A^2: Smaller Self-Supervised ViTs Localize Better than Larger Ones

Sreehari Rammohan[a] **Huy Ha**[b] **Carl Vondrick**[a]

[a] Columbia University [b] Stanford University



Figure 1: **Robust Recognition through Exclusion.** Images contain spurious correlations that often mislead classifiers, e.g. a man holding a rein is usually associated with horses. A^2 removes spatially spurious correlations by automatically selecting crops based on a pre-trained model's attention map. While the original model is misled (left), our A^2 trained classifier explicitly attends to only the animal body and correctly predicts "cow" (right).

Abstract

Robust visual classification often depends on localizing the main foreground objects in an image while ignoring contextual distractors. Surprisingly, we find that the last layer attention maps of smaller self-supervised ViTs localize foreground objects better than those of larger ViTs. However, we still need large ViTs, because they extract richer representations from each patch. To get the best of both worlds, good localization and rich representations, we propose A^2, a simple method that leverages this inverse scaling finding by decoupling where to look (a small attention model) from what to extract (a large embedding model): we crop around the attention peaks of a small model and embed the crops with a larger model. A^2 uses entirely pretrained features, requires no group labels, and does not require per-dataset attention or backbone training. Across 5 benchmarks, A^2 is competitive with backbone-matched loss-level methods like DFR, and outperforms end-to-end attention training under stronger distribution shifts. Code can be found here: `https://github.com/SreehariRamMohan/A2`

1 Introduction

Large vision models can approximate a wide range of functions given sufficient training data. However, this flexibility also means the model will exploit all biases and shortcuts [14] in the dataset to minimize the training loss. For example, Figure 1 (left) shows a linear classifier trained on ViT [12] embeddings that falsely predicts that the animal is a horse, when it is actually a cow. Spurious

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

correlations, such as the co-occurrence of horses and people, are pervasive in visual data. Despite progress in curating benchmarks and developing debiasing algorithms, recognition in the presence of such distractors remains a longstanding challenge for computer vision.

Existing methods have tried to correct for these biases by altering the loss function or augmenting the training data. These methods use group labels [27, 17], multiple training passes [20, 39], end-to-end training of attention [2], an LLM to identify spurious features [1], or prompts specifying what to segment [26, 4]. Most of these methods try to correct for the spurious correlation after it has already entered the model, or require external prompts or supervision specifying the bias.

A natural question is whether a pretrained attention model can be used to filter out these spurious cues on its own. In many cases, these tasks can simply be solved by isolating the primary foreground object, something that pretrained vision transformers are already good at doing. We find a surprising result: the attention maps in smaller self-supervised ViTs are better at localizing the foreground object than the attention maps in larger ViTs. This inverse scaling trend runs contrary to the common belief that larger parameter-count models are always better [38, 33, 10].

We design A^2 to leverage this finding by decoupling where to look (the attention model) from what to extract (the embedding model): one model attends to another model's attention, hence *Attending on Attention*. While pretrained attention might not always be enough to localize the features of interest, for many real-world applications (e.g. security camera footage, species identification, medical imaging) the foreground objects are the most task-relevant. This property is also baked into common distribution shift benchmarks (Waterbirds, Spawrious, MetaShift).

Contributions. We make four contributions: 1) we find that smaller pretrained self-supervised ViTs localize foreground objects substantially better than larger ones in last-layer attention, and that localization quality tracks downstream task performance, 2) we leverage this finding to propose A^2, a simple method that decouples *where to look* (the attention model) from *what to extract* (the embedding model) using only frozen pretrained features and is competitive with purpose-built methods for learning under distribution shift, 3) we show that larger models localize in earlier layers, and using an intermediate layer (chosen on a held-out ImageNet validation set) can regain some of the performance of smaller models, and 4) we introduce the MetaShift Animals Dataset, an 8-class animal benchmark with disjoint train and test contexts.

2 Attention in Smaller Models Localizes Better

To directly test whether smaller self-supervised ViTs [5] localize better than larger ones, we use ImageNet [11] (1000 object classes, N=50,000) and Waterbirds (2 classes, N=5,794) where annotated bounding boxes are available. We measure the proportion of last-layer CLS attention captured within the bounding box (taking the union if there are multiple) and also report the hit rate (whether or not the highest attention map value falls within a bounding box).

Figure 2 shows that the trend appears across all five self-supervised families we evaluate. The original DINO [5] shows it at its smallest scale: ViT-S/16 puts 68.9% of its attention mass inside Waterbirds ground truth bounding boxes vs. 64.9% for ViT-B/16. For DINOv2 [24], there is a monotonic decrease in attention mass and hit rate as the ViT gets larger. For ImageNet Val, ViT-S puts 74.5% of its mass inside the ground truth object box vs. 67.6% for ViT-G. The same pattern persists at a much larger scale with DINOv3 [31]: on Waterbirds, ViT-S/16 (21M) puts 76.7% of its mass inside GT bboxes vs. only 57.7% for ViT-7B/16 (6.7B), a 19-point drop while scaling $320\times$ in parameters. Although not monotonic, we observe a similar trend in another self-supervised ViT trained with reconstruction loss, MAE [15], with the model best at localization being the smallest, ViT-B/16 (for Waterbirds, ViT-B/16 puts 47.4% of its mass inside the ground truth object box vs. 37.7% for ViT-H/14). The same monotonic decrease in attention mass holds for iBOT [40], a fifth self-supervised family combining masked image modeling with self-distillation: ViT-S/16 puts 69.6% of its mass inside Waterbirds GT bboxes vs. 55.7% for ViT-L/16. In contrastive language-image pretrained ViTs (OpenCLIP) the trend does not hold: ViT-B/16 is both the smallest and the worst at localization with a particularly low hit rate of 0.166 compared with ViT-L/14 (0.815). Appendix Table 4 reports precise values for all metrics.

On patch size, objective, and pretraining data. Finer patches localize better. DINOv1 ViT-S/8 and ViT-B/8 beat their corresponding same-sized patch size 16 counterparts (ViT-S/8 outperforms

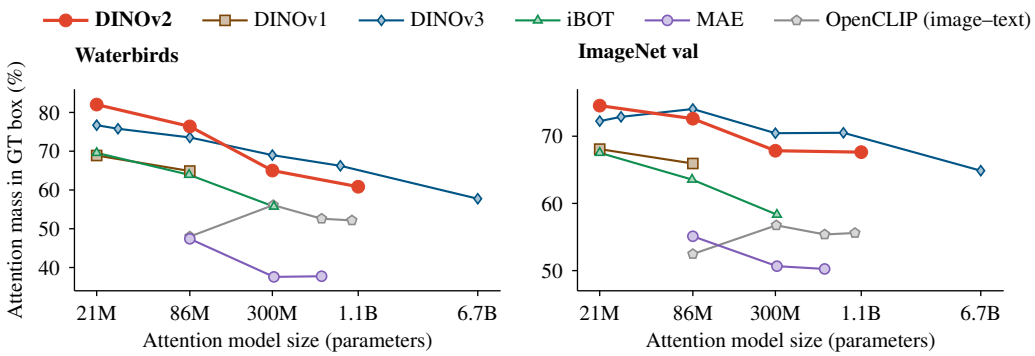

Figure 2: **Smaller self-supervised ViTs localize better.** We show the proportion of last layer attention mass that is inside ground-truth bounding boxes versus model size across six pretraining families, on Waterbirds (left) and ImageNet Val (right). All five self-supervised families (DINOv1, DINOv2, DINOv3, MAE, iBOT) slope downward as model size grows; the contrastive image-text family (OpenCLIP, gray) does not.

ViT-S/16). Keeping the patch size fixed, the smaller models localize better (ViT-S/16 is better than ViT-B/16). The plateau of MAE in localization when going from MAE ViT-L/16 to MAE ViT-H/14 (Figure 2) is likely due to the smaller patch size offsetting the inverse scaling trend when moving up in model size. When going from MAE B/16 to MAE L/16, Waterbirds mass in bbox declines 0.098, consistent with inverse scaling, but when going from MAE L/16 to MAE H/14, the smaller patch size appears to counteract inverse scaling, resulting in a slight increase (+0.002) in Waterbirds mass in bbox.

For objective, we compare DeiT-S/16 (supervised) with DINOv1 ViT-S/16 (self-distilled) as these models are identical except for the objective (both trained on ImageNet-1K). Small size alone is not enough for strong localization, as the DeiT model localizes poorly.

Finally, the image-text pretraining trend is not stable, varying the pretraining data and comparing OpenCLIP vs. OpenAI CLIP (LAION-2B vs. WIT-400M) suggests that image-text pretrained ViTs show no consistent trend with localization. The same architecture and objective trained on two different corpora yield very different attention hit rates (0.815 vs. 0.172) on Waterbirds. Full results in Appendix Section B.

3 Attending on Attention

Motivated by this finding, we propose Attending on Attention (A^2), a method where a larger model attends to the attention of a smaller model. A^2 has two variants: A^2_{ZS}, which is fully zero-shot, and A^2_{LR} which fits a logistic regression head using crop embeddings from a frozen pretrained model. As shown by Figure 3, A^2 operates in two stages: 1) selection of crops 2) classification based on crops.

Given an image x, we predict the label $\hat{y}$ by independently embedding image crops c_i using a transformer $\mathcal{T}$, concatenating the representations, and making a prediction using a classifier f. The main idea of our approach is that the crops are computed using the attention maps from a second (possibly different) transformer:

$$\hat{y} = f([\mathcal{T}(c_1),\ \ldots,\ \mathcal{T}(c_k)]) \quad \text{where} \quad c_i = \text{Crop}_i(x;\ \text{Attn}(x)) \tag{1}$$

where $\mathcal{T}(\cdot)$ is the CLS embedding of a vision transformer. This approach effectively turns the soft-attention Attn into a hard attention [36]. While $\mathcal{T}$ and Attn could be from the same model, this decoupling also allows us to use one transformer to estimate the attention and a different transformer to predict the label.

3.1 Crops as Hard Attention

We crop the image, x, around the highest-attention regions in the attention map Attn(x) computed from a vision transformer. This converts the transformer's soft attention to hard attention, allowing only the most attended-to and relevant regions to enter the downstream representation.

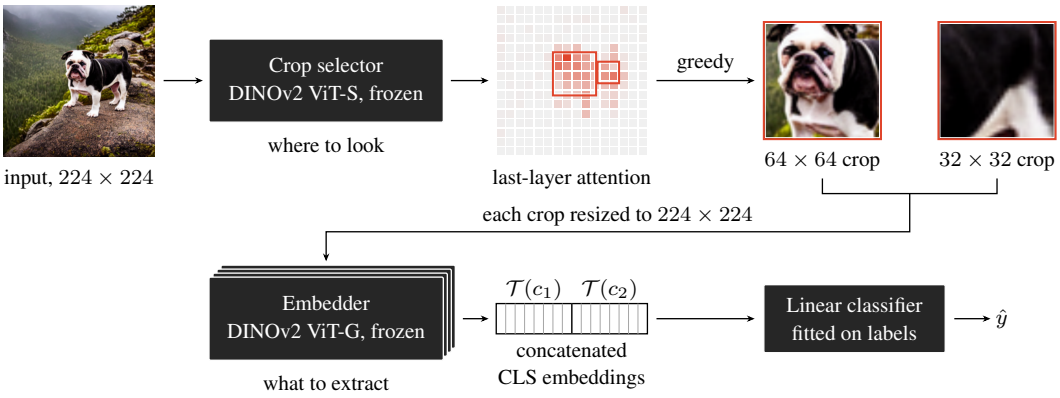

Figure 3: **Overview of A^2.** Given an input image, x, we use an attention model to select crops, $\text{Crop}_i(x;\ \text{Attn}(x))$, which are resized and embedded using a (possibly different) embedding model, $\mathcal{T}$. The concatenated embeddings $[\,\mathcal{T}(c_1),\ \ldots,\ \mathcal{T}(c_k)\,]$ are passed to a classifier f to predict $\hat{y}$.

Concretely, we use a ViT (e.g., DINOv2/v3 or CLIP) to produce an attention map by taking the dot product of the CLS token's query with every key [34]. Because large ViT models have multiple heads, we choose to compute the mean attention map over all heads (same attention computation as is used in Section 2). We upsample the attention matrix to be the same size as the input image using bilinear interpolation.

Given an $M \times M$ attention map, we greedily select non-overlapping crops $c_1, \ldots, c_n$ from a predefined list of crop sizes $s_1, \ldots, s_n$ (ordered from largest to smallest). Each crop is chosen so as to maximize the sum of attention mass within it while also not intersecting with a previously placed crop. Formal definitions for both head aggregation and greedy crop selection are given in Appendix C.

3.2 Classification

Each crop, c_i, is resized to 224×224 before being embedded using $\mathcal{T}$ into a d dimensional representation. $\mathcal{T}$ need not be the same model used to compute Attn(x). In the A^2_{LR} variant, we concatenate the crop embeddings $[\,\mathcal{T}(c_1),\ \ldots,\ \mathcal{T}(c_k)\,] \in \mathbb{R}^{k \cdot d}$ and fit a logistic regression classifier to predict $\hat{y}$. Training happens on these frozen crop embeddings and does not require group labels. In the zero-shot variant, A^2_{ZS}, we select a single crop, c_1, embed it using CLIP [25] ($\mathcal{T}$ = CLIP), and classify by comparing the cosine similarity of the crop embedding with those of the possible label embeddings ("a photo of a {}").

While the attention map produced from the SSL model is task agnostic, its large scale pretraining allows it to appropriately focus on the relevant foreground objects [5] – this is often sufficient for performing well on tasks, as shown by A^2's performance. Learning a task conditioned selector using an approach like iFAM [2] or GradCAM [28] often overfits to tasks with spurious distractors, underperforming task agnostic selection.

4 Experiments

In our experiments, we use five datasets: Spawrious O2O Hard, Spawrious M2M Hard [23], Waterbirds [27], MetaShift Cat vs. Dog [19], and a new dataset we created, MetaShift Animals (Figure 4). Each MetaShift image carries a *context tag* from Visual Genome (e.g., `tree`, `water`, `rope`) which acts as the spurious feature; we split the context tags in each class such that those appearing in training are disjoint from those at test time. The full dataset card with all eight classes and per-class image counts is in Appendix D. Code to construct the MetaShift Animals dataset can be found here: `https://github.com/SreehariRamMohan/MetaShift_Animals`

We design experiments to answer the following questions:

- **Robustness**: Does A^2 improve robustness to common spatial spurious correlations over full-image, last-layer retraining (DFR [17]), and attention retraining baselines [2]? Can A^2 be used with existing methods for spurious correlations that operate at the loss level?

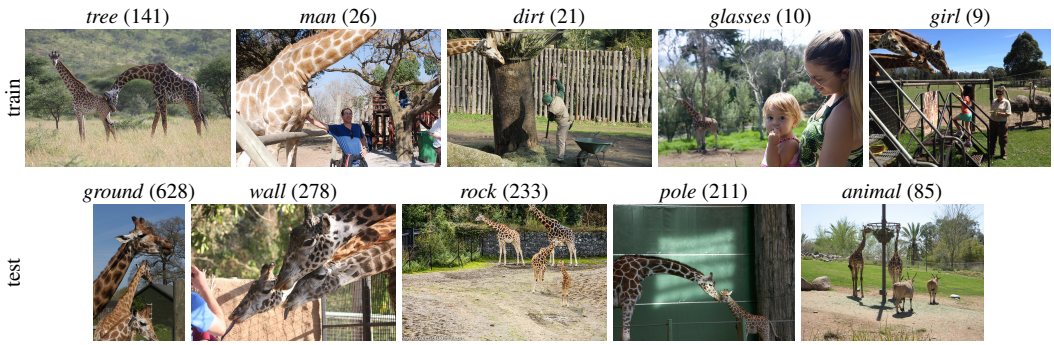

Figure 4: **MetaShift Animals (ours): disjoint train/test contexts.** One of our eight classes (`giraffe`) shown across several Visual Genome context tags. The top row of each pair contains training images and the bottom row contains test images; the per-image label is the context tag (spurious feature). Train tags (e.g., `tree`, `man`, `dirt`) and test tags (e.g., `ground`, `wall`, `rock`) are disjoint by construction. Full panels for four representative classes in Appendix Figure 14.

Table 1: **Worst-Group Accuracy.** This table reports worst-group accuracy (WGA, %) on five distribution-shift benchmarks. MetaShift Animals reports worst-class accuracy (WC) since its groups are class-context pairs with disjoint train/test contexts. A^2 improves worst-group accuracy over full image ViT-G linear probes on three of five benchmarks. Methods that require group labels at training time are shown above the horizontal line and are not directly comparable to the methods below. Backbones are DINOv2 [24] unless prefixed (e.g., OpenCLIP); for A^2 rows, the Backbone column reads attention $\times$ embedding. Bold indicates the best value per column among methods not requiring group labels. Values are means over 5 random seeds where applicable; standard deviations in Appendix Tables 18, 20, 21, 22.

Method	Backbone (attn $\times$ emb)	Spawrious O2O	Spawrious M2M	Waterbirds	Cat vs. Dog	Animals (WC)
Deep Feature Reweighting (requires group labels)						
DFR [17]	ViT-S (21M)	74.9	49.9	90.4	54.5	72.0
DFR [17]	ViT-B (86M)	84.5	69.7	94.0	65.9	73.5
DFR [17]	ViT-G (1.1B)	89.9	73.9	96.4	74.2	76.6
DFR† + A^2	ViT-S $\times$ ViT-S	88.8	64.6	91.0	67.7	72.9
Linear probe on full image (no crops)						
LR (full image)	ViT-S (21M)	71.7	25.0	72.7	56.0	63.2
LR (full image)	ViT-G (1.1B)	77.7	52.4	92.8	76.8	**75.9**
Attention learning						
iFAM [2] (K=4)	ViT-B (86M)	80.6	71.4	96.0	72.1	68.8
iFAM [2] (K=8)	ViT-B (86M)	76.1	72.4	**96.1**	73.5	63.5
GradCAM [28]	ViT-S (21M)	45.5	16.4	68.4	43.7	61.7
Zero-shot CLIP-based references (no crops)						
TTR [22]	OpenCLIP ViT-L-14	70.2	83.6	76.9	82.7	49.1
OpenCLIP ZS (full image)	ViT-L-14	82.0	82.1	46.9	**88.8**	58.8
Attending on Attention (ours)						
A^2_{LR} cross-model	ViT-S $\times$ ViT-G	79.9	64.6	95.3	76.6	70.8
A^2_{LR}	ViT-S $\times$ ViT-S	80.2	58.1	80.4	68.3	67.0
A^2_{ZS}	ViT-S $\times$ OpenCLIP ViT-L-14	**85.0**	**85.4**	64.5	**88.8**	54.7

$\dagger$ = uses group labels via the DFR step.

- **Decoupling**: Can decoupling the model used to compute attention and embed crops improve performance as observed in Section 2? Is localization performance tied to downstream task performance?

For our main A^2_{LR} cross model results, we use a shared crop configuration across all five benchmarks: one 64×64 crop followed by one 32×32 crop. Crops are chosen using DINOv2 ViT-S attention and embedded with DINOv2 ViT-G. We believe these settings work well in most cases, but investigate various crop configurations and settings (sizes, upscale options, top-k attention patches instead), along with computational costs in Appendix E.

Table 2: **O.O.D. Classification Accuracy.** A^2 does not sacrifice test accuracy, it exceeds full image ViT-G linear probes on three of the five benchmarks and comes within 1.7 points on the other two. Layout and notation are the same as Table 1.

Method	Backbone (attn × emb)	Spawrious O2O	Spawrious M2M	Waterbirds	Cat vs. Dog	Animals
Deep Feature Reweighting (requires group labels)						
DFR [17]	ViT-S (21M)	86.1	73.4	92.4	58.7	88.3
DFR [17]	ViT-B (86M)	90.1	84.9	95.5	68.6	90.3
DFR [17]	ViT-G (1.1B)	95.7	88.0	98.1	76.0	91.4
DFR† + A^2	ViT-S × ViT-S	95.2	86.2	91.9	69.4	88.9
Linear probe on full image (no crops)						
LR (full image)	ViT-S (21M)	80.5	66.6	88.7	59.8	83.2
LR (full image)	ViT-G (1.1B)	87.8	79.7	97.8	78.6	**90.6**
Attention learning						
iFAM [2] (K=4)	ViT-B (86M)	92.0	86.1	**98.9**	78.3	86.6
iFAM [2] (K=8)	ViT-B (86M)	91.2	86.0	98.8	81.1	88.2
GradCAM [28]	ViT-S (21M)	67.1	50.7	86.3	48.1	77.7
Zero-shot CLIP-based references (no crops)						
TTR [22]	OpenCLIP ViT-L-14	84.4	92.1	87.2	88.1	69.2
OpenCLIP ZS (full image)	ViT-L-14	89.8	92.4	73.5	**92.4**	90.4
Attending on Attention (ours)						
A^2_{LR} cross-model	ViT-S × ViT-G	89.2	86.0	98.3	77.3	88.9
A^2_{LR}	ViT-S × ViT-S	91.0	82.4	93.0	69.8	85.1
A^2_{ZS}	ViT-S × OpenCLIP ViT-L-14	**92.8**	**94.0**	80.3	89.7	89.7

† = uses group labels via the DFR step.

4.1 A^2 improves robustness and complements loss-level methods

We show our consolidated results in Tables 1 (worst-group accuracy) and 2 (test accuracy), with the full results and ablations in Appendix Tables 18, 19, 20, 21, 22. To keep the main comparison backbone-matched (ViT throughout), we omit the historical ResNet-50 baselines (ERM, CORAL, CausIRL, JTT, GroupDRO) from Tables 1 and 2. They are reported in Appendix Tables 18 and 20, where A^2 is competitive with them on out-of-domain test accuracy.

A^2 significantly improves performance when the spatial spurious signal dominates. At fixed backbone, adding A^2 crops to a ViT-S linear probe improves Spawrious M2M Hard WGA from 25.0% to 58.1% ($+33.1\%$). In zero-shot, holding the OpenCLIP ViT-L-14 embedder fixed, A^2 crops improve Waterbirds WGA from 46.9% to 64.5% ($+17.6\%$). On benchmarks where the full-image ViT-G probe is already strong (MetaShift Animals, Cat vs. Dog accuracy), A^2 trails slightly.

A^2 is complementary to loss-level methods (e.g. DFR, which does last-layer retraining on an equally weighted dataset) and can be used in tandem with these methods to improve performance. Fixing the backbone size to ViT-S, A^2_{LR} outperforms DFR on $4/5$ datasets in test accuracy ($3/5$ in worst-group accuracy). A^2 + DFR adds another $+13.9$ points WGA over DFR alone on Spawrious O2O Hard. A^2_{ZS} outperforms TTR (an input filtering method [22]) on $4/5$ datasets. On the remaining dataset, Waterbirds, A^2_{LR} outperforms TTR by 5.8 points in accuracy and 3.5 points in WGA. Full details of our TTR implementation are in Appendix P.

A^2 vs. end-to-end attention training (iFAM). We compare against iFAM [2], an attention tuning method that learns its attention head and ViT-B backbone jointly on a target dataset. We ran iFAM at both K=4 and K=8 part-discovery configurations on every benchmark and report both in Tables 1 and 2. In Waterbirds, where train and test contexts overlap, iFAM wins (96.1% vs. our 95.3% WGA at K=8), though iFAM's WGA drops 18.2 points when tested on a subset of off-center birds (while A^2 drops 10.6 points), showing it may be overfitting to the center bias of the birds in the dataset (Appendix Table 30). On the harder distribution-shift dataset MetaShift Animals, A^2 wins in both accuracy and worst-class accuracy. The pattern suggests that frozen pretrained DINOv2 attention generalizes across context shifts in a way that iFAM's per-dataset training does not.

In MetaShift Cat vs. Dog (binary classification), the OpenCLIP [6] zero-shot classifier outperforms the A^2_{LR} classifier. The OpenCLIP classifier possesses very strong image-text grounding for the cat and dog labels through its large-scale pretraining. The logistic regression classifier does not start with these priors and the slightest amount of spurious information that leaks into the representations can lead to decreased performance. When the task becomes more difficult (8-way classification in MetaShift Animals), A^2_{LR} improves over OpenCLIP zero-shot, $+12$-point gain in worst-class accuracy.

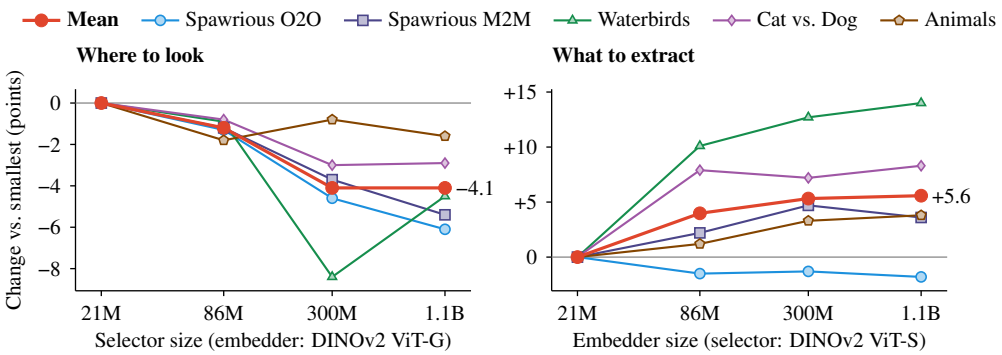

Figure 5: **Where and What Scaling Laws.** Larger embedding models but smaller attention models improve performance. Each panel plots each dataset's standard metric (test accuracy for Spawrious; worst-group accuracy for Waterbirds and Cat vs. Dog; worst-class accuracy for Animals) relative to the smallest model (thin colored) and the mean of those relative changes across 5 datasets (bold red). **Left (Where to Look)**: With the embedding model fixed to DINOv2 ViT-G, every dataset curve slopes downward as the attention model grows. **Right (What to Extract)**: With the attention model fixed to DINOv2 ViT-S, most dataset curves slope upward as the embedding model grows. Absolute values are in Appendix Tables 23, 24, 25, 26, and 27. The DINOv3 [31] family shows the same inverse-scaling-of-attention pattern (Appendix Figure 21).

Finally, in Table 3, to isolate the additional benefit of attention-guided selection over generic cropping, we compare various crop selection policies while keeping the embedding model and crop configuration constant. Small model attention for crop selection achieves the highest worst group accuracy, outperforming random, center, and GradCAM chosen crops.

Table 3: A^2 vs. other generic cropping policies

Crop selector	Spawrious O2O	Waterbirds
Full image (no crops)	77.7	92.8
Random placement	25.9 ± 1.8	47.4 ± 2.0
Center placement	43.5	86.3
GradCAM [28]	49.2	62.9
ViT-G attention, last layer	73.8	89.9
ViT-S attention, last layer (A^2)	**79.9**	**94.4**
GT-box-centered crop (uses annotations)	—	93.8

4.2 Localization quality predicts downstream task performance

A natural question is whether downstream A^2 performance mimics the trends seen in Section 2.

For A^2_{LR}, we fix the crop embedder to be DINOv2 ViT-G, and vary the model used to compute attention (left panel in Figure 5). To test the direct scaling of embedding size, we fix the attention model to be DINOv2 ViT-S, and vary the model used to embed the crops (right panel in Figure 5). For A^2_{ZS}, we fix the embedding model to OpenCLIP ViT-L/14 and vary the crop selector within the DINOv2 family and OpenCLIP family.

Across both A^2 variants, localization ability of the underlying self-supervised model is indicative of downstream task performance. We see the same downward sloping trend in task performance as the DINOv2 model used to select a crop scales (Figure 6). OpenCLIP does not show this trend, and worst-group accuracy follows the same family pattern (Appendix Figure 23). On Spawrious and Waterbirds, A^2_{ZS} using ViT-S attention improves the test accuracy and worst-group accuracy over standard CLIP classification using the full image (improving WGA for Waterbirds from 46.9% to 64.5%). Across all 5 datasets, in $18/20$ cases, the best results are achieved with ViT-S (the smallest model). In fact, pairing a small selector with a large embedder (ViT-S $\times$ ViT-G) outperforms using the large model for both (ViT-G $\times$ ViT-G) on every dataset. Full cross-model pairings in Appendix Tables 23-27.

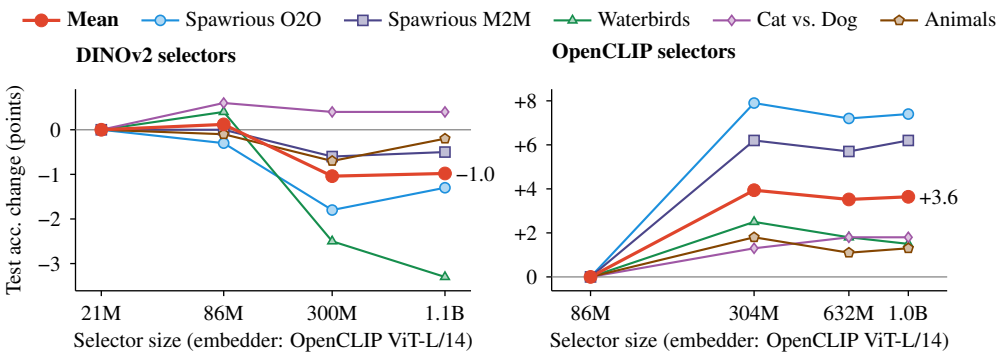

Figure 6: **Downstream Task Performance.** A^2_{ZS} tracks localization quality instead of model size. We fix the embedder to OpenCLIP ViT-L/14 and vary the attention model. Each panel plots per-dataset performance relative to the smallest model in that family (thin colored) and the mean of those relative changes across 5 datasets (bold red). **Left**: With a DINOv2 crop selector, performance *decreases* as the attention model grows on most datasets (curves below zero). **Right**: An OpenCLIP selector does not show the same inverse scaling, ViT-B/16 (the smallest) is the worst. Appendix Figure 23 shows the worst-group accuracy version; Appendix Table 42 reports the precise absolute values.

5 Discussion

Inverse Scaling in Self-Supervised ViTs.

We first rule out simple head averaging as the contributor (ViT-G has 24 attention heads vs. 6 in ViT-S, and averaging more heads should, by concavity of entropy, yield more diffuse attention). We explored trying different head aggregation strategies (Appendix Table 35), and while this ultimately reduced the attention entropy of the selection map, it did not increase downstream task performance (the models were concentrating attention on objects besides the foreground).

Next, we look at downstream localization from the attention map at every layer in the ViT — we notice a premature peak in localization ability for larger models (which also coincides with the model shifting a majority of its attention from patch tokens to register tokens, Figure 12). In Appendix Figure 11, we are able to reproduce the observation in [9] of high-norm tokens appearing in the attention maps of larger ViTs midway through the model, but note that this inverse scaling effect still holds in DINOv2 *with registers* (see Figure 7).

Lappe et al. [18] observed that even with registers, while the attention maps in models with registers are visually cleaner, they do not properly aggregate local information and pay a disproportionate amount of attention to register tokens over patch tokens. In ViT-G on Waterbirds, the fraction of CLS attention on register tokens grows from 41% at the peak-localization layer (block 32) to 55% at the last block, while the patch-side attention falls from 57% to 40% and the patch-side mass-in-bbox drops 13 points over the same span. ViT-S and ViT-B never undergo this register takeover and continue to localize through their last block without a premature peak.

Rethinking the use of last layer attention

The late stage degradation seen in Appendix Figure 12 naturally raises the question of whether we can use the same model for selection and embedding, provided we select from the image at the peak localization layer.

We use the ImageNet Validation set to calibrate, choosing the layer whose attention map places most mass in the ground truth bounding boxes. Once found, we keep this selection choice fixed for our five experiments to test transfer. We see particularly pronounced gains on the largest model size from the MAE backbone, where using block 18 instead of block 31 for attention yields over 20 points more in worst group/class accuracy for Waterbirds and MetaShift Animals (Appendix Table 45). In the 60 cases where the selected layer index differs from the terminal index, 40 result in improved performance, 19 reduce performance, and 1 leads to no change.

Using the smallest model in the family as a dedicated selector is still preferable over using the peak localization layer in the largest model in each family for three reasons: 1) using the smallest model

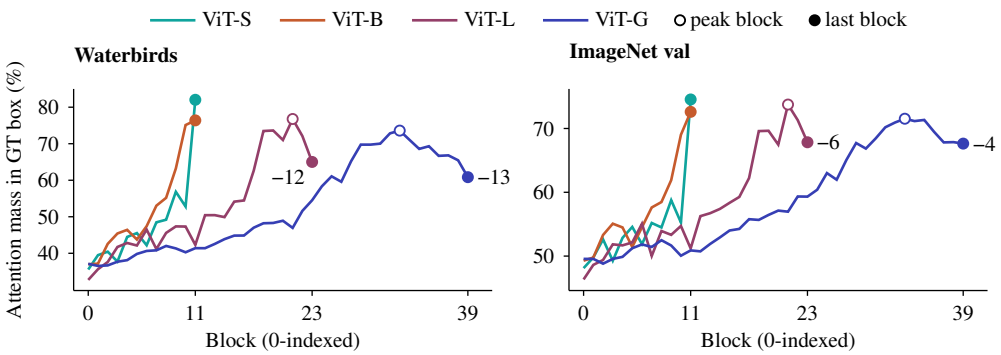

Figure 7: **Per-layer CLS-to-patches attention mass for DINOv2 with register tokens [9].** Filled dots mark each model's last block (the layer used in our other figures) and open dots its peak; numbers give the drop from peak to last block, in points. ViT-S and ViT-B peak at the last block; ViT-L peaks at block 21 of 24 and ViT-G at block 32 of 40, dropping 12–13 points on Waterbirds (left) and 4–6 points on ImageNet val (right) by the final block. Appendix Figure 10 shows the same plot for DINOv2 *without* registers; per-layer hit-rate counterparts are in Appendix Figures 9 and 11.

in a family as a dedicated selector scores higher in $16/25$ cases compared with using the largest model's peak localization layer (see Appendix Table 46), 2) the smallest model in the family is computationally more efficient for attention computation, and 3) A^2 forgoes the need for calibration to find the peak localization layer.

End-to-end attention training overfits; pretrained attention generalizes. End-to-end attention training (iFAM) [2] wins on Waterbirds, where there is no strong distribution shift between train and test, but the dataset is heavily biased toward placing the bird at the center; iFAM appears to lean on this bias: its WGA drops 18.2 points from 100.0% on the 1% most-centered test images to 81.8% on the 1% most off-center, while A^2's drop is roughly half as steep ($83.3\% \rightarrow 72.7\%$; Appendix Table 30). As the test distribution strays further from train, A^2 actually outperforms iFAM despite using entirely pretrained features and a linear head: on MetaShift Animals, where train and test contexts are disjoint, A^2 beats iFAM by 0.7 points in accuracy and 2 points in worst-class accuracy. Pretrained attention offers a gentler degradation curve than fine-tuned attention, which tends to overfit train-task distribution biases.

6 Related Work

6.1 Localization of SSL models

Isolated instances of inverse scaling of localization appear in prior work, but if present, they are entangled within the method itself. To the best of our knowledge, we are the first to observe inverse scaling of localization as a general property of self-supervised model pretraining. The authors of LOST [30] found that DINOv1 ViT-S was slightly better at localizing using their method for single object discovery than DINOv1 ViT-B (61.9/64.0/50.7 vs. 60.1/63.3/50.0) in CorLoc performance on VOC07/VOC12/COCO_20K — however TokenCut [35] mentions that the seed expansion heuristic in LOST is very sensitive to backbone, so identifying this as a scaling trend through the method is tenuous. In the vision language space, GEM [3] found that GEM-ViT-B/16 consistently outperforms GEM-ViT-L/14 on localization tasks, but this is only after using self-self attention rather than the model's own attention.

6.2 Data augmentation

Approaches that modify what the model attends to fall into two broad buckets: those that try to include more visual information and those that try to include less. S^2 [29] represents the former, showing that increased performance can be achieved by constructing representations based on the merging of multiple image scales (e.g., concatenating 224^2 and 448^2 features), and that scaling on scales can outperform scaling on model size. In contrast, A^2 focuses on reducing the number of image tokens the model sees.

In the space of approaches that try to include less visual information, LookWhere [13] also splits vision into where to look followed by what to extract. A selector (distilled from a self-supervised teacher) predicts patch importance and passes the top-k patches (along with low-resolution selector tokens) to a deeper extractor network for processing. iFAM [2] uses a two-stage system trained jointly on each target dataset: stage 1 learns task-specific binary attention masks via part-discovery losses, and stage 2 fine-tunes a ViT classifier on the masked input using class labels. TTR [22] improves CLIP's zero-shot classification abilities by first isolating a "core object" at test time for each image, which is found by a combination of PCA and Kmeans clustering on the CLIP patch embeddings together with a hand-crafted prompt ("a photo of a bird"). LookWhere, iFAM and TTR are similar to A^2 in that they can keep the spurious feature from entering the classifier; however, iFAM and LookWhere train their selectors (LookWhere's focus is on efficiency, not robustness) and TTR uses clustering and a separate prompt to isolate the main cluster. A^2 uses solely frozen attention from pretrained SSL models.

Segmentation models like SAM2 [26] or SAM3 [4] require concept prompts, clicks, boxes, or masks as input to the model to know what to segment. This requires the user to know the relevant object to segment at evaluation time. A^2 identifies the salient regions using the model's own attention.

6.3 Regularization, reweighting, and retraining

Assuming one has access to group labels, GroupDRO [27] trains a model to minimize the maximum loss over all groups. In Deep Feature Reweighting (DFR) [17], the authors show that an ERM trained classifier can be modified to perform well on spurious correlation tasks by simply re-training the final layer on a small validation set where all groups are represented equally. When benchmarking against DFR, we use the same ViT backbone to ensure fair comparisons.

When one doesn't have access to group labels, JTT [20] proposes a two-stage system, first fitting an ERM model and then training a second model that upweights the data the first model has difficulty predicting. CnC, Correct-N-Contrast [39], similarly uses a two-stage ERM-then-contrastive system, encouraging same-class samples from different groups to be close together in representation space using supervised contrastive learning [16].

Finally in the zero-shot setting, Roboshot [1] uses a language model given a task description to generate spurious and helpful text descriptions, then projects these descriptions into the input embedding space to amplify helpful components and suppress spurious ones. TIE [37] uses translation instead of projection, translating the image embedding away from a spurious vector.

All these methods intervene after the spurious feature has entered the representations through loss reweighting, retraining, or embedding adjustment. A^2 operates upstream at the input level, preventing the spurious correlation from entering the representation.

7 Scope, Limitations, and Future Work

A^2 helps when the label is associated with the dominant foreground object and the distractor or irrelevant aspects of the image can be spatially separated. CelebA is a good example of a failure case where default attention is not task relevant, requiring a small attention adapter (documented in Appendix Section H). Our inverse scaling finding can be potentially utilized in future self supervised model training to improve attention map quality and mitigate the effects of inverse scaling. For example, the peak localization layer's attention map can serve as a distillation target for the final layer's attention. It would be interesting to see if doing this gives large SSL models some of the robustness benefits of A^2 in a single forward pass.

8 Conclusion

Self-supervised ViTs show inverse scaling of localization: smaller models are better than larger models at identifying foreground objects. Furthermore, this trend directly translates into A^2, with performance decreasing as the size of the model used to select crops increases. By preventing distractors from entering the model through hard crops, A^2 is competitive with loss-level baselines and beats end-to-end attention training (iFAM) under stronger distribution shifts while being simple and interpretable.

Acknowledgements

SR and CV are partially supported by the National Science Foundation's Center for Smart Streetscapes (CS3), the National Science Foundation's AI Institute for Artificial and Natural Intelligence, the NSF ACED Award #2435757, and the NSF RI Award #2403016.

References

[1] Dyah Adila, Changho Shin, Linrong Cai, and Frederic Sala. Zero-Shot Robustification of Zero-Shot Models. *arXiv e-prints*, art. arXiv:2309.04344, September 2023. doi: 10.48550/arXiv.2309.04344.

[2] Ananthu Aniraj, Cassio F. Dantas, Dino Ienco, and Diego Marcos. Two-stage vision transformers and hard masking offer robust object representations. In *International Conference on Pattern Recognition (ICPR)*, 2026.

[3] Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers. *arXiv preprint arXiv:2312.00878*, 2023.

[4] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment Anything with Concepts. *arXiv e-prints*, art. arXiv:2511.16719, November 2025. doi: 10.48550/arXiv.2511.16719.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. *arXiv e-prints*, art. arXiv:2104.14294, April 2021. doi: 10.48550/arXiv.2104.14294.

[6] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *arXiv e-prints*, art. arXiv:2212.07143, December 2022. doi: 10.48550/arXiv.2212.07143.

[7] Mathieu Chevalley, Charlotte Bunne, Andreas Krause, and Stefan Bauer. Invariant Causal Mechanisms through Distribution Matching. *arXiv e-prints*, art. arXiv:2206.11646, June 2022. doi: 10.48550/arXiv.2206.11646.

[8] Jean-Baptiste Cordonnier, Aravindh Mahendran, Alexey Dosovitskiy, Dirk Weissenborn, Jakob Uszkoreit, and Thomas Unterthiner. Differentiable patch selection for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2351–2360, 2021.

[9] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=2dnO3LLiJ1`.

[10] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd Van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetic, Dustin Tran, Thomas Kipf, Mario Lucic, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors,

Proceedings of the 40th International Conference on Machine Learning, volume 202 of *Proceedings of Machine Learning Research*, pages 7480–7512. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/dehghani23a.html.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv e-prints*, art. arXiv:2010.11929, October 2020. doi: 10.48550/arXiv.2010.11929.

[13] Anthony Fuller, Yousef Yassin, Junfeng Wen, Daniel G. Kyrollos, Tarek Ibrahim, James R. Green, and Evan Shelhamer. Lookwhere? efficient visual recognition by learning where to look and what to see from self-supervision, 2025. URL https://arxiv.org/abs/2505.18051.

[14] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut Learning in Deep Neural Networks. *arXiv e-prints*, art. arXiv:2004.07780, April 2020. doi: 10.48550/arXiv.2004.07780.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[16] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning. *arXiv e-prints*, art. arXiv:2004.11362, April 2020. doi: 10.48550/arXiv.2004.11362.

[17] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

[18] Alexander Lappe and Martin Giese. Register and [cls] tokens induce a decoupling of local and global features in large vits. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 1009–1029. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/01bcbd34b02f3da9700a3ddd0480c156-Paper-Conference.pdf.

[19] Weixin Liang and James Zou. MetaShift: A Dataset of Datasets for Evaluating Contextual Distribution Shifts and Training Conflicts. *arXiv e-prints*, art. arXiv:2202.06523, February 2022. doi: 10.48550/arXiv.2202.06523.

[20] Evan Z Liu, Behzad Haghgoo, Annie S Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6781–6792. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/liu21f.html.

[21] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[22] Shenyu Lu, Zhaoying Pan, and Xiaoqian Wang. Think twice: Test-time reasoning for robust clip zero-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2919–2929, October 2025.

[23] Aengus Lynch, Gbètondji J-S Dovonon, Jean Kaddour, and Ricardo Silva. Spawrious: A benchmark for fine control of spurious correlation biases, 2023.

[24] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. *arXiv e-prints*, art. arXiv:2103.00020, February 2021. doi: 10.48550/arXiv.2103.00020.

[26] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. URL `https://arxiv.org/abs/2408.00714`.

[27] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization. *arXiv e-prints*, art. arXiv:1911.08731, November 2019. doi: 10.48550/arXiv.1911.08731.

[28] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[29] Baifeng Shi, Ziyang Wu, Maolin Mao, Xin Wang, and Trevor Darrell. When Do We Not Need Larger Vision Models? *arXiv e-prints*, art. arXiv:2403.13043, March 2024. doi: 10.48550/arXiv.2403.13043.

[30] Oriane Siméoni, Gilles Puy, Huy V. Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing Objects with Self-Supervised Transformers and no Labels. *arXiv e-prints*, art. arXiv:2109.14279, September 2021. doi: 10.48550/arXiv.2109.14279.

[31] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv e-prints*, art. arXiv:2508.10104, August 2025. doi: 10.48550/arXiv.2508.10104.

[32] Baochen Sun and Kate Saenko. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. *arXiv e-prints*, art. arXiv:1607.01719, July 2016. doi: 10.48550/arXiv.1607.01719.

[33] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era . In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 843–852, Los Alamitos, CA, USA, October 2017. IEEE Computer Society. doi: 10.1109/ICCV.2017.97. URL `https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.97`.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, art. arXiv:1706.03762, June 2017. doi: 10.48550/arXiv.1706.03762.

[35] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. TokenCut: Segmenting Objects in Images and Videos with Self-supervised Transformer and Normalized Cut. *arXiv e-prints*, art. arXiv:2209.00383, September 2022. doi: 10.48550/arXiv.2209.00383.

[36] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *arXiv e-prints*, art. arXiv:1502.03044, February 2015. doi: 10.48550/arXiv.1502.03044.

[37] Yu Yang, Besmira Nushi, Hamid Palangi, and Baharan Mirzasoleiman. Mitigating Spurious Correlations in Multi-modal Models during Fine-tuning. *arXiv e-prints*, art. arXiv:2304.03916, April 2023. doi: 10.48550/arXiv.2304.03916.

[38] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, June 2022.

[39] Michael Zhang, Nimit S. Sohoni, Hongyang R. Zhang, Chelsea Finn, and Christopher Ré. Correct-N-Contrast: A Contrastive Approach for Improving Robustness to Spurious Correlations. *arXiv e-prints*, art. arXiv:2203.01517, March 2022. doi: 10.48550/arXiv.2203.01517.

[40] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022.

A Attention Localization and Hit Rate across model families

Table 4: Attention localization vs. ground-truth bounding boxes across six pretraining families. Mass is the fraction of attention mass inside the ground truth bounding boxes; Hit is whether the $\arg\max$ of attention lies inside any ground truth box. Bold = best per family per column. Blue rows mark the smallest model under self-supervised pretraining, where the smallest ViT beats the largest on every column. Peach row marks the smallest model under contrastive image-text pretraining, where the trend breaks (smallest is not the best).

		Waterbirds		ImageNet val	
Model	**Params**	**Mass ↑**	**Hit ↑**	**Mass ↑**	**Hit ↑**
DINOv1 (self-distillation, original)					
ViT-S/16	21M	**0.689**	**0.939**	**0.681**	**0.849**
ViT-B/16	86M	0.649	0.934	0.659	0.829
DINOv2 (self-distillation)					
ViT-S	21M	**0.820**	**0.947**	**0.745**	**0.834**
ViT-B	86M	0.764	0.875	0.726	0.818
ViT-L	300M	0.650	0.823	0.678	0.767
ViT-G	1.1B	0.608	0.819	0.676	0.765
DINOv3 (self-distillation)					
ViT-S/16	21M	**0.767**	**0.913**	0.723	0.821
ViT-S+/16	29M	0.758	0.900	0.729	0.821
ViT-B/16	86M	0.735	0.877	**0.740**	**0.840**
ViT-L/16	300M	0.690	0.888	0.705	0.824
ViT-H+/16	840M	0.662	0.870	0.705	0.808
ViT-7B/16	6.7B	0.577	0.798	0.649	0.756
MAE (reconstruction)					
ViT-B/16	86M	**0.474**	**0.879**	**0.551**	**0.658**
ViT-L/16	307M	0.376	0.370	0.507	0.499
ViT-H/14	632M	0.377	0.451	0.503	0.554
iBOT (masked self-distillation)					
ViT-S/16	21M	**0.696**	0.909	**0.675**	**0.824**
ViT-B/16	85M	0.639	**0.949**	0.635	0.809
ViT-L/16	307M	0.557	0.902	0.583	0.750
OpenCLIP (contrastive image-text)					
ViT-B/16	86M	0.480	0.166	0.525	0.291
ViT-L/14	304M	**0.561**	**0.815**	**0.567**	0.543
ViT-H/14	632M	0.526	0.431	0.554	0.457
ViT-g/14	1.0B	0.522	0.585	0.556	**0.555**

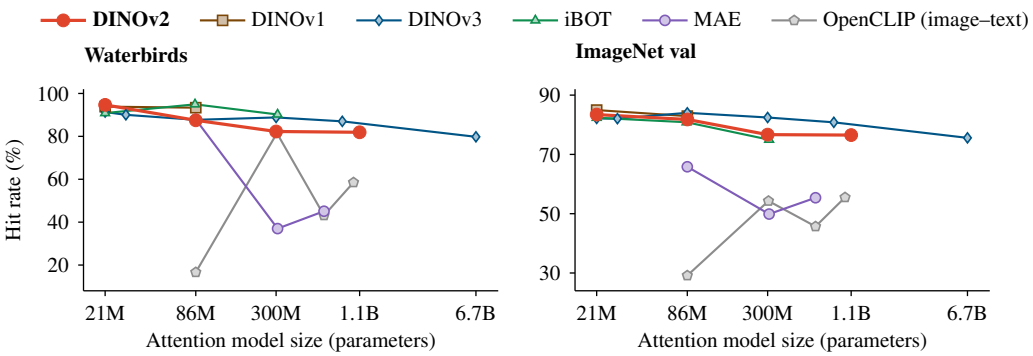

Figure 8: Attention hit rate vs. model size across six pretraining families, on Waterbirds (left) and ImageNet val (right). Hit rate is the fraction of images where the $\arg\max$ of the attention map falls inside the ground-truth bounding box. The DINOv2 and DINOv3 families show the cleanest inverse-scaling trend; MAE drops sharply from ViT-B/16 to ViT-L/16 on Waterbirds; OpenCLIP (gray) breaks the trend, with ViT-B/16 at 16.6% on Waterbirds (well below the next size). Companion to Figure 2; see Table 4 for precise values.

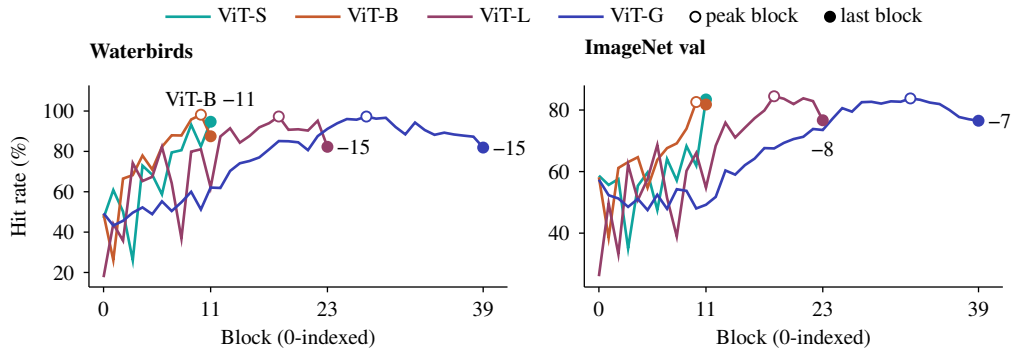

Figure 9: **Per-layer attention hit rate for DINOv2 with register tokens [9].** Hit-rate counterpart to Figure 7 in the main paper. Filled dots mark each model's last block and open dots its peak; numbers give the drop from peak to last block, in points (drops under 2 points are not labeled). Hit rate stays high across late layers for all sizes, reflecting that registers absorb the high-norm artifact tokens that would otherwise capture the CLS attention argmax.

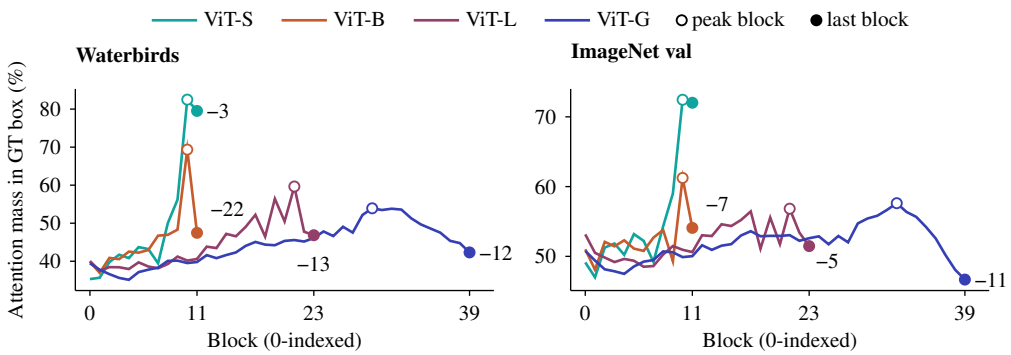

Figure 10: **Per-layer CLS-to-patches attention mass for DINOv2 *without* register tokens.** Without registers to absorb the late-layer high-norm "artifact" tokens identified by Darcet et al. [9], the inverse scaling trend in Figure 7 (with registers, main paper) becomes substantially sharper: the last-layer mass drop from ViT-S to ViT-G widens from 21 points to 37 points on Waterbirds and from 7 to 25 points on ImageNet val. ViT-S is largely unaffected (within 2.5 points of its registered counterpart at the last layer), while ViT-B/L/G's last-layer attention drifts off-object in late blocks. The per-layer hit-rate version (Figure 11) shows the effect even more starkly: for large ViTs without registers, the attention $\arg\max$ falls *outside* the GT bounding box on more than 80% of images at the final block. Registers do not eliminate the inverse scaling property we observe. Filled dots mark each model's last block and open dots its peak; numbers give the drop from peak to last block, in points.

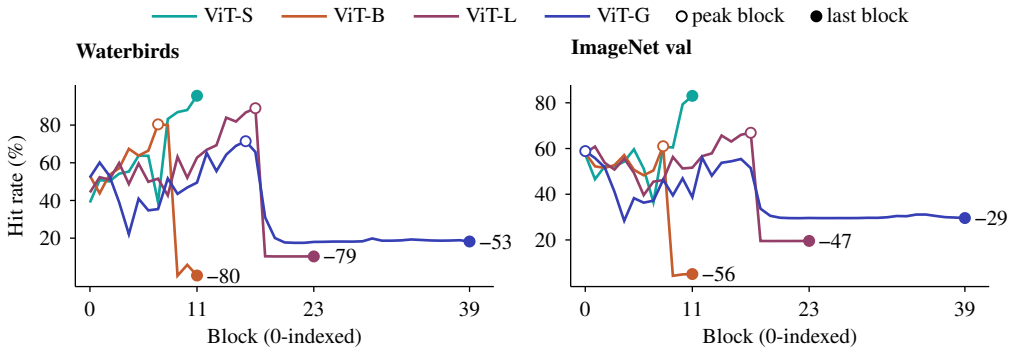

Figure 11: **Per-layer attention hit rate for DINOv2 *without* register tokens.** Hit-rate counterpart to Figure 10. Without registers, the hit rate of ViT-B/L/G collapses sharply partway through the network (ViT-B layer 9, ViT-L layer 18, ViT-G layer 18), consistent with attention being captured by high-norm background tokens in late blocks. ViT-S is unaffected and remains the most reliable localizer at the last block. Numbers give the drop from each model's peak (open dot) to its last block (filled dot), in points.

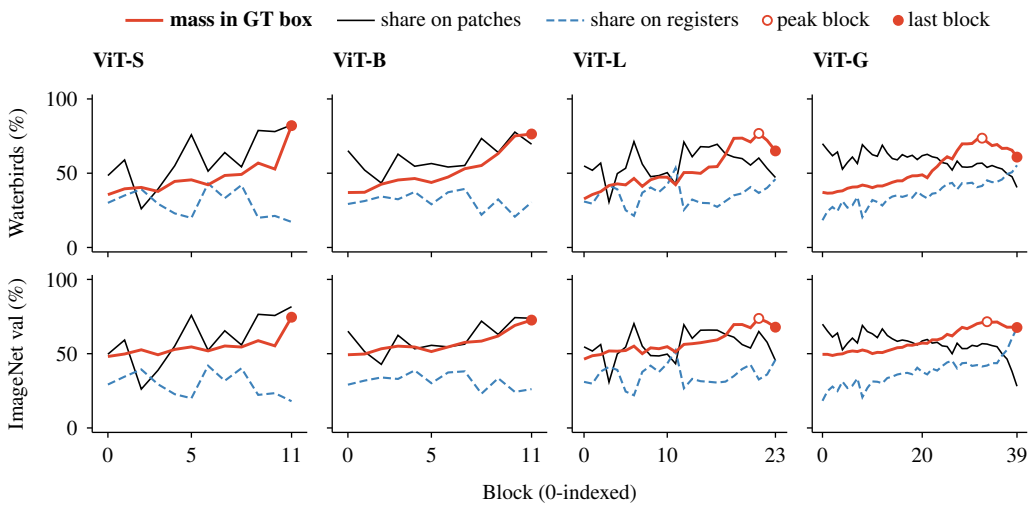

Figure 12: **The dip in patch-side localization coincides with attention shifting to register tokens.** For DINOv2 with registers on Waterbirds (top) and ImageNet val (bottom), we plot the per-layer fraction of CLS softmax mass on patch tokens (black) vs. register tokens (blue, dashed), alongside the patch-side mass-in-GT-bbox metric from Figure 7 (red). For ViT-L and ViT-G, register tokens absorb a growing share of CLS attention in late layers, eventually *exceeding* patch attention in ViT-G's final block (55% registers vs. 40% patches on Waterbirds); the patch-side localization metric drops correspondingly ($\sim$13 points from peak-localization layer to last block in ViT-G). ViT-S and ViT-B never experience this register takeover and continue to concentrate attention on patch tokens through their last block.

B Patch Size, Pretraining Objective, and Other Confounders

The impact of patch size on localization. We compare DINOv1 ViT-S and ViT-B at the /8 and /16 patch size.

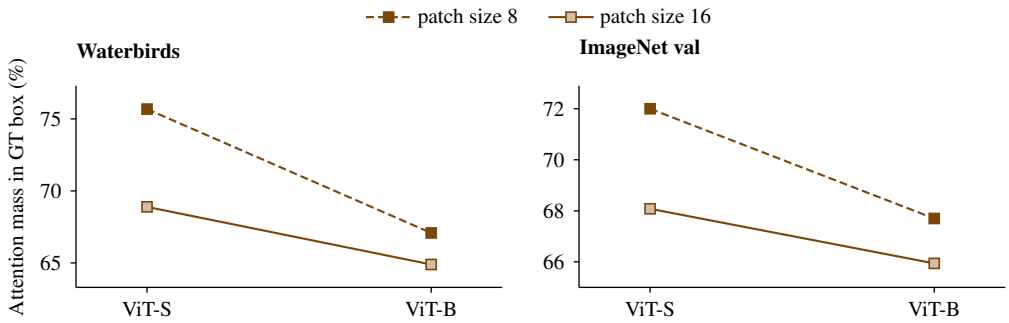

Figure 13: Finer patch sizes improve localization, but within each patch size family, we see inverse scaling with model size. Table 5 reports precise values.

Table 5: **DINOv1 patch-size vs. localization.**

	Waterbirds		ImageNet val	
DINOv1 model	**Mass ↑**	**Hit ↑**	**Mass ↑**	**Hit ↑**
ViT-S/16	0.689	0.939	0.681	0.849
ViT-S/8	0.757	0.978	0.720	0.868
ViT-B/16	0.649	0.934	0.659	0.829
ViT-B/8	0.671	0.950	0.677	0.842

Pretraining objective. We compare DeiT-S/16 (supervised) and DINOv1 ViT-S/16 (self-distilled), matching architecture, patch size, and ImageNet-1K pretraining corpus.

Table 6: **DINOv1 vs. DeiT localization.** Both models use ViT-S/16 and ImageNet-1K pretraining.

	Waterbirds		ImageNet val	
Model / pretraining	**Mass ↑**	**Hit ↑**	**Mass ↑**	**Hit ↑**
DeiT-S/16 (supervised)	0.394	0.157	0.442	0.301
DINOv1 ViT-S/16 (self-distilled)	**0.689**	**0.939**	**0.681**	**0.849**

Table 7: Fixed-patch-size comparisons across SSL families. Mass is the fraction of last-layer patch attention within ground-truth bounding boxes; Δ is larger minus smaller model mass. WB denotes Waterbirds and IN denotes ImageNet validation. MAE uses B/16 versus L/16 to hold patch size fixed, rather than the larger H/14 checkpoint.

Family	Corpus	Size pair	WB mass	Δ WB	Δ IN
DINOv1 (/16)	IN-1k	S $\to$ B	$0.689 \to 0.649$	-0.040	-0.021
DINOv2 (/14)	LVD-142M	S $\to$ g	$0.820 \to 0.608$	-0.212	-0.069
DINOv3 (/16)	LVD-1689M	S $\to$ 7B	$0.767 \to 0.577$	-0.190	-0.074
iBOT (/16)	IN-1k	S $\to$ L	$0.696 \to 0.557$	-0.139	-0.092
MAE (/16)	IN-1k	B $\to$ L	$0.474 \to 0.376$	-0.098	-0.044

Table 8: Waterbirds attention mass / hit rate for image–text pretrained ViTs. Parameter counts refer to the vision encoder.

Model family / corpus	B/16 86M	L/14 304M	H/14 632M	g/14 1.0B
OpenCLIP / LAION-2B	0.480 / 0.166	0.561 / 0.815	0.526 / 0.431	0.522 / 0.585
OpenAI CLIP / WIT-400M	0.426 / 0.166	0.483 / 0.172	—	—

Table 9: Reported downstream comparison with and without registers for the DINOv2 ViT-S selector / ViT-G embedder cross model result. Our method does not rely on registers. Waterbirds uses one 64×64 crop; Spawrious O2O Hard uses two crops of sizes 64×64 and 32×32. Cells are test accuracy / worst-group accuracy (%).

Register setting	Waterbirds	Spawrious O2O Hard
With registers	98.2 / 94.4	89.2 / 79.9
Without registers	98.2 / 94.4	91.7 / 82.1

C Attention Aggregation and Crop Selection

Head aggregation. The CLS token's attention in the last block of an H-head ViT yields per-head maps $a^h \in \mathbb{R}^N$ given the query q^h_{CLS} and key matrix $K^h \in \mathbb{R}^{N \times d_k}$ (with d_k the per-head embedding dimension):

$$a^h = \text{softmax}\left(\frac{q^h_{\text{CLS}} \cdot (K^h)^\intercal}{\sqrt{d_k}}\right) \in \mathbb{R}^N.$$

We aggregate by mean over heads, $a = \frac{1}{H}\sum_{h=1}^{H} a^h \in \mathbb{R}^N$, before bilinearly upsampling to the input resolution. Alternative aggregators over heads (entropy-weighted, lowest-entropy $k = 6$ mean, and max) are ablated in Appendix Tables 33-35, and none perform better than the mean over all heads. Max lowers Spawrious O2O downstream accuracy by 1.3 points.

Greedy non-overlapping crop selection. Given the upsampled $M \times M$ attention map a and a list of crop sizes $s_1, \ldots, s_n$ ordered largest-to-smallest, we sequentially place crops $c_1, \ldots, c_n$. Each crop $c_k = (x_k, y_k, s_k)$ is the location that maximizes the sum of attention mass within it, subject to non-overlap with previously placed crops. Letting $\mathrm{BoxSum}(a, x, y, x + s, y + s)$ denote the sum of attention values inside the crop starting at (x, y) with size s:

$$c_k = \underset{\substack{c=(x,y,s_k) \\ 0 \le x,y \le M - s_k \\ c \cap c_j = \varnothing,\ \forall j < k}}{\arg\max} \ \mathrm{BoxSum}(a, x, y, x + s_k, y + s_k), \qquad k = 1, \ldots, n.$$

For a new dataset, a user can start from these defaults (k=2, sizes [64, 32]). This configuration is used across all five datasets we tried.

D MetaShift Animals Dataset Card

MetaShift [19] is a dataset created from the Visual Genome in order to study learning under distribution shift. The dataset is grouped into subsets, where a subset is a co-occurrence of a particular class with another spurious feature. For example, an elephant with a person, an elephant with a bike, etc.

We created a subset of the MetaShift dataset focused on 8-class animal classification under distribution shift. Our dataset includes the classes `bear, bird, cow, elephant, giraffe, horse, sheep, zebra` in various contexts. For example, elephant train contexts include {`people, rope, wall, tree,...`}, while elephant test contexts include {`water, umbrella, camera, dirt,...`}. Within each class, the train and test set context tags are disjoint; we split them equally (the context count) between the two sets. We remove any image IDs present in the test set from the train set to enforce zero overlap between train and test (this deduplication step is why the train set ends up being far smaller than the test set).

Per-class train/test image counts are summarized in Table 10, and visual examples across context tags for four representative classes are shown in Figure 14. The dataset can be found at `https://github.com/SreehariRamMohan/MetaShift_Animals`.

Table 10: MetaShift Animals (8-class benchmark) class counts.

Class	**Train**	**Test**
bear	503	978
bird	1,122	977
cow	125	968
elephant	367	1,580
giraffe	238	2,368
horse	1,188	586
sheep	90	1,142
zebra	57	2,230
Total images	3,690	10,829
Class-context groups	73	77
Unique context tags	54	47
Unique image IDs	2,385	6,043

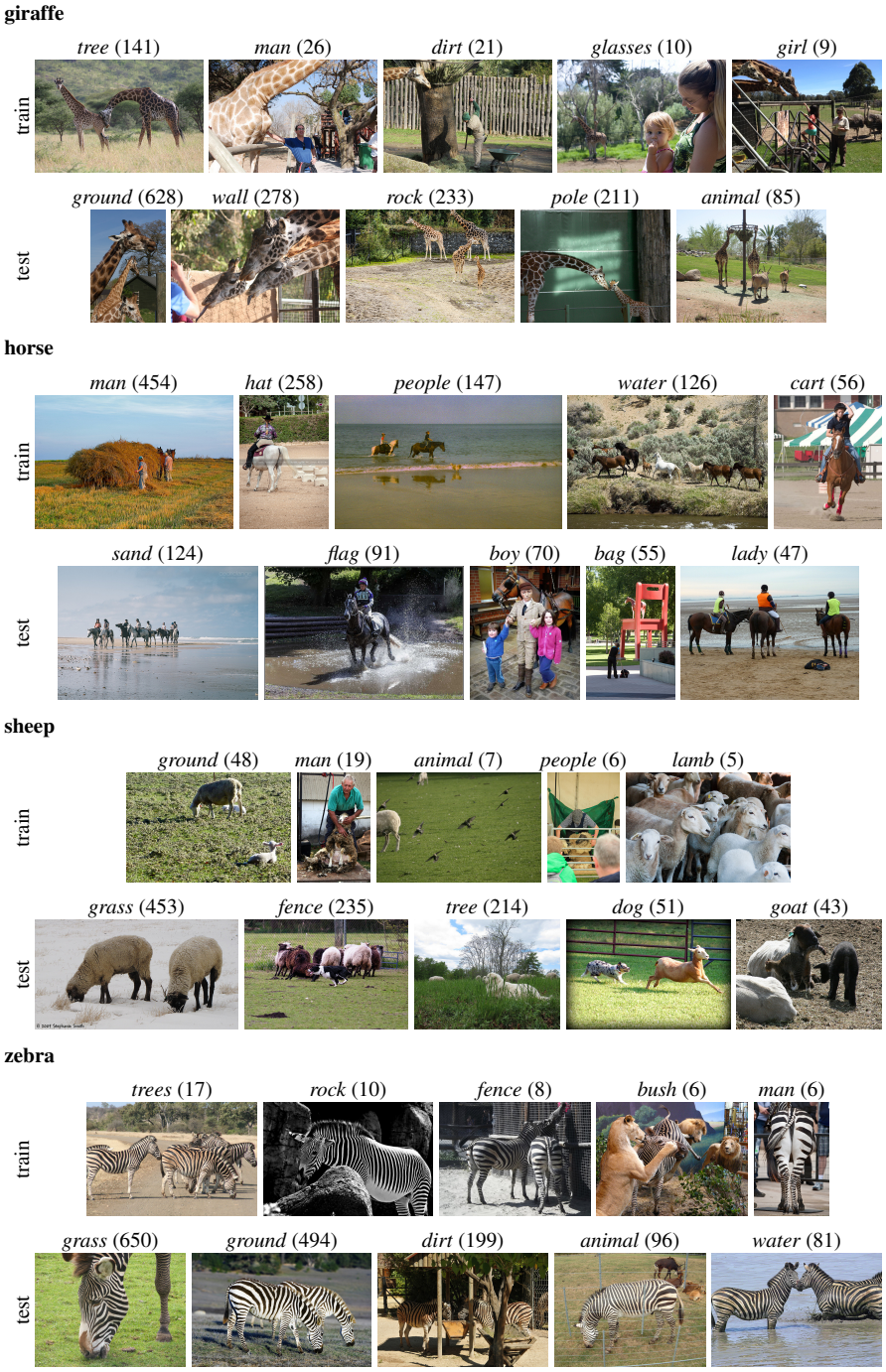

Figure 14: **MetaShift Animals: full context visualization.** Companion to Figure 4. Each panel shows one class (giraffe, horse, sheep, zebra) across multiple Visual Genome context tags. Within each panel, the top row contains training images and the bottom row contains test images; the per-image label is the context tag, which acts as the spurious feature. Train and test context tags are disjoint by construction.

E Crop Count Ablations, Sensitivity, Configurations, and Computational Cost

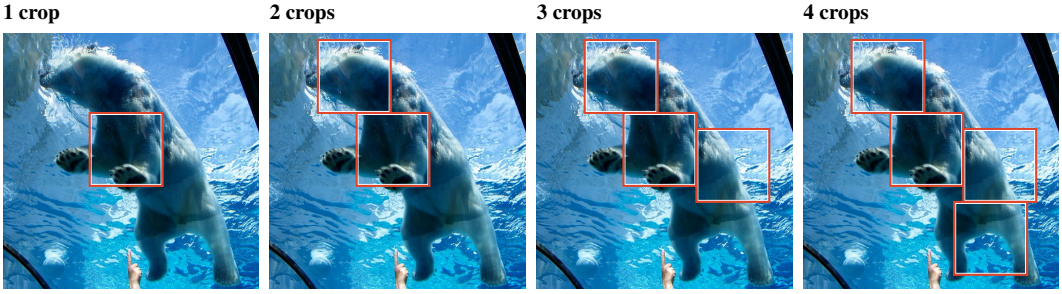

Figure 15: **Qualitative example of A^2 attention-guided crops.** One to four 64×64 crops selected by ViT-S attention on a MetaShift Animals bear. Crop regions are outlined in red. A^2 focuses on the bear and excludes the surrounding water and aquarium context.

We run with a fixed crop configuration of two crops, the first 64×64 and the second 32×32.

Table 11: Ablating the number of crops (64×64) using ViT-S for both attention and embedding. Coverage % is the proportion of the image seen by the classifier.

		Spawrious O2O	MetaShift Animals	
# Crops	**Coverage (%)**	**Test Acc (%)**	**Test Acc (%)**	**Worst-Class (%)**
1	8.2	**91.5**	83.1	61.8
2	16.3	88.3	**85.5**	67.0
3	24.5	87.0	84.7	66.2
4	32.7	86.0	85.0	**68.3**

E.1 Inference cost

We report FLOPs and inference latency on one NVIDIA A100 GPU using FP32 and batch size one. A^2 requires one selector forward pass followed by one embedding forward pass per crop. Tables 12–14 compare cross-model, zero-shot, and same-backbone configurations.

Table 12: **Cross-model inference cost.** DINOv2 ViT-S selects crops and DINOv2 ViT-G embeds them at 224×224. The $k = 2$ row corresponds to the shared main-paper crop count; $k = 1$ is the one-crop variant. Latencies include crop selection.

Configuration	GFLOPs	FLOPs ratio	Latency (ms)
ViT-G, full image (224^2)	304	1.00×	54
A^2, $k = 1$	311	1.02×	98
A^2, $k = 2$	615	2.02×	153
ViT-G, full image (518^2)	1790	5.9×	251

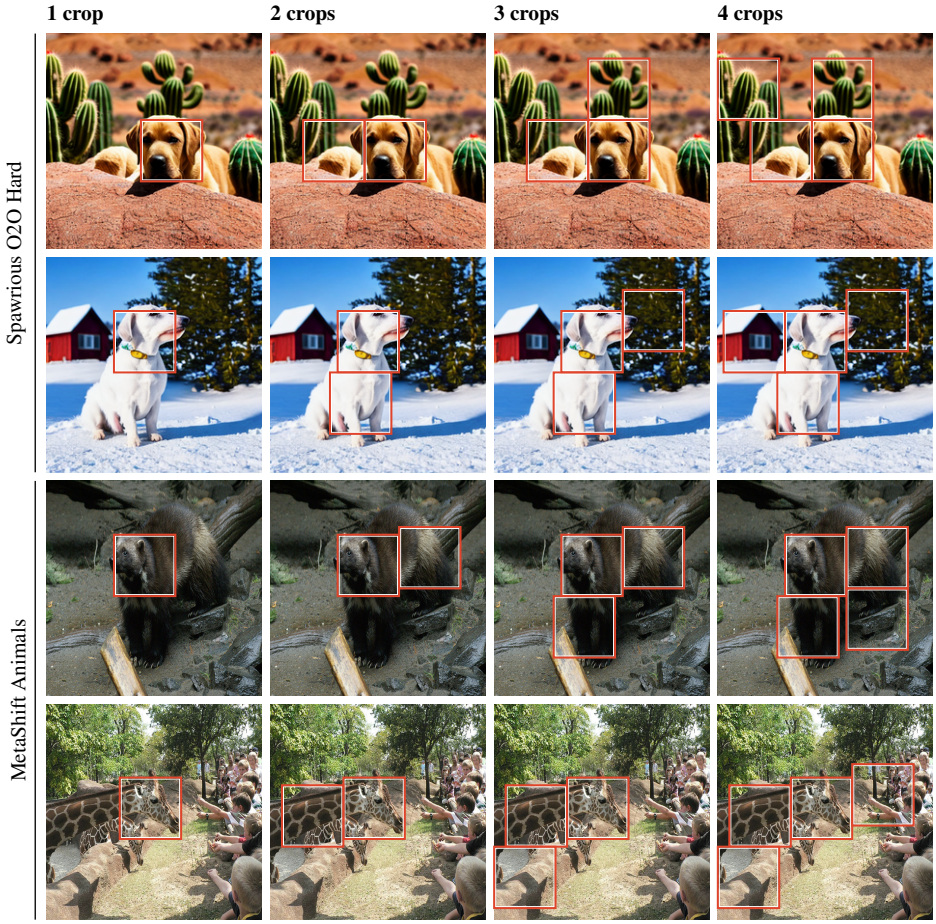

Figure 16: Crop count ablation visualization. Each column shows 1–4 attention-guided 64×64 crops selected by ViT-S. Rows 1–2: Spawrious O2O Hard, Rows 3–4: MetaShift Animals. Crop regions are outlined in red.

Table 13: **Zero-shot inference cost.** OpenCLIP ViT-L/14 embeds images or crops; A^2_{ZS} uses a ViT-S selector and one crop. TTR's reported latency covers two CLIP passes and excludes clustering, so its full end-to-end latency is not reported here.

Configuration	GFLOPs	FLOPs ratio	Latency (ms)
Full-image CLIP zero-shot	81	1.00×	16
A^2_{ZS}, $k = 1$	87	1.07×	60
TTR (two CLIP passes)	162	2.0×	31†

† Excludes clustering time.

Table 14: **Same-backbone inference cost.** DINOv2 ViT-S is used for both selection and embedding. S^2 uses five 224×224 views through `s2wrapper`. Latencies for A^2 include crop selection.

Configuration	GFLOPs	FLOPs ratio	Latency (ms)
ViT-S, full image (224^2)	6.2	1.0×	5
A^2, $k = 1$	12	2.0×	48
A^2, $k = 2$	19	3.0×	53
S^2 (five 224^2 views)	31	5.0×	12
ViT-S, full image (518^2)	47	7.5×	11

E.2 Crop Configuration Options and TopK

We ablate the design choices taken in A^2 for cropping. In particular, we test our decision to resize each crop to 224×224 rather than keeping its cropped size fed into the embedding model — also exploring other intermediate sizes like 112×112. This answers the question of whether classification performance benefits from having the small crop resized into a larger (blurry) image. The blurriness of the crop only marginally hurts performance, costing 0.5 WGA on Waterbirds. We note that on our other benchmarks, cropping from native resolution shows no consistent benefit over the existing paper pipeline. Upsampling the crop to 224^2 is helpful over processing in the crop native size.

We compare square crops to instead taking the patches with the highest attention and feeding them to the embedding model (with positional encodings preserved). We find that square cropping is an inductive bias that forces the model to focus its attention at a continuous unimodal, bimodal, or k-modal region(s) of the image. With top-k selection, the model runs the risk of selecting artifacts or disjoint regions of the image which can lead it astray (as it does in Spawrious M2M). Square crops are also nice because they do not require any special hacking to the embedder and can be used off the shelf with A^2_{ZS} (CLIP). The top-k option has the efficiency argument of avoiding multiple forward embedding passes.

Table 15: **Crop pixel source and embedder input resolution on Waterbirds.** DINOv2 ViT-S selects one crop and the frozen DINOv2 ViT-G embeds it. The crop width is 64 in the 224×224 attention frame. "Near-native" rounds each original-resolution crop dimension to the nearest multiple of 14 pixels for patch compatibility.

Crop pixel source	Embedder input	Tokens	Acc.	WGA
224×224 frame	224×224	256	98.2	94.4
224×224 frame	70×70	~ 25	97.3	92.5
224×224 frame	112×112	64	97.9	93.1
Original resolution	224×224	256	98.4	94.9
Original resolution	Near-native	~ 80	98.0	93.8

Table 16: **Crop size and resize-target sensitivity.** Worst-group accuracy (%) with a DINOv2 ViT-S selector and ViT-G embedder. All crop pixels come from the resized 224×224 frame. Waterbirds uses one crop of width s; Spawrious O2O Hard uses two crops of widths $[s, s/2]$. The patch-aligned condition maps $s = 32, 64, 128$ to input widths $28, 70, 126$, respectively.

Benchmark	Embedder input	$s = 32$	$s = 64$	$s = 128$
Waterbirds	112×112	87.5	93.1	94.2
	224×224	89.9	**94.4**	93.8
	Patch-aligned	54.4	92.5	93.9
Spawrious O2O Hard	112×112	76.6	78.4	68.9
	224×224	76.5	**80.0**	76.6
	Patch-aligned	64.8	74.8	64.7

Table 17: **Square crops versus top-K attention-ranked patch tokens.** The selector is DINOv2 ViT-S and the frozen embedder is DINOv2 ViT-G. The token budget is chosen to approximately match the image area covered by the crops before resizing. Square crops use one 64×64 crop for Waterbirds and two crops of widths $[64, 32]$ for the other benchmarks; each crop is resized to 224×224 before embedding.

		Square crops		Top-K tokens	
Benchmark	K	Acc.	WGA/WC	Acc.	WGA/WC
Waterbirds	21	98.2	94.4	97.7	94.9
Spawrious O2O Hard	26	89.2	79.9	86.7	79.7
Spawrious M2M Hard	26	86.0	64.6	82.3	50.9
MetaShift Cat vs. Dog	26	77.3	76.6	77.0	75.8
MetaShift Animals	26	88.9	70.8	89.1	70.1

F Full Results for Spawrious, Waterbirds, and MetaShift

F.1 Spawrious

Table 18: Spawrious Benchmark (O2O Hard + M2M Hard). We report the out of domain test accuracy. Note, by default the native resolution of all images is 224^2. In cases where we use a larger image, we have bilinearly interpolated the image to be bigger. A^2 uses two square crops of width [64, 32] for Spawrious.

Method	**Backbone**	**O2O-Hard**	**M2M-Hard**
Baselines using DINOv2 backbone			
Random Crops	ViT-S (21M)	38.2	22.1
Center Crop	ViT-S (21M)	80.2	68.4
Center Crop	ViT-G (1.1B)	72.6	61.4
Prior work (trained on Spawrious) [23]:			
ERM [23, 20]	ResNet-50	71.32	58.70
GroupDRO† [23, 20, 27]	ResNet-50	76.99	60.86
IRM†	ResNet-50	74.90	60.93
CORAL [23, 32]	ResNet-50	79.65	67.97
CausIRL [23, 7]	ResNet-50	80.40	68.93
DFR†	ViT-S (21M)	86.1 ± 0.4	73.4 ± 1.2
DFR†	ViT-B (86M)	90.1 ± 0.3	84.9 ± 1.2
DFR†	ViT-L (300M)	94.3 ± 0.2	86.1 ± 0.5
DFR†	ViT-G (1.1B)	95.7 ± 0.3	88.0 ± 0.5
DFR† + A^2	ViT-S (21M)	95.2 ± 0.1	86.2 ± 0.4
DINOv2 backbone			
Full only	ViT-S (21M)	80.5	66.6
Full only (518^2 image)	ViT-S (21M)	86.2	72.6
Full only	ViT-B (86M)	78.8	73.4
Full only (518^2 image)	ViT-B (86M)	82.5	82.4
Full only	ViT-L (300M)	80.8	81.9
Full only (518^2 image)	ViT-L (300M)	86.3	86.7
Full only	ViT-G (1.1B)	87.8	79.7
Attending on Attention (A^2)‡	ViT-S (21M)	91.0	82.4
Attending on Attention (A^2)‡	ViT-B (86M)	88.4	83.8
Attending on Attention (A^2)‡	ViT-L (300M)	82.1	83.5
Attending on Attention (A^2)‡	ViT-G (1.1B)	83.1	80.6
A^2 cross-model (Table 24)‡	ViT-S $\times$ ViT-L	89.7	87.1
Scaling on Scales [29] (scales=[1, 2] 224^2)	ViT-S (21M)	82.2	60.0
OpenCLIP backbone			
Full only	ViT-L-14	71.0	50.6
OpenCLIP ZS prompt (full)	ViT-L-14	89.8	92.4

† Requires group labels during training.

‡ Key rows were rerun across 5 seeds. Values are reported as means and show no variation (std $= 0.0$) since we use the lbfgs solver with sklearn Logistic Regression.

Table 19: Worst-group accuracy on the test set of Spawrious O2O Hard and M2M Hard. DFR rows are means $\pm$ standard deviation over 5 seeds (DFR's group-balanced retraining is stochastic). The A^2_{LR} cross-model row uses lbfgs Logistic Regression on the full training set and is therefore deterministic; we report the single fit. Groups are (breed, location) pairs; WGA is the worst-performing group on the test split.

Configuration	**Backbone**	**O2O-Hard WGA**	**M2M-Hard WGA**
DFR	ViT-S (21M)	74.9 ± 0.4	49.9 ± 1.7
DFR + A^2	ViT-S (21M)	88.8 ± 1.2	64.6 ± 0.8
DFR	ViT-B (86M)	84.5 ± 1.0	69.7 ± 1.5
DFR	ViT-L (300M)	89.1 ± 0.4	72.3 ± 0.8
DFR	ViT-G (1.1B)	89.9 ± 0.5	73.9 ± 0.7
A^2_{LR} cross-model	ViT-S $\times$ ViT-G	79.9	64.6

F.2 Waterbirds

Table 20: Waterbirds Benchmark. We report the out of domain test average accuracy and worst-group accuracy (%). Image size is 224^2 by default; rows tagged (518^2 image) use bilinear upsampling. A^2_{LR} (non cross model) uses one 64×64 attention-guided crop. A^2_{LR} cross model uses our shared 2 crop configuration of one 64×64 crop and one 32×32 crop. A^2_{ZS} uses ViT-S to select a 128×128 crop. Key rows (‡) averaged over 5 seeds.

Method	**Backbone**	**Avg Acc**	**Worst-Group**
Baselines using DINOv2 backbone			
Random Crops	ViT-S (21M)	70.7	32.4
Center Crop	ViT-S (21M)	90.9	78.7
Center Crop	ViT-G (1.1B)	98.0[§]	94.9
Prior work (trained on Waterbirds) [20]:			
ERM [23, 20]	ResNet-50	97.3	72.6
CVaR DRO	ResNet-50	96.0	75.9
LfF	ResNet-50	91.2	78.0
JTT [20]	ResNet-50	93.3	86.7
Group DRO[†] [23, 20, 27]	ResNet-50	93.5	91.4
DFR[†]	ResNet-50 [17]	94.2 ± 0.4	92.9 ± 0.2
DFR[†]	ViT-S (21M)	92.4 ± 0.4	90.4 ± 0.6
DFR[†]	ViT-B (86M)	95.5 ± 0.2	94.0 ± 0.5
DFR[†]	ViT-L (300M)	97.7 ± 0.1	97.1 ± 0.1
DFR[†]	ViT-G (1.1B)	98.1 ± 0.1	96.4 ± 0.1
DFR[†] + A^2	ViT-S (21M)	91.9 ± 0.5	91.0 ± 0.6
DINOv2 backbone			
Full only	ViT-S (21M)	88.7	72.7
Full only (518^2 image)	ViT-S (21M)	91.9	81.6
Full only	ViT-B (86M)	94.4	84.0
Full only (518^2 image)	ViT-B (86M)	96.3	89.9
Full only	ViT-L (300M)	96.9	89.3
Full only (518^2 image)	ViT-L (300M)	97.7	90.3
Full only	ViT-G (1.1B)	97.8	92.8
Attending on Attention (A^2)[‡]	ViT-S (21M)	93.0	80.4
Attending on Attention (A^2)[‡]	ViT-B (86M)	96.1	88.6
Attending on Attention (A^2)[‡]	ViT-L (300M)	93.8	85.2
Attending on Attention (A^2)[‡]	ViT-G (1.1B)	95.8	89.9
A^2 cross-model [‡]	ViT-S $\times$ ViT-G	98.3	95.3
Scaling on Scales [29] (scales=[1, 2] 224^2)	ViT-S (21M)	88.3	73.1
OpenCLIP backbone:			
Full only	ViT-L-14	87.5	69.2
OpenCLIP ZS prompt (full)	ViT-L-14	73.5	46.9

[†] Requires group labels during training.

[‡] Key rows were rerun across 5 seeds. Values are reported as means and show no variation (std $= 0.0$) since we use the lbfgs solver with sklearn Logistic Regression.

[§] See Section I for explanation as to why center crop does so well in Waterbirds (dataset center bias)

F.3 MetaShift Cat vs. Dog

Table 21: MetaShift Cat vs. Dog [19]. Test accuracy and worst-group accuracy (%) reported. Training has 100% spurious correlation (cat=indoor, dog=outdoor); test has flipped correlation. Image size is 224^2 by default; rows tagged (518^2 image) use bilinear upsampling. A^2_{LR} uses two attention-guided crops of width $[64, 32]$; A^2_{ZS} uses ViT-S to select a 128×128 crop. Key rows are reported over 5 seeds.

Method	Backbone	Test Acc	Worst-Group
Baselines using DINOv2 backbone			
Random Crops	ViT-S (21M)	44.0	40.6
Center Crop	ViT-S (21M)	64.0	61.3
Center Crop	ViT-G (1.1B)	76.9	74.8
DINOv2 backbone			
Full only	ViT-S (21M)	59.8	56.0
Full only (518^2 image)	ViT-S (21M)	65.2	60.9
Full only	ViT-B (86M)	68.5	68.2
Full only (518^2 image)	ViT-B (86M)	75.3	73.9
Full only	ViT-L (300M)	80.0	79.2
Full only (518^2 image)	ViT-L (300M)	83.2	82.4
Full only	ViT-G (1.1B)	78.6	76.8
DFR†	ViT-S (21M)	58.7 ± 2.0	54.5 ± 3.9
DFR† + A^2	ViT-S (21M)	69.4 ± 0.8	67.7 ± 1.5
DFR†	ViT-B (86M)	68.6 ± 3.4	65.9 ± 5.1
DFR†	ViT-L (300M)	73.6 ± 1.9	70.4 ± 3.5
DFR†	ViT-G (1.1B)	76.0 ± 2.0	74.2 ± 3.3
Attending on Attention (A^2)‡	ViT-S (21M)	69.8	68.3
Attending on Attention (A^2)‡	ViT-B (86M)	73.9	71.8
Attending on Attention (A^2)‡	ViT-L (300M)	72.5	70.6
Attending on Attention (A^2)‡	ViT-G (1.1B)	76.1	73.7
A^2 cross-model (Table 26)‡	ViT-S $\times$ ViT-G	77.3	76.6
Scaling on Scales [29] (scales=[1, 2] 224^2)	ViT-S (21M)	59.7	56.9
OpenCLIP ZS prompt (full)	ViT-L-14	92.4	88.8

† Requires group labels during training.

‡ Key rows were rerun across 5 seeds. Values are reported as means and show no variation (std $= 0.0$) since we use the lbfgs solver with sklearn Logistic Regression.

F.4 MetaShift Animals

Table 22: MetaShift Animals (8-class Domain Generalization). Test accuracy and worst-class accuracy (%) reported. Within each class, training and test contexts are disjoint. Image size is 224^2 by default; rows tagged (518^2 image) use bilinear upsampling. A^2_{LR} uses two attention-guided crops of width $[64, 32]$; A^2_{ZS} uses ViT-S to select a 128×128 crop. Key rows (‡) averaged over 5 seeds.

Method	Backbone	Test Acc	Worst-Class
Baselines using DINOv2 backbone			
Random Crops	ViT-S (21M)	58.3	34.6
Center Crop	ViT-S (21M)	82.6	62.7
Center Crop	ViT-G (1.1B)	89.2	72.4
DINOv2 backbone			
Full only	ViT-S (21M)	83.2	63.2
Full only (518^2 image)	ViT-S (21M)	87.5	68.3
Full only	ViT-B (86M)	88.3	70.3
Full only (518^2 image)	ViT-B (86M)	89.6	72.7
Full only	ViT-L (300M)	89.7	75.9
Full only (518^2 image)	ViT-L (300M)	90.1	77.6
Full only	ViT-G (1.1B)	90.6	75.9
DFR†	ViT-S (21M)	88.3 ± 0.1	72.0 ± 2.0
DFR† + A^2	ViT-S (21M)	88.9 ± 0.1	72.9 ± 1.3
DFR†	ViT-B (86M)	90.3 ± 0.3	73.5 ± 2.2
DFR†	ViT-L (300M)	91.4 ± 0.0	77.9 ± 0.8
DFR†	ViT-G (1.1B)	91.4 ± 0.3	76.6 ± 1.9
Attending on Attention (A^2)‡	ViT-S (21M)	85.1	67.0
Attending on Attention (A^2)‡	ViT-B (86M)	87.5	67.9
Attending on Attention (A^2)‡	ViT-L (300M)	85.2	69.1
Attending on Attention (A^2)‡	ViT-G (1.1B)	87.7	69.2
A^2 cross-model (Table 27)‡	ViT-S × ViT-G	88.9	70.8
Scaling on Scales [29] (scales=[1, 2] 224^2)	ViT-S (21M)	84.7	65.5
OpenCLIP backbone			
OpenCLIP ZS prompt (full)	ViT-L-14	90.4	58.8

† Requires group labels during training.

‡ Key rows were rerun across 5 seeds. Values are reported as means and show no variation (std = 0.0) since we use the lbfgs solver with sklearn Logistic Regression.

G Cross Model Matrices

Cross model matrices across all 5 datasets are shown below. In each matrix, **Bold** = best in column (best attention for each embedding) and *Italic* = diagonal entries which use the same model for embedding and attention computation. Each matrix reports that dataset's standard metric: test accuracy for Spawrious O2O/M2M, worst-group accuracy for Waterbirds and Cat vs. Dog, and worst-class accuracy for Animals.

G.1 Spawrious

Table 23: Spawrious O2O Hard: Cross-model attention × embedding matrix. Rows = attention source, columns = embedding model. Values are test accuracy.

Attn \ Emb	ViT-S	ViT-B	ViT-L	ViT-G
ViT-S attn	***91.0***	**89.5**	**89.7**	**89.2**
ViT-B attn	89.6	*88.4*	88.2	87.9
ViT-L attn	85.2	82.7	*82.1*	84.6
ViT-G attn	86.3	84.3	84.6	*83.1*

Table 24: Spawrious M2M Hard: Cross-model attention × embedding matrix. **ViT-S attn → ViT-L emb (87.1) is the overall best**, beating all diagonal entries. Values are test accuracy.

Attn \ Emb	ViT-S	ViT-B	ViT-L	ViT-G
ViT-S attn	***82.4***	**84.6**	**87.1**	**86.0**
ViT-B attn	81.0	*83.8*	85.3	84.8
ViT-L attn	76.5	80.2	*83.5*	82.3
ViT-G attn	74.7	79.9	81.2	*80.6*

G.2 Waterbirds

Table 25: Waterbirds: Cross-model attention × embedding matrix. **ViT-S attn → ViT-G emb (94.4 WGA)** beats all diagonal entries including ViT-G → ViT-G (89.9). Values are worst-group accuracy. All runs in this matrix use a single 64×64 crop.

Attn \ Emb	ViT-S	ViT-B	ViT-L	ViT-G
ViT-S attn	***80.4***	**90.5**	**93.1**	**94.4**
ViT-B attn	77.9	*88.6*	91.0	93.5
ViT-L attn	67.9	75.5	*85.2*	86.0
ViT-G attn	71.7	81.6	86.4	*89.9*

G.3 MetaShift Cat vs. Dog

Table 26: MetaShift Cat vs. Dog: Cross-model attention × embedding matrix. ViT-S attention is best for 3/4 embedding columns. Values are worst-group accuracy.

Attn \ Emb	ViT-S	ViT-B	ViT-L	ViT-G
ViT-S attn	*68.3*	**76.2**	**75.5**	**76.6**
ViT-B attn	**69.3**	*71.8*	74.0	75.8
ViT-L attn	64.4	70.8	*70.6*	73.6
ViT-G attn	66.2	70.6	73.2	*73.7*

G.4 MetaShift Animals

Table 27: MetaShift Animals: Cross-model attention × embedding matrix. ViT-S attention is best for $3/4$ embedding columns. Values are worst-class accuracy.

Attn \ Emb	ViT-S	ViT-B	ViT-L	ViT-G
ViT-S attn	***67.0***	**68.2**	70.3	**70.8**
ViT-B attn	64.6	*67.9*	69.6	69.0
ViT-L attn	62.1	65.8	*69.1*	70.0
ViT-G attn	64.0	64.8	**70.4**	*69.2*

H Adapting Attention

A^2 assumes the attention map naturally concentrates attention on the task-relevant features of the image. One prominent counter-example is CelebA [21], where the task is to predict whether a person has blond hair or not given a photo of their face, with the spurious feature being gender (since blond men are a rarity). The default attention maps will focus attention on the face or lower body (Figure 17 shows this). To address this limitation, we initialize a small MLP adapter network ($3{,}000$ params) whose input is the latent activations of the ViT, and whose output is added to the attention map from the ViT. The MLP is trained end-to-end, by backpropagating through the logistic regression fit and patch selection process (using DPS [8]). This proof-of-concept approach allows us to refocus attention to the hair in CelebA and improves worst-group accuracy (blond men) from 23.33% to

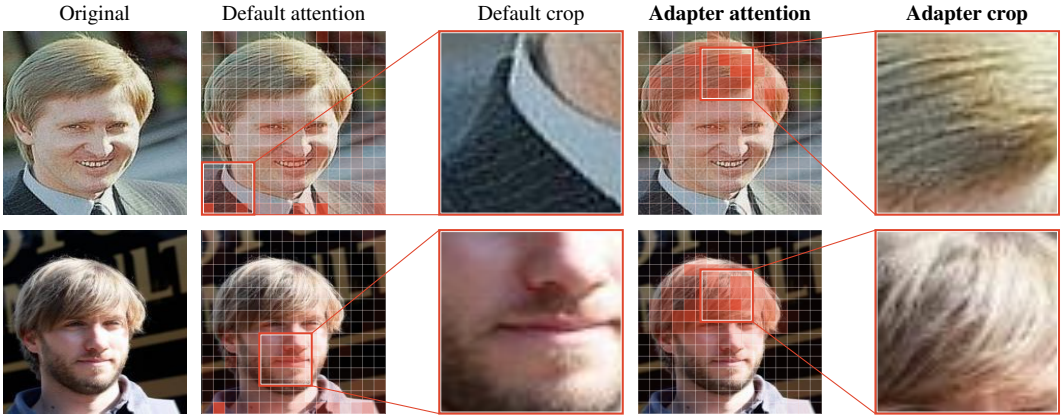

Figure 17: **Visualizing the MLP Adapter.** The MLP Adapter changes the attention map for the blond male class in the CelebA dataset. Default attention focuses on the face or lower body; adapted attention focuses on the hair, which is the label-defining feature.

66.11%, outperforming a baseline logistic regression fit on ViT-S embeddings of the full image by 35.01% (baseline gets 31.1% WGA). See Appendix Tables 28, 29 for full results.

The zero-shot method assumes that the features the attention map focuses on are the ones relevant to the task; this assumption holds for the 5 benchmarks we test, but it may not always be the case. In this scenario, we can learn a simple MLP adapter network ($\approx 3{,}000$ parameters), which operates on the patch features from the ViT to produce an augmented $\tilde{A}$ which allows us to $\text{Crop}_i(x;\ \tilde{A})$, embed with $\mathcal{T}(c_i)$, and fit a classifier using crops from a better attention map. Figure 18 illustrates the full pipeline.

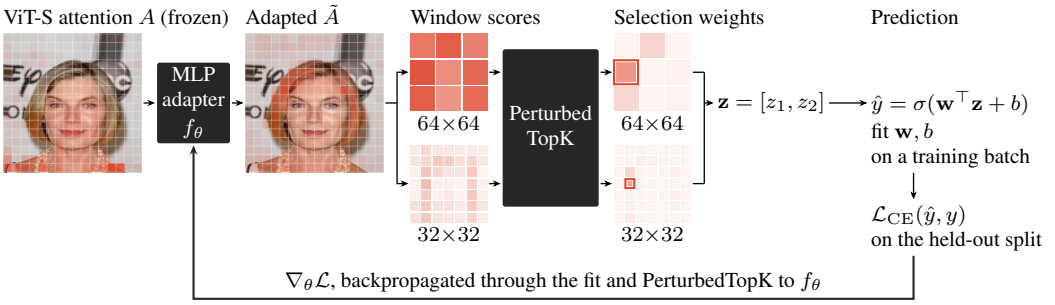

Figure 18: Learning the MLP attention adapter by differentiating through classifier fit using differentiable patch selection. The adapter learns to focus on the hair.

We use differentiable patch selection (DPS) [8] with a fixed set of 58 possible crops (9 64×64 and 49 32×32 crops in the 224×224 image) to get k weighted crops $c_1(x), \ldots, c_k(x)$ from the attention map, $\tilde{A}$. We compute the embedding z by concatenating the embedded crops using the embedding model, $\mathcal{T}$:

$$z(x) = [\,\mathcal{T}(c_1(x)),\ \ldots,\ \mathcal{T}(c_k(x))\,] \in \mathbb{R}^{k \cdot d}$$

We then fit a logistic regression classifier by taking a few gradient steps on w, b, the classifier weights using cross entropy loss ℓ.

$$(w^{t+1}, b^{t+1}) = (w^t, b^t) - \eta \nabla_{w,b} \ell\left(z(x)w^t + b^t, y\right), \quad t = 0, \ldots, T-1.$$

We evaluate the classifier on a held out partition of the training set, compute the cross entropy loss, and backpropagate through the classifier fitting and patch selection to optimize the MLP adapter weights.

Table 28: MLP Adapter Evaluation on CelebA Dataset. Adapter and default attention results are reported as mean ± std over 5 seeds.

Configuration	Test	Worst-Group Acc.
A^2 with MLP Adapter (384-dim)	$91.07 \pm 0.16\%$	**66.11** $\pm$ **1.57**%
A^2 default attention (384-dim)	$78.41 \pm 0.19\%$	$23.33 \pm 0.93\%$
ViT-S classifier (full image 224×224)	**94.2**%	31.1%

Table 29: Per-Group Accuracy Breakdown on CelebA Test Set. The MLP adapter increases performance on the minority class (blond men). Adapter and default attention results are reported as mean ± std over 5 seeds.

Group	MLP Adapter	Default Attn	ViT-S Classifier
Non-blond female (n=9767)	$86.96 \pm 0.37\%$	$67.52 \pm 0.37\%$	95.1%
Non-blond male (n=7535)	$96.01 \pm 0.18\%$	$92.90 \pm 0.25\%$	99.4%
Blond female (n=2480)	$94.12 \pm 0.38\%$	$81.31 \pm 0.39\%$	78.8%
Blond male (n=180)	$66.11 \pm 1.57\%$	$23.33 \pm 0.93\%$	31.1%

I On center crop as a strong baseline for Waterbirds (Table 20)

Simply center cropping the image to 128×128, embedding the crop, and fitting a logistic regression classifier seems like a competitive baseline. This is because Waterbirds is heavily biased toward placing the main subject (bird) at the center of the image. To illustrate this point, we sorted the test set birds by distance to the image center, and showed the performance of a center crop classifier on various percentiles (of bird distance to center). The results (Table 30) show a clear trend that as the main subject drifts away from the center, the A^2 fitted classifier outperforms the center crop classifier. In the 58 test set images with the bird furthest away from the image centroid, A^2 outperforms center crop by 6.9 points, while for the 58 test set images closest to the image centroid, A^2 actually underperforms the center crop classifier by 1.7 points (the bird is exactly centered here so center cropping is the most optimal crop to take). For reference, we also include iFAM K=8 results in the same table; iFAM relies on the center bias as well, and performance drops when going from subset bottom1 to top1 ($100.0\% \rightarrow 81.8\%$), consistent with iFAM exploiting Waterbirds' center bias more than A^2 does. We show some images from the top 1% and bottom 1% in Figure 19.



Figure 19: Examples from the bottom 1% (left three) subset and top 1% (right three) subsets. Red boxes show the center crop. Notice how in the top 1%, the center crop completely misses the bird, which explains the large drop off in Table 30.

Table 30: Distance stratification on Waterbirds test set. Center crop uses a fixed 128×128 center crop with a ViT-S logistic regression classifier; A^2 uses one 64×64 attention-guided crop with a ViT-S logistic regression classifier; iFAM K=8 is trained on the full Waterbirds train set (see Appendix Q). Models are trained once on the full Waterbirds training set and evaluated on each test-set subset. Test set is sorted from the bird being closest to the image center ("bottom") to furthest from the image center ("top"); the "top1" subset is the 58 test birds furthest from the image centroid. WGA results for the top and bottom 1st and 5th percentiles may have higher noise due to minority groups having < 20 samples in those subsets, but are included for reference.

Subset	N	Center Crop Acc	Center Crop WGA	A^2 Acc	A^2 WGA	iFAM $K{=}8$ Acc	iFAM $K{=}8$ WGA
top1	58	84.5	63.6	91.4	72.7	98.3	81.8
top5	290	89.0	75.6	93.8	87.8	98.3	92.7
top10	580	89.8	78.4	93.1	84.6	98.4	92.3
bottom10	580	91.6	84.5	92.2	78.0	98.6	96.6
bottom5	290	92.4	85.8	92.4	74.1	99.7	100.0
bottom1	58	93.1	86.7	91.4	83.3	100.0	100.0
full	5794	90.9	78.7	93.0	80.4	98.8	96.1

J DFR Baseline Implementation Details

All DFR rows run only Stage 2 of [17], last-layer retraining on a group-balanced subset, applied to frozen pretrained DINOv2 features. We do not fine-tune the backbone (Stage 1 ERM) on the biased train set; this is the natural baseline because A^2 also operates on frozen DINOv2 features, so comparing against balanced last-layer retraining on the same backbone isolates the effect of group balancing versus attention-guided cropping. We base our code on the public DFR repo `https://github.com/izmailovpavel/spurious_feature_learning/blob/main/dfr_evaluate_spurious.py` (modified to use frozen DINOv2 embeddings) and fit L1-regularized logistic regression (sklearn `liblinear`) on standardized features, grid-searching $C \in \{1.0, 0.7, 0.3, 0.1, 0.07, 0.03, 0.01\}$ and (for binary tasks) per-class weights to maximize worst-group accuracy on a tuning split, then averaging weights over 20 retrains on independently subsampled group-balanced sets. **Waterbirds** uses DFR(Val) (official val split 50/50 for tuning vs. retraining). **Spawrious O2O/M2M Hard**, **MetaShift Cat vs. Dog**, and **MetaShift Animals** have no provided val, so we use DFR(Train): 20% of train held out as pseudo-val for hyperparameter tuning, group-balanced retraining on the remaining 80%.

K Attention Concentration and Different Head Aggregators

Table 31: Attention concentration across DINOv2 model sizes (ImageNet validation, 50k images). Interpretation, 3rd column, first row: 48% of attention goes to 10% of patches.

Model	Entropy (norm)	Top-10%	Top-25%
ViT-S (21M)	$\mathbf{0.9219 \pm 0.0382}$	$\mathbf{0.4814 \pm 0.1504}$	$\mathbf{0.7430 \pm 0.1259}$
ViT-B (86M)	0.9292 ± 0.0316	0.4653 ± 0.1305	0.7207 ± 0.1098
ViT-L (300M)	0.9343 ± 0.0271	0.4528 ± 0.1090	0.7052 ± 0.0991
ViT-G (1.1B)	0.9351 ± 0.0226	0.4522 ± 0.0941	0.7043 ± 0.0840

Table 32: Attention concentration on Waterbirds validation set.

Model	Entropy (norm)	Top-10%	Top-25%
ViT-S (21M)	$\mathbf{0.8763 \pm 0.0303}$	$\mathbf{0.6742 \pm 0.1173}$	$\mathbf{0.8742 \pm 0.0612}$
ViT-B (86M)	0.8980 ± 0.0245	0.6089 ± 0.1049	0.8246 ± 0.0606
ViT-L (300M)	0.9187 ± 0.0271	0.5195 ± 0.1119	0.7636 ± 0.0850
ViT-G (1.1B)	0.9307 ± 0.0196	0.4738 ± 0.0842	0.7234 ± 0.0688

Table 33: Different attention head aggregation methods on Spawrious O2O Hard (cross-model ViT-G attention and ViT-S embeddings). We report A^2 crops-only OOD test accuracy, where crops are chosen using the attention map resulting when using the following aggregation method (on the last-layer self-attention across all 24 ViT-G heads).

Aggregation Method	A^2 **Test OOD (%)**
Mean	**86.3**
Entropy-weighted	**86.3**
Lowest-k mean (k=6)	85.7
Max	85.0

Table 34: Attention concentration on Spawrious O2O Hard test split for ViT-G using different head aggregation methods

Aggregation Method	**Entropy (norm)**	**Top-10%**	**Top-25%**
Mean	0.9202 ± 0.0210	0.5029 ± 0.0914	0.7698 ± 0.0717
Entropy-weighted	0.9170 ± 0.0224	0.5142 ± 0.0950	0.7780 ± 0.0728
Lowest-entropy $k = 6$ mean	$\mathbf{0.8561 \pm 0.0338}$	$\mathbf{0.7185 \pm 0.1056}$	$\mathbf{0.9111 \pm 0.0513}$
Max	0.9000 ± 0.0268	0.5817 ± 0.0969	0.8117 ± 0.0680

Table 35: Attention concentration on ImageNet (50k validation split) for ViT-G using different head aggregation methods

Aggregation Method	**Entropy (norm)**	**Top-10%**	**Top-25%**
Mean	0.9351 ± 0.0226	0.4522 ± 0.0941	0.7043 ± 0.0840
Entropy-weighted	0.9325 ± 0.0241	0.4619 ± 0.0983	0.7117 ± 0.0857
Lowest-entropy $k = 6$ mean	$\mathbf{0.8731 \pm 0.0411}$	$\mathbf{0.6624 \pm 0.1271}$	$\mathbf{0.8654 \pm 0.0764}$
Max	0.9218 ± 0.0273	0.5050 ± 0.0998	0.7383 ± 0.0794

L Adapter Learning Curves

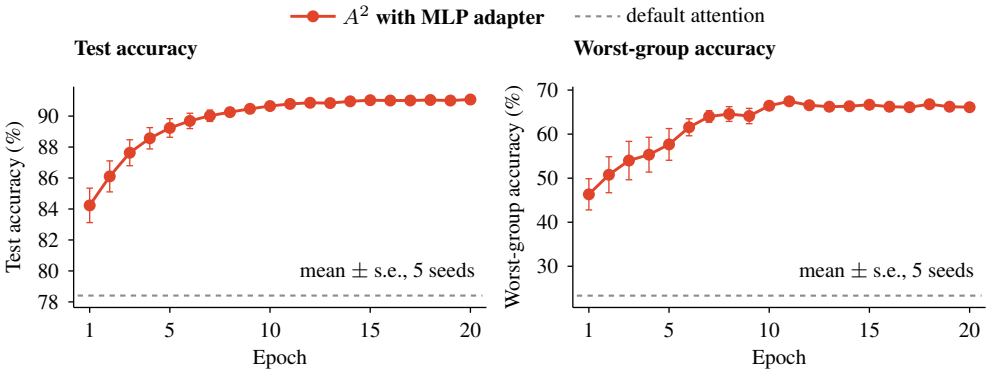

Figure 20: MLP Adapter Learning curves on CelebA. Results averaged across 5 seeds.

M DINOv3 Attention Scaling Graphs and Results

DINOv3 Attention Scaling. The Attention model is varied from DINOv3 ViT-S/16 (21M) to DINOv3 ViT-7B/16 (6.7B) with the embedding model fixed to DINOv2 ViT-G (1.1B). We show the general trend of decreasing A^2 performance (Test Acc and WGA) as the attention model size grows also holds for DINOv3.

Table 36: Mean A^2_{LR} performance across all evaluated datasets, ordered from smallest to largest DINOv3 attention model.

Attention model	Params	Test Acc (%)	WGA (%)
ViT-S/16	21M	86.76	76.16
ViT-S+/16	29M	86.38	75.40
ViT-B/16	86M	86.66	75.52
ViT-L/16	300M	84.98	72.12
ViT-H+/16	840M	84.38	72.44
ViT-7B/16	6,716M	84.76	73.38

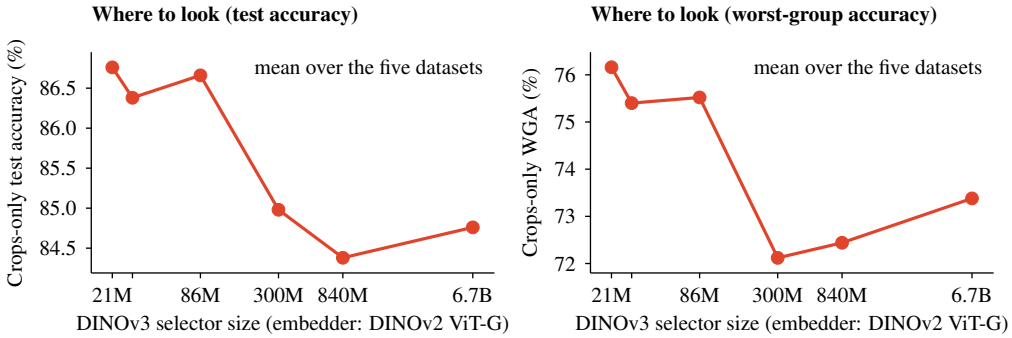

Figure 21: DINOv3 where-to-look scaling (mean across datasets) with fixed DINOv2 ViT-G embeddings. Absolute values are reported in Table 36.

N Additional DINOv3 Where-to-Look Scaling Results

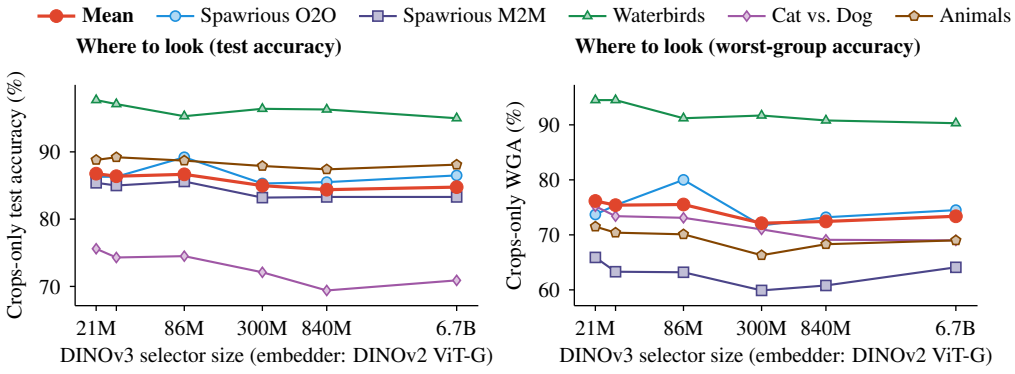

Figure 22: DINOv3 where-to-look scaling per dataset (colored lines) with the mean trend (red), using fixed DINOv2 ViT-G embeddings. Per-dataset absolute values in Tables 37, 38, 39, 40, and 41.

N.1 Per-dataset DINOv3 Where-to-Look Results

Table 37: MetaShift Animals (A^2_{LR}) with fixed DINOv2 ViT-G embeddings.

Attention model	Params	Test Acc (%)	WC (%)
ViT-S/16	21M	88.8	71.5
ViT-S+/16	29M	89.2	70.4
ViT-B/16	86M	88.7	70.1
ViT-L/16	300M	87.9	66.3
ViT-H+/16	840M	87.4	68.3
ViT-7B/16	6,716M	88.1	69.0

Table 38: MetaShift Cat vs. Dog (A^2_{LR}) with fixed DINOv2 ViT-G embeddings.

Attention model	Params	Test Acc (%)	WGA (%)
ViT-S/16	21M	75.6	75.2
ViT-S+/16	29M	74.3	73.4
ViT-B/16	86M	74.5	73.1
ViT-L/16	300M	72.1	71.0
ViT-H+/16	840M	69.4	69.1
ViT-7B/16	6,716M	70.9	69.0

Table 39: Spawrious O2O Hard (A^2_{LR}) with fixed DINOv2 ViT-G embeddings.

Attention model	Params	Test Acc (%)	WGA (%)
ViT-S/16	21M	86.3	73.7
ViT-S+/16	29M	86.3	75.4
ViT-B/16	86M	89.2	80.0
ViT-L/16	300M	85.3	71.7
ViT-H+/16	840M	85.5	73.2
ViT-7B/16	6,716M	86.5	74.5

Table 40: Spawrious M2M Hard (A^2_{LR}) with fixed DINOv2 ViT-G embeddings.

Attention model	Params	Test Acc (%)	WGA (%)
ViT-S/16	21M	85.4	65.9
ViT-S+/16	29M	85.0	63.3
ViT-B/16	86M	85.6	63.2
ViT-L/16	300M	83.2	59.9
ViT-H+/16	840M	83.3	60.8
ViT-7B/16	6,716M	83.3	64.1

Table 41: Waterbirds (A^2_{LR}) with fixed DINOv2 ViT-G embeddings.

Attention model	Params	Test Acc (%)	WGA (%)
ViT-S/16	21M	97.7	94.5
ViT-S+/16	29M	97.1	94.5
ViT-B/16	86M	95.3	91.2
ViT-L/16	300M	96.4	91.7
ViT-H+/16	840M	96.3	90.8
ViT-7B/16	6,716M	95.0	90.3

O OpenCLIP ZS Family Scaling: Full Results

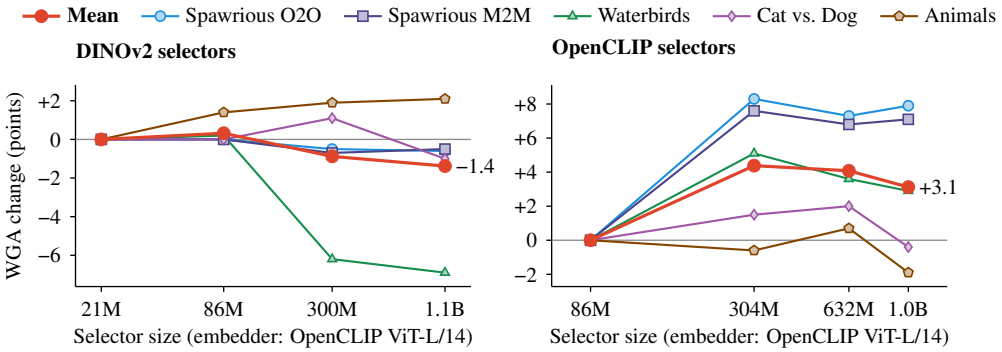

Figure 23: Worst-group accuracy version of Figure 6, also plotted relative to the smallest model in each family. The same family contrast holds: DINOv2 attention shows a mild downward WGA trend with size, while OpenCLIP attention shows large positive offsets at ViT-L/14 and larger (Section 2). Absolute per-dataset values are reported in Table 42.

Table 42: OpenCLIP ViT-L-14 zero-shot classification on A^2 attention-guided 128×128 crops. The rows represent the model used to compute attention (and select crops). The backbone used to embed crops and class text labels is fixed to OpenCLIP ViT-L-14. DINOv2 attention shows inverse scaling (smaller is better); OpenCLIP attention is non-monotonic (ViT-B-16 underperforms despite being the smallest), this finding mirrors the localization finding from Section 2. For OpenCLIP, we report the number of parameters in the vision encoder.

		O2O-Hard		**M2M-Hard**		**Waterbirds**		**Cat vs. Dog**		**Animals**	
Attn Family	**Attn Model**	Acc	WGA	Acc	WGA	Acc	WGA	Acc	WGA	Acc	WC
Full image (no crops)		89.8	82.0	92.4	82.1	73.5	46.9	92.4	88.8	90.4	58.8
DINOv2	ViT-S(21M)	**92.8**	**85.0**	**94.0**	**85.4**	**80.3**	64.5	89.7	88.8	**89.7**	54.7
	ViT-B(86M)	92.5	**85.0**	**94.0**	**85.4**	80.7	**64.7**	**90.3**	88.8	89.6	56.1
	ViT-L(300M)	91.0	84.5	93.4	84.7	77.8	58.3	90.1	**89.9**	89.0	56.6
	ViT-G(1.1B)	91.5	84.4	93.5	84.9	77.0	57.6	90.1	87.8	89.5	56.8
OpenCLIP	ViT-B-16(86M)	84.0	75.9	87.2	76.8	77.1	58.4	87.7	87.5	87.5	59.4
	ViT-L-14(304M)	91.9	84.2	93.4	84.4	79.6	63.5	89.0	89.0	89.3	58.8
	ViT-H-14(632M)	91.2	83.2	92.9	83.6	78.9	62.0	89.5	89.5	88.6	**60.1**
	ViT-G-14(1.01B)	91.4	83.8	93.4	83.9	78.6	61.3	89.5	87.1	88.8	57.5

P TTR Implementation Details

We reimplement TTR [22] based on the author's official codebase `https://github.com/ShengyuLu/TTR`. We use the OpenCLIP ViT-L-14 backbone (`laion2b_s32b_b82k`) for TTR and A^2 to ensure fair comparison. For Waterbirds, we were able to reproduce the reported results (achieving slightly higher WGA than was originally reported). We were unable to match the MetaShift Cat vs. Dog result, and report our best numbers instead.

Table 43: TTR hyperparameters used in our reimplementation

	Waterbirds	Spawrious O2O/M2M	MetaShift Cat vs. Dog	MetaShift Animals
Queue length	28	28	24	24
Gaussian σ	0.72	0.72	0.71	0.71
KMeans K	3	3	3	3
PCA components	3	3	3	3
SVD rank (q)	16	16	16	16

Below we also include our class prompts and general prompts (to isolate the main subject in each image from the clusters)

Table 44: TTR text prompts used in our reimplementation.

Dataset	Class prompts	General label
Waterbirds	"a photo of a landbird.", "a photo of a waterbird."	"a photo of a bird."
Spawrious O2O/M2M	"a photo of a {breed}." for each breed	"a photo of a dog."
MetaShift Cat vs. Dog	"a photo of a cat.", "a photo of a dog."	"a photo of a pet."
MetaShift Animals	"a photo of a {class}." for each class	"a photo of an animal."

Q iFAM Comparison Runs

We re-trained iFAM [2] on every benchmark in Tables 1 and 2 using the official code release with all default hyperparameters from the iFAM paper (its Appendix A).

Number of foreground parts K. The iFAM paper reports $K{=}8$ as its Waterbirds default and $K{=}4$ as its MetaShift Cat vs. Dog default; the remaining benchmarks we evaluate (Spawrious O2O Hard, Spawrious M2M Hard, MetaShift Animals) are not in the iFAM paper, so no published default exists. To give iFAM the strongest comparison, we trained at both $K{=}4$ and $K{=}8$ on every benchmark and report both rows in Tables 1 and 2.

Our Waterbirds numbers ($K{=}8$: 98.8 accuracy, 96.1 WGA) match the iFAM paper's reported Waterbirds results closely, which we take as evidence that our implementation is faithful; the same default hyperparameters are used across all five benchmarks (excluding K which we varied).

R Comparing A^2 to using a peak-localization selector

Table 45: **ImageNet-calibrated layer transfer.** Changes in downstream worst-group accuracy, computed as the ImageNet-selected layer minus the last layer. MetaShift Animals reports worst-class accuracy. For each selector, the chosen layer maximizes mean patch-normalized attention mass within ImageNet validation bounding boxes and stays fixed across all five tasks. Layer indices are zero-indexed. Positive changes are bold. Values of 0.0 are present when the last layer is the best attention localization layer on ImageNet validation.

Selector	Terminal → selected index	Waterbirds	Spawrious O2O	Spawrious M2M	MetaShift CatDog	MetaShift Animals
DINOv1-S	11 → 10	−0.47	−1.74	−2.62	−3.23	**+1.55**
DINOv1-B	11 → 9	−3.12	**+1.70**	**+6.47**	−0.78	−2.38
DINOv2-S	11 → 11	0.00	0.00	0.00	0.00	0.00
DINOv2-B	11 → 11	0.00	0.00	0.00	0.00	0.00
DINOv2-L	23 → 21	**+8.72**	**+2.16**	**+8.05**	**+3.88**	**+0.61**
DINOv2-G	39 → 33	**+2.96**	**+2.89**	**+6.12**	**+4.01**	−2.15
iBOT-S	11 → 9	−3.27	−1.50	**+0.22**	**+1.55**	**+2.56**
iBOT-B	11 → 9	−0.93	**+2.64**	**+5.65**	**+0.39**	**+3.38**
iBOT-L	23 → 15	**+2.96**	**+5.57**	−1.48	**+6.86**	−0.27
MAE-B	11 → 11	0.00	0.00	0.00	0.00	0.00
MAE-L	23 → 15	**+15.26**	−8.10	−3.72	**+2.07**	**+12.60**
MAE-H	31 → 18	**+23.52**	**+13.43**	**+2.90**	**+13.84**	**+20.97**
DINOv3-S	11 → 11	0.00	0.00	0.00	0.00	0.00
DINOv3-S+	11 → 11	0.00	0.00	0.00	0.00	0.00
DINOv3-B	11 → 11	0.00	0.00	0.00	0.00	0.00
DINOv3-L	23 → 22	0.00	−1.17	**+2.59**	**+4.14**	−4.61
DINOv3-H+	31 → 30	**+3.58**	**+2.87**	**+1.89**	**+0.52**	−4.91
DINOv3-7B	39 → 34	**+1.40**	**+2.07**	−0.38	**+0.52**	**+1.02**

Table 46: We compare A^2 performance (using the smallest model in a family to select crops and the largest to embed) with a classifier that instead selects crops using the layer that maximizes mean patch attention mass in ImageNet validation bounding boxes. Once calibrated on ImageNet validation, this layer is used throughout all five tasks. A^2 uses the last layer attention from the smallest model in the family, and the largest model in the family to embed those crops. Scores shown are worst group accuracy except for MetaShift Animals which is worst class accuracy. A positive delta favors A^2. A^2 outperforms in 16 out of 25 cases, showing that decoupled selection and embedding is still useful even after calibration. Layers are 0 indexed.

Selector (model)	Layer	Waterbirds	Spawrious O2O	Spawrious M2M	MetaShift CatDog	MetaShift Animals
DINOv1 (crop embedder: ViT-B/16)						
Largest at peak (B)	9	68.54	**65.83**	**37.75**	64.17	53.41
A^2 (S, terminal)	11	**72.12**	64.84	37.37	**66.49**	**55.79**
Δ: A^2 − largest (pp)	–	**+3.58**	−0.99	−0.38	**+2.33**	**+2.38**
DINOv2 (crop embedder: ViT-G/14)						
Largest at peak (G)	33	95.02	76.44	**67.87**	**77.75**	67.04
A^2 (S, terminal)	11	**95.33**	**80.10**	64.65	76.58	**70.83**
Δ: A^2 − largest (pp)	–	**+0.31**	**+3.66**	−3.22	−1.16	**+3.79**
iBOT (crop embedder: ViT-L/16)						
Largest at peak (L)	15	73.36	64.05	27.02	66.36	62.85
A^2 (S, terminal)	11	**76.17**	**64.79**	**28.47**	**69.21**	**63.15**
Δ: A^2 − largest (pp)	–	**+2.80**	**+0.74**	**+1.45**	**+2.85**	**+0.31**
MAE (crop embedder: ViT-H/14)						
Largest at peak (H)	18	**39.56**	42.60	18.40	**43.86**	**29.65**
A^2 (B, terminal)	11	37.23	**46.64**	**21.02**	41.27	26.76
Δ: A^2 − largest (pp)	–	−2.34	**+4.04**	**+2.62**	−2.59	−2.89
DINOv3 (crop embedder: ViT-7B/16)						
Largest at peak (7B)	34	88.32	**76.80**	64.49	59.90	**76.77**
A^2 (S, terminal)	11	**92.83**	73.56	**65.03**	**67.14**	76.36
Δ: A^2 − largest (pp)	–	**+4.52**	−3.24	**+0.54**	**+7.24**	−0.41

S Compute Resources

All experiments were run on a single node with 8×NVIDIA A100 GPUs (80 GB memory each).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the main claims in the abstract and introduction reflect the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Section 7 (Scope, Limitations, and Future Work).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include any theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All datasets are open source, experiment setup is described in the appendix, and full code is released with a link provided in the main paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have released our code `https://github.com/SreehariRamMohan/A2` and the MetaShift Animals Dataset `https://github.com/SreehariRamMohan/MetaShift_Animals`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details (splits, hyperparameters, optimizers) are included in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: A^2 uses a deterministic algorithm to fit linear probes on frozen backbones, the experiments with stochasticity are run over multiple seeds and standard deviations are written in the tables where applicable.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This is discussed in the appendix Section S.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, the research does conform to the code of ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Robust visual classifiers are essential for fairness – it goes the other way as well, classifiers can always be used for harm.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not pose these risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, all datasets, pretrained models, and existing implementation code used is cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release a new dataset based on MetaShift, MetaShift Animals, and this is documented in the appendix Section D (The Data Card). This dataset is released and the link is provided in the main paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Does not involve crowdsourcing experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Declaration is not required.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.