Improving OOD Generalization of Pre-trained Encoders via Aligned Embedding-Space Ensembles

Shuman Peng, Arash Khoeini, Sharan Vaswani, Martin Ester
School of Computing Science
Simon Fraser University
{shumanp, akhoeini}@sfu.ca vaswani.sharan@gmail.com ester@cs.sfu.ca

Abstract

The quality of self-supervised pre-trained embeddings on out-of-distribution (OOD) data is poor without fine-tuning. A straightforward and simple approach to improving the generalization of pre-trained representation to OOD data is the use of deep ensembles. However, obtaining an effective ensemble in the embedding space with only unlabeled data remains an unsolved problem. We first perform a theoretical analysis that reveals the relationship between individual hyperspherical embedding spaces in an ensemble. We then design a principled method to align these embedding spaces in an unsupervised manner. Experimental results on the MNIST dataset show that our embedding-space ensemble method improves pre-trained embedding quality on in-distribution and OOD data compared to single encoders.

1 Introduction

Self-supervised learning techniques enable the pre-training of deep neural network (DNN) encoders on widely available unlabeled data. These pre-trained encoders, once fine-tuned, are highly transferable to various downstream tasks [5, 3]. However, a key challenge remains: without fine-tuning, the quality of pre-trained features is noticeably lower on out-of-distribution (OOD) data, impairing the performance of pre-trained models on subsequent OOD downstream tasks [7]. This issue is particularly critical when the downstream task has insufficient data for fine-tuning, making it essential for the pre-trained encoders to generalize well to OOD data in a zero-shot manner without fine-tuning.

A straightforward approach to improve the generalizability of pre-trained representation quality to OOD data is the use of deep ensembles. Deep ensembles (DEs) [9], consist of DNNs independently trained with different initializations and data orders (e.g. seeds). DEs have been shown to improve the predictive performance over single DNNs on both in-distribution (ID) and OOD data [14, 1].

Existing DE approaches typically aggregate models either in the predictive output space [1, 9, 14] (e.g., logits for classification) or in the weight space [21, 15, 16]. Aggregating in the predictive space confines the ensemble to a single task, e.g., classification with a fixed set of classes, and is inapplicable for self-supervised pre-trained encoders. In contrast, aggregating models in the weight space offers the flexibility to accommodate various tasks by discarding the predictive layer and retaining only the ensembled encoder. However, this approach sacrifices the interpretability that predictive-space ensembles provide, particularly in how individual model outputs are combined and unified. In essence, existing DE techniques exhibit an undesirable trade-off between interpretability and flexibility.

In this paper, we take a novel perspective to ensemble self-supervised pre-trained encoders for improved zero-shot OOD generalization. Our approach offers both the flexibility of weight-space ensembling and the interpretability of predictive-space ensembling. We propose an embedding-space ensemble, called `Ensemble-InfoNCE`, in which we aggregate the ensemble mean in the

NeurIPS 2024 Workshop: Self-Supervised Learning - Theory and Practice.

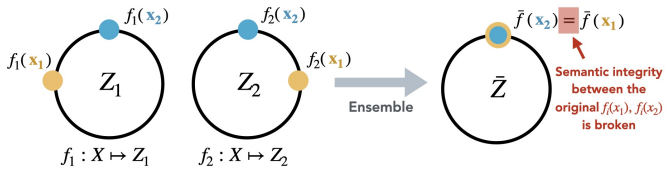


Figure 1: Need for embedding alignment: The ensemble mean of two different embeddings (yellow, blue) in misaligned embedding spaces Z_1, Z_2 collapses to the same vector in $\bar{Z}$, although they have different semantic meanings.

hyperspherical latent embedding space of encoders pre-trained using the widely used InfoNCE contrastive loss [13, 3].

Obtaining the ensemble mean of embedding vectors is less straightforward compared to taking the mean of predictive outputs or model weights. To take the ensemble mean of embeddings, the embedding spaces must be aligned such that the embeddings (produced by the different encoders) corresponding to semantically similar samples have a similar direction in the hyperspherical space. Taking the mean of misaligned embeddings that point in different directions can harm the semantic integrity of the embedding space (Figure 1). Existing approaches align embeddings using class labels [22]. However, aligning embeddings without access to labels remains an unsolved problem. To this end, we extend the theoretical results in [23], and use this to propose a principled unsupervised approach (referred to as `Ensemble-InfoNCE`) to align the embeddings. Our theoretical results demonstrate that an ensemble of encoders with aligned embedding spaces recovers the correct (ground truth) embeddings. Finally, we experimentally show improved pre-trained embedding quality on in-distribution (ID) and OOD data for `Ensemble-InfoNCE` compared to single InfoNCE encoders on the MNIST dataset.

2 Background and preliminaries

InfoNCE loss. We consider encoders pre-trained using the InfoNCE loss, a widely used contrastive (self-supervised) learning objective [13, 2, 3, 5, 18, 20]. These encoders f map the input space $\mathcal{X}$ to an L_2-normalized unit-hyperspherical embedding space $\mathcal{Z} = \mathbb{S}^{D-1}$. Encoders trained with the InfoNCE loss (3) have the desirable property of mapping semantically similar pairs of samples close together in the embedding space, while also ensuring that dissimilar samples are mapped far apart.

InfoNCE-trained encoders recover the correct latents. Zimmermann et al. [23] theoretically demonstrate that an encoder f minizing the contrastive loss in (3) recovers the ground truth latents $z \in \mathcal{Z}$ up to orthogonal transformations. Specifically, $z_1^\top z_2 = h(z_1)^\top h(z_2)$, where z_1, z_2 are ground truth latents, and $h = f \circ g$ composes of encoder f and a generative process $g : \mathcal{Z} \mapsto \mathcal{X}$.[1]

3 Embedding-space ensembles via unsupervised alignment

In this section, we introduce, to the best of our knowledge, the first approach of ensembling self-supervised pre-trained encoders in the embedding space, which offers the interpretability of predictive-space ensembles and the flexibility of weight-space ensembles. Our embedding-space ensemble approach, referred to as `Ensemble-InfoNCE`, produces mean embedding vectors $\bar{z} = \bar{f}(x) \in \mathcal{Z}$ for a given ensemble of $M > 1$ encoders $\{f_i : \mathcal{X} \mapsto \mathcal{Z}_i\}_{i=1}^M$ pre-trained using the InfoNCE loss (3). Each $\mathcal{Z}_i$ is a unit-hyperspherical embedding space $\mathcal{Z}_i = \mathbb{S}^{D-1}$. Before taking the ensemble mean, we must first *align* the embedding spaces $\{\mathcal{Z}_i\}_{i=1}^M$ so that each of the M embeddings $\{f_i(x)\}_{i=1}^M$ for the same input x points in a similar direction on a hypersphere. However, aligning embedding spaces without labels in an unsupervised manner remains a challenging problem.

To tackle the challenge of performing unsupervised embedding space alignment, we first conduct a theoretical analysis and reveal a critical orthogonality relationship between different embedding spaces (Section 3.1). This relationship allows us to extend the guarantees on the recovery of the correct latents[2] from single encoders to an ensemble. Furthermore, this relationship enables us to align the embedding spaces by recovering the orthogonal transformation matrix and forms the basis of our unsupervised embedding space alignment approach (Section 3.2). Finally, the aligned ensemble embeddings are aggregated using the Karcher Mean algorithm [17] (Section 3.3).

[1] We refer interested readers to Theorem 2 of [23].

[2] In this paper, we use the terms "latents", "features", and "embeddings" interchangeably.

3.1 Theoretical analysis

An encoder's latent space is an orthogonal transformation of another encoder's. Extending the main theoretical result of [23] (Section 2) from single encoders to an ensemble of these encoders, we reveal that an encoder's embedding space is an orthogonal transformation of another encoder's. This is formally stated below in Proposition 1 with the corresponding proof in Appendix A.2.

Assumption. For an ensemble of encoders f_1, f_2 trained on the same data $D = \{x_i\}_{i=1}^N$ with different random seeds, we *assume* that $f_1(x) = R_1 z$ and $f_2(x) = R_2 z$. In other words, both f_1 and f_2 recover the correct (ground truth) latents z up to different orthogonal transformations R_1, R_2.[3]

Proposition 1 (Orthogonal transformation relationship)**.** *Under the above assumption, f_1 and f_2 learn the same latents up to an orthogonal transformation R, that is, $f_1(x) = Rf_2(x)$.*

An ensemble of aligned embeddings recovers the correct latents. Based on Proposition 1, in Proposition 2 we generalize the theoretical guarantee on the correctness of the learned latents for each ensemble member to the ensemble as a whole. The proof is provided in Appendix A.2.

Proposition 2 (Ensemble recovers correct latents)**.** *The ensemble mean $\bar{f}(x)$ of aligned embeddings $f_1(x)$ and $Rf_2(x)$ are the correct latents z up to orthogonal transformation R_1, that is, $\bar{f}(x) = R_1 z$.*

Approximate orthogonal relationship on real-world data. With real-world data, there may not exist an orthogonal transformation that perfectly aligns *all* corresponding embedding vectors of the same input between different embedding spaces. This discrepancy arises due to violations of the data generation and modeling assumptions. Under such violated conditions, Zimmermann et al. [23] demonstrated that these encoders still recover the true latents to a moderate to high degree. By extending this result to our problem setting, we infer that the relationship between encoders f_i and f_j can be approximated as $f_i(x) \approx Rf_j(x)$ for all $x \in \mathcal{X}$. In deep ensembles, this approximate orthogonal transformation relationship preserves some degree of diversity between embedding spaces, preventing the ensemble from collapsing into a single model.

3.2 Unsupervised embedding space alignment via learning orthogonal matrices

Our goal is to align the M hyperspherical embedding spaces $Z_1, ..., Z_M$ induced by encoders $\{f_i\}_{i=1}^M$ so that the same in-distribution pre-training sample is mapped to embeddings that have a similar direction across $\{Z_i\}_{i=1}^M$. From Section 3.1, we know that $f_i(x) \approx Rf_j(x)$ for all $x \in \mathcal{X}$, which means that learning the orthogonal transformation matrix R would naturally align $f_j(x)$ with $f_i(x)$. We note that R does not need to be strictly orthogonal for aligning the embedding spaces. To learn R in a D-dimensional embedding space, we use a single-layer neural network with D input and output nodes. The $D \times D$ weight matrix within this single layer neural network represents the orthogonal transformation matrix R that we want to learn.

To align the embedding spaces, we randomly select one **anchor** encoder from the set of M encoders to align the remaining $M - 1$ encoders. To align each Z_j with the anchor embedding space Z_i, we propose an objective function that enforces R to be as close to orthogonal as possible by imposing orthogonality as a soft constraint, while simultaneously maximizing the alignment between pairs of embeddings:

$$\mathcal{L}_{\text{align}} = \underset{R \in \mathbb{R}^{D \times D}}{\arg\min} \frac{1}{N} \sum_{n=1}^{N} d(f_i(x_n), Rf_j(x_n)) + \lambda \|R^T R - I_D\|_F^2 \tag{1}$$

where N denotes the number of samples, λ is a hyperparameter that controls the strength of the orthogonality constraint, and $d(.,.)$ is a function that quantifies the discrepancy between pairs of vectors. Given that the embedding vectors reside on the surface of a unit sphere, we use the geodesic distance, which measures the shortest path between two points on a Riemannian manifold and accounts for the spherical geometry. The objective function then becomes:

$$\mathcal{L}_{\text{align}} = \underset{R \in \mathbb{R}^{D \times D}}{\arg\min} \frac{1}{N} \sum_{n=1}^{N} \arccos(f_i(x_n), Rf_j(x_n)) + \lambda \|R^T R - I_D\|_F^2 \tag{2}$$

[3] For our theoretical analysis, we consider the case of an ensemble of $M = 2$ encoders for simplicity. The results can be extended to $M > 2$ encoders.

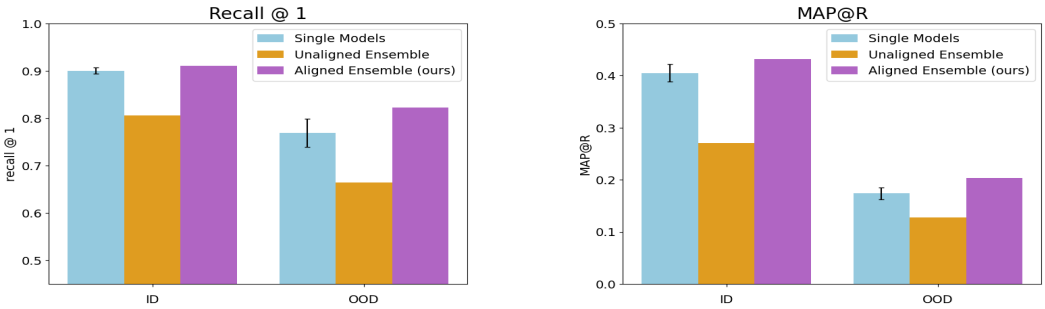


Figure 2: Comparing embedding qualities of single models (blue), an ensemble of unaligned embedding spaces (orange), and an ensemble of aligned embedding spaces (purple) in the ID and OOD settings. Recall@1 and MAP@R are presented. Higher values indicate better performance. The mean and standard deviation (error bars) of the performance metrics are reported for the 5 single models. The ensembles do not have standard deviation since all 5 models are combined into one.

3.3 Embedding mean in an ensemble

An ensemble of M pre-trained encoders with aligned embedding spaces produces a mean embedding vector for each sample, i.e., $\bar{z} = \bar{f}(x) = \text{mean}(f_1(x), R_2 f_2(x), ..., R_M f_M(x))$. Given the hyperspherical nature of the embedding spaces, we apply the Karcher Mean algorithm from [17] to compute meaningful mean vectors on the surface of a sphere. The algorithm projects hyperspherical data points onto a linear tangent space, calculates the mean in this tangent space, and then projects the result back onto the sphere. This process iterates until the mean in the tangent space approaches a near-zero norm, indicating that the mean has converged.[4]

4 Experiments

Dataset and training setup We use the MNIST dataset [10]. For ID evaluation, the test set is used as is. For OOD evaluation, each test sample is randomly colored. Since random coloring was not applied during pre-training, colored versions of the images are considered OOD. A total of $M = 5$ encoders are trained for the ensemble, as this ensemble size has been shown to be sufficient to produce good results [14]. Details of the dataset, model architecture and training are provided in Appendix A.3 and A.4.

Evaluation metrics In line with the representation learning literature, we assess the quality of embeddings using the **Recall at 1** (R@1) and **Mean Average Precision at R** (MAP@R) metrics [7, 11]. R@1 measures the semantic quality of the embedding space by verifying if each embedding's nearest neighbor belongs to the same class. MAP@R evaluates the proportion of each embedding's R nearest neighbors that belong to the same class, while accounting for the ranking of correct retrievals [11]. R is set to the total number of samples in a class [11].

4.1 Experimental results

Figure 2 shows that our `Ensemble-InfoNCE` model with aligned embedding spaces (shown in purple) improves the quality of embeddings over single models and unaligned ensembles in both the in-distribution (ID) and out-of-distribution (OOD) settings. The embedding quality improvement achieved by our method is more pronounced in the OOD setting, with a 6.99% improvement over the mean R@1 of the $M = 5$ single models and a 17.38% improvement over the mean MAP@R of single models. Our results also highlight the importance of aligned embedding spaces for ensembles. Taking an ensemble of misaligned embedding spaces consistently hurts the embedding quality, achieving lower values of R@1 and MAP@R than a single model for both ID and OOD settings. Table 1 provides the numerical values for Figure 2, and additional results are provided in Appendix A.5.

5 Conclusion

We improved the generalizability of pre-trained encoders to OOD data by taking an ensemble of encoders in the embedding space. We constructed embedding-space ensembles by effectively aligning the individual embedding spaces. Preliminary experiments on the MNIST dataset demonstrate that our aligned embedding-space ensemble significantly enhances the OOD embedding quality compared to individual models. In the future, we will focus on scaling our method to larger datasets, such as ImageNet, and incorporating OOD data from real datasets [4, 8, 19, 12].

[4]Details of the Karcher Mean algorithm can be found in [17].

Acknowledgments and Disclosure of Funding

This research was supported by the NSERC Discovery Grant. We would like to thank Dr. Ke Li for his early feedback. We are grateful to Shichong Peng for providing insightful discussions and thorough feedback throughout the course of this research. We also appreciate the anonymous reviewers for their constructive feedback and suggestions.

References

[1] Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *ArXiv*, abs/2110.10832, 2021. URL `https://api.semanticscholar.org/CorpusID:239049452`.

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

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

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

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

[6] Michael Kirchhof, Enkelejda Kasneci, and Seong Joon Oh. Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs. In *International Conference on Machine Learning*, pages 17085–17104. PMLR, 2023.

[7] Michael Kirchhof, Bálint Mucsányi, Seong Joon Oh, and Dr Enkelejda Kasneci. Url: A representation learning benchmark for transferable uncertainty estimates. *Advances in Neural Information Processing Systems*, 36:13956–13980, 2023.

[8] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[9] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[10] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[11] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 681–699. Springer, 2020.

[12] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012, 2016.

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

[14] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

[15] Alexandre Rame, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, Patrick Gallinari, and Matthieu Cord. Diverse weight averaging for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35:10821–10836, 2022.

[16] Alexandre Ramé, Kartik Ahuja, Jianyu Zhang, Matthieu Cord, Léon Bottou, and David Lopez-Paz. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *International Conference on Machine Learning*, pages 28656–28679. PMLR, 2023.

[17] Julian Straub, Jason Chang, Oren Freifeld, and John Fisher III. A dirichlet process mixture model for spherical data. In *Artificial Intelligence and Statistics*, pages 930–938. PMLR, 2015.

[18] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020.

[19] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[20] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR, 2020.

[21] Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7949–7961, 2021. URL `https://api.semanticscholar.org/CorpusID:237420687`.

[22] Jianqing Xu, Shen Li, Ailin Deng, Miao Xiong, Jiaying Wu, Jiaxiang Wu, Shouhong Ding, and Bryan Hooi. Probabilistic knowledge distillation of face ensembles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3489–3498, 2023.

[23] Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pages 12979–12990. PMLR, 2021.

A Appendix / supplemental material

A.1 The InfoNCE contrastive loss

The InfoNCE loss is defined as

$$L_{\text{contr}}(f;\tau,M) \quad := \underset{\substack{(x,\,x^+)\sim p_{\text{pos}} \\ \{x_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\text{data}}}}{\mathbb{E}} \left[-\log \frac{e^{f(x)^\top f(x^+)/\tau}}{e^{f(x)^\top f(x^+)/\tau} + \sum_{i=1}^{M} e^{f(x)^\top f(x_i^-)/\tau}} \right], \tag{3}$$

where pairs (x, x^+) are drawn of the distribution of positive samples p_{pos}, and $M > 0$ negative samples are drawn from the distribution of all observations p_{data}.

A.2 Proofs for Section 3.1

Proposition 1 *Under the above assumption, f_1 and f_2 learn the same latents up to an orthogonal transformation R, that is, $f_1(x) = Rf_2(x)$.*

Proof. Let $f_1(x) = R_1 z$ and $f_2(x) = R_2 z$ where R_1, R_2 are orthogonal matrices, i.e., $R_1^T = R_1^{-1}, R_2^T = R_2^{-1}$, we have

$$\begin{aligned}
&\Rightarrow R_1^{-1} f_1(x) = z \quad \text{and} \quad R_2^{-1} f_2(x) = z \\
&\Rightarrow R_1^{-1} f_1(x) = R_2^{-1} f_2(x) \\
&\Rightarrow R_1 R_1^{-1} f_1(x) = R_1 R_2^{-1} f_2(x) \\
&\Rightarrow f_1(x) = R f_2(x) \quad (\text{Letting } R = R_1 R_2^{-1})
\end{aligned}$$

Since R_1 and R_2^{-1} are both orthogonal matrices, their product R is also an orthogonal matrix, i.e., $R^T = R^{-1}$. Therefore, f_1 and f_2 learns the same latents up to an orthogonal transformation R. □

Proposition 2 (The ensemble also recovers the correct latents)**.** *The ensemble mean $\bar{f}(x)$ of aligned embeddings $f_1(x)$ and $Rf_2(x)$ are the correct latents z up to orthogonal rotation R_1, that is, $\bar{f}(x) = R_1 z$.*

Proof. Let us denote the ensemble mean as $\bar{f}(x) = \text{mean}(f_1(x), Rf_2(x))$, where mean(.) is a general notion of the mean, which can be the arithmetic mean in Euclidean spaces or the Karcher Mean in Reimmanian manifolds. For the simplicity of this proof, we will use the arithmetic mean, but the results also apply to the Karcher Mean.

Defining $\bar{f}(x)$ using the arithmetic mean, we have:

$$\bar{f}(x) = \frac{1}{2}[f_1(x) + Rf_2(x)] \tag{4}$$

Since $f_1(x) = Rf_2(x)$, we have:

$$\bar{f}(x) = \frac{1}{2}[f_1(x) + f_1(x)] = f_1(x) = R_1 z \tag{5}$$

□

A.3 Additional implementation and training details

Contrastive pre-training architecture For contrastive pre-trained encoders, two convolution blocks with max-pooling and ReLU activations are used, with a linear layer attached at the end to project the embeddings down to $D = 8$ dimensions. The first convolution block consists of (1) a Conv2d with 3 input channels, 16 output channels, a kernel size of 5, stride of 1, and padding of

2; (2) max-pooling with kernel size 2; (3) a ReLU activation; and finally (4) a dropout layer with dropout rate p (we used $p = 0$ in our experiments). Similarly, the second convolution block consists of identical components, except with a Conv2d that consists of 16 input channels and 32 output channels.

Supervised contrastive pre-training For part of our experiments (Figures 4 and 5), we follow [6] and **use class labels to generate positive and negative pairs**, which better preserves the theoretical assumptions and guarantees for encoders trained with the InfoNCE loss [23]. Positive pairs consist of samples from the same class, while negative pairs are from different classes. We refer to this approach as *supervised contrastive pre-training*. Each encoder is contrastive trained for 2000 epochs using different random seeds $(10, 11, 12, 13, 14)$ and weight initializations. A batch size of 128 is used, and each sample is paired with 16 negative samples, following [6]. The learning rate is set to 0.01 and the AdamW optimizer is used.

Unsupervised contrastive pre-training For unsupervised contrastive pre-training, which is the conventional contrastive pre-training approach (also used for Figures 2 and 6 and Table 1), we **apply random rotations (± 30 degrees) to generate positive pairs**. Positive pairs are created by taking two randomly rotated views of the same sample, while negative pairs are formed from randomly rotated views of different samples. We also experimented with using random cropping to generate positive pairs, but found that this resulted in unstable model training. This instability is likely due to the nature of the MNIST images, which consist of white digits on a black background. Images that are cropped to include too much background and insufficient detail of the digit can result in distinct samples being mistakenly identified as similar. Each encoder is trained for 20 epochs using different random seeds $(10, 11, 12, 13, 14)$ and weight initializations. We use a batch size of 1024 and a learning rate of 0.1 with the LAMB optimizer, which is better suited for larger batch sizes.

Unsupervised embedding space alignment For unsupervised embedding space alignment, a linear layer with D input and output dimensions is used. To align the **supervised contrastive pre-trained encoders**, the linear alignment layer layer is trained for 20 epochs with a learning rate of 0.1, and an orthogonality regularization factor $\lambda = 0.5$ is applied. To align the **unsupervised contrastive pre-trained encoders**, which exhibit lower degrees of orthogonality compared to the supervised ones (as expected due to the violation of theoretical assumptions regarding the conditional distribution used to generate positive pairs [23]), the linear layer is trained for 20 epochs with a learning rate of 0.1 and $\lambda \in \{0.1, 0.3, 0.5\}$ was applied. A lower λ relaxes the orthogonality constraint for encoders that have slightly weaker orthogonal relationships. In both cases, the linear layer weights were optimized using stochastic gradient descent (SGD).

Computing resources We used a single RTX 3090 GPU for our experiments. For MNIST scale experiments, any GPU with more than 8GB of VRAM would be sufficient.

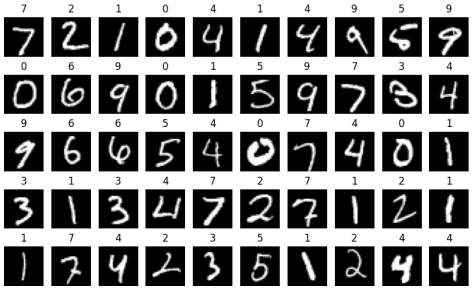
A.4 Additional dataset details

We converted single-channel grayscale MNIST images to three-channel black-and-white images. The training set is used to perform contrastive pre-training of the encoders and to align the embedding spaces. The test set is used for ID and OOD evaluation of the pre-trained encoders.

A.4.1 Data for OOD evaluation

Colored version We randomly colored the images in the MNIST test set to create an OOD evaluation set. Since only black-and-white images were used during pre-training, colored versions of the images are considered OOD compared to the original images. Colored versions of the images are illustrated in Figure 3b.

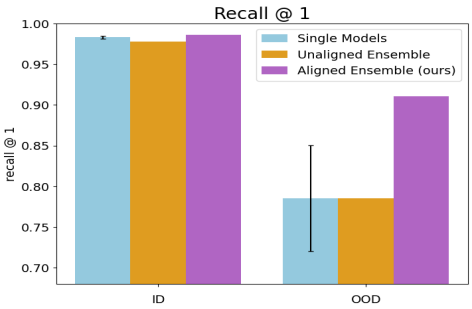
Cropped version For further OOD evaluation, each test sample is randomly cropped to `crop_size` $\sim \text{Unif}([0.25, 1])$ percent of their original size, following [6, 7]. Since no random cropping was applied during pre-training, cropped versions of the images are considered OOD compared to the original images.



(a) Original (ID) MNIST

(b) Colored (OOD) MNIST

Figure 3: For in-distribution (ID) evaluation, images like those in (a) were used. For out-of-distribution (OOD) evaluation, images like those in (b) were used.



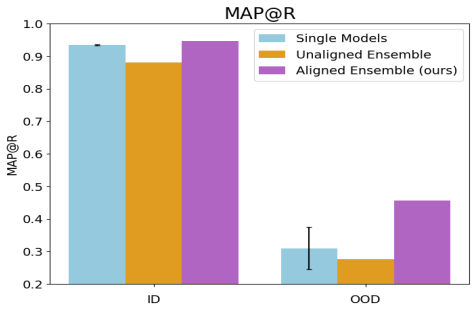


Figure 4: Supervised contrastive pre-training with Colored MNIST as OOD evaluation data. Comparing embedding qualities of single models (blue), an ensemble of unaligned embedding spaces (orange), and an ensemble of aligned embedding spaces (purple) in the ID and OOD settings. Recall@1 and MAP@R are presented. Higher values indicate better performance. The mean and standard deviation (error bars) of the performance metrics are reported for the 5 single models. The ensembles do not have standard deviation since all 5 models are combined into one.

A.5 Additional results

Supervised contrastive pre-training with colored OOD images Figure 4 compares the embedding qualities of single models and ensembles of aligned and unaligned embedding spaces. The $M = 5$ models are trained using the supervised contrastive pre-training procedure discussed in Appendix A.3. OOD evaluation is performed on Colored MNIST images (Appendix A.4.1).

Supervised contrastive pre-training with cropped OOD images Figure 5 compares the embedding qualities of single models and ensembles of aligned and unaligned embedding spaces. The $M = 5$ models are trained using the supervised contrastive pre-training procedure discussed in Appendix A.3. Cropped MNIST (A.4.1) images are used as OOD evaluation data.

Unsupervised contrastive pre-training with cropped OOD images Figure 6 compares the embedding qualities of single models and ensembles of aligned and unaligned embedding spaces. The $M = 5$ models are trained using the unsupervised contrastive pre-training procedure discussed in A.3. Cropped MNIST (A.4.1) images are used as OOD evaluation data.

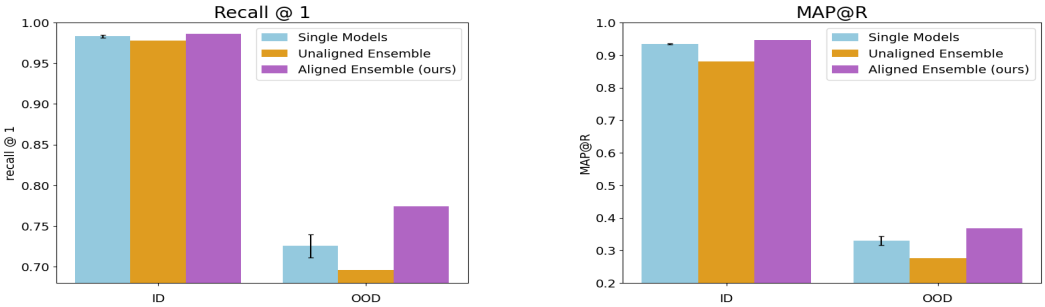


Figure 5: Supervised contrastive pre-training with Cropped MNIST as OOD evaluation data. Comparing embedding qualities of single models (blue), an ensemble of unaligned embedding spaces (orange), and an ensemble of aligned embedding spaces (purple) in the ID and OOD settings. Recall@1 and MAP@R are presented. Higher values indicate better performance. The mean and standard deviation (error bars) of the performance metrics are reported for the 5 single models. The ensembles do not have standard deviation since all 5 models are combined into one.

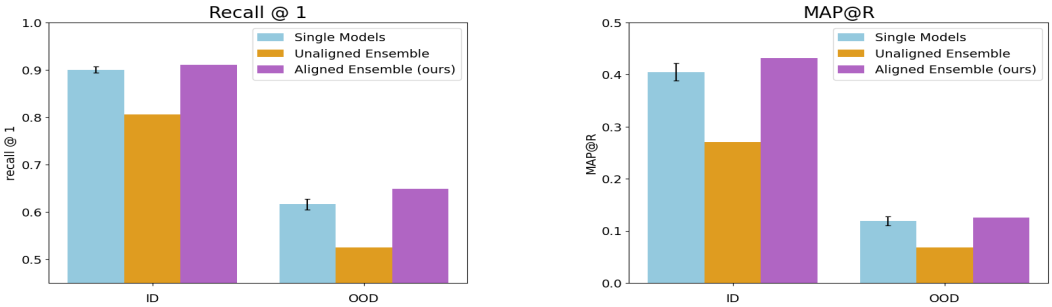


Figure 6: Unsupervised contrastive pre-training with Cropped MNIST as OOD evaluation data. Comparing embedding qualities of single models (blue), an ensemble of unaligned embedding spaces (orange), and an ensemble of aligned embedding spaces (purple) in the ID and OOD settings. Recall@1 and MAP@R are presented. Higher values indicate better performance. The mean and standard deviation (error bars) of the performance metrics are reported for the 5 single models. The ensembles do not have standard deviation since all 5 models are combined into one.

Table 1: Comparison of the embedding qualities of $M = 5$ **single models**, unaligned embedding-space ensemble (**Unaligned Encoders**), and aligned embedding-space ensemble (**Aligned Encoders**) in both in-distribution (ID) and out-of-distribution (OOD) settings. The mean and standard deviation of performance metrics (Recall@1, MAP@R) are reported for the single models. Standard deviation is not shown for the ensembles since all 5 models are combined into one. The top three rows represent Recall@1 performance, and the bottom three rows represent MAP@R performance. The $\%\Delta$ column shows the percentage change in performance for each ensemble type relative to single models. All models are trained on the MNIST dataset using the InfoNCE contrastive loss, where positive pairs are created by applying random rotations to the same input image.

		Single Models	**Unaligned Encoders**		**Aligned Encoders**	
		Mean $\pm$ Std	Ensemble	$\%\ \Delta$	Ensemble	$\%\ \Delta$
Recall@1 (↑)	ID	0.900 ± 0.006	0.806	-10.48%	**0.911**	+1.18%
	OOD (Color)	0.769 ± 0.029	0.664	-13.68%	**0.823**	+6.99%
	OOD (Crop)	0.616 ± 0.011	0.525	-14.80%	**0.649**	+5.32%
MAP@R (↑)	ID	0.405 ± 0.017	0.271	-33.12%	**0.432**	+6.61%
	OOD (Color)	0.174 ± 0.012	0.128	-26.35%	**0.204**	+17.38%
	OOD (Crop)	0.119 ± 0.009	0.068	-42.95%	**0.126**	+5.71%