

Task-Aware Routing of Representations for Invariant-Equivariant Self-Supervised Learning

Supplementary Material

A Additional Experiments

A.1 Comparison with STL

Table A.1 compares the linear evaluation performance of our method against STL [36] and STL combined with AugMix [13] across 11 downstream classification benchmarks. We report results under two pretraining settings: STL10 with ResNet-18 and ImageNet100 with ResNet-50. Following the experimental setting of the STL paper, we evaluate on the AWS Flowers dataset, which differs from the Oxford Flowers [24] in its train/test split. Except for this replacement, all other evaluation protocols are kept identical.

Table A.1: **Image Classification.** Linear evaluation accuracy (%) across 11 datasets. Models are pretrained on STL10 using ResNet-18 or on ImageNet100 using ResNet-50. Symbol * denotes performance reported in [36], based on 200 epochs for STL10 and 500 epochs for ImageNet100. For Flowers, we use AWS Flowers instead of the Oxford Flowers.

Pretraining	Method	CIFAR10	CIFAR100	Food	MIT67	Pets	Flowers	Caltech101	Cars	Aircraft	DTD	SUN397	Mean
ImageNet100	STL*	86.55	66.84	64.32	56.64	65.00	94.51	81.83	35.44	45.42	64.68	44.69	64.18
	STL + AugMix*	87.19	67.70	66.12	59.70	67.10	94.87	84.61	38.48	46.14	69.57	45.75	66.11
	Ours	89.61	71.35	66.62	67.29	76.68	92.81	86.08	49.00	49.66	69.65	52.79	70.16
STL10	STL*	85.22	60.13	38.05	43.53	46.57	73.50	71.36	18.85	30.25	45.34	31.63	49.49
	STL + AugMix*	86.01	62.07	40.16	44.90	46.69	77.37	73.29	19.32	30.87	48.71	33.44	51.17
	Ours	87.45	64.78	41.24	46.82	51.10	78.92	74.76	22.74	35.61	49.75	35.50	53.52

Across both pretraining regimes, our method consistently outperforms STL and its AugMix variant on the majority of datasets. Even when compared to STL models trained for longer epochs (200 for STL10, 500 for ImageNet100), our model achieves superior transferability, as reflected in higher mean accuracy. These results demonstrate the effectiveness of our method in producing more generalizable representations.

A.2 Training Time vs. Performance.

Figure A.1 extends Figure 5 from the main paper by including additional equivariant SSL approaches (CARE [11] and STL [36]) and evaluating up to longer training durations. In the in-domain setting (Figure A.1a), we observe that all methods except CARE exhibit similar performance trends, with most converging to comparable accuracy levels as training progresses.

However, the distinction becomes more pronounced in the out-domain setting (Figure A.1b), where transferability is the primary focus of SSL. Our method consistently outperforms all methods, including those trained for longer durations, and maintains this advantage throughout training. This demonstrates that the proposed approach not only achieves higher transfer accuracy but also reaches this level efficiently, making it suitable for scenarios where generalization to unseen domains and training efficiency are critical.

A.3 ViT Backbone

To assess the backbone independence of our proposed method, we conduct additional experiments using the Vision Transformer (ViT) [6] architecture. Specifically, we adopt MoCo-v3 [3], which utilizes

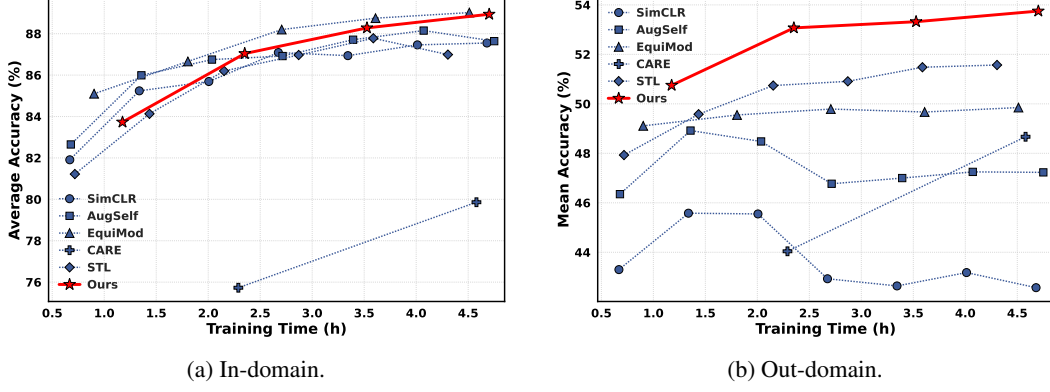


Figure A.1: **Training Time vs. Performance.** Mean accuracy (%) of different methods in (a) in-domain and (b) out-domain settings. Markers are shown every 100 training epochs.

ViT as its backbone, and modify SimCLR by replacing the original ResNet18 or ResNet50 with ViT. Following the experimental setup introduced in [25], ViT-Small is pretrained on ImageNet100 for 200 epochs with a batch size of 256. The training setup commonly employs the AdamW optimizer [21], with a linear warm-up of the learning rate during the first 40 epochs, a momentum of 0.9, and a weight decay of 0.1. A cosine learning rate schedule [20] is used for both the encoder and the projector, while in the case of MoCo-v3, the same schedule is also applied to the predictor.

MoCo-v3. We adopt the original parameter settings, with a learning rate of $1.5e-4$ and a temperature of 0.2 for both contrastive and equivariant learning. The exponential moving average (EMA) coefficient is initialized at 0.99 and gradually increased to 1. We use a 3-layer MLP for each expert, with hidden and output dimensions set to 4096 and 256, respectively. The equivariant predictor is a single-layer MLP with an input dimension of 512 and an output dimension of 256.

SimCLR. The learning rate is set to $1.5e-3$, and the temperature for contrastive and equivariant learning is also fixed at 0.2. We use a 3-layer MLP for each expert, with hidden and output dimensions set to 4096 and 256, respectively. The equivariant predictor is a single-layer MLP with an input dimension of 512 and an output dimension of 256.

Table A.2: **Backbone Ablation Study.** Linear evaluation accuracy (%) of ViT-S/16 pretrained on ImageNet100.

Baseline	Method	CIFAR10	CIFAR100	Food	MIT67	Pets	Flowers	Caltech101	Cars	Aircraft	DTD	SUN397	Mean
SimCLR	-	85.19 \pm 0.41	65.17 \pm 0.10	56.67 \pm 0.19	57.21 \pm 1.37	66.24 \pm 0.42	84.89 \pm 0.22	75.04 \pm 0.18	30.67 \pm 0.46	35.66 \pm 0.33	61.15 \pm 0.96	44.91 \pm 0.22	60.25
	AugSelf	85.69 \pm 0.34	65.88 \pm 0.44	57.36 \pm 0.22	57.39 \pm 0.31	66.89 \pm 0.18	84.89 \pm 0.26	75.38 \pm 0.39	30.67 \pm 0.32	36.00 \pm 0.14	61.44 \pm 1.13	45.42 \pm 0.18	60.64
	EquiMod	86.81 \pm 0.28	67.47 \pm 0.50	59.34 \pm 0.14	60.40\pm0.37	69.86 \pm 0.86	86.90 \pm 0.32	78.12 \pm 0.15	33.71 \pm 0.52	38.67 \pm 0.27	62.87 \pm 0.52	47.08\pm0.03	62.84
	Ours	86.95\pm0.21	68.24\pm0.18	59.73\pm0.12	59.75\pm0.59	68.49\pm0.20	87.43\pm0.40	79.42\pm0.75	35.96\pm0.70	39.96\pm0.26	62.91\pm0.18	46.84\pm0.23	63.24
MoCo-v3	-	84.96 \pm 0.23	64.85 \pm 0.20	57.74 \pm 0.04	57.74 \pm 1.32	65.99 \pm 0.48	84.69 \pm 0.38	75.85 \pm 0.27	30.45 \pm 0.39	35.91 \pm 0.37	60.89 \pm 0.57	45.48 \pm 0.15	60.41
	AugSelf	85.83 \pm 0.30	66.41 \pm 0.24	58.66 \pm 0.23	58.21 \pm 0.32	66.00 \pm 0.46	85.57 \pm 0.17	76.58 \pm 0.16	30.57 \pm 0.72	36.12 \pm 0.42	60.67 \pm 0.48	45.71 \pm 0.23	60.94
	EquiMod	85.98 \pm 0.17	66.52 \pm 0.38	59.69 \pm 0.26	59.75\pm0.85	67.88\pm0.35	87.30 \pm 0.17	78.01 \pm 0.76	32.01 \pm 0.58	37.76 \pm 0.49	63.21\pm0.24	46.82 \pm 0.11	62.23
	Ours	86.72\pm0.08	67.89\pm0.20	60.38\pm0.24	60.40\pm0.19	67.57 \pm 0.30	87.84\pm0.22	79.64\pm0.32	35.27\pm0.32	39.20\pm0.53	62.93\pm1.18	47.73\pm0.16	63.23

In Table A.2, our method achieves better performance than existing equivariant representation learning approaches across most datasets. This highlights the effectiveness of the proposed MMoE projection module, which is applicable to the ViT backbone.

A.4 Hyperparameter Optimization

We study how variation of hyperparameters can influence our model. For that purpose, we train ResNet-18 on STL10 with 16 experts for each factor modification, and present results for both in-domain and out-domain scenarios. We mainly inspect the influence of λ , the balancing coefficient of the equivariant loss (see Eq. 6), and τ , the temperature parameter used in the equivariant learning objective (see Eq. 14). As shown in Table A.3, both parameters have a notable impact on performance.

Table A.3: **Hyperparameter Tuning for λ and τ .** λ is the loss balancing coefficient, and τ is the temperature parameter used in the equivariant learning objective. We report performance on in-domain and out-domain settings under varying values of each.

λ	In-domain	Out-domain
0.1	86.47	50.37
0.2	86.46	51.27
0.5	86.65	52.25
1	86.74	53.07
2	86.17	52.52
5	84.95	50.88
10	82.64	48.87

(a) λ : Loss balancing coefficient

τ	In-domain	Out-domain
0.05	79.43	49.31
0.1	84.45	52.90
0.2	86.74	53.07
0.5	85.31	50.39
1	83.79	48.55

(b) τ : Equivariant temperature parameter

We observe that λ controls the relative strength of the equivariant learning signal. Increasing λ improves performance up to a certain point, with the best results observed at $\lambda = 1$, beyond which performance drops due to the overemphasis on equivariance. This highlights the importance of balancing equivariant and invariant objectives to prevent one from dominating the learning process. For τ , which modulates the sharpness of the similarity distribution in the equivariant contrastive loss, we find that $\tau = 0.2$ achieves optimal results. Smaller values such as 0.05 may cause representational collapse due to overly confident similarity distributions, whereas larger values (e.g., $\tau = 1$) reduce the discriminative power of the model. These trends are consistent across both in-domain and out-domain evaluations, emphasizing the necessity of careful calibration of hyperparameters for generalizable representation learning.

A.5 Qualitative Analysis of Learned Representations

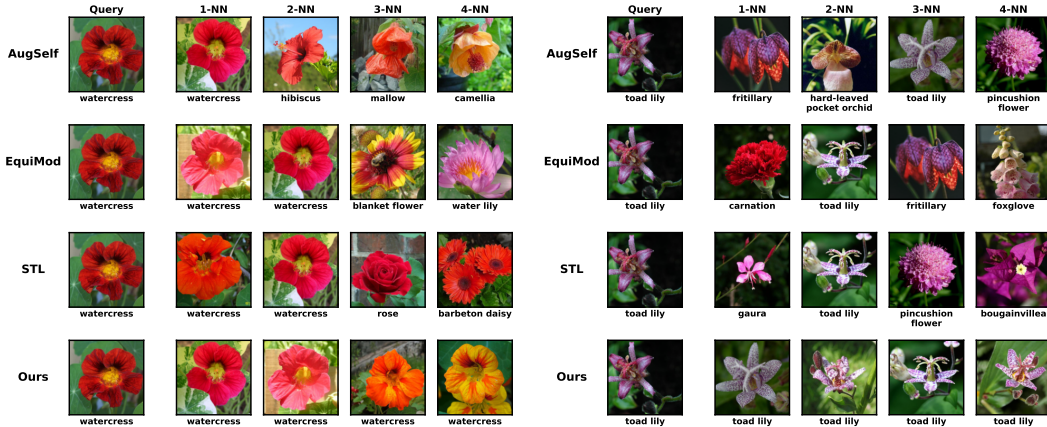


Figure A.2: **k -NN Retrieval on Flowers Test Set.** Results of k -NN retrieval using backbone features learned by Equivariant SSL methods.

Figure A.2 presents qualitative results of k -NN retrieval on the Flowers test set using backbone features from equivariant SSL methods. All models are based on ResNet-18 pretrained on STL10. We observe that most methods exhibit strong sensitivity to color information, often retrieving visually similar but semantically incorrect samples. For example, methods such as AugSelf and EquiMod frequently return instances from different classes that share similar color distributions with the query. STL also tends to favor samples with matching low-level visual cues, rather than consistently retrieving semantically correct instances. This indicates that their learned representations may not fully capture semantic object-level features.

In contrast, our method consistently retrieves samples from the same class as the query while still preserving sensitivity to fine-grained visual details such as color and texture. This suggests that our proposed method enables the model to encode features that are consistent with the class more effectively. Quantitatively, this advantage is reflected in a performance improvement of approximately 5 percentage points compared to the method that performs best on the Flowers dataset.

A.6 Extended Retrieval Visualization of Individual Experts

Figure A.3 provides full k -NN retrieval results using the output embeddings of all individual experts. While Figure 3b in the main paper focuses on three specific experts, namely the shared expert (Expert 1), the invariant expert (Expert 3), and the equivariant expert (Expert 7), this extended visualization enables a more comprehensive examination of the specialization exhibited by all experts.

We observe diverse retrieval patterns across experts. Expert 1 consistently retrieves semantically similar instances, indicating a focus on object-level meaning. Experts 2 through 6 often retrieve samples that differ in color from the query, suggesting that these experts have learned to ignore color and instead emphasize more abstract features. In contrast, Expert 7 and Expert 8 tend to retrieve samples with similar color characteristics, indicating that these experts have learned to capture information related to the augmentations.

This qualitative evidence supports the notion that the MMoE architecture induces meaningful division of representational roles among experts, with some specializing in robust semantic consistency and others in transformation sensitivity. Such functional diversity contributes to the model’s capacity to generalize across tasks and domains.

B Experimental Setup

B.1 Datasets

Table B.1: **Dataset Information.** Overview of the datasets used in the experiments. This table lists dataset names, the number of classes, and the counts for training, validation, and test samples, along with the evaluation metrics.

Category	Dataset	# of classes	Training	Validation	Test	Metric
(a) Pretraining	STL10 [5]	10	105,000	-	-	-
	ImageNet100 [31, 32]	1000	126,689	-	-	-
(b) Linear Evaluation	CIFAR10 [18]	10	45,000	5,000	10,000	Top-1 accuracy
	CIFAR100 [18]	100	45,000	5,000	10,000	Top-1 accuracy
	Food [1]	101	68,175	7,575	25,250	Top-1 accuracy
	MIT67 [29]	67	4,690	670	1,340	Top-1 accuracy
	Pets [27]	37	2,940	740	3,669	Mean per-class accuracy
	Flowers [24]	102	1,020	1,020	6,149	Mean per-class accuracy
	Caltech101 [8]	101	2,525	505	5,647	Mean per-class accuracy
	Cars [17]	196	6,494	1,650	8,041	Top-1 accuracy
	Aircraft [22]	100	3,334	3,333	3,333	Mean per-class accuracy
	DTD (split 1) [4]	47	1,880	1,880	1,880	Top-1 accuracy
(c) Few-shot	SUN397 (split 1) [35]	397	15,880	3,970	19,850	Top-1 accuracy
	FC100 [26]	20	-	-	12,000	Average accuracy
	CUB200 [33]	200	-	-	11,780	Average accuracy
(d) Object Detection	Plant Disease [23]	38	-	-	54,305	Average accuracy
	VOC2007+2012 [7]	20	16,551	-	4,952	Average precision

Table B.1 summarizes the datasets used in our experiments. Category (a) includes the pretraining datasets, STL10 and ImageNet100, which are exclusively used during the unsupervised pretraining phase. Category (b) covers the datasets used for linear evaluation, with each dataset annotated by the number of classes and the number of samples in the training, validation, and test splits. For datasets without an official validation split, validation samples are randomly selected from the training set. Category (c) consists of few-shot benchmarks, including the meta-test split of FC100 [26], as well as the full datasets of CUB200 [33] and Plant Disease [23]. Finally, Category (d) comprises object detection datasets, where we use the `trainval` split of VOC07+12 [7] for training and the `test` split for evaluation.



Figure A.3: k -NN Retrieval on STL10 Test Set. k -NN retrieval results using the output embeddings of all individual experts.

103 B.2 Pretraining Setups

104 B.2.1 ImageNet100 Pretraining

105 We pretrain a ResNet-50 backbone [12] on ImageNet100, a 100-class subset of ImageNet [31],
 106 following the dataset splits in [32]. The model is trained using the SimCLR framework [2], employing
 107 stochastic gradient descent (SGD) for 200 epochs with a batch size of 256. A cosine annealing

learning rate schedule [20] is used, initialized at 0.03 and without restarts, and a weight decay of 0.0005 is applied. The architecture includes 8 experts $\{E_i\}_{i=1}^8$, each implemented as a 3-layer MLP with a hidden dimension of 2048 and an output dimension of 128. Batch normalization [14] is excluded from the final layer of each expert. The equivariant predictor ϕ_T consists of 3 layers, each with a hidden dimension of 512. The gating networks G^{inv} and G^{eq} are implemented as single-layer MLPs that output 8-dimensional vectors corresponding to the number of experts, followed by softmax activations to produce normalized weights over experts. Pretraining on ImageNet100 is performed using 4 NVIDIA RTX 4090 GPUs.

116 B.2.2 STL10 Pretraining

We pretrain a ResNet-18 backbone on STL10 using stochastic gradient descent (SGD). Training is conducted for 200 epochs with a batch size of 256. A cosine annealing learning rate schedule without restarts is used, with the initial learning rate set to 0.03, except for SimSiam where a learning rate of 0.05 is used. A weight decay of 0.0005 is applied. The gating networks are implemented in the same manner as those used for ImageNet100 pretraining. For STL10, pretraining is conducted on a single NVIDIA RTX 4090 GPU.

SimCLR. We use a 3-layer expert architecture with 16 experts $\{E_i\}_{i=1}^{16}$, each with hidden and output dimensions of 512 and 128, respectively. Batch normalization is excluded from the final layer of each expert. The equivariant predictor is a 3-layer MLP with a hidden dimension of 512. A temperature parameter of 0.2 is used consistently for both contrastive and equivariant learning objectives.

MoCo. A 3-layer expert architecture with 8 experts is employed, where each expert has a hidden dimension of 512 and an output dimension of 128. Batch normalization is excluded from the final layer. The equivariant predictor is a single-layer MLP. A temperature parameter of 0.2 is used consistently for both contrastive and equivariant learning objectives.

SimSiam. We use a 2-layer expert architecture with 4 experts, each having hidden and output dimensions of 2048. Batch normalization is excluded from the final layer. The equivariant predictor is a single-layer MLP with input and output dimensions of 2048. A temperature of 0.1 is used for equivariant learning.

BYOL. A 2-layer expert architecture is used, consisting of 4 experts with a hidden dimension of 4096 and an output dimension of 256. Batch normalization is excluded from the final layer. The equivariant predictor is a 2-layer MLP with a hidden dimension of 512. A temperature of 0.1 is used for equivariant learning.

140 B.3 Evaluation Protocol

Linear Evaluation. We adopt the standard linear evaluation protocol [2, 10, 16], where a linear classifier is trained on top of frozen features extracted from center-cropped images of size 224×224 (or 96×96 when pretrained on STL10), without any data augmentation. Specifically, each image is first resized so that its shorter side is 224 pixels, followed by a center crop of size 224×224 . The classifier is optimized using an ℓ_2 -regularized cross-entropy objective with L-BFGS. The regularization strength is selected based on validation accuracy from 45 logarithmically spaced values ranging from 10^{-6} to 10^5 , and the final test accuracy is reported using the best model. We set the maximum number of L-BFGS iterations to 5000 and employ warm-start initialization by using the previous solution as the starting point for the next optimization step.

Few-Shot Classification. To evaluate representations in few-shot benchmarks, we perform logistic regression on top of frozen features using $N \times K$ support samples, without any fine-tuning or data augmentation, within each N -way K -shot episode.

Object Detection. We train a Faster R-CNN [30] with a R50-C4 backbone on the VOC2007+2012 trainval split containing 16551 images. To assess the quality of learned representations, we freeze all convolutional layers from C1 to C4 and train only the region proposal network (RPN) and the object classification head C5. The model is optimized for 24000 iterations with a batch size of 16

using synchronized batch normalization. The learning rate is set to 0.1 initially and decays by a factor of 10 at 18000 and 22000 iterations. A linear warmup [9] is applied during the first 1000 iterations with slope 0.333.

B.4 Augmentations

In this section, we describe how augmentation parameters are defined based on the transformations used in AugSelf [19] including random crop, horizontal flip, color jitter, grayscale, and Gaussian blur. Each parameter set is defined according to the specific configuration of each transformation. In our method, all parameters are normalized using the empirical mean and standard deviation of each transformation-specific variable before being projected into the embedding space. These normalized parameters are then projected into the same dimensional space as the equivariant embedding through a single linear layer.

- **RandomResizedCrop**. The parameter is defined by the center coordinates H_{center} and W_{center} of the crop, along with the crop size given by height H and width W . The crop is applied to images resized to 96×96 for STL10 and 224×224 for ImageNet100.
- **RandomHorizontalFlip**. This transformation is applied with a probability of 0.5. Since the operation is binary, the parameter is defined as either 0 or 1.
- **ColorJitter**. Color jitter includes four parameters: brightness, contrast, saturation, and hue. It is applied with a probability of 0.8. The maximum strength is set to 0.4 for brightness, contrast, and saturation, and 0.1 for hue. Each parameter is sampled independently from the ranges $[0.6, 1.4]$ for brightness, contrast, and saturation, and $[-0.1, 0.1]$ for hue. The transformations are applied in a random order rather than a fixed sequence. If **ColorJitter** is not applied, a default parameter of $[1, 1, 1, 0]$ is used.
- **RandomGrayScale**. Grayscale conversion is applied with a probability of 0.2. Similar to flipping, the parameter is binary with values of 0 or 1.
- **GaussianBlur**. The parameter consists of both the standard deviation of the blur, which ranges from 0.1 to 2.0, and a binary flag indicating whether the transformation was applied.

C Theoretical Analysis

This section provides a theoretical motivation for our architectural design by formally analyzing the structural limitations of existing two-branch equivariant SSL approaches. As discussed in Section 1, such methods employ separate branches to learn invariant and equivariant representations, but inevitably suffer from redundancy due to the potential overlap in the information they capture.

To support this claim, we examine the dependency structure induced by the underlying data generation process. In particular, we show that invariant and equivariant embeddings are not fully independent, as they are both generated from a shared latent representation. This structural entanglement, evident in the causal diagram, reveals that even with architectural separation, the two branches are dependent through a common parent.

C.1 Preliminaries

Definition C.1 (Definition 1.2.3 (rephrased) [28]). *Let \mathcal{G} be a Directed Acyclic Graph (DAG), and let X, Y, Z be disjoint sets of nodes in \mathcal{G} . We say that X and Y are d -separated by Z , denoted $d\text{-sep}_{\mathcal{G}}(X; Y \mid Z)$, if every undirected path between any node in X and any node in Y is blocked by Z . A path is said to be blocked by Z if it contains a node m such that one of the following holds:*

1. *The path includes a chain $i \rightarrow m \rightarrow j$, or a fork $i \leftarrow m \rightarrow j$, and the middle node $m \in Z$;*
2. *The path includes a collider $i \rightarrow m \leftarrow j$, and neither m nor any of its descendants are in Z .*

Lemma C.1 (Theorem 3.5 (rephrased) [15]). *Let X, Y, Z be random variables corresponding to nodes in the causal diagram \mathcal{G} . For almost all distributions P that factorize over a causal diagram \mathcal{G} (i.e. except for a measure zero set in the space of conditional probability distributions), the set of conditional independencies in P coincides with the set of d -separations in \mathcal{G} :*

$$\mathcal{I}(P) = \mathcal{I}(\mathcal{G}), \quad \mathcal{I}(\mathcal{G}) = \{(X \perp\!\!\!\perp Y \mid Z) : d\text{-sep}_{\mathcal{G}}(X; Y \mid Z)\},$$

204 where $\mathcal{I}(\mathcal{G})$ denotes the set of independencies that correspond to d-separation in \mathcal{G} .

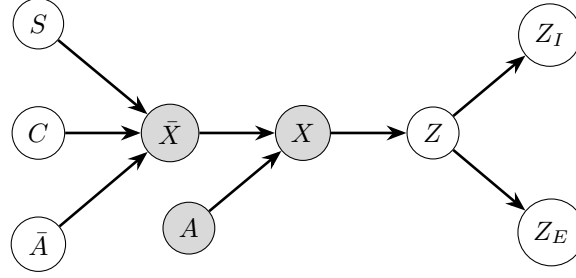


Figure C.1: **Causal Diagram of the Equivariant SSL Process.** We consider an extension of the causal diagram proposed in [34], which illustrates the generative process of images using several latent variables. These include the style variable S , which captures features irrelevant to semantics such as color and texture; the class variable C , which represents the semantic identity of the object, such as whether it is a cat or a dog; and the intrinsic equivariance factor \bar{A} , which captures semantically meaningful variations that preserve class identity, such as an object’s pose, orientation, or viewpoint. These latent factors jointly generate a raw image \bar{X} , which is then transformed by an explicit augmentation A into an observed image X . The image X is subsequently encoded into a latent representation Z . We extend this diagram by introducing two distinct embeddings derived from Z : an invariant embedding Z_I and an equivariant embedding Z_E . These embeddings are designed to separate augmentation-invariant and augmentation-aware features, respectively.

205 C.2 Confounding Effect in Equivariant SSL

206 **Theorem C.1** (Confounding Between Invariant and Equivariant Embedding). *If the data generation*
 207 *process obeys the fork-structured diagram in Figure C.1, then almost surely*

$$Z_I \not\perp\!\!\!\perp Z_E, \quad I(Z_I; Z_E) > 0.$$

208 Theorem C.1 demonstrates that the invariant and equivariant embeddings, Z_I and Z_E , are marginally
 210 dependent due to their shared causal parent in the generative process. From an information-theoretic
 211 perspective, this implies that these two embeddings share mutual information, i.e., $I(Z_I; Z_E) > 0$.
 212 Therefore, the invariant and equivariant embeddings share structural information, as implied by their
 213 mutual dependence.

214 C.3 Proof of Theorem C.1

215 *Proof.* To establish the marginal dependence between Z_I and Z_E , we analyze the causal diagram in
 216 Figure C.1 under the d-separation criterion.

217 In the diagram, both Z_I and Z_E are direct children of Z , forming the structure $Z_I \leftarrow Z \rightarrow Z_E$.
 218 According to Definition C.1, Z_I and Z_E are d-separated only if the node Z blocks the path between
 219 them, i.e., Z is observed. Since Z is not observed in this case, the path is not blocked, and thus Z_I
 220 and Z_E are *not* d-separated.

221 Then, by Lemma C.1, which asserts that d-separation implies conditional independence for almost all
 222 distributions that factorize over the graph, we conclude that Z_I and Z_E are marginally dependent,
 223 i.e., $Z_I \not\perp\!\!\!\perp Z_E$.

224 Finally, this statistical dependence implies that the mutual information between Z_I and Z_E is strictly
 225 positive: $I(Z_I; Z_E) > 0$. \square

226 References

- 227 [1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative
 228 components with random forests. In *ECCV*, 2014.
- 229 [2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework
 230 for contrastive learning of visual representations. In *ICML*, 2020.

- 231 [3] Xinlei Chen*, Saining Xie*, and Kaiming He. An empirical study of training self-supervised
232 vision transformers. In *ICCV*, 2021.
- 233 [4] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi.
234 Describing textures in the wild. In *CVPR*, 2014.
- 235 [5] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsuper-
236 vised feature learning. In *AISTATS*, 2011.
- 237 [6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai,
238 Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly,
239 Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image
240 recognition at scale. In *ICLR*, 2021.
- 241 [7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman.
242 The pascal visual object classes (voc) challenge. *IJCV*, 2010.
- 243 [8] Li Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training
244 examples: An incremental bayesian approach tested on 101 object categories. In *CVPR*
245 *Workshop*, 2004.
- 246 [9] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo
247 Kyröla, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD:
248 training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.
- 249 [10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena
250 Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar,
251 Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a
252 new approach to self-supervised learning. In *NeurIPS*, 2020.
- 253 [11] Sharut Gupta, Joshua Robinson, Derek Lim, Soledad Villar, and Stefanie Jegelka. Structuring
254 representation geometry with rotationally equivariant contrastive learning. In *ICLR*, 2024.
- 255 [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image
256 recognition. In *CVPR*, 2016.
- 257 [13] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshmi-
258 narayanan. AugMix: A simple data processing method to improve robustness and uncertainty.
259 In *ICLR*, 2020.
- 260 [14] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training
261 by reducing internal covariate shift. In *ICML*, 2015.
- 262 [15] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*.
263 MIT Press, 2009.
- 264 [16] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better?
265 In *CVPR*, 2019.
- 266 [17] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for
267 fine-grained categorization. In *CVPR Workshops*, 2013.
- 268 [18] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images.
269 Technical report, University of Toronto, 2009.
- 270 [19] Hankook Lee, Kibok Lee, Kimin Lee, Honglak Lee, and Jinwoo Shin. Improving transferability
271 of representations via augmentation-aware self-supervision. In *NeurIPS*, 2021.
- 272 [20] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. In *ICLR*,
273 2016.
- 274 [21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

- 275 [22] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-
276 grained visual classification of aircraft. Technical report, Visual Geometry Group, University of
277 Oxford, 2013.
- 278 [23] Sharada P. Mohanty, David P. Hughes, and Marcel Salathé. Using deep learning for image-based
279 plant disease detection. *Frontiers in Plant Science*, 2016.
- 280 [24] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large
281 number of classes. In *ICVGIP*, 2008.
- 282 [25] Jeongheon Oh and Kibok Lee. On the effectiveness of supervision in asymmetric non-contrastive
283 learning. In *ICML*, 2024.
- 284 [26] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive
285 metric for improved few-shot learning. In *NeurIPS*, 2018.
- 286 [27] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In
287 *CVPR*, 2012.
- 288 [28] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009.
- 289 [29] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *CVPR*, 2009.
- 290 [30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time
291 object detection with region proposal networks. In *NeurIPS*, 2015.
- 292 [31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng
293 Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei.
294 Imagenet large scale visual recognition challenge. *IJCV*, 2015.
- 295 [32] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*,
296 2020.
- 297 [33] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The
298 caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.
- 299 [34] Yifei Wang, Kaiwen Hu, Sharut Gupta, Ziyu Ye, Yisen Wang, and Stefanie Jegelka. Under-
300 standing the role of equivariance in self-supervised learning. In *NeurIPS*, 2024.
- 301 [35] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun
302 database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.
- 303 [36] Jaemyung Yu, Jaehyun Choi, Dong-Jae Lee, HyeongGwon Hong, and Junmo Kim. Self-
304 supervised transformation learning for equivariant representations. In *NeurIPS*, 2024.