Learning New Dimensions of Human Visual Similarity using Synthetic Data Supplementary Materials

Anonymous Author(s)
Affiliation
Address
`email`

For supplemental materials, we include an HTML page, *index.html*, containing qualitative results, including additional triplets from our dataset and visualizations of image retrieval and image reconstruction, and a text file listing the text categories used to generate our dataset, *classes.txt*. In this document, Sec. A contains additional methodological details on dataset collection and model training, Sec. B provides more analyses of our model, and Sec. C discusses the broader impact of our work, limitations, and licensing.

A Method

A.1 AMT Details

User interface. During the 2AFC study, users are instructed with the following prompt:

```
You will see three images: one labeled "Reference",
one labeled "Image A", and one labeled "Image B".
Select whether Image A or B is more similar to the Reference.
```

For each task, users see an image triplet with the reference in the center and distortions on either side (randomized). Each user completes 2 practice tasks, 50 real tasks, and 10 sentinels (randomly placed), averaging 3 minutes for the entire assignment. We discard responses from users who do not respond with 100% sentinel accuracy. See Figure 1 (left) for an example of the user interface.

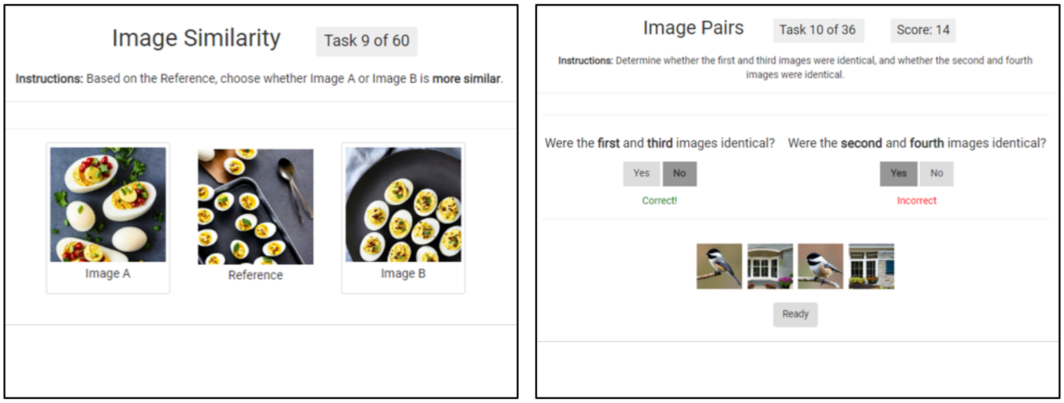
Instructions for JND are as follows:

```
You will see four images one after the other.
Determine whether the first and third images are identical, then whether
the second and fourth images are identical.
Each correct answer earns 1 point.
```

Users are shown four images for 500 ms each with a 1 second gap inbetween, and are then prompted to answer the two questions in Figure 1 (right). They are given feedback and a score, though these have no bearing on the JND results themselves. Each user completes 2 practice tasks, 24 tasks with “different" pairs, and 12 tasks with “same" pairs, averaging 10 minutes for the entire assignment.

The object retrieval study instructs users with the following text:

```
You will see a reference image and five other images under it.
Pick the image that is most similar to the reference.
```



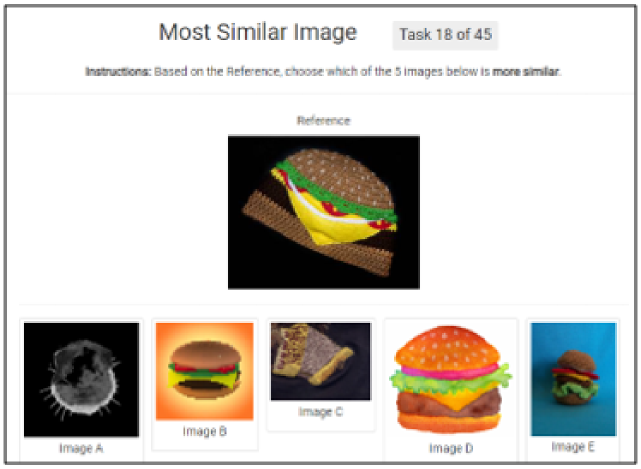
Instructions: Based on the Reference, choose whether Image A or Image B is **more similar**.

Instructions: Determine whether the first and third images were identical, and whether the second and fourth images were identical.

Were the **first** and **third** images identical?

Were the **second** and **fourth** images identical?

Figure 1: User interface for AMT studies. (Left) One image triplet shown to a user in 2AFC, who is prompted to pick "Image A" or "Image B". (right) In each JND task, users are shown a sequence of images and asked whether the image pairs were identical. Upon answering, they are given the correct answers.



Instructions: Based on the Reference, choose which of the 5 images below is **more similar**.

Figure 2: User interface for image retrieval study. To evaluate object retrieval performance, users are asked to pick the image (A-E) most similar to the reference.

See Figure 2 for an example of an object retrieval task. Each user completes 2 practice tasks, 40 real tasks, and 5 sentinels (randomly placed), averaging 3 minutes for the entire assignment. We discard responses from users who do not respond with 100% sentinel accuracy.

Cost breakdown. Across 10 rounds of 2AFC studies, we show users 477,964 triplets (see Table 1 for the full breakdown). Each user is paid \$0.50 for one assignment consisting of 50 triplets, averaging \$10.00/hr. In total, we pay users \$4779.64 to collect 2AFC judgments on our dataset.

We run JND on image triplets in our test set that received >5 2AFC judgments (1,824 triplets total). Each user is paid \$2.00 for one assignment consisting of 48 image pairs, averaging \$12.00/hr. In total, we pay users \$156.00 to collect JND judgments on our test set.

Our object retrieval user study is conducted over 200 query images with 10 nearest neighbors. Users are paid \$0.50 for one assignment consisting of 40 tasks, averaging \$10.00/hr. In total, we pay users \$40.50 to collect object retrieval judgments.

A.2 Additional Dataset Details

2AFC task. Following Sec. 3.2 in the main text, we filter the images over 10 rounds to obtain cognitively impenetrable triplets where humans tend to vote the same way despite various differences between the images. Statistics for each round of filtering is reported in Tab. 1, which leaves us with roughly 20% of the original triplets containing unanimous votes. We discard all votes from any turker who fails the sentinel task. As a result, not all triplets have the same number of votes. During

Round	# unanimous	# additional sentinel failures	# kept
1	100,000	0	100,000
2	74,346	6,750	81,096
3	63,423	2,411	65,834
4	51,097	592	51,689
5	43,615	289	43,904
6	37,696	113	37,809
7	30,420	16	30,436
8	25,079	0	25,079
9	22,098	0	22,098
10	20,019	0	20,019

Table 1: Filtering for cognitively impenetrable triplets. We start with 100K triplets, and advance triplets to the subsequent round if the human vote remains unanimous, or if the added vote came from a user who did not pass the sentinels and thus the vote is inconclusive (the vote from this user is discarded).

training, we further discard triplets with five or fewer unanimous votes. The resulting sizes of the train, validation, and test splits and additional statistics on each split are reported in Tab. 2-top.

JND task. The JND triplets are meant to capture the decision boundary where two different images are similar enough to be confused as identical, in the presence of a masking image. Each triplet is divided into two pairs – Ref vs. A and Ref vs. B. These pairs are presented to different turkers in different interleaving sequences, and we collect three judgments for each pair and take the majority vote among the three judgments, thus six judgments in total per triplet (Tab. 2-bottom).

Dataset	Split	# Samples	Avg # Votes	Consensus Type
2AFC	Train	15941	7.11	Unanimous
	Validation	1958	7.07	Unanimous
	Test	2120	7.04	Unanimous
JND	Test	608	6	Majority

Table 2: Dataset Statistics. For each dataset, we report the number of samples and the average number of votes for each triplet after filtering for sentinel failures. Labels for the 2AFC dataset are based on a unanimous vote, while for JND we take the majority vote over three trials per pair (six trials per triplet).

A.3 Model Training.

For all fine-tuned models (both `Tuned - MLP` and `Tuned - LoRA`) we use the NIGHTS training dataset, with an 80%-10%-10% split as described in Tab. 2, and images of size 768×768 resized to 224×224. We train on a single NVIDIA GeForce RTX 3090 or NVIDIA TITAN RTX GPU with an Adam optimizer, learning rate of $3e-4$, weight decay of 0, and batch size of 512 (non-ensemble models) and 16 (ensemble models). We tune the number of training epochs using the validation set; for the `Tuned - LoRA` ensemble model (DreamSim) we train for 6 epochs. For `Tuned - LoRA` models we use rank $r = 16$, scaling $\alpha = 0.5$, and dropout $p = 0.3$. Training time is approximately 30 min/epoch for LoRA-tuned models, and 15 min/epoch for MLP-tuned models.

B Experiments

B.1 Additional Evaluations

Full Evaluation on Large Vision Models. In Sec. 5.1 and Fig. 4 in the main text we report results using the best-performing setting of various large vision model backbones. Tab. 4 evaluates additional model settings, spanning different ViT model sizes, patch sizes, and strides. Tab. 3 shows the experimental variation over multiple runs, in which the LoRA variation consistently outperforms the MLP variation.

	Independent training seeds						
Ensemble tuning	**Trial 1**	**Trial 2**	**Trial 3**	**Trial 4**	**Trial 5**	**Avg.**	**Stdev.**
MLP	93.4	93.1	93.1	93.9	92.9	93.3	0.326
LoRA	96.2	96.3	95.1	96.0	95.8	95.9	0.416

Table 3: Experimental variation for ensemble models. We train the `Tuned - MLP` and `Tuned - LoRA` ensemble models on 5 seeds each and record their test accuracies at the same epoch as was recorded in the main paper.

Model				**Alignment**		
Model Class	**Model Name**	**Model Type**	**Feature**	**Overall**	**ImageNet**	**Non-ImageNet**
Base Models	PSNR	–	–	57.2	57.3	57.0
	SSIM	–	–	57.0	58.5	55.8
Prior-Learned Metrics	LPIPS	AlexNet-Linear	–	70.8	69.3	72.7
	DISTS	VGG16	–	86.0	87.1	84.5
Base Models	CLIP	ViT B/16	Embedding	82.2	82.6	81.7
		ViT B/32	Embedding	83.1	83.8	82.1
		ViT L/14	Embedding	89.8	83.3	79.8
	DINO	ViT S/8	CLS	89.0	89.7	88.0
		ViT S/16	CLS	89.6	90.2	88.8
		ViT B/8	CLS	88.6	88.6	88.5
		ViT B/16	CLS	90.1	90.6	89.5
	MAE	ViT B/16	CLS	81.6	81.7	81.5
		ViT L/16	CLS	81.5	81.1	82.0
		ViT H/14	CLS	81.7	81.4	82.2
	OpenCLIP	ViT B/16	Embedding	87.1	87.8	86.2
		ViT B/32	Embedding	87.5	87.5	87.6
		ViT L/14	Embedding	85.9	86.7	84.9
	Ensemble	ViT B/16	Mixed	90.8	91.6	89.8
Tuned MLP	CLIP	ViT B/32	Embedding	87.3	88.2	86.2
	DINO	ViT B/16	CLS	91.2	91.8	90.3
	MAE	ViT B/16	CLS	84.9	85.3	84.3
	OpenCLIP	ViT B/32	Embedding	89.9	91.0	88.5
	Ensemble	ViT B/16	Mixed	93.3	94.2	92.2
Tuned LoRA	CLIP	ViT B/32	Embedding	93.8	94.0	93.6
	DINO	ViT B/16	CLS	94.6	94.6	94.5
	MAE	ViT B/16	CLS	86.5	87.3	85.4
	OpenCLIP	ViT B/32	Embedding	95.5	96.5	94.1
	Ensemble	ViT B/16	Mixed	96.1	96.6	95.5

Table 4: Alignment on NIGHT test set. We evaluate alignment on additional model settings, and separate the test set into ImageNet categories and non-ImageNet categories.

As some models are adapted from backbones trained on ImageNet [3] (including the prior learned metrics and DINO), we split our dataset into categories contained in ImageNet and those not in ImageNet, and evaluate alignment on each split. Performance on both splits is highly correlated (Fig. 3), suggesting that the notions of visual similarity are related regardless of whether or not the triplet was generated from an ImageNet category, and whether or not the model was trained only on ImageNet.

Alignment on Alternative Datasets. As depicted in the left plot of Figure 4, training on our dataset (with either tuning method) indeed improves BAPPS metric-human alignment in nearly every model, suggesting that some of these patch-based distortions are implicitly still captured in our dataset. We observe that MAE exhibits the best out-of-the-box performance, indicating a greater sensitivity to

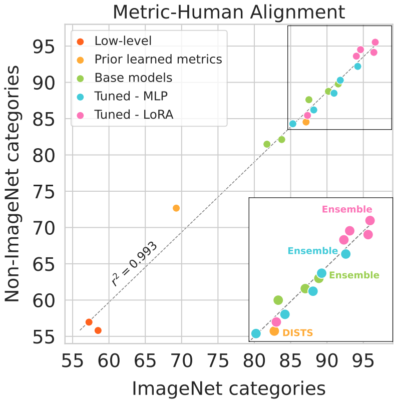


Figure 3: Alignment on ImageNet and Non-ImageNet triplets. We split the test set into triplets generated from ImageNet categories and Non-ImageNet categories, as some model backbones are trained only on ImageNet images. For all models, alignment is highly correlated between the two splits.

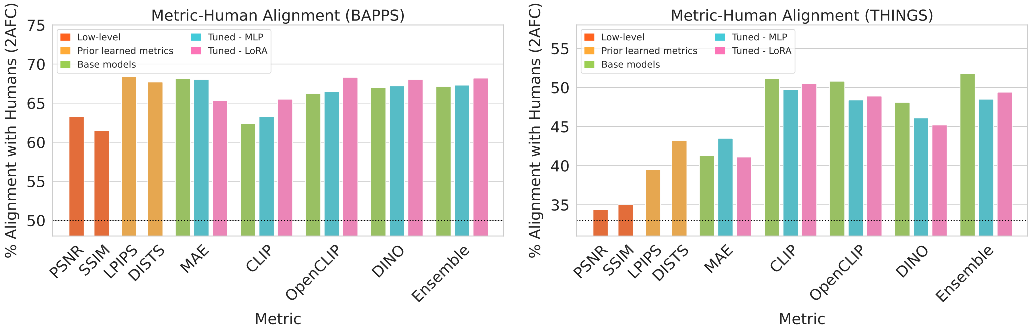


Figure 4: Evaluation on existing low-level and high-level similarity datasets. (Left) Despite never being trained for low-level similarity, LoRA-finetuned models on OpenCLIP, DINO, and Ensemble achieve similar human alignment to LPIPS, which was directly trained on the BAPPS dataset. (Right) The THINGS dataset measures high-level conceptual similarity, rather than appearance similarity. As such, we find that LoRA finetuning on our dataset degrades performance, as our triplets contain appearance similarity, by design.

lower-level image distortions (e.g. color and shape) than DINO, CLIP or OpenCLIP. Surprisingly however, it is the only model whose performance decreases on BAPPS as it is further tuned. DINO, CLIP and OpenCLIP are not as sensitive to the image distortions in BAPPS, suggesting that before tuning, they are more attuned to higher-level image attributes that the dataset does not capture.

On THINGS, further training actually *diminshes* alignment with humans (see right plot of Figure 4). CLIP and OpenCLIP's superior performance on this dataset supports our hypothesis that they are more well-adjusted to higher-level image attributes, which THINGS aims to capture, rather than appearance-level variations.

Our evaluations across these three datasets show that, as we train perceptual metrics that align more closely with human perceptual similarity, we also improve on low-level similarity but perform slightly worse on high-level image distortions. These results suggests that humans, when making an automatic judgment, are more inclined to focus on immediate visual differences (captured in BAPPS and our dataset) rather than the image's category, context, or related words.

Metric	Image Part	Ours	DINO	OpenCLIP	DISTS	LPIPS
Color (RGB)	Foreground	0.583	0.537	0.512	0.594	0.605
Color (RGB)	Background	0.587	0.572	0.555	0.64.1	0.647
Color (RGB)	Total	0.584	0.558	0.541	0.629	0.653
Luminance	Foreground	0.545	0.519	0.499	0.568	0.556
Luminance	Background	0.555	0.543	0.541	0.577	0.544
Luminance	Total	0.541	0.529	0.515	0.569	0.556
Depth	Total	0.542	0.536	0.533	0.547	0.558
Category Histogram	Things	0.587	0.583	0.551	0.553	0.538
Category Histogram	Stuff	0.595	0.588	0.579	0.625	0.613
Presence of Person	-	0.553	0.526	0.552	0.518	0.538
Presence of Furniture	-	0.531	0.522	0.542	0.534	0.536
Presence of Textiles	-	0.528	0.524	0.531	0.516	0.536

Table 5: Automated Metrics on COCO. Alignment of hand-crafted metrics with model decisions on the COCO dataset, which provides ground-truth semantic labels.

Metric	Image Part	Ours	DINO	OpenCLIP	DISTS	LPIPS
Color (RGB)	Foreground	0.717	0.70	0.693	0.687	0.606
Color (RGB)	Background	0.654	0.662	0.647	0.664	0.62
Color (RGB)	Total	0.698	0.679	0.676	0.66	0.643
Luminance	Foreground	0.631	0.626	0.622	0.614	0.561
Luminance	Background	0.593	0.595	0.588	0.603	0.568
Luminance	Total	0.594	0.606	0.598	0.592	0.565
Depth	Total	0.542	0.536	0.533	0.547	0.558

Table 6: Automated Metrics on NIGHTS. Alignment of hand-crafted metrics with model decisions on our dataset.

Alignment with low-level features. In Sec. 5.2 and Fig. 7 of the main text we report results on the alignment between our metric, OpenCLIP, DINO, LPIPS, and DISTS with low-level and semantic metrics for the COCO dataset. In Tab. 5 we also report additional, fine-grained results for COCO triplets. We use CarveKit [1] to segment out the foreground and background of each image, and then breakdown how well each metric agrees with RGB color histogram similarity, luminance histogram similarity, and depth map distance, for foreground, background, and the full image. For color histograms we use 32 bins for each channel, and for luminance histograms we use 10 bins.

We also examine semantic features. For each image, we find the percentage of area that each semantic category occupies, and then compute the alignment between the difference in area for each category and perceptual metrics. Note that when the difference in area is the same for both pairs in a triplet, it is counted as 50% alignment. When the difference in area is smaller for the pair chosen by the perceptual metric, it is counted as 100% alignment (and 0% in the case of disagreement). In Tab. 5 we show the five semantic categories most aligned with our metric. Our metric has a 55% alignment score with the "people" category, however does not seem to align well above chance for other categories.

In Tab. 6 we show alignment with low-level metrics for our dataset (which does not have semantic annotations). On our dataset there is higher alignment with color, luminance, and depth across all metrics, as compared to COCO triplets. This is likely because the images in each of our dataset's triplets all share the same semantic category, making lower-level features more important than for the randomly-chosen COCO triplets. Our model aligns significantly better with foreground metrics – particularly foreground color – whereas LPIPS aligns slightly better with background.

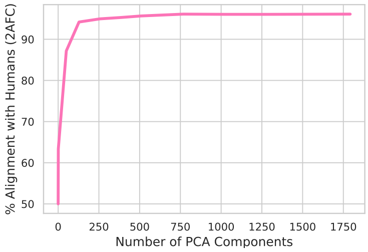


Figure 5: Ablating Feature Dimension. We apply a PCA decomposition to the output features of our model and vary the number of dimensions kept.

# PCA Components	2AFC Score
1	63.4
512	95.7
768	96.1
1792	96.1

Table 7: Feature PCA Decomposition. We list 2AFC scores as a function of the number of PCA components kept, beating both the CLIP/OpenCLIP dimensionality (512) and DINO (768).

Dimensionality reduction with PCA. Our model consists of the concatenation of the DINO, OpenCLIP, and CLIP backbones, and therefore uses a higher-dimensional feature space to compute similarity compared to each of these models independently. To investigate whether the increased dimensionality is critical for improving human alignment, we ablate feature dimensions by applying PCA, taking a certain number of the top components, as seen in Fig. 5 and Tab. 7. We can achieve comparable performance using just 500 of the top components, similar to the 512 dimensions of the CLIP and OpenCLIP embedding outputs, suggesting that the improved alignment is not just due to the higher-dimensional feature space used to compute similarity, but rather the additional capacity and model priors obtained from ensembling different models.

B.2 Additional Visualizations

Qualitative metric comparisons. In Sec. 5.2 and Fig. 6-7 of the main text we quantitatively analyze the differences between metrics. Here, we also provide a qualitative analysis by comparing image pairs that achieve high and low similarity scores for each metric.

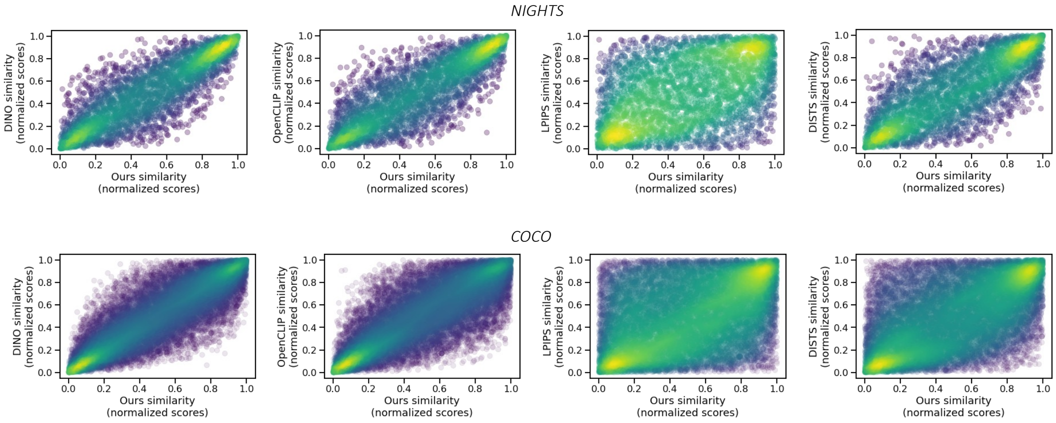


Figure 6: Correlation between metric scores. For image pairs from our dataset (above), and the COCO dataset (below), we plot similarity scores from our metric against similarity scores from DINO, OpenCLIP, LPIPS, and DISTS. Our metric's scores are most correlated with other ViT-based metrics, and correlate better with DISTS than LPIPS.

In Fig. 6 we plot, for each image pair in our dataset, the similarity scores from our metric against similarity scores from DINO, OpenCLIP, DISTS, and LPIPS. We show the same for image pairs drawn from the COCO dataset. In Fig. 7 and Fig. 8 we show the pairs where our metric most agrees and most disagrees with DINO, OpenCLIP, DISTS, and LPIPS, for our dataset's test set and the COCO dataset. Note that for COCO we draw random pairs of images that share at least instance category so that pairs have some semantic commonality, enabling better visualization of qualitative differences between metrics.

Comparing DISTS, the top-performing prior learned similarity metric, to our model, we find that DISTS is sensitive to structural changes despite similar overall appearance (such as the width of

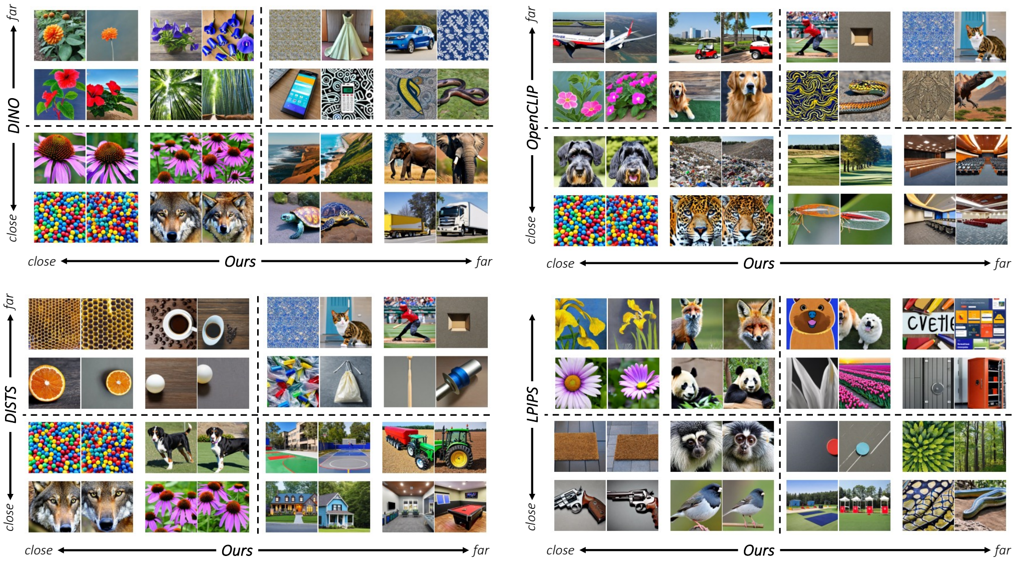


Figure 7: Visualizing differences between our model and other metrics for NIGHTS. We show the examples where our metric most agrees and disagrees with other metrics. Primary differences are that our metric is more sensitive to major changes in pose, semantics, and color. It is less sensitive to granular changes in structure when overall appearance is preserved, such as the honeycomb example in the DISTS quadrant and the forest example in the DINO quadrant.

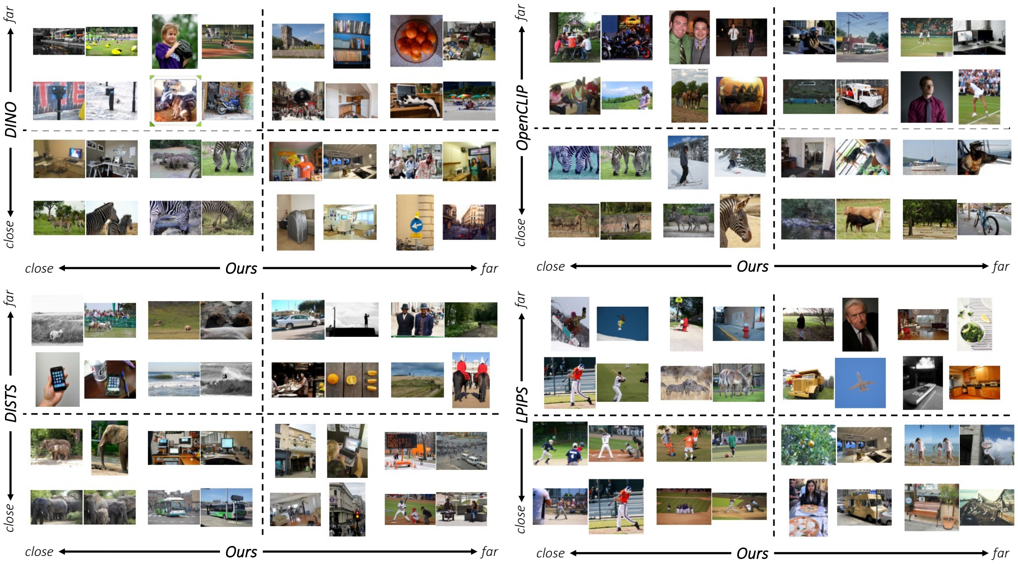


Figure 8: Visualizing differences between our model and other metrics for COCO. We show examples where our metric most agrees and disagrees with other metrics for pairs drawn from the COCO dataset. These pairs share fewer appearance similarities than pairs drawn from our dataset. Our metric seems particularly sensitive to foreground semantic similarities, such as the horse pair in in the OpenCLIP quadrant and the snowboarders in the LPIPS quadrant.

the honeycomb or the position of similar objects), while our model rates these pairs as nearby. On the other hand, pairs that are far in our feature space but close in DISTS feature space have less appearance similarity (*e.g*. the houses and rooms of different colors). Comparing to deep ViT features (DINO, OpenCLIP) our model is more likely to rate pairs with similar foreground color/appearance as similar, and less likely for pairs that are similar semantically but not appearance-wise. For COCO pairs, where there are fewer appearance similarities than in our dataset, our model chooses pairs that are similar semantically first, and only then appearance-wise.

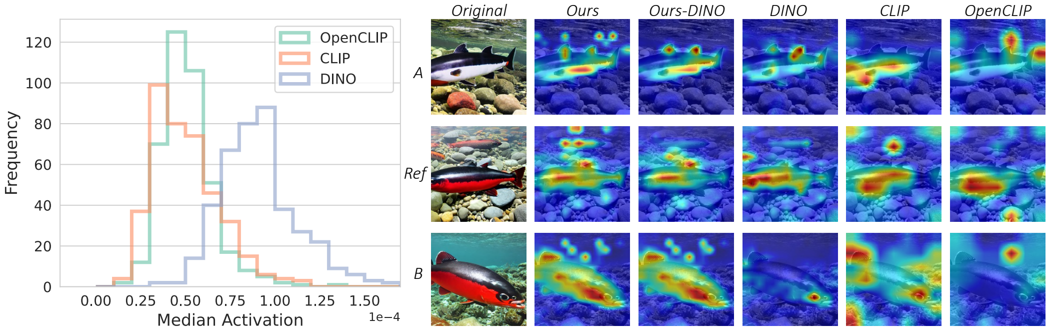


Figure 9: Visualizing attention maps. The finetuned DINO branch of our model has the largest contribution in the attention map [2], with the largest median activation compared to the CLIP and OpenCLIP branches (computed over 400 test set images). As such, in our model, the overall attention map is similar to the attention map from only the DINO branch. Compared to the pretrained backbones, our model better captures the entire object of interest (the fish body) while reducing spurious attention in background regions.

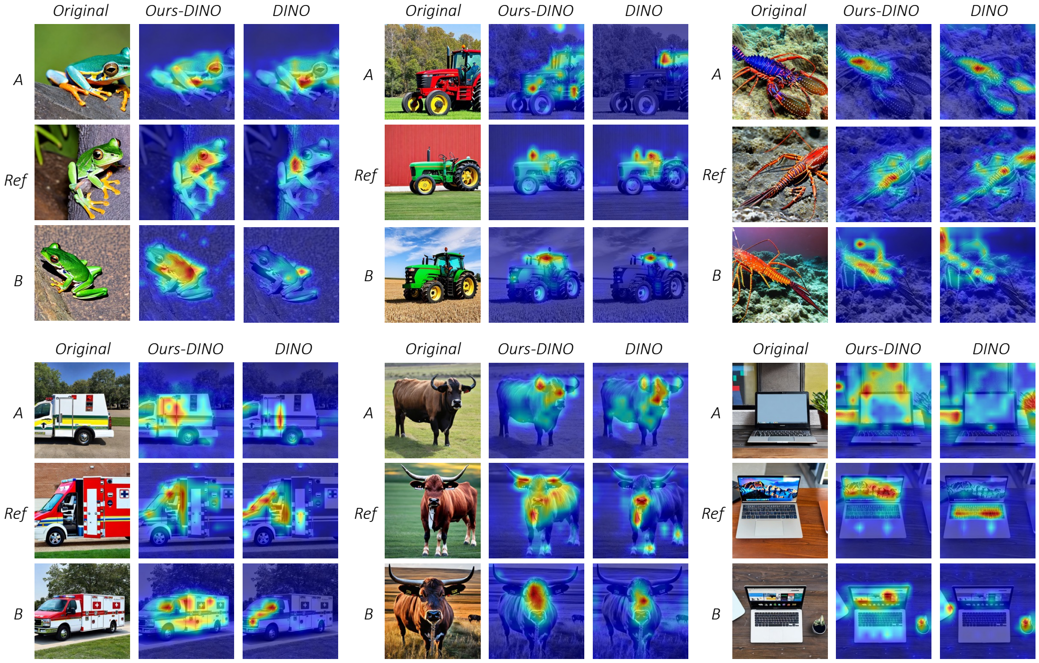


Figure 10: Comparing our model and baseline attention maps from the DINO branch. Focusing on the DINO branch, which has the largest contribution in our model's attention map, we compare the attention maps of the pretrained and finetuning models. The finetuned attention map in our model better captures the foreground object or relevant regions of interest. In all examples, the DINO baseline selects the A image, while humans and our model select the B image.

Attention Map Visualizations. As an alternate way of understanding our model's similarity decisions, we visualize the transformer attention maps following Chefer et al. [2]. Our model consists of finetuned version of DINO, CLIP, and OpenCLIP backbones, and the resulting attention map places the largest activations on the finetuned DINO backbone (Fig. 9-left). Accordingly, the overall attention map looks largely similar to the attention map constructed from only the finetuned DINO branch. Compared to the pretrained versions of each model backbone, the finetuned model better captures full object identity (such as over the entire body of the fish), while also minimizing spurious attention values in the background (Fig. 9-right). Consistent with earlier analysis, this supports the fact that the foreground plays a larger role in the DreamSim similarily judgement than the background.

As the DINO model has the largest contribution in the attention maps, Fig. 10 shows additional examples focusing on the difference between the finetuned DINO backbone within our model, and the pretrained DINO backbone prior to finetuning. This visualizes how the DINO backbone changes as it is finetuned on our dataset. Our finetuned model better captures the foreground object, while attention maps from the pretrained DINO backbone may only focus on small portions of the object. In the lobster example (Fig. 10 top-right), our model places attention on the relevant parts of the object, such as the lobster body rather than the claws in the A image as the claws do not appear in the other two images.

B.3 Additional Results on Applications

Please see additional results on image retrieval and image reconstruction in the attached HTML page. To quantitatively evaluate our image retrieval results, we conduct a user study to collect preferences across returned results from LPIPS, DISTS, DINO, OpenCLIP, and DreamSim. In Figure 11, we provide a detailed breakdown of user preferences for neighbors 1 through 10 across ImageNet-R and COCO, along with error bars marking 1 standard deviation above and below each result.

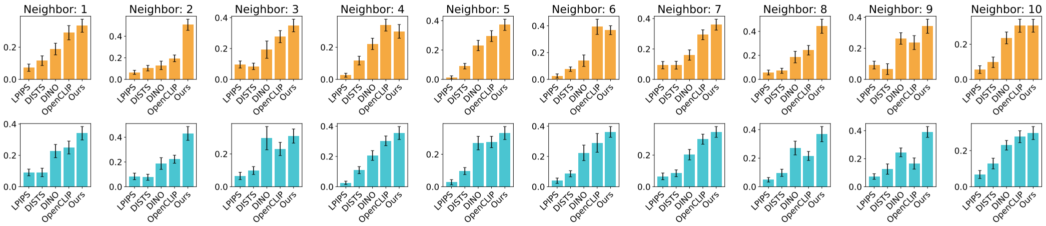


Figure 11: User preferences for image retrieval results by metric. We conduct a user study that collects preferences for retrieval results output by LPIPS, DISTS, DINO, OpenCLIP, and DreamSim. For most instances of the first 10 nearest neighbors, users preferred our metric's output, followed by DINO and OpenCLIP. We visualize one standard deviation above and below each bar.

C Discussion

Broader Impacts and Limitations. Our model and dataset are developed from pretrained Stable Diffusion, CLIP, OpenCLIP and DINO backbones. As such, our model can inherit and propagate biases existing in these models for decisions on downstream tasks. Our dataset is generated using prompts that describe a single-word category and is filtered to remove images containing sensitive topics, such as violence or human faces. As a result, the dataset has a large focus on object-centric domains, and content containing humans is considered out-of-domain. The resulting dataset and model does not capture the full range of human similarity judgements, but only the variations that we can capture in our synthetically-generated dataset.

Licenses. The Icons for the teaser figure as well as the images for the inversion experiments are licensed by Adobe Stock, under the Adobe Stock Standard License, and by Pixabay, under their content license.

IRB Disclosure. We received IRB approvals for our AMT experiments from all of the institutions involved. Accordingly, we took measures to ensure participant anonymity and refrained from showing them potentially offensive content.

References

[1] CarveKit. `https://github.com/OPHoperHPO/image-background-remove-tool/`.

[2] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021.

[3] 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. Ieee, 2009.