Homomorphic Self-Supervised Learning

T. Anderson Keller
Apple
t.anderson.keller@gmail.com

Xavier Suau
Apple
xsuaucuadros@apple.com

Luca Zappella
Apple
lzappella@apple.com

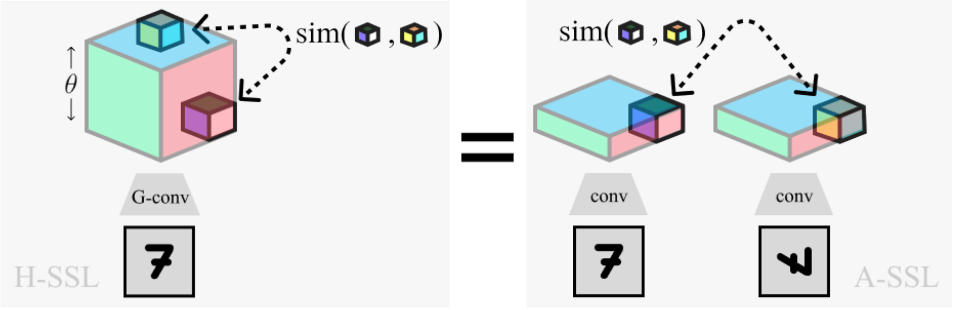
Abstract

In this work, we observe that many existing self-supervised learning algorithms can be both unified and generalized when seen through the lens of equivariant representations. Specifically, we introduce a general framework we call *Homomorphic Self-Supervised Learning*, and theoretically show how it may subsume the use of input-augmentations provided an augmentation-homomorphic feature extractor. We validate this theory experimentally for simple augmentations, demonstrate how the framework fails when representational structure is removed, and further empirically explore how the parameters of this framework relate to those of traditional augmentation-based self-supervised learning. We conclude with a discussion of the potential benefits afforded by this new perspective on self-supervised learning.

1 Introduction

Many self-supervised learning (SSL) techniques can be colloquially defined as representation learning algorithms which extract approximate supervision signals directly from the input data itself [23]. In practice, this supervision signal is often obtained by performing symmetry transformations of the input with respect to task-relevant information, meaning the transformations leave task-relevant information unchanged, while altering task-irrelevant information. Numerous theoretical and empirical works have shown that by combining such symmetry transformations with contrastive objectives, powerful lower dimensional representations can be learned which support linear-separability, identifiability of generative factors, and reduced sample complexity [1, 5, 6, 12, 15, 17, 24, 34, 36–38, 40, 42]. One rapidly developing domain of deep learning research which is specifically focused on the structured and accurate representation of the input with respect to symmetry transformations is that of equivariant neural networks [7, 8, 13, 14, 44–46]. In this work, we study the properties of SSL algorithms when equivariant neural networks are used as backbone feature extractors. Interestingly, we find a convergence of existing loss functions from the literature, and ultimately generalize these with the framework of *Homomorphic Self-Supervised Learning*.

Figure 1: Overview of Homomorphic-SSL (left) and its relation to traditional Augmentation-based SSL (right). Positive pairs extracted from the lifted dimension (θ) of a rotation equivariant network (G-conv) are equivalent to pairs extracted from the separate representations of two rotated images.



36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2 Background

Equivariance The map $f : \mathcal{X} \to \mathcal{Z}$ is said to be equivariant with respect to the group $\mathcal{G} = (G, \cdot)$ if

$$\exists \Gamma_g \quad \text{such that} \quad f(T_g[\boldsymbol{x}]) = \Gamma_g[f(\boldsymbol{x})] \quad \forall g \in G \;, \tag{1}$$

where G is the set of all group elements, $\cdot$ is the group operation, T_g is the representation of the transformation $g \in G$ in input space $\mathcal{X}$, and Γ_g is the representation of the same transformation in output space $\mathcal{Z}$. If T_g and Γ_g are formal group representations [31] such maps f are termed group-homomorphisms since they can be seen to preserve the structure of the group in the output space. There are many different methods for constructing group equivariant neural networks, resulting in different representations of the transformation in feature space Γ_g. In this work, we consider only discrete groups $\mathcal{G}$ and networks which admit regular representations for Γ (see Appendix B for an example). Specifically, we denote the output of our network $f(\boldsymbol{x}) = \boldsymbol{z} \in \mathbb{R}^{C \times |G|}$, where C is the number of output channels. As a simple example, a standard convolutional layer would have all height (H) and width (W) spatial coordinates as the set G, giving $\boldsymbol{z} \in \mathbb{R}^{C \times HW}$. A group-equivariant neural network [8] which is equivariant with respect to the the group of all integer translations and 90-degree rotations ($p4$) would thus have a feature multiplicity four times larger ($\boldsymbol{z} \in \mathbb{R}^{C \times 4HW}$), since each spatial element is associated with the four distinct rotation elements $(0^o, 90^o, 180^o, 270^o)$. Such a rotation equivariant network is depicted in Figure 1 with the 'lifted' rotation dimension extended along the vertical axis (θ). In both the translation and rotation cases, the regular representation Γ_g acts by permuting the representation along the group dimension, leaving the feature channels unchanged.

Notation The vector of features (channels) at a specific group element g is sometimes called a 'fiber' [7]. In this work we use the following shorthand for indexing fibers according to their group element g: $\boldsymbol{z}(g) \equiv \boldsymbol{z}_{:,g} \in \mathbb{R}^C$. Similarly, the set of fibers corresponding to an ordered set of group elements $\boldsymbol{g}$ can be called a 'fiber bundle' which we denote: $\boldsymbol{z}(\boldsymbol{g}) = [\boldsymbol{z}(g) \mid g \in \boldsymbol{g}] \in \mathbb{R}^{|\boldsymbol{g}|C}$. Fiber bundles can be seen in Figure 1 as the small cubes being compared (with the fibers themselves extending into the undepicted fourth dimension). Using this notation, we can define the action of Γ_g as: $\Gamma_g[\boldsymbol{z}(\boldsymbol{g}_0)] = \boldsymbol{z}(g^{-1} \cdot \boldsymbol{g}_0)$. Thus Γ_g can be seen to move the fibers from 'base' locations $\boldsymbol{g}_0$ to a new ordered set of locations $g^{-1} \cdot \boldsymbol{g}_0$. We highlight that order is critical for our definition since a transformation such as rotation may simply permute $\boldsymbol{g}_0$ while leaving the unordered set intact.

Augmentation-based SSL (A-SSL) Many state of the art SSL approaches rely on input augmentations in order to selectively extract task-relevant information. One prominent framework, SimCLR [5], trains a backbone feature extractor $f(\cdot)$ to minimize a contrastive loss applied to the representations of two augmented versions of an image. Specifically, given a batch of N input images $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, a similarity function $\mathrm{sim}(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a}^T \boldsymbol{b}}{||\boldsymbol{a}|| \cdot ||\boldsymbol{b}||}$, and a non-linear 'projection head' $h : \mathcal{Z} \to \mathcal{Y}$, the SimCLR loss is given as:

$$\mathcal{L}_{\text{A-SSL}}(\mathbf{X}) = -\frac{1}{N} \sum_i^N \mathbb{E}_{g_1, g_2 \sim G} \log \frac{\exp\Big(\mathrm{sim}\big(h(f(T_{g_1}[\boldsymbol{x}_i])), h(f(T_{g_2}[\boldsymbol{x}_i]))\big)/\tau\Big)}{\sum_{k \neq i}^N \sum_{j,l}^2 \exp\Big(\mathrm{sim}\big(h(f(T_{g_j}[\boldsymbol{x}_i])), h(f(T_{g_l}[\boldsymbol{x}_k]))\big)/\tau\Big)} \;, \tag{2}$$

where G is the set of all augmentations, $T_g[\boldsymbol{x}]$ denotes the action of the sampled augmentation g on the input, and τ is the 'temperature' of the softmax. In this work, we will focus on the SimCLR objective for simplicity, but our analysis also applies to non-contrastive frameworks such as BYOL [15] and SimSiam [6], provided the backbone is equivariant (i.e. augmentation-homomorphic).

3 Homomorphic Self-Supervised Learning

In this section we introduce Homomorphic Self-Supervised Learning (H-SSL) as a general framework for SSL with homomorphic encoders, and further show how many existing SSL algorithms can be both unified and generalized. To begin, consider an A-SSL objective such as Equation 2 when f is equivariant with respect to the input augmentation. By the definition of equivariant maps in Equation 1, the augmentation commutes with the feature extractor: $h(f(T_g[\boldsymbol{x}])) = h(\Gamma_g[f(\boldsymbol{x})])$. Thus, replacing $f(\boldsymbol{x}_i)$ with its output $\boldsymbol{z}_i = \boldsymbol{z}(\boldsymbol{g}_0)$, and applying the definition of the operator, we get:

$$\mathcal{L}_{\text{H-SSL}}(\mathbf{X}) = -\frac{1}{N} \sum_i^N \mathbb{E}_{g_1, g_2 \sim G} \log \frac{\exp\Big(\mathrm{sim}\big(h(\boldsymbol{z}_i(g_1^{-1} \cdot \boldsymbol{g}_0)), h(\boldsymbol{z}_i(g_2^{-1} \cdot \boldsymbol{g}_0))\big)/\tau\Big)}{\sum_{k \neq i}^N \sum_{j,l}^2 \exp\Big(\mathrm{sim}\big(h(\boldsymbol{z}_i(g_j^{-1} \cdot \boldsymbol{g}_0)), h(\boldsymbol{z}_k(g_k^{-1} \cdot \boldsymbol{g}_0))\big)/\tau\Big)} \;. \tag{3}$$

Ultimately, we see that $\mathcal{L}_{\text{H-SSL}}$ subsumes the use of input augmentations by defining the 'positive pairs' in the numerator as two fiber bundles from *the same representation* $\boldsymbol{z}_i$, simply indexed using two differently transformed base spaces $g_1^{-1} \cdot \boldsymbol{g}_0$ and $g_2^{-1} \cdot \boldsymbol{g}_0$ (depicted in Figure 1). Interestingly, this loss highlights the base space $\boldsymbol{g}_0$ as a parameter choice previously unexplored in the A-SSL frameworks.

A second interesting consequence of this derivation is the striking similarity of the $\mathcal{L}_{\text{H-SSL}}$ objective and other existing SSL objectives which operate without explicit input augmentations to generate multiple views. Specifically, when applied to images, Greedy InfoMax (GIM) [26] and Contrastive Predictive Coding (CPC) [27][1] use a similar InfoNCE-inspired loss (as in SimCLR) but between different spatial locations of a convolutional filter stack for a single image. Similarly, the 'local' term in the Deep InfoMax objective (DIM(L)) [16] operates entirely within convolutional feature maps. Consequently, these losses are contained in our framework where $\mathcal{G}$ is set to the 2D translation group, and $\boldsymbol{g}_0$ is a small subset of the spatial coordinates. Since $\mathcal{L}_{\text{H-SSL}}$ is also derived directly from $\mathcal{L}_{\text{A-SSL}}$ (when f is equivariant), we see that it provides a means to unify these previously distinct sets of SSL objectives. In Section 4 we validate this theoretical equivalence empirically. Furthermore, since $\mathcal{L}_{\text{H-SSL}}$ is defined for transformation groups beyond translation, it can be seen to generalize these augmentation-free objectives in a way that we have not previously seen exploited in the literature. In Section 4 we include a preliminary exploration this generalization to scale and rotation groups.

4 Experiments

We now empirically validate the derived equivalence of A-SSL and H-SSL in practice, and further reinforce our stated assumptions by demonstrating how H-SSL objectives are ineffective when representational structure is removed. We study how the parameters of H-SSL (topographic distance) relate to those traditionally used in A-SSL (augmentation strength), and finally explore how new parameter generalizations afforded by our framework (such as choices of $\boldsymbol{g}_0$ and $\mathcal{G}$) impact performance.

Empirical Validation For perfectly equivariant networks f, and sets of transformations which exactly satisfy the group axioms, the equivalence between Equations 2 and 3 is exact. However, in practice, due to discretization, boundary effects, and sampling artifacts, even for simple transformations such as translation, equivariance has been shown to not be strictly satisfied [48]. In Table 1 we empirically validate our proposed theoretical equivalence between $\mathcal{L}_{\text{A-SSL}}$ and $\mathcal{L}_{\text{H-SSL}}$, showing a tight correspondence between the downstream accuracy of linear classifiers trained on representations learned via the two frameworks. Precisely, for each transformation (Rotation, Translation, Scale), we use a backbone network which is equivariant specifically with respect to that transformation (e.g. rotation equivariant CNNs, regular CNNs, and Scale Equivariant Steerable Networks (SESN) [33]). In all settings, we take $\boldsymbol{z}$ from the final possible layer and set $\boldsymbol{g}_0$ to be a single fiber of dimension 128.

Table 1: MNIST [22], CIFAR10 [20] and Tiny ImageNet [21] top-1 test accuracy (mean ± std. over 3 runs) of a detached classifier trained on the representations from SSL methods with different backbones. We compare $\mathcal{L}_{\text{A-SSL}}$ and $\mathcal{L}_{\text{H-SSL}}$ with random frozen and fully supervised backbones. We see equivalence between A-SSL and H-SSL as desired from the first two columns, and often a significant improvement in performance for H-SSL methods when moving from Translation to generalized groups such as Scale. Full experiment details can be found in Appendix B.

Dataset	Transformation	Backbone	A-SSL	H-SSL	Frozen	Supervised
MNIST	Rotation	Rot-Eq.	68.2 ± 2.5	70.3 ± 5.4	*87.2 ± 0.8*	*99.4 ± 0.1*
	Translation	CNN	95.9 ± 0.3	96.0 ± 1.3	*94.1 ± 0.3*	*99.2 ± 0.1*
	Scale	SESN	98.6 ± 0.1	98.3 ± 0.2	*94.7 ± 0.6*	*99.3 ± 0.1*
CIFAR10	Rotation	Rot-Eq.	46.1 ± 0.6	48.3 ± 0.5	*38.4 ± 0.1*	*73.0 ± 1.1*
	Translation	CNN	39.2 ± 0.5	36.3 ± 1.1	*40.4 ± 0.2*	*76.2 ± 1.4*
	Scale	SESN	59.4 ± 0.2	56.7 ± 0.4	*41.1 ± 0.6*	*78.0 ± 0.2*
Tiny ImageNet	Rotation	Rot-Eq.	14.9 ± 0.3	13.5 ± 0.5	*6.1 ± 0.2*	*22.5 ± 0.1*
	Scale	SESN	16.2 ± 0.4	14.0 ± 1.3	*6.4 ± 0.2*	*23.7 ± 0.2*

[1] In CPC, the authors use an autoregressive encoder to encode one element of the positive pairs. In GIM, they find that in the visual domain, this autoregressive encoder is not necessary, and thus the loss reduces to a standard contrastive loss between the representations from raw spatial patches, as defined here.

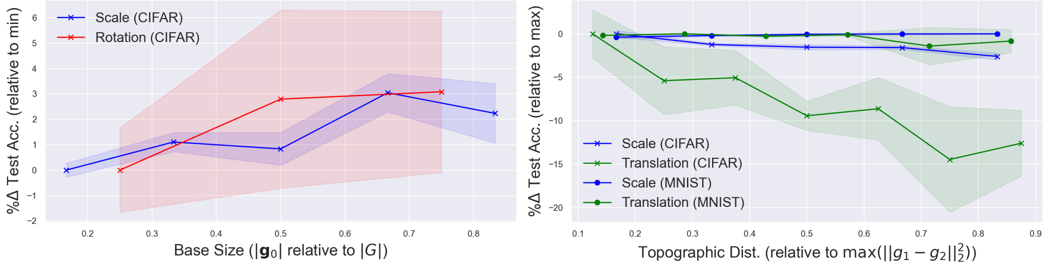
H-SSL Without Structure To validate our assertion that $\mathcal{L}_{\text{H-SSL}}$ requires a homomorphism, in Table 2 we show the same models from Table 1 without equivariant backbones. We observe $\mathcal{L}_{\text{H-SSL}}$ models perform significantly below their input-augmentation counterparts, and similarly to a 'frozen' randomly initialized backbone baseline – indicating the learning algorithm is no longer effective.

Table 2: An extension of Table 1 with non-equivariant backbones. We see that the H-SSL methods perform similar to, or worse than, the frozen baseline when equivariance is removed, as expected.

Dataset	Transformation	Backbone	A-SSL	H-SSL	Frozen	Supervised
MNIST	Translation	MLP	87.6 ± 0.2	58.2 ± 0.5	83.0 ± 0.8	98.6 ± 0.1
	Scale	CNN ($6 \times CHW$)	95.2 ± 0.1	87.2 ± 2.4	87.2 ± 0.6	99.3 ± 0.1
CIFAR10	Scale	CNN ($6 \times CHW$)	53.6 ± 0.2	37.5 ± 0.1	43.6 ± 0.3	67.9 ± 2.1

Parameters of H-SSL As discussed, The H-SSL framework highlights new parameter choices such as the base space $\boldsymbol{g}_0$. In Figure 2 (left) we plot the $\%$-change in top-1 accuracy on CIFAR-10 as we increase the total size of $\boldsymbol{g}_0$ from 1 (akin to DIM(L) losses) to $|G| - 1$ (akin to SimCLR). We see a minor increase in performance as we increase the size, but note relative stability, again suggesting greater unity between A-SSL and H-SSL. In Figure 2 (right), we explore how the traditional notion of augmentation 'strength' can be equated with the 'topographic distance' between g_1 and g_2 and their associated fiber bundles. Here we approximate topographic distance as euclidean distance between group elements for simplicity ($||g_1 - g_2||_2^2$), where a more correct measure would be computed using the topology of the group. We see, in alignment with prior work [35], that the strength of augmentation (and specifically translation distance) is an important parameter for effective self supervised learning, likely relating to the mutual information between fibers as a function of distance.

Figure 2: Study of the impact of new H-SSL parameters top-1 test accuracy. (Left) Test accuracy marginally increases as we increase total base space size $\boldsymbol{g}_0$. (Right) Test accuracy is constant or decreases as we increase the maximum distance between fiber bundles considered positive pairs.



5 Discussion

In this work we have studied the impact of combining augmentation-homomorphic feature extractors with augmentation-based SSL objectives. In doing so, we have introduced a new framework which we call Homomorphic-SSL which illustrates an equivalence between previously distinct SSL methods when the homomorphism constraint is satisfied. Since it is not currently known how to construct neural networks which are analytically equivariant with respect to all input augmentations used in modern SSL, this constraint is precisely the greatest current limitation of this framework, and we expand on this limitation in Appendix A. We therefore propose this work not as an improvment to the state of the art, but rather as a new perspective on SSL which provides a bridge to previously distant literature. Specifically, one field of research which appears particularly promising for future work is the integration of learned homomorphisms [9, 11, 18, 19, 28] with H-SSL. In the H-SSL framework, a learned homomorphism can be seen as equivalent to a learned augmentation, providing a potential new avenue for approaching the extremely challenging [3] but fruitful [32] goal of learned image augmentations.

We additionally present this work as an attempt to renew interest in SSL objectives which operate without multiple inferences of a transformed image, such as Deep InfoMax [16] and Greedy InfoMax [26], by allowing them to exploit the theoretical foundations developed for multi-view SSL [1, 12, 36–38, 40]. Although DIM-like methods have to-date not yielded the same performance as their A-SSL counterparts, we believe the coupling between objective and network architecture is likely to yield more parallelizable algorithms which are therefore more scalable and biologically plausible [26].

References

[1] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning, 2019. URL `https://arxiv.org/abs/1902.09229`.

[2] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

[3] Arno Blaas, Xavier Suau, Jason Ramapuram, Nicholas Apostoloff, and Luca Zappella. Challenges of adversarial image augmentations. In Melanie F. Pradier, Aaron Schein, Stephanie Hyland, Francisco J. R. Ruiz, and Jessica Z. Forde (eds.), *Proceedings on "I (Still) Can't Believe It's Not Better!" at NeurIPS 2021 Workshops*, volume 163 of *Proceedings of Machine Learning Research*, pp. 9–14. PMLR, 13 Dec 2022. URL `https://proceedings.mlr.press/v163/blaas22a.html`.

[4] Gabriele Cesa, Leon Lang, and Maurice Weiler. A program to build E(N)-equivariant steerable CNNs. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=WE4qe9xlnQw`.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020.

[7] Taco Cohen and M. Welling. Steerable cnns. *ArXiv*, abs/1612.08498, 2017.

[8] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999, 2016.

[9] Marissa Connor, Gregory Canal, and Christopher Rozell. Variational autoencoder with learned latent structure. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2359–2367. PMLR, 13–15 Apr 2021. URL `http://proceedings.mlr.press/v130/connor21a.html`.

[10] Rumen Dangovski, Li Jing, Charlotte Loh, Seungwook Han, Akash Srivastava, Brian Cheung, Pulkit Agrawal, and Marin Soljacic. Equivariant self-supervised learning: Encouraging equivariance in representations. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=gKLAAfiytI`.

[11] Nima Dehmamy, Robin Walters, Yanchen Liu, Dashun Wang, and Rose Yu. Automatic symmetry discovery with lie algebra convolutional network. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=NPOWF_ZLfC5`.

[12] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. Learning robust representations via multi-view information bottleneck, 2020. URL `https://arxiv.org/abs/2002.07017`.

[13] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3165–3176. PMLR, 13–18 Jul 2020. URL `http://proceedings.mlr.press/v119/finzi20a.html`.

[14] Marc Finzi, Max Welling, and Andrew Gordon Gordon Wilson. A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3318–3328. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/finzi21a.html`.

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

[16] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bklr3j0cKX`.

[17] Wenlong Ji, Zhun Deng, Ryumei Nakada, James Zou, and Linjun Zhang. The power of contrast for feature learning: A theoretical analysis, 2021.

[18] T. Anderson Keller and Max Welling. Topographic VAEs learn equivariant capsules. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AVWROGUWpu.

[19] Hamza Keurti, Hsiao-Ru Pan, Michel Besserve, Benjamin F. Grewe, and Bernhard Schölkopf. Homomorphism autoencoder – learning group structured representations from observed transitions, 2022. URL https://arxiv.org/abs/2207.12067.

[20] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

[21] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015.

[22] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

[23] Yann LeCun and Ishan Misra. Self-supervised learning: The dark matter of intelligence, Mar 2021. URL https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/.

[24] Jason D Lee, Qi Lei, Nikunj Saunshi, and JIACHENG ZHUO. Predicting what you already know helps: Provable self-supervised learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 309–323. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/02e656adee09f8394b402d9958389b7d-Paper.pdf.

[25] Bo Li, Qili Wang, and Gim Hee Lee. Filtra: Rethinking steerable cnn by filter transform. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6515–6522. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/li21v.html.

[26] Sindy Löwe, Peter O'Connor, and Bastiaan Veeling. Putting an end to end-to-end: Gradient-isolated learning of representations. In *Advances in Neural Information Processing Systems*, 2019.

[27] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018. URL https://arxiv.org/abs/1807.03748.

[28] Dipan K. Pal and Marios Savvides. Non-parametric transformation networks, 2018. URL https://arxiv.org/abs/1801.04520.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[30] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021. URL https://arxiv.org/abs/2102.12092.

[31] Jean-Pierre Serre. *Linear representations of finite groups.*, volume 42 of *Graduate texts in mathematics*. Springer, 1977. ISBN 978-3-540-90190-7.

[32] Yuge Shi, N Siddharth, Philip HS Torr, and Adam R Kosiorek. Adversarial masking for self-supervised learning. In *International Conference on Machine Learning*, 2022.

[33] Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgpugrKPS.

[34] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding, 2019. URL https://arxiv.org/abs/1906.05849.

[35] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[36] Yuandong Tian, Lantao Yu, Xinlei Chen, and Surya Ganguli. Understanding self-supervised learning with dual deep networks, 2020. URL https://arxiv.org/abs/2010.00578.

[37] Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models, 2020. URL https://arxiv.org/abs/2008.10150.

[38] Yao-Hung Hubert Tsai, Yue Wu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Self-supervised learning from a multi-view perspective, 2020. URL https://arxiv.org/abs/2006.05576.

[39] Yao-Hung Hubert Tsai, Tianqin Li, Martin Q. Ma, Han Zhao, Kun Zhang, Louis-Philippe Morency, and Ruslan Salakhutdinov. Conditional contrastive learning with kernel. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=AAJLBoGt0XM.

[40] 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 M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 16451–16467. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/8929c70f8d710e412d38da624b21c3c8-Paper.pdf.

[41] Yifei Wang, Zhengyang Geng, Feng Jiang, Chuming Li, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Residual relaxation for multi-view representation learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=rEBScZF6G70.

[42] 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, 2022.

[43] Maurice Weiler and Gabriele Cesa. General E(2)-Equivariant Steerable CNNs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[44] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 10402–10413, Red Hook, NY, USA, 2018. Curran Associates Inc.

[45] Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/f04cd7399b2b0128970efb6d20b5c551-Paper.pdf.

[46] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7168–7177, 2017.

[47] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=CZ8Y3NzuVzO.

[48] Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pp. 7324–7334. PMLR, 2019.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Discussion Section 5 and Appendix A
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix D
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section 3
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] May be released at a later date
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix B
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 4
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Appendix B
   (b) Did you mention the license of the assets? [Yes] See Appendix B
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix A Limitations

Despite the demonstrated unification of existing methods, and benefits from generalization, we note that this approach is still significantly limited. Specifically, the equivalence between $\mathcal{L}_{\text{A-SSL}}$ and $\mathcal{L}_{\text{H-SSL}}$, and benefits afforded by this equivalence, can only be realized if it is possible to analytically construct a neural network which is equivariant with respect to transformations of interest. Although the field of equivariant deep learning has made significant progress in recent years, state of the art techniques are still restricted to E(n) and continuous compact and connected Lie Groups [4, 13, 14, 43]. We believe in this regard, our analysis sheds some light on the success of methods which perform data augmentation over those which operate directly in feature space in recent literature – it is simply too challenging with current methods to construct models with structured representations for the diversity of transformations needed to induce a sufficient set of invariances for linear separability of classes.

In light of this, we believe that our framework specifically suggests a novel path forward via learned homomorphisms, [9, 11, 18, 19, 28], as mentioned in the Discussion Section 5. Specifically, if new methods can learn symmetries from the data unsupervised, or minimally supervised, our framework yields a natural avenue by which existing SSL advances can be extended to feature-space methods. As a potential interim solution, recent work [2] has additionally shown that it is possible to train 'hybrid' models that leverage both A-SSL and H-SSL objectives simultaneously, suggesting that this limitation may be circumvented by applying H-SSL losses only where beneficial, and otherwise supplementing with input space augmentation.

Appendix B Experiment Details

Model Architectures All models presented in this paper were built using the convolutional layers from the SESN [33] library for consistency and comparability (`https://github.com/ISosnovik/sesn`). For scale equivariant models, we used the set of 6 scales $[1.0, 1.25, 1.33, 1.5, 1.66, 1.75]$. To construct the rotation equivariant backbones, we use only a single scale of $[1.0]$ and augment the basis set with four 90-degree rotated copies of the basis functions at $[0^o, 90^o, 180^o, 270^o]$. These rotated copies thus defined the group dimension. This technique of basis or filter-augmentation for implementing equivariance is known from prior work and has been shown to be equivalent to other methods of constructing group-equivariant neural networks [25]. For translation models, we perform no basis-augmentation, and again define the set of scales used in the basis to a single scale $[1.0]$, thereby leaving only the spatial coordinates of the final feature maps to define the output group.

On MNIST [22], we used a backbone network f composed of three SESN convolutional layers with # channels (32, 64, 128), kernel sizes (11, 7, 7), effective sizes (11, 3, 3), strides (1, 2, 2), padding (5, 3, 3), no biases, basis type 'A', BatchNorm layers after each convolution, and ReLU activations after each BatchNorm. The output of this final ReLU was then considered our $\boldsymbol{z}$ for contrastive learning (with $\mathcal{L}_{A-SSL}$ and $\mathcal{L}_{H-SSL}$) and was of shape $(128, S \times R, 8, 8)$ where S was the number of scales for the experiment (either 1 or 6), and R was the number of rotation angles (either 1 or 4). For experiments where the transformation studied was not translation, we average pool over the spatial dimensions before applying the projection head h to achieve a consistent dimensionality of 128. For classification, an additional SESN convolutional layer was placed on top with kernel size 7, effective size 3, stride 2, and no padding, thereby reducing the spatial dimensions to 1, and the total dimensionality of the input to the final linear classifier to 128.

On CIFAR10 we used a ResNet20 model composed of an initial SESN lifting layer with kernel size 7, effective size 7, stride 1, padding 3, no bias, basis type 'A', and 9 output channels. This lifted representation was then processed by a following SESN convolutional layer of kernel size 7, effective size 3, stride 1, padding 3, no bias, basis type 'A', and 64 output channels. This initial layer was followed by a BatchNorm and ReLU before being processed by three ResNet blocks of output sizes (128, 256, 512) and initial strides of (1, 2, 2). Each ResNet block is composed of 3 SESN Basic blocks as defined here (`https://github.com/ISosnovik/sesn/blob/master/models/stl_ses.py#L19`). The output of the third ResNet block was taken as our $\boldsymbol{z}$ for contrastive learning (again for $\mathcal{L}_{A-SSL}$ and $\mathcal{L}_{H-SSL}$) of shape $(512, S \times R, 7, 7)$. Again, as for MNIST, for experiments where the transformation studied was not translation, we average pool over the spatial dimensions before applying the projection head h to achieve a consistent dimensionality of 512. For classification, the vector $\boldsymbol{z}$ was first max-pooled along the scale/rotation group-axis ($S \times R$), followed by a BatchNorm, a ReLU, and average pooling over the remaining 7×7 spatial dimensions. Finally, we apply BatchNorm to this 512-dimensional vector before applying the non-linear projection head h.

On Tiny ImageNet we use a Resnet20 model which has virtually the same structure as the CIFAR10 model, but instead uses 4 ResNet blocks of output sizes (64, 128, 256, 512) and strides (1, 2, 2, 2). Furthermore, each ResNet block is composed of only 2 BasicBlocks for TIN instead of 3 for CIFAR10. Overall this results in a $\boldsymbol{z}$ of shape $(512, S \times R, 4, 4)$, and a final vector for classification of size 512. We note that we do not include Translation results in Table 1 for Tiny ImageNet precisely because the spatial dimensions of the feature map with this architecture are too small to allow for effective H-SSL training in the settings we used for other methods.

All models used a detached linear classifier for computing the reported downstream classification accuracies, while the Supervised baselines used an attached linear layer (implying gradients with respect to the classification loss back-propagated though the whole network). All models additionally used an attached non-linear projection head h constructed as an MLP with three linear layers. For MNIST these layers have of output sizes (128, 128, 128), while for CIFAR10 and TIN they have sizes (512, 2048, 512). There is a BatchNorm after each layer, and ReLU activations between the middle layers (not at the last layer).

Training Details For training we use the LARS optimizer with an initial learning rate of 0.1, and a batch size of 4096 for all models. We use an NCE temperature (τ) of 0.1, half-precision training, a learning rate warm-up of 10 epochs, a cosine lr-update schedule, and weight decay of 1×10^{-4}. On MNIST we train for 500 epochs and on CIFAR10 and Tiny ImagNet (TIN) we train for 1300 epochs. On average each MNIST run took 1 hour to complete distributed across 8 GPUs, and each CIFAR10/TIN run took 10 hours to complete distributed across 64 GPUs. In total this amounts to roughly 85,000 GPU hours.

Empirical Validation For the experiments in Table 1, we use two different methods for data augmentation, and similarly two different methods for selecting the representations ultimately fed to the contrastive loss for the A-SSL and H-SSL settings.

For A-SSL we augment the input at the pixel level by: randomly translating the image by up to $\pm$ 20% of its height/width (for translation), randomly rotating the image by one of $(0^o, 90^o, 180^o, 270^o)$ (for rotation), or randomly downscaling the image between 0.57 and 1.0 of its original scale. For S-SSL we use no input augmentations.

For both methods we use only a single fiber, meaning the base size $|\boldsymbol{g}_0|$ is 1. For A-SSL, we randomly select the location $\boldsymbol{g}_0$ for each example, but we use the same $\boldsymbol{g}_0$ between both branches. For example, in translation, we compare the feature vectors for two translated images *at the same pixel location*. Similarly, for scale and rotation, we pick a single scale or rotation element to compare for both branches. For H-SSL, we randomly select the location $\boldsymbol{g}$ independently for each example *and independently for each branch*, effectively mimicing the latent operator.

H-SSL Without Structure In Table 2, we use the same overall model architectures defined above (3-layer model or ResNet20), but replace the individual layers with non-equivariant counterparts. Specifically, for the MLP, we replace the convolutional layers with fully connected layers with outputs (784, 1024, 2048). For the convolutional models (denoted CNN ($6 \times CHW$)), we replace the SESN kernels with fully-parameterized, non-equivariant counterparts, otherwise keeping the output dimensionality the same (resulting in the 6 $\times$ larger output dimension).

Furthermore, for these un-structured representations, in the H-SSL setting, we 'emulate' a group dimension to sample 'fibers' from. Specifically, for the MLP we simply reshape the 2048 dimensional output to (16,128), and select one of the 16 rows at each iterations. For the CNN, we similarly use the 6 times larger feature space to sample $\frac{1}{6}^{th}$ of the elements as if they were scale-equivariant.

Parameters of H-SSL For Figure 2 (left), we select patches of sizes from 1 to $|G| - 1$ with no padding. In each setting, we similarly increase the dimensionality of the input layer for the non-linear projection head h to match the multiplicative increase in the dimension of the input representation $\boldsymbol{z}(\boldsymbol{g})$. For the topographic distance experiments (right), we keep a fixed base size of $|\boldsymbol{g}_0| = 1$ and instead vary the maximum allowed distance between randomly sampled pairs g_1 & g_2.

Appendix C Extended Background

Related Work Our work is undoubtedly built upon the the large literature base from the fields equivariant deep learning and self-supervised learning as outlined in Sections 1 and 2. Beyond this background, our work is highly related in motivation to a number of studies specifically related to equivariance in self-supervised learning. Most prior work, however, has focused on the undesired invariances learned by A-SSL methods [39, 47] and on developing methods by which to avoid this through learned approximate equivariance [10, 41]. Our work is, to the best of our knowledge, the first to suggest and validate that the primary reason for the success of feature-space SSL objectives such as DIM(L) [16] and GIM [26] is due to their exploitation of equivariant backbones.

Group-Convolutional Neural Networks As discussed in Section 2, we assume that the backbones used in this work are equivariant with respect to input augmentations, and further that they admit regular representations of those transformations in feature space. In this section we detail how such group-equivariant convolutional neural networks may be constructed via the group-convolution [8]: For a discrete group $\mathcal{G}$, we denote the pre-activation output of a $\mathcal{G}$-equivariant convolutional layer l as $\boldsymbol{z}^l$, with a corresponding input $\boldsymbol{y}^l$. In practice these values are stored in finite arrays with a feature multiplicity equal to the order of the group in each space. Explicitly, $\boldsymbol{z}^l \in \mathbb{R}^{C_{out}\times|G_{out}|}$, and $\boldsymbol{y}^l \in \mathbb{R}^{C_{in}\times|G_{in}|}$ where G_{out} and G_{in} are the set of group elements in the output and input spaces respectively. We use the following shorthand for indexing $\boldsymbol{z}^l(g) \equiv \boldsymbol{z}^{l,:,g} \in \mathbb{R}^{C_{out}}$ and $\boldsymbol{y}^l(g) \equiv \boldsymbol{y}^{l,:,g} \in \mathbb{R}^{C_{in}}$, denoting the vector of feature channels at a specific group element ('fiber'). Then, the value $z^{l,c}(g) \in \mathbb{R}$ of a single output at layer l, channel c and element g is

$$z^{l,c}(g) \equiv [\boldsymbol{y}^l \star \boldsymbol{\psi}^{l,c}](g) = \sum_{h\in G_{in}} \sum_{i}^{C_{in}} y^{l,i}(h)\psi_i^{l,c}(g^{-1}\cdot h) \ , \tag{4}$$

where $\boldsymbol{\psi}_i^{l,c}$ is the filter between the i^{th} input channel (subscript) and the c^{th} output channel (superscript), and is similarly defined (and indexed) over the set of input group elements G_{in}. We highlight that the composition $g^{-1}\cdot h = k \in G_{in}$ is defined by the action of the group and yields another group element by closure of the group. The representation Γ_g and can then be defined as $\Gamma_g[\boldsymbol{z}^l(h)] = \boldsymbol{z}^l(g^{-1}\cdot h)$ for all $l > 0$ when $\mathcal{G}^l_{in} = \mathcal{G}^l_{out} = \mathcal{G}^0_{out}$. From this definition it is straightforward to prove equivariance from: $[\Gamma_g[\boldsymbol{y}^l] \star \boldsymbol{\psi}^l](h) = \Gamma_g[\boldsymbol{y}^l \star \boldsymbol{\psi}^l](h) = \Gamma_g[\boldsymbol{z}^l](h)$. Furthermore, we see that Γ_g is a 'regular representation' of the group, meaning that it acts by permuting features along the group dimension while leaving feature channels intact. Group equivariant layers can then be composed with pointwise non-linearities and biases to yield a fully equivariant deep neural network (e.g. $\boldsymbol{y}_i^{l+1} = \text{ReLU}(\boldsymbol{z}^l + \boldsymbol{b})$ where $\boldsymbol{b} \in \mathbb{R}^{C_{out}}$ is a learned bias shared over the output group dimensions). For $l = 0$, $\boldsymbol{y}^0 = \boldsymbol{x}$, the raw input, and typically $\mathcal{G}^0_{in} = (\mathbb{Z}^2_{HW}, +)$, the group of all 2D integer translations up to height H and width W. $\mathcal{G}^0_{out}$ is then chosen by the practitioner and is typically a larger group which includes translation as a subgroup, e.g. the roto-translation group, or the group of scaling & translations.

DIM(L) in H-SSL In this section we outline precisely how the Deep Infomax Local loss DIM(L) relates to the H-SSL framework proposed in Section 3. Specifically, in Deep InfoMax (DIM(L)) the same general form of the loss function is applied (often called InfoNCE), but the cosine similarity is replaced with a log-bilinear model: $\text{sim}(\boldsymbol{a}, \boldsymbol{b}) = \exp\left(\boldsymbol{a}^T W \boldsymbol{b}\right)$. Additionally, and most importantly to this work, rather than computing the similarity between two differently augmented versions on an image, the loss is applied between different spatial locations of the representation for a single image, again with a head h applied afterwards. If we let $\boldsymbol{g} \sim \mathbb{Z}^2_{HW}$ refer to sampling a contiguous patch from the spatial coordinates of a convolutional feature map, we can write this general Feature-Space InfoMax loss ($\mathcal{L}_{\text{FSIM}}$) as:

$$\mathcal{L}_{\text{FSIM}}(\mathbf{X}) = -\frac{1}{N}\sum_{i}^{N} \mathbb{E}_{\boldsymbol{g}_1,\boldsymbol{g}_2\sim\mathbb{Z}^2_{HW}} \log \frac{\exp\Big(\text{sim}\big(h(\boldsymbol{z}_i(\boldsymbol{g}_1)), h(\boldsymbol{z}_i(\boldsymbol{g}_2))\big)/\tau\Big)}{\sum_{k\neq i}^{N}\sum_{j,l}^{2} \exp\Big(\text{sim}\big(h(\boldsymbol{z}_i(\boldsymbol{g}_j)), h(\boldsymbol{z}_k(\boldsymbol{g}_l))\big)/\tau\Big)} \ . \tag{5}$$

To show that this is equivalent to our $\mathcal{L}_{\text{H-SSL}}$, we see that the randomly sampled spatial patches $\boldsymbol{g}_1, \boldsymbol{g}_2$ can equivalently be described as a single base patch $\boldsymbol{g}_0$ shifted by randomly sampled translations g_1

and g_2. Explicitly,

$$\mathcal{L}_{\text{FSIM}}(\mathbf{X}) = -\frac{1}{N}\sum_{i}^{N} \mathbb{E}_{g_1,g_2\sim G} \log \frac{\exp\Big(\text{sim}\big(h(\boldsymbol{z}_i(g_1^{-1}\cdot \boldsymbol{g}_0)), h(\boldsymbol{z}_i(g_2^{-1}\cdot \boldsymbol{g}_0))\big)/\tau\Big)}{\sum_{k\neq i}^{N}\sum_{j,l}^{2}\exp\Big(\text{sim}\big(h(\boldsymbol{z}_i(g_j^{-1}\cdot \boldsymbol{g}_0)), h(\boldsymbol{z}_k(g_l^{-1}\cdot \boldsymbol{g}_0))\big)/\tau\Big)} \ . \tag{6}$$

Thus, we see that Feature-Space InfoMax losses are included in our framework, and can therefore be seen to be equivalent to input-augmentation based losses with an equivariant backbone, where the set of augmentations is limited to the translation group $G \equiv \mathbb{Z}^2_{HW}$, and the $\boldsymbol{g}_0$ base size is a single spatial coordinate ($|\boldsymbol{g}_0| = 1$) rather than the size of the full representation ($|\boldsymbol{g}_0| = |G|$).

Appendix D Broader Impact

This work is primarily related to understanding and improving self-supervised learning – a training method for deep neural networks which is able to leverage large amounts of unlabeled data from the internet, making it one of the most used methods for state of the art image and text generative models today [29, 30]. Such models have significant broader impact and potential negative consequences which are beyond the scope of this work. We refer readers to discussions of those paper for further information. Specifically, this work aims to improve such SSL techniques, thereby inheriting the broader impact of these models.