Appendix

This appendix is organized as follows:

- Section A presents mathematical definition for the equivariant representation (Section 1) and derivation of the interventional ERM (Eq. (1)).
- Section B provides the implementation details of the Fig. 2 and Eq. (6). We also elaborate the details of the used Colored MNIST dataset (Fig. 5) and the NICO dataset (Section 5.1).
- Section C shows the results with MAE pretrained feature (Section 4); the attention map visualizations (Section 5.4) and the algorithm complexities (Section 5.4).

A Mathematical Definition & Derivation

A.1 The Mathematical Definition of Equivariance

Let $\mathcal{U}$ be a set of (unseen) semantics, *e.g.*, attributes such as "digit" and "color". There is a set of *independent and causal mechanisms* [13] $\varphi : \mathcal{U} \to \mathcal{I}$, generating images from semantics, *e.g.*, writing a digit "0" when thinking of "0" [18]. A **visual representation** is the inference process $\phi : \mathcal{I} \to \mathcal{X}$ that maps image pixels to vector space features, *e.g.*, a neural network. We define **semantic representation** as the functional composition $f : \mathcal{U} \to \mathcal{I} \to \mathcal{X}$. Let $\mathcal{G}$ be the group acting on $\mathcal{U}$, *i.e.*, $g \cdot u \in \mathcal{U} \times \mathcal{U}$ transforms $u \in \mathcal{U}$, *e.g.*, a "turn green" group element changing the semantic from "red" to "green".

Definition 1. (Equivariant Representation) *Suppose there is a direct product decomposition* $\mathcal{G} = g_1 \times \ldots \times g_m$ *and* $\mathcal{U} = \mathcal{U}_1 \times \ldots \times \mathcal{U}_m$*, where* g_i *acts on* $\mathcal{U}_i$ *respectively. A feature representation is equivariant if there exists a group* $\mathcal{G}$ *acting on* $\mathcal{X}$ *such that:*

$$f(g \cdot u) = g \cdot f(u), \quad \forall g \in \mathcal{G}, \forall u \in \mathcal{U} \tag{A1}$$

e.g., the feature of the changed semantic: "red" to "green" in $\mathcal{U}$*, is equivalent to directly change the color vector in* $\mathcal{X}$ *from "red" to "green".*

As stated in Section 3.2, we follow the definition and implementation in [19] to achieve the sample-equvariant by using contrastive loss. Specifically, by assuming $\mathbf{x} \in \mathcal{X}$ as the feature, we can write the contrastive loss briefly as $\ell = -\log \frac{\exp(\mathbf{x}_i^T \mathbf{x}_j)}{\sum_{\mathbf{x} \in \mathcal{X}} \exp(\mathbf{x}_j^T \mathbf{x})}$. Then, if we use all the samples in the denominator of the loss, we can approximate to $\mathcal{G}$-equivariant features given limited training samples. This is because the loss minimization guarantees $\forall (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{X} \times \mathcal{X}, i \neq j \to \mathbf{x}_i \neq \mathbf{x}_j$. We provide the proof in the following.

Suppose that the training loss ℓ is minimized, yet $\exists \mathbf{x}_a = \mathbf{x}_b \in \mathcal{X}$ for $a \neq b$. Let $\mathbf{x}_i \in \mathcal{X}$ in the denominator, and we have $\mathbf{x}_j^T \mathbf{x}_i = \cos(\theta_{i,j}) \|\mathbf{x}_i\| \|\mathbf{x}_j\|$, where $\theta_{i,j}$ is the angle between the two vectors. When $\mathbf{x}_i = \mathbf{x}_j$, $\cos(\theta_{i,j}) = 1$. So keeping $\|\mathbf{x}_i\| \|\mathbf{x}_j\|$ constant (*i.e.*, the same regularization penalty such as L2), $\mathbf{x}_j^T \mathbf{x}_i$ can be

further reduced if $\mathbf{x}_i \neq \mathbf{x}_j$, which reduces the training loss. This contradicts with the earlier assumption. Hence by minimizing the training loss, we can achieve sample-equivariant, *i.e.*, different samples have different features. Note that this does not necessarily mean group-equivariant. However, the variation of training samples is all we know about the group action of $\mathcal{G}$, and we establish that the action of $\mathcal{G}$ is transitive on $\mathcal{X}$, hence we use the sample-equivariant features as the approximation of $\mathcal{G}$-equivariant features.

A.2 Derivation of Eq. (1)

In this section, we will show the derivation for the backdoor adjustment formula using the three rules of *do*-calculus [15], whose detailed proof can be found in [15,14]. For a causal directed acyclic graph $\mathcal{G}$, let X, Y, Z and W be arbitrary disjoint sets of nodes. We use $\mathcal{G}_{\overline{X}}$ to denote the manipulated graph where all incoming arrows to node X are deleted. Similarly $\mathcal{G}_{\underline{X}}$ represents the graph where outgoing arrows from node X are deleted. We use lower case x, y, z and w for specific values taken by each set of nodes: $X = x, Y = y, Z = z$ and $W = w$. For any interventional distribution compatible with $\mathcal{G}$, we have the following three rules:

Rule 1 Insertion/deletion of observations. If $(Y \perp\!\!\!\perp Z|X, W)_{\mathcal{G}_{\overline{X}}}$:

$$P(y|do(x), z, w) = P(y|do(x), w), \tag{A2}$$

Rule 2 Action/observation exchange. If $(Y \perp\!\!\!\perp Z|X, W)_{\mathcal{G}_{\overline{X}\underline{Z}}}$,

$$P(y|do(x), do(z), w) = P(y|do(x), z, w), \tag{A3}$$

Rule 3 Insertion/deletion of actions. If $(Y \perp\!\!\!\perp Z|X, W)_{\mathcal{G}_{\overline{X}\overline{Z(W)}}}$,

$$P(y|do(x), do(z), w) = P(y|do(x), w), \tag{A4}$$

where $Z(W)$ is the set of nodes in Z that are not ancestors of any W-node in $\mathcal{G}_{\overline{X}}$.

In our causal graph, the desired interventional distribution $P(Y|do(X))$ can be derived by:

$$P(Y|do(X)) = \sum_z P(Y|do(X), Z = z)P(Z = z|do(X)s) \tag{A5}$$

$$= \sum_z P(Y|do(X), Z = z)P(Z = z) \tag{A6}$$

$$= \sum_z P(Y|X, Z = z)P(Z = z), \tag{A7}$$

where Eq. (A5) follows the law of total probability; Eq. (A6) uses Rule 3 given $S \perp\!\!\!\perp X$ in $\mathcal{G}_{\overline{X}}$; Eq. (A7) uses Rule 2 to change the intervention term to observation as $(Y \perp\!\!\!\perp X|Z)$ in $\mathcal{G}_{\underline{X}}$. Therefore, by imposing Eq. (A7) into Eq. (1) of the main paper, we can have:

$$\mathcal{R} = \sum_x \sum_y \sum_z \mathcal{L}(f(\phi(x)), y)P(y|x, z)P(z)P(x). \tag{A8}$$

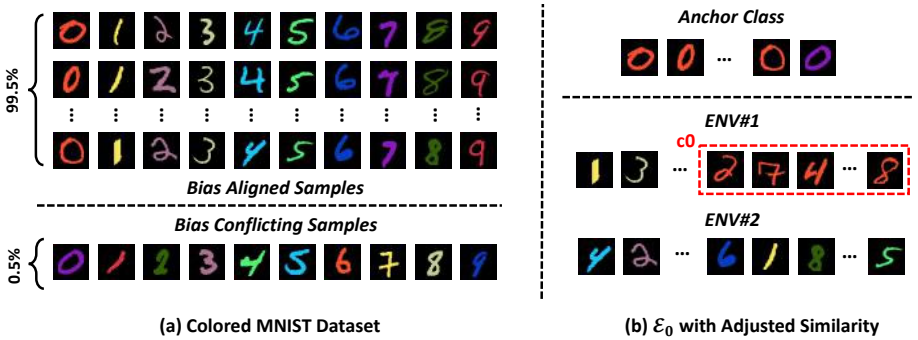

Fig. A1: We illustrate (a) the example images of the Colored MNIST dataset; (b) the generated environment $\mathcal{E}_0$ with adjusted similarity.

B Implementation Details

B.1 Implementation Details of Fig. 2

The t-SNE Visualization. We adopt the t-SNE visualization here to reflect the true data distribution and expect the feature extraction model to be able to accurately structure the relationships between images. Recently, Contrastive Language-Image Pretraining (CLIP) [16] is proposed for solving vision tasks by exploiting contrastive learning with very large-scale noisy image-text pairs. Such large data makes it nearly a "Sufficient Data Situation". And the conventional ERM algorithm can achieve optimal performance, as we introduced in Section 3 of the main paper. Indeed, CLIP achieves inspirational performances on various visual classification tasks. In this paper, we utilize the CLIP pretrained backbone ("ViT-Base/32") to extract feature of the training and testing images of class `Hen` which is randomly chosen. Then we draw t-SNE visualization with the open-source codebase[1].

The Test Accuracy. We trained a ResNet-50 model from scratch on each training set and evaluated on testing images. We calculated accuracy on class `Hen`.

B.2 Implementation Details of Eq. (6)

In Section 4 Step 3, we propose the class-wise IRM to regularize the invariance within each class as stated in Eq. (6). In practice, we adopt a more practical version named REx [11] of Eq. (6) which improves the vanilla IRM in the covariate shift situation. Specifically, [11] discards the dummy classifier w and changes the penalty term of IRM to the variance of risks as the regularization:

$$\mathcal{L}_k = \lambda \mathrm{Var}(\{\ell(1), ..., \ell(\mathrm{e})\}) + \sum\nolimits_{e \in \mathcal{E}_k} \ell(e). \tag{A9}$$

Same as IRM, REx aims to encourage the invariance across different environments, but provides a simpler, stabler and more effective implementation [11].

[1] https://github.com/DmitryUlyanov/Multicore-TSNE

Table A1: Construction of the NICO subset [9,20] for OOD multi-classification. **Context** denotes the context class name, while **Class** represents the object class name. "Long-Tailed Contexts" is the training contexts arranged by the sample number order (from more to less) and "Zero-shot Contexts" represents the context labels only appear in testing rather than training.

Class \ Context	Long-Tailed Contexts							Zero-shot Contexts		
Dog	on grass	in water	in cage	eating	on beach	lying	running	at home	in street	on snow
Cat	on snow	at home	in street	walking	in river	in cage	eating	in water	on grass	on tree
Bear	in forest	black	brown	eating grass	in water	lying	on snow	on ground	on tree	white
Sheep	eating	on road	walking	on snow	on grass	lying	in forest	aside people	in water	at sunset
Bird	on ground	in hand	on branch	flying	eating	on grass	standing	in water	in cage	on shoulder
Rat	at home	in hole	in cage	in forest	in water	on grass	eating	lying	on snow	running
Horse	on beach	aside people	running	lying	on grass	on snow	in forest	at home	in river	in street
Elephant	in zoo	in circus	in forest	in river	eating	standing	on grass	in street	lying	on snow
Cow	in river	lying	standing	eating	in forest	on grass	on snow	at home	aside people	spotted
Monkey	sitting	walking	in water	on snow	in forest	eating	on grass	in cage	on beach	climbing

B.3 Details of Colored MNIST dataset in Fig. 5

Figure A1 (a) shows the example images of the Colored MNIST [12] dataset. As we introduced in Section 4 of the main paper, the Colored MNIST dataset injects *color* bias on each digit. There are 99.5% bias-aligned samples and only 0.5% images are non-bias samples. For example, most of `0` are red. Figure A1 (b) illustrates the generated environment $\mathcal{E}_0$ of `anchor` class `0` with the adjusted similarity using real images. We can clearly observe that the biased color `c0` (*i.e.*, red) of digit `0` distributes differently in Env#1 and Env#2, while other semantics keep invariant. This will encourages the bias color to be removed during the following class-wise IRM process.

B.4 Details of the NICO dataset

In our experiment, we use the NICO subset proposed in [20] as a challenging benchmark to test the proposed EqInv and baselines. Specifically, images in NICO are labeled with a context class (*e.g.*, "on grass"), besides the object class (*e.g.*, "dog"). During training, 7 context classes (Long-Tailed Contexts as shown in Table A1) are chosen for each object class. Next, a long-tailed training dataset is formed by selecting part of the images in each context class with multiplying a ratio. In particular, the ratio for w-th context class ($w \in \{0, \dots, 6\}$) is given by:

$$\text{ratio} = \text{IR}^{w/6}, \tag{A10}$$

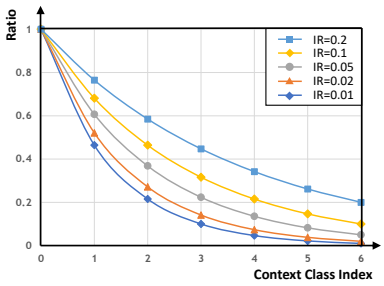

Fig. A3: Plot of context class index against its corresponding ratio under various imbalance ratio (IR).

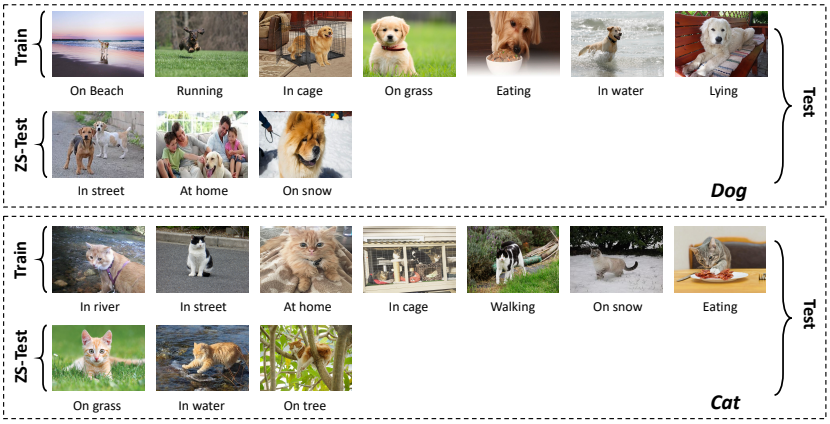

Fig. A2: We list the sample images of each context class using "Dog" and "Cat" as the example in the utilized NICO data subset. **Train**, **Test** and **ZS-Test** denote samples for training, testing and zero shot testing respectively. Note that there is no overlap between training and testing images.

where IR is a hyper-parameter that denotes the imbalance ratio. The effect of IR on ratio is shown in Fig. A3 — lower ratio leads to the harder OOD problem. In the main paper we keep IR = 0.02. During testing, the number of test samples across the 7 context classes is balanced, *i.e.*, 50 samples per context. Moreover, 3 zero-shot context classes are added for each object class as shown in Table A1 (last three columns). These zero-shot context classes have the larger number of test samples (100 samples per context). Therefore, a model that performs well in such split must be robust to both long-tailed and zero-shot problems *w.r.t.* the context class. Fig. A2 shows an example of the NICO subset for "cat" and "dog" during training and testing.

B.5 Experimental Details

More Implementation Details of the Equivariant Learning. As stated in the main paper, we utilize different Self-Supervised Learning techniques in Step 1 Equivariant Learning. For implementation, we train for 800 epochs using ResNet-50/-18 and ViT-Base/16 as the encoder architecture. We just follow the original methods to use the default parameters and training schedules except for some slight changes to adapt to the VIPriors and NICO dataset. Specifically, we set the queue size as 16384 for MoCo-v2 [8,4]. For the NICO dataset, we train the MAE for 2000 epochs and adopt the mixup version of IP-IRM [19] to achieve the reasonable performance. Moreover, please note that we follow the other team's solution [21] of VIPriors Challenge to use both train and val set for the Stage 1 SSL pretraining with no need of the label for all the comparison methods. Then for the second fine-tuning stage, as stated in the main paper, we only use the train set images and labels for training.

Table A2: Recognition accuracies (%) on the VIPriors-10 and NICO datasets with ViT-Base/16 as the feature backbone and MAE [7] as the SSL method. "Aug." represents augmentation. Our results are highlighted.

Model	VIPriors-10		NICO	
	Val	Test	Val	Test
Train from Scratch				
Baseline (ViT-Base/16)	4.74	4.50	32.23	31.46
Random Aug. [5]	5.40	4.92	33.54	31.92
Train from SSL				
MAE [7]	16.04	14.63	54.10	**52.29**
+ EqInv (Ours)	**16.93**	**15.48**	**56.26**	**52.29**
MAE [7] + Random Aug.	16.70	15.39	56.11	52.91
+ EqInv (Ours)	**17.53**	**16.00**	**57.96**	**54.14**

More Implementation Details of the Downstream Fine-tuning. In the main paper, we have introduced most training schedules for ResNet model. Besides the training schedules introduced in the main paper, we set $\lambda = 2, 10, 100$ for VIPriors-10, 20, 50 dataset. This parameter choice also makes sense from intuitive since the demand of the invariance regularization is decreasing with more training samples. Please note that, for fair comparison with data-efficient learning methods, we did not apply any strong data augmentation in our downstream training (after SSL), even though it is common in SSL works. For MAE with ViT-Base/16, we follow the default end-to-end fine-tuning schedule: AdamW as the optimizer with base learning rate 5e-4 using the cosine learning rate decay; the layer-wise learning rate decay is set to 0.65 and weight decay is set to 0.05; the drop path is set to 0.1 and the warmup epochs are 5. We decrease the batch size to 256 and not use the advanced augmentation (*i.e.*, cutmix, mixup, label smoothing) to keep consistent with the ResNet model. For our proposed class-wise IRM, the hidden size of the MLP g is 512 with batch normalize layer and ReLU activation. The output dimension of g is 128, same with SimCLR [3]. We also utilize the weight normailzation [17] on the fc layer f for the stable training.

C Additional Results

C.1 Results with MAE [7] feature

Table A2 shows the recognition accuracies on the VIPriors-10 and NICO datasets with ViT-Base/16 as the feature backbone and MAE [7] as the SSL method. Similar to the Table 1 in the main paper, we can observe that compared to training from scratch, both imposing equivariance and invariance inductive bias can boost the performance. However, we also find that the improvements of considering invariance inductive bias are not such huge compared to that of the ResNet structure. The possible reason is the Visual Transformer [6,10] structure

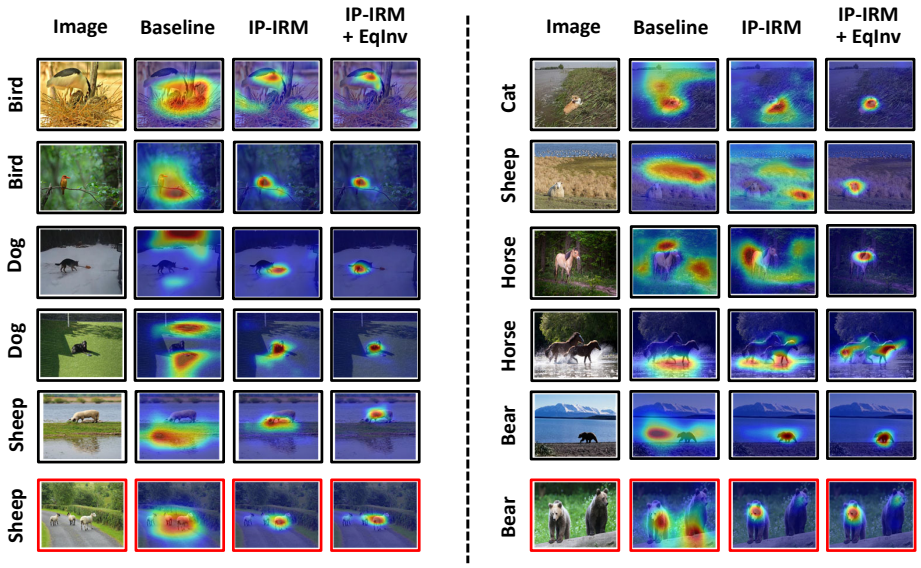

Fig. A4: Visual attention visualizations on NICO dataset with our proposed EqInv and baseline methods. We adopt IP-IRM [19] as the SSL method. The red box represents the failure case.

itself is more robust [2,1] to the distribution shift than the CNN model (*e.g.*, ResNet). That is, the self-attention-like architectures of Visual Transformer have partially achieved the invariance.

C.2 Algorithm Complexities

Table A3: The model size and computational cost comparison between our proposed EqInv and baseline models with different feature backbones.

Model	Params (M)	Flops (G)	MACs (G)	Time (s)
ResNet-18	11.180	3.644	1.822	0.025
+ EqInv (Ours)	11.510	3.646	1.823	0.105
ResNet-50	25.560	8.244	4.122	0.061
+ EqInv (Ours)	26.680	8.246	4.123	0.342
ViT-Base/16	86.570	33.712	16.856	0.065
+ EqInv (Ours)	87.030	33.712	16.856	0.297

We show the model sizes and the computational costs in Table A3. The "Time (s)" denotes the forwarding process time with bs images as input. bs is set to 128 for ResNet and 64 for ViT, based on GPU memory consumption. We can see

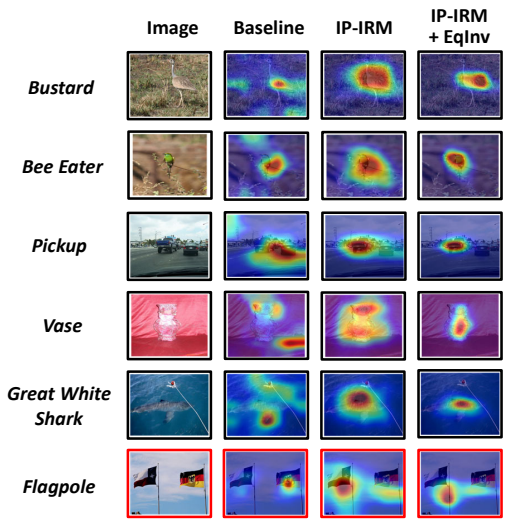

Fig. A5: Visual attention visualizations on VIPriors-10 dataset with our proposed EqInv and baseline methods. We adopt IP-IRM [19] as the SSL method. The red box represents the failure case.

that compared to baseline models, deploying EqInv does not cause many extra network parameters and computation costs. This is because in our EqInv, the invariance inductive bias is implemented by only one learnable mask layer and one MLP layer, bringing little overhead.

C.3 Attention Visualizations

Figure A4 and A5 show the qualitative attention map comparisons between baseline (*i.e.*, training from scratch), incorporating SSL pretraining and our proposed EqInv. We utilize CAM [22] for the visualization. We can clearly observe that:

- Training from scratch (the second column) produces many inaccurate attentions, even totally misses the object area (*e.g.*, the last three rows of Fig. A4 left). This indicates the severe environmental bias of the model trained with the insufficient samples (*cf.* Section 3.1 of the main paper).
- Though incorporating SSL pretraining (the third column) greatly alleviates such problem by imposing the equivariance inductive bias, the model still attends to partial context area. It means the model may still be confounded by the environmental feature during the downstream fine-tuning.
- By further imposing the invariance inductive bias with our proposed EqInv (the last column), the model can achieve much more accurate and tighter attention focusing on the object area. We also display the failure cases in red boxes. We can find that our EqInv cannot accurately attend to multiple

objects (*e.g.*, three sheeps and two bears in Fig. A4) or small objects (*e.g.*, the flagpole in Fig. A5). But our EqInv can still achieve relatively better attention maps compared to comparison methods. We will explore such failure cases in the future work.

References

1. Bai, Y., Mei, J., Yuille, A.L., Xie, C.: Are transformers more robust than cnns? NeurIPS **34** (2021) 7
2. Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., Veit, A.: Understanding robustness of transformers for image classification. In: ICCV. pp. 10231–10241 (2021) 7
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML. pp. 1597–1607. PMLR (2020) 6
4. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020) 5
5. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: CVPR Workshops. pp. 702–703 (2020) 6
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 6
7. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377 (2021) 6
8. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722 (2019) 5
9. He, Y., Shen, Z., Cui, P.: Towards non-iid image classification: A dataset and baselines. Pattern Recognition **110**, 107383 (2021) 4
10. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. ACM Computing Surveys (CSUR) (2021) 6
11. Krueger, D., Caballero, E., Jacobsen, J.H., Zhang, A., Binas, J., Zhang, D., Priol, R.L., Courville, A.: Out-of-distribution generalization via risk extrapolation (rex). arXiv preprint (2020) 3
12. Nam, J., Cha, H., Ahn, S., Lee, J., Shin, J.: Learning from failure: De-biasing classifier from biased classifier. NeurIPS **33**, 20673–20684 (2020) 4
13. Parascandolo, G., Kilbertus, N., Rojas-Carulla, M., Schölkopf, B.: Learning independent causal mechanisms. In: Proceedings of the 35th ICML. pp. 4036–4044 (2018) 1
14. Pearl, J.: Causal diagrams for empirical research. Biometrika **82**(4), 669–688 (1995) 2
15. Pearl, J.: Causality. Cambridge university press (2009) 2
16. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763. PMLR (2021) 3
17. Salimans, T., Kingma, D.P.: Weight normalization: A simple reparameterization to accelerate training of deep neural networks. NeurIPS **29** (2016) 6
18. Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., Mooij, J.: On causal and anticausal learning. In: Proceedings of the 29th ICML (2012) 1
19. Wang, T., Yue, Z., Huang, J., Sun, Q., Zhang, H.: Self-supervised learning disentangled group representation as feature. In: NeurIPS (2021) 1, 5, 7, 8
20. Wang, T., Zhou, C., Sun, Q., Zhang, H.: Causal attention for unbiased visual recognition. In: ICCV (2021) 4

21. Zhao, B., Wen, X.: Distilling visual priors from self-supervised learning. In: ECCV. pp. 422–429. Springer (2020) 5
22. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR. pp. 2921–2929 (2016) 8