A Data augmentation for pre-training

We use the strong contrastive baseline MoCo-v2 [10] 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 [38] 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

The Random model was not pre-trained and thus did not use any augmentations. The Supervised model was pre-trained using both spatial and appearance-based augmentation (crop, h. flip and a version of color jitter) [23].

B Synthetic transforms

For Table 2 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 [21]. 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.

C.2 Alternative metrics

We note that there are alternative metrics for (in)variance such as alignment and uniformity [47] and ACR [48]. 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 2, showing the Appearance

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**

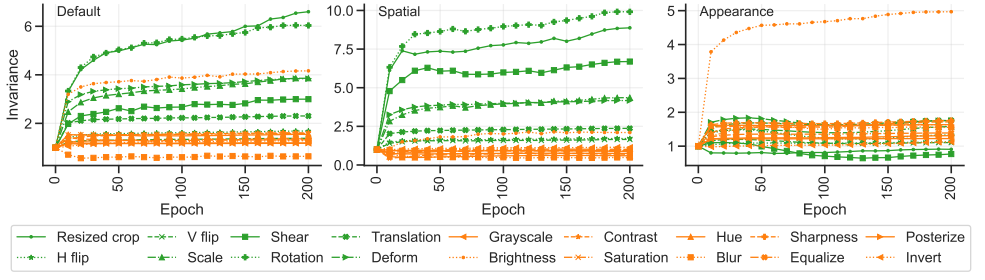

Figure C.1: Invariances as measured by cosine similarity during pre-training. Different invariances are learned at different speeds. After 200 epochs many of the invariances are still steadily increasing, suggesting longer training helps achieve stronger invariances.

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 [5, 7].

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 [32] and iNaturalist 2021 validation set [45]. 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 Datasets with real-world transforms

Here we describe the datasets used in Section 4.2: **Flickr1024:** [43] contains stereo image pairs. We use this to measure invariance to small lateral changes in viewpoint. **COIL-100:** [35] is a dataset of 7,200 images of 100 objects photographed under varying pose conditions against a black background. Each object is imaged at every 5 degree angle of rotation around the vertical axis

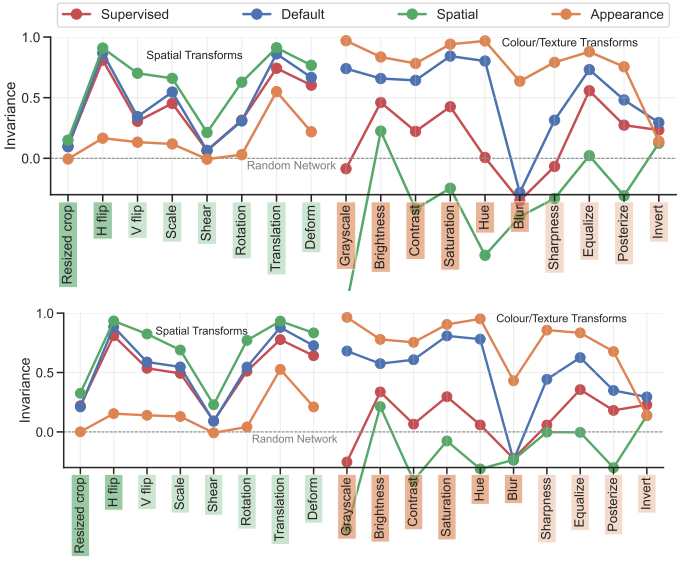

Figure C.2: Invariances to synthetic transforms hold for cluttered images from COCO (top) and in-the-wild images from iNaturalist (bottom).

resulting in 72 images. This creates a suitable dataset for us to investigate the real-world viewpoint invariance of our models. **ALOI:** [19] contains 1,000 objects captured under varying poses, camera angles and lighting conditions, allowing measurements of both viewpoint and illumination invariance. **ALOT:** [4] is a similar dataset in structure but contains photos of 250 textures instead of objects. **ExposureErrors:** [1] is composed of 24,000 images rendered at different exposures from raw RGB data. **RealBlur:** [41] has 4,500 geometrically aligned pairs of blurred and sharp images, thereby providing an ideal setup for measuring real-world blur invariance.

E Correlations

E.1 Between synthetic and real-world transforms

To establish the ability of synthetic transforms to capture the invariances found in the real world, we compute the correlations between our synthetic and real-world invariance measurements. In Figure E.1 we see that invariance to synthetic spatial transforms correlate very strongly with spatial real-world transforms. Furthermore, within the spatial family all transforms are correlated. These effects are similar for our appearance family. Interestingly, illumination changes on real-world data correlate strongly with spatial transforms like cropping and flipping. Intuitively, we would expect it to correlate stronger with synthetic changes in brightness. However, due to the very low illumination in the ALOI/ALOT data, only small regions of the objects are illuminated in each view, meaning that across views it can appear that the object is moving. An outlier in these results is the invert transform which correlates stronger with real-world spatial changes. These results suggest that many common synthetic transforms used in self-supervised pre-training lead to invariances that transfer to corresponding real-world transforms.

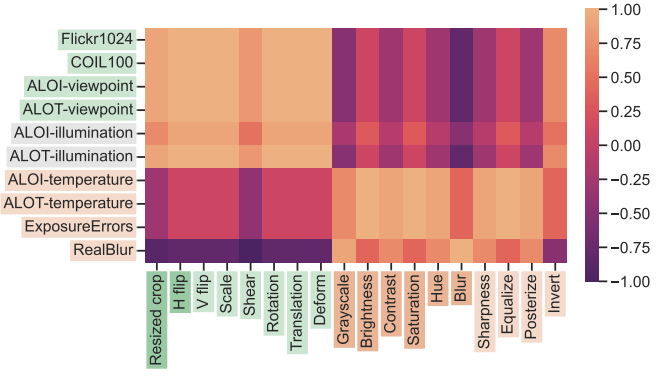

Figure E.1: Rank correlation between invariances to synthetic and real-world transforms. Invariance to synthetic transforms is highly correlated with invariance to the corresponding family of real-world transforms.

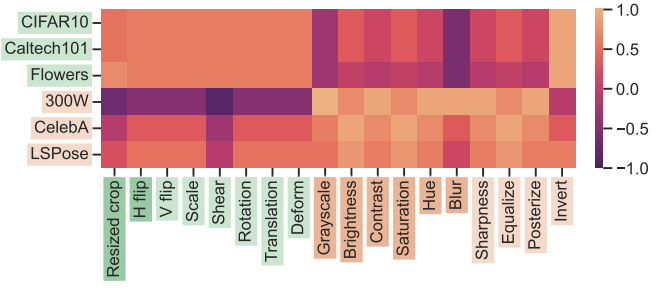

Figure E.2: Rank correlation between synthetic invariances and downstream tasks of our set of models in Table 5 (top). A high value means that learning a stronger invariance to the transform is highly correlated with getting better at the task.

E.2 Between synthetic and real-world transforms

Figure E.2 shows the correlations between synthetic invariances and downstream model performances. We again see a block-like structure, where spatial invariances correlate with classification tasks and 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.

F 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 [42] we use images from both the

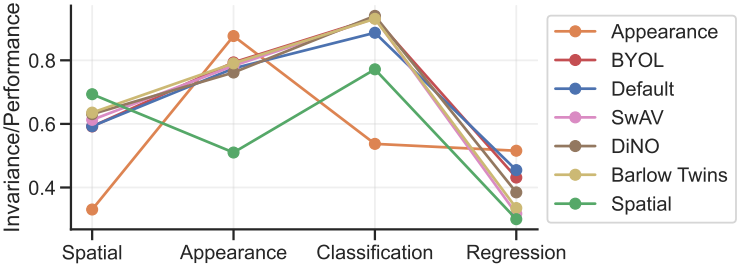

Figure F.1: A parallel coordinates plot demonstrating the spatial/appearance invariance (measured in cosine similarity) and the performance on downstream tasks (measured in accuracy or R^2) for different existing state-of-the-art self-supervised models.

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

F.1 Comparison of other State of the Art

Our experiment so far focused on pre-training 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 4.1) and the downstream tasks studied in Section 5. We evaluate BYOL [22], SwAV [5], Barlow Twins [51] and DiNO [4] along with our MoCo default, Spatial and Appearance models. All use a ResNet50 backbone.
Results: From the plot in Figure F.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 (with all existing methods substantively outperforming both our appearance and spatial models), it is poorly tuned for regression, where our Appearance model performs best. In fact, all other existing models shown here underperform MoCo-v2 on regression tasks, further highlighting the continuing overfitting of SOTA self-supervision to recognition tasks. It provides more evidence that the default augmentation suite has been overfitted to the most common benchmarks, and that more thought is necessary to design augmentation distributions suitable for more diverse downstream tasks.