Table 8: Test Accuracies on Hard-Spurious-ImageNet-10 with highly spurious backgrounds.

Model	Clean Accuracy	Object Resolution	Group Accuracies CeO	CoO	CeR	CoR
Convnext-Base	85.8	56^2	57.4	38.6	8.2	1.0
		84^2	79.0	71.0	27.0	11.4
		112^2	83.2	79.8	45.4	17.0
ResNet-50	82.2	56^2	45.00	29.6	6.4	0.0
		84^2	70.8	58.4	17.0	9.8
		112^2	79.4	78.6	35.6	28.6
CoATNet	83.4	56^2	20.9	21.8	0.8	0.0
		84^2	71.8	49.0	11.0	4.0
		112^2	75.40	80.8	18.2	23.0
Hiera	85.8	56^2	46.8	22.8	1.0	0.0
		84^2	75.6	55.6	4.0	1.8
		112^2	78.6	74.6	16.0	10.6
MVitv2	86.6	56^2	29.0	15.0	0.4	0.0
		84^2	72.6	47.8	11.2	1.2
		112^2	72.4	66.0	18.4	12.6

A Benchmark Results

The results for Hard-Spurios-ImageNet-10 are given in 8. Similar to main paper, We test the performance of the datasets on 5 different pre-trained architectures: ConvNext-Base (Liu et al., 2022), ResNet-50 (He et al., 2016), CoATNet (Dai et al., 2021), Hiera-Base with MAE (Ryali et al., 2023), and MVit2-small (Li et al., 2022). All models are pretrained on ImageNet1k only. We see that Hard-Spurious-ImageNet-10 has far worse performance on groups CeR and CoR across all architectures and sizes. This indicates that the strength of spurious backgrounds is far greater than that of core features when the size of core features starts to decrease.

B Biases in ImageNet

Figure 11 shows the distribution of center and size scores for different classes in the training data of ImageNet. We calculate these scores using the available bounding boxes for ImageNet training data. Figure 3 refers to the distribution for the validation data.

C Inpaint Anything

The predicted masks from Segment Anything are dilated by a kernel size of 15 to avoid edge effects when the "hole" is filled by LaMa. Some examples of the inpainted data are given in Figure 10.

Figure 10: **left**: Original images with their resized inpainted versions. **right**: Despite inpainting, the background (Bg) consists of cues that help the model predict the background label.

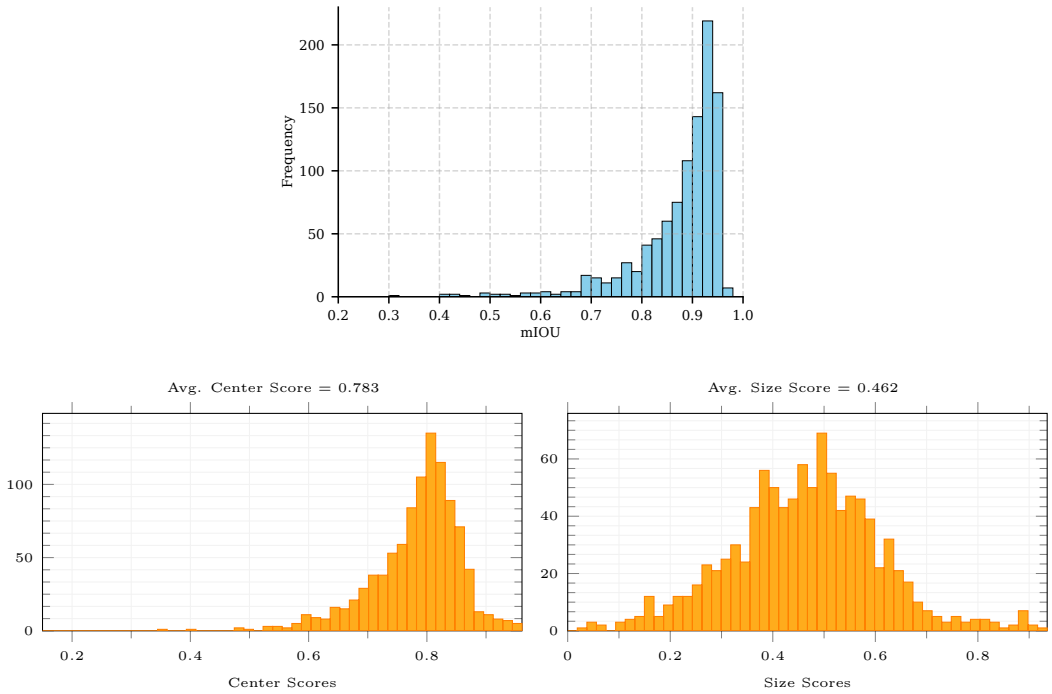

Figure 11: **(top)**: Class-wise mIOU scores between Grounding DINO predictions and ImageNet annotations on the validation set. Averaged mIOU is 0.875. **(bottom)**:Histograms showing distribution of scores in different classes of train data in ImageNet1k dataset. In the main paper, we show the distribution for validation data. This is the same plot for training data. The distributions are very similar.

D True Objects in Background

Ensuring that the backgrounds do not contain true objects depends on the fidelity of provided ImageNet annotations. We perform an additional analysis with a foundation model, Grounding DINO Liu et al. (2024), to extract bounding boxes from the images. We consider similarity scores between Grounding DINO predictions and the ImageNet annotations to analyze the correctness of ImageNet annotations. For ImageNet validation data, we get an overall mIOU of 0.8675 across all classes between both sets of bounding boxes with 139 classes having mIOU value less than 0.8 (see Figure 11 for a histogram by mIOU). This shows that the majority of the classes in ImageNet data have correct bounding boxes and the amount of objects from the foreground class in the background is negligible.

E Hard-Spurious-ImageNet with SAM

We also experiment with using the Segment Anything (Kirillov et al., 2023) model to obtain masks for the objects inside a bounding box and resize it to 3 different sizes (56, 84, and 112). The resized masks are then placed in the center and corner of the inpainted image, similar to the setting described in the main paper. At the moment, we only consider one object per image. Since we have access to ImageNet-annotated bounding boxes, we use them as prompts to be given to SAM. The results are shown in Table 9. Compared to the results in Table 1, the results with SAM are worse, mainly because the resized SAM object masks are not entirely accurate in cases where objects are small and thin, such as insects, etc. Hence, we prefer human-annotated ImageNet bounding boxes.

Table 9: Test Accuracies on Hard-Spurious-ImageNet with SAM Masks.

Model	Clean Accuracy	Object Resolution	Group Accuracies			
			CeO	CoO	CeR	CoR
Convnext-Base	84.43	56^2	46.07	36.07	13.86	6.21
		84^2	61.18	53.92	31.04	22.30
		112^2	67.78	64.69	42.91	13.84
ResNet-50	81.21	56^2	29.33	24.34	6.68	4.36
		84^2	45.17	40.63	19.09	16.24
		112^2	55.24	52.56	31.34	29.87
CoATNet	82.39	56^2	30.57	27.61	7.91	3.93
		84^2	50.94	44.66	21.03	15.63
		112^2	60.60	56.73	33.00	29.30
MVit2	84.49	56^2	37.94	25.88	9.08	2.92
		84^2	54.89	44.73	24.74	15.15
		112^2	63.73	57.94	36.80	30.60
Hiera	83.77	56^2	39.88	27.06	10.34	3.198
		84^2	56.36	46.18	25.13	15.72
		112^2	66.14	60.38	39.15	31.63

F Group Robustness Methods

We use pretrained ResNet-50 trained on ImageNet1k for our experiments. The Base model is fine-tuned with batch size 256, constant learning rate of 0.001 for 20 epochs. The input images are randomly cropped with an aspect ratio in the bounds (0.75,1.33) and finally resized to 224×224. Horizontal flipping is applied afterward. A momentum of 0.9 and weight decay of 0.001 is used. For DFR, we normalize the embeddings using mean and standard deviation of validation data used to train the last layer, and use the same statistics to normalize embeddings of test data. We re-train the last layer for 1000 epochs, learning rate of 1, cosine learning rate scheduler and SGD optimizer with full-batch. We use ℓ_2 regularization with λ set to 100. These hyperparamters are similar to the ones set by Kirichenko et al. (2023) for optimizing the last layer for ImageNet-9 dataset (Xiao et al., 2021). Since, the data distribution in the proposed dataset and ImageNet-9 is similar, we assumed the same hyperparamteres. In case of JTT, models have the same hyperparameters as the ERM trained model. λ_{up} is set to 50.

After extracting the embeddings from the pre-trained ERM model, the embeddings are normalized using **fit_transform()** and **transform()** functions of **sklearn.preprocessing.StandardScaler** for val and test data, respectively. For the JTT model, the images are applied with random resized cropping followed by horizontal flipping. No additional data augmentation is applied afterward. We also experimented with ConvNext-tiny pre-trained on ImageNet-22k and fine-tuned on ImageNet1k. We fine-tune the pre-trained model on the proposed data under various settings. ERM is trained by replicating the long-tailed distribution of the data, while ERM^{easy} is trained only with the easy group. ERM^{all} is trained with equal data points from all groups. DFR is trained by extracting embeddings from ERM, and re-training the last layer only. The number of train and test images is similar to the data setting described in the main paper.

G Dataset Details

Table 11 provides details on the Hard Spurious ImageNet dataset such as the available number of train, validation and test images per resolution and group.

Table 10: Test Performance of different methods on Easy, Medium, and Hard categories in Hard-Spurious-ImageNet. Average accuracy is the average test performance of all the groups combined. The model is Convnext-tiny.

Methods	**Easy**	**Medium**	**Hard**	**Average**
Pretrained	71.14	54.93	29.21	51.75
ERM	76.91	70.63	63.48	70.34
ERM^{easy}	77.82	68.33	51.39	65.85
DFR	74.82	68.66	61.68	68.39

Table 11: Number of Images in every group of Hard Spurious ImageNet.

Set	**Resolution**	**CeO**	**CoO**	**CeR**	**CoR**	**Total**
Train	56	10k	10k	10k	10k	400k
	84	80k	80k	10k	10k	
	112	80k	80k	10k	10k	
Test	56	50k	50k	50k	50k	600k
	84	50k	50k	50k	50k	
	112	50k	50k	50k	50k	
Val	56	20k	20k	20k	20k	240k
	84	20k	20k	20k	20k	
	112	20k	20k	20k	20k	