How Learning by Reconstruction Produces Uninformative Features for Perception

Randall Balestriero [1] **Yann LeCun** [2]

Abstract

Input space reconstruction is an attractive representation learning paradigm. Despite interpretability benefit of reconstruction and generation, we identify a misalignment between learning to reconstruct, and learning for perception. We show that the former allocates a model's capacity towards a subspace of the data explaining the observed variance–a subspace with uninformative features for the latter. For example, the supervised TinyImagenet task with images projected onto the top subspace explaining 90% of the pixel variance can be solved with 45% test accuracy. Using the bottom subspace instead, accounting for only 20% of the pixel variance, reaches 55% test accuracy. Learning by reconstruction is also wasteful as the features for perception are learned last, pushing the need for long training schedules. We finally prove that learning by denoising can alleviate that misalignment for some noise strategies, e.g., masking. While tuning the noise strategy without knowledge of the perception task seems challenging, we provide a solution to detect if a noise strategy is never beneficial regardless of the perception task, e.g., additive Gaussian noise.

1. Introduction

One of the far reaching mandate of deep learning is to provide a self-contained methodology to learn intelligible and universal representations of data (LeCun et al., 2015). That is, to learn a nonlinear transformation of the data producing a parsimonious representation able to solve numerous downstream tasks. Significant progress has been made through the lens of supervised learning, i.e., by learning a representation that maps the observed data to provided labels of an

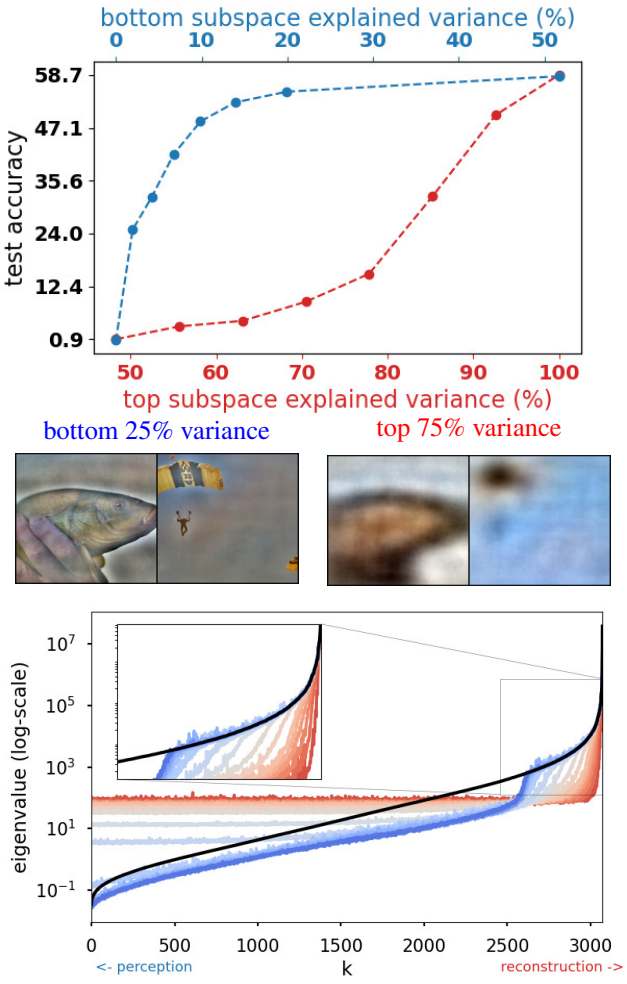

Figure 1. **Features for reconstruction are uninformative for perception (top):** TinyImagenet ResNet9 top-1 accuracy when trained and validated on images projected on the top-subspace (red) or bottom subspace (blue) of explained variance, corresponding images displayed in the **middle** and in Fig. 10. **Perception features are learned last (bottom):** training loss evolution (red to blue) of reconstructed training images from a deep Autoencoder projected onto the eigenspace of the original data (black). The top eigenspace (right) is learned first, and then, if training lasts long enough, the features most useful for perception (left) are finally learned. This explains why learning by performances on perception task keep increasing long after reconstructed samples look appealing.

[1]Brown University [2]NYU. Correspondence to: Randall Balestriero <rbalestr@brown.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

priori known downstream task (Krizhevsky et al., 2012). Labels being costly and over-specialized, much progress was also made through the lens of unsupervised learning (Barlow, 1989; Ghahramani, 2003) roughly falling into three camps. First, reconstruction-based methods that produce (compressed) latent representations of the data nonetheless sufficient to recover most of the original data, e.g., Denoising/Variational/Masked Auto-Encoders (Vincent et al., 2010; Kingma & Welling, 2013; He et al., 2022) and Autoregressive models (Van den Oord et al., 2016; Chen et al., 2018). Second, score matching which is often solved by setting up a surrogate supervised task of classifying observed samples from noise (Hyvärinen & Dayan, 2005). Third, Self-Supervised Learning (SSL) (Chen et al., 2020; Zbontar et al., 2021; Bardes et al., 2021; Balestriero et al., 2023)–whose contrastive methods can be thought of as generalized versions of score matching and Noise Contrastive Estimation (Gutmann & Hyvärinen, 2010)–combining an invariance term bringing together representations of data known to be semantically similar, or generated to be so, and an anti-collapse term making sure that not all representations become similar.

In recent years, SSL emerged as the preferred solution–regularly reaching new state-of-the-arts through careful experimental design (Garrido et al., 2023). Yet, reconstruction-based methods maintain a large presence due to their ability to provide reconstructed samples that are human interpretable, enabling informed quality assessment of a model (Selvaraju et al., 2016; Bordes et al., 2021; Brynjolfsson et al., 2023; Baidoo-Anu & Ansah, 2023). Despite that benefit, reconstruction-based learning falls behind SSL as it requires fine-tuning to reach the state-of-the-art. One of the most popular reconstruction-based learning strategy that emerged as the solution of choice in recent years is the Masked-Autoencoder (He et al., 2022).

We ask the following question: *Why reconstruction based learning easily produces compelling reconstructed samples but fails to provide competitive latent representations for perception?* We can pinpoint that observation to at least three reasons.

R1: Misaligned. The features with most reconstructive power are the least informative for perceptual tasks as depicted in Fig. 1 (top). Instead, images projected onto the subspace least useful for reconstruction still contain sufficient information for perception.

R2: Ill-conditioned. From the misalignment, the features useful for perception are learned last for a reconstruction task as depicted in Fig. 1 (bottom), meaning that learning by reconstruction is wasteful as it requires long training schedules to capture informative features for perception.

R3: Ill-posed. Due to the "irrelevance" of perception features to affect reconstruction error, learning by reconstruction can produce drastically different representation quality between training settings. For example, we can find a same model with two different set of parameters that produce the same train and test reconstruction error while exhibiting significant performance gaps for perceptual tasks as depicted in Fig. 7, where for a given reconstruction error the top-1 accuracy on Imagenet-10 can vary from 50% to almost 90%.

These findings provide first clues as to why learning by reconstruction requires long training time and fine-tuning–observations that are common knowledge within the representation learning community but still lack theoretical justification. Those findings alone do not answer the following question: *Why Masked Autoencoders were able to provide a significant improvement in the quality of the learned representation for solving perception tasks?* We will prove that some denoising tasks can push back some of the limitations of learning by reconstructions. In particular, we will demonstrate that masking is provably beneficial while other noise distributions, e.g., additive Gaussian noise, are not. We summarize our technical contributions below

- We provide a novel measure of alignment between reconstruction and classification task from first principles (Section 3.1, Eq. (7), and Fig. 2)
- We identify the root cause for reconstruction and perceptual classification tasks to be misaligned in term of the spectral property of the data covariance matrix (Section 3.2 and Fig. 4). The misalignment increases with presence of background and increased number of objects in the images
- We prove that denoising tasks introduce a mechanism through the noise distribution to improve alignment between reconstruction and perception: masking provides great alignment benefits while additive Gaussian noise is provably ineffective(Fig. 6 and Corollary 3.1)

We hope that our findings will help skew further research in learning by reconstruction to explore alternative noise distributions as they are the main driver of learning useful representations for perception. Codebase provided at `github.com/RandallBalestriero/LearningByReconstruction`.

2. Background and Notations

Notations: We denote by $\boldsymbol{x}_n \in \mathbb{R}^D, n = 1, \dots, N$ the n^{th} input sample, e.g., a (H, W, C) image flattened to a $D = H \times W \times C$ dimensional vector. The entire training set is collected into the matrix $\boldsymbol{X} \triangleq [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N] \in \mathcal{M}_{D,N}(\mathbb{R})$ where $\mathcal{M}_{n,m}(\mathbb{R})$ is the vector space of real $n \times m$ matrices. Throughout our study, vectors will always

be column-vectors, and matrices are built by horizontally stacking column-vectors, i.e., they are column major. Unless specified otherwise, we assume that the input matrix $\boldsymbol{X}$ is full-rank. In practice, if this is not the case, one can easily disregard the subspace associated with 0 singular values and apply our analysis on that filtered matrix instead.

Learning by reconstruction. Learning representations by fitting a model's parameters θ to produce a reconstruction of presented inputs as in

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}} \left[d\left(g_\theta \left(f_\theta(\boldsymbol{x}) \right), \boldsymbol{x} \right) \right], \tag{1}$$

is common (Bottou, 2012; Kingma & Ba, 2014; LeCun et al., 2015), given some distance measure d. The reconstruction provides a qualitative signal enabling one to easily assess the quality of the model, and even interpret trained classification models with g_θ being trained and f_θ the (frozen) pretrained classifier (Zeiler & Fergus, 2014; Mahendran & Vedaldi, 2015; Olah et al., 2017; Shen et al., 2020). In its simplest form, the encoder $f_\theta : \mathbb{R}^D \mapsto \mathbb{R}^K$ and the decoder $g_\theta : \mathbb{R}^K \mapsto \mathbb{R}^D$ are linear, possibly with shared parameters. In such settings, the optimal parameters are obtained from Principal Component Analysis (Wold et al., 1987). Many variants of Eq. (1) have emerged, such as denoising and masked autoencoders (MAEs)(Vincent et al., 2010; He et al., 2022). The objective remains similar: learn a low-dimensional latent embedding of the data that is able to reconstruct the original samples while being robust to some noise perturbation added onto the samples as in

$$\min_{\theta} \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}} \left[\mathbb{E}_{\boldsymbol{x}'\sim p_{\boldsymbol{x}'|\boldsymbol{x}}} \left[d\left(g_\theta \left(f_\theta(\boldsymbol{x}') \right), \boldsymbol{x} \right) \right] \right], \tag{2}$$

with $p_{\boldsymbol{x}'|\boldsymbol{x}}$ applying some (conditional) noise transformation to the original input, e.g., $\boldsymbol{x}' \sim \mathcal{N}(\boldsymbol{x}, \epsilon \boldsymbol{I}), \epsilon > 0$.

Known limitations. Learning by reconstruction is widely popular and thus heavily studied. Major axes of research evolve around (i) deriving novel loss functions for specific datasets that better align with semantic distance (Wang et al., 2004; Kulis et al., 2013; Ballé et al., 2016; Bao et al., 2017), (ii) explaining the learned embedding dimensions (Tran et al., 2017; Esmaeili et al., 2019; Mathieu et al., 2019), and (iii) imposing structure in the embedding space such as clustered embeddings (Jiang et al., 2016; Dilokthanakul et al., 2016; Lim et al., 2020; Seydoux et al., 2020). Despite the rich literature, the current solutions of choice to learn representation in computer vision still rely on the Mean Squared Error loss in pixel space with the possible application of structured noise ($p_{\boldsymbol{x}'|\boldsymbol{x}}$ in Eq. (2)), e.g., as employed by the current state-of-the-art solution of masked-autoencoders (MAEs) (He et al., 2022). Yet, even that solution learns a representation that needs to be fine-tuned to compete with state-of-the-art, e.g., MAE's performances are drastically determined by two parameters (i) the training time, i.e., evaluation performance of the learned representation does not plateau even after 1600 epochs on Imagenet, and (ii) the need to fine tune, i.e., evaluation performance with and without finetuning have a significant gap going from 70% to 84% on top-1 Imagenet classification task.

Closer to our study, (Locatello et al., 2019) showed a first possible theoretical limitation of learning disentangled representations through reconstruction. That limitation is also reinforced by the large empirical study of (Zamir et al., 2018) concluding that learning by reconstruction and learning by denoising produce similar representations that are misaligned from supervised learning.

Our study will propose some first hints as to why even current state-of-the-art reconstruction-based solutions are poised with slow training, and the need to finetune (Section 3). We will conclude by proving that MAEs's masking strategy partially alleviates those limitations (Section 4), showing that the most rewarding findings for learning by reconstruction may emerge from novel denoising strategies, hence bringing some nuance into the alignment results from (Zamir et al., 2018).

3. Rich Features for Reconstruction are Poor Features For Perception

This section provides the theoretical ground and empirical validation of **R1** and **R2** from Section 1, namely, that learning by reconstruction learns features that are misaligned with common perception tasks. We start by deriving a closed form alignment measure between those two tasks in Section 3.1 and conclude by empirically measuring that mismatch in Section 3.2.

3.1. How To Measure The Alignment Between Reconstruction and Supervised Tasks

As a starting point to our study, we will build intuition and obtain theoretical results in the linear regime. As we will see at the end of this Section 3.1, this seemingly simplified setting turns out to be informative of practical cases.

Let's consider an encoder mapping $\boldsymbol{V} \in \mathcal{M}_{D,K}(\mathbb{R})$, a decoder mapping $\boldsymbol{Z} \in \mathcal{M}_{K,D}(\mathbb{R})$, and a predictor head $\boldsymbol{W} \in \mathcal{M}_{K,C}(\mathbb{R})$, where C is the number of target dimensions, or classes. The targets for $\boldsymbol{X} \in \mathcal{M}_{D,N}(\mathbb{R})$ are given by $\boldsymbol{Y} \in \mathcal{M}_{C,N}(\mathbb{R})$. The combination of the supervised and reconstruction losses is given by

$$\mathcal{L}(\boldsymbol{V}, \boldsymbol{W}, \boldsymbol{Z}) = \|\boldsymbol{W}^\top \boldsymbol{V}^\top \boldsymbol{X} - \boldsymbol{Y}\|_F^2 + \lambda \|\boldsymbol{Z}^\top \boldsymbol{V}^\top \boldsymbol{X} - \boldsymbol{X}\|_F^2, \tag{3}$$

where the latent representation, $\boldsymbol{V}^\top \boldsymbol{X}$, is shared between the two losses, $\lambda \geq 0$ controls the trade-off between the two terms. Quantifying how the optimal parameters of Eq. (3)

vary with λ will be key to assess how the two losses are aligned. As a starting point, let's formalize below the optimal parameters of this loss function using the notations $\boldsymbol{M} = \boldsymbol{P_M D_M P_M^\top}$ for the eigendecomposition of a symmetric semi-definite positive matrix $\boldsymbol{M}$. We will also denote, to lighten notations, $\boldsymbol{A} \triangleq \boldsymbol{X}\left(\boldsymbol{Y}^\top\boldsymbol{Y} + \lambda\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{X}^\top$.

Theorem 1. *The loss function from Eq. (3) is minimized for*

$$\boldsymbol{V}^* \text{ spans } \boldsymbol{P_{XX^\top}} \boldsymbol{D}_{\boldsymbol{XX}^\top}^{-\frac{1}{2}} (\boldsymbol{P_H})_{\cdot,1:K}, \tag{4}$$

$$\boldsymbol{W}^* = \left(\boldsymbol{V}^{*\top}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{V}^*\right)^{-1}\boldsymbol{V}^{*\top}\boldsymbol{X}\boldsymbol{Y}^\top, \tag{5}$$

$$\boldsymbol{Z}^* = \left(\boldsymbol{V}^{*\top}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{V}^*\right)^{-1}\boldsymbol{V}^{*\top}\boldsymbol{X}\boldsymbol{X}^\top, \tag{6}$$

where $\boldsymbol{H} \triangleq \boldsymbol{D}_{\boldsymbol{XX}^\top}^{-\frac{1}{2}} \boldsymbol{P}_{\boldsymbol{XX}^\top}^\top \boldsymbol{A} \boldsymbol{P_{XX^\top}} \boldsymbol{D}_{\boldsymbol{XX}^\top}^{-\frac{1}{2}}$. *(Proof in Section 5.1, empirical validation in Fig. 9.)*

We observe that the optimal solutions from Theorem 1 continuously interpolates between the standard least square (OLS) problem ($\lambda = 0$) and the unsupervised linear autoencoder or Principal Component Analysis (PCA) setting ($\lambda \to \infty$). We formalize below that we recover the optimal solutions for each of those extreme cases from Theorem 1.

Corollary 1.1. *The solution from Theorem 2 recovers the OLS solution for* $\boldsymbol{W}^{*\top}\boldsymbol{V}^{*\top}$ *as* $\lambda \to 0$*, and the PCA solution for* $\boldsymbol{Z}^{*\top}\boldsymbol{V}^{*\top}$ *as* $\lambda \to \infty$*. (Proof in Section 5.2.)*

That observation should comfort the reader that Eq. (3) accurately conveys both ends of the spectrum from supervised learning to reconstruction based learning, while continuously interpolating in-between.

Condition for perfect alignment. The result from Theorem 1 enables us to formalize the condition for perfect alignment between the two tasks, i.e., under which condition the solution $\boldsymbol{V}^*$ is not impacted by λ.

Proposition 1. *The supervised and reconstruction tasks are aligned (the optimal solutions do not depend on* λ*) iff the intersection of the top-*K *eigenspaces of* $\boldsymbol{X}^\top\boldsymbol{X}$ *and* $\boldsymbol{Y}^\top\boldsymbol{Y}$ *is of dimension* K*.*

In other words, whenever the condition in Proposition 1 holds, the matrix $\boldsymbol{P_H}$ (recall Theorem 1) will include the same eigenvectors (up to rotation) for any λ, making the optimal parameters (Eqs. (4) to (6)) independent of λ. In practice, we will see that Proposition 1's alignment condition is never fulfilled, pushing the need to define a more precise measure of alignment between the two tasks.

Continuous measure of alignment. As we aim to measure the tasks alignment more precisely that in a yes/no setting (Proposition 1), we propose the following continuous measure

$$\text{alignment}(k) \triangleq \frac{\|\boldsymbol{Y}^\top\boldsymbol{Y}(\boldsymbol{P_{XX^\top}})_{1:k}\|_F^2}{\|\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{P_{XX^\top}}\|_F^2}, \tag{7}$$

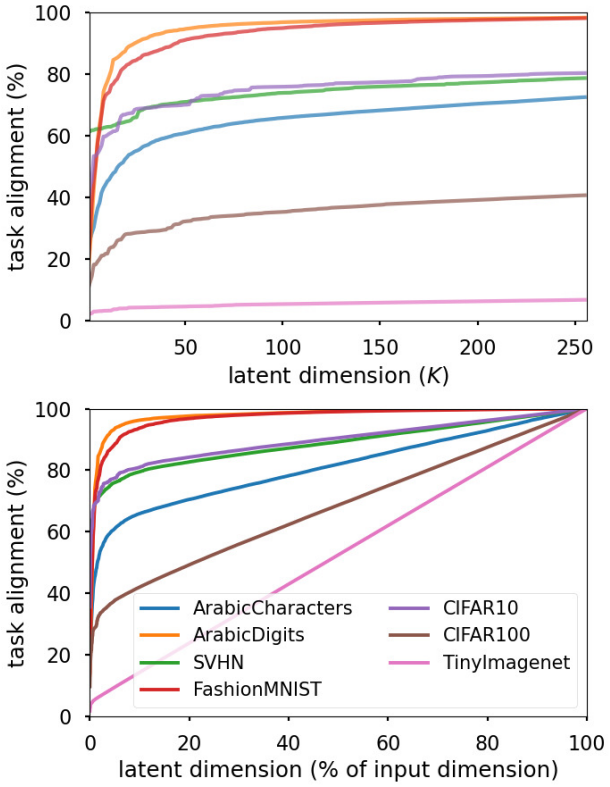

Figure 2. Depiction of the closed form alignment measure from Eq. (7) measuring the minimum supervised training error achievable given the optimal reconstruction parameters, as per Theorem 1 and Corollary 1.2. **Top:** depiction in terms of the latent dimension K. **Bottom:** depiction in terms of the ratio of the latent dimension K to the input dimension D. We clearly observe that as the dataset becomes more realistic (going from background-free images to CIFAR and TinyImagenet), as the alignment between the reconstruction and supervised task lessens. In particular, when going to TinyImagenet, we observe that the alignment only increases linearly with respect to the latent space dimension.

where $\|\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{P_{XX^\top}}\|_F^2$ simplifies as $\|\boldsymbol{Y}^\top\boldsymbol{Y}\|_F^2$ when $D = N$. We assume that the supervised task can be at least partially solved from $\boldsymbol{X}$, ensuring $\|\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{P_{XX^\top}}\|_F^2 > \epsilon$. In words, Eq. (7) is the (scaled and negated) supervised training error achieved given the representation $(\boldsymbol{V}^\top\boldsymbol{X})$ minimizing the reconstruction loss, which is measured by how much of the matrix $\boldsymbol{Y}^\top\boldsymbol{Y}$ can be reconstructed from the top-k subspace of $\boldsymbol{X}^\top\boldsymbol{X}$, as formalized below.

Corollary 1.2. *alignment*(k) *from Eq. (7) increases with* k*, has value* 0 *iff the two losses are misaligned, and has value* 1 *iff the two losses are aligned. (Proof in Section 5.3.)*

Although the measure from Eq. (7) is motivated from the linear setting of Theorem 1, we will demonstrate at the end of this Section 3.1 that it actually aligns well with practical

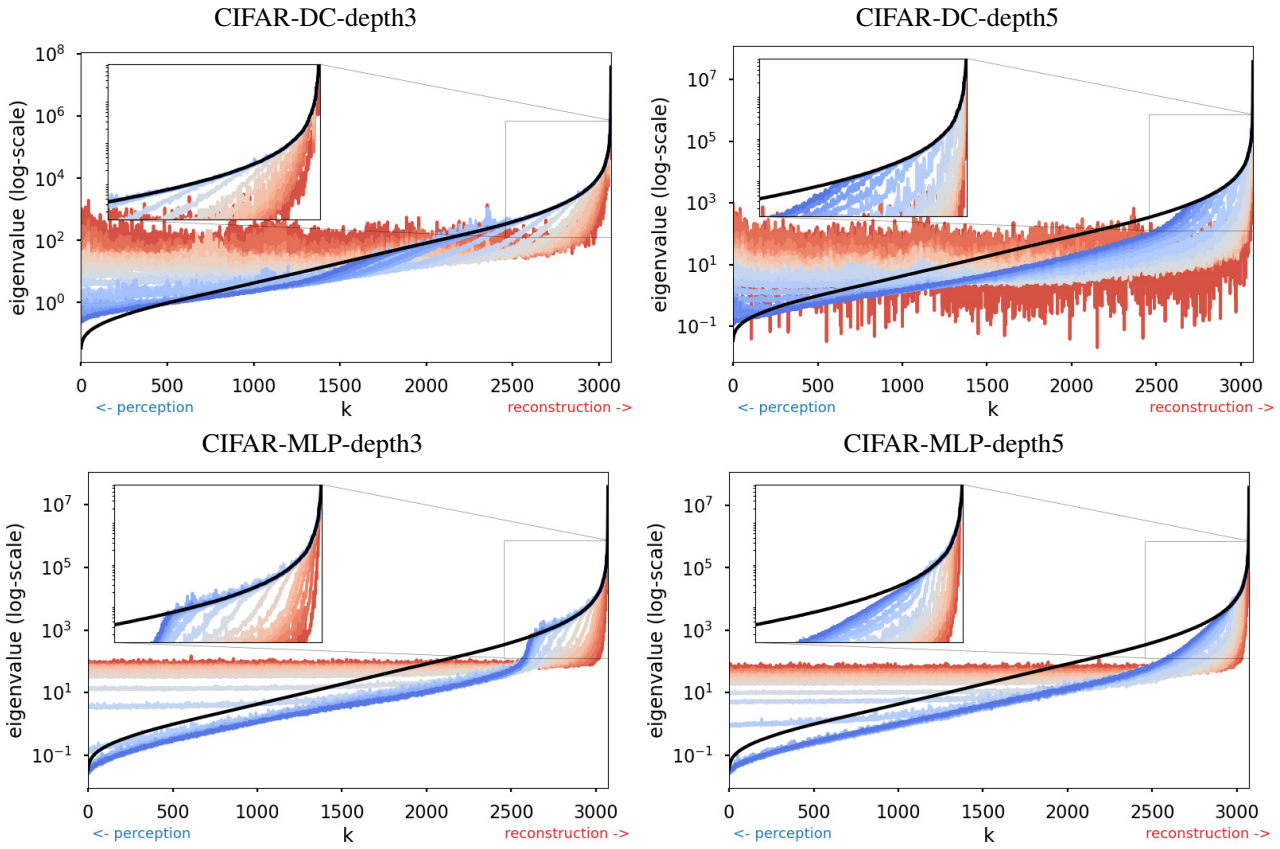

Figure 3. Reprise of Fig. 1 for additional autoencoder architectures: convolutional encoder and deconvolutional decoder (**top**) and MLP encoder and decoder (**bottom**). We clearly observe that the top subspace is learned first during training, which is the one that best minimize the reconstruction loss but that contains the least informative features for perception, as per Fig. 4.

settings.

Findings. We now propose to evaluate the closed form alignment metric (Eq. (7)) on a few datasets. Note that it can be implemented efficiently as detailed in Section 5.8. In Fig. 2, we measure the metric of Eq. (7) for a sweep of latent dimension K over 7 different datasets. We observe three striking regimes. First, for images without background, reconstruction and classification tasks are very much aligned even for small latent dimension, as low as 20% of the input dimension. Second, when comparing datasets with same image distributions but different number of classes (CIFAR10 to CIFAR100) the misalignment increases between the two task, especially for small embedding dimension. This follows our intuition that additional budget must be devoted to separate more classes. And as the subspace of the data used for reconstruction and classification do not align (recall Proposition 1), a greater misalignment is measured. And third, when looking at more realistic images (higher resolution and more diverse) such as tiny-imagenet, we observe that the alignment only increases linearly with the latent space dimension K, requiring $K = D$ in that case to ensure alignment. We thus conclude that *the presence of background, finer classification tasks, and higher resolution images, are all factors that drastically decrease the alignment between learning features for perception tasks and learning features that reconstruct those images.*

Linear regime results are informative. We have focused on the linear encoder, decoder, and classification head of Eq. (3). Albeit insightful, one might wonder how much of those insights transfer to the more realistic setting of employing nonlinear mappings. We note that it remains common to keep the classification head linear therefore leading to the following generalization of Eq. (3) as

$$\mathcal{L}(\boldsymbol{W}, \theta, \gamma) = \|\boldsymbol{W}^\top f_\theta(\boldsymbol{X}) - \boldsymbol{Y}\|_F^2 + \lambda \|g_\gamma(f_\theta(\boldsymbol{X})) - \boldsymbol{X}\|_F^2. \tag{8}$$

The encoder is now the nonlinear mapping $f_\theta : \mathbb{R}^D \mapsto \mathbb{R}^K$ and the decoder is the nonlinear mapping $g_\gamma : \mathbb{R}^K \mapsto \mathbb{R}^D$. We formalize below a result that will reinforce the legitimacy of our linear regime analysis (Theorem 1, Proposition 1, and Corollary 1.2) by showing that it is (i) a correct model during the early phase of training, and even (ii) a correct model throughout training when the decoder being employed is under-parametrized.

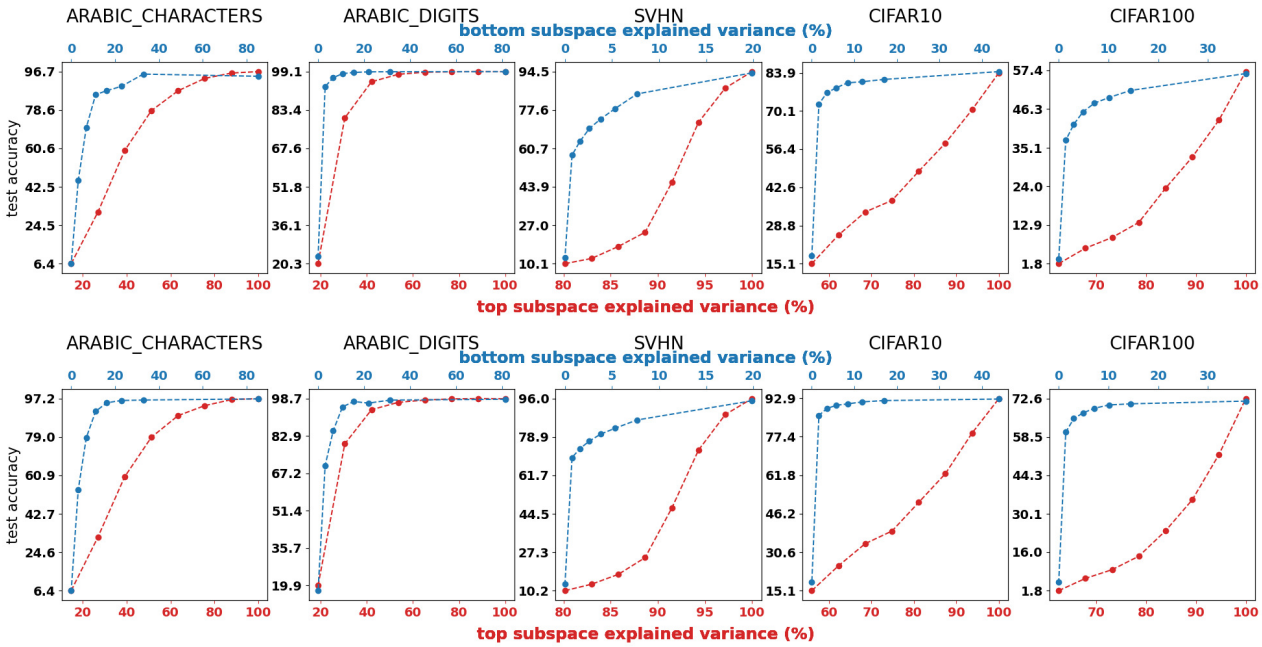

Figure 4. We depict the classification accuracy of a ResNet9 DNN when trained and tested on images that have been projected onto the top (**red**) and bottom (**blue**) subspace as ordered per the eigenvalues of the data covariance matrix, without data-augmentation (**top**) and with data-augmentation (**bottom**). We clearly observe that except for datasets without background and for which reconstruction and classification are better aligned (recall Fig. 2), the final performance is greater when employing the subspace of the data that explains the least the pixel variation, i.e., the bottom subspace. Training was done with cross-entropy loss, Fig. 8 provides the MSE case with one-hot labels as target showcasing the same trends.

Theorem 2. *For any high-capacity encoder f_θ, studying Eq. (3) and Eq. (8) is equivalent at initialization for any decoder, and is always equivalent when the decoder is linear. (Proof in Section 5.6.)*

Combining Theorems 1 and 2, we obtain that even with DNs, during the early stages of learning, the encoder-decoder mapping focuses on the principal subspace of the data, i.e., the space that explains most of the reconstruction error in the linear regime. As our study strongly hinges on that claim, we propose to empirically validate it in the following Section 3.2.

3.2. Reconstruction and Perception Features Live In Different Subspaces of the Data

We characterized in the previous Section 3.1 how classification and reconstruction tasks fail to align when it comes to learning common features. In particular, Section 3.1 and Fig. 2 validated how training focuses first on the top subspace of the data. We now reinforce our claim by showing that supervised tasks can not be solved when restricting the images on the subspace that is learned first by reconstruction.

Perception can not be solved from the principal subspace of the data. We first propose a controlled experiment where we artificially remove some of the original data subspace. In particular, we consider two settings. First, we gradually remove the subspace associated to the top eigenvectors of the data covariance matrix effectively removing what is most useful for reconstruction but also what we claim to be least useful for perception. Second, we gradually remove the subspace associated to the bottom eigenvectors (the one least useful for reconstruction but that we claim to be most useful for perception). This procedure is applied to the entire dataset (train and test images) before any DN training occurs. Hence the DN is only presented with the filtered images. We report the top-1 accuracy over numerous datasets in Fig. 1 (top) and in Figs. 4 and 8 for cross-entropy and MSE loss respectively. Recall that it has been previously observed that cross-entropy or MSE with one-hot labels produce similar representations (Hui & Belkin, 2020). We obtain a few key observations. First, *for any % of filtering, keeping the bottom subspace of the data produces higher test accuracy that when keeping the top subspace*. That is, the subspace that is most useful for reconstruction (top) is least useful for perception. Second, *the accuracy gap is impacted by the presence of background, finer-grained classes, and higher resolution images.* This further validates our theoretical

observations from Fig. 2 and the result from Theorem 2. We will now focus on validating the second part of our claim that the subspace used for perception (bottom) is learned last, and slowly.

Useful features for perception are learned last. The above results demonstrate that the top subspace of the data–explaining most of the pixel variance–is not aligned with the perception tasks. Yet, perfect reconstruction implies capturing both the top and bottom subspace. Albeit correct, we demonstrate in Fig. 1 (bottom) and in Fig. 3 that the rate at which the bottom subspace is learned is exponentially slower than the rate at which the top subspace is learned. This empirically validates Theorem 2. For the reader familiar with optimization (Benzi, 2002), or power iteration methods, this observation is akin to their converge rate which is a function of the eigengap (Booth, 2006; Xu et al., 2018), i.e., the difference between λ_i and λ_{i+1}, where λ are the sorted eigenvalues. Because natural images have an exponential decay of their eigenvalues (Van der Schaaf & van Hateren, 1996; Ruderman, 1997), the rate at which the top subspace is approximated is exponentially faster than the bottom one, therefore making the learning of useful features for perception occur only late during training. This finding also resonates with previous studies on the spectral bias of DNs in classification and generative settings (Chakrabarty & Maji, 2019; Rahaman et al., 2019; Schwarz et al., 2021).

Combining the observations from this section supports **R1** and **R2** from Section 1. It remains to study **R3** which states that since features for perception lie within a negligible subspace (as measured by the reconstruction loss), one can find two separate models that equally solve the reconstruction task (same train and test loss values) but provide drastically different perception task performances.

4. Learning By Reconstruction Needs Guidance

We now turn to **R3** stating that features for perceptions lie within a space with negligible impact on the reconstruction error–therefore motivating the need to add additional guidance to the training, e.g., through denoising tasks. To do so, we will show that it is possible to obtain two DNs with same reconstruction error but one with perception capabilities far greater than the other (Section 4.1). Lastly we will prove that some guidance can be provided to the learned representation to reduce that gap and focus towards more useful features through careful design of the denoising task (Section 4.2).

4.1. Learning By Reconstruction Can Produce Optimal Representations

One interesting benefit emerging from the observations made for **R1** and **R2** is that guiding a DN to focus on the subspace containing informative features for perception has minimal impact on the reconstruction loss–as they focus on different subspaces. Therefore, we now propose a simple experiment to demonstrate the above argument. We take a resnet34 autoencoder train it with the usual reconstruction loss (MSE) on Imagenette. This gives us a model that (as per R1 and R2) fails to properly focus on discriminative feature. To obtain the second DN with improved classification performance, we simply add a classification head on top of the embedding of the encoder. That is, the same embedding that is fed to the decoder for reconstruction, is also fed to the linear classifier with supervised training loss (recall Eq. (8)). We obtain the key insight of **R3** which is that one can produce two DNs with same training loss (reconstruction) and validation loss (reconstruction), but with significantly different classification performance, as reported in Figs. 5 and 7.

To further understand that observation, we can recall the results from Theorem 1. We demonstrated that the encoder ($\boldsymbol{V}$) having K dimensions at its disposal, is optimal when selecting the top singular vectors of the data matrix $\boldsymbol{X}$. If K is large enough that it encompasses both the top subspace of the data (which is learned first and has greatest impact on the reconstruction loss) and the bottom subspace of the data (which is useful for perception as per Figs. 1 and 4), then both objective can coexist (recall Proposition 1) as long as enough capacity is given to the encoder. Therefore, we obtain the following key insight. *Whenever the capacity of the autoencoder is large, the encoder embedding can (and will at the end of training) include features useful for perception all while being able to reconstruct its inputs*. Again, for this to happen one requires the capacity of the encoder to grow with the image resolution (as more and more dimensions will be taken up by the top subspace), and with the complexity of the image background (again taking more dimensions in the top subspace) (recall Section 3.2).

The above observation demonstrates that learning to reconstruct needs an additional training signal to focus towards discriminative features. As we will prove below, learning by denoising offers such a solution.

4.2. Provable Benefits of Learning by Denoising

Recalling the Denoising Autoencoder setting from Eq. (2), we aim to obtain a closed form solution of the linear loss Eq. (3) in order to find some hints as to why masking and additive Gaussian noise produce representations of different quality for perception.

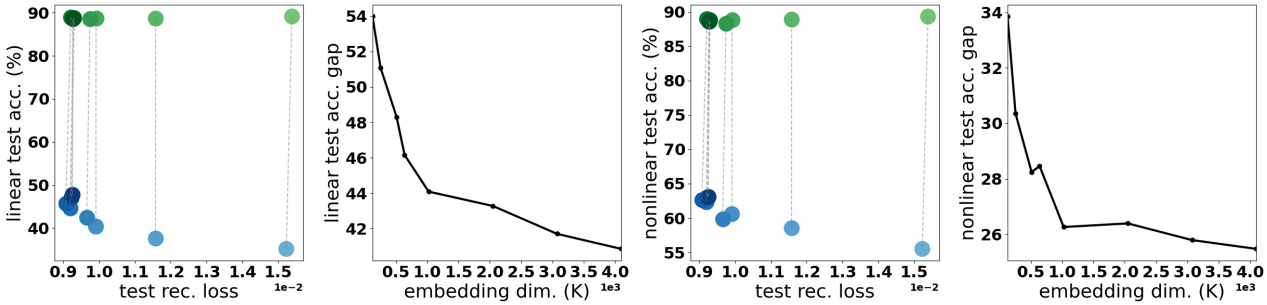

Figure 5. Depiction of multiple resnet34 autoencoders with varying embedding dimensions (**light to dark**) some trained only to reconstruct the input samples with data-augmentations (**blue**) and others with an additional supervised loss signal (as per Eq. (8)) (**green**). We report the test set accuracy and the relative difference (**y-axis**) for each of the "paired" models, i.e., the ones with every training setting identical except for the use of the supervised signal, as a function of the train and test rec loss. We clearly observe that for any embedding dimension and reconstruction loss, one can find two set of parameters with drastically different ability to solve perception tasks. Reconstructed samples and training curves are provided in Fig. 7.

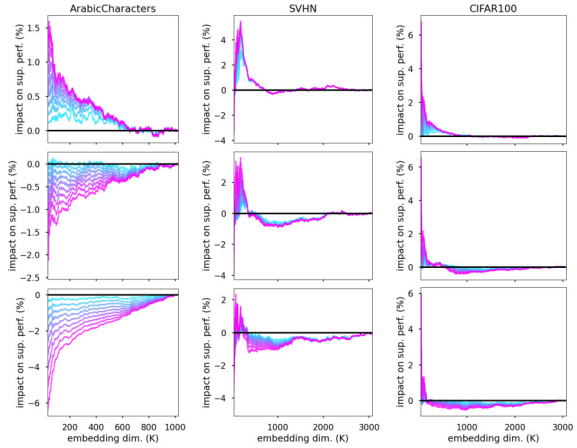

Figure 6. Depiction of the relative alignment difference when employing denoising tasks (recall Eq. (9)) with masking noise, with probability of dropping ranging from 0% to 99% (**cyan** to **pink**) for patch size of $(1, 1)$ recovering multiplicative dropout (**top**), $(2, 2)$ (**middle**), and $(4, 4)$ (**bottom**) on various datasets. A positive number indicates a beneficial impact of using the denoising loss on the supervised performance of the learned representation. We observe that for datasets such as ArabicDigits that already have a strong alignment between the two tasks (recall Fig. 2), the use of any form of masking is detrimental except with shape $(1, 1)$. However for datasets such as CIFAR100 (**right column**) with originally poor alignment, masking is beneficial and increases the alignment between the two tasks. As the original alignment increases with K, as the benefit of masking reduces.

Our goal is therefore to study the misalignment metric (Eq. (7)) under the denoising setting which, as per Corollary 1.2, is given by

$$\text{alignment}(k) \triangleq \min_{\boldsymbol{W}} \|\boldsymbol{W}^\top \boldsymbol{V}^{*\top} \boldsymbol{X} - \boldsymbol{Y}\|_F^2,$$
$$\boldsymbol{V}^* = \arg\min_{\boldsymbol{V}} \min_{\boldsymbol{Z}} \mathbb{E}_{\boldsymbol{X}'|\boldsymbol{X}} \left[\|\boldsymbol{Z}^\top \boldsymbol{V}^\top \boldsymbol{X}' - \boldsymbol{X}\|_F^2\right], \quad (9)$$

which is the minimum supervised loss that can be attained using the representation from $\boldsymbol{V}^*$ that minimizes the denoising loss. We ought to highlight that we can obtain a closed form solution under the expectation over the noise distribution $(\boldsymbol{X}'|\boldsymbol{X})$ as formalized below, where we denote $G \triangleq \mathbb{E}_{\boldsymbol{X}'|\boldsymbol{X}} \left[\boldsymbol{X}'\boldsymbol{X}'^\top\right]$ and $\boldsymbol{S} \triangleq \mathbb{E}_{\boldsymbol{X}'|\boldsymbol{X}} \left[\boldsymbol{X}'\right]$.

Theorem 3. *The closed form solution for $\boldsymbol{V}^*$ from Eq. (9) is given by $\boldsymbol{V}^*$ spans $\boldsymbol{P_G} \boldsymbol{D_G}^{-\frac{1}{2}} (\boldsymbol{P_H})_{.,1:K}$, where $\boldsymbol{H} \triangleq \boldsymbol{D_G}^{-\frac{1}{2}} \boldsymbol{P_G}^\top \boldsymbol{S}\boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{S}^\top \boldsymbol{P_G} \boldsymbol{D_G}^{-\frac{1}{2}}$. (Proof in Section 5.4.)*

The above result demonstrates that even when employing denoising autoencoders with additive Gaussian noise or masking, we can obtain a closed form solution for $\boldsymbol{V}$, and from that obtain all the alignment metrics studied so far. In particular, Section 5.4 also provides the closed form solutions for $\boldsymbol{G}$ and $\boldsymbol{S}$. We illustrate the alignment between reconstruction and perception tasks in the denoising autoencoder setting (Eq. (9)) in Fig. 6 for the case of random masking, as per the MAE setting. We clearly observe that the denoising task has the ability to increase the alignment between the two tasks–especially for small embedding dimension (K). We however observe that the size of the masked patches that provides the best gains vary with the dataset, hinging at another challenge of denoising autoencoders: the cross-validation of the denoising task. Another formal result we propose below will reinforce that point.

Denoting by $\boldsymbol{V}^*(\sigma)$ the optimal denoising autoencoder parameters when employing additive isotropic Gaussian noise with standard deviation σ, we obtain the following statement showcasing that this type of denoising task does not help supervised tasks.

Corollary 3.1. *Under the settings of Theorem 3, additive Gaussian noise has no impact in the supervised task performance as $\boldsymbol{W}^{*\top} \boldsymbol{V}^*(\sigma)^\top = \boldsymbol{W}^{*\top} \boldsymbol{V}^*(0)^\top, \forall \sigma \geq 0$, regardless of the supervised task. (Proof in Section 5.5.)*

We therefore obtain the following insights. *Denoising tasks offer a powerful guidance to skew learned representations to better align with perception tasks (Fig. 6) but some noise distributions such as additive Gaussian noise are provably unable to help*. A challenge that naturally emerges is in selecting the adequate denoising tasks, e.g., to avoid Corollary 3.1 in a setting where labels are not available and the supervised tasks to be tackled may not be known a priori. An interesting portal that we obtained (Corollary 3.1) in our study is that it is possible to assess if a denoising task has any impact on the perception task without yet knowing what is that supervised task. That alone could help in at least focusing on denoising tasks that do have an impact, albeit it will remain unknown if that impact will be beneficial or not.

5. Conclusion

We proposed to study the transferability of representations learned by reconstruction towards perception tasks. In particular, we obtain that the two objectives are fundamentally misaligned, with a degree of misalignment that grows with the presence of complicated background, with greater number of classes for the perception task, and with higher image resolutions. While our study focused on bringing those limitations forward from a theoretical and empirical angle, we also opened new avenues to reduce those limitations in the future. For example, we obtained a closed form solution to measure the impact of noise distributions to better align the learned representation to the downstream perception task. This novel methodology opens the door to a priori selecting noise distribution candidates. Even when the downstream task is unknown, we found that some noise distributions, such as additive Gaussian noise, are effectively unable to provide any benefit for better aligning reconstruction and perception tasks. On the opposite, we validated that masking is a valid strategy, albeit requiring some per-dataset tuning. That finding is in line with MAE's performances going from about 50% to 74% on Imagenet top-1 accuracy when masking is employed. We hope that our study will also open new avenues to study reconstruction methods for other modalities such as time-series and NLP.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Baidoo-Anu, D. and Ansah, L. O. Education in the era of generative artificial intelligence (ai): Understanding the potential benefits of chatgpt in promoting teaching and learning. *Journal of AI*, 7(1):52–62, 2023.

Balestriero, R., Misra, I., and LeCun, Y. A data-augmentation is worth a thousand samples: Analytical moments and sampling-free training. *Advances in Neural Information Processing Systems*, 35:19631–19644, 2022.

Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian, Y., et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end optimization of nonlinear transform codes for perceptual quality. In *2016 Picture Coding Symposium (PCS)*, pp. 1–5. IEEE, 2016.

Bao, J., Chen, D., Wen, F., Li, H., and Hua, G. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE international conference on computer vision*, pp. 2745–2754, 2017.

Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Barlow, H. B. Unsupervised learning. *Neural computation*, 1(3): 295–311, 1989.

Benzi, M. Preconditioning techniques for large linear systems: a survey. *Journal of computational Physics*, 182(2):418–477, 2002.

Booth, T. E. Power iteration method for the several largest eigenvalues and eigenfunctions. *Nuclear science and engineering*, 154(1):48–62, 2006.

Bordes, F., Balestriero, R., and Vincent, P. High fidelity visualization of what your self-supervised representation knows about. *arXiv preprint arXiv:2112.09164*, 2021.

Bottou, L. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade: Second Edition*, pp. 421–436. Springer, 2012.

Brynjolfsson, E., Li, D., and Raymond, L. R. Generative ai at work. Technical report, National Bureau of Economic Research, 2023.

Chakrabarty, P. and Maji, S. The spectral bias of the deep image prior. *arXiv preprint arXiv:1912.08905*, 2019.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

Chen, X., Mishra, N., Rohaninejad, M., and Abbeel, P. Pixelsnail: An improved autoregressive generative model. In *International Conference on Machine Learning*, pp. 864–872. PMLR, 2018.

Dilokthanakul, N., Mediano, P. A., Garnelo, M., Lee, M. C., Salimbeni, H., Arulkumaran, K., and Shanahan, M. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

Esmaeili, B., Wu, H., Jain, S., Bozkurt, A., Siddharth, N., Paige, B., Brooks, D. H., Dy, J., and Meent, J.-W. Structured disentangled representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2525–2534. PMLR, 2019.

Garrido, Q., Balestriero, R., Najman, L., and Lecun, Y. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *International Conference on Machine Learning*, pp. 10929–10974. PMLR, 2023.

Ghahramani, Z. Unsupervised learning. In *Summer school on machine learning*, pp. 72–112. Springer, 2003.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Hui, L. and Belkin, M. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *arXiv preprint arXiv:2006.07322*, 2020.

Hyvärinen, A. and Dayan, P. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Jiang, Z., Zheng, Y., Tan, H., Tang, B., and Zhou, H. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Kulis, B. et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521 (7553):436–444, 2015.

Lim, K.-L., Jiang, X., and Yi, C. Deep clustering with variational autoencoder. *IEEE Signal Processing Letters*, 27:231–235, 2020.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Mahendran, A. and Vedaldi, A. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196, 2015.

Mathieu, E., Rainforth, T., Siddharth, N., and Teh, Y. W. Disentangling disentanglement in variational autoencoders. In *International conference on machine learning*, pp. 4402–4412. PMLR, 2019.

Olah, C., Mordvintsev, A., and Schubert, L. Feature visualization. *Distill*, 2(11):e7, 2017.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pp. 5301–5310. PMLR, 2019.

Ruderman, D. L. Origins of scaling in natural images. *Vision research*, 37(23):3385–3398, 1997.

Schwarz, K., Liao, Y., and Geiger, A. On the frequency bias of generative models. *Advances in Neural Information Processing Systems*, 34:18126–18136, 2021.

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., and Batra, D. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

Seydoux, L., Balestriero, R., Poli, P., Hoop, M. d., Campillo, M., and Baraniuk, R. Clustering earthquake signals and background noises in continuous seismic data with unsupervised deep learning. *Nature communications*, 11(1):3972, 2020.

Shen, Y., Gu, J., Tang, X., and Zhou, B. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9243–9252, 2020.

Tran, L., Yin, X., and Liu, X. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1415–1424, 2017.

Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

Van der Schaaf, v. A. and van Hateren, J. v. Modelling the power spectra of natural images: statistics and information. *Vision research*, 36(17):2759–2770, 1996.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3): 37–52, 1987.

Xu, P., He, B., De Sa, C., Mitliagkas, I., and Re, C. Accelerated stochastic power iteration. In *International Conference on Artificial Intelligence and Statistics*, pp. 58–67. PMLR, 2018.

Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., and Savarese, S. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3712–3722, 2018.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pp. 818–833. Springer, 2014.

SUPPLEMENTARY MATERIALS

5.1. Proof of Theorem 1

Proof. The first part of the proof finds the optimum $\boldsymbol{W}^*$ and $\boldsymbol{Z}^*$ as a function of $\boldsymbol{V}$ which is direct since we are in a least-square style setting for each of them. The second part will consist in showing that the optimal $\boldsymbol{V}$ can be found as the solution of a generalized eigenvalue problem. The third and final step will be to express the solution for $\boldsymbol{V}$ in close-form that is also friendly for computations.

Step 1. Recall that our loss function is given by

$$\mathcal{L} = \|\boldsymbol{W}^\top \boldsymbol{V}^\top \boldsymbol{X} - \boldsymbol{Y}\|_F^2 + \lambda \|\boldsymbol{Z}^\top \boldsymbol{V}^\top \boldsymbol{X} - \boldsymbol{X}\|_F^2,$$

recalling that $\|\boldsymbol{M}\|_F^2 = \mathrm{Tr}(\boldsymbol{M}^\top \boldsymbol{M})$, the above simplifies to

$$\begin{aligned}\mathcal{L} =& \|\boldsymbol{Y}\|_F^2 - 2\,\mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{W}\boldsymbol{Y}\right) + \mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{W}\boldsymbol{W}^\top \boldsymbol{V}^\top \boldsymbol{X}\right) \\ &+ \lambda \|\boldsymbol{X}\|_F^2 - 2\lambda\,\mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{Z}\boldsymbol{X}\right) + \lambda\,\mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{Z}\boldsymbol{Z}^\top \boldsymbol{V}^\top \boldsymbol{X}\right),\end{aligned}$$

we are now going to find the optimal $\boldsymbol{W}$ and $\boldsymbol{Z}$ which are unique by convexity of the loss and of their domain. Recall that we assume $\boldsymbol{Y}$ and $\boldsymbol{X}$ to be full-rank (therefore also making $\boldsymbol{V}$ full-rank). Recalling the derive of traces, we obtain

$$\begin{aligned}\nabla_{\boldsymbol{W}} \mathcal{L} &= -2\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{Y}^\top + 2\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{W}, \\ \nabla_{\boldsymbol{Z}} \mathcal{L} &= -2\lambda \boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top + 2\lambda \boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{Z},\end{aligned}$$

setting it to zero (we assume here $\lambda > 0$ otherwise we can not solve for $\boldsymbol{Z}$ since its value does not impact the loss) and solving leads

$$\begin{aligned}\boldsymbol{W}^* &= (\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V})^{-1} \boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{Y}^\top, \\ \boldsymbol{Z}^* &= (\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V})^{-1} \boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top.\end{aligned}$$

We now have solved for $\boldsymbol{W}, \boldsymbol{Z}$ as a function of $\boldsymbol{V}$, i.e., the loss is now only a function of $\boldsymbol{V}$, which we are going to solve for now.

Step 2. We will first proceed by plugging the values for $\boldsymbol{W}^*, \boldsymbol{Z}^*$ back into the loss, which will now be only a function of $\boldsymbol{V}$. Let's first simplify our derivations by noticing that

$$\begin{aligned}\mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{W}^*\boldsymbol{W}^{*\top} \boldsymbol{V}^\top \boldsymbol{X}\right) &= \mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}(\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V})^{-1} \boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{Y}^\top \boldsymbol{Y}\boldsymbol{X}^\top \boldsymbol{V}(\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V})^{-1} \boldsymbol{V}^\top \boldsymbol{X}\right) \\ &= \mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}(\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V})^{-1} \boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{Y}^\top \boldsymbol{Y}\right) \\ &= \mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{W}^*\boldsymbol{Y}\right),\end{aligned}$$

and similarly

$$\mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{Z}^*\boldsymbol{Z}^{*\top} \boldsymbol{V}^\top \boldsymbol{X}\right) = \mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{Z}^*\boldsymbol{X}\right),$$

finally making the entire loss simplify as follows

$$\begin{aligned}\mathcal{L} &= \|\boldsymbol{Y}\|_F^2 - 2\,\mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{W}^*\boldsymbol{Y}\right) + \mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{W}^*\boldsymbol{W}^{*\top} \boldsymbol{V}^\top \boldsymbol{X}\right) \\ &\qquad + \lambda \|\boldsymbol{X}\|_F^2 - 2\lambda\,\mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{Z}^*\boldsymbol{X}\right) + \lambda\,\mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{Z}^*\boldsymbol{Z}^{*\top} \boldsymbol{V}^\top \boldsymbol{X}\right) \\ &= \|\boldsymbol{Y}\|_F^2 - \mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{W}^*\boldsymbol{Y}\right) + \lambda \|\boldsymbol{X}\|_F^2 - \lambda\,\mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}\boldsymbol{Z}^*\boldsymbol{X}\right) \\ &= \|\boldsymbol{Y}\|_F^2 - \mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}(\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V})^{-1} \boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{Y}^\top \boldsymbol{Y}\right) \\ &\qquad + \lambda \|\boldsymbol{X}\|_F^2 - \lambda\,\mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}(\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V})^{-1} \boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{X}\right) \\ &= \|\boldsymbol{Y}\|_F^2 + \lambda \|\boldsymbol{X}\|_F^2 - \mathrm{Tr}\left(\boldsymbol{X}^\top \boldsymbol{V}(\boldsymbol{V}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{V})^{-1} \boldsymbol{V}^\top \boldsymbol{X}\left(\boldsymbol{Y}^\top \boldsymbol{Y} + \lambda \boldsymbol{X}^\top \boldsymbol{X}\right)\right).\end{aligned}$$

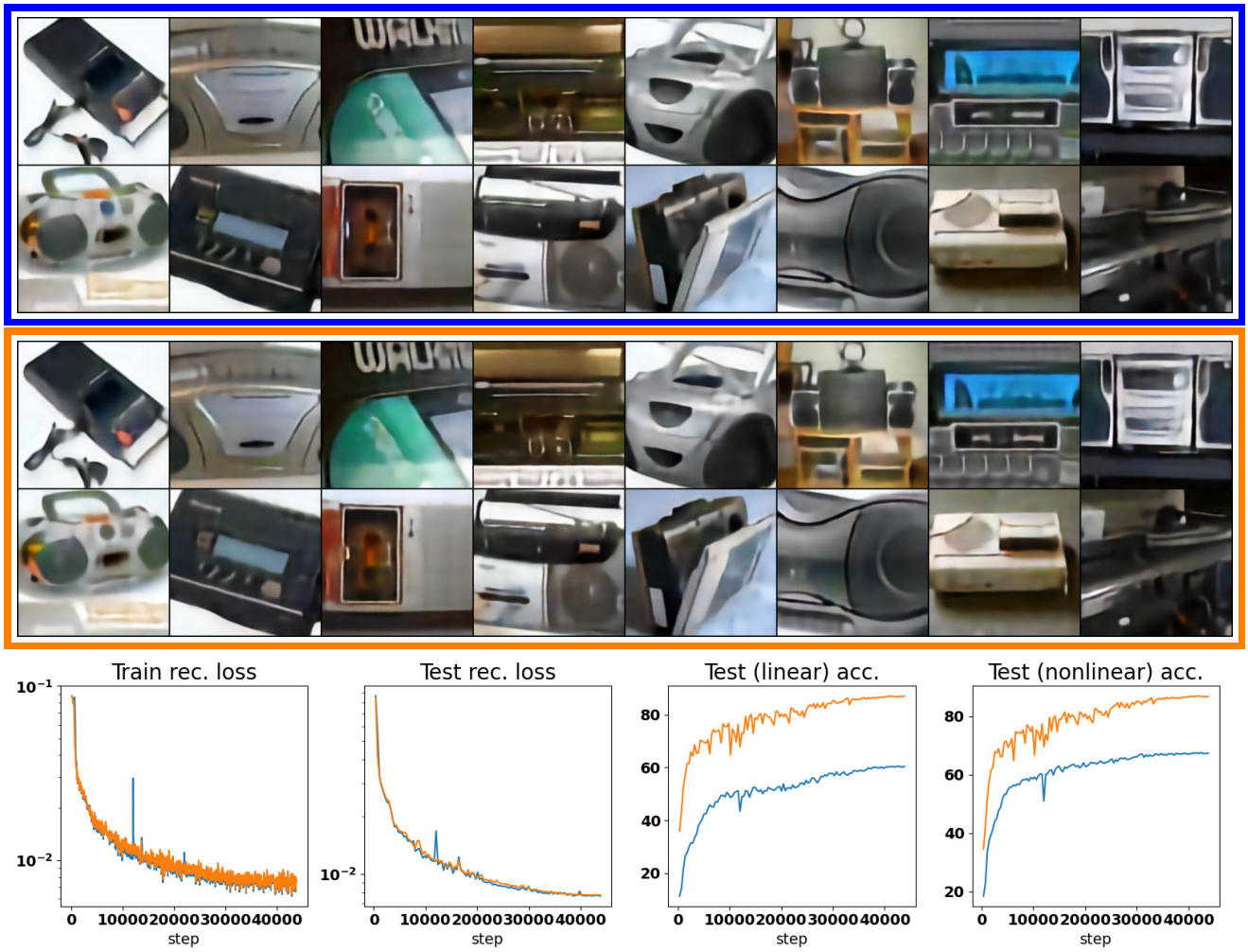

Figure 7. Depiction of two resnet34 autoencoders trained on Imagenette (Imagenet-10) images, one (orange) with an additional training signal that favors latent representations suited for classification, and the other (blue) that is only the reconstruction loss. As per **R1** and **R2** the latter naturally focuses on suboptimal features as showcases in the test accuracy, both when using a linear or a nonlinear probe. Crucially, the autoencoder with the additional signal produces representations with much greater discriminative power in both the linear and nonlinear setting. Yet, and despite popular belief, doing so has no impact on the reconstruction losses on the train or test set, and thus no impact on the quality of the reconstruction presented at the **top**. Therefore validating **R3**.

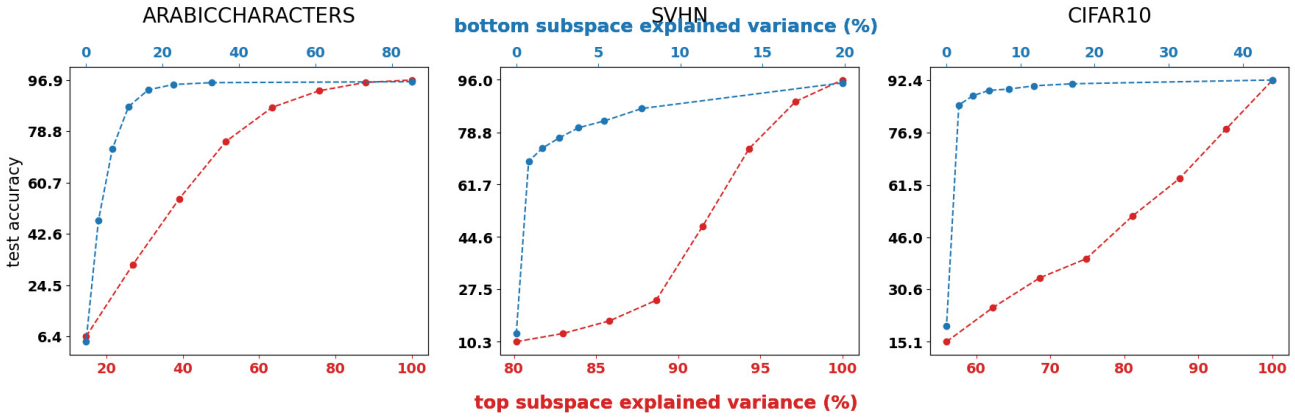

Figure 8. Reprise of Fig. 4 but now with MSE loss, the same observations can be made.

First, notice that both $\boldsymbol{X}\left(\boldsymbol{Y}^\top\boldsymbol{Y}+\lambda\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{X}^\top$ and $\boldsymbol{X}\boldsymbol{X}^\top$ are symmetric. Therefore, we can minimize the loss by solving the following generalized eigenvalue problem:

$$\text{find } \boldsymbol{V}\in\mathcal{M}_{D,D}(\mathbb{R}) \text{ so that } \boldsymbol{X}\left(\boldsymbol{Y}^\top\boldsymbol{Y}+\lambda\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{X}^\top\boldsymbol{V}=\boldsymbol{v}^\top\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{V}\Lambda,$$

where $\boldsymbol{V}$ are the eigenvectors of the generalized eigenvalue problem, and Λ the eigenvalues, then the solution to our problem we be any rotation of any K eigenvectors, but the minimum will be achieve for the top-K ones.

Step 3. We will first demonstrate the general solution for the generalized eigenvalue problem. Given that solution, it will be easy to take the top-K eigenvectors that solve the considered problem. Denoting $\boldsymbol{A}\triangleq\boldsymbol{X}\left(\boldsymbol{Y}^\top\boldsymbol{Y}+\lambda\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{X}^\top$ and $\boldsymbol{B}\triangleq\boldsymbol{X}\boldsymbol{X}^\top$, and $\boldsymbol{H}\triangleq\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{B}}^\top\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{B}}\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}$, we have

$$\begin{aligned}
&\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{B}}\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{H}}=\boldsymbol{B}\boldsymbol{P}_{\boldsymbol{B}}\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{H}}\Lambda\\
\iff\ &\boldsymbol{P}_{\boldsymbol{H}}^\top\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{B}}^\top\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{B}}\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{H}}=\boldsymbol{P}_{\boldsymbol{H}}^\top\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{B}}^\top\boldsymbol{B}\boldsymbol{P}_{\boldsymbol{B}}\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{H}}\Lambda \qquad (\boldsymbol{P}_{\boldsymbol{H}}^\top\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{B}}^\top \text{ bijective})\\
\iff\ &\boldsymbol{P}_{\boldsymbol{H}}^\top\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{B}}^\top\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{B}}\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{H}}=\Lambda\\
&\iff \boldsymbol{D}_{\boldsymbol{H}}=\Lambda,
\end{aligned}$$

therefore the eigenvalues are given by $\boldsymbol{D}_{\boldsymbol{H}}$ and the eigenvectors are given by $\boldsymbol{P}_{\boldsymbol{B}}\boldsymbol{D}_{\boldsymbol{B}}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{H}}$ or equivalently $(\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K}$. So the optimal $\boldsymbol{V}$ is any rotation of the top-K eigenvectors. The above is simple to use as-is whenever $N>D$, if not, then we can obtain a solution without having to compute any $D\times D$ matrix, thus making the process more efficient. To that end, we can obtain

$$\begin{aligned}
\boldsymbol{M}&\triangleq\boldsymbol{Y}^\top\boldsymbol{Y}+\lambda\boldsymbol{X}^\top\boldsymbol{X}\\
\boldsymbol{S}&\triangleq\boldsymbol{D}_{\boldsymbol{M}}^{\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{M}}^\top\boldsymbol{X}^\top\\
\boldsymbol{P}_{\boldsymbol{A}}=\boldsymbol{V}_{\boldsymbol{S}},&\ \boldsymbol{D}_{\boldsymbol{A}}=\boldsymbol{\Sigma}_{\boldsymbol{S}}^2,
\end{aligned}$$

that only involves $D\times\min(D,N)$ matrices instead of $D\times D$.

□

5.2. Proof of Corollary 1.1

Proof. We will start with the fully supervised (least-square) proof obtained when $\lambda=0$. Also notice that in any case, we have that

$$\boldsymbol{X}=\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{\frac{1}{2}}\boldsymbol{V}_{\boldsymbol{X}}^\top.$$

Ordinary Least Square recovery. since $\lambda = 0$ we also have

$$\boldsymbol{A} = \boldsymbol{X}\left(\boldsymbol{Y}^\top\boldsymbol{Y} + \lambda\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{X}^\top = \boldsymbol{X}\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{X}^\top$$

when then lead to

$$\begin{aligned}
\boldsymbol{H} =& \boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}^\top\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}} \\
=& \boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}^\top\boldsymbol{X}\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{X}^\top\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}} \\
=& \boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}^\top\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{\frac{1}{2}}\boldsymbol{V}_{\boldsymbol{X}}^\top\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{V}_{\boldsymbol{X}}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}^\top\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}} \\
=& \boldsymbol{V}_{\boldsymbol{X}}^\top\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{V}_{\boldsymbol{X}},
\end{aligned}$$

from the above, we can simply plug those values in the analytical form for $\boldsymbol{V}^*$ from Eq. (4) to obtain

$$\boldsymbol{V} = \boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{H}} = \boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}}\boldsymbol{V}_{\boldsymbol{X}}^\top\boldsymbol{V}_{\boldsymbol{Y}} = \boldsymbol{X}^\dagger(\boldsymbol{V}_{\boldsymbol{Y}})_{.,1:K}$$

since we easily see that $\boldsymbol{P}_{\boldsymbol{H}} = \boldsymbol{V}_{\boldsymbol{X}}^\top\boldsymbol{V}_{\boldsymbol{Y}}$. We also have the optimum for $\boldsymbol{W}$ from Eq. (5) to be

$$\begin{aligned}
\boldsymbol{W} =& (\boldsymbol{V}^{*\top}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{V}^*)^{-1}\boldsymbol{V}^{*\top}\boldsymbol{X}\boldsymbol{Y}^\top \\
=& (\boldsymbol{V}_{\boldsymbol{Y}}^\top\boldsymbol{V}_{\boldsymbol{Y}})^{-1}\boldsymbol{V}_{\boldsymbol{Y}}^\top\boldsymbol{V}_{\boldsymbol{X}}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}^\top\boldsymbol{X}\boldsymbol{Y}^\top \\
=& \boldsymbol{V}_{\boldsymbol{Y}}^\top\boldsymbol{V}_{\boldsymbol{X}}\boldsymbol{V}_{\boldsymbol{X}}^\top\boldsymbol{Y}^\top \\
=& \boldsymbol{\Sigma}_{\boldsymbol{Y}}\boldsymbol{U}_{\boldsymbol{Y}}^\top
\end{aligned}$$

and finally the product of both matrices (which produce the supervised linear model) is obtained as

$$\boldsymbol{W}^\top\boldsymbol{V}^\top = \boldsymbol{U}_{\boldsymbol{Y}}\boldsymbol{\Sigma}_{\boldsymbol{Y}}(\boldsymbol{V}_{\boldsymbol{Y}}{}_{.,1:K})^\top(\boldsymbol{X}^\dagger)^\top = \boldsymbol{Y}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{X}^\top)^{-1}, K \geq C,$$

therefore recovering the OLS optimal solution. Note that if $K < C$ then we have a bottleneck and we therefore obtain an interesting alternative solution that looks at the top subspace of $\boldsymbol{Y}$ (this is however never the case in OLS settings).

Principal Component Analysis recovery. We now consider the case where we only employ the unsupervised loss (akin to $\lambda \to \infty$). In this setting we get

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{X}^\top,$$

and we directly obtain

$$\begin{aligned}
\boldsymbol{H} =& \boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}^\top\boldsymbol{A}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}} \\
=& \boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}^\top\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}} \\
=& \boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top},
\end{aligned}$$

therefore the optimal form for $\boldsymbol{V}$ will be

$$\boldsymbol{V} = \boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}}(\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K} = \boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}})_{.,1:K},$$

which will select the top-K subspace of $\boldsymbol{X}$ (recall that the eigenvalues of $\boldsymbol{H}$ are $\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}$ and therefore its top-K eigenvectors are selected the top-K dimension of the subspace. Then the solution for $\boldsymbol{Z}$ from Eq. (6) gives

$$\begin{aligned}
\boldsymbol{Z} =& (\boldsymbol{V}^{*\top}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{V}^*)^{-1}\boldsymbol{V}^{*\top}\boldsymbol{X}\boldsymbol{X}^\top \\
=& \boldsymbol{V}^{*\top}\boldsymbol{X}\boldsymbol{X}^\top \\
=& ((\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{\frac{1}{2}})_{.,1:K})^\top\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}^\top,
\end{aligned}$$

and lastly the product of $\boldsymbol{Z}$ and $\boldsymbol{V}$ (which produce the final linear transformation processing $\boldsymbol{X}$ takes the form

$$\boldsymbol{Z}^\top\boldsymbol{V}^\top = \boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{\frac{1}{2}})_{.,1:K})((\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}^{-\frac{1}{2}})_{.,1:K})^\top\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}^\top = (\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K}((\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K})^\top,$$

which is the projection matrix onto the top-K subspace of the data $\boldsymbol{X}$ i.e. recovering the optimal solution of Principal Component Analysis. □

5.3. Proof of Corollary 1.2

Proof. Recall that in the $\lambda \to \infty$ regime, we have that $\boldsymbol{V}^* = \boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}(\boldsymbol{D}^{-\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K}$ and $\boldsymbol{W}^* = \boldsymbol{P}^\top_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{Y}^\top$. We thus develop

$$
\begin{aligned}
\|\boldsymbol{W}^{*\top}\boldsymbol{V}^{*\top}\boldsymbol{X} - \boldsymbol{Y}\|_F^2 =& \|\boldsymbol{Y}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}((\boldsymbol{D}^{-\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K})^\top\boldsymbol{P}^\top_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{X} - \boldsymbol{Y}\|_F^2 \\
=& \|\boldsymbol{Y}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}((\boldsymbol{D}^{-\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K})^\top\boldsymbol{P}^\top_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}^{\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{P}^\top_{\boldsymbol{X}\boldsymbol{X}^\top} - \boldsymbol{Y}\|_F^2 \\
=& \|\boldsymbol{Y}(\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K}((\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K})^\top - \boldsymbol{Y}\|_F^2 \\
=& \|\boldsymbol{Y}\|_F^2 - 2\operatorname{Tr}\left(\boldsymbol{Y}(\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K}((\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K})^\top\boldsymbol{Y}^\top\right) + \|\boldsymbol{Y}(\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K}((\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K})^\top\|_F^2 \\
=& \|\boldsymbol{Y}\|_F^2 - \operatorname{Tr}\left(\boldsymbol{Y}(\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K}((\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K})^\top\boldsymbol{Y}^\top\right) \\
=& \|\boldsymbol{Y}\|_F^2 - \|\boldsymbol{Y}(\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K}\|_F^2,
\end{aligned}
$$

as $\|\boldsymbol{Y}\|_F^2$ is a constant with respect to the parameters, we consider $\|\boldsymbol{Y}(\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top})_{.,1:K}\|_F^2$ as our alignment measure (the greater, the better the supervised loss can be minimized from the parameters). Since this quantity lives in the range $[0, \|\boldsymbol{Y}\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\|_F^2]$, we see that by using the reparametrization from Eq. (7) we obtain the proposed measure of alignment rescaled to $[0, 1]$. □

5.4. Proof of Theorem 3

We need to find the optimal reconstruction solution $(\boldsymbol{V}^*, \boldsymbol{Z}^*)$ first (corresponding to the case of $\lambda \to \infty$, and then plug it into the supervised loss with optimal $\boldsymbol{W}$.

$$
\mathbb{E}_{\boldsymbol{X}'_n \sim p_{\boldsymbol{X}'|\boldsymbol{X}}} \|\boldsymbol{W}^\top\boldsymbol{V}^\top\boldsymbol{X}' - \boldsymbol{X}\|_2^2,
$$

from which we obtain

$$
\min_{\boldsymbol{W},\boldsymbol{V}} \sum_{n=1}^{N} \mathbb{E}_{\boldsymbol{x}'_n \sim p_{\boldsymbol{x}'_n|\boldsymbol{x}_n}} \operatorname{Tr}\left(\boldsymbol{W}^\top\boldsymbol{V}^\top\boldsymbol{X}'\boldsymbol{X}'^\top\boldsymbol{V}\boldsymbol{W}\right) - 2\operatorname{Tr}\left(\boldsymbol{W}^\top\boldsymbol{V}^\top\boldsymbol{X}'\boldsymbol{X}^\top\right) + \text{cst},
$$

leading to $\boldsymbol{W}^* = (\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top]\boldsymbol{V})^{-1}\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}']\boldsymbol{X}^\top$ which we can plug back into the loss to obtain

$$
\begin{aligned}
&\operatorname{Tr}\left(\boldsymbol{X}\mathbb{E}[\boldsymbol{X}']^\top\boldsymbol{V}(\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top]\boldsymbol{V})^{-1}\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top]\boldsymbol{V}(\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top]\boldsymbol{V})^{-1}\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}']\boldsymbol{X}^\top\right) \\
&- 2\operatorname{Tr}\left(\boldsymbol{X}\mathbb{E}[\boldsymbol{X}']^\top\boldsymbol{V}(\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top]\boldsymbol{V})^{-1}\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}']\boldsymbol{X}^\top\right) + \text{cst}, \\
=& -\operatorname{Tr}\left(\boldsymbol{X}\mathbb{E}[\boldsymbol{X}']^\top\boldsymbol{V}(\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top]\boldsymbol{V})^{-1}\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}']\boldsymbol{X}^\top\right) + \text{cst}, \\
=& -\operatorname{Tr}\left(\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}']\boldsymbol{X}^\top\boldsymbol{X}\mathbb{E}[\boldsymbol{X}']^\top\boldsymbol{V}(\boldsymbol{V}^\top\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top]\boldsymbol{V})^{-1}\right) + \text{cst},
\end{aligned}
$$

whose solution is given (assuming $\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top]$ is full-rank) by the solution of the generalized eigenvalue problem

$$
\arg\max_{\boldsymbol{v}} \frac{\boldsymbol{v}^\top\mathbb{E}[\boldsymbol{X}']\boldsymbol{X}^\top\boldsymbol{X}\mathbb{E}[\boldsymbol{X}']^\top\boldsymbol{v}}{\boldsymbol{v}^\top\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top]\boldsymbol{v}}.
$$

Given those optimum we can thus obtain the alignment measure same as before as the sueprvised loss obtain from V^* from the unsupervised loss and $\boldsymbol{Z}^*$ from the supervised one:

$$
\boldsymbol{Z}^* = \arg\min_{\boldsymbol{Z}} \mathbb{E}_{\boldsymbol{X}'_n \sim p_{\boldsymbol{X}'|\boldsymbol{X}}} \|\boldsymbol{Z}^\top\boldsymbol{V}^{*\top}\boldsymbol{X}' - \boldsymbol{Y}\|_2^2,
$$

whose optimum is therefore given by $\boldsymbol{Z}^* = (\boldsymbol{V}^{*\top}\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top]\boldsymbol{V}^*)^{-1}\boldsymbol{V}^{*\top}\mathbb{E}[\boldsymbol{X}']\boldsymbol{Y}^\top$ which can then be plugged back to produce the (unnormalized) measure. The analytical form of can be obtained (following the derivations of (Balestriero et al., 2022)) as follows for example for the case of additive Gaussian noise which is trivial and commonly done before e.g. showing the link between ridge regression and additive dropout $\mathbb{E}[\boldsymbol{X}'] = \boldsymbol{X}$ and $\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top] = \boldsymbol{X}'\boldsymbol{X}'^\top + \sigma\boldsymbol{I}$. The perhaps more interesting derivations concern the masking as employed by MAE.

5.5. Proof of Corollary 3.1

In the case of additive, centered Gaussian noise, we have $\mathbb{E}[\boldsymbol{X}'] = \boldsymbol{X}$ and $\mathbb{E}[\boldsymbol{X}'\boldsymbol{X}'^\top] = \boldsymbol{X}'\boldsymbol{X}'^\top + \sigma\boldsymbol{I}$. Therefore the optimal value for $\boldsymbol{V}$, is given by solving the generalized eigenvalue problem $(\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{X}^\top, \boldsymbol{X}\boldsymbol{X}^\top + \sigma\boldsymbol{I})$. Recalling the derivations of the optimal solution for such problem from Section 5.1, we have that

$$\boldsymbol{V} = \boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top+\sigma\boldsymbol{I}}\boldsymbol{D}^{-\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top+\sigma\boldsymbol{I}}(\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K} = \boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{-\frac{1}{2}}(\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K},$$

with

$$\begin{aligned}\boldsymbol{H} =&\boldsymbol{D}^{-\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top+\sigma\boldsymbol{I}}\boldsymbol{P}^\top_{\boldsymbol{X}\boldsymbol{X}^\top+\sigma\boldsymbol{I}}(\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top+\sigma\boldsymbol{I}}\boldsymbol{D}^{-\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top+\sigma\boldsymbol{I}}\\ =&\boldsymbol{D}^{-\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top+\sigma\boldsymbol{I}}\boldsymbol{P}^\top_{\boldsymbol{X}\boldsymbol{X}^\top}(\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{D}^{-\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top+\sigma\boldsymbol{I}}\\ =&(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top})^2(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{-1},\end{aligned}$$

and the important property to notice is that the ordering of the eigenvalues of $\boldsymbol{H}$ which are given by $(\boldsymbol{D}^\top_{\boldsymbol{X}\boldsymbol{X}^\top})^2(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{-1}$ are the same as the ordering of $\boldsymbol{X}\boldsymbol{X}^\top$ which are given by $\boldsymbol{D}^\top_{\boldsymbol{X}\boldsymbol{X}^\top}$. That is, the top-$K$ subspace that will be picked up by $\boldsymbol{P}_{\boldsymbol{H}}$ are the same for any noise standard deviation σ. Now given the close form for $\boldsymbol{V}$ we can obtain the close form of the classifier weights $\boldsymbol{W}$ from Eq. (5) to be

$$\begin{aligned}\boldsymbol{W} =&(\boldsymbol{V}^{*\top}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{V}^*)^{-1}\boldsymbol{V}^{*\top}\boldsymbol{X}\boldsymbol{Y}^\top\\ \boldsymbol{W} =&\Big(((\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K})^\top(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{-\frac{1}{2}}\boldsymbol{P}^\top_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{P}_{\boldsymbol{X}\boldsymbol{X}^\top}(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{-\frac{1}{2}}(\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K}\Big)^{-1}\\ &\times((\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K})^\top(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{-\frac{1}{2}}\boldsymbol{P}^\top_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{X}\boldsymbol{Y}^\top\\ \boldsymbol{W} =&\big(((\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K})^\top(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{-1}\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top}(\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K}\big)^{-1}((\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K})^\top(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{-\frac{1}{2}}\boldsymbol{D}^{\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{V}_{\boldsymbol{X}}^\top\boldsymbol{Y}^\top\\ \boldsymbol{W} =&((\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K})^\top(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})\boldsymbol{D}^{-1}_{\boldsymbol{X}\boldsymbol{X}^\top}(\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K}((\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K})^\top(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{-\frac{1}{2}}\boldsymbol{D}^{\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top}\boldsymbol{V}_{\boldsymbol{X}}^\top\boldsymbol{Y}^\top\\ \boldsymbol{W} =&((\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K})^\top\left((\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{\frac{1}{2}}\boldsymbol{D}^{-\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top}\right)_s\boldsymbol{V}_{\boldsymbol{X}}^\top\boldsymbol{Y}^\top,\end{aligned}$$

where the s subscript indicates that only the top $K \times K$ part of the diagonal matrix is nonzero. Lastly, the product of both matrices (producing the supervised linear model) is obtained as

$$\begin{aligned}\boldsymbol{W}^\top\boldsymbol{V}^\top =&\boldsymbol{Y}\boldsymbol{V}_{\boldsymbol{X}}\left((\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{\frac{1}{2}}\boldsymbol{D}^{-\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top}\right)_s(\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K}((\boldsymbol{P}_{\boldsymbol{H}})_{.,1:K})^\top(\boldsymbol{D}_{\boldsymbol{X}\boldsymbol{X}^\top} + \sigma\boldsymbol{I})^{-\frac{1}{2}}\boldsymbol{P}^\top_{\boldsymbol{X}\boldsymbol{X}^\top}\\ =&\boldsymbol{Y}\boldsymbol{V}_{\boldsymbol{X}}(\boldsymbol{D}^{-\frac{1}{2}}_{\boldsymbol{X}\boldsymbol{X}^\top})_s\boldsymbol{P}^\top_{\boldsymbol{X}\boldsymbol{X}^\top}\end{aligned}$$

therefore recovering the OLS optimal solution $\boldsymbol{Y}\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{X}^\top)^{-1}$ whenever $K \geq D$, and otherwise recovers the projection onto the top subspace of $\boldsymbol{X}$–in any case the final parameters are invariant to the choice of the standard deviation of the additive Gaussian noise (σ) during the denoising autoencoder pre-training phase.

5.6. Proof of Theorem 2

Proof. We prove separately the two cases of linear and nonlinear decoder. The linear decoder setting relies on less assumptions about the data distributions but is also the less general statement. The nonlinear decoder setting is only valid near initialization as it assumes that the Jacobian matrix of the decoder is at random initialization, i.e., not yet aligned with the data manifold.

Linear decoder setting. The first part of the proof is to rewrite the joint objective classification and reconstruction objective with an arbitrary encoder network f_θ

$$\min_{\theta\in\mathbb{R}^P, \boldsymbol{W}\in\mathcal{M}_{C,K}(\mathbb{R}), \boldsymbol{V}\in\mathcal{M}_{D,K}(\mathbb{R})} \|\boldsymbol{W}f_\theta(\boldsymbol{X}) - \boldsymbol{Y}\|_F^2 + \lambda\|\boldsymbol{V}f_\theta(\boldsymbol{X}) - \boldsymbol{X}\|_F^2,$$

as the nonparametric version

$$\min_{\boldsymbol{Z}\in\mathcal{M}_{K,N}(\mathbb{R}), \boldsymbol{W}\in\mathcal{M}_{C,K}(\mathbb{R}), \boldsymbol{V}\in\mathcal{M}_{D,K}(\mathbb{R})} \|\boldsymbol{W}\boldsymbol{Z} - \boldsymbol{Y}\|_F^2 + \lambda\|\boldsymbol{V}\boldsymbol{Z} - \boldsymbol{X}\|_F^2,$$

both being identical if we assume that the encoder is powerful enough to reach any representation, which is a realistic assumption given current architectures. Given that nonparametric objective, we can now solve for both the optimal decoder weight $\boldsymbol{V}$ and the optimal representation $\boldsymbol{Z}$ as follows

$$
\begin{aligned}
&\min_{\boldsymbol{Z}\in\mathcal{M}_{K,N}(\mathbb{R}),\boldsymbol{W}\in\mathcal{M}_{C,K}(\mathbb{R}),\boldsymbol{V}\in\mathcal{M}_{D,K}(\mathbb{R})} \|\boldsymbol{W}\boldsymbol{Z}-\boldsymbol{Y}\|_F^2+\lambda\|\boldsymbol{V}\boldsymbol{Z}-\boldsymbol{X}\|_F^2\\
=&\min_{\boldsymbol{Z}\in\mathcal{M}_{K,N}(\mathbb{R})} \|\boldsymbol{Y}\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}-\boldsymbol{Y}\|_F^2+\lambda\|\boldsymbol{X}\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}-\boldsymbol{X}\|_F^2\\
=&\min_{\boldsymbol{Z}\in\mathcal{M}_{K,N}(\mathbb{R})} \operatorname{Tr}\left(\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\right)-2\operatorname{Tr}\left(\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\right)+\|\boldsymbol{Y}\|_F^2\\
&+\lambda\operatorname{Tr}\left(\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\right)-2\lambda\operatorname{Tr}\left(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\right)+\lambda\|\boldsymbol{X}\|_F^2\\
=&\min_{\boldsymbol{Z}\in\mathcal{M}_{K,N}(\mathbb{R})} \operatorname{Tr}\left((\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{Z}^\top\right)-2\operatorname{Tr}\left(\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\right)+\|\boldsymbol{Y}\|_F^2\\
&+\operatorname{Tr}\left((\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{Z}^\top\right)-2\lambda\operatorname{Tr}\left(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\right)+\lambda\|\boldsymbol{X}\|_F^2\\
=&\min_{\boldsymbol{Z}\in\mathcal{M}_{K,N}(\mathbb{R})} -\operatorname{Tr}\left(\boldsymbol{Y}^\top\boldsymbol{Y}\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\right)+\|\boldsymbol{Y}\|_F^2-\lambda\operatorname{Tr}\left(\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\right)+\lambda\|\boldsymbol{X}\|_F^2\\
=&\min_{\boldsymbol{Z}\in\mathcal{M}_{K,N}(\mathbb{R})} -\operatorname{Tr}\left(\left(\boldsymbol{Y}^\top\boldsymbol{Y}+\lambda\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{Z}^\top(\boldsymbol{Z}\boldsymbol{Z}^\top)^{-1}\boldsymbol{Z}\right)+\|\boldsymbol{Y}\|_F^2+\lambda\|\boldsymbol{X}\|_F^2\\
=&\min_{\boldsymbol{Z}\in\mathcal{M}_{K,N}(\mathbb{R}):\boldsymbol{Z}^\top\boldsymbol{Z}=\boldsymbol{I}} -\operatorname{Tr}\left(\left(\boldsymbol{Y}^\top\boldsymbol{Y}+\lambda\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{Z}^\top\boldsymbol{Z}\right)+\|\boldsymbol{Y}\|_F^2+\lambda\|\boldsymbol{X}\|_F^2
\end{aligned}
$$

which is solved by $\boldsymbol{Z}$ being any orthogonal matrix in the subspace of the top-K eigenvectors of $\left(\boldsymbol{Y}^\top\boldsymbol{Y}+\lambda\boldsymbol{X}^\top\boldsymbol{X}\right)$. Now as $\lambda\to\infty$ as the encoder will become more and more linear, ultimately converging to $f_\theta(\boldsymbol{x})=\boldsymbol{U}\boldsymbol{x}$ with $\boldsymbol{U}\in\operatorname{span}\{\operatorname{eigvec}(\boldsymbol{X}^\top\boldsymbol{X})_1,\ldots,\operatorname{eigvec}(\boldsymbol{X}^\top\boldsymbol{X})_K\}$

Nonlinear decoder setting. To prove the result in the nonlinear decoder setting, our argument will look at the contribution of the data principal subspace to the gradient norm used to learn the encoder network parameters. In short, we will see that most of the gradient energy comes from the principal subspace of the data, effectively forcing the autoencoder model to focus on that subspace first. Let's first derive the gradient of the loss with respect to the encoder output

$$
\begin{aligned}
\nabla_{\mathcal{L}}(f(\boldsymbol{x}_n)) &= \sum_{n=1}^{N}\boldsymbol{J}_g(f(\boldsymbol{x}_n))^\top(g(f(\boldsymbol{x}_n))-\boldsymbol{x}_n)\\
&=\sum_{n=1}^{N}\boldsymbol{J}_g(f(\boldsymbol{x}_n))^\top g(f(\boldsymbol{x}_n))-\sum_{n=1}^{N}\boldsymbol{J}_g(f(\boldsymbol{x}_n))^\top\boldsymbol{x}_n\\
&=\sum_{n=1}^{N}\boldsymbol{J}_g(f(\boldsymbol{x}_n))^\top g(f(\boldsymbol{x}_n))-\sum_{n=1}^{N}\sum_{k=1}^{K}\langle\boldsymbol{x}_n,\boldsymbol{v}_k(\boldsymbol{X})\rangle\boldsymbol{J}_g(f(\boldsymbol{x}_n))^\top\boldsymbol{v}_k(\boldsymbol{X})\\
&=\sum_{n=1}^{N}\boldsymbol{J}_g(f(\boldsymbol{x}_n))^\top g(f(\boldsymbol{x}_n))-\sum_{k=1}^{K}\sigma_k(\boldsymbol{X})\left(\sum_{n=1}^{N}\boldsymbol{J}_g(f(\boldsymbol{x}_n))^\top\boldsymbol{U}_{n,k}(\boldsymbol{X})\right)\boldsymbol{v}_k(\boldsymbol{X}),
\end{aligned}
$$

where we recall that $\boldsymbol{v}_k(\boldsymbol{X})$ is a unit-norm vector (right singular vector of $\boldsymbol{X}$), and we have singular values in descending order, $\sigma_i\geq\sigma_i,\forall i\leq j$. As a result, we first notice that the gradient will involve the sum of backpropagated right singular vector of $\boldsymbol{X}$ weighted by the corresponding singular values. Since we know that

$$
\sigma_{\min}^2\left(\sum_{n=1}^{N}\boldsymbol{J}_g(f(\boldsymbol{x}_n))\boldsymbol{U}_{n,k}(\boldsymbol{X})\right)\leq\left\|\left(\sum_{n=1}^{N}\boldsymbol{J}_g(f(\boldsymbol{x}_n))^\top\boldsymbol{U}_{n,k}(\boldsymbol{X})\right)\boldsymbol{v}_k(\boldsymbol{X})\right\|_2^2\leq\sigma_{\max}^2\left(\sum_{n=1}^{N}\boldsymbol{J}_g(f(\boldsymbol{x}_n))\boldsymbol{U}_{n,k}(\boldsymbol{X})\right)
$$

we can guarantee that the amount of information coming from the first principal singular vector of $\boldsymbol{X}$ is greater than the information coming form the last singular vector as soon as

$$
\frac{\sigma_1^2(\boldsymbol{X})}{\sigma_K^2(\boldsymbol{X})}\geq\frac{\sigma_{\max}^2\left(\sum_{n=1}^{N}\boldsymbol{J}_g(f(\boldsymbol{x}_n))\boldsymbol{U}_{n,K}(\boldsymbol{X})\right)}{\sigma_{\min}^2\left(\sum_{n=1}^{N}\boldsymbol{J}_g(f(\boldsymbol{x}_n))\boldsymbol{U}_{n,1}(\boldsymbol{X})\right)}.
$$

Given that $\boldsymbol{U}$ is a unitary matrix, and that (at initialization) $\boldsymbol{J}_g(f(\boldsymbol{x}_n))$ is well-behaved, we conclude that the above is always true at initialization in most practical settings where $\sigma_1^2(\boldsymbol{X}) \gg \sigma_K^2(\boldsymbol{X})$. Lastly, the fact that the principal subspace of $\boldsymbol{X}$ impacts the gradient of the loss with respect to $f(\boldsymbol{x})$ is sufficient since training is done through backpropagation. Hence, any bias that $\nabla_{\mathcal{L}}(f(\boldsymbol{x}_n))$ has towards the principal subspace of the data will naturally flow through the backpropagation steps to all the internal parameters of f.

□

5.7. Eigendecomposition

Given a matrix $\boldsymbol{X} \in \mathcal{M}_{D,N}(\mathbb{R})$ with $D > N$, computing the eigendecomposition of $\boldsymbol{X}\boldsymbol{X}^\top$, a $D \times D$ matrix is $\mathcal{O}(D^3)$ which instead can be obtained in $\mathcal{O}(N^3 + DN^2)$ as

```
def fast_gram_eigh(X, major="C", unit_test=False):
    """
    compute the eigendecomposition of the Gram matrix:
    - XX.T using column (C) major notation
    - X.T@X using row (R) major notation
    """
    if major == "C":
        X_view = X.T
    else:
        X_view = X

    if X_view.shape[1] < X_view.shape[0]:
        # this case is the usual formula
        U, S = np.linalg.eigh(X_view.T @ X_view)
    else:
        # in this case we work in the tranpose domain
        U, S = np.linalg.eigh(X_view @ X_view.T)
        S = X_view.T @ S
        S[U>0] /= np.sqrt(U[U>0])
        # ensuring that we have the correct values
        if unit_test:
            Uslow, Sslow = np.linalg.eigh(X_view.T @ X_vew)
            assert np.allclose(U, Uslow)
            assert np.allclose(S, Sslow)
    return U, S
```

since we have the relation

$$\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{v} = \lambda \boldsymbol{v} \iff \boldsymbol{X}^\top \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{v}) = \frac{\lambda}{\|\boldsymbol{X}^\top \boldsymbol{v}\|_2^2}(\boldsymbol{X}^\top \boldsymbol{v}),$$

and thus we can simply compute the eigenvectors of the $K \times K$ matrix $\boldsymbol{X}^\top \boldsymbol{X}$ and get the eigenvectors of the $N \times N$ matrix $\boldsymbol{X}\boldsymbol{X}^\top$ by left-multiplying them by $\boldsymbol{X}^\top$, and their corresponding eigenvalues are rescaled by $\frac{1}{\|\boldsymbol{X}^\top \boldsymbol{v}\|_2^2}$.

5.8. Fast Implementation of Alignment Metric (Eq. (7))

We want to sweep over the latent dimension K. As such, we can avoid recomputing the metric for each value and get them all at once as below. We again use the column-major notations as per Section 2:

```
def alignment_sweep(X, Y, major="C"):
    U, S, Vh = np.linalg.svd(X, full_matrices=False)
    if major == "C":
        denom = np.square(np.linalg.norm(Y @ Y.T))
        numer = np.linalg.multi_dot([Y.T, Y, Vh.T])
    else:
        denom = np.square(np.linalg.norm(Y.T @ Y))
        numer = np.linalg.multi_dot([Y,Y.T, U])
    numer = np.linalg.norm(numer, axis=0)**2
    return np.cumsum(numer) / denom
```

5.9. Additional figures

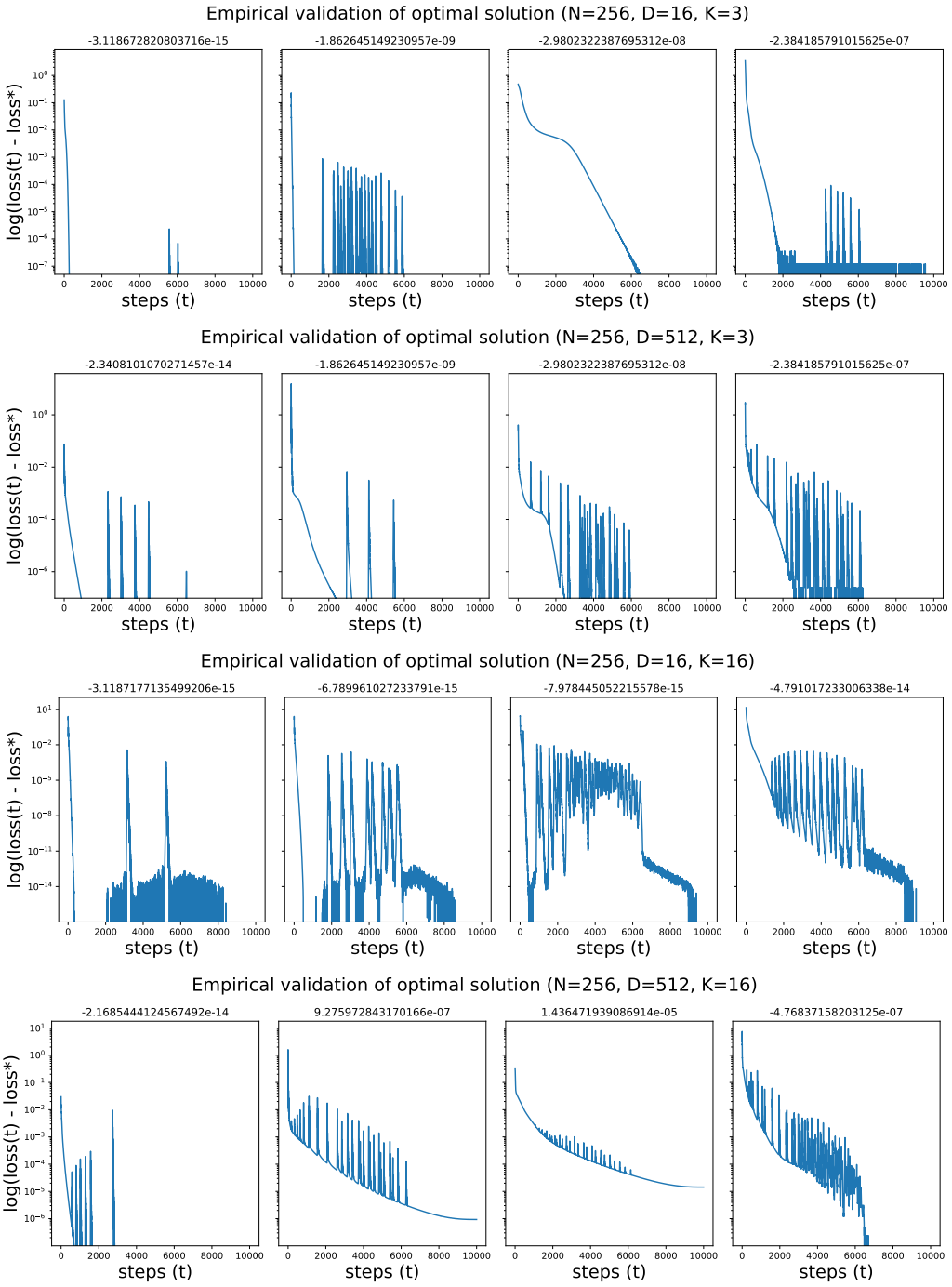

Figure 9. Empirical validation of Theorem 1 comparing the loss value at the optimum (from Eqs. (4) to (6)) against the one minimized with gradient descent (Adam optimizer) (**y-axis**) during gradient steps (t, **x-axis**). We expect that different to get close to 0 as the gradient updates converge to the minimum value of the loss. Although that quantity (loss(optimum) - loss(t)) is nonnegative in theory, we observe that its minimum value (reported in the title of each subplot) is sometimes negative with negligible value due to round off error. We compare numerous values of K, D, N as given in the titles of each **row**, and different values of $\lambda \in \{0.0, 0.1, 1, 10\}$ (**column**).

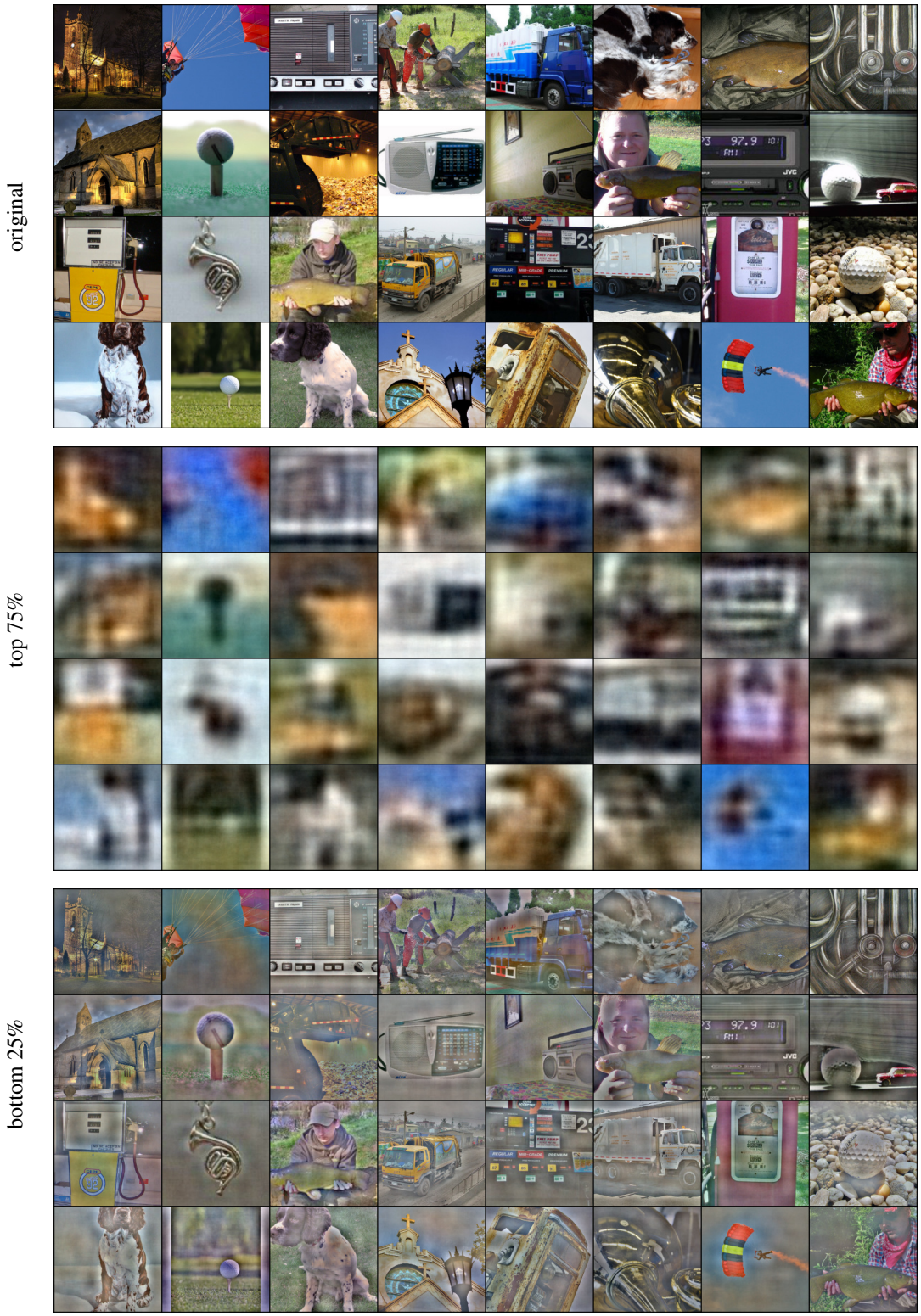

Figure 10. Depiction of Imagenet images (**top**) projected onto different subspaces obtained from Principal Component Analysis corresponding to the subspace explaining the top 75% of pixel variance (**middle**) and bottom 25% of pixel variance (**bottom**). We clearly observe that the image representation preserved after projection onto the bottom subspace makes the perception tasks (classification) easier to solve that if projected onto the top subspace, where the lower frequency information is insufficient to classify what is the object depicted (recall the classification performances of DNs applied onto those different projections from Figs. 1 and 4).