Why Do Self-Supervised Models Transfer? On the Impact of Invariance on Downstream Tasks

Linus Ericsson[1], **Henry Gouk**[1], **Timothy M. Hospedales**[1,2]
[1]University of Edinburgh [2]Samsung Research
{linus.ericsson,henry.gouk,t.hospedales}@ed.ac.uk

Abstract

Self-supervised learning is a powerful paradigm for representation learning on unlabelled images. A wealth of effective new methods based on instance matching rely on data-augmentation to drive learning, and these have reached a rough agreement on an augmentation scheme that optimises popular recognition benchmarks. However, there is strong reason to suspect that different tasks in computer vision require features to encode different (in)variances, and therefore likely require different augmentation strategies. In this paper, we measure the invariances learned by contrastive methods and confirm that they do learn invariance to the augmentations used and further show that learned invariances strongly affect downstream task performance and confirm that different downstream tasks benefit from polar opposite (in)variances, leading to performance loss when the standard augmentation strategy is used. Finally, we demonstrate that a simple fusion of representations with complementary invariances ensures wide transferability to all the diverse downstream tasks considered.

1 Introduction

Self-supervised learning (SSL) has made rapid progress in representation learning, with contrastive methods driven by semantics-preserving data augmentation achieving particular success in computer vision [12, 4]. In this paradigm, the properties and efficacy of the learned representation are largely determined by the augmentation distribution used during self-supervision. To this end a rough consensus has emerged among many state of the art methods as to a good default distribution that leads to strong performance on the downstream benchmarks, especially on the ubiquitous ImageNet object recognition benchmark [6]. For example, image cropping, flipping, colour perturbation and blurring, are widely applied [12, 5, 7]. However, if augmentation leads to invariance to the corresponding transformation, then we should ask: do our self-supervised algorithms provide the right invariances for diverse downstream tasks of interest? For example, while an object categorisation task might benefit from pose invariance, other tasks such as pose estimation may require strong spatial sensitivity. If different tasks require contradictory (in)variances, using a single default data augmentation scheme for all may provide sub-optimal performance for some tasks.

To investigate this issue, we group augmentations into two categories: *spatial* and *appearance*. Using a representative state of the art contrastive learner MoCo-v2+ResNet50 [5], we train models exclusively with spatial-style and appearance-style augmentations and compare them to the model produced by the default augmentation scheme. In particular, we evaluate their resulting invariances to synthetic transforms and their performance on a suite of diverse real-world downstream tasks.

Based on the experimental design outlined above, we attempt to better understand *why contrastive SSL works* by answering the following specific questions, among others, with associated results summarised in Figure 1. **Q1** *Given that there are multiple types invariances of potential interest to learn. Is there a trade-off between learning different types of invariances?* A2: Yes. Promoting spatial-style invariances decreases appearance-style ones and vice-versa (Fig. 1 left). We also show that all existing state-of-the-art learners suffer from this trade-off. **Q2** *Do different downstream tasks of interest benefit from different invariances?* A3: Yes. Across a suite of downstream tasks, we see that recognition-style

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

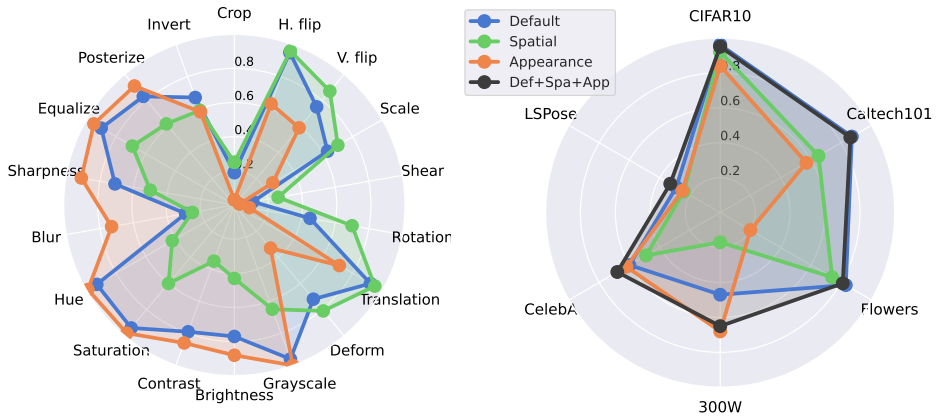

Figure 1: Our Spatial and Appearance models lead to strong spatial and colour/texture invariance, respectively (left). Simple feature fusion (black) dominates specialised models, as well as state of the art 'default' augmentation, providing more consistent performance across all tasks (right).

tasks prefer a representation trained on default or spatial-style augmentations, while pose-related tasks benefit from appearance-style augmentations. In particular, default augmentations [5] under-perform in pose-related tasks (Fig. 1 right, blue line on LSPose/CelebA/300W). **Q3** *Given that different tasks prefer polar-opposite augmentations, is there a simple way to achieve high performance across all tasks?* A4: Yes. Simple fusion of multiple representations tuned for different (in)variances leads to consistent strong performance across all tasks considered (Fig. 1 right, black line).

Related Work: Though data augmentation has become even more vital in practical self-supervision, understanding its role has lagged behind practical engineering lore. Self-supervised contrastive learners with strong augmentation have been shown to learn occlusion-invariant representations, but not to capture viewpoint and category instance invariance [21]. There exists approaches for integrating augmentation-aware information into learned representations by; separating different information into different features, i.e. colour, orientation etc. [26]; or predicting the difference of augmentation parameters [15]. While in this paper, our main focus is on gaining insights into the effects of data augmentation on downstream performance, our fusion of complementary features serves as another example of this.

2 Methods

Our main focus is on analysing the properties of self-supervised models pre-trained with different augmentation strategies. In particular, we choose MoCo-v2 [5] as a representative self-supervised learner that is widely used and near state-of-the-art. MoCo-v2 matches images with their augmented counterparts, while using negative pairs in a contrastive loss to encourage feature dissimilarity between semantic objects, and to avoid features all collapsing to the same vector. We pre-train three models using MoCo-v2 [5] with ResNet50 architectures [11] on ImageNet [6] for 200 epochs. **Default:** The default [5, 3, 4, 16, 10] model uses the standard array of data augmentations, which includes crops, horizontal flips, color jitter, grayscale and blur. **Spatial:** The spatial model uses only the spatial subset of default augmentations, including crops and horizontal flips. By learning invariance to these spatial transforms, the model has to put larger focus on colour and texture. **Appearance:** The appearance model uses only the appearance-based augmentations of color jitter, grayscale and blur. As baselines, we also compare a CNN with **Random** weights, and one pre-trained by **Supervised** learning. Details on the augmentations used in pretraining each model is summarised in Tab. A.1.

3 Measuring invariances

In this work we wish to establish whether downstream tasks benefit from different feature invariances. Thus, we must first quantify the invariances of our pre-trained models. We use two measures of invariance in our experiments, Mahalanobis distance and cosine similarity (full details in Appendix). A further set of measures are reported in the Appendix with results supporting those in the main paper.

Table 1: ImageNet pre-trained ResNet50 with MoCo-v2 (200 epochs) evaluated on invariances to transforms on 1000 ImageNet validation images. Top group: Mahalanobis distance where a low value means strong invariance. Bottom group: cosine similarity in a normalised feature space where a value close to 1 means strong invariance. Column colours indicate the type of invariance evaluated and row colours indicate the augmentation expected to lead to high-performing specialised models. Similarity results within {Default, Spatial, Appearance} that are statistically significantly the best are annotated with a •. The hypothesis testing procedure used to determine this is described in the supplemental material.

		Crop	H flip	V flip	Scale	Shear	Rotat.	Transl.	Deform	Graysc.	Brightn.	Contrast	Saturat.	Hue	Blur	Sharpn.	Equalize	Posterize	Invert
Distance	↓																		
	Random	69.40	34.14	35.35	67.03	69.57	71.65	56.33	65.28	22.81	73.25	59.03	46.63	42.39	49.17	52.59	27.46	27.46	32.88
	Supervised	**57.44**	12.33	24.07	40.37	63.93	47.67	22.51	34.25	19.87	40.43	37.05	26.61	35.95	54.92	42.15	17.44	22.74	27.32
	Default	58.72	9.92	21.30	35.75	56.58	43.49	16.45	30.16	7.78	26.07	25.66	12.17	13.72	65.84	31.07	12.81	17.95	**24.71**
	Spatial	59.43	**8.17**	**15.57**	**32.05**	**56.50**	**32.93**	**13.85**	**26.08**	26.58	46.25	61.67	39.34	47.33	63.37	48.83	25.41	38.46	28.95
	Appearance	64.35	27.52	29.05	56.33	71.81	62.49	33.14	52.20	**2.57**	**19.71**	**22.84**	**6.98**	**5.77**	**30.38**	**16.86**	**9.55**	**12.18**	29.24
Similarity	↑																		
	Random	0.03	0.56	0.54	0.16	0.04	0.07	0.40	0.20	0.81	0.17	0.52	0.59	0.60	0.48	0.51	0.68	0.70	0.52
	Supervised	0.18	0.92	0.71	0.57	0.11	0.43	0.87	0.69	0.81	0.54	0.64	0.79	0.64	0.29	0.55	0.84	0.77	0.64
	Default	0.19	0.95	0.75	0.63	0.11	0.45	0.92	0.72	0.96	0.77	0.79	0.94	0.93	0.29	0.71	0.90	0.83	**0.67**
	Spatial	**0.25•**	**0.96**	**0.87•**	**0.70•**	**0.26•**	**0.70•**	**0.95•**	**0.81•**	0.65	0.43	0.35	0.60	0.42	0.25	0.50	0.69	0.62	0.59
	Appearance	0.03	0.63	0.59	0.26	0.03	0.09	0.71	0.33	**1.00**	**0.88•**	**0.86•**	**0.98**	**0.99•**	**0.73•**	**0.91•**	**0.95•**	**0.91•**	0.58

Setup: We focus on task-agnostic metrics of invariances introduced in Sec C.1. Other extrinsic measures of invariance like identifiability/classification performance under different transformations are inherently biased towards that task. We therefore use invariance metrics that apply to feature vectors directly. We evaluate our Default, Spatial and Appearance methods on 1,000 images from the ImageNet (ILSVRC12) validation set [6] against a wider array of synthetic transformations than used for training (Tab A.1), but still group these into appearance and spatial-style transforms. We compute our measurements between augmented and unaugmented images, averaged over all images considered.

Results: The results in Tab. 1 evaluate the invariance of different transformations at test-time (columns) for the different pre-trained models (rows). Using a method described in the supplemental material, we carry out statistical hypothesis tests to determine which of the Default, Appearance, and Spatial models reliably exhibit the most invariance, as measured by the similarity metric. Statistically significant results (at the 95% confidence level) are marked with a •. We make the following observations. For spatial transformations like rotation and translation, the Spatial model is the most invariant, due to its use of such augmentations during pre-training. Likewise, the Appearance model has the strongest invariance to transformations in colour and texture, except for the invert transform. The Default model tends to fall in between the two specialised models suggesting strong invariance to any one transformation is traded off for a reasonable variance across the board. The Random model tends to have the highest variance.

While the Spatial model has very low variance to spatial transforms, it has a high variance to colour and texture. Its sensitivity to these transforms is available for solving tasks that depend on colour or texture. Likewise, the Appearance model is sensitive to spatial information which it could use to solve spatially sensitive tasks. In fact, since the Appearance model is more spatially sensitive than the Default model, it might achieve better performance on such tasks. We investigate this in Sec. 4. Overall the results confirm that invariances are indeed learned by contrastive learning with corresponding augmentations. Furthermore, augmentations do tend to increase invariance to other transforms in the corresponding appearance/spatial family, rather than only the specific subset used for training.

4 Do Downstream Tasks Prefer Different Invariances?

We have showed how contrastive training under data augmentation learns invariance to synthetic transformations. We also confirmed the appearance sensitivity of the Spatial model and the spatial sensitivity of the Appearance model. In terms of real-world benchmarks, self-supervised methods are widely evaluated on ImageNet recognition, with the literature having a lesser focus and lack of consistency in evaluation of other non-recognition tasks. Since the default augmentations are largely chosen to optimise recognition benchmarks, there is a chance that it may be overfit to these tasks and perform less well on others. We therefore investigate how learned invariances affect a more diverse suite of real downstream tasks of interest, hypothesising that different features may be preferred, depending on the (in)variance needs of each downstream task.

Experimental Details: Our suite of downstream tasks consists of object recognition on standard benchmarks **CIFAR10** [14], **Caltech101** [8] and **Flowers** [19]; as well as a set of spatially sensitive tasks including facial landmark detection on **300W** [22] and **CelebA** [18], and pose estimation on **Leeds Sports Pose** [13]. We freeze the backbones and extract features from just after the average pooling layer of the ResNet50 architectures. We fit a ridge or logistic regression model on these features, depending on the task in question. To tune the $\ell 2$ regularisation value we perform 5-fold cross-validation over a grid of 45 logarithmically spaced values between 10^{-6} to 10^{5}, following [7, 3].

Table 2: mean and standard deviation of 5-fold cross-validation on all data for each downstream task. Random refers to a randomly initialised feature extractor and '+' refers to feature concatenation. On the left datasets we report the classification accuracy and on the right the R^2 regression metric. Row colours indicate whether Appearance or Spatial turned out better for the given task.

	CIFAR10	Caltech101	Flowers	300W	CelebA	LSPose	Avg.
Random	0.55 ± 0.004	0.25 ± 0.008	0.21 ± 0.009	0.24 ± 0.024	0.47 ± 0.002	0.10 ± 0.007	0.30 ± 0.009
Supervised	$\mathbf{0.98 \pm 0.001}$	$\mathbf{0.90 \pm 0.005}$	$\mathbf{0.86 \pm 0.007}$	0.17 ± 0.028	0.49 ± 0.002	0.20 ± 0.015	0.60 ± 0.010
Default	0.96 ± 0.002	0.87 ± 0.006	0.83 ± 0.004	0.47 ± 0.014	0.60 ± 0.002	$\mathbf{0.29 \pm 0.025}$	$\mathbf{0.67 \pm 0.009}$
Spatial	0.92 ± 0.003	0.65 ± 0.008	0.74 ± 0.010	0.17 ± 0.030	0.49 ± 0.001	0.24 ± 0.020	0.54 ± 0.013
Appearance	0.84 ± 0.003	0.57 ± 0.007	0.20 ± 0.009	$\mathbf{0.68 \pm 0.018}$	$\mathbf{0.62 \pm 0.003}$	0.25 ± 0.021	0.53 ± 0.010
3×Default	$\mathbf{0.96 \pm 0.004}$	0.87 ± 0.004	$\mathbf{0.83 \pm 0.007}$	0.52 ± 0.012	0.67 ± 0.001	0.31 ± 0.016	0.69 ± 0.007
Default(×3)	$\mathbf{0.96 \pm 0.002}$	$\mathbf{0.88 \pm 0.003}$	0.82 ± 0.009	0.42 ± 0.030	0.65 ± 0.005	0.31 ± 0.028	0.67 ± 0.013
Spa+App	0.95 ± 0.002	0.74 ± 0.009	0.68 ± 0.005	0.62 ± 0.021	0.64 ± 0.002	0.29 ± 0.007	0.65 ± 0.008
Def+Spa+App	0.95 ± 0.003	0.86 ± 0.009	0.81 ± 0.006	$\mathbf{0.65 \pm 0.020}$	$\mathbf{0.68 \pm 0.002}$	$\mathbf{0.33 \pm 0.010}$	$\mathbf{0.71 \pm 0.008}$

We report the mean and standard deviation for the hyperparameter choice with highest mean. The performance is reported as accuracies (between 0 and 1) for classification tasks and R^2 values for regression tasks. For comparison we also evaluate random and supervised backbones.

Results: Table 2 shows the linear readout performance on all tasks considered. On the datasets most similar to ImageNet: CIFAR10, Caltech101 and Flowers, the Default or Supervised models achieve the highest classification accuracy, followed by the Spatial and then the Appearance model. On the spatially sensitive tasks the Appearance model outperforms the Spatial model substantively, with the Appearance model performing best overall on 300W. These results show some evidence that the Default (and to a lesser extent Spatial model) model is well suited for object recognition on ImageNet-like datasets, but both are weak in comparison to a model with more spatial sensitivity when solving the pose-related tasks. Overall this supports the hypothesis that different (in)variances are required for best performance on different types of tasks.

Improving Performance Through Feature Fusion: Our analysis has shown that different real-world tasks prefer different invariances. The default model tries to satisfy them all by using a mix of augmentations to obtain a moderate amount of invariance to all transformations (Sec 3), but appearance/spatial specialised features can be better for particular tasks (Table 2, above). We therefore explore whether a fusion of specialised features can perform competitively across the board. In particular we explore Spatial-Appearance fusion, as well as three way Default-Spatial-Appearance fusion.

Experimental Details: The evaluation follows the setup above, but as our fused representations have higher dimensionality, we shift the $\ell 2$ search space for Spa+App to 10^{-5} to 10^{6} and Def+Spa+App 10^{-4} to 10^{7}. To compare the concatenated features of Def+Spa+App, we evaluate a second Default model with a $3\times$ wider architecture – ResNet50($\times 3$) – trained with MoCo-v2 for 200 epochs on ImageNet like our other models and uses the same hyperparameter search space as Def+Spa+App. A final baseline of three fused separately trained Default models forms $3\times$Default.

Results: Table 2 (bottom) shows that the Spa+App model and Def+Spa+App fusion models perform strongly across the board. The $3\times$Default and Default($\times 3$) models are unsurprisingly best for recognition tasks, but only by a small margin; while the 3-way Def+Spa+App fusion is dramatically better for 300W, and the most consistent performer. This result is noteworthy, as a goal of SSL is to provide a single feature extractor that provides excellent performance for diverse downstream tasks. We have shown the Default model falls down in this regard, but our fused feature performs well consistently. We therefore recommend it to practitioners who want a single feature with which to perform diverse tasks.

5 Discussion

We have performed the first thorough evaluation of SSL in terms of augmentations used for training, and resulting downstream invariance and task impact. Our main findings are: (1) CNNs trained contrastively learn invariances corresponding to augmentations, and specialising CNNs to particular spatial/appearance augmentations can lead to greater corresponding invariances (Table 1). (2) Different real-world downstream tasks benefit from different invariances (Table 2), and invariance-specialised features can outperform the standard default augmentation, e.g., for spatially sensitive tasks. (3) Fusing different specialised extractors provides a consistently high performing strategy (Table 2). This outperforms the Default model on pose related tasks, suggesting it was over-tuned for recognition. Our feature ensemble strategy is promising for providing high performance *general purpose* real-world features. Based on these results we encourage the SSL community to evaluate on more diverse downstream task types.

References

[1] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*, 2020. URL http://arxiv.org/abs/2006.09882.

[2] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, 2020.

[4] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big Self-Supervised Models are Strong Semi-Supervised Learners. In *NeurIPS*, 2020. URL http://arxiv.org/abs/2006.10029.

[5] X. Chen, H. Fan, R. Girshick, and K. He. Improved Baselines with Momentum Contrastive Learning. *arXiv*, 2020. URL http://arxiv.org/abs/2003.04297.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[7] L. Ericsson, H. Gouk, and T. M. Hospedales. How Well Do Self-Supervised Models Transfer? In *CVPR*, 2021. URL http://arxiv.org/abs/2011.13377.

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshops*, 2004. ISBN 9780000000002. doi: 10.1109/CVPR.2004.383.

[9] I. J. Goodfellow, Q. V. Le, A. M. Saxe, H. Lee, and A. Y. Ng. Measuring Invariances in Deep Networks. In *NeurIPS*, 2009.

[10] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. In *NeurIPS*, 2020. URL http://arxiv.org/abs/2006.07733.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. ISBN 9781467388504. doi: 10.1109/CVPR.2016.90.

[12] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*, 2019. URL http://arxiv.org/abs/1911.05722.

[13] S. Johnson and M. Everingham. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In *BMVC*, 2010.

[14] A. Krizhevsky and G. Hinton. Learning Multiple Layers of Features from Tiny Images. *arXiv*, 2009.

[15] H. Lee, K. Lee, K. Lee, H. Lee, and J. Shin. Improving Transferability of Representations via Augmentation-Aware Self-Supervision. In *NeurIPS*, 2021. ISBN 9781713845393. doi: 10.48550/arxiv.2111.09613. URL https://arxiv.org/abs/2111.09613v1.

[16] J. Li, P. Zhou, C. Xiong, R. Socher, and S. C. H. Hoi. Prototypical Contrastive Learning of Unsupervised Representations. In *ICLR*, 2021. URL http://arxiv.org/abs/2005.04966.

[17] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. doi: 10.1007/978-3-319-10602-1{_}48.

[18] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[19] M. E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, 2008. ISBN 9780769534763. doi: 10.1109/ICVGIP.2008.47.

[20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[21] S. Purushwalkam and A. Gupta. Demystifying Contrastive Self-Supervised Learning: Invariances, Augmentations and Dataset Biases. *arXiv*, 2020. URL http://arxiv.org/abs/2007.13916.

[22] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 Faces In-The-Wild Challenge. *Image and Vision Computing*, 2016. ISSN 02628856. doi: 10.1016/J.IMAVIS.2016.01.002. URL https://dl.acm.org/doi/abs/10.1016/j.imavis.2016.01.002.

[23] G. Van Horn, E. Cole, S. Beery, K. Wilber, S. Belongie, and O. Mac Aodha. Benchmarking Representation Learning for Natural World Image Collections. In *CVPR*, 2021.

[24] T. Wang and P. Isola. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *ICML*, 2020. URL http://arxiv.org/abs/2005.10242.

[25] Y. Wang, Q. Zhang, Y. Wang, J. Yang, and Z. Lin. Chaos is a Ladder: A New Understanding of Contrastive Learning. In *ICLR*, 2022. URL http://arxiv.org/abs/2102.06866.

[26] T. Xiao, X. Wang, A. A. Efros, and T. Darrell. What Should Not Be Contrastive in Contrastive Learning. In *ICLR*, 2021. URL https://arxiv.org/abs/2008.05659v2.

[27] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. In *ICML*, 2021. URL https://arxiv.org/abs/2103.03230v3.

Table A.1: Augmentations used during pre-training of our Spatial and Appearance models, along with the standard default augmentations [5]. The color jitter augmentation is a combination of individual jitter in brightness, contrast, saturation and hue.

	Resized crop	Horizontal flip	Color jitter	Grayscale	Blur
Default	✓	✓	✓	✓	✓
Spatial	✓	✓			
Appearance			✓	✓	✓

A Data augmentation for pre-training

We use the strong contrastive baseline MoCo-v2 [5] for pre-training our models and use its augmentation policy as our basis for our experiments. Our Default model is trained using the full set of augmentations detailed below in PyTorch [20] code.

```
transforms.Compose([
    transforms.RandomResizedCrop(224, scale=(0.2, 1.)),
    transforms.RandomApply([
        transforms.ColorJitter(0.4, 0.4, 0.4, 0.1)
    ], p=0.8),
    transforms.RandomGrayscale(p=0.2),
    transforms
    .RandomApply([moco.loader.GaussianBlur([.1, 2.])], p=0.5),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms
    .Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

Listing A.1: Default augmentation policy

The Spatial model only uses resized crops and horizontal flips.

```
transforms.Compose([
    transforms.RandomResizedCrop(224, scale=(0.2, 1.)),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms
    .Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

Listing A.2: Spatial augmentation policy

And finally the Appearance model uses grayscale, color jitter and blurring.

```
transforms.Compose([
    transforms.Resize(224),
    transforms.CenterCrop(224),
    transforms.RandomApply([
        transforms.ColorJitter(0.4, 0.4, 0.4, 0.1)
    ], p=0.8),
    transforms.RandomGrayscale(p=0.2),
    transforms
    .RandomApply([moco.loader.GaussianBlur([.1, 2.])], p=0.5),
    transforms.ToTensor(),
    transforms
    .Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

Listing A.3: Appearance augmentation policy

Table A.1 summarises the augmentations used by each model.

B Synthetic transforms

For Table 1 we compute the invariance metrics based on the following synthetic transforms.

- Resized crop: We generate 256 crops with anchor points positioned between 0 and 64 pixels from the top and left of the original image (of size 256×256). A crop is between 25% and 75% of the image in both height and width. After, the crop is resized to 224×224.
- Horizontal flip: A single horizontally flipped image is generated in addition to the unaugmented image.
- Vertical flip: A single vertically flipped image is generated in addition to the unaugmented image.
- Scale: We generate 256 images rescaled between $\frac{1}{4}$ to 2 times its original size.
- Shear: We generate 256 images with horizontal and vertical shear of -160 to 160 degrees.
- Rotation: We generate 256 images with rotation angles between 0 and 360 degrees.
- Translation: We generate 256 images with horizontal and vertical translation of -16 to 16 pixels.
- Deform: We generate 256 images with the ElasticTransform function of the albumentations package, with σ between 10 and 50.
- Grayscale: A single grayscale image is generated in addition to the unaugmented image.
- Brightness: We generate 256 images where brightness is $\frac{1}{4}$ to 5 times its original value.
- Contrast: We generate 256 images where contrast is $\frac{1}{4}$ to 5 times its original value.
- Saturation: We generate 256 images where saturation is $\frac{1}{4}$ to 5 times its original value.
- Hue: We generate 256 images where hue is set to one of 5 values spread over the colour circle.
- Blur: We generate 256 images with Gaussian blur where σ is between 10^{-5} to 20.
- Sharpness: We generate 256 images where the sharpness is adjusted by a factor of 1 to 30.
- Equalize: A single image with an equalized histogram is generated in addition to the unaugmented image.
- Posterize: We generate seven images by reducing the number of bits for each colour channel to 1-7 in addition to the full 8-bit unaugmented image.
- Invert: A single image with inverted colours is generated in addition to the unaugmented image.

C Invariance measurement details and further results

C.1 Measuring invariances

A key contribution of this paper is measuring the degree of invariance to various synthetic and real-world transformations. Previous studies have focused on measuring invariance at the neuronal level [9]. We consider instead the invariance properties of entire feature vectors under input transformations. To this end we explore two metrics.

Mahalanobis distance: A vector can be said to be invariant to a transformation if it remains unchanged after applying that transformation. We can measure the invariances of a feature extractor model by looking at how much its feature vectors change under different transformations. Given a pre-trained feature extractor f, whose feature space has a covariance of $\mathbf{\Sigma}$, a transformation t_ϕ parameterised by ϕ and an image x, we compute the variance of f to transformation t_ϕ as the Mahalanobis distance

$$l_f^{t_\phi}(x) = \sqrt{\Big(f(x) - f(t_\phi(x))\Big)\mathbf{\Sigma}^{-1}\Big(f(x) - f(t_\phi(x))\Big)^T} = \|Gf(x) - Gf(t_\phi(x))\|_2 \tag{C.1}$$

where $\mathbf{\Sigma}^{-1} = GG^T$, and G can be computed using the Cholesky decomposition.

Cosine similarity: Alternatively, we can measure the angle instead of the distance by first standardising the vectors using the mean feature, $\bar{f}$, of f and G, giving us

$$z = G\Big(\bar{f} - f(x)\Big), \qquad z_{t_\phi(x)} = G\Big(\bar{f} - f(t_\phi(x))\Big), \tag{C.2}$$

and then using cosine similarity to measure the angle between features, giving us an invariance measure of

$$l_f^{t_\phi}(x) = \frac{z \cdot z_{t_\phi(x)}}{\|z\| \|z_{t_\phi(x)}\|}. \tag{C.3}$$

The distance or similarity is computed over a range of transformation parameters, $\phi \in \Phi$—e.g. from 0° to 360° for rotation. Additionally, we average over all images in a dataset D. The global measurement is then

$$L_f^{T_\Phi}(D) = \frac{1}{|D||\Phi|} \sum_{x \in D} \sum_{\phi \in \Phi} l_f^{t_\phi}(x), \tag{C.4}$$

where $T_\Phi = \{t_\phi\}_{\phi \in \Phi}$. A model with zero Mahalanobis distance (variance) to a transformation is invariant to it. Likewise, a model with maximum cosine similarity is invariant.

Table C.1: Top left: alignment metric on synthetic transforms, and top right: uniformity on the three augmentation families. Bottom left: alignment metric on real-world transforms and, bottom right: average confusion ratio (ACR) on the three augmentation families (C=10, k=10).

	Alignment																
	Crop	H flip	V flip	Scale	Shear	Rotat.	Transl.	Deform	Graysc.	Bright.	Contr.	Satur.	Hue	Blur	Sharpn.	Equal.	Poster.	Invert
Random	1.94	0.88	0.93	1.66	1.92	1.86	1.32	1.61	0.38	1.66	0.97	0.81	0.80	1.04	0.97	0.61	0.56	0.98
Supervised	1.63	0.16	0.58	0.86	1.77	1.14	0.27	0.61	0.38	0.92	0.72	0.43	0.72	1.42	0.91	0.31	0.45	0.72
Default	1.63	0.11	0.53	0.74	1.77	1.09	0.15	0.50	0.09	0.60	0.34	0.12	0.14	1.38	0.60	0.20	0.34	**0.67**
Spatial	**1.50**	**0.07**	**0.24**	**0.58**	**1.48**	**0.61**	**0.10**	**0.36**	0.78	1.28	1.28	0.91	1.27	1.52	1.06	0.62	0.76	0.82
Appearance	1.95	0.75	0.82	1.48	1.94	1.81	0.58	1.30	**0.01**	**0.33**	**0.23**	**0.06**	**0.03**	**0.51**	**0.20**	**0.10**	**0.17**	0.84

	Uniformity		
	Default	Spatial	Appearance
Supervised	3.93	3.87	3.84
Default	**3.84**	3.83	3.67
Spatial	3.86	**3.78**	3.88
Appearance	3.98	3.98	**3.61**

	Flickr1024	COIL100	ALOI	ALOT	ALOI	ALOT	ALOI	ALOT	ExposureErrors	RealBlur
	Stereo	Pose/Scale	Viewp.	Viewp.	Illumin.	Illumin.	Temp.	Temp.	Exposure	Blur
Random	0.75	1.17	1.04	1.51	0.85	1.13	0.20	0.55	1.19	0.17
Supervised	0.20	0.60	0.72	0.66	0.47	0.51	0.06	0.21	0.33	0.22
Default	**0.12**	0.49	0.46	0.60	0.33	0.48	0.04	0.07	0.20	0.20
Spatial	**0.12**	**0.32**	**0.24**	**0.56**	**0.20**	**0.40**	0.11	0.44	0.62	0.36
Appearance	0.52	0.92	0.73	1.37	0.41	0.78	**0.01**	**0.02**	**0.09**	**0.04**

	ACR		
	Default	Spatial	Appearance
Random	0.02	0.05	0.44
Supervised	0.69	0.86	0.86
Default	**0.84**	0.86	**0.90**
Spatial	0.32	**0.89**	0.45
Appearance	0.35	0.35	**0.90**

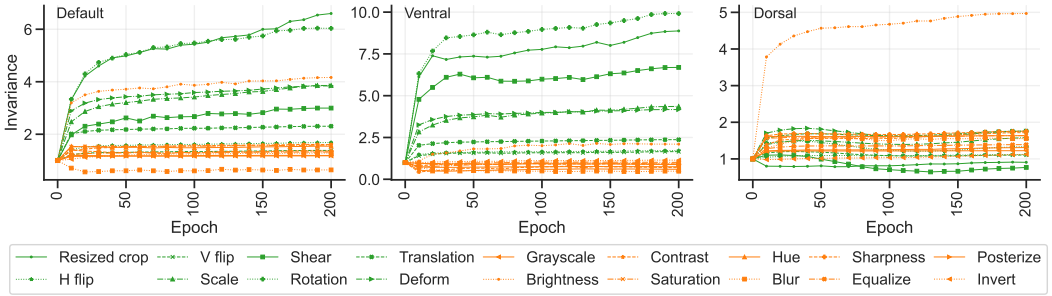

Figure C.1: Invariances as measured by cosine similarity during pre-training. Different invariances are learned at different speeds. After 200 epochs many of the invariances are still steadily increasing, suggesting longer training helps achieve stronger invariances.

C.2 Alternative metrics

We note that there are alternative metrics for (in)variance such as alignment and uniformity [24] and ACR [25]. We provide results with these metrics here in Table C.1, and they lead to the same conclusions reported in the main paper.

C.3 How do invariances change during pre-training?

It is clear from the results above that the use of augmentations leads to invariances to those transforms. But how do these invariances change as the model learns? Figure C.1 shows how the invariances evolve during pre-training. The results echo those in Table 1, showing the Appearance and Spatial models quickly specialise to greater corresponding invariances than the default model which has a moderate invariance to all transforms. In terms of the temporal dynamics, while some invariances stabilise quickly, other are continuing to increase at 200 epochs. This suggests that longer training may lead to further increases in invariance, and may explain why several state of the art learners achieve best performance with a very large number of iterations [1, 3].

C.4 Do learned invariances hold for uncurated images?

We also evaluate the invariance to synthetic transforms on 100 images from the MS COCO val2017 set [17] and iNaturalist 2021 validation set [23]. As can be seen in Fig. C.2, the invariances of all models match those in Fig. 1 (top) on ImageNet, showing that these learned invariances are not limited to highly curated object-centric images, but also to the cluttered images of COCO and the in-the-wild nature of iNaturalist.

D Correlations

D.1 Between synthetic and downstream tasks

Figure D.1 shows the correlations between synthetic invariances and downstream model performances. We see a block-like structure, where spatial invariances correlate with classification tasks and

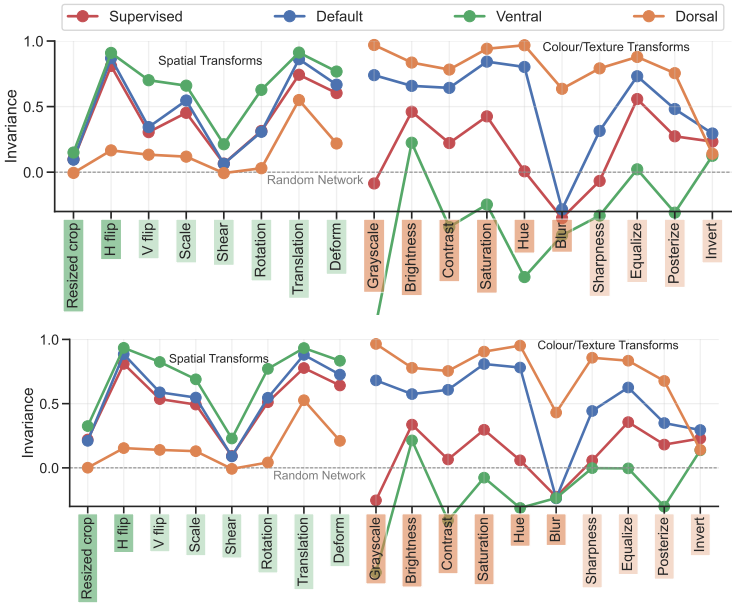

Figure C.2: Invariances to synthetic transforms hold for cluttered images from COCO (top) and in-the-wild images from iNaturalist (bottom).

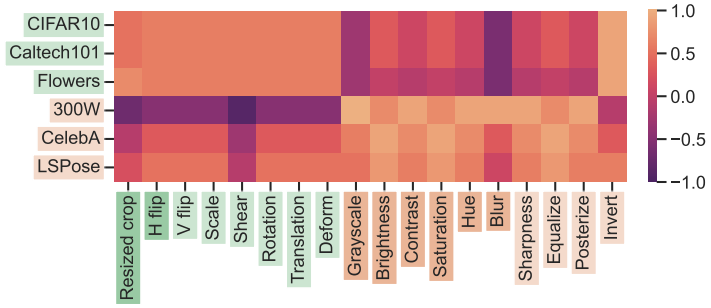

Figure D.1: Rank correlation between synthetic invariances and downstream tasks of our set of models in Table 2 (top). A high value means that learning a stronger invariance to the transform is highly correlated with getting better at the task.

appearance invariances correlate with regression tasks. The task that stands out the most is 300W, where invariance to spatial transforms is very destructive for performance. The invert transform is also here an outlier as it correlates strongly with classification.

E Downstream evaluation

For all datasets, we use the full sets of images (combining train, val and test sets) for 5-fold cross-validation. For classification datasets we stratify the folds to ensure class balance. On Caltech101 and Flowers this means that we randomly select 30 and 20 images per class, respectively, to form the train set in the current fold and test on the rest. On 300W [22] we use images both the indoor and outdoor sets. For CIFAR10 we report accuracy and for Caltech101 and Flowers, mean per-class accuracy. On 300W and CelebA we perform facial landmark regression and report the R^2 regression metric and for Leeds Sports Pose we perform pose estimation and report R^2.

We follow the evaluation of [3], but additionally perform 5-fold cross-validation. We extract features from a frozen backbone and, for classification datasets (CIFAR10, Caltech101, Flowers) fit a logistic

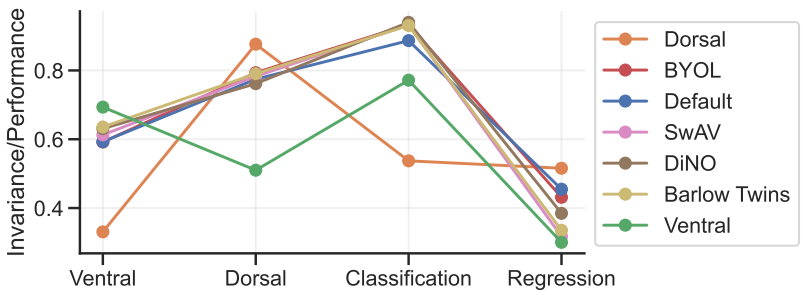

Figure E.1: A parallel coordinates plot demonstrating the impact of different levels of spatial/appearance invariance (measured in cosine similarity) on the performance of downstream problems for different types of tasks (measured in accuracy or R^2).

regression classifier and for regression datasets (300W, CelebA and Leeds Sports Pose) we fit ridge regression. For both settings, the $\ell 2$ hyperparameter search range is 45 logarithmically spaced values between 10^{-6} to 10^{5}. For the larger and fused models we shift the range to 10^{-5} to 10^{6} for Spa+App and 10^{-4} to 10^{7} for Def+Spa+App, $3\times$Default and Default($\times 3$).

E.1 Comparison of other State of the Art

Our experiment so far focused on retraining a representative MoCo model with different augmentations. In the final experiment, we broaden our scope and evaluate a suite of existing pre-trained methods on our suite of tests for spatial and appearance invariances (Section **??**) and the downstream tasks studied in Section 4. We evaluate BYOL [10], SwAV [1], Barlow Twins [27] and DiNO [2] along with our MoCo default, Spatial and Appearance models. All use a ResNet50 backbone.

Results: From the plot in Figure E.1, we can see that: (i) All the standard models fall between the performance of the spatial and appearance models in terms of spatial and appearance invariance (left two metrics). This shows that it is not possible to achieve high appearance and spatial invariance simultaneously. (ii) While the particular suite of default invariances has been well tuned for classification (substantively outperforming both our appearance and spatial models), it is poorly tuned for regression, where our Appearance model performs best. This suggests that the default augmentation suite has been overfitted to the most common benchmarks, and more thought is necessary on designing augmentation distributions suitable for more diverse downstream tasks.

F Hypothesis testing for similarities

We make use of Hoeffding's inequality, which for a sum of random variables, $S_n = \frac{1}{n}\sum_{i=1}^{n} X_i$, where each X_i is in the range $[0,1]$ with probability one, tells us that

$$P(S_n - \mathbb{E}[S_n] \geq t) \leq e^{-2nt^2}. \tag{F.1}$$

Setting the left-hand side equal to δ and rearranging for t yields

$$t \leq \sqrt{\frac{\ln(1/\delta)}{2n}}. \tag{F.2}$$

This fact can be used to test the null hypothesis that the expected value of S_n is zero: set δ to the threshold that will be applied to a p-value, and check whether S_n is greater than the right-hand side of Eq. F.2. If it is greater, then one can reject the null hypothesis. By setting S_n equal to the mean difference in representation similarity for two different methods, we can test whether one method is statistically significantly more invariant than the other. Bonferroni correction is applied when we carry out multiple hypothesis tests to perform a three-way comparison.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1; and the MOD University Defence Research Collaboration (UDRC) in Signal Processing. This project was supported by the Royal Academy of Engineering under the Research Fellowship programme.