InfoNCE: Identifying the Gap Between Theory and Practice

Evgenia Rusak[1,2*] , **Patrik Reizinger**[2,4*] , **Attila Juhos**[2,4*] , **Oliver Bringmann**[1],
Roland S. Zimmermann[2,4†] , **Wieland Brendel**[2,3,4†]
University of Tübingen[1] Max-Planck Institute for Intelligent Systems[2]
ELLIS Institute Tübingen[3] Tübingen AI Center[4]

Abstract

Previous theoretical work on contrastive learning (CL) with InfoNCE showed that, under certain assumptions, the learned representations uncover the ground-truth latent factors. We argue these theories overlook crucial aspects of how CL is deployed in practice. Specifically, they assume that within a positive pair, all latent factors either vary to a similar extent or some do not vary at all. However, in practice, positive pairs are often generated using augmentations such as strong cropping to just a few pixels. Hence, a more realistic assumption is that all latent factors change, with a continuum of variability across these factors. We introduce a new contrastive loss, AnInfoNCE, a generalization of InfoNCE containing additional learnable parameters. We provably explain the role of the learnable element in uncovering the latent factors of this anisotropic setting, broadly generalizing previous identifiability results in CL. We validate our identifiability results in controlled experiments and show that AnInfoNCE increases the recovery of previously collapsed information in CIFAR10 and ImageNet, albeit at the cost of downstream accuracy. Upon analyzing our newly developed model it is revealed that CL extracts exactly the shortcut features relevant for discriminating the positive pair from the negative ones.

1. Introduction

Self-Supervised Learning (SSL) employs surrogate objectives to learn useful representations from large, unlabeled data sets. A particularly successful and simple sub-family, called Contrastive Learning (CL), minimizes the representational distance between similar samples (e.g., augmentations of the same sample) while maximizing the distance between dissimilar samples (drawn i.i.d.) [5–7, 10, 41]. Two similar samples form a *positive pair*, while two dissimilar samples form a *negative pair*. Most of today's successful SSL techniques build on this principle, including non-contrastive ones that only use positive pairs [9, 22, 69]. InfoNCE [41] is a commonly used objective for CL. Zimmermann et al. [74] have shown that under certain assumptions about the Data Generating Process (DGP), a model can recover the ground-truth latent factors if trained with InfoNCE. Their identifiability result requires an isotropic change across all latent factors in the positive pair. von Kügelgen et al. [60] relaxed the isotropic assumption on the positive conditional distributions to two partitions: content (δ distribution) and style (non-degenerate distribution). This comes at the cost of a weaker block-identifiability result.

* Equal contribution. † Joint senior authors. Correspondence to evgenia.rusak@uni-tuebingen.de. Code at github.com/brendel-group/AnInfoNCE. Website at brendel-group.github.io/AnInfoNCE.

© .

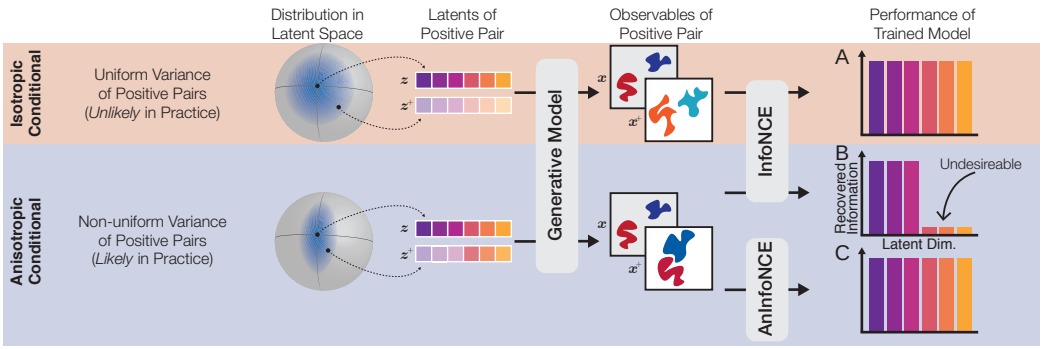

Figure 1: **Illustration of the mismatch between the standard Contrastive Learning (CL) model and practice: A:** CL with the commonly used InfoNCE objective is identifiable when all latents change to the same extent across the positive pair [74], which is unlikely to happen in practice. **B:** The more likely scenario (augmentations affect different latents to a different extent) leads to dimensional collapse and information loss. **C:** Our proposed objective, Anistropic InfoNCE (AnInfoNCE), models features that can vary to a different degree in the positive pair, avoiding collapse.

In real-world datasets, however, positive pairs for CL are generated using augmentations. In practice, a strong cropping augmentation, often extracting just a tiny area of the image, is crucial for high downstream performance [6]. Assuming an underlying Latent Variable Model (LVM), these augmentations correspond to changes in latent factors, modeled by the positive conditional distribution. A strong cropping augmentation discards a significant amount of information and, hence, may harshly change some latent factors. Other latents might change moderately, such as the sharpness of the objects following the Gaussian blurring augmentation, which is applied half of the time. Lastly, latents alluding to class information are invariably left intact. We posit that augmentations imply anisotropic changes on a continuum for different latent factors—a setting previously not covered theoretically. In fact, we believe that this gap between theory and practice can in part explain why empirical work shows that CL loses information [6, 28] *despite* the identifiability guarantees of von Kügelgen et al. [60], Zimmermann et al. [74]. To close this gap, we generalize existing identifiability theory to anisotropic latent distributions by introducing a new variation of InfoNCE, called AnInfoNCE, and proving identifiability for latent factors with varying stability. We validate our results and show promising properties when applied to CIFAR10 and ImageNet. We then discuss the remaining gap between CL theory and practice and extend our theory towards hard negative mining and loss ensembles. Our **contributions** are:

- We introduce a new trainable loss architecture, AnInfoNCE, a generalization of InfoNCE hat can provably uncover the latent factors assuming anisotropic positive pair distributions (Sec. 3).
- We propose a model for hard negative mining that is provably identifiable (Appx. B.2) and also extend our main identifiability result to loss ensembles (Appx. B.3).
- Experimentally, we verify our loss in synthetic and well-controlled image experiments, having full knowledge of the ground-truth generative process (Sec. 4.1 & Appx. F).
- We further demonstrate the efficacy of AnInfoNCE on CIFAR10 and ImageNet in recovering the latent factors, but observe a trade-off with linear readout accuracy (Appx. G).
- After carefully analyzing our newly developed model, we link the observed accuracy-identifiability trade-off on real-world data to the use of augmentations: CL extracts exactly the shortcut features

that are relevant for discriminating the positive pair from the negative ones. We analyze remaining mismatches between CL theory and practice and explore practical mitigation strategies (Appx. H).

2. Background

Self-Supervised/Contrastive Learning. Contrastive Learning (CL) is an SSL paradigm where an encoder $\boldsymbol{f} : \mathcal{X} \to \mathcal{Z}$ learns to map observations $\boldsymbol{x}$ to latent vectors $\boldsymbol{z}$. CL uses positive pairs $(\boldsymbol{x}, \boldsymbol{x}^+)$ (similar data points, e.g., augmentations of the same sample), and negative pairs $(\boldsymbol{x}, \boldsymbol{x}^-)$ (dissimilar data points, e.g., uniformly drawn from the dataset). CL mostly employs the InfoNCE loss [23, 41, 52] with the form:

$$\mathcal{L}_{\text{INCE}}(\boldsymbol{f}) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{x},\boldsymbol{x}^+ \\ \{\boldsymbol{x}^-{}_i\}}} \left[-\ln \frac{e^{\boldsymbol{f}^\top(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}^+)/\boldsymbol{\tau}}}{e^{\boldsymbol{f}^\top(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}^+)/\boldsymbol{\tau}} + \sum_{i=1}^{M} e^{\boldsymbol{f}^\top(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x}^-{}_i)/\boldsymbol{\tau}}} \right], \tag{1}$$

where $\boldsymbol{\tau}$ is the scalar temperature, and $\boldsymbol{x}^+ \sim p(\boldsymbol{x}^+|\boldsymbol{x})$ and $(\boldsymbol{x}, \boldsymbol{x}^-) \sim^{i.i.d.} p(\boldsymbol{x})$ are the positive and negative observation pairs. For analyzing the behavior of CL with InfoNCE, it is beneficial to describe observations by a Data Generating Process (DGP) $\boldsymbol{g} : \mathcal{Z} \to \mathcal{X}$. Here, the samples follow a marginal distribution $p(\boldsymbol{z})$ over the latent space $\mathcal{Z}$, and the positive samples follow a conditional $p(\boldsymbol{z}^+|\boldsymbol{z})$ [74]. Under certain assumptions, CL inverts the DGP, i.e., the composition $\boldsymbol{h} = \boldsymbol{f} \circ \boldsymbol{g}$ is a trivial map (including scaling, permutation, or affine transformations, depending on the assumptions) [74]. More details on the background and related work are provided in Appx. A.

3. Identifiable Anisotropic CL

Setup and Intuition We argue that the augmentations usually employed in Contrastive Learning (CL) do not change all latents evenly. For example, color distortions do not drastically affect latents encoding semantic information. We use a Latent Variable Model (LVM) to model CL, where a positive conditional $p(\boldsymbol{z}^+|\boldsymbol{z})$ relates the positive pairs $\boldsymbol{z}^+$ to the anchor sample $\boldsymbol{z}$. The augmentations used in practice induce a specific form of the positive conditional $p(\boldsymbol{z}^+|\boldsymbol{z})$, e.g., by assigning larger variance to latents strongly affected by augmentations. Following the practice of CL, we assume the latent space to be a $(d-1)$-dimensional hypersphere: $\mathcal{Z} = \mathcal{S}^{d-1}$. As per previous work [74], the anchor latent $\boldsymbol{z}$, and all negative pairs $\{\boldsymbol{z}^-{}_i\}$ are assumed to be uniformly distributed on $\mathcal{Z}$. We allow non-uniform variances for the individual latent factors in the positive pair $\boldsymbol{z}^+$, akin to [31]. We model this in the positive conditional by scaling the latents with *unknown* positive *concentration parameters*, collected in the diagonal matrix $\boldsymbol{\Lambda}$:

$$p(\boldsymbol{z}) = \text{const}, \quad p(\boldsymbol{z}^+|\boldsymbol{z}) \propto e^{-(\boldsymbol{z}^+ - \boldsymbol{z})^\top \boldsymbol{\Lambda} (\boldsymbol{z}^+ - \boldsymbol{z})}, \tag{2}$$

where a higher value $\boldsymbol{\Lambda}_{ii}$ means a higher concentration of $\boldsymbol{z}^+$ around the anchor $\boldsymbol{z}$ along dimension i. We assume that observations $\boldsymbol{x}, \boldsymbol{x}^+, \{\boldsymbol{x}^-{}_i\}$ are the result of passing the respective latents through an invertible generator $\boldsymbol{g} : \mathcal{Z} \to \mathcal{X} \subseteq \mathbb{R}^D$. Our goal is to train an encoder $\boldsymbol{f} : \mathbb{R}^D \to \mathcal{Z}$ such that the reconstructed latents, $\hat{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{x})$, recover $\boldsymbol{z}$ up to permissible transformations. To train $\boldsymbol{f}$, we introduce the **AnInfoNCE** loss:

$$\mathcal{L}_{\text{AINCE}}(\boldsymbol{f}, \hat{\boldsymbol{\Lambda}}) = \underset{\substack{\boldsymbol{x},\boldsymbol{x}^+ \\ \{\boldsymbol{x}^-{}_i\}}}{\mathbb{E}} \left[-\ln \frac{e^{-\left\|\boldsymbol{f}(\boldsymbol{x}^+)-\boldsymbol{f}(\boldsymbol{x})\right\|^2_{\hat{\boldsymbol{\Lambda}}}}}{e^{-\|f(\boldsymbol{x}^+)-\boldsymbol{f}(\boldsymbol{x})\|^2_{\hat{\boldsymbol{\Lambda}}}} + \sum_{i=1}^{M} e^{-\|\boldsymbol{f}(\boldsymbol{x}^-{}_i)-\boldsymbol{f}(\boldsymbol{x})\|^2_{\hat{\boldsymbol{\Lambda}}}}} \right], \tag{3}$$

where $\hat{\boldsymbol{\Lambda}}$ is a *trainable* diagonal matrix, and $-\|\Delta\boldsymbol{f}(\boldsymbol{x}^+)\|^2_{\hat{\boldsymbol{\Lambda}}} = -(\Delta\boldsymbol{f}(\boldsymbol{x}^+))^\top \hat{\boldsymbol{\Lambda}}(\Delta\boldsymbol{f}(\boldsymbol{x}^+))$, is the *similarity function*. Our idea is that the similarity function should flexibly accommodate latents anisotropically perturbed by the augmentations. We summarize our assumptions in Appx. B.1.

Identifiability of Anisotropic CL When studying identifiability, we aim to recover the ground-truth latent $\boldsymbol{z}$ from observations $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{z})$ with as little ambiguity as possible. Our main contribution is proving the identifiability of the DGP from Assum. 1, generalizing previous results [60, 74] (details and proof in Appx. D):

Theorem 3.1 (Identifiability of Anisotropic CL) *Under Assum. 1, if a pair $(\boldsymbol{f}, \hat{\boldsymbol{\Lambda}})$ minimizes $\mathcal{L}_{\text{AINCE}}$ (3), then $\boldsymbol{f} \circ \boldsymbol{g}$ is a block-orthogonal transformation, where each block acts on latents with equal weight $\boldsymbol{\Lambda}_{ii}$. In other words, $\boldsymbol{z}$ is identified up to a block-orthogonal transformation.*

Thm. 3.1 means that if the encoder $\boldsymbol{f}$, usually parametrized by a neural network, is expressive enough to minimize AnInfoNCE, then the reconstructed latent $\hat{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{x})$ is related to the ground-truth $\boldsymbol{z}$ via a simple orthogonal linear transformation. This simple ambiguity remains inconsequential as solving any downstream task involves at least one linear layer trained on top of the backbone $\boldsymbol{f}$.

We extended our identifiability result to two very important cases: hard-negative mining and ensemble losses. The statements and intuition can be found in Appx. B.

4. Experiments

We now validate our theoretical results in experiments with increasing complexity. In this Section, we focus on our main result and demonstrate the efficacy of AnInfoNCE in various settings. We include additional experimental ablations, results for hard negative mining and ensembling AnInfoNCE as well as detailed compute requirements for all our experiments in the Appendix.

4.1. Synthetic Experiments

Preliminaries. We follow Zimmermann et al. [74] and consider a fully-controlled Data Generating Process (DGP), implementing Assum. 1. That is, the ground-truth latent space is a hypersphere ($\mathcal{Z} = \mathcal{S}^{d-1}$), and positive/negative pairs are defined through conditional sampling in the latent space. We model the generative process $\boldsymbol{g}$ as an invertible Multi-Layer Perceptron (MLP) with the same input, intermediate, and output dimensionality d and with Leaky ReLU non-linearities. We set $d = 10$. The encoder $\boldsymbol{f}$ is also parameterized by an MLP with the same input, intermediate, and output dimensionality, and with Leaky ReLU non-linearities. The encoder is trained only with observations, which are obtained by passing $\boldsymbol{z}$ through $\boldsymbol{g}$. As we have access to the ground-truth $\boldsymbol{z}$, we can post-hoc compute the information captured in the inferred latents $\hat{\boldsymbol{z}}$ about the ground-truth $\boldsymbol{z}$. We do this by fitting a linear map $\mathbf{A}$ between $\hat{\boldsymbol{z}}$ and $\boldsymbol{z}$ and compute the average coefficient of determination R^2 [65] between the relevant features (e.g. all of them, content-only dimensions, style-only dimensions) of $\mathbf{A}\hat{\boldsymbol{z}}$ and $\boldsymbol{z}$.

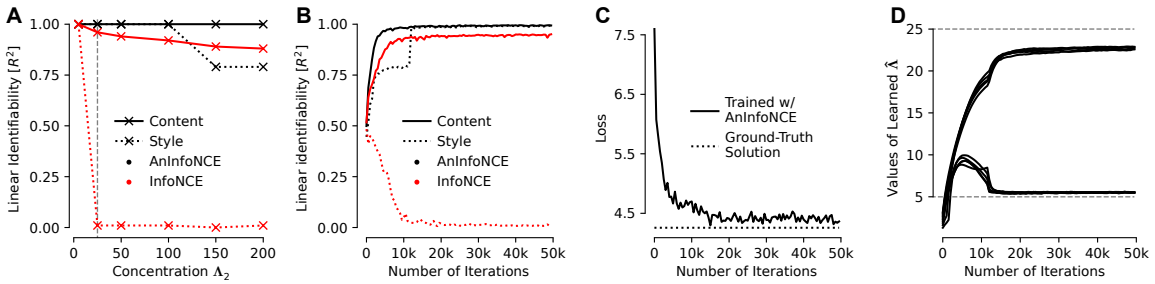

Figure 2: **Behavior of AnInfoNCE on synthetic data.** ($\Lambda_1 = 5$) **A:** We maintain high R^2-scores with AnInfoNCE for both content and style dimensions, while the style dimensions are lost when training with the regular InfoNCE loss. For $\Lambda_2 = 25$ (dotted black vertical line), we show: **B:** The evolution of the linear R^2 scores during training; **C:** AnInfoNCE reaches the global minimum, computed based on ground-truth latents; **D:** The evolution of the learned $\hat{\mathbf{\Lambda}}$ values.

Results. We study the effect of anisotropy in the positive conditional by varying the ground-truth $\mathbf{\Lambda}$: We keep the value of five latent dimensions fixed at $\Lambda_1 = 5$, while we vary the value of the other five dimensions (i.e., Λ_2), resulting in a ground-truth diagonal $\mathbf{\Lambda} = \operatorname{diag}(\Lambda_1, \dots, \Lambda_1, \Lambda_2, \dots \Lambda_2)$. Higher (lower) values reflect more content-like (style-like) dimensions. We train an encoder on the observed data and summarize the results in Fig. 2**A**. In contrast to regular InfoNCE, AnInfoNCE produces high R^2 scores over a wide range of $\mathbf{\Lambda}$-values for both content and style latents (markers in Fig. 2**A** indicate converged training runs). We also analyze the setting where the conditionals of five latents are distributed according to $\mathbf{\Lambda}_2 = 25$ in more detail in Fig. 2**B-D**. We observe that content dimensions are learned before style (Fig. 2**B**). Our loss converges to the global minimum value, which can be computed based on the ground-truth latents (Fig. 2**C**). When learning $\mathbf{\Lambda}$, we observe that the learned $\hat{\mathbf{\Lambda}}$ values almost match the ground-truth values (dashed black lines in Fig. 2**D**). In Appx. F we perform experiments with a VAE trained on MNIST [15] and have similar results.

Real world experiments. In Appx. G we conduct real world experiments on CIFAR10 [33] and ImageNet [48]. Tab. 1 summarizes our results. For details and interpretation, see Appx. G.

5. Conclusion

Our work advances CL theory by generalizing InfoNCE. The proposed framework, AnInfoNCE-based CL, is identifiable and relies on more realistic assumptions on the data, reducing the gap between CL theory and practice: It can account for anisotropic latent conditionals that more likely reflect how augmentations affect the latent factors in practice. In addition, we showed extensions of our results towards other empirical techniques such as hard negative mining and loss ensembling. We demonstrated that our framework accounts for certain known limitations of CL, such as collapsing information, by better capturing style information. At the same time, we showed that there exists a trade-off between recovered style information and downstream classification.

We hope that further investigations into overcoming the remaining gap between theory and practice will improve practical self-supervised learning techniques and our understanding of them. See discussions in Appx. H and Appx. I.

Author Contributions

WB conceived the project idea. ER led the project. ER, RSZ, and WB designed the experiments and ER executed them. ER conducted the analysis with support from RSZ and PR. ER and RSZ jointly created all figures. PR conceived how to incorporate anisotropy in SimCLR. This idea led to WB and RSZ developing the main theorem. AJ developed corollaries for hard negative mining and loss ensembling with help from PR and feedback from RSZ. RSZ and WB jointly supervised the project. AJ was responsible for presenting the theoretical results in a standardized way. ER, PR, AJ, RSZ, and WB contributed to writing the manuscript.

Acknowledgements

The authors thank Julian Bitterwolf, Jack Brady, Steffen Schneider, Thaddäus Wiedemer, and Zac Cranko for helpful discussions. We would also like to thank Mihály Weiner for all the insights that aided us in filling edge cases of our proof. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Evgenia Rusak, Patrik Reizinger, Attila Juhos, and Roland S. Zimmermann. Patrik Reizinger acknowledges his membership in the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program. This work was supported by the German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A. WB acknowledges financial support via an Emmy Noether Grant funded by the German Research Foundation (DFG) under grant no. BR 6382/1-1 and via the Open Philantropy Foundation funded by the Good Ventures Foundation. WB is a member of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645. This research utilized compute resources at the Tübingen Machine Learning Cloud, DFG FKZ INST 37/1057-1 FUGG.

References

[1] Mido Assran, Randall Balestriero, Quentin Duval, Florian Bordes, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, and Nicolas Ballas. The hidden uniform cluster prior in self-supervised learning. In *ICLR*. OpenReview.net, 2023.

[2] Randall Balestriero and Yann LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. In *NeurIPS*, 2022.

[3] Ihab Bendidi, Adrien Bardes, Ethan Cohen, Alexis Lamiable, Guillaume Bollot, and Auguste Genovesio. No Free Lunch in Self Supervised Representation Learning. *arXiv preprint*, 2023.

[4] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL http://www.blender.org.

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*, 2020.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*. PMLR, 2020.

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big Self-Supervised Models are Strong Semi-Supervised Learners. *Advances in neural information processing systems*, 2020.

[8] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. In *NeurIPS*, 2021.

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*. Computer Vision Foundation / IEEE, 2021.

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved Baselines with Momentum Contrastive Learning. *arXiv preprint*, 2020.

[11] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *NeurIPS*, 2020.

[12] Romain Cosentino, Anirvan Sengupta, Salman Avestimehr, Mahdi Soltanolkotabi, Antonio Ortega, Ted Willke, and Mariano Tepper. Toward a Geometrical Understanding of Self-supervised Contrastive Learning. *arXiv preprint*, 2022.

[13] Jingyi Cui, Weiran Huang, Yifei Wang, and Yisen Wang. AggNCE: Asymptotically Identifiable Contrastive Learning. In *NeurIPS 2022 Workshop: Self-Supervised Learning - Theory and Practice*, 2022.

[14] Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljačić. Equivariant Contrastive Learning. *arXiv preprint*, 2021.

[15] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, (6), 2012.

[16] Yann Dubois, Stefano Ermon, Tatsunori B. Hashimoto, and Percy Liang. Improving self-supervised learning by characterizing idealized representations. In *NeurIPS*, 2022.

[17] Cian Eastwood, Julius von Kügelgen, Linus Ericsson, Diane Bouchacourt, Pascal Vincent, Bernhard Schölkopf, and Mark Ibrahim. Self-Supervised Disentanglement by Leveraging Structure in Data Augmentations. *arXiv preprint*, 2023.

[18] Facebook Research. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. https://github.com/facebookresearch/vicreg, 2022.

[19] FOLDING.LAR.SYSTEMS. Gpu folding overview. https://folding.lar.systems/gpu_ppd/brands/nvidia/folding_profile/ga100_a100_pcie_40gb, Accessed on May 14th, 2024.

[20] Quentin Garrido, Yubei Chen, Adrien Bardes, Laurent Najman, and Yann LeCun. On the duality between contrastive and non-contrastive self-supervised learning. In *ICLR*. OpenReview.net, 2023.

[21] Luigi Gresele, Paul K. Rubenstein, Arash Mehrjou, Francesco Locatello, and Bernhard Schölkopf. The incomplete rosetta stone problem: Identifiability results for multi-view nonlinear ICA. In *UAI*. AUAI Press, 2019.

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *NeurIPS*, 2020.

[23] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*. JMLR.org, 2010.

[24] Jeff Z. HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. In *NeurIPS*, 2021.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*. IEEE Computer Society, 2016.

[26] Patrick Hua. Simsiam: A pytorch implementation. https://github.com/PatrickHua/SimSiam, 2021.

[27] Aapo Hyvärinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ICA. In *NeurIPS*, 2016.

[28] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *ICLR*. OpenReview.net, 2022.

[29] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

[30] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[31] Michael Kirchhof, Karsten Roth, Zeynep Akata, and Enkelejda Kasneci. A non-isotropic probabilistic take on proxy-based deep metric learning. In *ECCV (26)*. Springer, 2022.

[32] Michael Kirchhof, Enkelejda Kasneci, and Seong Joon Oh. Probabilistic contrastive learning recovers the correct aleatoric uncertainty of ambiguous inputs. In *ICML*. PMLR, 2023.

[33] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

[34] Hankook Lee, Kibok Lee, Kimin Lee, Honglak Lee, and Jinwoo Shin. Improving transferability of representations via augmentation-aware self-supervision. In *NeurIPS*, 2021.

[35] Bingbin Liu, Elan Rosenfeld, Pradeep Kumar Ravikumar, and Andrej Risteski. Analyzing and improving the optimization landscape of noise-contrastive estimation. In *ICLR*. OpenReview.net, 2022.

[36] Qi Lyu, Xiao Fu, Weiran Wang, and Songtao Lu. Latent Correlation-Based Multiview Learning and Self-Supervision: A Unifying Perspective. *arXiv preprint*, 2021.

[37] Zhuang Ma and Michael Collins. Noise Contrastive Estimation and Negative Sampling for Conditional Models: Consistency and Statistical Efficiency. *arXiv preprint arXiv:1809.01812*, 2018.

[38] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *ACM International Conference on Multimedia*, 2010.

[39] Stefan Matthes, Zhiwei Han, and Hao Shen. Towards a Unified Framework of Contrastive Learning for Disentangled Representations. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[40] Dirk Merkel. Docker: Lightweight linux containers for consistent development and deployment. *Linux J.*, 2014(239), March 2014. ISSN 1075-3583.

[41] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint*, 2018.

[42] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[43] Goutham Rajendran, Patrik Reizinger, Wieland Brendel, and Pradeep Ravikumar. An interventional perspective on identifiability in gaussian lti systems with independent component analysis. *arXiv preprint*, 2023.

[44] Rensmart. Carbon emissions calculator. https://www.rensmart.com/Calculators/KWH-to-CO2, Accessed on May 14th, 2024.

[45] Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? In *NeurIPS*, 2021.

[46] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *ICLR*. OpenReview.net, 2021.

[47] Geoffrey Roeder, Luke Metz, and Durk Kingma. On linear identifiability of learned representations. In *ICML*. PMLR, 2021.

[48] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, (3), 2015.

[49] Nikunj Saunshi, Jordan Ash, Surbhi Goel, Dipendra Misra, Cyril Zhang, Sanjeev Arora, Sham Kakade, and Akshay Krishnamurthy. Understanding contrastive learning requires incorporating inductive biases. In *ICML*. PMLR, 2022.

[50] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *ICLR*, 2014.

[51] James B Simon, Maksis Knutins, Liu Ziyin, Daniel Geisz, Abraham J Fetterman, and Joshua Albrecht. On the stepwise nature of self-supervised learning. In *ICML*, 2023.

[52] Kihyuk Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. *Advances in neural information processing systems*, 29, 2016.

[53] Sylabs Inc. Singularity 2.5.2 - linux application and environment containers for science. https://doi.org/10.5281/zenodo.1308868, 2018.

[54] Alex Tamkin and Xiluo He. Feature Dropout: Revisiting the Role of Augmentations in Contrastive Learning. In *NeurIPS 2022 Workshop: Self-Supervised Learning - Theory and Practice*, 2022.

[55] Techpowerup. Gpu specs database. https://www.techpowerup.com/gpu-specs, Accessed on May 14th, 2024.

[56] Yuandong Tian. Understanding the role of nonlinearity in training dynamics of contrastive learning. In *ICLR*. OpenReview.net, 2023.

[57] Travel Navigator. https://travelnav.com/emissions-from-new-york-ny-to-london-unite Accessed on May 14th, 2024.

[58] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: https://doi.org/10.1038/s41592-019-0686-2.

[59] Julius von Kügelgen, Paul K. Rubenstein, Bernhard Schölkopf, and Adrian Weller. Optimal experimental design via bayesian optimization: active causal structure learning for gaussian process networks. *arXiv preprint*, 2019.

[60] Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. In *NeurIPS*, 2021.

[61] Diane Wagner, Fabio Ferreira, Danny Stoll, Robin Tibor Schirrmeister, Samuel Müller, and Frank Hutter. On the Importance of Hyperparameters and Data Augmentation for Self-Supervised Learning. *arXiv preprint*, 2022.

[62] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *CVPR*. Computer Vision Foundation / IEEE, 2021.

[63] Yifei Wang, Qi Zhang, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Chaos is a ladder: A new theoretical understanding of contrastive learning via augmentation overlap. In *ICLR*. OpenReview.net, 2022.

[64] Zixin Wen and Yuanzhi Li. Toward understanding the feature learning process of self-supervised contrastive learning. In *ICML*. PMLR, 2021.

[65] Sewall Wright. Correlation and causation. *Journal of Agricultural Research*, (7), 1921.

[66] Mike Wu, Chengxu Zhuang, Milan Mosse, Daniel Yamins, and Noah Goodman. On Mutual Information in Contrastive Learning for Visual Representations. *arXiv preprint*, 2020.

[67] Tete Xiao, Xiaolong Wang, Alexei A. Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. In *ICLR*. OpenReview.net, 2021.

[68] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations*, 2023.

[69] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. In Marina Meila and Tong Zhang, editors, *ICML*. PMLR, 2021.

[70] Runtian Zhai, Bingbin Liu, Andrej Risteski, Zico Kolter, and Pradeep Ravikumar. Understanding Augmentation-based Self-Supervised Representation Learning via RKHS Approximation. *arXiv preprint*, 2023.

[71] Junbo Zhang and Kaisheng Ma. Rethinking the augmentation module in contrastive learning: Learning hierarchical augmentation invariance with expanded views. In *CVPR*, 2022.

[72] Qi Zhang, Yifei Wang, and Yisen Wang. Identifiable contrastive learning with automatic feature importance discovery. In *NeurIPS*, 2023.

[73] Huangjie Zheng, Xu Chen, Jiangchao Yao, Hongxia Yang, Chunyuan Li, Ya Zhang, Hao Zhang, Ivor Tsang, Jingren Zhou, and Mingyuan Zhou. Contrastive Attraction and Contrastive Repulsion for Representation Learning. *arXiv preprint*, 2021.

[74] Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive Learning Inverts the Data Generating Process. In *ICML*. PMLR, 2021.

[75] Liu Ziyin, Ekdeep Singh Lubana, Masahito Ueda, and Hidenori Tanaka. Loss Landscape of Self-Supervised Learning. In *NeurIPS 2022 Workshop: Self-Supervised Learning - Theory and Practice*, 2022.

Appendix A. Related work

Practical Shortcomings of CL. The identifiability of SSL is has been analyzed by numerous works, e.g. by Lyu et al. [36], von Kügelgen et al. [59] and Cui et al. [13], with insightful connections between contrastive and non-contrastive methods in [1, 2, 20].

Despite the identifiability guarantees of CL methods [13, 60, 74], practitioners struggle to make models learn all underlying latent factors. Chen et al. [6] noted that the representation could become invariant to some input transformations. Several works connected data augmentations to failing to learn all latent factors [3, 12, 24, 28, 61, 63, 66, 70].

A potential way to improve augmentation readout (a proxy for capturing the information in highly-varying, i.e., style, latents) is to introduce inductive biases into the training process. Lee et al. [34] add a regularizer to predict the augmentation parameters between two views, whereas Dangovski et al. [14] use a regularizer to induce equivariance by predicting rotation parameters. Xiao et al. [67] introduce Leave-one-out Contrastive Learning (LooC) with multiple latent spaces. They train multiple embeddings by leaving out one augmentation at a time, thus inducing different invariances, which results in better downstream and augmentation readout performance. Zhang and Ma [71] propose a similar strategy with different augmentation partitions and calculate the contrastive loss at different hierarchical levels in the encoder, meant to induce invariances at different feature levels. Recently, Eastwood et al. [17] proposed a similar method to LooC [67], but with strong theoretical guarantees.

Content-style Partitioning of $\mathcal{Z}$. von Kügelgen et al. [60] present an identifiability result for an assumed partitioning of $\boldsymbol{z} \in \mathcal{Z}$ into invariant (*content*) $\boldsymbol{z}_c \in \mathcal{Z}_c$ and variant (*style*) $\boldsymbol{z}_s \in \mathcal{Z}_s$ dimensions with $\mathcal{Z} = \mathcal{Z}_c \times \mathcal{Z}_s$. That is, the content latents $\boldsymbol{z}_c$ in the positive pairs are assumed to follow a δ-distribution, whereas the distribution of $\boldsymbol{z}_s$ is not degenerate, yielding $p\left(\boldsymbol{z}^+|\boldsymbol{z}\right) = \delta\left(\boldsymbol{z}_c^+ - \boldsymbol{z}_c\right) p(\boldsymbol{z}_s^+|\boldsymbol{z}_s)$. Note that Jing et al. [28] implicitly use a similar distinction—defined as different augmentation amplitudes. Thus, their approach is akin to generalized content and style variables with variances $\boldsymbol{\sigma}_s^2 \gg \boldsymbol{\sigma}_c^2 > 0$; Dubois et al. [16] also have a similar approach. Other means to partition the latent variables are tied to the "simplicity" of features [8] or sparsity arguments [64]. In concurrent work, Eastwood et al. [17] propose a *relative* notion of content and style, assuming the presence of multiple environments, each with distinct content and style dimensions.

Zimmermann et al. [74] assumes that the latent variables are interchangeable for both their marginals and conditionals are the same. This gives rise to specific symmetries in the latent space $\mathcal{Z}$, which might become restrictive in real-world scenarios. von Kügelgen et al. [60] introduces a two-fold partitioning of $\mathcal{Z}$, where the marginal is assumed to be the same for all components z_i of the latent vector $\boldsymbol{z}$. Namely, they assume that some latent factors (termed content variables $\boldsymbol{z}_c$) have a delta distribution as conditional, i.e., those variables do not change across the positive pairs. This model has been deemed useful in understanding why dimensionality collapse happens. However, such an assumption might be too simplistic in specific scenarios (at least for some of the latents), which depends on the augmentation strategy [8, 16, 54, 56, 64, 70, 75]. Thus, we can think of Jing et al. [28] as a further generalization, though the authors do not make the connection explicit. They consider augmentations, but their reasoning concerns the observation space $\mathcal{X}$. They conclude—similar to [16, 54]—that the stronger the augmentation, the more prevalent the collapse. Both to generalize and to formalize these insights, we introduce the notion of partitioning of $\mathcal{Z}$:

Definition A.1 (Partitions of $\mathcal{Z}$) *Given a latent space $\mathcal{Z}$ and a latent vector $\boldsymbol{z} \in \mathcal{Z}$, we say that subsets of latent components $z_i^k, \dots, z_{i+n-1}^k$ belong to partition $\mathcal{Z}^k$ of dimension n, when in the ground truth DGP their conditional variances $\sigma^2_{\tilde{z}_i|\boldsymbol{z}}$ are the same, i.e.:*

$$\forall j \neq l \in \{i, \dots, i+n-1\} : \sigma^2_{z_j^+|\boldsymbol{z}} = \sigma^2_{z_l^+|\boldsymbol{z}}. \tag{4}$$

When all conditional variances are equal, then we get back the setting of Zimmermann et al. [74]; when the number of partitions is two, with one of the conditional variances being zero, then we are in the regime of von Kügelgen et al. [60]. We formalize this scenario as follows:

Definition A.2 (Content–style partitioning of $\mathcal{Z}$) *Following von Kügelgen et al. [60], the latent space $\mathcal{Z}$ is said to be partitioned into content ($\boldsymbol{z}_c$, invariant) and style ($\boldsymbol{z}_s$, changing) subspaces if and only if*

i) $\mathcal{Z} = \mathcal{Z}_c \times \mathcal{Z}_s : \mathcal{Z}_c \cap \mathcal{Z}_s = \emptyset; \quad \dim \mathcal{Z}_c + \dim \mathcal{Z}_s = \dim \mathcal{Z}$;
ii) the conditional distribution of $\boldsymbol{z}_c$ latents is a δ-distribution [60, Assum. 3.1];
iii) the conditional distribution of $\boldsymbol{z}_s$ latents is non-degenerate [60, Assum. 3.2];

yielding the following factorization of the latent conditional $p(\boldsymbol{z}^+|\boldsymbol{z}) = \delta(\boldsymbol{z}^{+,c} - \boldsymbol{z}_c)\, p(\boldsymbol{z}^{+,s}|\boldsymbol{z}_s)$.

That is, our DGP does not cover the exact setting of von Kügelgen et al. [60], as we cannot have zero variance. If having zero variance (or equivalently, infinite concentration in $\boldsymbol{\Lambda}$), then we might say that our results incorporate the content-style setting.

Appendix B. Theoretical assumptions. Hard Negative Mining and Ensemble Loss

B.1. Theoretical assumptions

Assumption 1 (Anisotropic CL on $\mathcal{S}^{d-1}$) *The **DGP** satisfies (cf. Defn. D.1):*

i) $\mathcal{Z} = \mathcal{S}^{d-1}$, $\mathcal{X} \subseteq \mathbb{R}^D$ *are latent and observation spaces.*
ii) Anchor is uniform on $\mathcal{Z}$, i.e., $p(\boldsymbol{z}) = \text{const}$.
iii) For $\boldsymbol{\Lambda} \in \mathbb{R}^{d \times d}$ PD diagonal matrix, the positive conditional $p(\boldsymbol{z}^+|\boldsymbol{z})$ has the form of (2).
iv) Negative pairs $\{\boldsymbol{z}^-{}_i\}$ are uniform on $\mathcal{Z}$.
v) Continuous, invertible generator $\boldsymbol{g}$ maps $\boldsymbol{z}, \boldsymbol{z}^\pm \in \mathcal{Z} \overset{\boldsymbol{g}}{\mapsto} \boldsymbol{x}, \boldsymbol{x}^\pm \in \mathcal{X}$.

*The **model** satisfies (cf. Defn. D.2):*

vi) Encoder $\boldsymbol{f} : \mathbb{R}^D \to \mathcal{S}^{d-1}$ is continuous.
vii) $\hat{\boldsymbol{\Lambda}} \in \mathbb{R}^{d \times d}$ is a PD diagonal matrix.
viii) $\boldsymbol{f}$ and $\hat{\boldsymbol{\Lambda}}$ are trained with $\mathcal{L}_{\text{AINCE}}$, i.e. (3).

Collectively i)-viii) define an anisotropic CL problem .

B.2. Extending Identifiability to Hard Negative Mining

Our generalized result in Sec. 3 accounts for anisotropically varying latent factors. Here, we build on Thm. 3.1 to theoretically model the practice of hard negative (HN) mining. This technique effectively uses a negative conditional distribution to select negative samples more similar to the anchor, instead of sampling from a uniform marginal. Put differently, hard negative (HN) mining

makes the classification problem of positive/negative samples *more difficult*, which is reasoned to help avoid latent collapse [11, 29, 45, 46, 62, 66, 68, 73]. Our proposed anisotropic loss still identifies the ground-truth latents, when the negative conditional has the same form as (2) (we denote its concentration parameter as $\mathbf{\Lambda}^-$, and the one of the positive conditional as $\mathbf{\Lambda}^+$). Thus, we further generalize the setting of Sec. 3, which is subsumed as $\mathbf{\Lambda}^- = \mathbf{0}$ (i.e., a uniform marginal). Intuitively, with a negative conditional, the density ratios will be the same as with a uniform marginal and a positive conditional with $\mathbf{\Lambda} = \mathbf{\Lambda}^+ - \mathbf{\Lambda}^-$.

Corollary B.1 (Identifiability with hard negative mining) *Under Assum. 1, but with a negative conditional in iv) as (2) with a concentration parameter $\mathbf{\Lambda}^-$, such that $\mathbf{\Lambda} = \mathbf{\Lambda}^+ - \mathbf{\Lambda}^-$ has full rank, the pair $(\boldsymbol{f}, \hat{\mathbf{\Lambda}})$ minimizing $\mathcal{L}_{\text{AINCE}}$ identifies the latent factors up to a block-orthogonal transformation.*

Proof [Proof Sketch (full proof in Appx. D.2)] The minimum loss is attained iff the similarity function matches the log-density-ratio between the positive and negative conditionals (Thm. C.1). As the conditionals have the same distributional form, the fraction can be simplified to one expression equivalent to the one obtained with a uniform marginal and a positive conditional with $\mathbf{\Lambda} = \mathbf{\Lambda}^+ - \mathbf{\Lambda}^-$. The rest of the proof is analogous to that of Thm. 3.1. ■

We include experimental results for hard negative mining in Appx. I.3. We empirically show that our theory correctly predicts the loss optimum at $\mathbf{\Lambda} = \mathbf{\Lambda}^+ - \mathbf{\Lambda}^-$, and that fine-tuning an encoder on a concatenation of hard negative samples and regular negative samples improves the identifiability score.

B.3. Extending Identifiability to Ensemble AnInfoNCE

Recent works started exploring both theoretically [17, 32] and empirically [67, 71] the advantages of ensemble CL losses, where each loss component is calculated with different (potentially partially overlapping) data augmentations. We show that an ensemble version of AnInfoNCE is also identifiable. For more details, see Appx. D.3.

Corollary B.2 (Identifiability of ensemble AnInfoNCE) *Assume k DGPs, each satisfying Assum. 1. Assume that they share $\boldsymbol{g}$, but have different concentration parameters $\{\mathbf{\Lambda}^i\}$, for $1 \leq i \leq k$. Assume a shared encoder $\boldsymbol{f}$ and k learnable $\hat{\mathbf{\Lambda}}^i$ parameters. Then, the tuple $(\boldsymbol{f}, \{\hat{\mathbf{\Lambda}}^i\})$ minimizing the ensemble loss $\sum_i^k \mathcal{L}_{\text{AINCE}}(\boldsymbol{f}, \hat{\mathbf{\Lambda}}^i)$ identifies the latents up to a block-orthogonal transformation.*

Proof $\boldsymbol{f} = \boldsymbol{g}^{-1}$ together with $\hat{\mathbf{\Lambda}}^i = \mathbf{\Lambda}^i$ minimizes the ith loss term. As $\boldsymbol{g}$ is shared, the individual losses can attain their minima simultaneously. Hence, this also holds for any minimum of the ensemble loss. Applying Thm. 3.1 to either of them concludes the proof. ■

Cor. B.2 shows that using an ensemble with proper augmentations could lead to stronger identifiability guarantees (e.g., up to permutation, cf. Rem. D.4). Ideally, we could design such sets of augmentations, but it is not evident that this is possible in practice.

We include experimental results for ensemble AnInfoNCE in Appx. I.3. To verify Cor. B.2, we define two DGPs that share a uniform anchor distribution and differ in their respective conditional

distributions. We sample data from both DGPs, calculate the individual losses, and train the encoder on the sum of both losses. We see improved linear identifiability scores with ensemble AnInfoNCE, compared to training the encoder with InfoNCE or AnInfoNCE.

Appendix C. Contrastive Learning: Bayes Optimum

For the sake of completeness, this section presents contrastive learning in a more general form, alongside a theorem regarding its unconditional (Bayes-) optimum. This theorem is referenced several times across the identifiability proofs in Appx. D. Although the content of this section is considered to be known by the research community, we still decided to include it to make the presentation of the theory more self-contained. Appx. D presents the novelties introduced by this paper.

Remark C.1 *In the following, we will refer to vectors as $\boldsymbol{y}$ to provide a unified treatment both for $\boldsymbol{y} = \boldsymbol{z}$ and $\boldsymbol{y} = \boldsymbol{x}$.*

Definition C.1 (General Contrastive Learning (CL) Problem) *Let $\mathcal{Y}$ be a data domain. Let $\boldsymbol{Y} := (\boldsymbol{y}, \boldsymbol{y}^+, \boldsymbol{y}_1^-, \dots, \boldsymbol{y}_M^-) \sim p_{\boldsymbol{Y}}$ be a random vector with all elements coming from $\mathcal{Y}$. We refer to $\boldsymbol{y}$ as the* anchor point*, to $\boldsymbol{y}^+$ as its* positive pair *and to $\boldsymbol{y}_i^-$ as one of its* negative pairs*. Let us assume that*

- *i) the pairs $\boldsymbol{y}^+, \boldsymbol{y}_1^-, \dots, \boldsymbol{y}_M^-$ are conditionally completely independent when conditioned on $\boldsymbol{y}$ and*
- *ii) the conditional distribution of negative pairs w.r.t. the anchor coincide, i.e., for any i, j we have $p_{\boldsymbol{y}_i^-|\boldsymbol{y}} = p_{\boldsymbol{y}_j^-|\boldsymbol{y}}$.*

Let $\mathcal{U}$ be a class of bivariate functions $u : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, hereinafter called similarity functions *and let us optimize the following objective amongst all functions $u \in \mathcal{U}$:*

$$\min_{u \in \mathcal{U}} \quad \mathcal{L}_g(u) := \mathbb{E}\left[-\ln \frac{e^{u(\boldsymbol{y}, \boldsymbol{y}^+)}}{e^{u(\boldsymbol{y}, \boldsymbol{y}^+)} + \sum_{j=1}^M e^{u(\boldsymbol{y}, \boldsymbol{y}_j^-)}} \right]. \tag{5}$$

The tuple $(\mathcal{Y}, p_{\boldsymbol{Y}}, \mathcal{U}, \mathcal{L}_g)$ is called a general contrastive learning (CL) problem*.*

Remark C.2 *Denoting $p_{\text{pos}} := p_{\boldsymbol{y}^+|\boldsymbol{y}}$ and $p_{\text{neg}} := p_{\boldsymbol{y}_j^-|\boldsymbol{y}}$, the latter being independent of the exact choice of j per assumption ii), the joint data distribution is the following:*

$$p_{\boldsymbol{Y}}(\boldsymbol{y}, \boldsymbol{y}^+, \boldsymbol{y}_1^-, \dots, \boldsymbol{y}_M^-) = p_{\boldsymbol{y}}(\boldsymbol{y})\, p_{\text{pos}}(\boldsymbol{y}^+|\boldsymbol{y}) \prod_{j=1}^M p_{\text{neg}}(\boldsymbol{y}_j^-|\boldsymbol{y}). \tag{6}$$

Theorem C.1 (Bayes-Optima of CL) *Let $(\mathcal{Y}, p_{\boldsymbol{Y}}, \mathcal{U}, \mathcal{L}_g)$ be a general CL problem (Defn. C.1) with $\boldsymbol{Y} := (\boldsymbol{y}, \boldsymbol{y}^+, \boldsymbol{y}_1^-, \dots, \boldsymbol{y}_M^-) \sim p_{\boldsymbol{Y}}$ a* continuous random vector*, such that:*

- *i) $\boldsymbol{y}$ is supported on $\mathcal{Y}$,*
- *ii) for any $\boldsymbol{y} \in \mathcal{Y}$, both conditional distributions $p_{\text{neg}}(\cdot|\boldsymbol{y}), p_{\text{pos}}(\cdot|\boldsymbol{y}) := p_{\boldsymbol{y}^+|\boldsymbol{y}}(\cdot|\boldsymbol{y})$ are supported on $\mathcal{Y}$.*

Then an arbitrary measurable function $u : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a global minimum of $\mathcal{L}_g$ (amongst all measurable u's) if and only if there exists a measurable function $c : \mathcal{Y} \to \mathbb{R}$ such that the following holds almost everywhere (w.r.t. any continuous measure of $\mathcal{Y}$):

$$u(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = c(\boldsymbol{y}) + \ln \frac{p_{\text{pos}}(\tilde{\boldsymbol{y}}|\boldsymbol{y})}{p_{\text{neg}}(\tilde{\boldsymbol{y}}|\boldsymbol{y})}. \tag{7}$$

This statement has already been proven by Matthes et al. [39], however we provide an alternative argument. The key ideas have also emerged in Gutmann and Hyvärinen [23], Ma and Collins [37], Sohn [52].

To prove the theorem, we rewrite the optimization objective as the classification objective of discriminating the positive pair from all the negative pairs, given an anchor point. More specifically, all the pairs $\boldsymbol{y}^+, \boldsymbol{y}_1^-, \ldots, \boldsymbol{y}_M^-$ are randomly shuffled, and, then, the task is to estimate the index K of the positive pair from the new sequence, $\tilde{\boldsymbol{y}}_0, \ldots, \tilde{\boldsymbol{y}}_M$, given the unchanged anchor $\boldsymbol{y}$. The optimization objective is a categorical cross-entropy (CE) with a predictor of the form (being the output of a softmax function, the predictor's output is normalized to a finite probability distribution):

$$\left[\frac{e^{u(\boldsymbol{y}, \tilde{\boldsymbol{y}}_i)}}{\sum_j e^{u(\boldsymbol{y}, \tilde{\boldsymbol{y}}_j)}} \middle| \, i \in \{0, 1, \ldots, M\} \right]. \tag{8}$$

We formalize the instance discrimination task as follows:

Definition C.2 (Instance Discrimination Data) *The* instance discrimination data $(\boldsymbol{y}, \tilde{\boldsymbol{y}}_0, \ldots, \tilde{\boldsymbol{y}}_M; K)$ *corresponding to a general CL problem $(\mathcal{Y}, p_{\boldsymbol{Y}}, \mathcal{U}, \mathcal{L}_g)$ is defined by the following process:*

Step 1) $\boldsymbol{y} \sim p_{\boldsymbol{y}}$
Step 2) $K \sim Uni(\{0, 1, \ldots, M\})$
Step 3) $\tilde{\boldsymbol{y}}_K \sim p_{\text{pos}}(\cdot|\boldsymbol{y})$
Step 4) $\forall j \in \{\, 0, 1, \ldots, K-1, K+1, \ldots, M \,\} : \tilde{\boldsymbol{y}}_j \sim p_{\text{neg}}(\cdot|\boldsymbol{y})$.

Remark C.3 *The joint data distribution of the instance discrimination data is:*

$$p(\boldsymbol{y}, \tilde{\boldsymbol{y}}_0, \ldots, \tilde{\boldsymbol{y}}_M, K = i) = p_{\boldsymbol{y}}(\boldsymbol{y}) \, \mathbb{P}(K = i) \, p_{\text{pos}}(\tilde{\boldsymbol{y}}_i|\boldsymbol{y}) \prod_{0 \le j \le M, \, j \ne i} p_{\text{neg}}(\tilde{\boldsymbol{y}}_j|\boldsymbol{y}). \tag{9}$$

Definition C.3 (Instance Discrimination Problem) *Let the instance discrimination data $(\boldsymbol{y}, \tilde{\boldsymbol{y}}_0, \ldots, \tilde{\boldsymbol{y}}_M; K)$ correspond to a general CL problem $(\mathcal{Y}, p_{\boldsymbol{Y}}, \mathcal{U}, \mathcal{L}_g)$ (Defn. C.2) that also satisfies the assumptions of Thm. C.1. We define the* instance discrimination problem *as:*

$$\min_{u \in \mathcal{U}} \quad \mathcal{L}^{id}(u) := \mathbb{E}\left[-\sum_{i=0}^{M} \delta_{K=i} \ln \frac{e^{u(\boldsymbol{y}, \tilde{\boldsymbol{y}}_i)}}{\sum_{j=0}^{M} e^{u(\boldsymbol{y}, \tilde{\boldsymbol{y}}_j)}} \right], \tag{10}$$

where δ_E is the indicator variable of event E.

Remark C.4 *When conditioned on $K = k$, the data distribution Eq. (9) can be slightly simplified to:*

$$p(\boldsymbol{y}, \tilde{\boldsymbol{y}}_0, \ldots, \tilde{\boldsymbol{y}}_M | K = k) = p_{\boldsymbol{y}}(\boldsymbol{y}) \, p_{\text{pos}}(\tilde{\boldsymbol{y}}_k|\boldsymbol{y}) \prod_{0 \le j \le M, \, j \ne k} p_{\text{neg}}(\tilde{\boldsymbol{y}}_j|\boldsymbol{y}). \tag{11}$$

Comparing Eq. (11) to Eq. (6) in Rem. C.2, this distribution acts as if $\tilde{\boldsymbol{y}}_k$ (or $\tilde{\boldsymbol{y}}_{j \ne k}$) was a positive pair (or negative pairs) in a contrastive learning problem w.r.t. anchor $\boldsymbol{y}$.

Lemma C.1 *Under the assumptions of Defn. C.3 we have $\mathcal{L}_g(u) = \mathcal{L}^{id}(u)$, for any $u \in \mathcal{U}$.*

Proof First, using the law of total expectation $\mathbb{E}[A] = \mathbb{E}[\mathbb{E}[A|B]]$, we expand $\mathcal{L}^{id}$ w.r.t. K:

$$\mathcal{L}^{id}(u) = \mathbb{E}_K\left[\mathbb{E}\left[-\sum_{i=0}^{M}\delta_{K=i}\ln\frac{e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_i)}}{\sum_{j=0}^{M}e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_j)}}\middle|K\right]\right] \tag{12}$$

$$= \frac{1}{M+1}\sum_{k=0}^{M}\mathbb{E}\left[-\sum_{i=0}^{M}\delta_{K=i}\ln\frac{e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_i)}}{\sum_{j=0}^{M}e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_j)}}\middle|K=k\right]. \tag{13}$$

However, for the individual summands:

$$\mathbb{E}\left[-\sum_{i=0}^{M}\delta_{K=i}\ln\frac{e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_i)}}{\sum_{j=0}^{M}e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_j)}}\middle|K=k\right] = \mathbb{E}\left[-\ln\frac{e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_k)}}{\sum_{j=0}^{M}e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_j)}}\middle|K=k\right] \tag{14}$$

Considering Rem. C.4, the last term equals the contrastive objective $\mathcal{L}_g(u)$.
Therefore, substituting into Eq. (12) we get

$$\mathcal{L}^{id}(u) = \frac{1}{M+1}\sum_{k=0}^{M}\mathcal{L}_g(u) = \mathcal{L}_g(u). \tag{15}$$

■

After having rewritten the general contrastive objective, we are ready to prove Thm. C.1.

Proof [Proof of Thm. C.1] By taking the corresponding instance discrimination problem $\mathcal{L}^{id}$, it remains to compute the optima of $\mathcal{L}^{id}$. For this, we again expand $\mathcal{L}^{id}$ with the law of total expectation, but now w.r.t. $(\boldsymbol{y}, \tilde{\boldsymbol{y}}_0, \dots, \tilde{\boldsymbol{y}}_M)$:

$$\mathcal{L}^{id}(u) = \mathbb{E}\left[\mathbb{E}\left[-\sum_{i=0}^{M}\delta_{K=i}\ln\frac{e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_i)}}{\sum_{j=0}^{M}e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_j)}}\middle|\boldsymbol{y},\tilde{\boldsymbol{y}}_0,\dots,\tilde{\boldsymbol{y}}_M\right]\right] \tag{16}$$

$$= \mathbb{E}\left[-\sum_{i=0}^{M}\mathbb{E}\left[\delta_{K=i}\middle|\boldsymbol{y},\tilde{\boldsymbol{y}}_0,\dots,\tilde{\boldsymbol{y}}_M\right]\ln\frac{e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_i)}}{\sum_{j=0}^{M}e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_j)}}\right] \tag{17}$$

$$= \mathbb{E}\left[-\sum_{i=0}^{M}\mathbb{P}\Big(K=i\Big|\boldsymbol{y},\tilde{\boldsymbol{y}}_0,\dots,\tilde{\boldsymbol{y}}_M\Big)\ln\frac{e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_i)}}{\sum_{j=0}^{M}e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_j)}}\right], \tag{18}$$

where we used the properties that $\mathbb{E}\big[A\,f(B)\big|B\big] = f(B)\,\mathbb{E}\big[A\big|B\big]$ and that for any event E, $\mathbb{E}[\delta_E] = \mathbb{P}(E)$.

Let us denote $\tilde{\boldsymbol{Y}} = (\boldsymbol{y}, \tilde{\boldsymbol{y}}_0, \dots, \tilde{\boldsymbol{y}}_M)$ and $\eta_i\big(\tilde{\boldsymbol{Y}}\big) := \mathbb{P}\big(K = i\big|\boldsymbol{y}, \tilde{\boldsymbol{y}}_0, \dots, \tilde{\boldsymbol{y}}_M\big)$, the quantity $\boldsymbol{\eta}\big(\tilde{\boldsymbol{Y}}\big) = (\eta_0, \dots, \eta_M)$ is the ground-truth conditional distribution of index K. Denoting $F_i\big(\tilde{\boldsymbol{Y}}\big) = e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_i)}/(\sum_{j=0}^{M}e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_j)})$, the quantity $\boldsymbol{F}\big(\tilde{\boldsymbol{Y}}\big) = (F_0, \dots, F_M)$ is the inferred conditional distribution of index K. Then,

$$\mathcal{L}^{id}(u) = \mathbb{E}_{\tilde{\boldsymbol{Y}}}\Big[-\sum_{i=0}^{M}\eta_i(\tilde{\boldsymbol{Y}})\ln F_i(\tilde{\boldsymbol{Y}})\Big] \tag{19}$$

$$= \mathbb{E}_{\tilde{\boldsymbol{Y}}}\Big[\sum_{i=0}^{M}\eta_i(\tilde{\boldsymbol{Y}})\ln\frac{\eta_i(\tilde{\boldsymbol{Y}})}{F_i(\tilde{\boldsymbol{Y}})}\Big] + \mathbb{E}_{\tilde{\boldsymbol{Y}}}\Big[-\sum_{i=0}^{M}\eta_i(\tilde{\boldsymbol{Y}})\ln\eta_i(\tilde{\boldsymbol{Y}})\Big] \tag{20}$$

$$= \mathbb{E}_{\tilde{\boldsymbol{Y}}}\Big[KL\Big(\boldsymbol{\eta}(\tilde{\boldsymbol{Y}})\Big\|\boldsymbol{F}(\tilde{\boldsymbol{Y}})\Big)\Big] + \mathbb{E}_{\tilde{\boldsymbol{Y}}}\Big[H\Big(\boldsymbol{\eta}(\tilde{\boldsymbol{Y}})\Big)\Big], \tag{21}$$

where KL and H denote the Kullback-Leibler Divergence (KL) and entropy of the finite distributions over indices $\{0, 1, \dots, M\}$, provided as arguments. Due to the non-negativity of the KL, we can lower bound $\mathcal{L}_g(u)$:

$$\mathcal{L}_g(u) = \mathcal{L}^{id}(u) \geq \mathbb{E}_{\tilde{\boldsymbol{Y}}}\Big[H\Big(\boldsymbol{\eta}(\tilde{\boldsymbol{Y}})\Big)\Big], \tag{22}$$

which is independent of the choice of the function u. Equality is attained if and only if the KL vanishes, which occurs if and only if the true and inferred conditional distribution of index K coincide almost everywhere w.r.t. $\tilde{\boldsymbol{Y}} \sim p(\boldsymbol{y}, \tilde{\boldsymbol{y}}_0, \dots, \tilde{\boldsymbol{y}}_M)$. The statement is in terms of the indices, since the instance discrimination task is about classifying which index belongs to the positive pair. Hence, for any i and almost everywhere w.r.t. $\tilde{\boldsymbol{Y}}$:

$$\frac{e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_i)}}{\sum_{j=0}^{M} e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_j)}} = F_i(\tilde{\boldsymbol{Y}}) = \eta_i(\tilde{\boldsymbol{Y}}) = \mathbb{P}\big(K = i\big|\boldsymbol{y}, \tilde{\boldsymbol{y}}_0, \dots, \tilde{\boldsymbol{y}}_M\big). \tag{23}$$

To rewrite the last expression, we use assumptions *i)* and *ii)* and the uniformity of K (step *Step 2)*). Thus, the data distribution Eq. (9) of the instance discrimination problem can be rewritten as

$$p(\boldsymbol{y}, K = i, \tilde{\boldsymbol{y}}_0, \dots, \tilde{\boldsymbol{y}}_M) = \frac{1}{M+1}\, p_{\boldsymbol{y}}(\boldsymbol{y})\, \frac{p_{\text{pos}}(\tilde{\boldsymbol{y}}_i|\boldsymbol{y})}{p_{\text{neg}}(\tilde{\boldsymbol{y}}_i|\boldsymbol{y})} \prod_{j=0}^{M} p_{\text{neg}}(\tilde{\boldsymbol{y}}_j|\boldsymbol{y}). \tag{24}$$

With this in mind, Eq. (23) becomes

$$\frac{e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_i)}}{\sum_{j=0}^{M} e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}}_j)}} = \mathbb{P}\big(K{=}i\big|\boldsymbol{y}, \tilde{\boldsymbol{y}}_0, \dots, \tilde{\boldsymbol{y}}_M\big) = \frac{p(\boldsymbol{y}, K{=}i, \tilde{\boldsymbol{y}}_0, \dots, \tilde{\boldsymbol{y}}_M)}{\sum_{j=0}^{M} p(\boldsymbol{y}, K{=}j, \tilde{\boldsymbol{y}}_0, \dots, \tilde{\boldsymbol{y}}_M)} = \frac{\frac{p_{\text{pos}}(\tilde{\boldsymbol{y}}_i|\boldsymbol{y})}{p_{\text{neg}}(\tilde{\boldsymbol{y}}_i|\boldsymbol{y})}}{\sum_{j=0}^{M} \frac{p_{\text{pos}}(\tilde{\boldsymbol{y}}_j|\boldsymbol{y})}{p_{\text{neg}}(\tilde{\boldsymbol{y}}_j|\boldsymbol{y})}}. \tag{25}$$

From this it can be seen that there exists a measurable, positive function $\tilde{c} : \mathcal{Y} \to \mathbb{R}$ such that

$$e^{u(\boldsymbol{y},\tilde{\boldsymbol{y}})} = \tilde{c}(\boldsymbol{y}) \cdot \frac{p_{\text{pos}}(\tilde{\boldsymbol{y}}|\boldsymbol{y})}{p_{\text{neg}}(\tilde{\boldsymbol{y}}|\boldsymbol{y})} \quad \text{holds almost everywhere (w.r.t. any continuous measure of } \mathcal{Y}). \tag{26}$$

By taking the logarithm we get the desired result. ■

Appendix D. Identifiability Theory

D.1. Identifiability of Anisotropic CL

Definition D.1 (Anisotropic data generating process, ADGP)
Let $\mathcal{Z} := \mathbb{S}^{d-1}$ be the $(d-1)$-dimensional unit hypersphere and let $\boldsymbol{g} : \mathcal{Z} \to \mathcal{X} \subseteq \mathbb{R}^D$ be an invertible and continuous function. Let $\boldsymbol{\Lambda} \in \mathbb{R}^{d \times d}$ be a positive definite diagonal matrix.

The following process is called an anisotropic data generating process (ADGP)*:*

Step 1° $\boldsymbol{z} \sim Uni(\mathcal{Z})$
Step 2° $\boldsymbol{z}^+ \sim p_{\boldsymbol{z}^+|\boldsymbol{z}}(\cdot|\boldsymbol{z})$*, where* $p_{\boldsymbol{z}^+|\boldsymbol{z}}(\boldsymbol{z}^+|\boldsymbol{z}) \propto e^{-\|\boldsymbol{z}^+ - \boldsymbol{z}\|^2_{\boldsymbol{\Lambda}}}$
Step 3° *for all* $j \in \{1, \dots, M\}, \boldsymbol{z}_j^- \sim Uni(\mathcal{Z})$
Step 4° $\boldsymbol{x}, \boldsymbol{x}^+, \{\boldsymbol{x}_j^-\} \xleftarrow{\boldsymbol{g}} \boldsymbol{z}, \boldsymbol{z}^+, \{\boldsymbol{z}_j^-\}$

The pair $(\boldsymbol{g}, \boldsymbol{\Lambda})$ is called the characterization *of the process. Sets $\mathcal{Z}, \mathcal{X}$ are called the* latent *and* observation spaces*, respectively.*

Remark D.1 *The random vector $(\boldsymbol{x}, \boldsymbol{x}^+, \{\boldsymbol{x}_j^-\})$ meets the assumptions of Defn. C.1, so a contrastive learning problem can be well-defined.*

Definition D.2 (Anistropic InfoNCE loss, AnInfoNCE) *Let an ADGP be characterized by $(\boldsymbol{g}, \boldsymbol{\Lambda})$ and let $(\boldsymbol{x}, \boldsymbol{x}^+, \{\boldsymbol{x}_j^-\})$ be the observed data (all coming from observation space $\mathcal{X} \subseteq \mathbb{R}^D$). Let us optimize the following objective amongst all positive definite (PD) diagonal matrices $\hat{\boldsymbol{\Lambda}}$ and all continuous encoders $\boldsymbol{f} : \mathbb{R}^D \to \mathcal{Z}$:*

$$\min_{\substack{\hat{\boldsymbol{\Lambda}} \text{ PD diag} \\ \boldsymbol{f} \text{ continuous}}} \mathcal{L}_{\text{AINCE}}(\boldsymbol{f}, \hat{\boldsymbol{\Lambda}}) := \mathbb{E}\left[-\ln \frac{e^{-\|\boldsymbol{f}(\boldsymbol{x}^+) - \boldsymbol{f}(\boldsymbol{x})\|^2_{\hat{\boldsymbol{\Lambda}}}}}{e^{-\|\boldsymbol{f}(\boldsymbol{x}^+) - \boldsymbol{f}(\boldsymbol{x})\|^2_{\hat{\boldsymbol{\Lambda}}}} + \sum_{j=1}^M e^{-\|\boldsymbol{f}(\boldsymbol{x}_j^-) - \boldsymbol{f}(\boldsymbol{x})\|^2_{\hat{\boldsymbol{\Lambda}}}}} \right]. \tag{27}$$

We call the objective $\mathcal{L}_{\text{AINCE}}$ the Anistropic InfoNCE (AnInfoNCE) loss.

Remark D.2 *The tuple $(\mathcal{X}, p(\boldsymbol{x}, \boldsymbol{x}^+, \{\boldsymbol{x}_j^-\}), \mathcal{U}, \mathcal{L}_{\text{AINCE}})$ is a special case of the general contrastive learning problem (Defn. C.1), where $\mathcal{U} = \{u \,|\, u(\boldsymbol{x}, \boldsymbol{x}^+) = -\|\boldsymbol{f}(\boldsymbol{x}^+) - \boldsymbol{f}(\boldsymbol{x})\|^2_{\hat{\boldsymbol{\Lambda}}}$ for any pair $(\boldsymbol{f}, \hat{\boldsymbol{\Lambda}})\}$ is the hypothesis class of similarity functions.*

Theorem 3.1b (Identifiability of Anisotropic CL) *Let an ADGP (Defn. D.1) be characterized by $(\boldsymbol{g}, \boldsymbol{\Lambda})$ with latent and observed data being $(\boldsymbol{z}, \boldsymbol{z}^+, \{\boldsymbol{z}_j^-\})$ and $(\boldsymbol{x}, \boldsymbol{x}^+, \{\boldsymbol{x}_j^-\})$. Let $\mathcal{L}_{\text{AINCE}}$ denote the AnInfoNCE loss (Defn. D.2).*

If a pair $(\boldsymbol{f}, \hat{\boldsymbol{\Lambda}})$ (globally) minimizes the $\mathcal{L}_{\text{AINCE}}$ loss, then:

i) $\hat{\boldsymbol{\Lambda}}$ *is equal to* $\boldsymbol{\Lambda}$ *up to a permutation of elements and*
ii) $\boldsymbol{f} \circ \boldsymbol{g}$ *is a block-orthogonal transformation, where each block acts on latents of equal weight* $\boldsymbol{\Lambda}_{ii}$.

In other words, latent $\boldsymbol{z}$ is identified up to a block-orthogonal transformation.

Proof [Proof of Thm. 3.1b] Our proof consists of two steps:

Step 1: Based on Appx. C, optimums of AnInfoNCE connect the density ratio to the similarity function u.
Step 2: Then, we solve the resulting functional equation.

Step 1: Connecting the density ratio to the similarity function. First, we are going rewrite the $\mathcal{L}_{\text{AINCE}}$ loss function into a form that favors the analysis of $\boldsymbol{f} \circ \boldsymbol{g}$. Plugging in $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{z})$ (*Step 4°* in Defn. D.1) into the $\mathcal{L}_{\text{AINCE}}$-loss:

$$\mathcal{L}_{\text{AINCE}}(\boldsymbol{f}, \hat{\boldsymbol{\Lambda}}) = \mathbb{E}\left[-\ln \frac{e^{-\|\boldsymbol{f}\circ\boldsymbol{g}(\boldsymbol{z}^+)-\boldsymbol{f}\circ\boldsymbol{g}(\boldsymbol{z})\|^2_{\hat{\boldsymbol{\Lambda}}}}}{e^{-\|\boldsymbol{f}\circ\boldsymbol{g}(\boldsymbol{z}^+)-\boldsymbol{f}\circ\boldsymbol{g}(\boldsymbol{z})\|^2_{\hat{\boldsymbol{\Lambda}}}} + \sum_{j=1}^{M} e^{-\|\boldsymbol{f}\circ\boldsymbol{g}(\boldsymbol{z}_j^-)-\boldsymbol{f}\circ\boldsymbol{g}(\boldsymbol{z})\|^2_{\hat{\boldsymbol{\Lambda}}}}} \right] =: \tilde{\mathcal{L}}(\boldsymbol{f} \circ \boldsymbol{g}, \hat{\boldsymbol{\Lambda}}), \tag{28}$$

with the optimization still over $\boldsymbol{f}$ (and $\hat{\boldsymbol{\Lambda}}$). However, as the generator $\boldsymbol{g}$ is continuously invertible on the *compact* set $\mathcal{Z}$, its inverse $\boldsymbol{g}^{-1}$ is automatically continuous as well. Therefore, any continuous function $\boldsymbol{h} : \mathcal{Z} \to \mathcal{Z}$ can take the role of $\boldsymbol{f} \circ \boldsymbol{g}$, by substituting $\boldsymbol{f} = \boldsymbol{h} \circ \boldsymbol{g}^{-1}$ continuous. Hence, minimizing $\mathcal{L}_{\text{AINCE}}(\boldsymbol{f}, \hat{\boldsymbol{\Lambda}})$ for $\boldsymbol{f}$ continuous is equivalent to minimizing the new, latent space loss $\tilde{\mathcal{L}}(\boldsymbol{h}, \hat{\boldsymbol{\Lambda}})$ for $\boldsymbol{h}$ continuous:

$$\min_{\substack{\hat{\boldsymbol{\Lambda}} \text{ PD diag} \\ \boldsymbol{f} \text{ cont}}} \mathcal{L}_{\text{AINCE}}(\boldsymbol{f}, \hat{\boldsymbol{\Lambda}}) = \min_{\substack{\hat{\boldsymbol{\Lambda}} \text{ PD diag} \\ \boldsymbol{h} \text{ cont}}} \tilde{\mathcal{L}}(\boldsymbol{h}, \hat{\boldsymbol{\Lambda}}). \tag{29}$$

The new hypothesis class is $\tilde{\mathcal{U}} = \{u \,|\, u(\boldsymbol{z}, \boldsymbol{z}^+) = -\|\boldsymbol{h}(\boldsymbol{z}^+) - \boldsymbol{h}(\boldsymbol{z})\|^2_{\hat{\boldsymbol{\Lambda}}} \text{ for any pair } (\boldsymbol{h}, \hat{\boldsymbol{\Lambda}})\}$.

We claim that $\tilde{\mathcal{L}}$ can attain its Bayes- or unconditional optimum (Thm. C.1), i.e., as if there were no constraints on the similarity function u. According to Thm. C.1, an arbitrary similarity function u minimizes $\tilde{\mathcal{L}}$ amongst all possible measurable u's (attaining the Bayes-optimum) if and only if for a suitable function c:

$$u(\boldsymbol{z}, \tilde{\boldsymbol{z}}) = c(\boldsymbol{z}) + \ln \frac{p_{\boldsymbol{z}^+|\boldsymbol{z}}(\tilde{\boldsymbol{z}}|\boldsymbol{z})}{p_{\boldsymbol{z}^-|\boldsymbol{z}}(\tilde{\boldsymbol{z}}|\boldsymbol{z})} \quad \text{holds almost everywhere on } \mathcal{Z}. \tag{30}$$

In our special case, u is restricted to take the form $u(\boldsymbol{z}, \tilde{\boldsymbol{z}}) = -\|\boldsymbol{h}(\tilde{\boldsymbol{z}}) - \boldsymbol{h}(\boldsymbol{z})\|^2_{\hat{\boldsymbol{\Lambda}}}$. Besides that, $\boldsymbol{z}_j^- \sim Uni(\mathcal{Z})$ with a constant density function and $p_{\boldsymbol{z}^-|\boldsymbol{z}} = p_{\boldsymbol{z}^-} \equiv \text{const}$. We also plug in the anisotropic positive conditional distribution $p_{\boldsymbol{z}^+|\boldsymbol{z}}(\tilde{\boldsymbol{z}}|\boldsymbol{z}) \propto e^{-\|\tilde{\boldsymbol{z}}-\boldsymbol{z}\|^2_{\boldsymbol{\Lambda}}}$. After merging every constant additive term (including normalization constants) into function c, we get the functional equation with unknowns $\boldsymbol{h}, \hat{\boldsymbol{\Lambda}}, c$:

$$\|\boldsymbol{h}(\tilde{\boldsymbol{z}}) - \boldsymbol{h}(\boldsymbol{z})\|^2_{\hat{\boldsymbol{\Lambda}}} = c(\boldsymbol{z}) + \|\tilde{\boldsymbol{z}} - \boldsymbol{z}\|^2_{\boldsymbol{\Lambda}} \quad \text{holds almost everywhere on } \mathcal{Z}. \tag{31}$$

Observe that the equation is trivially satisfied for $\boldsymbol{h} = id_{\mathcal{Z}}, \hat{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}, c \equiv 0$. This special case corresponds to having $\boldsymbol{f} = \boldsymbol{g}^{-1}$, i.e., we inverted the exact generator and also reconstructed $\boldsymbol{\Lambda}$. Consequently, our loss function attains its Bayes-minimum even in our restricted hypothesis class. Now leveraging again Thm. C.1 in the other direction, the above shows us that all minimizers are, in fact, solutions of Eq. (31).

Let's take an arbitrary solution $\boldsymbol{h}, \hat{\boldsymbol{\Lambda}}, c$. Noticeably, the single-variable, measurable function c is equal (almost everywhere) to a two-variable continuous function. This is only possible if the latter function, depending on $(\boldsymbol{z}, \tilde{\boldsymbol{z}})$ is a single-variable, continuous function of $\boldsymbol{z}$. In this case, swapping c to this exact function will also reveal a good solution and, therefore, it is only required to solve Eq. (31) for c continuous and on the full space $\mathcal{Z}$. By taking the special case of $\tilde{\boldsymbol{z}} \leftarrow \boldsymbol{z}$, we see that $c(\boldsymbol{z}) = 0$ follows.

Step 2: Solving the functional equation. What is left to solve is the following functional equation in terms of the variables $\boldsymbol{h}, \hat{\boldsymbol{\Lambda}}$:

$$\|\boldsymbol{h}(\tilde{\boldsymbol{z}}) - \boldsymbol{h}(\boldsymbol{z})\|_{\hat{\boldsymbol{\Lambda}}}^2 = \|\tilde{\boldsymbol{z}} - \boldsymbol{z}\|_{\boldsymbol{\Lambda}}^2 \quad \text{holds for any } \tilde{\boldsymbol{z}}, \boldsymbol{z} \text{ on } \mathcal{Z}. \tag{32}$$

For this, let $\boldsymbol{z}_0 \in \mathcal{Z}$ be fixed and let us define the following functions:

$$\boldsymbol{a}(\boldsymbol{z}) = \hat{\boldsymbol{\Lambda}}^{1/2}\big(\boldsymbol{h}(\boldsymbol{z}) - \boldsymbol{h}(\boldsymbol{z}_0)\big) \quad \text{and} \quad \boldsymbol{b}(\boldsymbol{z}) = \boldsymbol{\Lambda}^{1/2}(\boldsymbol{z} - \boldsymbol{z}_0). \tag{33}$$

In this case, the following properties are easily verified:

$$\|\boldsymbol{a}(\tilde{\boldsymbol{z}}) - \boldsymbol{a}(\boldsymbol{z})\|^2 = \|\boldsymbol{b}(\tilde{\boldsymbol{z}}) - \boldsymbol{b}(\boldsymbol{z})\|^2, \tag{34}$$

$$\boldsymbol{a}(\boldsymbol{z}_0) = \boldsymbol{b}(\boldsymbol{z}_0) = 0. \tag{35}$$

By substituting $\tilde{\boldsymbol{z}} \leftarrow \boldsymbol{z}_0$ into Eq. (34) and using Eq. (35), we get that

$$\|\boldsymbol{a}(\boldsymbol{z})\|^2 = \|\boldsymbol{b}(\boldsymbol{z})\|^2 \text{ holds for any } \boldsymbol{z} \in \mathcal{Z}. \tag{36}$$

After expanding Eq. (34) and using Eq. (36) twice, we get that, similarly to the distance and norm, the scalar product is equivalently transformed by $\boldsymbol{a}$ and $\boldsymbol{b}$:

$$\langle \boldsymbol{a}(\tilde{\boldsymbol{z}}), \boldsymbol{a}(\boldsymbol{z})\rangle = \langle \boldsymbol{b}(\tilde{\boldsymbol{z}}), \boldsymbol{b}(\boldsymbol{z})\rangle \text{ for any } \tilde{\boldsymbol{z}}, \boldsymbol{z} \in \mathcal{Z}. \tag{37}$$

From this, it can be proven that there exists an orthogonal linear transformation $\boldsymbol{O}$ of $\mathbb{R}^d$ such that:

$$\boldsymbol{a}(\boldsymbol{z}) = \boldsymbol{O}\boldsymbol{b}(\boldsymbol{z}) \text{ holds for any } \boldsymbol{z} \in \mathcal{Z}. \tag{38}$$

To see this, we first prove that the following mapping is well-defined on the linear span of $\boldsymbol{b}(\mathcal{Z})$:

$$\boldsymbol{O} : \sum_{i=1}^n \alpha_i \boldsymbol{b}(\boldsymbol{z}_i) \mapsto \sum_{i=1}^n \alpha_i \boldsymbol{a}(\boldsymbol{z}_i) \tag{39}$$

Assume there exist two equal expansions (with potentially zero coefficients):

$$\sum_{i=1}^n \alpha_i \boldsymbol{b}(\boldsymbol{z}_i) = \sum_{i=1}^n \beta_i \boldsymbol{b}(\boldsymbol{z}_i). \tag{40}$$

In this case the difference and the distance square vanish:

$$\left\|\sum_{i=1}^n (\alpha_i - \beta_i)\boldsymbol{b}(\boldsymbol{z}_i)\right\|^2 = 0. \tag{41}$$

However, due to the equivalent transformation of the scalar product (Eq. (37)), after the expansion of Eq. (41) we can essentially change every occurrence of $\boldsymbol{b}$ to $\boldsymbol{a}$ and receive that:

$$\left\|\sum_{i=1}^n (\alpha_i - \beta_i)\boldsymbol{a}(\boldsymbol{z}_i)\right\|^2 = 0, \tag{42}$$

which may hold if and only if

$$\sum_{i=1}^{n} \alpha_i \boldsymbol{a}(\boldsymbol{z}_i) = \sum_{i=1}^{n} \beta_i \boldsymbol{a}(\boldsymbol{z}_i), \tag{43}$$

proving that $\boldsymbol{O}$ in Eq. (39) is well-defined on the linear span of $\boldsymbol{b}(\mathcal{Z})$.

Secondly, the facts that $\boldsymbol{O}$ is linear and that $\boldsymbol{a}(\boldsymbol{z}) = \boldsymbol{O}\boldsymbol{b}(\boldsymbol{z})$ hold trivially. We also see that the image of $\boldsymbol{b}$, i.e. $\boldsymbol{b}(\mathcal{Z}) = \boldsymbol{b}(\mathbb{S}^{d-1})$ is full dimensional and hence $\boldsymbol{O}$ is defined on the entire $\mathbb{R}^d$. The orthogonality is a direct consequence of the scalar product preservation property (Eq. (37)) and the definition of $\boldsymbol{O}$ (Eq. (39)).

Consequently, we have proven that there exists an orthogonal linear transformation $\boldsymbol{O}$ of $\mathbb{R}^d$ such that:

$$\hat{\boldsymbol{\Lambda}}^{1/2}(\boldsymbol{h}(\boldsymbol{z}) - \boldsymbol{h}(\boldsymbol{z}_0)) = \boldsymbol{O}\boldsymbol{\Lambda}^{1/2}(\boldsymbol{z} - \boldsymbol{z}_0) \text{ holds for any } \boldsymbol{z} \in \mathcal{Z}. \tag{44}$$

As $\hat{\boldsymbol{\Lambda}}$ is positive definite and invertible, it follows (after merging constant terms containing $\boldsymbol{z}_0$), that:

$$\boldsymbol{h}(\boldsymbol{z}) = \hat{\boldsymbol{\Lambda}}^{-1/2}\boldsymbol{O}\boldsymbol{\Lambda}^{1/2}\boldsymbol{z} + \boldsymbol{c} \text{ holds for any } \boldsymbol{z} \in \mathcal{Z}, \text{ for some constant } c. \tag{45}$$

Let us denote $\boldsymbol{H} = \hat{\boldsymbol{\Lambda}}^{-1/2}\boldsymbol{O}\boldsymbol{\Lambda}^{1/2}$. To complete the proof, we have to show that $\boldsymbol{c} = 0$ and $\boldsymbol{H}$ is orthogonal. For this, we use the fact that $\boldsymbol{h}$ is a transformation of the unit hypersphere $\mathcal{Z} = \mathbb{S}^{d-1}$:

$$\langle \boldsymbol{H}\boldsymbol{z}, \boldsymbol{c} \rangle = \frac{1}{4}\Big(\|\boldsymbol{H}\boldsymbol{z} + \boldsymbol{c}\|^2 - \| -\boldsymbol{H}\boldsymbol{z} + \boldsymbol{c}\|^2 \Big) \tag{46}$$

$$= \frac{1}{4}\Big(\|\boldsymbol{h}(\boldsymbol{z})\|^2 - \|\boldsymbol{h}(-\boldsymbol{z})\|^2 \Big) = \frac{1}{4}(1-1) = 0, \text{ for any } \boldsymbol{z} \in \mathcal{Z}. \tag{47}$$

As $\boldsymbol{H}$ is invertible and, thus, with a full range of $\mathbb{R}^d$, $\boldsymbol{c} = 0$ is concluded. Therefore, $\boldsymbol{h}$ is linear. As unit vectors are mapped to unit vectors, $\boldsymbol{h}$ is norm-preserving and, thus, $\boldsymbol{h} = \boldsymbol{f} \circ \boldsymbol{g}$ is orthogonal.

Now, rewriting $\boldsymbol{H} = \hat{\boldsymbol{\Lambda}}^{-1/2}\boldsymbol{O}\boldsymbol{\Lambda}^{1/2}$ gives us:

$$\boldsymbol{O}\boldsymbol{\Lambda}^{1/2} = \hat{\boldsymbol{\Lambda}}^{1/2}\boldsymbol{H} = \boldsymbol{H}\big(\boldsymbol{H}^\top \hat{\boldsymbol{\Lambda}}^{1/2}\boldsymbol{H}\big), \tag{48}$$

where the first and last formulas provide polar decompositions of the same operator. As the polar decomposition of invertible operators is unique, we conclude that $\boldsymbol{O} = \boldsymbol{H}$, and

$$\boldsymbol{\Lambda}^{1/2} = \boldsymbol{H}^\top \hat{\boldsymbol{\Lambda}}^{1/2}\boldsymbol{H}. \tag{49}$$

After squaring both sides, we get:

$$\boldsymbol{\Lambda} = \boldsymbol{H}^\top \hat{\boldsymbol{\Lambda}}\boldsymbol{H}. \tag{50}$$

Then the right-hand side of Eq. (50) is the eigendecomposition of $\boldsymbol{\Lambda}$. Due to the uniqueness of eigenvalues, there must exist a permutation $\mathbf{P}_{\boldsymbol{h}}$ such that $\hat{\boldsymbol{\Lambda}} = \mathbf{P}_{\boldsymbol{h}}\boldsymbol{\Lambda}\mathbf{P}_{\boldsymbol{h}}^\top$. Then

$$\boldsymbol{\Lambda} = \boldsymbol{H}^\top \mathbf{P}_{\boldsymbol{h}}\boldsymbol{\Lambda}\mathbf{P}_{\boldsymbol{h}}^\top \boldsymbol{H} \tag{51}$$

$$\Leftrightarrow \mathbf{P}_{\boldsymbol{h}}^\top \boldsymbol{H}\boldsymbol{\Lambda} = \boldsymbol{\Lambda}\mathbf{P}_{\boldsymbol{h}}^\top \boldsymbol{H}. \tag{52}$$

In other words, $\mathbf{P}_{\boldsymbol{h}}^\top \boldsymbol{H}$ and $\boldsymbol{\Lambda}$ commute. Then $\mathbf{P}_{\boldsymbol{h}}^\top \boldsymbol{H}$ must be a diagonal matrix if all values on the diagonal of $\boldsymbol{\Lambda}$ are distinct. Should some values be the same, then the corresponding submatrix of $\mathbf{P}_{\boldsymbol{h}}^\top \boldsymbol{H}$ is orthogonal.

To see this more formally, we define $\mathbf{O} := \mathbf{P}_{\boldsymbol{h}}^\top \boldsymbol{H}$. Then, for all i, j,

$$\sum_k \mathrm{O}_{ik}\lambda_k\delta_{kj} = \sum_k \lambda_i\delta_{ik}\mathrm{O}_{kj}, \tag{53}$$

$$\Leftrightarrow \mathrm{O}_{ij}\lambda_j = \lambda_i\mathrm{O}_{ij}, \tag{54}$$

$$\Leftrightarrow \mathrm{O}_{ij}(\lambda_j - \lambda_i) = 0. \tag{55}$$

Hence, for distinct eigenvalues, all off-diagonal entries of $\mathbf{O}$ are zero. Within the space of equivalent eigenvalues, $\mathbf{O}$ is only constrained by the requirement to be orthogonal.

From $\mathbf{O} = \mathbf{P}_{\boldsymbol{h}}^\top \boldsymbol{H}$ we then know that $\boldsymbol{H}$ is — up to permutation — an orthogonal block-diagonal matrix where the size of each block corresponds to the multiplicity of the values on the diagonal of $\boldsymbol{\Lambda}$.

■

D.2. Including hard negative sampling

Definition D.3 (Anisotropic DGP with hard negatives (HN)) *Let $\boldsymbol{g} : \mathcal{Z} := \mathbb{S}^{d-1} \to \mathcal{X} \subseteq \mathbb{R}^D$ be an invertible and continuous function. Let $\boldsymbol{\Lambda}^+, \boldsymbol{\Lambda}^- \in \mathbb{R}^{d\times d}$ be positive definite diagonal matrices such that $\boldsymbol{\Lambda}^+_{ii} > \boldsymbol{\Lambda}^-_{ii}$ (or equivalently, $\boldsymbol{\Lambda}^+ - \boldsymbol{\Lambda}^-$ is still PD).*

The following process is called an anisotropic DGP with hard negatives (HN)*, characterized by the triple $(\boldsymbol{g}, \boldsymbol{\Lambda}^+, \boldsymbol{\Lambda}^-)$:*

Step 1) $\boldsymbol{z} \sim Uni(\mathcal{Z})$

Step 2) $\boldsymbol{z}^+ \sim p_{\boldsymbol{z}^+|\boldsymbol{z}}(\cdot|\boldsymbol{z})$*, where* $p_{\boldsymbol{z}^+|\boldsymbol{z}}(\boldsymbol{z}^+|\boldsymbol{z}) \propto e^{-\|\boldsymbol{z}^+-\boldsymbol{z}\|^2_{\boldsymbol{\Lambda}^+}}$

Step 3) for all $j \in \{1, \dots, M\}, \boldsymbol{z}_j^- \sim p_{\boldsymbol{z}^-|\boldsymbol{z}}(\cdot|\boldsymbol{z})$*, where* $p_{\boldsymbol{z}^-|\boldsymbol{z}}(\boldsymbol{z}^-|\boldsymbol{z}) \propto e^{-\|\boldsymbol{z}^--\boldsymbol{z}\|^2_{\boldsymbol{\Lambda}^-}}$

Step 4) $\boldsymbol{x}, \boldsymbol{x}^+, \{\boldsymbol{x}_j^-\} \xleftarrow{\boldsymbol{g}} \boldsymbol{z}, \boldsymbol{z}^+, \{\boldsymbol{z}_j^-\}$.

Corollary B.1b (Identifiability of anisotropic DGPs with HN) *Let an ADGP with HN (Defn. D.3) be characterized by $(\boldsymbol{g}, \boldsymbol{\Lambda}^+, \boldsymbol{\Lambda}^-)$ with latent and observed data being $(\boldsymbol{z}, \boldsymbol{z}^+, \{\boldsymbol{z}_j^-\})$ and $(\boldsymbol{x}, \boldsymbol{x}^+, \{\boldsymbol{x}_j^-\})$. Let $\mathcal{L}_{\text{AINCE}}$ denote the AnInfoNCE loss (Defn. D.2), evaluated w.r.t. $(\boldsymbol{x}, \boldsymbol{x}^+, \{\boldsymbol{x}_j^-\})$.*

If a pair $(\boldsymbol{f}, \hat{\boldsymbol{\Lambda}})$ (globally) minimizes the $\mathcal{L}_{\text{AINCE}}$ loss, then:

i) $\hat{\boldsymbol{\Lambda}}$ is equal to $\boldsymbol{\Lambda}^+ - \boldsymbol{\Lambda}^-$ up to a permutation of elements and

ii) $\boldsymbol{f} \circ \boldsymbol{g}$ is an orthogonal transformation, or latent $\boldsymbol{z}$ is identified up to an orthogonal transformation.

Proof We are following the steps of the Proof of Thm. 3.1b.

First, we again rewrite the $\mathcal{L}_{\text{AINCE}}$ by plugging in $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{z})$ and receiving $\mathcal{L}_{\text{AINCE}}(\boldsymbol{f}, \hat{\boldsymbol{\Lambda}}) = \tilde{\mathcal{L}}(\boldsymbol{f} \circ \boldsymbol{g}, \hat{\boldsymbol{\Lambda}})$, with optimization over $\boldsymbol{f}$ (and $\hat{\boldsymbol{\Lambda}}$). As $\boldsymbol{h} = \boldsymbol{f} \circ \boldsymbol{g}$ attains all possible continuous functions, we may optimize $\tilde{\mathcal{L}}(\boldsymbol{h}, \hat{\boldsymbol{\Lambda}})$. The hypothesis class becomes, again, $\tilde{\mathcal{U}} = \{u \,|\, u(\boldsymbol{z}, \boldsymbol{z}^+) = -\|\boldsymbol{h}(\boldsymbol{z}^+) - \boldsymbol{h}(\boldsymbol{z})\|^2_{\hat{\boldsymbol{\Lambda}}}$ for any pair $(\boldsymbol{h}, \hat{\boldsymbol{\Lambda}})\}$.

Secondly, according to Thm. C.1, an arbitrary similarity function u minimizes $\tilde{\mathcal{L}}$ amongst all possible measurable u's (attaining the Bayes-optimum) if and only if for a suitable function c:

$$u(\boldsymbol{z}, \tilde{\boldsymbol{z}}) = c(\boldsymbol{z}) + \ln \frac{p_{\boldsymbol{z}^+|\boldsymbol{z}}(\tilde{\boldsymbol{z}}|\boldsymbol{z})}{p_{\boldsymbol{z}^-|\boldsymbol{z}}(\tilde{\boldsymbol{z}}|\boldsymbol{z})} \quad \text{holds almost everywhere on } \mathcal{Z}. \tag{56}$$

We are about to plug in all our constraints. The major difference compared to the Proof of Thm. 3.1b is that besides the positive conditional distribution being $p_{\boldsymbol{z}^+|\boldsymbol{z}}(\tilde{\boldsymbol{z}}|\boldsymbol{z}) \propto e^{-\|\tilde{\boldsymbol{z}}-\boldsymbol{z}\|^2_{\boldsymbol{\Lambda}^+}}$, the negative conditional also changes to $p_{\boldsymbol{z}^-|\boldsymbol{z}}(\tilde{\boldsymbol{z}}|\boldsymbol{z}) \propto e^{-\|\tilde{\boldsymbol{z}}-\boldsymbol{z}\|^2_{\boldsymbol{\Lambda}^-}}$. After plugging in the restricted form of u and merging every constant normalizing term into function c, Eq. (56) becomes:

$$\begin{aligned}
-\|\boldsymbol{h}(\tilde{\boldsymbol{z}}) - \boldsymbol{h}(\boldsymbol{z})\|^2_{\hat{\boldsymbol{\Lambda}}} &= c(\boldsymbol{z}) + \ln \frac{e^{-\|\tilde{\boldsymbol{z}}-\boldsymbol{z}\|^2_{\boldsymbol{\Lambda}^+}}}{e^{-\|\tilde{\boldsymbol{z}}-\boldsymbol{z}\|^2_{\boldsymbol{\Lambda}^-}}} && (57)\\
&= c(\boldsymbol{z}) - \|\tilde{\boldsymbol{z}} - \boldsymbol{z}\|^2_{\boldsymbol{\Lambda}^+} + \|\tilde{\boldsymbol{z}} - \boldsymbol{z}\|^2_{\boldsymbol{\Lambda}^-} && (58)\\
&= c(\boldsymbol{z}) - \langle \tilde{\boldsymbol{z}} - \boldsymbol{z}, \boldsymbol{\Lambda}^+(\tilde{\boldsymbol{z}} - \boldsymbol{z})\rangle + \langle \tilde{\boldsymbol{z}} - \boldsymbol{z}, \boldsymbol{\Lambda}^-(\tilde{\boldsymbol{z}} - \boldsymbol{z})\rangle && (59)\\
&= c(\boldsymbol{z}) - \langle \tilde{\boldsymbol{z}} - \boldsymbol{z}, (\boldsymbol{\Lambda}^+ - \boldsymbol{\Lambda}^-)(\tilde{\boldsymbol{z}} - \boldsymbol{z})\rangle = c(\boldsymbol{z}) - \|\tilde{\boldsymbol{z}} - \boldsymbol{z}\|^2_{\boldsymbol{\Lambda}^+ - \boldsymbol{\Lambda}^-}. && (60)
\end{aligned}$$

Hence, we get the functional equation with unknowns $\boldsymbol{h}, \hat{\boldsymbol{\Lambda}}, c$:

$$\|\boldsymbol{h}(\tilde{\boldsymbol{z}}) - \boldsymbol{h}(\boldsymbol{z})\|^2_{\hat{\boldsymbol{\Lambda}}} = c(\boldsymbol{z}) + \|\tilde{\boldsymbol{z}} - \boldsymbol{z}\|^2_{\boldsymbol{\Lambda}^+ - \boldsymbol{\Lambda}^-} \quad \text{holds almost everywhere on } \mathcal{Z}. \tag{61}$$

We conclude our proof by continuing the Proof of Thm. 3.1b by substituting $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^+ - \boldsymbol{\Lambda}^-$ PD.

■

Remark D.3 (Subsuming the uniform marginal in the negative conditional with $\boldsymbol{\Lambda}^- = 0$) *When using a negative conditional in the form of Eq. (2), $\boldsymbol{\Lambda}^- = 0$ accounts for a uniform marginal.*

D.3. Identifiability of a loss ensemble

In this setting the observations $\boldsymbol{x}$ come from k different DGPs, with a *shared* generator $\boldsymbol{g}$ and separate concentration parameters $\{\boldsymbol{\Lambda}_i\}_{i=1}^k$. That is, this setting is akin to works investigating identifiability in multiple environments [21, 27, 43] and reflects the practice in CL of optimizing multiple loss components with different data augmentations [17, 67, 71].

Corollary B.2b (Identifiability of ensemble AnInfoNCE)
Assume k DGPs, each satisfying Assum. 1. Assume that they share $\boldsymbol{g}$, but have different concentration parameters $\{\boldsymbol{\Lambda}^i\}$, for $1 \leq i \leq k$. Assume a shared encoder $\boldsymbol{f}$ and k learnable $\hat{\boldsymbol{\Lambda}}^i$ parameters. Then, the tuple $(\boldsymbol{f}, \{\hat{\boldsymbol{\Lambda}}^i\})$ minimizing the ensemble loss $\sum_i^k \mathcal{L}_{\text{AINCE}}(\boldsymbol{f}, \hat{\boldsymbol{\Lambda}}^i)$ identifies the latents up to a block-orthogonal transformation.

Proof Provided in the main text. ■

Remark D.4 (Sufficiency condition for identifiability up to permutations for the ensemble AnInfoNCE loss) *Ensemble AnInfoNCE is identifiable up to permutations if, under the conditions of Cor. B.2, there are sufficiently many DGPs such that for all distinct latent factors z_k, z_l, there exists a DGP with concentration parameter $\boldsymbol{\Lambda}^i$ such that $\boldsymbol{\Lambda}^i_{kk} \neq \boldsymbol{\Lambda}^i_{ll}$.*

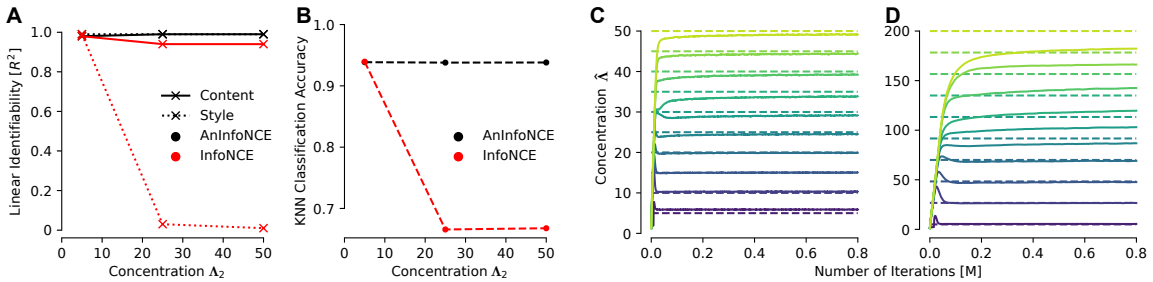

Figure 3: **Behavior of AnInfoNCE on MNIST. A:** Linear identifiability (R^2) scores for an encoder trained on VAE-generated MNIST samples, when varying Λ_2 and keeping $\Lambda_1 = 5$ for the positive conditional. Style dimensions collapse ($R^2 = 0$) for regular InfoNCE. **B:** KNN-accuracy evaluated on the regular MNIST dataset using the encoders trained as described in **A**. KNN accuracy degrades when style dimensions are lost. **C & D:** The evolution of learned $\hat{\mathbf{\Lambda}}$ during training. We set the diagonal entries of $\mathbf{\Lambda}$ to ten different values by linearly interpolating between 5 and 50 (**C**) or 5 and 200 (**D**). The ground-truth $\mathbf{\Lambda}$ values are indicated by dashed lines. The learned $\hat{\mathbf{\Lambda}}$ are shown in the corresponding colors as solid lines.

Appendix E. Experimental details

Appendix F. VAE Experiments on MNIST

Preliminaries. We now move closer to actual images by replacing the MLP in our generative model with a ten-dimensional Variational Autoencoder (VAE; 30) trained on MNIST [15]. This setup is beneficial as we have valid images while fully controlling the DGP. Thus, we still sample ground-truth latents, which we then pass through the VAE decoder to generate observations to train on. Both the encoder and the decoder are fully connected three-layer MLPs with Leaky ReLU activations. Generated samples for different concentration parameters $\mathbf{\Lambda}$ are shown in Appx. I.5, Fig. 13.

Results. We start with the same anisotropic ground-truth $\mathbf{\Lambda}$ as in Sec. 4.1. When training an encoder on MNIST samples with AnInfoNCE, we observe perfect linear identifiability scores (R^2) for all latents, whereas training with the standard InfoNCE loss yields zero R^2 scores for the style latents (Fig. 3**A**). The identifiability scores correlate with KNN classification accuracy, which is stable across a wide range of $\mathbf{\Lambda}$-values when training with AnInfoNCE, but degrades when training with InfoNCE. As our theoretical claims are only valid for the representations after the final layer of the model, we evaluate those. This means we deviate from previous literature and do not use a projection head.

Next, we model each latent with a distinct concentration parameter $\mathbf{\Lambda}_{ii}$ by linearly sampling values between 5 and a maximum of either 50 or 200. When training the encoder with AnInfoNCE, we recover all latent dimensions: The linear identifiability scores are 99.7% and 99.6%, respectively; However, training with the regular InfoNCE loss leads to substantially lower scores of 77.4% and 76.3%, respectively. Fig. 3**C & D** show how the learned $\hat{\mathbf{\Lambda}}$ values change during training. We observe that values of $\hat{\mathbf{\Lambda}}$ approach the ground truth, except for very high $\mathbf{\Lambda}$ values: This is visible in Fig. 3**D** when $\mathbf{\Lambda}_{ii} > 100$). We analyze this behavior further in Appx. I.

Table 1: **Better augmentations readout does not imply better classification performance.** Comparison of different training objectives for linear downstream classification (*Classes*) and augmentation readout (*Grayscale* to *Saturate*, *Avg.* denotes the average over those) performance. Learning $\mathbf{\Lambda}$ results in higher readout accuracy of the used augmentations, which can be interpreted as a better recovery of style latents. However, better augmentations readout does not result in better linear classification accuracy. We note that our theory and all synthetic and VAE experiments apply to the post-projector (backb.+proj.) features which are typically not used in practice because using backbone features results in better performance.

	Training Objective	Features	Classes	Grayscale	Brightness	Contrast	Hue	Saturate	Avg.
CIFAR10	InfoNCE	backbone	90.9	56.1	59.3	62.6	55.4	55.1	57.7
	AnInfoNCE	backbone	88.4	80.1	93.4	91.1	65.5	72.3	80.5
	AnInfoNCE, w/o ℓ_2-norm.	backbone	83.2	96.2	98.0	96.6	76.8	88.4	91.2
	InfoNCE	backb.+proj.	89.8	52.0	52.0	52.0	51.8	51.0	51.8
	AnInfoNCE	backb.+proj.	85.4	71.4	71.4	77.1	59.7	65.5	69.0
	AnInfoNCE, w/o ℓ_2-norm.	backb.+proj.	79.1	65.5	65.5	87.6	70.5	83.8	74.6
ImageNet	InfoNCE	backbone	68.2	98.7	71.6	72.5	68.0	61.2	74.4
	AnInfoNCE	backbone	59.0	98.8	81.5	82.3	71.6	62.5	79.3
	InfoNCE	backb.+proj.	57.7	58.6	51.9	51.9	51.5	51.1	53.0
	AnInfoNCE	backb.+proj.	26.7	71.0	55.2	54.1	52.6	52.7	57.1

Appendix G. Real-world Experiments

Preliminaries. For our CIFAR10 [33] experiments, we train a ResNet18 [25] for 1000 epochs with the code from Hua [26]. For ImageNet [48] experiments, we train a ResNet50 for 100 epochs with the code from Facebook Research [18].

Results. On real-world datasets, we do not have access to true generative factors and the latent conditional distribution. Therefore, we need to create views using regular SimCLR augmentations. We calculate the linear readout accuracy on the augmentations used during training to judge how well we recover the latent factors. Chance-level performance is 50%. Our experiments on CIFAR10 and ImageNet show that our loss with a trainable $\hat{\mathbf{\Lambda}}$ leads to much higher readout accuracy on the used augmentations, i.e., we successfully recover more latent dimensions compared to the regular InfoNCE loss (Tab. 1). While we reduce the information loss, this surprisingly does not lead to better downstream accuracy (Tab. 1, *Classes* column). Removing ℓ_2-normalization results in an encoder with the highest augmentations readout accuracy and simultaneously the lowest classification accuracy. We analyze this discrepancy, the validity of our assumptions for real-world data, and the required steps for closing the remaining gap between CL theory and practice in Appx. H.

Appendix H. Analysis

Sec. 4 shows some scenarios in which recovering more (latent) information results in better downstream performance (e.g., readout accuracy on MNIST). While AnInfoNCE outperforms InfoNCE in these controlled scenarios, it fails in others: On CIFAR10 and ImageNet, we observe a trade-off between the augmentation readout and linear classification readout accuracies. While higher accuracy

in augmentation readout indicates a better capture of style latents (i.e., more latent information is recovered), it does not coincide with higher classification accuracy. Scaling up synthetic experiments to ImageNet comes with changes to various aspects of the training: The encoder's architecture is switched to a ResNet from an invertible MLP, views are sampled using augmentations instead of the ground-truth generative model, and the encoder's output dimensionality is orders of magnitude higher. To control for these differences, we conduct a minimal-change experiment on MNIST, changing only how the views are generated while the encoder and the training procedure remain the same. We demonstrate that using augmentations instead of generating the views using the ground-truth DGP is the main reason for the observed trade-off.

Minimal-change experiment on MNIST exhibits the same identifiability-accuracy trade-off. To visualize how augmentations influence the positive conditional distribution in latent space, we train a VAE on augmented MNIST images and then project these augmented images onto the learned VAE's latent space. We observe that the latents of positive pairs do not follow a projected Gaussian distribution and can even follow a bimodal one (see Fig. 7, Appx. I). Our theory does not cover this case; neither do other theories we are aware of, and so this mismatch presents a fruitful direction for future research on closing the gap between theory and practice.

Further, to better analyze the interplay between augmentations and disentanglement, we train a model either using InfoNCE or AnInfoNCE on MNIST when views are generated using SimCLR augmentations (except color augmentations as MNIST is grayscale):

Loss	Class	Lin. ident. $[R^2]$	Crop	Gaussian Blur
InfoNCE	**0.97**	0.34	0.53	0.58
AnInfoNCE	0.94	**0.44**	**0.59**	**0.63**

We observe the same accuracy-identifiability trade-off on MNIST as before on ImageNet and CIFAR10: While the augmentations' prediction accuracy improves, the classification accuracy decreases. Using the VAE we trained on MNIST, we can also calculate linear identifiability scores, which are higher for AnInfoNCE (but still moderately low). Note that training with AnInfoNCE on MNIST when views are generated by passing ground-truth latents through the VAE leads to perfect linear identifiability scores ($R^2 = 1$). In that case, we do not observe a trade-off between linear identifiability scores and downstream accuracy.

We observe the accuracy-identifiability trade-off on the Causal 3DIdent dataset. The Causal 3DIdent dataset (C-3DIdent, 60, 74) is a computer-generated image dataset with seven classes and ten ground-truth latent factors, rendered with Blender [4]. This dataset is more complex than MNIST and thus constitutes an interesting intermediate step between MNIST and visually complex datasets such as ImageNet.

Further ablation studies. We analyze the learning dynamics of AnInfoNCE and conduct a thorough hyperparameter study in Appx. I. We observe that the performance of AnInfoNCE is strongly influenced by sample efficiency, i.e., the batch size and the latent dimensionality. Further, we observe that larger concentration parameters also require larger batch sizes. We link this observation to the role of the *augmentation overlap* [63]: Higher Λ means a more concentrated conditional and requires larger batch sizes for sufficient augmentation overlap. Investigating the evolution of the identifiability scores over time, we observe a step-like behavior in how the latents are learned,

corroborating the results by Simon et al. [51]. We discuss connections between AnInfoNCE and TriCL [72] and show how the losses compare experimentally in Appx. I.4.

Discussion. Given our analysis above, we identify the use of augmentations in real-world datasets for generating views as the main cause for the accuracy-identifiability trade-off: Assuming an underlying LVM, these augmentations correspond to changes in latent factors, modeled by the positive conditional distribution. This points to a potential cause for unexpectedly low performance: The conditional distribution implied by augmentations may not correspond to the one assumed by the loss. First, the form of the conditional implicitly defined by the augmentations may not be Gaussian or even uni-modal. Second, the corresponding concentration parameters may be high, leading to a flat loss landscape and optimization issues in practice.

Appendix I. Additional Analysis: Ablations and Parameter Sensitivity

In Section H of the main manuscript, we identified the use of augmentations as the primary issue in observing a trade-off between accuracy and identifiability on real-world data. In this Section, we analyze other reasons why scaling AnInfoNCE to real-world data is challenging. In particular, we observe that the interplay between the batch size, the latent dimensionality and the value of the concentration parameter plays a crucial role. To demonstrate this, we run controlled experiments on synthetic data to investigate the assumptions and properties of our theory. Since our analysis equally applies to the isotropic and anisotropic cases, we use the isotropic one for our experiments for simplicity but consider the general anisotropic case in our additional theoretical results.

I.1. Issues when Scaling up AnInfoNCE

Batch Size Needs to Scale with the Dimensionality. One potential difference between the fully-controlled and the real-world experiments is the latent dimensionality, which will (most likely) be substantially higher for more complex image data. Therefore, we investigate the influence of the latent dimension and batch size on the linear identifiability score (Fig. 4A). We observe a significant degradation of performance with a smaller batch size and higher latent dimensionality, corroborating the theoretical predictions of Wang et al. [63] and in line with the empirical observation that CL performs better with a higher batch size [6]. Even for a modest dimensionality, a rather high batch size is required. Extrapolating our results, feature encoders on the order of what is typically used on CIFAR10 and ImageNet (512 to 2048 dimensions) would require extremely large batch sizes to reach high identifiability scores even if all assumptions match our theory.

Batch Size Needs to Scale with the Concentration $\mathbf{\Lambda}$. Another important difference between our controlled and real-world experiments concerns the choice of concentration parameters and its influence on the batch size for identifiability. More concretely, the more concentrated the conditionals are, and the higher the dimensionality, the less training signal CL has to structure the representation [63]. This was denoted by Wang et al. [63] as the role of *augmentation overlap*. Our anisotropic positive conditional distribution could help explain the role of augmentation overlap from an identifiability perspective via the concentration parameter $\mathbf{\Lambda}$, whose effect is illustrated in Fig. 5. Larger $\mathbf{\Lambda}$ means a more concentrated conditional and requires larger batch sizes for sufficient augmentation overlap; otherwise, the R^2 score will degrade (see Fig. 4B for $d = 20$). In

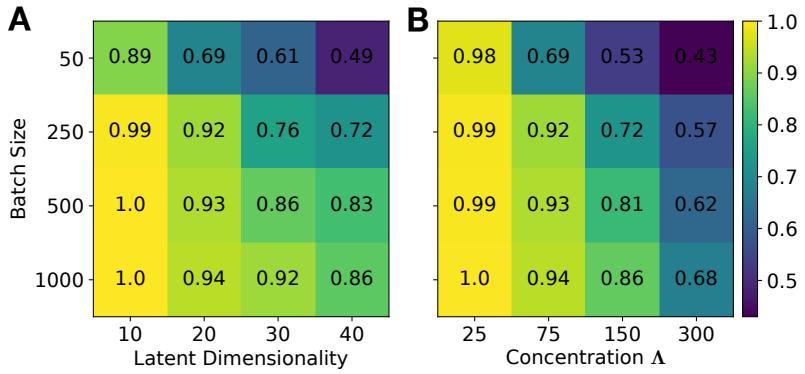

Figure 4: **Latent dimensionality, concentration, and batch size influence identifiability.** Linear identifiability, quantified by the R^2 score between reconstructed and ground-truth latents, degrades with **A:** higher latent dimensionality d; and **B:** larger concentration parameter $\mathbf{\Lambda}$ in the ground-truth positive conditional. Both detrimental effects can be countered by increasing the batch size.

real-world experiments, the positive pairs distribution is determined by augmentations. Likely, the shape or the class of an object are minimally affected by the augmentations, which corresponds to those latent dimensions having a large concentration parameter, thereby again increasing the necessary batch size to reach high identifiability scores.

Augmentations can Violate our Conditional Assumption. Next, we consider the role of different positive pair distributions. In Sec. 3, we assume a normalized Gaussian distribution over a hypersphere. However, it is unclear whether practical augmentation strategies follow such a conditional. We test our assumption's validity by training a Variational Autoencoder (VAE) on augmented images from MNIST with only crops and horizontal flips as augmentations since MNIST images are grayscale. Training an encoder on MNIST using these augmentations leads to a KNN accuracy of 96.7%, validating our augmentation choice. We visualize an unaugmented image and VAE reconstructions of its augmented versions in Fig. 6A. To show the conditional distribution, we project augmented images onto the learned VAE latent space and demonstrate in Fig. 6B that the conditional distribution does not follow a normalized Gaussian distribution for most dimensions and can even be bimodal; we show more examples in Fig. 7. This holds even though the latent space of the VAE across all augmented data samples is Gaussian by construction. Our theory does not cover this case and so this mismatch presents a fruitful direction for future research on closing the gap between theory and practice.

Augmentations can Violate our Uniformity Assumption. Our theory assumes the same uniform marginal $p(\boldsymbol{z})$ for both training and testing. However, this assumption's validity is unclear in practice: During training, even anchor points are produced by data augmentation, creating a potential mismatch to the test domain which lacks these augmentations. We test the influence of violations of this assumption by interpolating the marginal between a uniform and a von Mises-Fisher (vMF) distribution with varying concentration. We center the vMF at the north (south) pole during training (testing) (Fig. 6C). We train a binary classifier to distinguish samples from the training and test distributions. For different concentrations, we compute the domain classification accuracy and the identifiability score (Fig. 6D), showing a stark anticorrelation. Our domain classification results

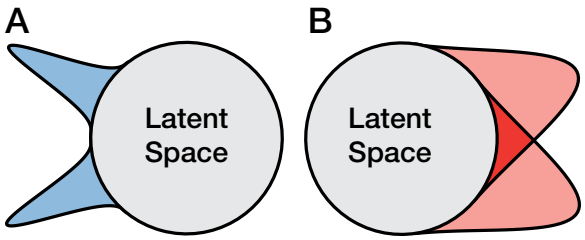

Figure 5: **Concentration of positive pairs influences augmentation overlap**. We figuratively visualize conditional distributions with large (**A**) and small (**B**) concentration parameters $\mathbf{\Lambda}$. For large concentration values, samples from the conditional distribution of two anchor points do not overlap, signifying missing augmentation overlap. This is not the case for small $\mathbf{\Lambda}$ values.

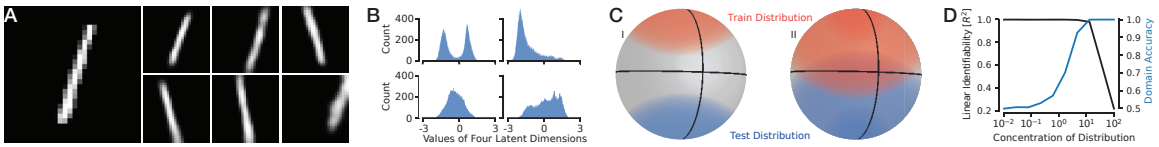

Figure 6: **Data augmentation can violate our data distribution assumptions.** We train a VAE on augmented MNIST images. **A.** We show an MNIST sample and six augmented versions generated by the VAE. **B.** We project augmented versions of an MNIST sample into the VAE's latent space and visualize the distribution of four dimensions. This demonstrates a violated conditional assumption of a normalized Gaussian. **C.** We visualize violations of the uniformity assumption for the marginal distribution during training and testing. **D.** We observe degrading identifiability scores when the train and test domains can be distinguished. Since an almost perfect domain classifier can be trained on real-world datasets, we conclude that the uniformity assumption is likely violated in practice.

with regular SimCLR augmentations (MNIST: 99.1%, CIFAR10: 99.2%, and ImageNet 98.7%) demonstrate that binary classifiers can almost perfectly distinguish augmented and non-augmented samples on real-world data. Connecting this to observed anticorrelation with identifiability, we conclude that the mismatch between distributions hurts the quality of representations.

The Conditional's Effect on Learning Dynamics. Finally, the generative process of real-world image data is most likely substantially richer compared to the previously studied synthetic DGPs, resulting in a generally harder learning problem requiring longer training time. In particular, we hypothesize that the conditional distribution's concentration partly influences the learning speed; this aligns with previous observations of flat loss landscapes for Noise Contrastive Estimation (NCE) [35]. Our anisotropic conditional of positive pairs provides a possible explanation: We hypothesize that similar to Saxe et al. [50], the factors controlling the variances (in our case, the concentration $\mathbf{\Lambda}$) affect the loss landscape, making some latents *slow* to learn. Thus, it is conceivable that more style-like (ground-truth) latents are learned slower (thus, harder) compared to more content-like ones (see, e.g., in Fig. 2B).

We demonstrate the influence of the concentration parameter on the learning dynamics as follows. We define two DGPs using the synthetic data from Section 4.1. We assume a uniform marginal and an isotropic normalized Gaussian conditional with two different $\mathbf{\Lambda}$ parameters: $\mathbf{\Lambda} = 25$ and $\mathbf{\Lambda} = 75$. A higher concentration parameter corresponds to a more narrow conditional distribution and

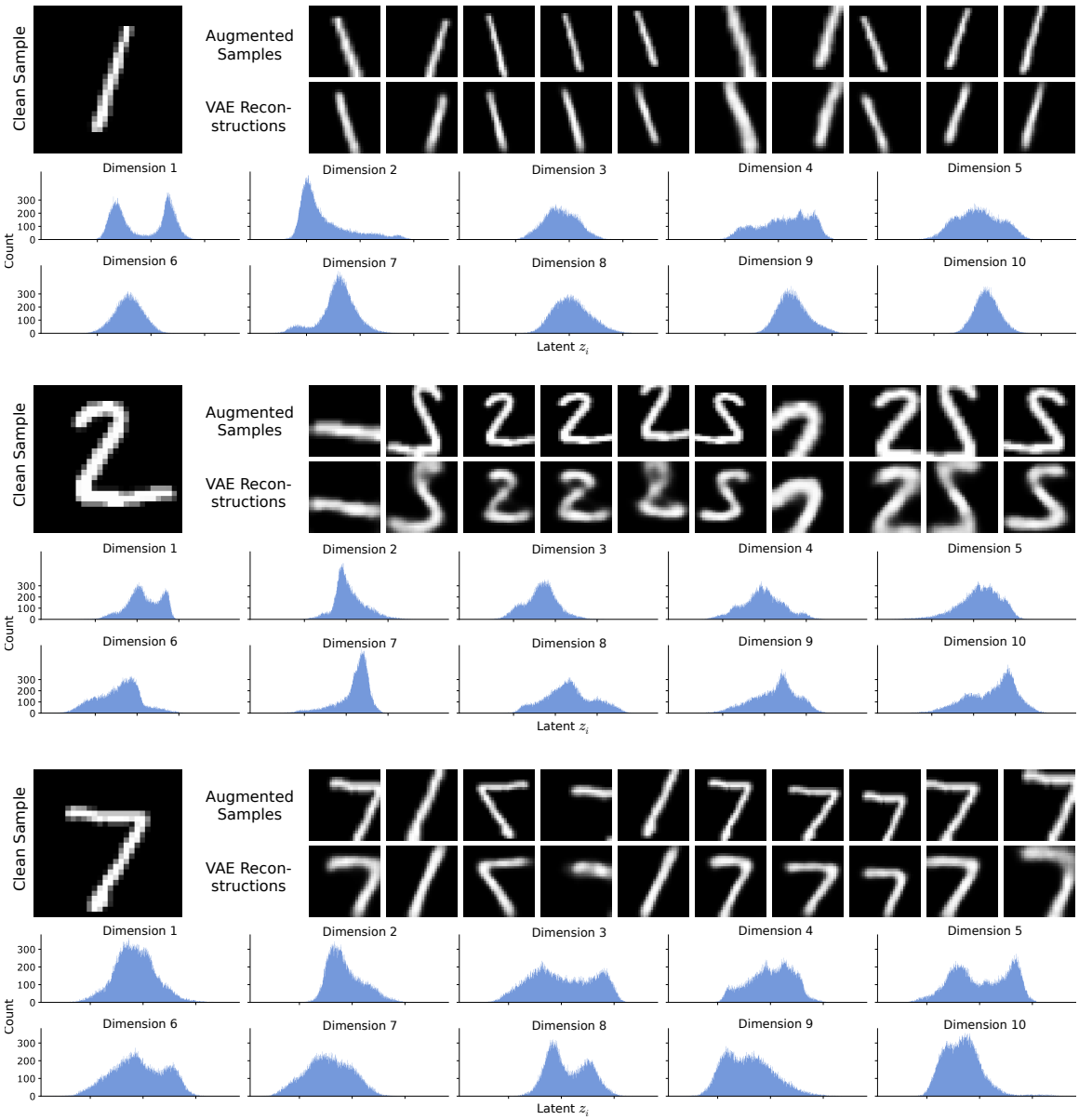

Figure 7: **When using augmentations to generate views for CL, the conditional generally no longer follows a Gaussian distribution.** We show augmented images as well as their reconstructions using a VAE trained on augmented images. We project the augmented images into the latent space of the trained VAE and observe that their distribution does not resemble a Gaussian distribution and can even be bimodal.

thus, lower variation between the anchor and its conditional sample in the latent space. We analyze how $\mathbf{\Lambda}$ affects learning dynamics (Fig. 8) and observe that learning the latents follows a step-like behavior, corroborating Simon et al. [51], where latents with larger $\mathbf{\Lambda}$ are learned slower (Fig. 8 A & D). Alternatively, reaching a target R^2 score requires more gradient steps unless the batch size is

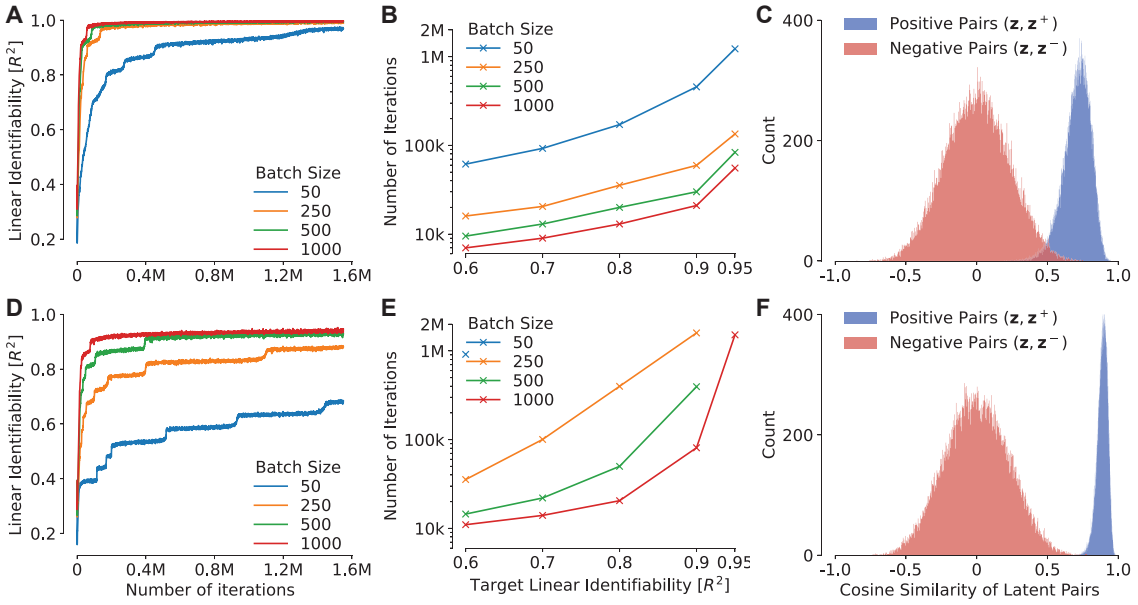

Figure 8: **Analysis of learning dynamics for a small ($\mathbf{\Lambda} = 25$, A-C) and a large ($\mathbf{\Lambda} = 75$, D-F) concentration parameter. A & D**: Linear identifiability measured by the R^2 scores show step-like behavior and generally need more training iterations to reach a certain R^2 score for higher $\mathbf{\Lambda}$. **B & E**: The number of training iterations needed for a certain target R^2 score grows exponentially. **C & F**: We explain degrading R^2 scores for higher concentration parameters $\mathbf{\Lambda}$ with missing overlap in the latent positive and negative distributions.

increased (Fig. 8 B & E). The cosine similarities' distribution for negative and positive pairs clearly shows that large $\mathbf{\Lambda}$ leads to insufficient augmentation overlap: Since the histograms are disjoint for large $\mathbf{\Lambda}$, the anchor's nearest neighbor is always a positive sample (Fig. 8 C & F). This makes distinguishing positive from negative samples trivial and allows for a short-cut solution. However, augmentation overlap cannot fully explain CL in practice: Saunshi et al. [49] show that practical augmentations often provide no significant overlap; nonetheless, CL still succeeds, potentially due to (architectural) inductive biases.

I.2. Dependence of Identifiability on Batch Size, Training Duration, and Dimensionality

We now further analyze the dependence of the batch size, training duration, and dimensionality of the latent space on the identifiability scores. To this end, we train models with different batch sizes on data with various latent dimensionality and investigate their identifiability scores. We use 3 random seeds per configuration and train all models for 500k iterations using the same hyperparameters as described above. We consider the usual conditional distribution of positive pairs with a concentration value $\mathbf{\Lambda} = 75$.

Fig. 9 clearly shows that for higher dimensional latent spaces, one needs to train (exponentially) longer and use an increased batch size. Note that even for the fairly low dimensional settings investigated here, large batch sizes and long training runs were necessary to achieve high identifiability scores. Comparing this to the dimensionalities usually assumed/used in practice, e.g., 512 dimensions when

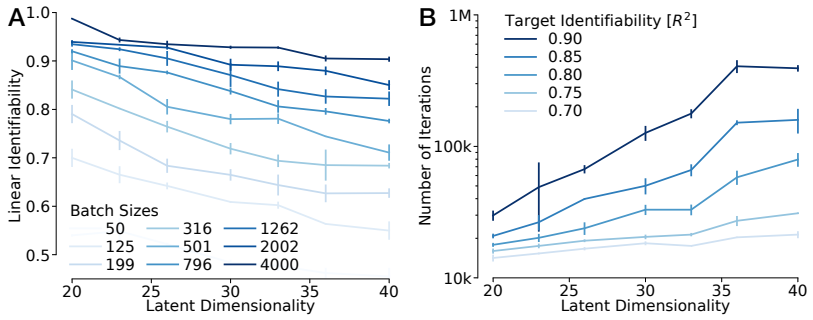

Figure 9: **Higher dimensional latent spaces require longer training with larger batch sizes. A**: The linear identifiability achieved after training for 500k steps decreases with an increasing latent dimensionality but increases with larger batch sizes. **B**: We display the required number of iterations to achieve various target linear identifiability scores (0.70 to 0.90) for a fixed batch size of 4000. Note that the number of iterations required grows exponentially with the latent dimensionality.

training on images, we argue that the usual batch sizes and training durations are too small to achieve high identifiability scores. This observation would explain the mixed performance observed earlier when testing our loss on real-world image data despite its identifiability guarantees.

I.3. Hard Negative Mining and Ensemble AnInfoNCE

Hard Negative Mining. In the previous section, we observed that high concentration parameters require higher batch sizes for successful learning. HN mining can help counter highly concentrated conditionals by increasing the sample density close to the anchors. First, we demonstrate in our synthetic setting for $d = 20$ that with HN mining, the optimum for the loss in terms of $\hat{\mathbf{\Lambda}}$ is indeed at the predicted value of $\mathbf{\Lambda} = \mathbf{\Lambda}^+ - \mathbf{\Lambda}^-$ (Fig. 10A). We also finetune an encoder using AnInfoNCE by adding three hard negatives for each sample. We observe better linear identifiability scores compared to continued regular training, supporting our Cor. B.1, (Fig. 10B).

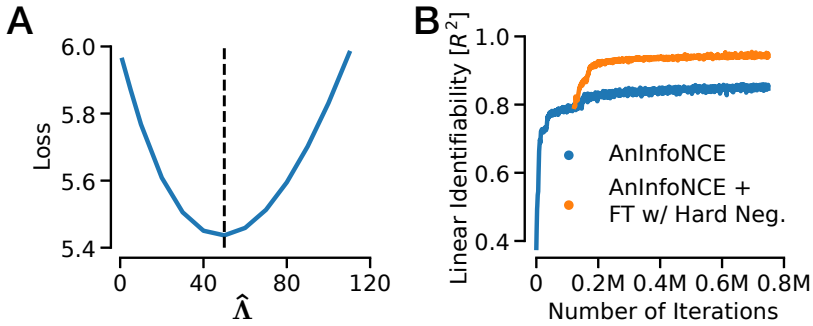

Figure 10: **Hard negative mining improves identifiability. A:** For $\mathbf{\Lambda}^+ = 150$ and $\mathbf{\Lambda}^- = 100$, we find a loss optimum at $\mathbf{\Lambda} = \mathbf{\Lambda}^+ - \mathbf{\Lambda}^- = 50$ as predicted by Cor. B.1; **B:** Finetuning (FT) an encoder on a concatenation of hard negative samples and regular negative samples improves the identifiability score.

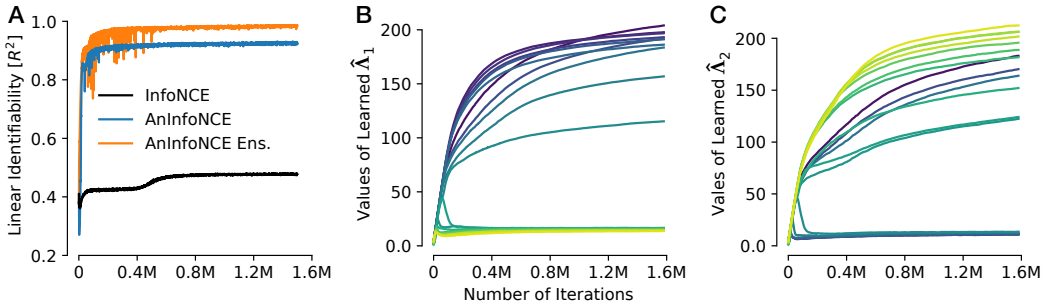

Figure 11: **Ensemble AnInfoNCE improves upon InfoNCE and AnInfoNCE:** We observe higher linear identifiability scores when training with Ensemble AnInfoNCE. When examining the learned $\hat{\Lambda}$ values, we see a specialization in the learned $\hat{\Lambda}$ values: The color scheme is chosen such that the same color indicates the same dimension, and we find that the learned $\hat{\Lambda}$-values for the two DGPs are anti-correlated with respect to each other.

Training With Ensemble AnInfoNCE. To test the benefit of ensembling AnInfoNCE for training, we define two DGPs with $d = 20$. The DGPs share a uniform marginal distribution and differ in their respective conditional distributions. To model latents that cannot be changed by either DGP, such as object class or shape, we set a high concentration parameter of 400 for three latents shared between both DGPs. For the remaining seventeen latents, we choose two $\mathbf{\Lambda}$ values of 15 and 250, modeling strongly and weakly varying latents and alternate them between the DGPs:

$$\mathbf{\Lambda}_1 = \operatorname{diag}(400, 400, 400, 15, \ldots, 15, 250, \ldots, 250)$$
$$\mathbf{\Lambda}_2 = \operatorname{diag}(400, 400, 400, 250, \ldots, 250, 15, \ldots, 15)\,.$$

We sample data with both DGPs and use two losses with two learnable concentration parameters $\hat{\mathbf{\Lambda}}_1, \hat{\mathbf{\Lambda}}_2$ to model both DGPs. We train the encoder on the sum of both losses. We see improved linear identifiability scores when training the encoder on the loss ensemble compared to InfoNCE or AnInfoNCE (see also Fig. 11A):

Loss	InfoNCE	AnInfoNCE	Ens. AnInfoNCE
Lin. ident. [R^2]	0.48	0.93	0.98

Further, we observe a specialization in the learned $\hat{\Lambda}$ values (Fig. 11B+C), where same colors indicate the same $\hat{\Lambda}$ dimension): We observe an anti-correlation between $\hat{\Lambda}_1$ and $\hat{\Lambda}_2$. Optimizing an ensemble loss makes it more likely that for each latent dimension, there exists an augmentation making the dimensions more content-like with a higher $\mathbf{\Lambda}$.

I.4. Comparison to TriCL [72]

We first discuss the conceptual differences, theoretical results and assumptions betwen TriCL and AnInfoNCE. TriCL extends and theoretically analyzes the spectral contrastive loss formulation (their extension to InfoNCE does not have a proof). Our paper extends InfoNCE and provides an identifiability result in that case. Note that the extensions are also different: We weigh the differences of latent coordinates instead of the latents themselves. The TriCL paper provides a result on encoder

output equivalence (similar to Roeder et al. [47], meaning that all encoders optimizing the TriCL loss are linearly related, or, more precisely $f_1(x) = Af_2(x)$. However, the encoder's output is not related to ground-truth latents in any way. Our paper provides a result on the equivalence of the output with the ground-truth latents, meaning that $f \circ g(z) = Oz$. On one hand, our type of result subsumes the TriCL type of result. Moreover, our result is well-suited for the qualitative assessment of inferred latent features, which is not possible with the latter. The theoretical result from TriCL requires that an additional decorrelation regularizer completely vanishes, whereas our method does not have any such term.

We also show an experimental comparison between TriCL and AnInfoNCE below. First, we noticed a small discrepancy between the paper's definition of TriCL and its experimental implementation in how the loss is computed. Based on the used implementation, the scale parameter is not applied to all terms in the sum but only to half of them. We are not sure how how this discrepancy affects the experiments.

We compare the evolution of linear identifiability scores when training with TriCL or AnInfoNCE in our synthetic experiment in Fig. 12. We set the latent dimensionality to 20 and test two cases for the positive conditional distribution. (1) Uniform Λ set to 100 (top row); (2) Dual partitioned Λ, such that half of the dimensions have a low Λ of 15 ("style"-like dimensions) and the other half high Λ of 250 ("content"-like dimensions) (bottom row). In both cases, we observe that AnInfoNCE behaves much more stably compared to TriCL and reaches higher linear identifiability scores, especially when there is an anisotropic conditional.

I.5. Training Details for VAE Training in MNIST

The VAE consists of three fully connected layers with Leaky ReLU activations and is trained on MNIST for 50 epochs. The latent dimensionality d of the VAE is set to 10. We use the trained VAE to generate training images in the following way: We first sample a latent vector z from the marginal distribution and then sample a vector conditioned on z. Since both vectors are normalized to the sphere surface, we multiply them with $\sqrt{d}$ to project them to a Gaussian vector. Finally, we use the VAE decoder to generate images which we train the encoder with the AnInfoNCE loss on.

To provide an intuition for the quality of samples generated by the VAE, we show images generated from random samples from the latent space in Fig. 13 (left). For the highlighted sample in the top-right corner, we also show samples from the conditional distribution for two different Λ values: $\Lambda = 5$ (middle) and $\Lambda = 50$ (right).

Appendix J. Compute requirements

Synthetic experiments. For our synthetic experiments, as described in Sections 4.1, F as well as in Appendix I, we used a single NVIDIA GeForce RTX 2080 Ti with 11GB of RAM for each experiment. We used 8 CPUs and 8 workers per experiment. The convergence time for each experiment varied, e.g. higher batch size or a lower concentration parameter $\hat{\mathbf{\Lambda}}$ generally imply faster convergence: We show the relationship between the batch size and the concentration parameter in Fig. 8. We set the run time for synthetic experiments to ten hours to make sure that all runs converge; the main experiments shown in Figs. 2 and 3 train within one hour to convergence.

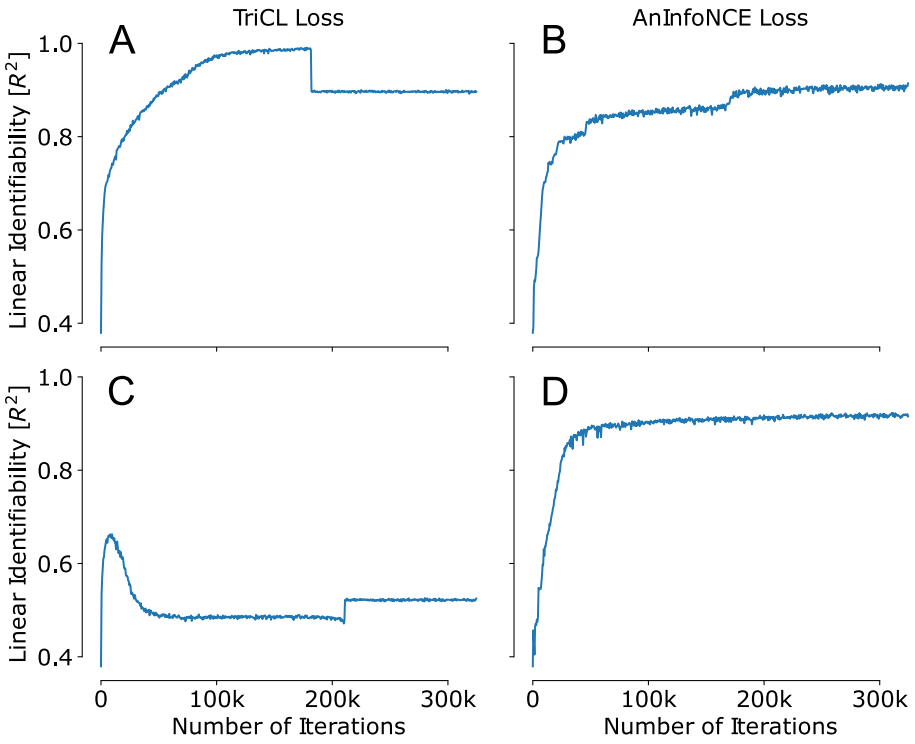

Figure 12: **Synthetic experiments for a latent dimensionality of d=20 when training with TriCL or AnInfoNCE.** Top row: Uniform Λ of 100. Bottom row: Dual partitioned Λ, such that half of the dimensions are varied according to a low Λ of 15 and the other half is varied with a high Λ of 250. In both cases, we observe that AnInfoNCE behaves much more stably compared to TriCL and reaches higher linear identifiability scores, especially when there is an anisotropic positive conditional.

Real-world experiments. On CIFAR10, we trained the ResNet18 model on a single NVIDIA A100 GPU with 40GB RAM and 8 CPUs (with 8 workers). The training took about seven hours per run. The linear readout evaluation as well as the binary augmentations readout took about 30 minutes on the same compute setup.

On ImageNet, we trained a ResNet50 on 8 NVIDIA A100 GPUs with 40GB RAM and 64 CPUs. We set the number of workers to 16. The training took about 15 hours on this setup. The linear readout evaluation as well as the binary augmentations readout took about 3 hours using one GPU.

Compute costs per experiment, main paper. While we used different GPUs between our synthetic and real-world experiments, we will use a metric of generic "GPU-hours" for both cases. As described below, both GPU types require about the same amount of power.

- *i*) Synthetic data, Fig. 2: Experiments showing the influence of linear identifiability scores when varying $\mathbf{\Lambda}$. We show results for 6 values of $\mathbf{\Lambda}$ for InfoNCE and AnInfoNCE. The GPU run time for this experiment is $6 \times 2 \times 10h = 120h$.
- *ii*) Synthetic data, Fig. 3: Experiments showing how AnInfoNCE behaves in a slightly more difficult experimental setting on MNIST. We show results for 3 values of $\mathbf{\Lambda}$ for InfoNCE and

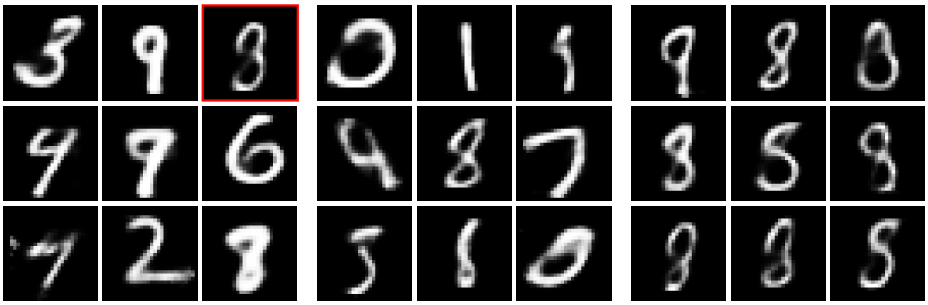

Figure 13: **Samples generated by the MNIST-VAE**. **Left**: Samples from the marginal distribution, used as anchor samples for the contrastive objective. **Middle & Right**: For the highlighted sample, we show potential positive samples from the conditional distribution for two different values of $\mathbf{\Lambda}$. A value of 5 leads to too much variation (middle) and leads to obvious class changes in the conditional samples, whereas a value of 50 leads to reasonable variation. We note that training an encoder on images which were sampled according to a DGP with either ground-truth Λ-value leads to perfect linear disentanglement scores. In other words, a low concentration parameter of $\Lambda = 5$ which changes the class of the conditional view does not pose an issue.

AnInfoNCE in Fig 3 A+B, and for 2 values of $\mathbf{\Lambda}$ when training with AnInfoNCE in Fig 3 C+D. The GPU run time for this experiment is $3 \times 2 \times 10h + 2 \times 10h = 80h$.

iii) Real-world data, Table 1: We show a trade-off between augmentations readout and downstream accuracy on CIFAR10 and ImageNet. We show results for training with InfoNCE and AnInfoNCE both on CIFAR10 and ImageNet. On CIFAR10, we additionally show results for training with AnInfoNCE without ℓ_2 normalization of the output before calculating the loss. We also conduct the linear readout evaluation on both the backbone and the post-projection features. The compute break-down is as follows:

  i) CIFAR10: Training: $3 \times 7h = 21h$, Linear readout: $3 \times 6 \times 0.5h \times 2 = 18h$. Total: $39h$

  ii) ImageNet: Training: $2 \times 15h \times 8$ GPUs$= 240h$, Linear readout: $2 \times 3h \times 6 \times 2 = 72h$. Total: $312h$

  The total compute for the real-world experiments is then 351 GPU hours.

iv) MNIST, inset Table in Sec.H; showing the accuracy-identifiability trade-off. The run time for this experiment has been: $2 \times 10h$=$20h$.

Compute costs per experiment, Appendix.

i) Synthetic data, Fig. 4. We show ablation experiments for varying the batch size, the latent dimensionality and the concentration. We show results for 32 different parameter combinations. The GPU run time for this experiment is $32 \times 10h = 320h$.

ii) Synthetic data, Fig. 8. We analyze the interplay between higher dimensional latent spaces and batch size in more detail. We ran 3 random seeds in this experiment, analyzed 9 different batch sizes across 5 different latent dimensionalities. The total run time for this experiment is $3 \times 9 \times 5 \times 10h = 1350h$.

iii) Synthetic data, Fig. 9. We utilize the previous runs from computing results for Fig. 4; this Figure does not incur additional compute costs.

iv) Hard-negative mining, Fig. 10. We show that hard-negative mining can improve identifiability. We finetune one model trained with regular InfoNCE on AnInfoNCE with hard negative examples. The run time for this experiment is $3h$.
v) Synthetic data, Fig. 11. We show that ensembling can improve upon InfoNCE and AnInfoNCE. The run time for this experiment is $3 \times 10h = 30h$.
vi) Synthetic data, Fig. 12. We show that AnInfoNCE is more stable and reaches higher linear identifiability scores compared to the TriCL loss. The run times for this experiment is $2 \times 10h = 20h$

Total compute estimation. Given the previous paragraphs, the training time estimation is 2942 GPU hours. The maximum power consumption for the 2080Ti GPUs is 250W [55] and 300W for the A100 GPUs [19]. We assume an upper bound on power consumption of 300W per GPU. Then, the energy spent by the GPUs for the experiments shown in this paper is 882.6kWh. We estimate an additional factor of 20 in terms of energy spent during the development stage of this project. Therefore, the total estimated project cost in terms of energy is about 17.7MWh. This corresponds to 3.5t CO_2 emissions [44] which is roughly equivalent to six single-person flights from London to New York [57].

Appendix K. Software stack

We use different open source software packages for our experiments, most notably Docker [40], Singularity [53], scipy and numpy [58], PyTorch [42], and torchvision [38].

Appendix L. Acronyms

CE cross-entropy

AnInfoNCE Anistropic InfoNCE

CL Contrastive Learning

DGP Data Generating Process

HN hard negative

i.i.d. independent and identically distributed

KL Kullback-Leibler Divergence

LooC Leave-one-out Contrastive Learning
LVM Latent Variable Model

MLP Multi-Layer Perceptron

NCE Noise Contrastive Estimation

PD positive definite

SSL Self-Supervised Learning

VAE Variational Autoencoder
vMF von Mises-Fisher

Appendix M. Nomenclature

R^2 coefficient of determination
$\mathcal{L}_{\text{AINCE}}$ AnInfoNCE loss function
$\mathcal{L}_{\text{INCE}}$ InfoNCE loss function
$\mathcal{S}$ hypersphere
$\boldsymbol{f}$ encoder map $\mathcal{X} \to \mathcal{Z}$
$\boldsymbol{g}$ decoder map $\mathcal{Z} \to \mathcal{X}$
$\boldsymbol{h}$ composition of encoder and decoder, i.e., $\boldsymbol{f} \circ \boldsymbol{g}$
$\boldsymbol{\tau}$ temperature in $\mathcal{L}_{\text{INCE}}$

Algebra

$\hat{\mathbf{\Lambda}}$ inferred diagonal matrix of weighting factors
$\mathbf{O}$ orthogonal matrix
$\mathbf{\Lambda}$ diagonal matrix of weighting factors
$\mathbf{P}$ permutation matrix

Latents

$\boldsymbol{z}_c$ content latent vector
$\boldsymbol{z}_s$ style latent vector
$\boldsymbol{z}$ latent vector
$\hat{\boldsymbol{z}}$ reconstructed latent vector
$\mathcal{Z}_c$ content
$\mathcal{Z}_s$ style
$\mathcal{Z}$ latents
d dimensionality of the latent space $\mathcal{Z}$
z latent single component
$\boldsymbol{z}^+$ positive latent vector
$\boldsymbol{z}^-$ negative latent vector

Observations

D dimensionality of the observation space $\mathcal{X}$
M number of negative samples
$\boldsymbol{x}^-$ negative observation vector
$\boldsymbol{x}$ observation vector
$\mathcal{X}$ observation space
$\boldsymbol{x}^+$ positive observation vector