_2)$ for augmentation parameters a_1 and a_2, then exact invariance would imply

$$E(x_1) = E\big(t(a_1, x_1)\big) = E(x_2) \tag{35}$$

in the first case, or, similarly,

$$E(x_1) = E\big(t(a_1, x_1)\big) = E\big(t(a_2, x_2)\big) = E(x_2) \tag{36}$$

in the second. As such, we refine our desired notion of invariance as follows.

Definition 3.1 *Let $\mu_X \in \mathcal{P}_p(X)$ and let $t : A \times X \to X$ be an augmentation with parameter distribution $\nu \in \mathcal{P}_p(A)$. Define the* $\mathbf{t}$-***augmentation set*** *$Aug(x, t)$ of x by*

$$Aug(x, t) = \{t(a, x) : a \in supp(\nu)\}. \tag{37}$$

*Further, define the set of **unique augmentations** $UAug_{\mu_X}(x, t)$ of x with respect to t and μ_X by*

$$UAug_{\mu_X}(x, t) := Aug(x, t) \cap \left(\bigcup_{w \in supp(\mu_X) \setminus \{x\}} Aug(w, t) \right)^c \cap supp(\mu_X)^c. \tag{38}$$

That is, $UAug_{\mu_X}(x, t)$ is the set of all collision-free augmentations of x.

Definition 3.2 *Again, let $\mu_X \in \mathcal{P}_p(X)$ and let $t : A \times X \to X$ be an augmentation with parameter distribution $\nu \in \mathcal{P}_p(A)$. A (deterministic) encoder $E : X \to Z$ is* $(\mathbf{t}, \boldsymbol{\mu}_\mathbf{X})$-***invariant*** *(or invariant with respect to t and μ_X)) if for all $x \in supp(\mu_X)$ and all $y = t(a, x) \in UAug_\mu(x, t)$, we have*

$$E(y) = E\big(t(a, x)\big) = E(x). \tag{39}$$

(We can alternatively say that this condition should hold almost everywhere, but we will ignore subtleties like this.)

This definition specifies that, for the purposes of representation learning, and encoder should only be invariant with respect to collision-free augmentations, as otherwise (partial) collapse of the latent space immediately follows. Since our primary aim is to add additional invariance properties to pretrained models while preserving the structure of the original latent space up to isometry, or at least up to some class of admissible maps, we introduce the following definition.

Definition 3.3 *Let $\mu_X \in \mathcal{P}_p(X)$ be a distribution, let $t : A \times X \to X$ be an augmentation with parameter distribution $\nu \in \mathcal{P}_p(A)$, let $F : X \to Z$ be a (pretrained) encoder, and let $\mathcal{V}(Z, W)$ be an admissible class of maps (e.g., isometries, bi-Lipschitz maps, etc.). An encoder $E : Z \to W$ is* $(\mathbf{t}, \boldsymbol{\mu}_\mathbf{X}, \mathbf{F}, \mathbf{V})$-***invariant*** *if $E \circ F$ is (t, μ_X)-invariant and if $(E \circ F)(x) = (V \circ F)(x)$ for all $x \in supp(\mu_X)$ where $V : Z \to W$ is in $\mathcal{V}(Z, W)$.*

That is, Definition 3.3 specifies that the composite encoder $E \circ F$ should be augmentation invariant with respect to t and the data distribution μ_X, but, additionally, we require that E preserve the (metric) structure of the latent distribution $F_\sharp\mu_X$ up to V. This is the precise definition of what we mean by **post-training augmentation invariance**: adding additional invariants to the latent space of a pretained network F without corrupting the structure of that space.

We stress that this notion of structure preservation has nothing to do with intrinsic, low-dimensional manifold structure in a dataset, nor does it have anything to do with class structure. We want to preserve the literal, point-cloud metric structure of the latent distribution as exactly as possible, since this implies that downstream representation quality on the original distribution is maintained. Also, we note that our first loss, Markov-Wasserstein minimization (see Section 3.3.1), is only defined for encoders $E : Z \to Z$ where the codomain is the same as the domain, and we take $V = \mathrm{id}_Z$. Wasserstein correlation maximization (Section 3.3.2) additionally allows for dimensionality reduction (W different from Z), and we then take $V : Z \to W$ to be an approximate local isometry. In fact, the Wasserstein correlation loss can also be defined without a pretrained network F in the mix at all. In this case, we will say that $E : X \to Z$ is $(\mathbf{t}, \mu_\mathbf{X}, \mathbf{V})$-invariant if E is (t, μ_X)-invariant and if there is some $V \in \mathcal{V}(X, Z)$ with $E(x) = V(x)$ on $\mathrm{supp}(\mu_X)$.

Finally, we extend Definitions 3.2 and 3.3 to collections of augmentations and to augmented encoders as follows. Let $T = \{t_1, \ldots, t_m\}$ be a collection of augmentations $t_j : A_j \times X \to X$ on X with parameter distributions $\nu_j \in \mathcal{P}_p(A_j)$. An encoder $E : X \to Z$ is (T, μ_X)-invariant if it is (t_j, μ_X)-invariant for each $t_j \in T$. We extend the definitions for (T, μ_X, F, V)-invariance and (T, μ_X, V)-invariance in a similar fashion.

For the case of augmented encoders, we say that $E^T : X \to \mathcal{P}_p(Z)$ is (T, μ_X)-invariant if the underlying deterministic encoder $E : X \to Z$ is (T, μ_X)-invariant, which, in turn, implies that $E^T(x) = \delta_{E(x)}$. Similarly, we say that the augmented encoder $(E \circ F)^T : X \to \mathcal{P}_p(W)$ for $E : Z \to W$ deterministic is (T, μ_X, F, V)-invariant if $E \circ F$ is, which implies that $(E \circ F)^T(x) = \delta_{(V \circ F)(x)}$.

3.3 Loss Functions

We now introduce two loss functions for the notions of augmentation invariance defined above, namely, Markov-Wasserstein minimization and Wasserstein correlation maximization.

3.3.1 Markov-Wasserstein Minimization

Interpreting the equality $(E \circ F)^T = \delta_F$ as an equality of Markov-Wasserstein kernels (see Remark 2.13 for a reminder on terminology) gives us our first loss function for augmentation invariance learning. Consider a parameterized collection of deterministic encoders $E_\theta : Z \to Z$, and let $F : X \to Z$ be a frozen, pretrained network. We want $(E_\theta \circ F)^T$ to be $(T, \mu_X, F, \mathrm{id}_Z)$-invariant for augmentations $T = \{t_1, \ldots, t_m\}$. We'll focus here on the case of a single, possibly composite, augmentation $T = \{t\}$.

To preserve the metric structure of the pretrained latent space exactly, we include the identity id_X as a trivial augmentation and set $T = \{\mathrm{id}_X, t\}$ with weights $w = \left(\frac{1}{s+1}, \frac{s}{s+1}\right)$. Then, we can approximate $(E_\theta \circ F)^T$ empirically at x using s samples:

$$(E_\theta \circ F)^T_s(x) = \frac{1}{s+1}\delta_{(E_\theta \circ F)(x)} + \frac{1}{s+1}\sum_{k=1}^{s} \delta_{(E_\theta \circ F)\left(t(a^k_x, x)\right)}. \tag{40}$$

Previously, we used the notation a^k_j to indicate dependence of the augmentation parameters on the jth augmentation. Here, with only one augmentation, we use the notation a^k_x to denote the fact that the augmentation parameters are not necessarily uniform across $\mathrm{supp}(\mu_X)$, but instead come from taking independent samples of $E^T(x)$. When we have a batch $\{x_i\}_{i=1}^N$ with $\mu_X \approx \frac{1}{N}\sum_{i=1}^N \delta_{x_i}$, we will instead write this dependence as a^k_i. Note also that

$$(E_\theta \circ F)^T_s(x) \otimes \delta_{F(x)} = \frac{1}{s+1}\delta_{(E_\theta \circ F)(x)} \otimes \delta_{F(x)} + \frac{1}{s+1}\sum_{k=1}^{s} \delta_{(E_\theta \circ F)(x)} \otimes \delta_{F(x)}. \tag{41}$$

Now, take $p = 2$ and assume that all spaces are standard Euclidean spaces. We use equality (41) together with the fact that couplings with Dirac masses are unique to reduce the Wasserstein computation below to a collection of squared $L2$ terms. Specifically, we define the (squared) Markov-Wasserstein (MaWa) minimization loss as

$$\theta_* = \arg\min_\theta \mathcal{L}_{\mathrm{MaWa}}(\theta) := MW_2^2\big((E_\theta \circ F)^T, \delta_F\big). \tag{42}$$

Then, the empirical batch loss is given by

$$\begin{aligned} & MW_2^2\big((E_\theta \circ F)_s^T, \delta_F\big) \\ &\quad = \int_X W_2^2\big((E_\theta \circ F)_s^T(x), \delta_{F(x)}\big)\mu(dx) \\ &\quad = \int_X \left(\int_{Z\times Z} \|z_1 - z_2\|_2^2 \big((E_\theta \circ F)_s^T(x) \otimes \delta_{F(x)}\big)(dz_1, dz_2)\right)\mu(dx) \\ &\quad = \int_X \left(\frac{1}{s+1}\|(E_\theta \circ F)(x) - F(x)\|_2^2 \right. \\ &\qquad\qquad \left. + \frac{1}{s+1}\sum_{k=1}^{s} \|(E_\theta \circ F)\big(t(a_x^k, x)\big)) - F(x)\|_2^2\right)\mu(dx) \\ &\quad \approx \frac{1}{N}\sum_{i=1}^{N}\left(\frac{1}{s+1}\|(E_\theta \circ F)(x_i) - F(x_i)\|_2^2 \right. \\ &\qquad\qquad \left. + \frac{1}{s+1}\sum_{k=1}^{s} \|(E_\theta \circ F)\big(t(a_i^k, x_i)\big)) - F(x_i)\|_2^2\right). \end{aligned} \tag{43}$$

Thus, the MaWa loss reduces, in the empirical case, to what is essentially an (anchored) mean squared error loss. The first term enforces preservation of F and the second term enforces augmentation invariance. More specifically, if all augmentations are collision free, then (43) reaches a minimum of zero when $E_\theta = E_*$ for a $(T, \mu_X, F, \mathrm{id}_Z)$-invariant encoder $E_* : Z \to Z$, since both terms zero out by definition.

Finally, we note that this loss avoids the collapse problem that plagues a naive mean squared invariance loss involving terms of the form $\|E_\theta\big(t(a,x)\big) - E_\theta(x)\|_2^2$, which can immediately be minimized by mapping everything in the latent space to a single point. The frozen network F acts similarly to dual network designs, serving as an analog to, say, a slowly updating teacher network or to a stop-gradient protected branch of a twin network.

3.3.2 Wasserstein Correlation Maximization

We now present the Wasserstein correlation (WaCo) maximization loss, which achieves invariance results similar to the Markov-Wasserstein minimization framework but that has additional, unique properties. Rather than operate as a loss between networks viewed as elements of a Markov-Wasserstein space, WaCo maximization instead acts on the joint distribution induced by the augmented encoder. Further, besides invariance learning, Wasserstein correlation maximization can alternatively, or simultaneously, be used for dimensionality reduction, with or without the presence of a frozen, pretrained network F.

Remark 3.4 *Crucially, we note that Wasserstein correlation maximization, by itself, is not suitable as a wholly general representation learning method comparable to, say, contrastive methods. The issue is that, by being an approximate (local) isometry on the input distribution, Wasserstein correlation maximization fundamentally cannot cluster points according to semantic similarity in the way that a contrastive method does. On the other hand, this is precisely what makes it suitable for post-training augmentation invariance, whereas other methods fail outright by altering the (metric) structure of the latent space, as demonstrated empirically in Section 4.*

Our setup is as follows. Let $\mu_X \in \mathcal{P}_p(X)$ be a distribution. As before, we focus on the case of a single, possibly composite distribution t, and we again include the identity as a trivial augmentation. That is, we take $T = \{\mathrm{id}_X, t\}$ with weights $w = \left(\frac{1}{s+1}, \frac{s}{s+1}\right)$. We note in passing that, whereas including the identity in the Markov-Wasserstein minimization loss is necessary for strong experimental results, we found that it can be left out of the Wasserstein correlation maximization loss without much degradation. Still, we include it by default.

Markov-Wasserstein minimization requires a frozen, pretrained network $F : X \to Z$ to match against, but for Wasserstein correlation maximization, we have two options. Let $E_\theta : X \to Z$ be a parameterized collection of deterministic encoders. Then, we define the ordinary Wasserstein correlation maximization loss by

$$\theta_* = \arg\max_\theta \mathcal{L}_{\text{WaCo}}(\theta) \coloneqq \text{WC}_p \left(\int_X \big(\delta_x \otimes E_\theta^T(x)\big)\mu_X(dx) \right). \tag{44}$$

That is, we maximize the Wasserstein correlation of the joint distribution induced by the augmented encoder, and there is no requirement that the left and right marginals of this joint distribution live in the same space. (Indeed, recall from Definition 2.9 that all Wasserstein computations here take place in $\mathcal{P}_p(X \times Z)$.) This can be viewed as an objective for (t, μ_X, V)-invariance. Additionally, if T is only the identity, then this loss serves purely to train an encoder that reduces dimensionality while acting as an approximate local isometry on the support of μ_X.

Remark 3.5 *Although the intuition for the MaWa loss is clear, the WaCo loss is more subtle. We leave a theoretical investigation of the WaCo loss for future work, but the general mechanism is related to the following. First, Theorem 2.2 in Wiesel (2022) shows that for the transport-based correlation measure defined there is equal to one if and only if the second marginal is the pushforward of the first marginal under a measurable function. Section 5 of Nies et al. (2021) gives similar results for a variety of transport-based correlation measures. While these results do not apply verbatim to our setting, the idea of using Wasserstein correlation maximization for augmentation invariance is this: In order for the joint distribution induced by the augmented encoder to be far away, in the Wasserstein distance, from the product distribution, subject to normalization constraints, the encoder must bring any augmented views of the same input closer together in the latent space so that the resulting joint distribution is as close as possible to the graph of a measurable function. That is, the encoder must become invariant to the augmentations, and, in practice, we observe that it can accomplish this without representational collapse.*

For the second case of post-training augmentation invariance for a frozen, pretrained network $F : X \to Z$, the loss can be written as

$$\theta_* = \arg\max_\theta \mathcal{L}^F_{\text{WaCo}}(\theta) \coloneqq \text{WC}_p \left(\int_X \big(\delta_{F(x)} \otimes (E_\theta \circ F)^T(x)\big)\mu_X(dx) \right), \tag{45}$$

where $E_\theta : Z \to W$ with W not necessarily the same as Z. That is, computations now take place in $\mathcal{P}_p(Z \times W)$.

Our computational implementations of these losses use sliced Wasserstein distances on empirical distributions.[2] We can approximate the induced joint distribution of a batch $\{x_i\}_{i=1}^N$ with s samples as

$$\begin{aligned} &P_N^s\big(X, E_\theta^T(X)\big) \\ &\quad = \frac{1}{(s+1)N} \sum_{i=1}^N \delta_{\left(x_i, E_\theta(x_i)\right)} + \frac{1}{(s+1)N} \sum_{i=1}^N \sum_{k=1}^s \delta_{\left(x_i, E_\theta(t(a_i^k, x_i))\right)}. \end{aligned} \tag{46}$$

Then, in place of the ordinary WaCo loss (44), we have

$$\theta_* = \arg\max_\theta \mathcal{L}_{\text{SWaCo}}(\theta) \coloneqq \text{SC}_2\Big(P_N^s\big(X, E_\theta^T(X)\big)\Big). \tag{47}$$

[2] While it is well known that sliced distances are an effective approximation of ordinary Wasserstein distances (see, e.g., Bonnotte (2013) or Nadjahi et al. (2019)), we note that to have a fully rigorous connection between sliced Wasserstein correlation (SWC) and ordinary Wasserstein correlation (WC), we would need to justify that SWC maximization is indeed a suitable proxy for WC maximization, similar to how InfoNCE losses can be shown to be a lower bound on mutual information maximization losses. This is left for future work. In any case, we demonstrate empirically that, if nothing else, SWC maximization does indeed serve as a suitable objective for (post-training) augmentation invariance.

Likwise, we define the corresponding sliced version of (45) in the obvious way. Again, our experiments for Wasserstein correlation maximization specifically use these sliced analogs, but, for simplicity, we retain the WaCo naming, instead of switching to SWaCo.

Before proceeding to our experimental results, we make two last remarks. First, when used for dimensionality reduction, the WaCo loss can be expanded to include a reconstruction term for a decoder. The natural loss, at the level of distributions, is given by

$$\mathcal{L}_{\text{dist}}(\phi, \theta) = \alpha W_p\big(\mu_X, (D_\phi \circ E_\theta)_\sharp \mu_X\big) - \beta \mathrm{WC}_p \left(\int_X \big(\delta_x \otimes E_\theta^T(x)\big) \mu_X(dx) \right), \tag{48}$$

and we can formulate a similar loss when reducing and reconstructing from a pretrained latent distribution $F_\sharp \mu_X$. When training with a decoder, we follow Bousquet et al. (2017) and Tolstikhin et al. (2017) and observe that

$$W_c\big(\mu_X, (D_\phi \circ E_\theta)_\sharp \mu_X\big) \leq \int_X c\big(x, (D_\phi \circ E_\theta)(x)\big) \mu_X(dx) \tag{49}$$

for any lower semicontinuous cost function c. We substitute this upper bound for the reconstruction error, as this allows us to compute a simple L^2 reconstruction term in the Euclidean case when $p = 2$. The batch loss is then written as

$$\mathcal{L}(\phi, \theta) = \alpha \frac{1}{N} \sum_{j=1}^{N} \big\| x_j - (D_\phi \circ E_\theta)(x_j) \big\|_2^2 - \beta \mathrm{SC}_2 \Big(P_N^s \big(X, E_\theta^T(X) \big) \Big), \tag{50}$$

Finally, we note that in order for the trained model to be simultaneously invariant to multiple augmentations, one must distinguish between composite versus non-composite augmentations. In general, training for invariance to augmentations $T = \{t_1, \ldots, t_m\}$ will only make the encoder separately invariant to each t_j. For example, if t_1 is rotation and t_2 is translation, then the encoder, at least when restricted to simple architectures, will not yet be invariant to inputs that have been both rotated and translated, as given by either $t = t_2 \circ t_1$ or $t = t_1 \circ t_2$. However, if invariance to non-composite augmentations is sufficient, then instead of working with (48) for $T = \{t_1, \ldots, t_m\}$, one can also consider taking the logarithm of the product of Wasserstein correlation scores—one for each augmentation t_j and the corresponding augmented encoder—since this allows for parallel computation and may be more efficient, depending on the use case.

4 Experiments

To isolate the effect of different training objectives, we employ one-hidden-layer multilayer perceptrons (MLPs) with ReLU activations, deliberately avoiding more complex architectures that involve, for example, convolution, attention, or normalization layers. (Exact details can be found in Appendix A.) This approach better ensures that the structure preservation and invariance properties in the trained models are not the result of inductive biases in the networks themselves. It also shows that the method is extremely lightweight and can be used to modify pretrained networks with little additional training burden.

4.1 Evaluation Framework

Our tests are designed to evaluate (t, μ_X, F, V)-invariance for various choices of F and V. For our primary tests targeting post-training augmentation invariance, we use the STL10 dataset[3] with ImageNet normalization statistics, and we take F to be either a ResNet50 network[4] with a 2048-dimensional latent space trained on the Swapping Assignments between Views (SwAV) objective (Caron et al., 2020), or else a vision

[3] Our choice of STL10 as the main testing ground comes from the need to balance two constraints: realistic feature processing through large pretrained networks and computational efficiency. Image datasets like CIFAR10 are arguably too low-resolution for convincing feature extraction with pretrained networks, specifically large-scale, state-of-the-art models like DINO. Additionally, given limited compute, and given the large number of tests we perform, it is infeasible to do systematic comparisons on ImageNet directly. STL10, we submit, is a reasonable compromise.

[4] Available at `https://github.com/facebookresearch/swav`

transformer[5] trained on the Self-Distillation with No Labels (DINO) objective (Caron et al., 2021), or, more specifically, the DINOv2 version (Oquab et al., 2023), which we still call DINO for simplicity.

In addition to our results for the MaWa minimization loss and the WaCo maximization loss, we compare against two alternative ways one might try to achieve (t, μ_X, F, V)-invariance. Specifically, we use SimCLR (Chen et al., 2020) (denoted by SCLR in our tables for more uniform presentation) as a representative contrastive method, and we use maximization of the Hilbert-Schmidt Independence Criterion (HSIC) as a possible alternative to the WaCo loss.

The SimCLR loss pushes augmented views closer together, which is desirable for invariance, but, as will be seen, when training an encoder E_θ this way on augmented views of the form $F\big(t(a, x)\big)$ for a pretrained network F, the original latent space of F can become badly corrupted, breaking one of our main requirements for post-training augmentation invariance.

For HSIC comparisons, we use the $O(1/N)$ biased estimator from Gretton et al. (2005), which was shown to be an effective loss for representation learning in Li et al. (2021). That is, given i.i.d. samples $\{(x_i, y_i\}_{i=1}^N$ from a joint distribution (X, Y),

$$\mathrm{HSIC}(X, Y) = \frac{1}{(N-1)^2}\mathrm{Tr}(KHLH) \tag{51}$$

for kernel matrices $K_{ij} = k(x_i, x_j)$, $L_{ij} = l(y_i, y_j)$ and centering matrix $H = I - \frac{1}{N}\mathbf{1}\mathbf{1}^T$. Specifically, we train E_θ using HSIC maximization on the joint distribution $P_N^s\big(X, E_\theta^T(X)\big)$ induced by the augmented encoder. However, as is the case with SimCLR, we'll see that this potential alternative is also ill-suited for post-training augmentation invariance. The latent space of F can again become badly corrupted.

4.1.1 Invariance Evaluation

To evaluate invariance, we compare classification accuracy of $C \circ E_\theta \circ F$ versus $C \circ F$ on augmented data for a classifier C trained on non-augmented data. In the first case, for $C \circ E_\theta \circ F$, C is either a linear classifier (LC), or else a nonlinear classifier (NC), specifically a simple, one-hidden-layer MLP with ReLU activation. We call $C \circ F$ the end-to-end classifier (EC) case, and C in this case is a nonlinear MLP with the same total depth and dimensions as $C \circ E_\theta$ for C linear. In all cases, we train on a standard cross-entropy loss for 50 epochs. It is certainly possible to obtain better classification results with longer training times and more fine-tuning, but we are primarily concerned with comparing relative accuracy between networks of the form $C \circ E_\theta \circ F$ and $C \circ F$.

Classification is a more practical test of invariance than direct measurement of whether we have (approximate) equalities $E_\theta\big(t(a, x)\big) = E_\theta(x)$, since in pretrained latent spaces, raw $L2$ distances between points in the same class can actually be quite large, making these calculations uninformative in practice. Classification accuracy instead shows in a task-relevant way that the augmented data lives within the same decision boundaries as the non-augmented data after passing through E_θ.

Finally, all augmentations are taken from the Kornia package (Riba et al., 2020). We use (arbitrary, 360 degree) rotations, affine transformations, noise, and crops as our augmentation types. Full experimental details and Kornia parameters can be found in Appendix A.

Remark 4.1 *What exactly is different from our approach to simply training a classifier on augmented data directly? A classifier C trained on data of the form $\{(F\big(t(a, x_i)\big), label(x_i)\}_{i=1}^N$ would indeed be an invariant classifier, but the latent space of F itself need not be invariant. Indeed, under natural assumptions, we can have $(C \circ F)\big(t(a, x)\big) = (C \circ F)(x)$, but $F(t(a, x))$ need not equal $F(x)$, even approximately. If one only wants an invariant classifier going into logit space $C : Z \to \mathbb{R}^m$ for m classes, then this is acceptable. However, the advantage of having an adapter E_θ that turns $E_\theta \circ F$ into a (t, μ_X, F, V)-invariant encoder is that one then obtains invariance for any downstream task, not just classification. Again, our aim is to alter the latent space of F itself without degrading its existing performance. This is why we train our classifiers on non-augmented data and then evaluate on augmented data: to show that the latent space itself has been*

[5] The dino_vits8 model available at `https://github.com/facebookresearch/dinov2`

reshaped. In particular, evaluating linear probes on the latent space is a standard evaluation protocol in (self-supervised) representation learning.

4.1.2 Structure Evaluation

To evaluate whether or not the original latent space is preserved, we test to what degree E_θ acts isometrically on $F_\sharp \mathrm{supp}(\mu_X)$. Specifically, we consider scatter plots of the form

$$\big\{(\|F(x) - F(y)\|_2, \|(E_\theta \circ F)(x) - (E_\theta \circ F)(y)\|_2)\big\}_{x,y \in \mathrm{supp}(\mu_X)}. \tag{52}$$

That is, we plot $\|(E_\theta \circ F)(x) - (E_\theta \circ F)(y)\|_2$ against $\|F(x) - F(y)\|_2$, and we compute the best linear fit. If E_θ is exactly isometric on the original, non-augmented features, then we'll have a straight line with slope one and intercept zero. Naturally, we don't expect a perfect isometry in practice, so we record the lower and upper Lipschitz constants L_1 and L_2, defined by

$$\begin{aligned} L_1 &:= \min_{x,y \in \mathrm{supp}(\mu_X)} \frac{\|(E_\theta \circ F)(x) - (E_\theta \circ F)(y)\|_2}{\|F(x) - F(y)\|_2}, \text{ and} \\ L_2 &:= \max_{x,y \in \mathrm{supp}(\mu_X)} \frac{\|(E_\theta \circ F)(x) - (E_\theta \circ F)(y)\|_2}{\|F(x) - F(y)\|_2}. \end{aligned} \tag{53}$$

Additionally, we record the slope m and the intercept b of the best least-squares fit to (52), and finally, we record the coefficient of determination

$$R^2 = 1 - \frac{\mathrm{SS}_{\mathrm{res}}}{\mathrm{SS}_{\mathrm{tot}}} \tag{54}$$

to evaluate the predictive power of this linear fit, where $\mathrm{SS}_{\mathrm{res}}$ is the sum of squared residual terms, and where $\mathrm{SS}_{\mathrm{tot}}$ is the sum of squared deviations from the mean. R^2 ranges from zero to one, with zero indicating no predictive power, and one indicating perfect predictive power. In the case of a perfect isometry, we'll have L_1, L_2, m, and R^2 all equal to one with $b = 0$.

These quantities provide the most direct evaluation for the structure-preservation constraint demanded by (t, μ_X, F, V)-invariance where V is a (local, approximate) isometry. That said, our evaluation code also contains optional computations of (normalized) spectral statistics and heat kernel statistics of k-nn feature graphs for a range of possible values of k. These tests are meant to further evaluate how well the encoder E_θ preserves local geometric properties of the latent space, and we found that our trained encoders consistently performed well. Overall, though, these tests were not as informative or illuminating as the direct isometry testing outlined above, so we leave them out to streamline our presentation.

Finally, we stress once more that we are interested in pretrained features taken as a point cloud in a Euclidean space. We are not concerned with structure in the sense of low-dimensional intrinsic manifolds or class structure. If the metric structure of the original pretrained latent space is perfectly preserved, then downstream tasks on the original distribution will continue to behave exactly as expected. More generally, for any downstream task involving, say, linear probes on the latent space, more or less distortion, which is to say, more or less rigidity for the choice of V, may be tolerated.

4.2 Results

Table 1 shows classification accuracy on both original and augmented data for the pretrained network $F =$ DINO and for one-hidden-layer MLP encoders E_θ trained with the four losses enumerated previously. The dimensions of all layers in E_θ are set to the same dimensionality as the latent space. The first number of each entry reports classification accuracy for non-augmented data, and the second number is the accuracy on augmented data. For example, the entry for row (LC, MaWa) and column Rotation, namely, the pair 98.41 | 90.41, should be read as follows: the composite network $C \circ E_\theta \circ F$ (for a linear classifier C trained on non-augmented data and for E_θ trained for rotation-invariance with MaWa minimization) achieves 98.41% classification accuracy on non-rotated STL10 DINO features (the first number). Further, on arbitrarily rotated images processed through F, $C \circ E_\theta \circ F$ still achieves 90.41% accuracy (the second number). This

Table 1: Classification comparison across models for linear (LC), nonlinear (NC), and end-to-end (EC) classifiers. First number is accuracy on original STL10 data processed through DINOv2 pretrained feature network. Second number is accuracy on augmented data.

Augmentation		Rotation	Affine	Noise	Crop
LC	MaWa	98.41 | 90.41	98.34 | 95.79	98.32 | 79.81	98.21 | 97.76
	WaCo	97.67 | 90.78	98.12 | 96.14	97.44 | 78.66	98.20 | 97.69
	SCLR	88.61 | 82.70	97.30 | 95.32	93.49 | 43.46	98.09 | 97.69
	HSIC	77.22 | 73.34	89.90 | 87.69	71.10 | 62.38	93.86 | 93.55
NC	MaWa	98.66 | 91.07	98.75 | 96.49	98.74 | 80.43	98.76 | **98.33**
	WaCo	98.06 | **91.88**	98.25 | **96.69**	97.78 | **80.44**	98.39 | 98.12
	SCLR	89.99 | 83.93	97.60 | 95.36	94.10 | 47.39	97.91 | 97.65
	HSIC	86.41 | 80.41	95.05 | 92.40	80.30 | 67.67	96.91 | 96.37
EC		98.66 | 71.13	98.66 | 95.12	98.66 | 61.82	98.66 | 98.32

is in contrast to the EC, or end-to-end classifier, row, which shows that the network $C \circ F$ (for C nonlinear with total depth and dimension equal to $C \circ E_\theta$) drops to 71.13% accuracy for arbitrarily rotated images.

Across augmentation types, the MaWa minimization loss and the WaCo maximization loss perform strongly and comparably, but the SCLR and HSIC losses are not competitive on rotations and noise augmentations. The success of the EC case for affine and crop augmentations, as well as the stronger results for SimCLR and HSIC on these augmentation types, has to do with the fact that the DINO model is already strongly (approximately) invariant to these augmentations. In particular, crops are central to how the model was trained, so the results for these cases are not unexpected. Given this, we will focus, going forward, on the results for rotation and noise augmentations.

Importantly, in all cases, the MaWa minimization loss and the WaCo maximization loss consistently preserve much of the structure of the original latent space, whereas the SimCLR and HSIC losses introduce major corruptions. Tables 2 and 3 show that MaWa minimization better preserves the pretrained latent space than WaCo maximization. For the case of rotations, Table 2 shows that E_θ trained with the MaWa loss is very close to an isometry on the latent distribution with $R^2 = 0.95$. For the WaCo loss, there is still a clear linear trend close to the isometry line ($y = x$), but there is more variance, with $R^2 = 0.57$ for the case where E_θ maintains the dimensionality of the latent space. As seen in the third column, when reducing the dimension of the latent space from $d = 384$ to $d = 96$, the WaCo loss still performs comparably to the case with no dimensionality reduction.

Table 2: Structure preservation for MaWa and WaCo rotation-invariant encoders on DINO features.

Structure		STL10 + DINO + Rotation		
Preservation		MaWa ($d = 384$)	WaCo ($d = 384$)	WaCo ($d = 96$)
Similarity	$L_1 \mid L_2$	0.69 | 1.03	0.48 | 1.34	0.52 | 1.17
Tests	$m \mid b$	0.95 | 0.01	0.94 | 0.01	0.90 | 0.01
	R^2	0.95	0.57	0.67
Classification	LC	98.41 | 90.41	97.67 | 90.78	97.04 | 89.18
	NC	98.66 | 91.07	98.06 | 91.88	97.54 | 90.73
	EC	98.66 | 71.13	98.66 | 71.13	98.66 | 71.13

The results for noise augmentations reported in Table 3 are similar. The drop in accuracy is now more pronounced, since noise is highly destructive, but the optimal transport-based losses still gain a nearly 20 percent increase in accuracy over the EC case. We also note that for the WaCo loss with $d = 96$, accuracy can be improved to 80.89% in the LC case and 81.68% in the NC case simply by increasing the depth of the network to two hidden layers, indicating that further optimizations are certainly possible.

Table 3: Structure preservation for MaWa and WaCo noise-invariant encoders on DINO features.

Structure		STL10 + DINO + Noise		
Preservation		MaWa ($d = 384$)	WaCo ($d = 384$)	WaCo ($d = 96$)
Similarity	L_1 | L_2	0.70 | 1.01	0.46 | 1.40	0.52 | 1.20
Tests	m | b	0.97 | 0.01	0.95 | 0.01	0.92 | 0.01
	R^2	0.97	0.53	0.65
Classification	LC	98.32 | 79.81	97.44 | 78.66	97.04 | 75.98
	NC	98.74 | 80.43	97.78 | 80.44	97.64 | 78.47
	EC	98.66 | 61.82	98.66 | 61.82	98.66 | 61.82

We also visualize these structure-preservation results directly by plotting the best linear fit for the scatter plot of input and encoded distance pairs. Figures 1 and 2 are for the MaWa and WaCo losses, respectively, in the case of rotation invariance, and we see clearly the additional variance introduced by the WaCo loss. The plots for noise invariance look nearly identical.

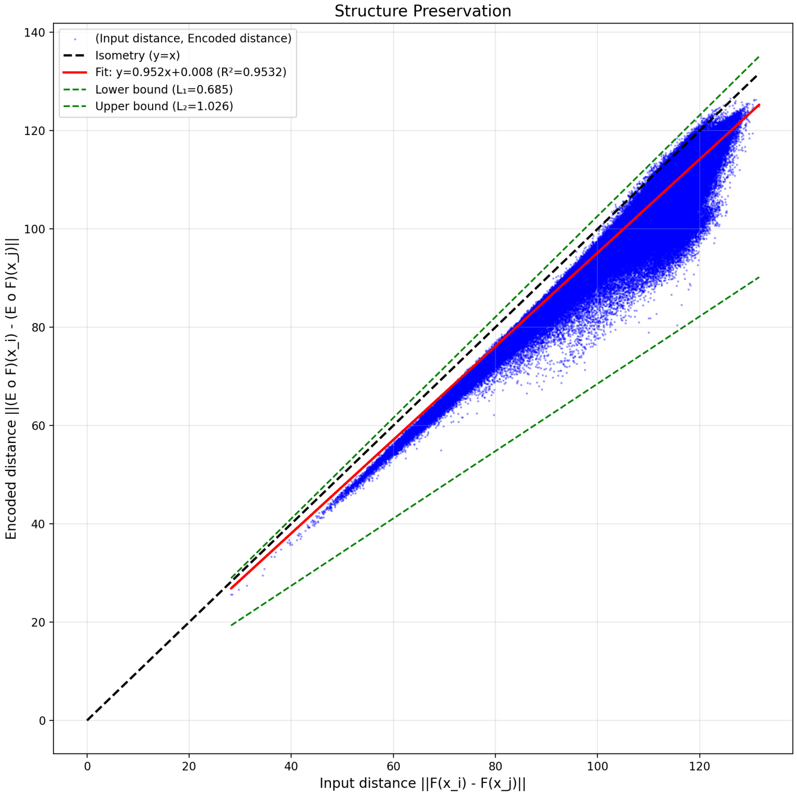


Figure 1: Structure preservation for MaWa rotation-invariant encoder on DINO features.

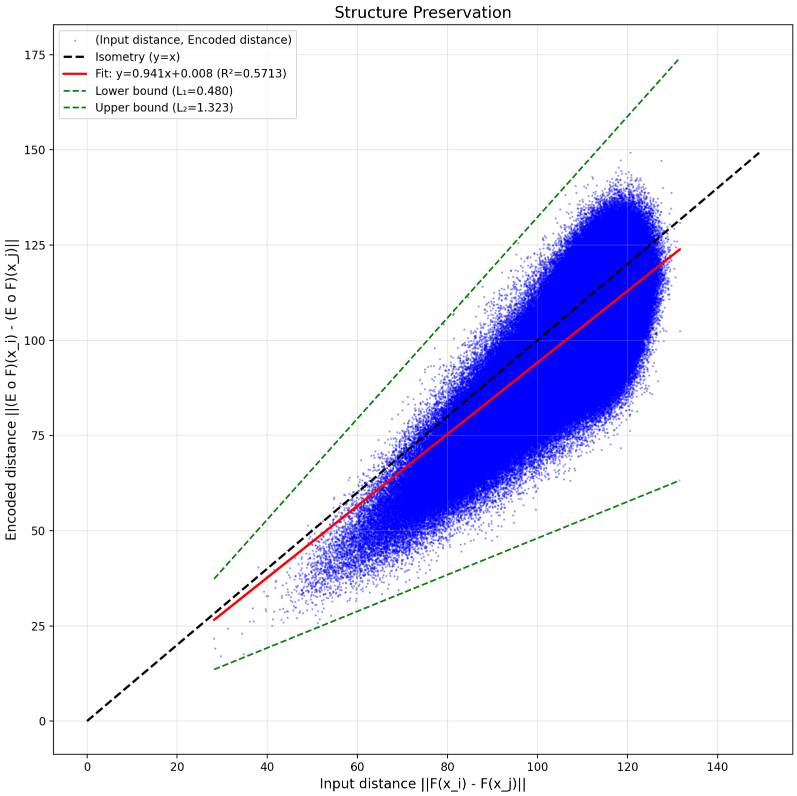


Figure 2: Structure preservation for MaWa rotation-invariant encoder on DINO features.

In contrast to the results for the MaWa and WaCo losses, the structural results for SimCLR and HSIC on rotation and noise augmentations in Tables 4 and 5 show that the predictive power (R^2) of the linear fit, which tells us how close we are to an isometry, plummets. Figures 3 and 4 illustrate the contrast more directly, and we see that the HSIC loss, in particular, leads to strong degeneration of the original latent space. As before, the corresponding plots for noise invariance are nearly identical.

Table 4: Structure preservation for SimCLR and HSIC rotation-invariant encoders on DINO features.

Structure		STL10 + DINO + Rotation	
Preservation		SCLR ($d = 384$)	HSIC ($d = 384$)
Similarity	$L_1 \mid L_2$	0.13 | 2.82	0.01 | 0.06
Tests	$m \mid b$	0.85 | 0.01	0.03 | 0.00
	R^2	0.06	0.23
Classification	LC	88.61 | 82.70	77.22 | 73.34
	NC	89.99 | 83.93	86.41 | 80.41
	EC	98.66 | 71.13	98.66 | 71.13

We provide one final comparison of the different methods for the case of $F = \text{DINO}$. Figure 5 shows the t-sne visualizations of the original STL10 DINO latent distribution $F_\sharp \mu_X$ compared to $(E_\theta \circ F)_\sharp \mu_X$ for encoders trained for noise invariance using one of the four losses. Both optimal transport-based losses strongly preserve the class structure (in virtue of approximately preserving metric structure), whereas SimCLR and HSIC produce noticeable corruption. The t-sne plots for rotation invariance show exactly similar patterns.

Table 5: Structure preservation for SimCLR and HSIC noise-invariant encoders on DINO features.

Structure		STL10 + DINO + Noise	
Preservation		SCLR ($d = 384$)	HSIC ($d = 384$)
Similarity	$L_1 \mid L_2$	0.37 | 3.01	0.01 | 0.06
Tests	$m \mid b$	1.29 | 0.01	0.03 | 0.00
	R^2	0.12	0.18
Classification	LC	93.49 | 43.46	71.10 | 62.38
	NC	94.10 | 47.39	80.30 | 67.67
	EC	98.66 | 61.82	98.66 | 61.82

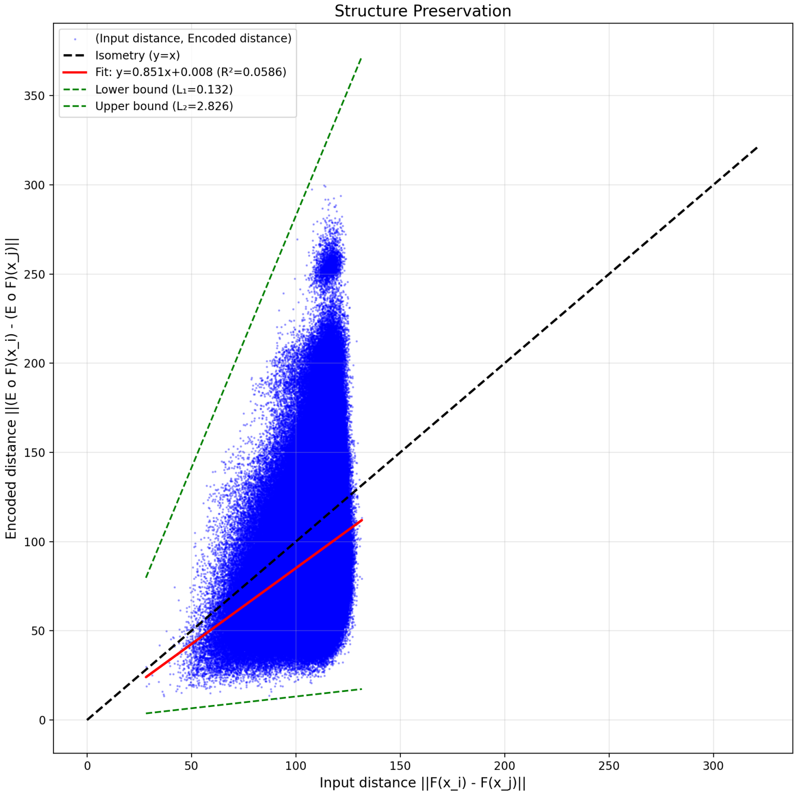


Figure 3: Structure preservation for SimCLR rotation-invariant encoder on DINO features.

We return now to the case of the MaWa and WaCo losses to illustrate how the choice of F changes things. For $F =$ SwAV, Tables 6 and 7 show that invariant classification for rotation and noise augmentations is not as strong as for the more modern DINO network, but the relative increase in accuracy over the EC case is actually greater. Looking at the case of noise, for example, the EC classifier is essentially unusable at 39% accuracy, whereas the MaWa encoder, for example, achieves 74% accuracy in both the linear and nonlinear cases. Interestingly, the WaCo loss with dimensionality reduction performs better than the one without.

We discuss one last test to better illuminate the properties of the WaCo loss, which, as noted, can be defined with or without a pretrained network F and can be used simultaneously for dimensionality reduction, or for dimensionality reduction alone. We train an encoder E_θ on MNIST using the ordinary WaCo loss (44) with, in the first case, $T = \{\mathrm{id}_X\}$ or with $T = \{\mathrm{id}_X, t\}$ in the second, where t is arbitrary rotations. Figure

Table 6: Structure preservation for MaWa and WaCo rotation-invariant encoders on SwAV features.

Structure		STL10 + SwAV + Rotation		
Preservation		MaWa ($d = 2048$)	WaCo ($d = 2048$)	WaCo ($d = 96$)
Similarity	$L_0 \mid L_1$	0.61 | 0.97	0.44 | 1.84	0.47 | 1.25
Tests	$m \mid b$	0.85 | 0.07	1.09 | 0.09	0.92 | 0.08
	R^2	0.94	0.70	0.80
Classification	LC	93.91 | 83.79	87.31 | 79.95	89.72 | 82.18
	NC	93.55 | 82.93	87.91 | 80.54	90.84 | 82.44
	EC	94.30 | 51.70	94.30 | 51.70	94.30 | 51.70

Table 7: Structure preservation for MaWa and WaCo noise-invariant encoders on SwAV features.

Structure		STL10 + SwAV + Noise		
Preservation		MaWa ($d = 2048$)	WaCo ($d = 2048$)	WaCo ($d = 96$)
Similarity	$L_0 \mid L_1$	0.63 | 0.96	0.35 | 2.25	0.41 | 1.29
Tests	$m \mid b$	0.87 | 0.07	1.12 | 0.10	0.89 | 0.08
	R^2	0.96	0.67	0.79
Classification	LC	94.10 | 74.29	85.81 | 67.95	89.75 | 71.30
	NC	93.88 | 74.21	86.49 | 67.55	90.97 | 71.23
	EC	94.30 | 39.28	94.30 | 39.28	94.30 | 39.28

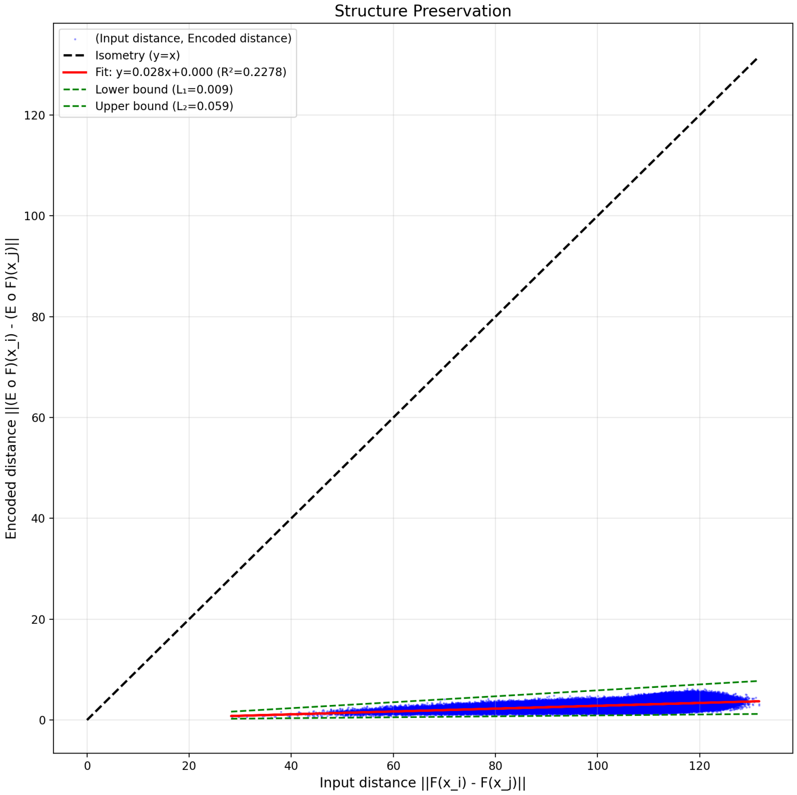


Figure 4: Structure preservation for HSIC rotation-invariant encoder on DINO features.

6 shows t-sne visualizations of the learned latent spaces for ordinary MNIST digits plus 90-degree rotations. In the first case, the 90-degree rotated digits are encoded as separate classes, whereas in the second case, we see near perfect overlap between the original classes and their rotated counterparts. (We emphasize again that although we visualize the case of 90-degree rotations, we trained on arbitrary rotations.) Finally, we note that the notion of invariance we have developed in this work depends on the choice of the initial distribution μ_X. If μ_X in this final test were instead 90-degree rotations of MNIST digits, or, say, the ordinary MNIST distribution plus an additional character, then that would be the structure preserved by the optimal transport-based losses, since augmentation invariance in our sense only applies to unique augmentations, as defined previously.

5 Conclusion

We have developed precise definitions of augmented encoders and augmentation invariance with emphasis on the problem of post-training augmentation invariance, in which our goal is to add additional invariance properties to a pretrained network without corrupting its behavior on the original, non-augmented distribution. Our experimental results show that both Markov-Wasserstein minimization and Wasserstein correlation maximization are effective losses for this objective, whereas a SimCLR loss or a HSIC maximization loss, by contrast, were both shown to be ineffective. Additionally, we have shown that the proposed methods are lightweight and do not depend on complicated architectures or fine hyperparameter tuning. With the right training objective, post-training augmentation invariance can be achieved, at least approximately, with only a one-hidden-layer MLP appended to the latent space of a frozen, pretrained network.

5.1 Limitations

While we have shown that our methods are effective for single augmentations, we note that invariance quality degrades and sample requirements increase dramatically when training for invariance to composite augmentations with large parameter spaces. For example, our random affine transformations both rotate and translate data to some degree, but if we apply completely arbitrary rotations and translations to, say, MNIST digits and train an encoder $E_\theta : \mathbb{R}^{784} \to \mathbb{R}^{64}$ on the WaCo loss, then $C \circ E_\theta$ for C linear only achieves around 50 percent accuracy on augmented data after 100 epochs, though this is still a strong improvement over the 13 percent accuracy found in the end-to-end case. Accuracy improves monotonically as the number of epochs increases, reaching, for example, about 68 percent accuracy after 1000 epochs (with the end-to-end classifier still at 13), but convergence is slow.

Improving sample efficiency and convergence for increasingly large augmentation parameter spaces may require going beyond simple MLP architectures in future work. Likewise, though we successfully demonstrated approximate post-training augmentation invariance for pretrained models with μ_X = STL10, architectural or training innovations may be needed when dealing with much larger, more complex input distributions.

5.2 Future Work

There are many promising directions for future work in this area. First, we pose the question: Are there other losses that can be used for post-training augmentation invariance? We have shown experimentally that the MaWa and WaCo losses are suitable candidates, though further theoretical investigation of their properties, especially the WaCo loss, is still needed. Finally, we highlight the experiments on noise invariance, since these results suggest that these methods may be promising for adversarial robustness, more generally. By treating adversarial attacks as augmentations, an encoder trained on either the MaWa loss or the WaCo loss could potentially produce robust latent spaces without any need for adversarial training.

References

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures.* Springer Science & Business Media, 2008.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

Xingjian Bai, Guangyi He, Yifan Jiang, and Jan Obloj. Wasserstein distributional robustness of neural networks. *Advances in Neural Information Processing Systems*, 36:26322–26347, 2023.

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.

Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Lower bounds on adversarial robustness from optimal transport. *Advances in Neural Information Processing Systems*, 32, 2019.

Vladimir Igorevich Bogachev and Maria Aparecida Soares Ruas. *Measure theory*, volume 1. Springer, 2007.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45, 2015.

Nicolas Bonnotte. *Unidimensional and evolution methods for optimal transportation.* PhD thesis, Université Paris Sud-Paris XI; Scuola normale superiore (Pise, Italie), 2013.

Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, Carl-Johann Simon-Gabriel, and Bernhard Schoelkopf. From optimal transport to generative modeling: the vegan cookbook. *arXiv preprint arXiv:1705.07642*, 2017.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

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.

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.

Kenta Cho and Bart Jacobs. Disintegration and bayesian inversion via string diagrams. *Mathematical Structures in Computer Science*, 29(7):938–971, 2019.

Florence Clerc, Vincent Danos, Fredrik Dahlqvist, and Ilias Garnier. Pointless learning. In *Foundations of Software Science and Computation Structures: 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings 20*, pp. 355–369. Springer, 2017.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in neural information processing systems*, 30, 2017.

Jared Culbertson and Kirk Sturtz. A categorical foundation for bayesian probability. *Applied Categorical Structures*, 22:647–662, 2014.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yonatan Dukler, Alessandro Achille, Hao Yang, Varsha Vivek, Luca Zancato, Benjamin Bowman, Avinash Ravichandran, Charless Fowlkes, Ashwin Swaminathan, and Stefano Soatto. Your representations are in the network: composable and parallel adaptation for large scale models. *Advances in Neural Information Processing Systems*, 36:28832–28860, 2023.

Keenan Eikenberry. *Bayesian Inference for Markov Kernels Valued in Wasserstein Spaces*. PhD thesis, Arizona State University, 2023.

Tobias Fritz. A synthetic approach to markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, 2020.

Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19338–19347, 2023.

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

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, 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.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International conference on machine learning*, pp. 1558–1567. PMLR, 2017.

Yuyang Ji, Zeyi Huang, Haohan Wang, and Yong Jae Lee. Customizing domain adapters for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 934–944, 2025.

Olav Kallenberg. *Foundations of Modern Probability.* Springer Nature, Switzerland AG, 3rd edition, 2021.

Daehee Kim, Youngjun Yoo, Seunghyun Park, Jinkyu Kim, and Jaekoo Lee. Selfreg: Self-supervised contrastive regularization for domain generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9619–9628, 2021.

Soheil Kolouri, Phillip E Pope, Charles E Martin, and Gustavo K Rohde. Sliced wasserstein auto-encoders. In *ICLR (Poster)*, 2019.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

L. Kunkel and M. Trabs. A wasserstein perspective of vanilla gans. *arXiv preprint arXiv:2403.15312*, 2024.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10285–10295, 2019.

Alexander Levine and Soheil Feizi. Wasserstein smoothing: Certified robustness against wasserstein adversarial attacks. In *International conference on artificial intelligence and statistics*, pp. 3938–3947. PMLR, 2020.

T. Li, J. Yu, and C. Meng. Scalable model-free feature screening via sliced-wasserstein dependency. *Journal of Computational and Graphical Statistics*, 32(4):1501–1511, 2023.

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. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Lang Liu, Soumik Pal, and Zaid Harchaoui. Entropy regularized optimal transport independence criterion. In *International Conference on Artificial Intelligence and Statistics*, pp. 11247–11279. PMLR, 2022.

Antoine Liutkus, Umut Simsekli, Szymon Majewski, Alain Durmus, and Fabian-Robert Stöter. Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *International Conference on machine learning*, pp. 4104–4113. PMLR, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pp. 875–884. PMLR, 2020.

Gilles Mordant and Johan Segers. Measuring dependence between random vectors via optimal transport. *Journal of Multivariate Analysis*, 189:104912, 2022.

Tamás F Móri and Gábor J Székely. The earth mover's correlation. *arXiv preprint arXiv:2009.04313*, 2020.

Kimia Nadjahi, Alain Durmus, Umut Simsekli, and Roland Badeau. Asymptotic guarantees for learning generative models with the sliced-wasserstein distance. *Advances in Neural Information Processing Systems*, 32, 2019.

Khai Nguyen, Tongzheng Ren, Huy Nguyen, Litu Rout, Tan Nguyen, and Nhat Ho. Hierarchical sliced wasserstein distance. *arXiv preprint arXiv:2209.13570*, 2022.

Van-Anh Nguyen, Trung Le, Anh Bui, Thanh-Toan Do, and Dinh Phung. Optimal transport model distributional robustness. *Advances in Neural Information Processing Systems*, 36:24074–24087, 2023.

Thomas Giacomo Nies, Thomas Staudt, and Axel Munk. Transport dependency: Optimal transport based dependency measures. *arXiv preprint arXiv:2105.02073*, 2021.

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

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Sherjil Ozair, Corey Lynch, Yoshua Bengio, Aaron Van den Oord, Sergey Levine, and Pierre Sermanet. Wasserstein dependency measure for representation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Evan Patterson. Hausdorff and wasserstein metrics on graphs and other structured data. *Information and Inference: A Journal of the IMA*, 10(4):1209–1249, 2021.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180. PMLR, 2019.

Svetlozar T Rachev and Ludger Rüschendorf. *Mass Transportation Problems: Volume 1: Theory*. Springer Science & Business Media, 2006.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017.

Ali Lotfi Rezaabad and Sriram Vishwanath. Learning representations by maximizing mutual information in variational autoencoders. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 2729–2734. IEEE, 2020.

Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3674–3683, 2020.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

Cédric Villani. *Topics in Optimal Transportation.* Number 58. American Mathematical Soc., 2003.

J. C. Wiesel. Measuring association with wasserstein distances. *Bernoulli*, 28(4):2816–2832, 2022.

Eric Wong, Frank Schmidt, and Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *International conference on machine learning*, pp. 6808–6817. PMLR, 2019.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7959–7971, 2022.

Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3713–3722, 2019.

Kaiwen Wu, Allen Wang, and Yaoliang Yu. Stronger and faster wasserstein adversarial attacks. In *International conference on machine learning*, pp. 10377–10387. PMLR, 2020.

Yijun Xiao and William Yang Wang. Disentangled representation learning with wasserstein total correlation. *arXiv preprint arXiv:1912.12818*, 2019.

Yi Xin, Siqi Luo, Haodi Zhou, Junlong Du, Xiaohong Liu, Yue Fan, Qing Li, and Yuntao Du. Parameter-efficient fine-tuning for pre-trained vision models: A survey. *arXiv e-prints*, pp. arXiv–2402, 2024.

Mengde Xu, Zheng Zhang, Fangyun Wei, Han Hu, and Xiang Bai. Side adapter network for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2945–2954, 2023.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pp. 5885–5892, 2019.

A Experiment Details

We use simple MLPs with ReLU activations for all tests. For our primary tests, the dimensions of the networks are of the form $(\text{input}, \text{hidden}, \text{output}) = (d, d, d)$, where d is the dimension of the pretrained latent space, namely $d = 384$ for DINO or $d = 2048$ for SwAV. For dimensionality reduction with WaCo, the networks have dimensions $(384, 192, 96)$ for DINO and $(2048, 512, 96)$ for SwAV. Our final MNIST test uses a two-hidden-layer network with dimensions $(784, 512, 256, 64)$ to achieve slightly better class separation in the latent space, but the results are essentially the same when restricting to a one-hidden-layer network with dimensions $(784, 256, 64)$.

In all of our tests, at least for the MaWa and WaCo losses, we found comparable results across a wide range of choices for the batch size, number of epochs, learning rate, optimizer, scheduler, and number of augmentation samples. Default settings are listed in Table 8. The one exception is that for SimCLR, we set the batch size to 1024 to match the total number of samples used in the other three losses, namely, 3×256 augmented samples plus an additional 256 non-augmented samples for a total of 1024.

All models were trained on either a single NVIDIA RTX 3070 GPU, a NVIDIA RTX 3090 GPU, or a NVIDIA A100 GPU. Compute times are fairly modest (on the order of tens of minutes), but when training for invariance to complicated, composite augmentations, training can extend to multiple hours when using

Table 8: Default hyperparameters

Parameter	**Value**
Batch size	256
Epochs	100
Learning rate	1×10^{-3}
Optimizer	AdamW (Loshchilov & Hutter, 2017) with weight decay 1×10^{-4}
Scheduler	CosineAnnealing with minimum rate 4×10^{-4}
Augmentation Samples s	3

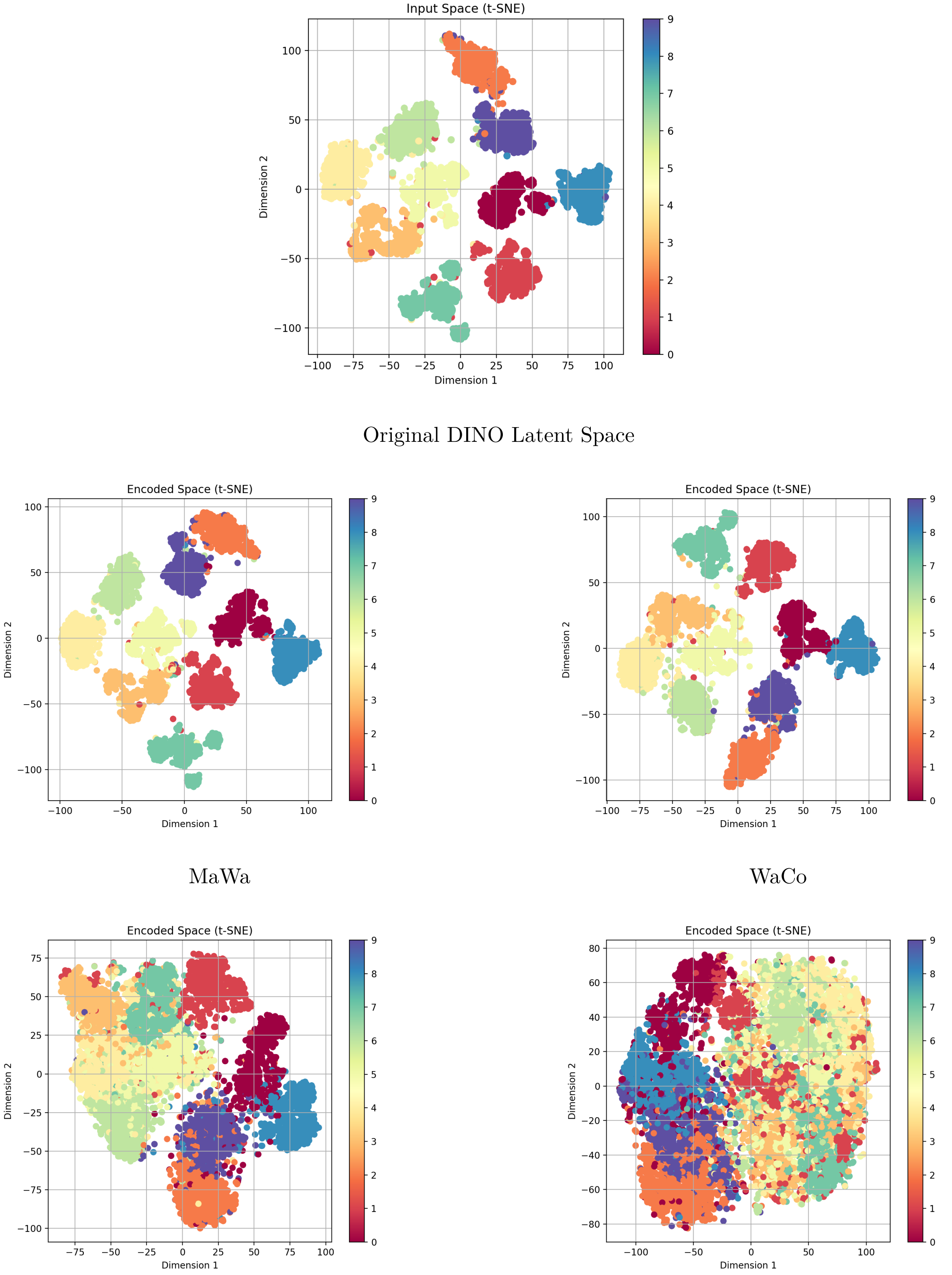
only a single GPU. Applying multiple augmentations and extracting pretrained features are two of the main bottlenecks.

For our invariance tests, all classifiers are trained on a standard cross-entropy loss, and the optimizer and scheduler settings are the same as in Table 8 but without weight decay for the optimizer (meaning that we are effectively using an ordinary Adam optimizer rather than AdamW for classification tests).

We use the Kornia package (Riba et al., 2020) for augmentations. We vary the augmentation parameters for each element of the batch (same_on_batch = False), but, in fact, results are comparable when the same augmentation is applied across the batch, which leads to better computational efficiency, when needed. Default augmentation parameters are listed in Table 9. We take, for example, 50 to 70 percent crops with aspect ratio $(0.75, 1.33)$ and then resize to the original dimensions of the data, namely $(96, 96)$ for STL10.

Table 9: Default augmentation parameters

Augmentation	**Parameters**	
RandomRotation	degrees $= (-180, 180)$	
RandomAffine	degrees $= (-30, 30)$,	translate $= (0.2, 0.2)$,
	scale $= (0.8, 1.2)$,	shear $= (-15, 15)$
RandomGaussianNoise	mean $= 0$,	std $= 2$ or 1
RandomResizedCrop	scale $= (0.5, 0.7)$,	resize to $= (0.75, 1.33)$
	output size $= (96, 96)$	



Original DINO Latent Space

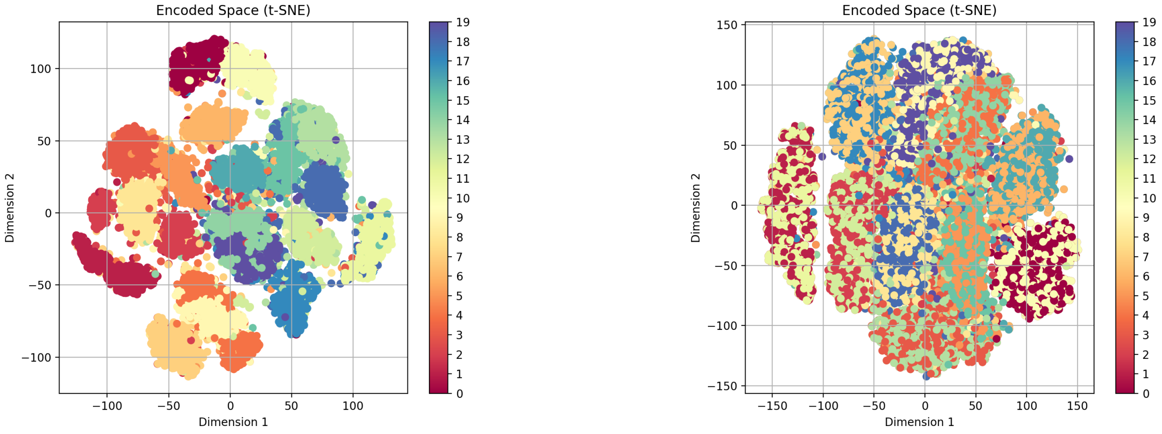
MaWa

WaCo

SCLR

HSIC

Figure 5: t-sne visualizations of the original STL10 DINO latent distribution $F_\sharp\mu_X$ compared to $(E_\theta \circ F)_\sharp\mu_X$ for encoders trained for noise invariance with one of the four losses.



Dim

Dim + Inv

Figure 6: t-sne visualizations of MNIST plus 90-degree rotated digits for E_θ with final dimension $d = 64$ trained on the WaCo loss with $T = \{\mathrm{id}_X\}$, which does dimensionality reduction (Dim) only, versus the case of $T = \{\mathrm{id}_X, t\}$, which does Dim plus invariance (Inv) to t, where t here is arbitrary rotations.