Evaluating Foundation Models' 3D Understanding Through Multi-View Correspondence Analysis

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper extends the Hummingbird framework with the Multi-View ImageNet (MVImgNet) dataset to evaluate how foundation model image encoders handle in-context object segmentation under unseen camera angles. We group MVImgNet object views and construct memory banks from selected viewpoints. We assess generalization by evaluating performance on held-out angles. Across six pretrained Vision Transformer (ViT) models—CLIP, DINO, DINOv2, SigLIP2, C-RADIOv2, and TIPS—we find that DINO-based models perform best: DINO leads when more context viewpoints are available, while DINOv2 is strongest with fewer reference views. These results highlight the benefits of contrastive pretraining for robust performance across large viewpoint shifts.

1 Introduction

In recent years, we have seen the rise of foundation models [2, 20, 11, 23]. These large-scale pretrained models support many tasks, including visual understanding, which enables direct comparison of their encoders.

Despite their impressive capabilities, state-of-the-art ViTs can suffer catastrophic failures when objects are viewed from unusual angles, as demonstrated by recent angle-sensitivity studies [15]. However, most vision benchmarks emphasize single-view tasks or dense synthesis [10], leaving segmentation robustness under camera rotations comparatively underexplored.

To address viewpoint variability, recent work has explored in-context learning (ICL) as a means to adapt models to unseen views without retraining. ICL is the ability of a model to perform new tasks by conditioning on a prompt [22]. An example is Hummingbird [1], a memory-augmented ViT for ICL. Its encoder extracts dense image features and projects them into key–value pairs stored in a dynamic memory. At query time, cross-attention over the keys assigns soft nearest-neighbor (NN) weights to aggregate values and predict novel views in-context. This modular design permits the use of any encoder.

In this work, we evaluate foundation model encoders for in-context object segmentation under novel camera angles. We extend the Hummingbird framework with the MVImgNet dataset [24], grouping object views into angular bins. By constructing dynamic memory banks from selected viewpoints and evaluating segmentation performance on held-out angles, we assess how well different encoders generalize across viewpoint shifts. For clarity, some terms are followed by their `variable name` to maintain cohesion with our codebase.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

2 Related work

Recent developments in view generation have significantly improved the understanding of 3D scene structure in computer vision, setting new benchmarks along the way. A notable example is NeRF [10], which achieves state-of-the-art novel view synthesis by optimizing a continuous volumetric function from sparse input images. While NeRF addresses generation, other studies investigate recognition in 3D understanding. Ruan et al. [15] demonstrate that modern recognition models (ResNets [5], ViTs [4], Swin [8], and Masked Autoencoders (MAEs) [6]) remain sensitive to viewpoint shifts in 3D scenes. With adversarial training, they improve invariance to 3D viewpoint changes beyond standard rotation-based augmentations.

In parallel, Shifman and Weiss [19] show that state-of-the-art encoders such as CLIP and DINOv2 change predictions under a single-pixel shift in up to 40% of cases, despite extensive augmentation during training. Although their analysis centers on 2D translation invariance, the findings highlight that robustness to geometric changes, whether in 2D or 3D, remains limited in current vision encoders.

Beyond model architectures, multi-view datasets such as MVImgNet and PASCAL3D+ offer rich multi-view annotations that support the evaluation of cross-view generalization in object recognition. The rise of self-supervised Transformers has also opened new possibilities for downstream tasks such as segmentation. In particular, DINOv2 [3] produces robust embeddings that transfer effectively to prediction tasks, despite training without labels.

Nevertheless, a gap remains across these benchmarks and models. Most prior work emphasizes detection, classification, or view-angle estimation, but not the measurement of 3D contextual understanding through multi-view segmentation. To our knowledge, no benchmark currently evaluates a model's ability to generalize segmentation across viewpoint shifts using dynamic memory. Our proposed evaluation method, therefore, represents a novel contribution to this area.

3 Methodology

We evaluate the view generalization ability of frozen ViT models in semantic segmentation using a non-parametric, retrieval-based framework. Our approach builds on Hummingbird [1], which applies in-context learning (ICL) to vision tasks. Hummingbird supports evaluation of spatial perception and semantic understanding, but does not analyze performance variation under viewpoint changes. We address this by introducing a viewpoint binning protocol and a curated dataset subset for cross-view robustness analysis.

3.1 Encoders

We evaluate six pretrained ViT models: CLIP [13], SigLIP2 [21], DINO [3], DINOv2 [12], C-RADIOv2 [14, 7], and TIPS [9]. All models use a ViT-B/16 backbone, except DINOv2, which is released only as ViT-B/14 and reported to outperform its ViT-B/16 counterpart. All encoders are publicly released under open-source licenses (see Appendix C).

3.2 Inference Pipeline

We follow the Hummingbird [1] inference pipeline, where patch-level features from a frozen ViT are stored in a memory bank with one-hot semantic labels. During inference, query features are matched to the memory using FAISS with cosine similarity and $k = 30$ nearest neighbours. Hummingbird's cross-attention decoder aggregates the retrieved labels to produce segmentation predictions. The framework is released under the MIT license (see Appendix C).

3.3 Dataset

Our study is based on MVImgNet, a large-scale dataset with over 6.5 million frames across 238 object categories [24]. Each frame includes a segmentation mask, camera extrinsics, and a reconstructed 3D point cloud.

This dataset is chosen for the following three key reasons: (1) consistent object annotations across wide viewpoint ranges, enabling view generalization analysis; (2) camera extrinsics that allow precise

angular binning; and (3) scale and diversity that capture a broad range of object appearances and spatial configurations. It is released with CC BY-NC 4.0 for its code, and the dataset itself is distributed under a custom Terms of Use provided by the authors (see Appendix C).

Viewpoint binning. To study viewpoint robustness, we discretize relative camera angles into seven bins spanning $0^\circ - 90^\circ$ in 15° steps. Using COLMAP [16–18] extrinsics, we compute the relative rotation R_{rel} between each frame and the canonical (first) frame of the instance [1]. The angular deviation θ is computed as:

$$\theta = \arccos\left(\frac{\text{trace}(R_{rel}) - 1}{2}\right). \tag{1}$$

For each object instance, we select one representative frame per bin by choosing the frame with the smallest angular error relative to the bin center. Images and masks are stored per bin for downstream use.

Dataset construction. To evaluate cross-view generalization, we curated a subset of MVImgNet. Our goal was to select categories with sufficient angular coverage and manageable size for repeated memory-based evaluation.

The two criteria for selecting a category are: (1) the total zipped folder size must be 1–6 GB to ensure feasible data loading and storage; and (2) at least one object instance must span all seven angular bins with a maximum variance of 6 degrees. Categories that did not meet these criteria were excluded. For example, category 23 (*laptop*) was excluded, as it did not have an instance that reached an angle of 90°. This selection yielded 15 segmentation classes, excluding the background.

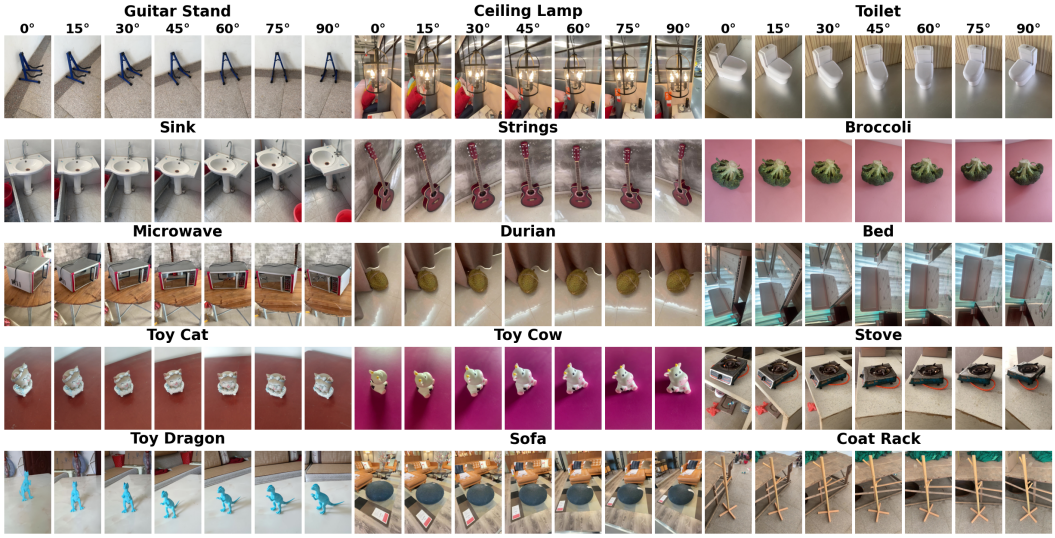

Figure 1: **Multi-view categories.** Each MVImgNet category is shown across all viewpoint bins (0°–90°) for the 15 selected classes.

For each category instance, we parsed the COLMAP camera pose data (`images.bin`) to extract camera extrinsics and compute relative angles. For each valid instance, we selected the closest frame to each angular bin center and organized the resulting RGB images and masks in a structured directory grouped by category and angle. Additionally, we computed angular selection errors per bin (see Table 5). Examples of multi-view objects from the resulting dataset are shown in Figure 1.

[1] COLMAP is a Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipeline that estimates camera poses and 3D scene geometry from a set of images. It is commonly used to recover camera extrinsics and sparse 3D reconstructions from unstructured image sets.

3.4 Model Input

The models we use have different native input sizes: 224×224 for CLIP ViT-B/16 [13], DINO ViT-B/16 [3] and C-RADIOv2 ViT-B/16-CPE [7, 14]; 512×512 for DINOv2 ViT-B/14 [12]; 504×504 for SigLIP2 B/16-512 [21]; and 448×448 for TIPS ViT-B/14-HR [9].

To ensure a fair comparison, the largest native input size across the models is used: 504×504 for models that use a patch size of 14 and 512×512 for the ones using a patch size of 16. The evaluation of models with a bigger input size than their original resolution requires an additional step of embedding interpolation. In our setup, we upsample the absolute positional embeddings using bicubic interpolation to align with the new input resolution, as it provides smooth spatial transitions and is widely used for resizing in vision models. All evaluations are performed with a batch size of 4. The full configuration details are provided in Table 6.

3.5 Metrics and Multi-view Evaluation

The segmentation quality is measured by mean Intersection-over-Union (mIoU) across all semantic classes, including the background. Each predicted segmentation mask consists of the background class (`jac_0`) and the 15 object classes (`jac_1`–`jac_15`)[2], where `jac_i` refers to the Intersection over Union (IoU) of class i.

We report three metrics per evaluation setting as follows: (1) class-wise IoU, (2) mIoU over all 16 classes (`jac_mean`), and (3) standard deviation (`jac_std`). Results are presented per model, reference bins, and validation bin to analyze viewpoint robustness. Although all models perform well on the background class (see Figure 4), we include it in the mIoU computation for completeness.

3.6 Hardware and software

All experiments were conducted on a high-performance computing cluster with 4 $\times$ A100 GPUs, 72 CPU cores, and a maximum wall time of 30 hours per job. Due to memory constraints, we used FAISS sharding to distribute the search index across GPUs and data parallelism to distribute the computation of a batch size of 4. For reproducibility, we fixed the random seed of Python, NumPy, PyTorch, and CUDA to 42.

4 Reproducing the Hummingbird benchmark

To validate our setup, we begin by reproducing the Hummingbird evaluation on Pascal VOC using a publicly available implementation. Our reproduction results are shown in Table 8, obtained with the same methodology and configuration described in the reference implementation. This step ensures our pipeline is consistent with the original benchmark and provides a reliable starting point for subsequent experiments.

5 Experiment A: cross-model generalization

We compare six pretrained ViTs in terms of their ability to generalize across viewpoints. The goal is to assess how different pretraining strategies affect performance when only limited reference views are available in the memory.

5.1 Experimental setup

We evaluate how well the models generalize to unseen viewpoints when trained on selected angular bins from the MVImgNet dataset. The reference and validation splits vary with the difficulty level, as defined in Table 1. Each model is evaluated using mIoU across all 16 classes (background plus 15 object categories) and all validation images. This setup enables a controlled comparison of each model's ability to interpolate between observed viewpoints and extrapolate to unseen ones.

[2]The 15 categories are ordered as follows: stove, sofa, microwave, bed, toy cat, toy cow, toy dragon, coat rack, guitar stand, ceiling lamp, toilet, sink, strings, broccoli and durian.

Table 1: **Reference and validation bin splits.** Difficulty levels are defined by the choice of reference bins (stored in the memory bank) and validation bins (unseen views for testing generalization).

Difficulty	Reference bins	Validation bins
Easy	$0°, 30°, 60°, 90°$	$15°, 45°, 75°$
Medium	$0°, 45°, 90°$	$15°, 30°, 60°, 75°$
Hard	$0°, 90°$	$15°, 30°, 45°, 60°, 75°$
Extreme	$0°$	$15°, 30°, 45°, 60°, 75°, 90°$

5.2 Results

We evaluated the models across four difficulty levels (see Table 1). As shown in Table 2, DINO achieved the highest mIoU in the first three difficulty levels, indicating strong generalization when trained on a wider range of bins. Under the Extreme difficulty, limited to a single reference bin ($0°$), DINOv2 surpassed all other models, suggesting greater robustness in extrapolating to unseen viewpoints.

Quantitative results illustrating these trends are shown in Figure 2, with per-category results in subsection D.2. Across all classes, DINO and DINOv2 produced clearer segmentations under more difficult conditions than the other models. For example, the class-specific breakdown on *ceiling lamp* (Figure 14) highlights DINOv2's robustness in unseen viewpoint bins.

Overall, CLIP consistently followed close behind DINO and DINOv2, while C-RADIOv2, TIPS, and SigLIP2 lagged significantly, especially under limited reference view conditions.

Table 2: **mIoU across difficulty levels.** Scores for each model are reported across increasing difficulty levels. DINO achieves the best performance under easier setups, whereas DINOv2 performs best when fewer reference views are available.

Model	Easy	Medium	Hard	Extreme
CLIP ViT-B/16	0.755 ± 0.130	0.748 ± 0.134	0.734 ± 0.140	0.701 ± 0.149
DINO ViT-B/16	**0.782 ± 0.132**	**0.774 ± 0.137**	**0.748 ± 0.153**	0.686 ± 0.171
DINOv2 ViT-B/14	0.763 ± 0.136	0.758 ± 0.139	**0.748 ± 0.143**	**0.728 ± 0.154**
C-RADIOv2 ViT-B/16-CPE	0.653 ± 0.129	0.636 ± 0.132	0.592 ± 0.141	0.506 ± 0.150
SigLIP 2 B/16-512	0.564 ± 0.150	0.551 ± 0.153	0.530 ± 0.157	0.481 ± 0.152
TIPS ViT-B/14-HR	0.667 ± 0.137	0.647 ± 0.146	0.588 ± 0.162	0.462 ± 0.169

5.3 Discussion

Although none of the models were explicitly trained for viewpoint understanding, the results show clear differences in their ability to generalize across views. DINO and DINOv2 perform best, indicating that their features capture object shape more reliably and remain consistent under viewpoint changes. Their self-supervised training, which aligns features from different augmentations of the same image, promotes consistency between local and global cues and leads to structured representations where similar object parts share similar embeddings.

DINO achieved the best results under the Easy difficulty, where multiple views of the object were available, while DINOv2 performed best under the Extreme difficulty with only a single reference bin. This contrast suggests that DINO's simpler self-distillation loss favors interpolation across views, whereas DINOv2's larger-scale self-supervised pretraining supports stronger extrapolation to unseen angles.

CLIP was consistently ranked third, reflecting some ability to encode visual–semantic regularities despite being trained for image–text alignment rather than geometric consistency. By comparison, SigLIP2, TIPS, and C-RADIOv2 performed substantially worse, indicating that objectives centered on semantic matching or mixed supervision may weaken part-level consistency needed for viewpoint-robust segmentation.

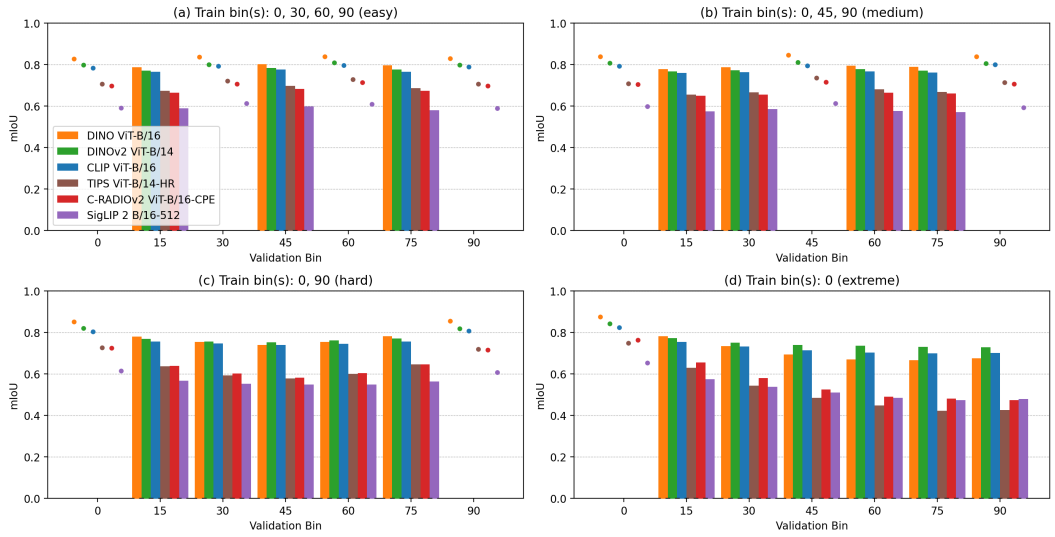

Figure 2: **Segmentation across viewpoint bins.** Each subplot (a)–(d) corresponds to a different difficulty. Bars show performance on validation bins, while dots show performance on the reference bins stored in the memory bank. Performance declines as validation angles move further from the reference bins, with DINO and DINOv2 degrading more slowly than multimodal encoders such as CLIP and SigLIP2.

6 Experiment B: breaking point analysis

We analyze whether and when each model experiences a sudden failure in viewpoint generalization. The goal is to identify the angular range within which performance remains stable under the Extreme setting defined in Table 1.

6.1 Experimental setup

We perform a breaking point analysis under the Extreme setting from Experiment A, where only a single reference bin ($0°$) is available in the memory and models are evaluated across the remaining bins. We define the *breaking point* as the earliest validation bin where a model's normalized performance drops significantly compared to the previous bin. Specifically, we compute

$$\Delta_i = \text{norm mIoU}_i - \text{norm mIoU}_{i-1} \tag{2}$$

$$\text{norm mIoU}_i = \frac{\text{mIoU}_i}{\text{mIoU}_{0°}}, \tag{3}$$

where the normalization by the $0°$ bin ensures that performance drops reflect viewpoint sensitivity rather than overall model scale.

A breaking point is recorded at bin i if $\Delta_i \leq -0.1$, indicating a relative drop of more than 10%. This analysis highlights each model's resistance to viewpoint shifts and identifies the angular range within which performance remains stable. We also report the normalized mIoU degradation curves as the viewpoint angle increases. The baseline that is used is an evaluation of the same setting, but with the validation bins being part of the memory.

6.2 Results

As illustrated in Figure 3, most models show a gradual decline in normalized mIoU as the validation angle increases. C-RADIOv2 and TIPS are the models that reach a breaking point, both at the $30°$

bin (see Table 3). TIPS shows the steepest drop, with a relative decrease of -0.125 between $15°$ and $30°$. By contrast, DINO, DINOv2, CLIP, and SigLIP2 degrade smoothly without reaching the breaking point.

Table 3: **Breaking points.** For each model in Experiment B, we report the validation bin where a breaking point occurs together with the corresponding normalized mIoU drop ($\Delta_i \leq -0.1$) relative to the previous bin.

Model	Breaking Point Bin	Δ (Drop)
CLIP ViT-B/16	None	–
DINO ViT-B/16	None	–
DINOv2 ViT-B/14	None	–
C-RADIOv2 ViT-B/16-CPE	30	-0.1049
SigLIP2 B/16-512	None	–
TIPS ViT-B/14-HR	30	-0.1250

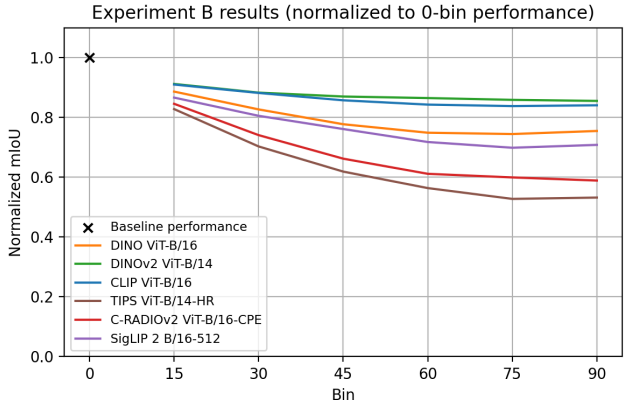

Figure 3: **Normalized mIoU.** We show normalized mIoU curves (Δ_i) per model in Experiment B, measured relative to the 0-bin performance.

6.3 Discussion

The breaking point analysis highlights differences in robustness across models and shows how their pretraining objectives affect stability under viewpoint shifts. DINOv2 is the most stable, maintaining gradual degradation without sudden drops. DINO shows a similar trend, though with slightly weaker overall performance. CLIP and SigLIP2 also decrease steadily, suggesting that they preserve some resilience despite being trained for image–text alignment rather than spatial structure.

In contrast, TIPS and C-RADIOv2 exhibit clear breakdowns, indicating that their learned features are less robust when faced with unseen viewpoints. For TIPS, this instability likely stems from its focus on semantic alignment rather than spatial consistency, while C-RADIOv2's mixed supervision appears to weaken the reliability of part-level representations.

Relative to the baseline, where validation bins are also included in memory, these findings confirm that the sudden decline comes from missing reference views rather than instability in the evaluation pipeline. Overall, the results show that self-supervised encoders (DINO, DINOv2) produce more geometry-aware features under strong viewpoint shifts, while multimodal or distilled approaches are more brittle. DINOv2's stability makes it the most reliable choice for applications that must handle abrupt perspective changes.

7 Experiment C: memory size robustness

In this experiment, we analyze how memory bank size affects viewpoint generalization, comparing the performance gains across difficulty levels against the added computational cost.

Table 4: **Memory size comparison.** Viewpoint-based segmentation performance (mIoU) is reported for three memory sizes: (a) 320,000, (b) 640,000, and (c) 1,024,000. Bold values mark the best score per difficulty-memory configuration, and each cell shows mean ± standard deviation.

Model	Easy	Medium	Hard	Extreme
(a) Memory = 320,000				
CLIP ViT-B/16	0.729 ± 0.138	0.725 ± 0.141	0.710 ± 0.149	0.681 ± 0.156
DINO ViT-B/16	**0.741 ± 0.147**	**0.736 ± 0.150**	0.712 ± 0.163	0.656 ± 0.178
DINOv2 ViT-B/14	0.738 ± 0.145	**0.736 ± 0.146**	**0.726 ± 0.152**	**0.709 ± 0.158**
C-RADIOv2 ViT-B/16-CPE	0.588 ± 0.136	0.579 ± 0.139	0.539 ± 0.146	0.467 ± 0.149
SigLIP 2 B/16-512	0.506 ± 0.154	0.501 ± 0.156	0.483 ± 0.157	0.443 ± 0.149
TIPS ViT-B/14-HR	0.600 ± 0.161	0.590 ± 0.166	0.539 ± 0.178	0.437 ± 0.175
(b) Memory = 640,000				
CLIP ViT-B/16	0.745 ± 0.132	0.740 ± 0.136	0.727 ± 0.141	0.694 ± 0.150
DINO ViT-B/16	**0.768 ± 0.137**	**0.761 ± 0.141**	0.736 ± 0.156	0.676 ± 0.173
DINOv2 ViT-B/14	0.754 ± 0.138	0.750 ± 0.142	**0.740 ± 0.147**	**0.721 ± 0.155**
C-RADIOv2 ViT-B/16-CPE	0.629 ± 0.132	0.615 ± 0.135	0.572 ± 0.142	0.491 ± 0.150
SigLIP 2 B/16-512	0.541 ± 0.152	0.531 ± 0.154	0.511 ± 0.157	0.466 ± 0.149
TIPS ViT-B/14-HR	0.644 ± 0.144	0.628 ± 0.153	0.571 ± 0.167	0.453 ± 0.170
(c) Memory = 1,024,000				
CLIP ViT-B/16	0.755 ± 0.130	0.748 ± 0.134	0.734 ± 0.140	0.701 ± 0.149
DINO ViT-B/16	**0.782 ± 0.132**	**0.774 ± 0.137**	**0.748 ± 0.153**	0.686 ± 0.171
DINOv2 ViT-B/14	0.763 ± 0.136	0.758 ± 0.139	**0.748 ± 0.143**	**0.728 ± 0.154**
C-RADIOv2 ViT-B/16-CPE	0.653 ± 0.129	0.636 ± 0.132	0.592 ± 0.141	0.506 ± 0.150
SigLIP 2 B/16-512	0.564 ± 0.150	0.551 ± 0.153	0.530 ± 0.157	0.481 ± 0.152
TIPS ViT-B/14-HR	0.667 ± 0.137	0.647 ± 0.146	0.588 ± 0.162	0.462 ± 0.169

7.1 Experimental setup

We follow the Hummingbird paper [1] and vary the number of entries in the memory bank. Three configurations are evaluated: 320k, 640k, and 1,024k. For each configuration we report segmentation accuracy and measure computational cost, including runtime and resource usage.

7.2 Results

The results in Table 4 show that increasing memory generally improves mIoU, with the largest gains observed under the Easy difficulty. Doubling memory from 320k to 640k increases performance by about 0.030 mIoU on Easy, while a further increase to 1,024k yields a smaller improvement of 0.017 (see Table 9).

DINO and DINOv2 achieve the highest absolute performance across all memory sizes. However, models such as C-RADIOv2, SigLIP2, and TIPS show the greatest relative gains. Each improves by more than 0.048 when moving from 320k to 1,024k, with most of that improvement coming from the first doubling.

The runtime analysis in Table 10 shows that larger memory banks increase computational cost. The highest configuration adds about two hours per job. Total memory usage remains similar (189–214GB), but CPU efficiency declines as memory grows: for example, C-RADIOv2 drops from 16.35% at 320k to 12.52% at 1,024k.

7.3 Discussion

These results confirm that larger memory banks improve generalization, but the benefits vary across models. Less robust encoders such as C-RADIOv2, SigLIP2, and TIPS gain the most from additional memory, while DINO and DINOv2 already perform strongly even with smaller banks. The diminishing gains beyond 640k and the higher computational cost suggest that the intermediate configuration offers the best trade-off. This shows that memory scaling can compensate for weaker representations, but stable self-supervised features remain more effective overall.

8 Qualitative analysis of segmentations

In addition to quantitative evaluations, we conducted a qualitative inspection of predicted segmentation masks to better understand model behaviour under the large-angle changes of the Extreme difficulty.

Localized shape recovery. As shown in Figure 27, the model successfully segments the dinosaur figure from an unfamiliar viewpoint. While the prediction (center) is not a perfect pixel-wise match to the ground truth (left), the overall contour and structure of the object are preserved. The prediction predicts that that the object extends further (e.g., tail tip and scattered patches), and the core body is segmented as a bigger area (encompassing more of the bony plates embedded in the back of the dinosaur).

Prediction versus ground truth overlays. In Figure 28, we visualize the predicted mask (center) versus the ground truth (left) when overlaid on the input image. These overlays reveal that the model tends to underestimate object boundaries in highly self-occluded regions or under harsh lighting (e.g., dark shadows like those under the dinosaur's core). Nonetheless, the predictions largely align with visually salient contours of the object, indicating strong local feature retrieval. When looking at the overlay of the ground truth over the prediction (right), we notice that the prediction actually covers more of the core and parts of the dinosaur's tail, an improvement as compared to the ground truth.

Prediction versus ground truth. We also visualize mask comparisons using color-coded difference maps (Figure 28 (right) or Figure 29). These maps highlight regions of over- and under-segmentation. Notably, misalignments are often located near object boundaries or thin structures (e.g., legs, tails), suggesting that memory retrieval may struggle with fine-grained spatial resolution under Extreme difficulty views. In addition, dark shadows are often included as parts of the segmentation mask. In some cases, the ground truth masks themselves are of poor quality, as illustrated in Figures 8 and 20–25 for the bed class. This leads to decreased scores even when the prediction masks are visually more accurate. Such cases highlight that annotation quality can limit the reliability of quantitative evaluation, since metrics are tied to imperfect labels rather than perceptual correctness.

Failures. In Figure 29, we show a failure case on a different *toy dragon* instance. Here, the predicted segmentation extends beyond the actual object boundary, indicating that the model retrieved visually similar but geometrically inconsistent patches. Segmentation failures such as this one are causing certain models (e.g., C-RADIOv2, TIPS) to exhibit performance drops beyond 30° angular deviation.

9 Conclusion

We studied the ability of frozen ViT encoders to generalize across unseen camera viewpoints in an in-context segmentation setting, using the Hummingbird architecture and MVImgNet dataset. Our benchmark covered three aspects: controlled evaluation under view shifts, robustness through breaking point analysis, and the effect of memory bank size.

Our findings show that self-supervised encoders, particularly DINO and DINOv2, provide more stable geometry-aware features under viewpoint changes, while multimodal and distilled approaches are more brittle. Larger memory banks help weaker models but offer diminishing returns for stronger encoders, raising questions about the cost–benefit trade-off.

Taken together, these results highlight the strengths and limitations of current ViTs for multi-view perception. While modern self-supervised training yields robust features, none of the models are immune to viewpoint-induced degradation. Addressing this gap will likely require objectives or architectures that enforce 3D consistency. Future work could include extending this evaluation to multi-object scenes, wider angular ranges, and compound rotations, providing a broader testbed for building truly viewpoint-robust representations.

References

[1] Ivana Balažević, David Steiner, Nikhil Parthasarathy, Relja Arandjelović, and Olivier J. Hénaff. Towards in-context scene understanding, 2023. URL https://arxiv.org/abs/2306.01667.

[2] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022. URL https://arxiv.org/abs/2108.07258.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. URL https://arxiv.org/abs/2104.14294.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakoba Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2020.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

[6] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. URL https://arxiv.org/abs/2111.06377.

[7] Greg Heinrich, Mike Ranzinger, Hongxu, Yin, Yao Lu, Jan Kautz, Andrew Tao, Bryan Catanzaro, and Pavlo Molchanov. Radiov2.5: Improved baselines for agglomerative vision foundation models, 2025. URL https://arxiv.org/abs/2412.07679.

[8] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. URL https://arxiv.org/abs/2103.14030.

[9] Kevis-Kokitsi Maninis, Kaifeng Chen, Soham Ghosh, Arjun Karpur, Koert Chen, Ye Xia, Bingyi Cao, Daniel Salz, Guangxing Han, Jan Dlabal, Dan Gnanapragasam, Mojtaba Seyedhosseini, Howard Zhou, and André Araujo. TIPS: Text-Image Pretraining with Spatial Awareness. In *ICLR*, 2025.

[10] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. URL https://arxiv.org/abs/2003.08934.

[11] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester

Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL `https://arxiv.org/abs/2303.08774`.

[12] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL `https://arxiv.org/abs/2304.07193`.

[13] 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, 2021. URL `https://arxiv.org/abs/2103.00020`.

[14] Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model – reduce all domains into one, 2024. URL `https://arxiv.org/abs/2312.06709`.

[15] Shouwei Ruan, Yinpeng Dong, Hang Su, Jianteng Peng, Ning Chen, and Xingxing Wei. Towards viewpoint-invariant visual recognition via adversarial training, 2023. URL `https://arxiv.org/abs/2307.10235`.

[16] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] Johannes Lutz Schönberger, True Price, Torsten Sattler, Jan-Michael Frahm, and Marc Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision (ACCV)*, 2016.

[18] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[19] Ofir Shifman and Yair Weiss. Lost in translation: Modern neural networks still struggle with small realistic image transformations, 2024. URL https://arxiv.org/abs/2404.07153.

[20] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

[21] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. URL https://arxiv.org/abs/2502.14786.

[22] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference, 2022. URL https://arxiv.org/abs/2111.02080.

[23] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[24] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Mvimgnet: A large-scale dataset of multi-view images, 2023. URL https://arxiv.org/abs/2303.06042.

A MVImgNet dataset

The statistics for the reorganized subset of MVImgNet [24] we used are shown in Table 5. The selected subset includes 15 object classes, each represented across 7 standardized angle bins: $[0°, 15°, 30°, 45°, 60°, 75°, 90°]$. Figure 1 shows an instance of each angle for each object class.

Table 5: **Angle selection accuracy per class.** Mean and standard deviation of angular error (in degrees) are reported for each object class after selecting views closest to the predefined angle bins.

Class Number	Category	Std. Error	Mean Error	# Images in each Angle bin
7	Stove	1.28	0.01	197
8	Sofa	1.15	-0.04	91
19	Microwave	1.44	-0.04	120
46	Bed	1.17	-0.00	23
57	Toy Cat	1.96	-0.01	783
60	Toy Cow	1.96	-0.02	735
70	Toy Dragon	1.62	0.01	627
99	Coat Rack	1.41	-0.02	97
113	Ceiling Lamp	1.64	-0.03	154
125	Toilet	1.31	0.02	58
126	Sink	1.20	-0.12	30
152	Strings	1.25	0.03	192
166	Broccoli	2.04	-0.03	210
196	Durian	1.65	0.03	758
100	Guitar Stand	1.53	0.04	218

B Model specifics

The following table details the ViT-based encoders used in our experiments, including their patch configurations and feature dimensionality.

Table 6: **Vision Transformer configurations.** We show the architectural and input settings of the models evaluated in our experiments.

Model	Architecture	Input Size	Patch Size	Patch Count	Feature Dim.	Batch Size
DINO	ViT-B/16	512	16	1024	768	4
DINOv2	ViT-B/14	504	14	1296	768	4
OpenAI CLIP	ViT-B/16	512	16	1024	768	4
C-RADIOv2	ViT-B/16-CPE	512	16	1024	768	4
SigLIP2	ViT-B/16-512	512	16	1024	768	4
TIPS	ViT-B/14-HR	504	14	1296	768	4

C Licenses for assets

All datasets and models used in this work are released under open licenses. We list them here for clarity:

Table 7: **Licenses for datasets and models.** All assets are used in compliance with their respective licenses.

Asset	License
MVImgNet	CC BY-NC 4.0 (code), dataset Terms of Use
Hummingbird	MIT License
CLIP	MIT
DINO	Apache 2.0
DINOv2	Apache 2.0
SigLIP2	Apache 2.0
C-RADIOv2	NVIDIA Open Model License
TIPS	Apache 2.0 (code), CC BY 4.0 (docs)

D Additional results

D.1 Reproduction

Table 8: **Reproduction results.** We report accuracy (%) at different evaluation scales. "Reported" indicates values taken from prior published results, while "Reproduced" refers to our reproduction with a batch size (BS). Bold values highlight the best result for each model-scale configuration, with reproduced results differing by less than 3%.

Model	Source	$\mathbf{1024x10^2}$	$\mathbf{1024x10^3}$	$\mathbf{1024x10^4}$
ViT-S/16	Reproduced, BS: 256	37.5	45.0	–
	Reproduced, BS: 64	**37.9**	**45.1**	**49.3**
	Reported, BS: 64	37.2	43.1	46.6
ViT-B/16	Reproduced, BS: 64	**48.0**	**54.7**	–
	Reproduced, BS: 32	47.8	54.6	–
	Reproduced, BS: 8	–	–	**57.9**
	Reported, BS: 64	44.9	50.8	55.7
ViT-S/14	Reproduced, BS: 64	69.6	**75.1**	**77.0**
	Reported, BS: 64	**70.2**	74.9	77.0
ViT-B/14	Reproduced, BS: 64	68.0	74.0	–
	Reproduced, BS: 8	–	–	**76.6**
	Reported, BS: 64	**69.1**	**74.6**	76.9
ViT-L/14	Reproduced, BS: 64	64.1	71.2	–
	Reproduced, BS: 8	–	–	74.4
	Reported, BS: 64	**64.6**	**71.7**	**74.8**
ViT-G/14	Reproduced, BS: 32	**62.4**	–	–
	Reproduced, BS: 16	–	**70.1**	–
	Reproduced, BS: 8	–	–	73.3
	Reported, BS: 64	62.3	69.9	**73.6**

D.2 Experiment A

In Experiment A, we assess model generalization across view-angle difficulties using per-class performance curves. The background class performs best, while bed, coat rack, guitar stand, and sink perform worst. For beds, the drop is explained by poor ground-truth annotations that likely reduce model scores.

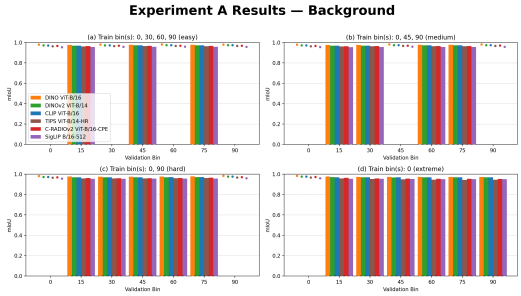

Figure 4: **Experiment A: background.** Viewpoint generalization for the class *background*, with four training difficulty settings: (a) easy; (b) medium; (c) hard; and (d) extreme.

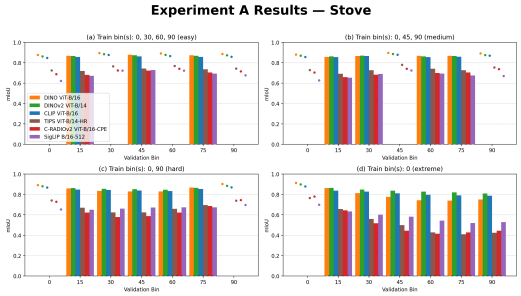

Figure 5: **Experiment A: stove.** SigLIP2 does relatively better than the average performance in Figure 2

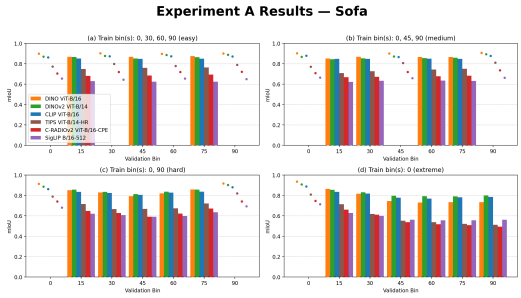

Figure 6: **Sofa.** Performance trends mirror the general degradation observed in Figure 2, with accuracy dropping consistently across difficulty settings.

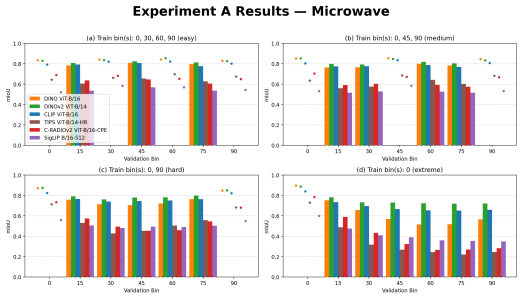

Figure 7: **Microwave.** TIPS appears to struggle when segmenting the microwave. Performs poorer than in Figure 2

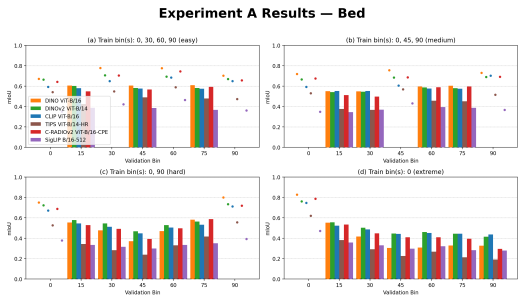

Figure 8: **Bed.** Beds have bad ground truths (seen in Figures 20–25 of Appendix D.2.1), resulting in a drop in overall performance

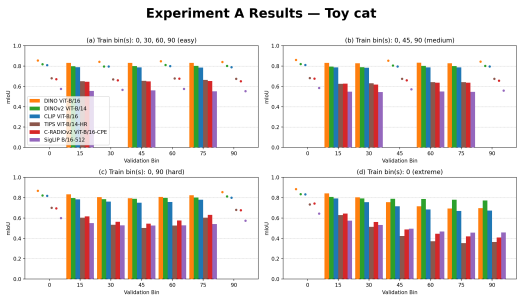

Figure 9: **Cat.** Performance trends mirror the general degradation observed in Figure 2, with accuracy dropping consistently across difficulty settings.

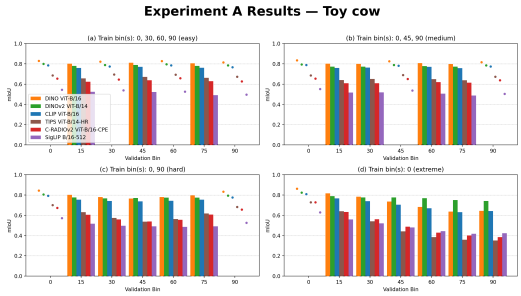

Figure 10: **Cow.** Performance trends mirror the general degradation observed in Figure 2, with accuracy dropping consistently across difficulty settings.

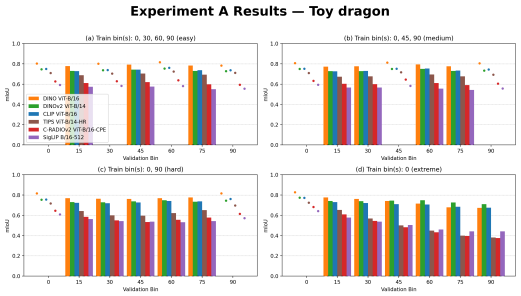

Figure 11: **Toy dragon.** Performance trends mirror the general degradation observed in Figure 2, with accuracy dropping consistently across difficulty settings.

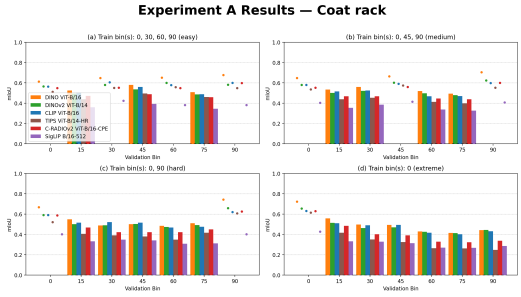

Figure 12: **Coat rack.** Performance degradation is observed, but its cause remains unclear and is a point for future investigation.

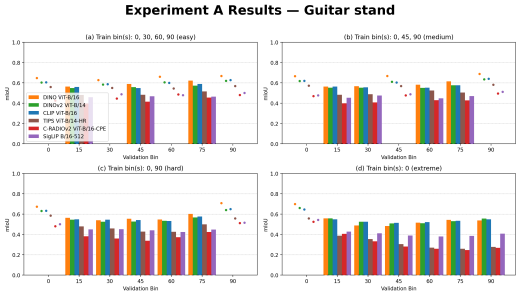

Figure 13: **Guitar stand.** Performance degradation is observed, but its cause remains unclear and is a point for future investigation.

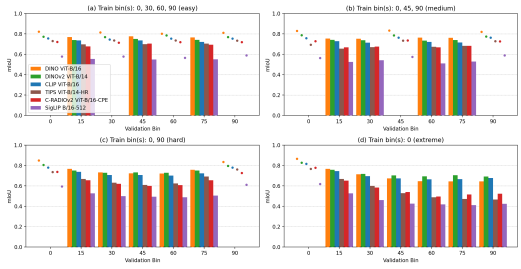

Figure 14: **Ceiling lamp.** Performance trends mirror the general degradation observed in Figure 2, with accuracy dropping consistently across difficulty settings.

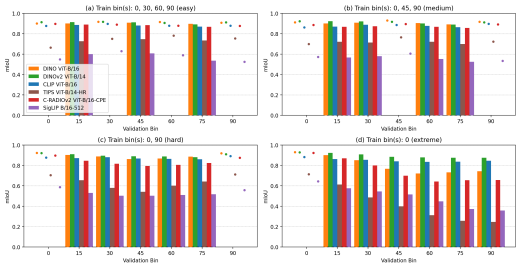

Figure 15: **Toilet.** C-Radio does significantly better on toilets. Resulting in a greater performance compared to Figure 2

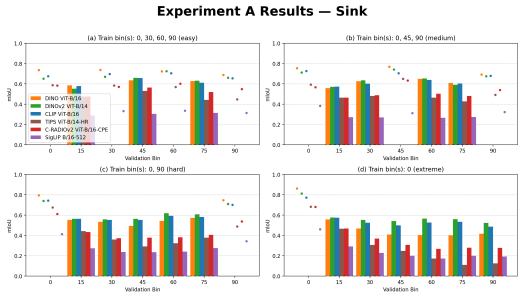

Figure 16: **Sink.** Performance degradation is observed, but its cause remains unclear and is a point for future investigation.

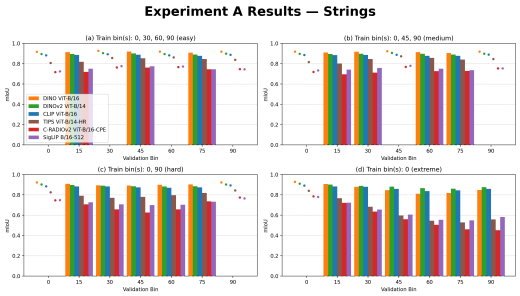

Figure 17: **Strings.** Performance trends mirror the general degradation observed in Figure 2, with accuracy dropping consistently across difficulty settings.

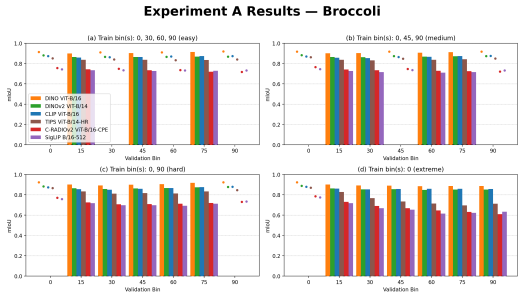

Figure 18: **Broccoli.** Performance trends mirror the general degradation observed in Figure 2, with accuracy dropping consistently across difficulty settings.

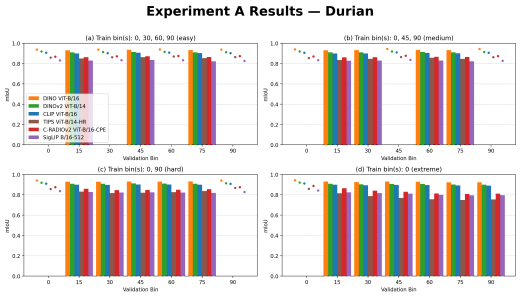

Figure 19: **Durian.** Performance trends mirror the general degradation observed in Figure 2, with accuracy dropping consistently across difficulty settings.

D.2.1 Bed category ground truth analysis

We investigated the poor performance of the 'bed' class (as seen in Figure 8) and identified inconsistent or incomplete ground truth annotations as a likely cause.

Figure 20: **Bed image example 1.** Original image used for annotation.

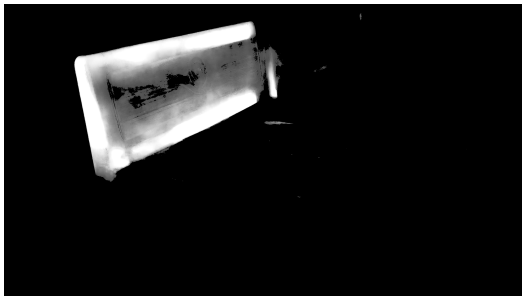

Figure 21: **Incomplete annotation.** Ground truth excludes parts of the bed (side railing, footboard) and ignores surrounding beds.

Figure 22: **Bed image example 2.** Original image used for annotation.

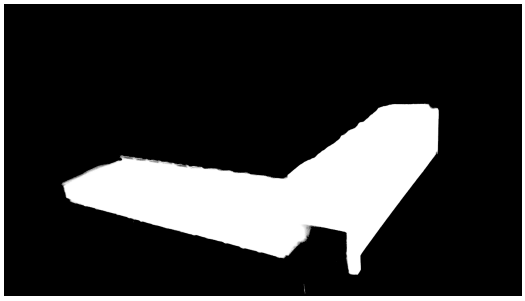

Figure 23: **Erroneous annotation.** The ground truth excludes large parts of the bed (headboard, mattress) and incorrectly includes the shadow between two beds.

Figure 24: **Bed image example 3.** Original image used for annotation.

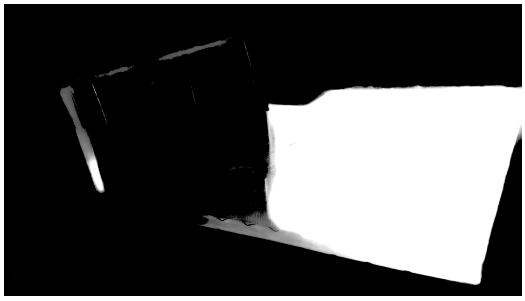

Figure 25: **Annotation error.** The ground truth omits the surrounding beds, incorrectly includes the gap between the main bed and the bed to its right, and only partially marks the headboard (side only).

D.3 Experiment B

As shown in Figure 3 and Figure 26, most models exhibit a gradual performance decline as the distance of the validation bin from the training bin ($0°$) increases. Notably, only C-RADIOv2 (ViT-B/16-CPE) and TIPS (ViT-B/14-HR) exhibit sharp performance drops with increasing validation angles, indicating limited generalization capacity under extreme viewpoint shifts.

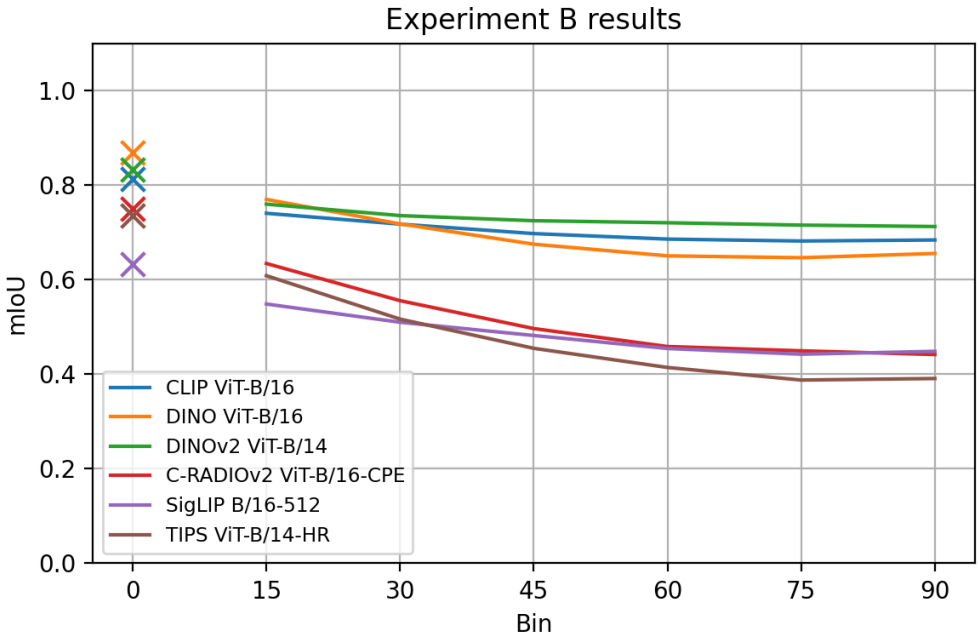

Figure 26: **Absolute mIoU.** We report absolute mIoU scores per validation bin. Trends follow the normalized plot, with DINOv2 showing the most consistent generalization across bins.

D.4 Experiment C

In Experiment C, we evaluate the impact of memory bank size on model performance. Table 9 reports the absolute mIoU gains per difficulty level when increasing the memory bank for enteries (a) from 320,000 to 640,000, (b) from 640,000 to 1,024,000, and (c) from 320,000 to 1,024,000. This highlights how memory sensitivity varies across model architectures. On the other hand, Table 10 outlines the computational trade-offs. We report the runtime and memory usage associated with each model under both memory configurations. This provides context for the computational cost underlying the performance results.

Table 9: **Performance gains from memory.** We report absolute mIoU improvements when increasing memory from (a) 320,000 to 640,000, (b) 640,000 to 1,024,000, and (c) 320,000 to 1,024,000. Values are computed as differences from Table 4, with the final column showing the average gain across all four difficulty levels.

Model	Gain (Easy)	Gain (Medium)	Gain (Hard)	Gain (Extreme)	Average Gain
(a) Memory = 320,000 → 640,000					
CLIP ViT-B/16	0.016	0.015	0.017	0.013	0.01525
DINO ViT-B/16	0.027	0.025	0.024	0.020	0.02400
DINOv2 ViT-B/14	0.016	0.014	0.014	0.012	0.01400
C-RADIOv2 ViT-B/16-CPE	0.041	0.036	**0.033**	**0.024**	**0.03350**
SigLIP 2 B/16-512	0.035	0.030	0.028	0.023	0.02900
TIPS ViT-B/14-HR	**0.044**	**0.038**	0.032	0.016	0.03250
Average per task	**0.030**	0.026	0.0247	0.018	
(b) Memory = 640,000 → 1,024,000					
CLIP ViT-B/16	0.010	0.008	0.007	0.007	0.00800
DINO ViT-B/16	0.014	0.013	0.012	0.010	0.01225
DINOv2 ViT-B/14	0.009	0.008	0.008	0.007	0.00800
C-RADIOv2 ViT-B/16-CPE	**0.024**	**0.021**	**0.020**	**0.015**	**0.02000**
SigLIP 2 B/16-512	0.023	0.020	0.019	**0.015**	0.01925
TIPS ViT-B/14-HR	0.023	0.019	0.017	0.009	0.01700
Average per task	**0.017**	0.015	0.0138	0.0105	
(c) Memory = 320,000 → 1,024,000					
CLIP ViT-B/16	0.026	0.023	0.024	0.020	0.02325
DINO ViT-B/16	0.041	0.038	0.036	0.030	0.03625
DINOv2 ViT-B/14	0.025	0.022	0.022	0.019	0.02200
C-RADIOv2 ViT-B/16-CPE	0.065	**0.057**	**0.053**	**0.039**	**0.05350**
SigLIP 2 B/16-512	0.058	0.050	0.047	0.038	0.04825
TIPS ViT-B/14-HR	**0.067**	**0.057**	0.049	0.025	0.04950
Average per task	**0.047**	0.041	0.0385	0.0285	

Table 10: **System performance under memory scaling.** We report wall-clock time, memory usage, CPU utilization, CPU efficiency, and memory efficiency for viewpoint-based segmentation with memory sizes (a) 320,000, (b) 640,000, and (c) 1,024,000. Some entries are marked as `(running)` or show `0.00` usage values; these correspond to cases where monitoring logs did not update correctly, even though the jobs completed successfully. Bold values highlight the largest drops in CPU efficiency for each model.

Model	Wall-clock Time	CPU Used	CPU Eff.	Memory Used	Mem. Eff.
(a) Memory = 320,000					
CLIP ViT-B/16	07:32:14	3-15:33:05	16.13%	198.14 GB	41.28%
DINO ViT-B/16	07:32:35	3-12:20:38	15.53%	214.11 GB	44.61%
DINOv2 ViT-B/14	08:39:50 (running)	00:00:00	0.00%	0.00 GB	0.00%
C-RADIOv2 ViT-B/16-CPE	07:27:39	3-15:50:23	16.35%	199.82 GB	41.63%
SigLIP 2 B/16-512	07:39:28	3-19:58:28	16.68%	189.89 GB	39.56%
TIPS ViT-B/14-HR	08:33:54 (running)	00:00:00	0.00%	0.00 GB	0.00%
(b) Memory = 640,000					
CLIP ViT-B/16	08:21:00	3-10:39:36	13.75%	177.58 GB	37.00%
DINO ViT-B/16	08:28:26 (running)	00:00:00	0.00%	0.00 GB	0.00%
DINOv2 ViT-B/14	09:43:52 (running)	00:00:00	0.00%	0.00 GB	0.00%
C-RADIOv2 ViT-B/16-CPE	08:18:48	3-11:02:33	13.87%	203.63 GB	42.42%
SigLIP 2 B/16-512	08:26:16	3-16:01:55	14.49%	181.39 GB	37.79%
TIPS ViT-B/14-HR	09:33:10 (running)	3-09:16:27	11.82%	209.87 GB	43.72%
(c) Memory = 1,024,000					
CLIP ViT-B/16	09:20:31	3-07:03:04	11.75%	207.41 GB	43.21%
DINO ViT-B/16	09:29:42	3-07:07:59	11.58%	213.23 GB	44.42%
DINOv2 ViT-B/14	11:10:37	3-11:36:03	10.39%	209.23 GB	43.59%
C-RADIOv2 ViT-B/16-CPE	09:28:53	3-13:29:38	12.52%	200.34 GB	41.74%
SigLIP 2 B/16-512	09:36:48 (running)	00:00:00	0.00%	0.00 GB	0.00%
TIPS ViT-B/14-HR	11:04:09	3-11:57:18	10.53%	211.38 GB	44.04%

D.5 Qualitative analysis

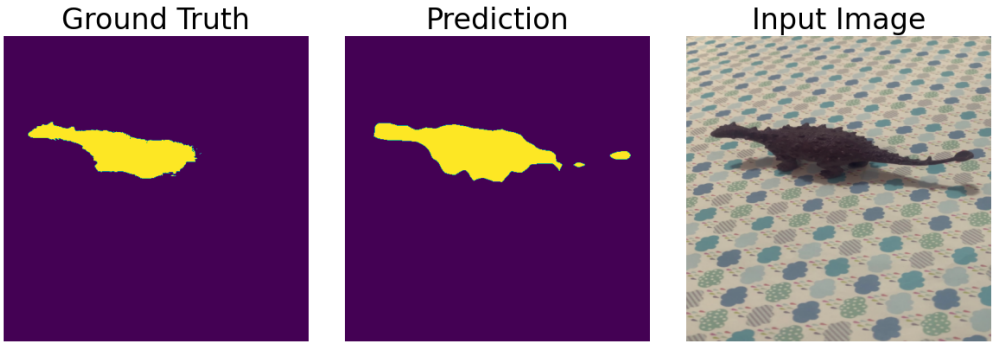

Figure 27: **Qualitative segmentation.** Results shown using DINO. Left: input image. Center: predicted mask. Right: ground truth mask. Interestingly, the prediction of the *toy dragon* aligns more closely with visible object boundaries than the ground truth, which appears coarser and less accurate at fine details (e.g., tail, horns, stomach occlusion).

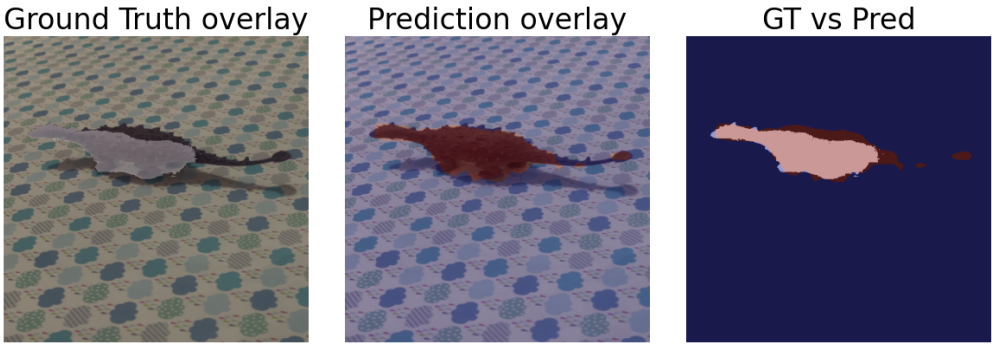

Figure 28: **Overlay comparison of ground truth and prediction.** The same instance as in Figure 27 is shown for better. Left: ground truth overlaid on the input image. Center: prediction overlaid on the input image. Right: difference visualization, with the ground truth overlaid on the predicted mask.

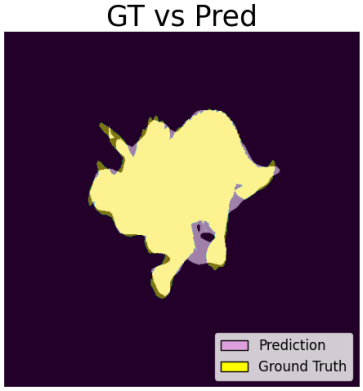

Figure 29: **Another overlay comparison of ground truth and prediction.** Shown using DINO. Yellow: ground truth mask. Purple: predicted mask. We note that discrepancies appear at fine-grained boundaries (e.g., tail, horns, nose) and in occluded or shadowed regions (e.g., under the stomach).

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]

   Guidelines:

   - The answer NA 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 NA 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]

   Guidelines:

   - The answer NA 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: [NA]

   Guidelines:

   - The answer NA 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]

   Guidelines:

   - The answer NA 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]

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Guidelines:

   - The answer NA 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]

   Guidelines:

   - The answer NA 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]

   Guidelines:

   - The answer NA 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]

   Guidelines:

   - The answer NA 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: [NA]

    Justification: Our work focuses on methodological evaluation of viewpoint generalization in vision encoders. As such, it does not, on its own, have direct positive or negative societal impacts. Any downstream societal considerations would depend on the specific deployment contexts of such encoders, which are outside the scope of this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Guidelines:

    - The answer NA 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]

    Guidelines:

    - The answer NA 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]

    Guidelines:

    - The answer NA 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: [NA]

    Guidelines:

    - The answer NA 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: [NA]

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.