The supplemental material contains additional analysis, visualization and ablation studies. All these are not included in the main paper due to the space limit.

A. Experimental Details

A.1. Datasets

For the source dataset training, we use ImageNet1K training set [12]. For downstream linear evaluation, we use 12 datasets from different domains to evaluate the transferability of different models. We divide the datasets into six groups - natural, satellite, symbolic, illustrative, medical, and texture. Unless otherwise mentioned, we use top-1 accuracy as evaluation metrics.

The most similar to ImageNet categories (i.e., natural images) are CropDisease, DeepWeeds and Flowers102. **CropDisease** [38] contains natural images of diseased crop leaves categorized into 38 different classes. **DeepWeeds** [41] contains 17,509 images of 8 different weed species native to Australia. **Flowers** [40] is a fine-grained dataset of 102 different flower categories each of which consists of 40 to 258 images.

In the satellite image category, we use EuroSAT adn Resisc45. **EuroSAT** [22] is a satellite imagery dataset consisting of 27,000 labeled images with 10 different land use and land cover classes. **Resisc** [9] is a remote sensing image classification dataset containing 31,500 images of 45 scene classes.

SVHN [39] is obtained from house numbers of google street view images. There are 10 classes, 1 for each digit from 0 to 9, consisting of around 73k training images and 26k testing images. **Omniglot** [35] contains 1623 different hand-writted characters from 50 different alphabets.

Both Kaokore and Sketch contain illustrative or hand-drawn images. **Kaokore** [46] dataset contains 8848 face images from japanese illustration. ImageNet **Sketch** [49] consists of 'black-and-white' sketches from each of ImageNet 1000 classes. Kaokore dataset contains two super classes based on gender and status. The 'gender' class contains male and female, and the 'status' class contains - noble, warrior, incarnation, and commoner. Combining both of them we get total 8 classes.

In the medical imagery domain, we have ChestX and ISIC dataset. **ChestX** [50] is comprised of X-Ray images, and **ISIC** [11] dataset contains dermoscopic images of skin lesions.

In the category of texture dataset, we use **DTD** [10], which consists of 5640 texture images from 47 categories. We use the official training and test split for most datasets if available. If the official split contains both training and validation split, we combine them for training. When there are multiple official splits available (e.g. DTD [10]), we use the first split for evaluation. If there is no official split available, we randomly select 30% of the total images from each category as test set and the remaining images for training. For few-shot learning, we use all the available images from both training and test splits, and randomly select images for support set and query set from 5 random classes (5-way few-shot learning) at each few-shot episode.

A.2. ImageNet Pretraining

We trained all models on the source dataset for 400 epochs with learning rate 0.01 and cosine scheduling with warm-up for 5 epochs. We set the temperature parameter of MoCo to $\tau = 0.07$, and queue size to $65,596$ [21]. The effect of queue size to the transfer learning performance is discussed in the Appendix D.1. For `CE` model, we use random crop(224x224), horizontal flip, and normalization data-augmentations during training. For `SelfSupCon`, `SupCon+SelfSupCon`, and `CE+SelfSupCon`, we use random-crop (224x224), color-jitter, random gray-scale, Gaussian blur, random horizontal flip, and normalization for training data augmentations. CE models with stronger augmentation is also discussed in Appendix. D.6. Unless otherwise mentioned, we use top-1 accuracy as the evaluation metric.

A.3. Linear Evaluation

For fixed-feature linear evaluation, we freeze the pretrained backbone, add a linear layer to train it on the downstream dataset. We add a BatchNorm layer without any affine parameter between the backbone and linear layer to make the extracted features comparable among different models. Note that the BatchNorm layer makes the models to have similar optimal hyperparameters during linear evaluation. We train all models for 50 epochs with step learning rate scheduler which decreases the learning rate by 0.1 at epoch 25 and 37. We also experimented with 100 epochs for all models, but did not notice any noticeable improvement over 50 epochs. As different datasets might require different hyperparameters, we perform extensive hyperparameter tuning. We split the training set on 70% training and 30% validation, and then train the models for

- learning rate: 0.001, 0.01, 0.1
- batch-size: 32, 128
- weight decay: 0, 1e-4, 1e-5

and chose the optimal hyperparamters among different runs based on the performance on the validation set. Nevertheless, we found that batch-size 128, learning rate 0.01, and weight-decay 0 can be chosen as a safe hyperparameter choice for most cases.

A.4. Object Detection

For object detection, we follow the setting in [21] to finetune the full network, and add batch normalization layer into the meta architecture of the detector, e.g., region-of-interested (ROI) header, feature pyramid network (FPN), etc., to minimize the effort on hyperparameter tuning.

A.5. Full-network Finetuning

We train the whole network, i.e., the pretrained backbone and linear layer on the downstream dataset. We train all models for 50 epochs with step learning rate scheduler which decreases the learning rate by 0.1 at epoch 25 and 37. Here, we also experimented with 100 epochs, but did not notice any noticeable improvement over 50 epochs. We add a BatchNorm layer between the backbone and linear layer to make the extracted features comparable among different models. We train the models for various learning rate - 0.01, 0.001, and 0.1, batch-sizes - 32 and 128, and weight decay 0, 1e-4, 1e-5 and select the optimal hyperparameter based on the performance on the validation set. We found that for most datasets learning rate 0.001 with batch-size 32 performs the best, hence this setting can be used in a scenario where hyperperameter tuning is not possible or expensive.

A.6. Few-shot Recogniton

For few-shot pretraining, we closely followed CDFSL benchmark paper [20]. For all models, we train for 300 epochs with SGD optimizer with learning rate 0.01 and batch-size 32, and cosine scheduling with warm-up for 5 epochs. For the contrastive models, we use number of negative samples to be 16384 and temperature parameter to be 0.07. Following [47], we train a logistic regression layer on top of the extracted features during meta-testing phase. We use the implementation from scikit-learn for logistic regression [1]. The accuracy is the mean of 600 randomly sampled tasks, and 95% confidence interval is also reported.

B. More Results and Analysis

B.1. Linear Evaluation

Detailed numbers for linear evaluation are provided in 8.

B.2. Object Detection

Table 9 shows object detection results with standard deviation over 5 runs.

We also conducted the experiments of object detection with longer training setting (2× schedule in Detectron2.) and tested another detector, RetinaNet-R50. The results are shown in Table 10 and Table 11. When training with more iterations (2× schedule), the results of `CE`, `SelfSupCon` and `SupCon` are more closer to `CE+SelfSupCon` and `SupCon+SelfSupCon` as pretraining between less important if the downstream has been trained for a long time. The results of the RetinaNet have the similar trend, which shows that the visual representations trained by `CE+SelfSupCon` and `SupCon+SelfSupCon` are transferable on different types of detectors.

B.3. Few-shot Recogniton

5-way 1-shot, 5-shot and 20-shot results are provided in Table 12.

B.4. Full-network Finetune

Detailed results for image classification with full-network finetuning are provided in Table 13.

C. More Analysis

C.1. Experimental Setup for Analysis on ImageNet Derivatives

For the evaluation on ImageNet1K, ImageNet-A, ImageNet-R, ImageNet-C, and Stylized ImageNet on Table 5 and Table 6 in the main paper, we train a classifier header on top of the ResNet-50 backbone for the contrastive models. The classifier header is trained on the ImageNet1K training set. We SGD optimizer, and tune the learning rate for 0.01, 0.1, 1, 2, 10, and chose the best performing model based on the validation set accuracy. We do not use any weight decay [21]. Note that this is performed only for `SelfSupCon`, `SupCon`, `SupCon+SelfSupCon`. For `CE` and `CE+SelfSupCon`, we use the header associated with the supervised branch.

C.2. Model Calibration on the Downstream Datasets

Table 14 reports calibration performance for the ImageNet pretrained models on the 12 downstream datasets in terms of Expected Calibration Error (ECE). We use fixed-feature linear evaluation to train the linear layer on the downstream tasks. We also perform hyperparameter sweaping on the validation set and report the best ECE score (lower is better). We observe that contrastive models have lower calibration error than cross-entropy model on average.

C.3. Robustness of Adversarial Attack

We used projected gradient descent (PGD) attack to test the robustness of each model. It is clear that without any adversarial training, all models are vulnerable. Nonetheless, we would like to analyze the region with smaller perturbation (i.e., small ε). Figure 8 show the relative accuracy degradation among those methods. Interestingly, `CE` is the most robust one among while all other methods are relatively vulnerable. We think it is because contrastive learning methods are more sensitive to the local changes as they

	CropDisease	DeepWeeds	Flowers102	EuroSAT	Resisc45	ISIC	ChestX	Omniglot	SVHN	Kaokore	Sketch	DTD	Mean
CE	97.18±.04	85.17±.04	84.73±.52	94.62±.12	84.96±.25	79.58±.21	44.70±.20	63.88±.61	66.31±.16	73.96±.16	67.41±.13	65.48±.40	75.67±.24
SelfSupCon	99.06±.04	87.88±.16	89.62±.14	96.76±.05	90.88±.10	81.51±.35	48.08±.13	69.66±.23	69.95±.06	81.67±.80	69.12±.13	72.21±.57	79.70±.23
SupCon	98.85±.03	87.37±.13	92.79±.15	96.05±.12	90.26±.24	79.78±.22	46.76±.32	72.99±.31	74.09±.12	80.94±.23	77.19±.10	74.00±.37	80.92±.20
CE+SelfSupCon	98.79±.03	87.72±.09	91.78±.14	96.67±.16	90.33±.06	80.43±.15	47.44±.17	68.38±.19	73.34±.14	79.48±.16	74.39±.15	73.48±.21	80.19±.14
SupCon+SelfSupCon	99.03±.03	87.79±.21	93.18±.05	96.67±.03	91.98±.08	80.15±.30	47.70±.13	72.84±.34	75.80±.13	78.86±.29	76.78±.06	74.80±.58	81.30±.19

Table 8: Top-1 accuracy of different models on the downstream datasets for fixed-feature extractor transfer learning. The models are pretrained on ImageNet1K dataset and we only train the final linear layer on top of the pretrained backbones. Mean and standard deviation over 5-runs are provided.

Datasets	VOC0712			MS COCO (Trained with 1× schedule)											
Detectors	FasterRCNN-R50-C4			MaskRCNN-R50-C4						MaskRCNN-R50-FPN					
Methods	AP^{bb}_{50}	AP^{bb}	AP^{bb}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
CE	81.58±.17	54.63±.28	60.17±.40	38.91±.02	59.05±.20	42.14±.16	33.98±.04	55.39±.14	36.10±.12	39.84±.14	60.54±.13	43.60±.22	36.54±.08	57.65±.10	39.25±.18
SelfSupCon	82.48±.19	57.33±.44	64.03±.42	39.14±.18	58.74±.28	42.40±.17	34.29±.18	55.53±.26	36.62±.20	39.19±.22	59.54±.20	42.74±.40	36.03±.21	56.79±.27	38.75±.27
SupCon	82.64±.36	56.12±.15	62.45±.27	39.63±.11	59.76±.20	42.77±.23	34.52±.06	56.26±.15	36.54±.15	40.40±.19	61.20±.21	44.35±.34	36.97±.14	58.31±.14	39.80±.34
CE+SelfSupCon	83.06±.26	57.07±.12	63.97±.62	39.68±.15	59.84±.19	42.80±.22	34.62±.12	56.31±.13	36.74±.30	40.65±.13	61.38±.17	44.65±.15	37.21±.21	58.57±.20	40.06±.45
SupCon+SelfSupCon	82.95±.08	57.26±.25	64.12±.44	39.95±.05	60.02±.18	43.12±.17	34.93±.08	56.59±.20	37.27±.18	40.33±.17	60.96±.21	44.20±.31	36.95±.24	58.19±.33	39.73±.33

The results of VOC0712 is the average of 5 runs. AP^{bb}: AP of objection detection; AP^{mk}: AP of instance segmentation.

Table 9: Object detection results on MS COCO.

Detectors	MaskRCNN-R50-C4						MaskRCNN-R50-FPN					
Methods	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
CE	41.04	60.86	44.63	35.48	57.29	37.82	41.43	62.12	45.38	37.89	59.29	40.53
SelfSupCon	41.00	60.77	44.43	35.68	57.48	38.21	41.15	61.53	45.05	37.63	58.84	40.59
SupCon	41.58	61.72	44.99	36.04	58.16	38.42	41.70	62.39	45.47	38.03	59.72	40.66
CE+SelfSupCon	41.28	61.26	45.06	35.75	57.60	38.34	41.77	62.33	45.78	38.15	59.63	41.10
SupCon+SelfSupCon	41.28	61.18	44.97	35.91	57.72	38.38	41.96	62.58	45.92	38.37	59.81	41.36

AP^{bb}: AP of objection detection; AP^{mk}: AP of instance segmentation.

Table 10: Object detection and instance segmentation results on MS COCO (2× schedule).

Detector	RetinaNet-R50					
	1× schedule			2× schedule		
Methods	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
CE	38.46	58.32	41.19	40.13	60.02	43.10
SelfSupCon	37.65	56.62	40.29	39.61	59.14	42.28
SupCon	38.77	58.48	41.52	40.35	60.23	43.26
CE+SelfSupCon	**39.33**	58.94	42.24	40.52	60.11	43.32
SupCon+SelfSupCon	39.24	58.67	42.23	40.50	60.34	43.50

AP^{bb}: AP of objection detection.

Table 11: Object detection results on MS COCO.

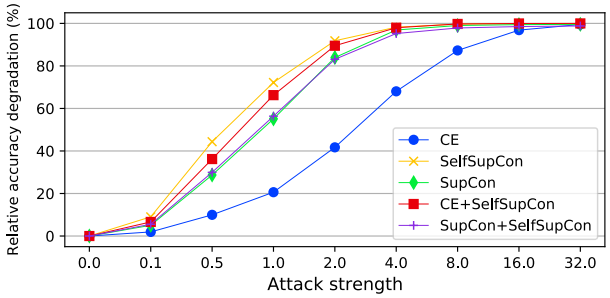

Figure 8: Performance degradation under different attack strengths (ε). L_∞PGD attack is used [37].

learn local features while CE does not; hence, CE can tolerance more perturbations.

C.4. Comparison with Other Self-supervised Models

We further evaluated two contrastive learning methods, SwAV [5] and BYOL [18]. These two contrastive learning approaches have different characteristics than MoCov2. SwAV is a clustering-based approach while BYOL does not need negative samples in contrastive loss. The detailed results with linear evaluation on all downstream tasks can be found in Table 15. SwAV and BYOL significantly outperform CE by 5.28% and 5.35% on mean transfer accuracy, respectively.Thus, those contrastive models could learn features for better transferability than the models trained with cross-entropy.

D. More Ablation Studies

D.1. Effect of Queue-size to Transferability

Table 16 shows the effect of queue size to transferability for the contrastive models. All models are trained on ImageNet1K training set with queue sizes 1024, 8192, and

	miniImagenet	CropDisease	DeepWeeds	Flowers102	EuroSAT	Resisc45	ISIC	ChestX	Omniglot	SVHN	Kaokore	Sketch	DTD	Mean
1-shot														
CE	53.96±.82	68.72±.85	35.73±.62	62.03±.92	63.47±.91	53.38±.75	33.33±.60	22.78±.41	83.73±.71	23.37±.41	28.92±.54	45.25±.79	41.43±.75	46.84±.69
SelfSupCon	48.88±.77	65.35±.85	38.68±.61	66.04±.85	66.07±.83	53.09±.86	33.73±.60	22.93±.43	81.88±.74	23.28±.40	31.70±.61	47.83±.79	44.19±.79	47.90±.70
SupCon	55.50±.78	63.59±.83	35.71±.59	63.17±.86	63.41±.83	55.14±.81	32.15±.60	22.73±.41	87.52±.66	25.82±.46	32.87±.59	53.55±.85	48.22±.82	48.66±.69
CE+SelfSupCon	56.84±.79	66.06±.80	38.07±.62	68.65±.81	65.88±.84	55.76±.86	33.52±.61	23.50±.42	87.20±.63	24.99±.44	33.87±.61	52.42±.86	49.36±.84	49.94±.70
SupCon+SelfSupCon	52.51±.79	65.27±.80	38.11±.59	68.74±.86	66.24±.84	54.90±.88	34.09±.62	23.94±.43	88.24±.65	24.58±.43	33.06±.60	50.90±.86	47.16±.80	49.60±.70
1-shot														
CE	72.47±.62	86.58±.56	48.33±.55	81.31±.68	78.51±.68	72.86±.67	44.28±.52	26.16±.43	94.26±.34	27.98±.48	37.63±.54	64.64±.74	57.64±.73	60.01±.58
SelfSupCon	67.71±.63	83.29±.62	49.90±.59	84.29±.57	81.65±.60	72.36±.74	45.20±.52	26.91±.44	93.61±.36	27.10±.46	42.75±.57	66.32±.72	61.35±.73	61.23±.58
SupCon	75.20±.62	83.44±.60	47.74±.58	82.93±.59	80.94±.63	74.48±.70	42.97±.52	26.23±.43	96.78±.23	33.22±.53	45.10±.57	74.24±.71	65.42±.76	62.79±.57
CE+SelfSupCon	76.13±.60	84.68±.59	50.02±.60	86.88±.54	82.63±.58	75.11±.71	44.66±.49	27.93±.45	96.19±.27	31.36±.52	45.32±.57	72.38±.68	67.21±.71	63.70±.56
SupCon+SelfSupCon	72.81±.59	84.26±.59	50.35±.59	86.72±.52	82.57±.61	74.94±.72	45.82±.51	28.19±.46	96.72±.24	30.67±.50	45.26±.57	71.09±.72	65.08±.72	63.47±.56
1-shot														
CE	80.81±.50	92.51±.39	58.43±.56	89.00±.53	84.77±.53	82.08±.55	52.88±.54	29.86±.44	97.76±.20	35.69±.51	46.87±.53	76.43±.61	67.85±.67	67.84±.51
SelfSupCon	76.95±.52	90.42±.45	58.99±.56	90.77±.42	87.72±.47	82.20±.56	53.57±.52	32.01±.45	97.70±.20	34.70±.53	52.71±.56	77.94±.57	71.00±.64	69.14±.49
SupCon	83.32±.46	90.73±.45	55.90±.58	90.45±.42	87.71±.45	83.40±.54	51.11±.50	31.06±.47	98.87±.13	44.98±.57	54.94±.51	85.23±.48	74.11±.63	70.71±.48
CE+SelfSupCon	84.24±.45	91.27±.42	58.63±.54	92.70±.39	89.13±.44	84.54±.54	52.80±.50	33.42±.47	98.63±.15	41.60±.57	54.72±.54	83.32±.52	75.78±.62	71.38±.48
SupCon+SelfSupCon	82.07±.47	91.54±.42	59.24±.55	92.51±.38	89.04±.43	84.18±.53	54.08±.51	33.92±.48	98.71±.15	40.84±.58	55.25±.55	82.84±.53	74.58±.63	71.39±.48

Table 12: Few-shot classification accuracies (%) on the Mini-ImageNet and 12 downstream datasets. Mean and 95% confidence interval over 600 tasks.

	CropDisease	DeepWeeds	Flowers102	EuroSAT	Resisc45	ISIC	ChestX	Omniglot	SVHN	Kaokore	Sketch	DTD	Mean
Full Dataset													
CE	99.92±.02	97.37±.13	93.47±.30	98.78±.09	96.03±.09	88.06±.36	55.67±.56	90.34±.29	97.03±.12	88.13±.40	79.49±.27	73.26±.56	88.13±.27
SelfSupCon	99.91±.02	97.39±.13	93.36±2.07	98.85±.04	96.28±.12	88.13±.13	56.41±.26	91.10±.20	97.25±.10	88.92±.36	77.16±.13	75.49±.17	88.35±.31
SupCon	99.91±.01	96.89±.11	95.37±.16	98.67±.12	96.01±.06	87.92±.33	55.08±.57	90.56±.19	97.11±.11	87.88±.66	80.14±.19	74.19±.20	88.31±.23
CE+SelfSupCon	99.88±.04	97.28±.22	95.30±.40	98.91±.06	96.18±.12	88.29±.45	54.77±.17	90.20±.22	97.14±.08	88.25±.66	80.85±.12	74.12±.47	**88.43**±.25
SupCon+SelfSupCon	99.91±.02	97.38±.07	96.17±.34	98.75±.05	96.21±.07	88.55±.47	55.12±.16	90.86±.14	97.04±.09	87.73±.25	80.17±.26	74.86±.47	**88.56**±.20
1000 training samples													
CE	93.89±.40	87.46±.57	88.31±.43	94.68±.30	79.33±.26	78.26±.96	40.42±.48	44.95±1.05	77.65±.52	77.28±1.13	14.87±1.10	60.43±.68	69.79±.66
SelfSupCon	93.95±.51	88.10±.78	88.92±.22	95.47±.41	81.14±.48	78.27±.36	43.09±.80	45.33±1.10	82.37±1.14	79.41±.85	10.57±.27	62.95±1.08	70.80±.67
SupCon	93.93±.08	86.89±.57	91.53±.23	94.73±.52	81.95±.21	78.51±.62	41.97±.24	44.48±1.53	79.25±.83	77.41±.38	16.31±.43	64.71±.31	**70.97**±.50
CE+SelfSupCon	93.60±.48	86.83±.67	91.01±.32	94.51±.46	80.43±.54	78.68±.32	41.65±.88	44.08±.43	79.85±.84	79.14±1.09	15.72±.24	65.07±1.08	70.88±.61
SupCon+SelfSupCon	93.82±.27	87.42±.64	91.93±.69	95.12±.47	81.28±.67	78.93±.42	41.77±.43	44.06±1.25	80.51±.61	78.84±.57	15.57±.26	66.02±.40	**71.27**±.56

Table 13: Performance of different models on the downstream datasets in terms of top-1 accuracy (%) (averaged over 5 runs) for full-network fine-tuning. Contrastive pretrained methods are slightly more effective in a limited data regime than cross-entropy based models.

	CropDisease	DeepWeeds	Flowers102	EuroSAT	Resisc45	ISIC	ChestX	Omniglot	SVHN	Kaokore	Sketch	DTD	Mean
CE	0.83±.06	1.41±.37	2.98±.33	0.64±.11	6.89±.32	1.74±.26	2.79±.26	9.30±.43	2.10±.28	4.84±.20	9.60±1.45	6.64±.55	4.15±.38
SelfSupCon	0.24±.03	3.58±.15	2.52±.12	0.78±.08	1.02±.21	2.92±.45	2.01±.47	19.02±.66	3.99±.23	3.93±.64	6.46±.18	7.08±.56	4.46±.31
SupCon	0.55±.03	1.80±.15	4.21±.12	0.99±.15	4.83±.21	2.05±.20	1.80±.25	3.82±.33	2.52±.25	3.37±.54	7.27±.23	3.53±.24	3.06±.23
CE+SelfSupCon	0.57±.06	7.16±.20	1.02±.32	0.68±.09	4.52±.15	4.11±.22	3.04±.29	4.78±.22	2.36±.28	4.79±.30	7.62±.14	4.96±.18	3.80±.20
SupCon+SelfSupCon	0.55±.04	2.40±.08	3.10±.27	0.78±.06	0.98±.05	4.69±.31	1.64±.23	4.87±.53	3.23±.43	6.96±.73	5.15±.30	6.38±.31	3.39±.28

Table 14: Expected calibration Error (%) of the models on the downstream datasets. The models are pre-trained on ImageNet1K dataset and we only train the final linear layer on top of the pretrained backbones.

	CropDisease	DeepWeeds	Flowers102	EuroSAT	Resisc45	ISIC	ChestX	Omniglot	SVHN	Kaokore	Sketch	DTD	Mean
SwAV	98.83	90.24	93.06	97.22	92.62	80.15	48.48	68.92	70.31	82.46	71.18	76.81	80.86
BYOL	98.76	87.52	92.86	96.80	91.69	80.49	47.92	70.69	73.03	81.36	75.65	74.36	80.93

Table 15: Linear evaluation with other contrastive learning methods.

65596. We perform fixed-feature linear evaluation on the downstream datasets and report the average accuracy. The table shows that higher queue size is better for transferability.

D.2. Results with torchvision-pretrained Model

Table 17 shows linear evaluation and full-network fine-tuning scores from torchvision pretrained model and our reconstructed ResNet-50 model. We note that the torchvision pretrained one achieves better accuracy in linear evaluation, but the accuracy is similar in full-network finetuning. Our CE model was trained for 400 epochs with cosine learning rate scheduler, whereas the torchvision one has been trained for 100 epochs with step learning scheduler. We hypothesize that the training schedule might have made difference in the transfer performance. However, although the torchvision pretrained model has better linear evaluation accuracy, it is not better in full-network fine-tuning, robustness or even ImageNet performance. The CE (torchvision) achieves 75.7% ImageNet accuracy, where our CE achieves 76.60% ImageNet accuracy. From Table 18, our CE achieves 60.80% mCE in ImageNet-C, whereas

Method	Queue size		
	1024	8192	65596
SelfSupCon	79.23	79.68	79.70
SupCon	78.11	79.88	80.92
CE+SelfSupCon	77.24	78.66	80.19
SupCon+SelfSupCon	79.42	80.50	81.30

Table 16: Ablation studies on the effect of **queue size** to the linear evaluation performance on the downstream datasets. Models trained with higher queue size generally performs better in transfer.

CE (torchvision) achieves much worse score 78.50%. We infer that our training strategy might be better for cross-entropy model in terms of calibration, robustness, or even in-domain accuracy. Overall, even with the torchvision pretrained ResNet-50 model, our main thesis does not change, as the performance is still significantly lower than the contrastive models.

D.3. ResNet50x2 as Backbone

We also use WideResNet-50-x2 as the backbone of the networks, and report fixed feature linear evaluation transfer and full-network transfer in Table 19 and. We found similar pattern with larger backbone that contrastive approaches provide more transferable representations.

D.4. Pretraining on Stylized ImageNet

We also train all models on the Stylized ImageNet training set so that the models learn more shape-based representations [16]. We then perform a fixed feature linear evaluation on the 12 downstream datasets. Table 20 shows the top-1 accuracy on the downstream datasets for the Stylized ImageNet trained model. The results reveal that contrastive approaches also provide better transferable representation than the cross-entropy model when trained on Stylized ImageNet.

D.5. Measuring Intra-class Similarity

Supervised learning models learn feature representations by objectives that also increase the inter-class separation. However, we argue that increasing the intra-class variation, though might be harmful for in-domain performance, is beneficial for learning rich feature representations in transfer learning. We compute the inter-class and intra-class separation, as follows [32]:

$$R_{\text{intra}} = \sum_{k=1}^{K} \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \frac{1 - \text{cosine}(\mathbf{x}_{k,i}, \mathbf{x}_{k,j})}{K N_k^2} \tag{4}$$

$$R_{\text{inter}} = \sum_{k=1}^{K} \sum_{\substack{1 \le m \le K \\ m \neq k}} \sum_{i=1}^{N_k} \sum_{j=1}^{N_m} \frac{1 - \text{cosine}(\mathbf{x}_{k,i}, \mathbf{x}_{m,j})}{K(K-1) N_k^2} \tag{5}$$

where $\text{cosine}(\cdot, \cdot)$ is the cosine similarity. Table 21 reports the intra-class and inter-class distance of the representations of penultimate layers. Surprisingly, SupCon has very high intra-class variation, although it was not trained using any such constraint. However, CE+SelfSupCon does not have higher intra-class distance as we expected from MoCo loss. Our intuition is that the way we have calculated intra and inter-class distance in Eq. 4 and Eq. 5 might not properly capture the embedding landscape. We, nonetheless, report the scores to inform the community about our observation.

D.6. Results for CE with Stronger Augmentation

Table 22 show comparison with CE and CE(strong), where CE(strong) is trained with similar data augmentation as MoCo. We note that, in linear evaluation setting, for SVHN, Sketch and Omniglot, CE(strong) performs better than CE; however, it performs worse in CropDisease and ISIC dataset. In general, CE(strong) might be helpful for transfer learning to a domain which is very different from the source domain. We also see that contrastive approaches generally perform better in most cases, which suggests that both contrastive loss itself is helpful for learning transferable representation. We also perform experiments where CE model is trained with similar augmentation and batch formation as in MoCo, i.e., strong augmentation and mini-batch formation with two different views of the same image, denoted as CE(MoCo-aug). Table 23 shows results of CE(MoCo-aug). Here, we also find that it performs similar as CE model.

	CropDisease	DeepWeeds	Flowers102	EuroSAT	Resisc45	ISIC	ChestX	Omniglot	SVHN	Kaokore	Sketch	DTD	Mean
Linear Evaluation	98.19±.05	87.58±.18	88.40±.18	95.21±.03	85.62±.06	76.87±.13	45.05±.23	64.39±.34	70.41±.11	75.79±.11	66.18±.43	71.72±.24	77.12±.17
Finetune	99.87±.01	97.53±.09	95.47±.20	98.91±.05	95.92±.19	88.74±.27	54.47±.35	89.99±.14	96.96±.04	87.73±.30	78.36±.14	73.84±.27	88.15±.17

Table 17: Top-1 accuracy of different models on the downstream datasets for ImageNet-pretrained ResNet-50 model from **torchvision** [42]. Mean and std over 5 runs.

Method	ImageNet-R		ImageNet-A		ImageNet-C
	Top-1(%)	ECE(%)	Top-1(%)	ECE(%)	mCE(%)
`CE`	35.83	19.45	3.35	55.03	60.80
`CE` (torchvision)	36.11	19.75	3.98	62.44	78.50
`CE(strong)`	41.24	17.90	8.01	50.81	68.90

Table 18: Robustness tests on ImageNet-R, ImageNet-A, and ImageNet-C datasets for `CE(strong)` and `CE` (torchvision). ECE is the expected calibration error (lower is better) and mCE (lower is better) is the mean of the (unnormalized) corruption errors of the Noise, Blur, Weather, and Digital corruptions. Models are trained only on clean ImageNet images.

	CropDisease	DeepWeeds	Flowers102	EuroSAT	Resisc45	ISIC	ChestX	Omniglot	SVHN	Kaokore	Sketch	DTD	Mean
Linear Evaluation													
CE	97.27	87.33	91.08	95.04	85.41	80.39	45.50	64.90	67.85	77.59	69.14	68.40	77.49
SelfSupCon	99.07	88.23	92.45	97.10	90.63	81.62	49.96	71.72	71.57	82.10	69.28	71.33	80.42
SupCon	98.52	86.70	94.90	95.99	90.55	79.32	47.62	70.26	74.76	81.24	76.62	74.36	80.90
CE+SelfSupCon	98.82	89.48	94.51	96.51	89.94	80.22	49.56	69.40	73.34	79.42	75.08	74.41	80.89
SupCon+SelfSupCon	98.95	89.21	96.27	96.95	91.50	80.35	49.46	73.52	73.11	81.61	76.24	74.89	81.84
Full-network finetuning													
CE	99.92	97.66	97.75	98.75	96.05	89.30	55.54	91.02	96.90	90.01	80.42	73.35	88.89
SelfSupCon	99.88	96.94	97.84	98.81	96.17	89.56	56.70	91.15	97.13	89.52	77.08	74.52	88.78
SupCon	99.86	96.99	98.63	98.62	95.89	88.76	55.68	90.77	97.00	89.52	80.98	75.32	89.00
CE+SelfSupCon	99.86	97.24	98.24	98.78	96.12	90.09	55.12	90.97	97.07	90.01	81.94	77.23	89.39
SupCon+SelfSupCon	99.89	97.60	97.94	98.74	96.29	89.96	55.83	91.05	97.05	90.74	81.69	76.60	89.45

Table 19: Performance of different models with **WideResNet-50-x2** backbone on the downstream datasets in terms of top-1 accuracy with fixed-feature **linear evaluation**. We present the best performing scores for different hyperparameters for each model.

	CropDisease	DeepWeeds	Flowers102	EuroSAT	Resisc45	ISIC	ChestX	Omniglot	SVHN	Kaokore	Sketch	DTD	Mean
CE	96.27	79.93	87.16	92.58	82.74	78.16	45.09	73.39	59.10	76.13	66.94	54.41	74.33
SelfSupCon	98.51	84.71	90.00	95.95	88.51	81.65	49.23	77.06	67.23	79.54	65.42	64.04	78.49
SupCon	97.69	79.72	90.49	94.23	87.19	78.26	47.92	74.84	66.57	79.90	73.94	61.91	77.72
CE+SelfSupCon	98.31	83.75	91.08	95.31	88.69	79.89	48.89	73.52	61.41	80.88	72.30	63.30	78.11
SupCon+SelfSupCon	98.37	83.60	93.14	95.16	89.66	80.25	49.14	75.06	63.29	81.85	73.85	64.41	78.98

Table 20: Performance of different Stylized-ImageNet pretrained models on the downstream datasets in terms of of top-1 accuracy for fixed feature linear transfer.

Method	R_{intra}	R_{inter}
CE	0.32 ± 0.0057	0.60 ± 0.0044
SelfSupCon	0.54 ± 0.0071	0.77 ± 0.0036
SupCon	0.45 ± 0.0085	0.71 ± 0.0055
CE+SelfSupCon	0.28 ± 0.0053	0.55 ± 0.0041
SupCon+SelfSupCon	0.48 ± 0.0073	0.73 ± 0.0040

Table 21: Intra-class and inter-class separation in the penultimate layer for different models on the ImageNet1K validation set. Results averaged over 10 runs.

CropDisease	DeepWeeds	Flowers102	EuroSAT	Resisc45	ISIC	ChestX	Omniglot	SVHN	Kaokore	Sketch	DTD	Mean
Linear Evaluation												
95.61±.08	83.19±.26	85.61±.09	94.18±.23	84.82±.05	76.77±.45	43.98±.33	66.38±.42	70.10±.06	72.39±1.49	69.97±.14	66.74±.86	75.81±.37
Few-shot Recognition (5-shot)												
82.13±.61	46.09±.58	80.83±.61	78.79±.63	71.76±.71	41.48±.53	26.47±.42	94.68±.32	34.99±.57	41.96±.55	72.13±.69	64.42±.71	61.31±.58
Full-network finetuning												
99.90±.03	97.17±.10	94.30±.14	98.65±.10	96.00±.09	87.34±.33	56.01±.33	90.64±.17	97.14±.11	88.12±.86	80.56±.21	73.42±.09	88.27±.21

Table 22: Results with `CE(strong)` (CE model with strong augmentation). As mentioned in the main paper, the performance difference between `CE` and `CE(strong)` is minor.

CropDisease	DeepWeeds	Flowers102	EuroSAT	Resisc45	ISIC	ChestX	Omniglot	SVHN	Kaokore	Sketch	DTD	Mean
Linear Evaluation												
95.49±.09	82.21±.43	86.94±.17	93.79±.03	84.29±.02	76.86±.46	43.82±.21	65.95±.24	68.61±.15	74.75±.81	69.39±.10	68.00±.46	75.84±.27
Few-shot Recognition (5-shot)												
79.47±.65	45.03±.55	77.92±.65	76.54±.64	69.62±.75	40.61±.52	25.44±.41	95.56±.29	37.68±.57	38.39±.53	71.26±.68	63.04±.74	60.05±.58
Full-network finetuning												
99.89±.02	97.31±.16	95.03±.11	98.87±.02	96.09±.02	88.38±.96	55.93±.39	90.39±.13	97.21±.03	88.15±.39	80.33±.23	75.59±.28	88.60±.23

Table 23: Results with `CE(MoCo-aug)` (CE model with similar augmentation and mini-batch formation as MoCo). We note minor performance variation between `CE` and `CE(MoCo-aug)`.