GECO: Geometrically Consistent Embedding with Lightspeed Inference

Supplementary Material

Contents

7. Implementation Details

7.1. Architecture Details

Feature Encoder Hyperparameters We identified notable differences in the optimal settings for DINOv2 and DINO that are not explicitly documented in their original papers. For DINOv2, we achieve the best performance by using patch tokens from the final transformer layer, particularly when working with higher-resolution images. In contrast, DINO performs better when using patch tokens from the fourth-to-last transformer layer, with optimal results at an input resolution of 244; performance degrades with higher resolutions.

Based on these findings, we use the DINOv2 backbone [42] with an input resolution of 518 and patch tokens from the final transformer layer in our model.

Architecture We learn low-rank matrices (rank 10) for each attention layer, which are added to the 768×768-dimensional weight matrices of the linear layers before being passed to the attention mechanism ($768 \times 10 \times 2$ parameters). For DINOv2-B, this results in $v = 768 \times 10 \times 2 \times 12 \times 2$ parameters, with a depth of 12 and updating query and value matrices equating $v \times 4$ bytes (≈ 1.4 MB) of learnable parameters. This is significantly fewer than the 19 MB of learnable parameters introduced by Geo [64]. This small parameter count and simple architecture enable us to increase the inference time of DINOv2-B by less than 1 millisecond.

7.2. Training Details

Training Hyperparameters The models are trained using the Adam optimizer with a learning rate of 0.0001, no weight decay, and no learning rate scheduler, with a batch size of 6. A key advantage of our approach is the ability to train directly on raw images without preprocessing, which allows for increased data augmentation and larger batch sizes. Training is conducted on a single GPU for 8 epochs. For the loss functions, weights are set as follows: 1 for the positive loss, 1 for the bin loss, and 10 for the negative loss.

Dataset Shuffeling Training is conducted jointly across the PFPascal, SPair [38], and APK [64] datasets. We employ a reweighting scheme that samples up to 800 examples per category. We found this hyperparameter by trading off the validation accuracies for the different datasets (see Sec. 7.4). In fact, if one dataset dominates the training, the model tends to overfit to that dataset.

Image Pair Augmentation We leverage existing datasets [22, 38, 55, 64] containing images of diverse category instances with varying shapes, textures, and motion deformations, captured under different lighting and camera setups. Thanks to our method's low computational cost, we use more data augmentation than [64] to improve generalization. We incorporate flipping, cropping, and color jitter into the augmentation pipeline, generating more positive, negative, and bin pairs for feature matching (see Fig. 2). Specifically, flipping creates challenging positive/negative pairs, while cropping increases bin pairs by promoting matches to the bin rather than to semantically similar but geometrically inconsistent features.

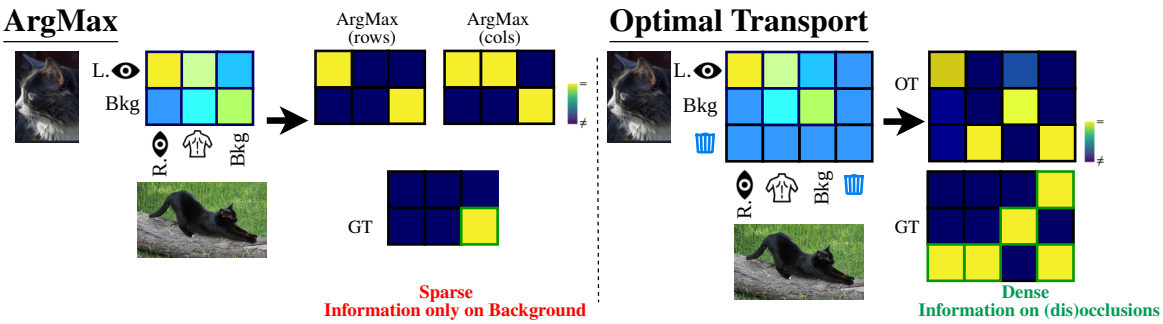


Figure 7. **Intuition on OT and geometry.** Standard ArgMax loss (left) provides positive supervision only for elements that appear in both images. Optimal Transport (right) uses the bin to provide explicit signals also on elements that appear in only one of the images. This is particularly common for non-rigid shapes (e.g., animals), as often a point and its symmetrical counterpart (e.g., left and right eyes) face (dis)occlusions. Intuitively, this leads to a stronger signal for learning features that better disambiguate symmetries.

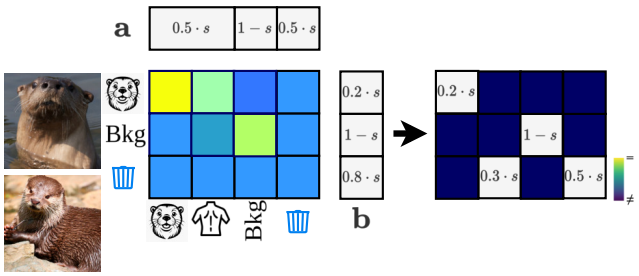


Figure 8. **Optimal Transport Marginals.** This figure illustrates the marginal distribution used in the Optimal Transport algorithm, assigning s to the shape and $1-s$ to the background. The visible content proportion is estimated from keypoint annotations by calculating the ratio of visible to total keypoints. In this example, the top image has a visibility ratio of 0.2 (only the head is visible), while the bottom image has a ratio of 0.5 (the head and one side of the body are visible) relative to the full shape.

	PFPascal [22]	APK [64]	Spair [38]	CUB [55]
GECO (Ours)-S	89.6	83.4	78.0	90.4
GECO (Ours)	**92.1**	**86.7**	**85.2**	**92.5**

Table 3. **Impact of model size on performance (PCK ↑).** Comparison of our approach using DINOv2-B and DINOv2-S backbones on PFPascall [22], CUB [55], SPair [38], and APK [64] datasets. The results indicate that model size plays a crucial role in overall performance.

7.3. Loss Function

Motivation for OT We argue that Optimal Transport (OT) provides stronger geometric supervision than Argmax, as shown in Fig. 7. As already pointed out in Sec. 3 there are several drawbacks when using Argmax to construct a loss function. Here, we highlight, that OT incorporates the supervision signal of points, visible in only one image, which often occurs with symmetric parts of non-rigid objects (e.g., eyes) by introducing a dustbin entry in the assignment matrix. Furthermore, due to the interaction between features, the OT loss enables to densely backpropagte the gradient. For example, a bin assignment signal pushes down the similarity to all other features, which is not possible with Argmax.

Designing marginal distributions Since correspondence data is valuable, a key contribution of our work is using it to construct marginal distributions see Fig. 8. To define the marginals for the Optimal Transport (OT) problem, we depart from the conventional approach, employed in SuperGlue [48], which assigns 0.9 to the image and 0.1 to the bin. Instead, we utilize automatically generated mask annotations to distinguish between foreground and background, leveraging an off-the-shelf segmentation tool [67].

In our formulation, the shape mass is assigned a value of $s = 0.9$, with the remaining $1-s$ allocated to the background, as illustrated in Fig. 8. The proportion of visible content, denoted as x, is estimated by calculating the ratio of visible keypoints to the total number of keypoints. This leads to $x \cdot s$ being assigned to the foreground and $(1-x) \cdot s$ to the bin.

This marginal distribution, together with the padded cosine similarity matrix of the features, is then fed into the Optimal Transport algorithm to determine the optimal transport plan.

7.4. Ablation Study on Design Choices

Impact of model size on performance (PCK) When scaling down the architecture to DINOv2-S, we observe a decline in performance, which we attribute to the reduced capacity of the model. While it still captures geometric relationships between keypoints, its performance does not match that of the larger variant, as shown in Tab. 3.

Impact of the Optimal Transport Marginals We evaluate the impact of the marginal distribution on the performance of our method. We compare the standard marginals used in SuperGlue [48] with our approach, which leverages mask annotations to distinguish between foreground

	PFPascal [22]	APK [64]	Spair [38]	CUB [55]
GECO (Ours)-N	91.5	86.0	83.7	89.5
GECO (Ours)	**92.1**	**86.7**	**85.2**	**92.5**

Table 4. **Impact of choice of marginals on performance (PCK ↑).** We evaluate the effect of incorporating more sophisticated marginals (Bottom row) in the loss function compared to the standard formulation (Top row). Our results show that while the performance difference on the test splits of APK, PFPascal, and SPair remains within a margin of 1.5, the performance drop on the generalization task is substantially larger, reaching a value of 3. This highlights the importance of incorporating improved marginals for better generalization.

	mean mIoU ↑	mean Acc ↑
GECO (Ours) (w/o KL)	36.4	88.5
GECO (Ours)	**37.9**	**89.0**

Table 5. **Impact of choice of marginals on performance (Segmentation Accuracy).** We evaluate using the segmentation metrics explained in Sec. 8.2. We show the effect of incorporating the KL-marginal regularization (Bottom row) in the loss function compared to the standard formulation with hard contraints on the marginal distributions (Top row).

and background. We found that our approach outperforms the standard marginals especially on the generalization to the CUB [55] dataset, as shown in Tab. 4.

Impact of the KL Divergence Regularization on the marginals The benefit of KL regularization is clear in the segmentation results, as shown in Tab. 5. Because the estimated marginals are imperfect, KL reduces overfitting to noise during training and should be increased when exact marginals are available (*e.g.*, images rendered from 3D shapes).

Impact of dataset shuffeling on the performance We evaluate the effect of dataset shuffling on performance across different datasets. As our method is trained jointly on SPair [38], PFPascal [22], and APK [64], shuffling enables better generalization. To address category imbalance, we sample up to K image pairs per category; if fewer are available, all are used.

Performance is highly sensitive to the choice of K. Low values (e.g., 100-400) lead to overfitting on PFPascal and poor generalization to SPair and CUB. Conversely, high K underrepresents PFPascal, limiting its validation accuracy. We find that $K = 800$ yields the best overall performance.

8. Additional Experiments

We provide more detailled experiments to analyze the PCK metric and its subsets in Sec. 8.1. In Sec. 5.2.1, we complement the qualitative analysis of the segmentation performance in the main paper with a quantitative and qualitative evaluation of the segmentation confusion matrix. In Sec. 8.3, we provide an additional experiment focusing on rigid shapes with diverse appearances and demonstrate our methods performance on keypoint matching by evaluating 2D-3D projections.

8.1. Analysis with (and of) PCK

Influence of different radii on unambiguous True Positives A critical choice of PCK is the radius in which a prediction is considered correct. This also has an impact on our PGCK subdivision. Here, we investigate how the choice of the PCK metric's radius influences the unambiguous true positive subset's cardinality (see Fig. 9). As expected, the unambiguous correct predictions decrease with the radius, while the overall PCK increases (see Fig. 10). For larger radius values, the PCK metric reflects the model's ability to identify general correspondences across the image, even when semantic parts are widely distributed rather than concentrated in a single area. In contrast, for smaller radius values, the metric emphasizes the model's precision in pinpointing correspondences with high spatial accuracy and its ability to distinguish between closely spaced keypoints, such as those on the left and right sides, when both are present.

Results In the main paper, we reported the subsets of the PGCK for one value of Fig. 9 on APK [64]. Here, we extend this analysis to the CUB [55] and SPair [38] datasets. As shown in Fig. 9a, our method achieves a slight improvement over previous work in unambiguous correct predictions on the CUB [55] dataset. For the APK [64] and SPair [38] datasets, performance varies: our method outperforms the competitor in one case, while in the other, the competitor demonstrates a more refined geometric understanding. These results suggest that geometric understanding is dataset-dependent.

Additionally, given the performance drop on CUB observed in competing methods on the semantic understanding task (see Fig. 10c), we argue that our approach strikes a favorable balance between geometric and semantic understanding. Unlike the competitors, our method preserves previously learned properties, making it a more robust and well-rounded solution. Moreover, it remains competitive with the state of the art while significantly reducing memory consumption and runtime from 2274 ms to 43 ms.

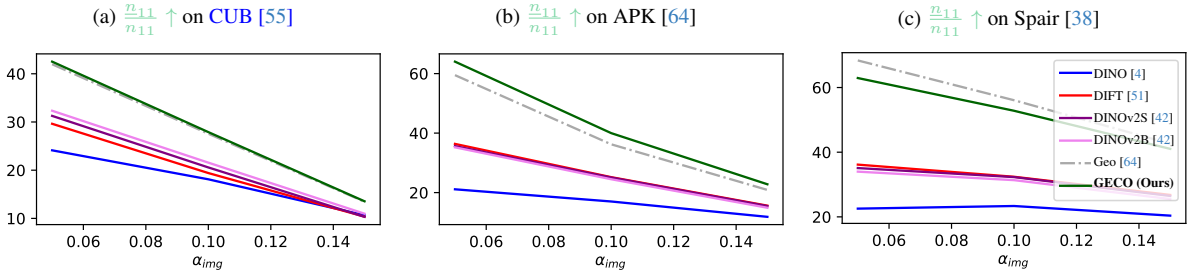


Figure 9. **Geometric ambiguity as a function of radius.** As the radius decreases, the set of unambiguous true positives $\frac{n_{11}}{n_{11}}$ grows, where the target keypoint is outside the radius of any incorrect matches. This figure illustrates the PGCK subset, specifically the unambiguous correct predictions, for various radius values ($@\alpha_{img}$) on CUB [55], APK [64], and Spair [38]. Our method outperforms previous work on two out of three datasets while achieving a reduction in runtime and memory usage by two orders of magnitude.

Influence of different radii on PCK As expected the PCK values are increasing with varying radius. It measures the general semantic knowledge of features with a part also measuring the geometric understanding.

Results We present detailed results, evaluated on the PF-Pascal [22], APK [64], SPair [38], and CUB [55] datasets. Specifically, in Fig. 10 we show the PCK values for different radii across these datasets. Notably, our method significantly outperforms the current state-of-the-art on three out of the four datasets, highlighting the superior effectiveness of our approach. While the competitor Geo [64] achieves comparable performance to ours in geometric understanding on the generalization task to CUB [55] (see Fig. 9a), we observe significant catastrophic forgetting in the n_{10} and n_{1x} splits leading to worse overall results than DINOv2-S (see Fig. 10c). In contrast, our method maintains consistent performance across all splits and outperforms Geo [64] in three out of four cases, demonstrating its robustness and reliability.

Qualitative results We illustrate various failure modes of existing approaches and demonstrate how our method addresses these issues. In Fig. 11, we show that our method can correctly assign a keypoint to the bin even when the actual match is occluded. This ability is learned through bin and negative losses, which encourage the assignment to the bin rather than to the symmetric counterpart. When the match is occluded, the model becomes better calibrated, generating minimal attention on the target image, including the symmetric counterpart. This represents a failure mode that the PCK metric does not capture, as the symmetric counterpart is absent in the target image, and consequently, this keypoint pair is excluded from the PCK evaluation.

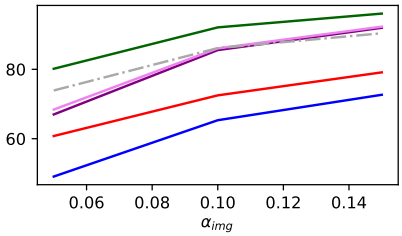
We also report performance on the geometrically relevant case, where multiple semantically similar keypoints are present in the target image. In this scenario, our method is more confident in the predictions, focusing only on the relevant areas of the image (see Fig. 12). As shown in Fig. 13, our features localize keypoints precisely in the target image without spreading similarity across irrelevant regions.

CUB [55]								
				Bir				
				11				
SPair [38]								
Aer	Bik	Bir	Boa	Bot	Bus	Car	Cat	Cha
34	53	23	30	97	43	40	82	81
Cow	Dog	Hor	Mot	Per	Pla	She	Tra	TV
80	46	61	69	34	73	72	90	62
APK [38]								
alo	ant	arg	bea	bis	bla	bob	bro	buf
37	31	28	21	20	32	36	33	34
cat	che	chi	cow	dee	dog	ele	fox	gir
42	30	37	38	36	36	36	33	24
gor	ham	hip	hor	jag	kin	leo	lio	mar
42	34	40	26	32	33	40	40	27
mon	moo	mou	noi	ott	pan	pan	pig	pol
37	24	20	38	28	23	46	35	30
rab	rac	rat	rhi	she	sku	sno	spi	squ
7	28	27	23	34	19	35	35	16
tig	uak	wea	wol	zeb				
32	39	36	34	24				

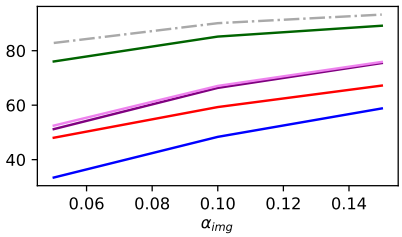
Table 6. **PGCK Dataset imbalance**. Completing Fig. 3, we report the geometric aware n_{11}/n, counts only the pairs for which a geometric mismatch is possible. Categories often present high imbalance, and geometrical error modes would have a different weight.

PGCK Dataset imbalance We analyze the cardinality of the geometrically aware subset in Fig. 3, which is able to assess the confusion between keypoints on opposite sides of the symmetry axis. To complete the table for all categories, we present a detailed report for the CUB [55], SPair [38], and APK [64] datasets in Tab. 6.

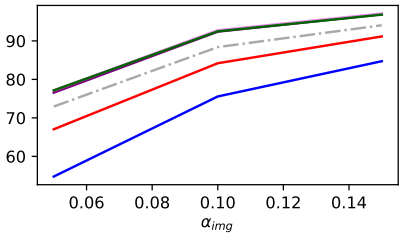
(a) PCK↑ on PFPascal [22]



(b) PCK↑ on Spair [38]



(c) PCK↑ on CUB [55]



(d) PCK↑ on APK [64]

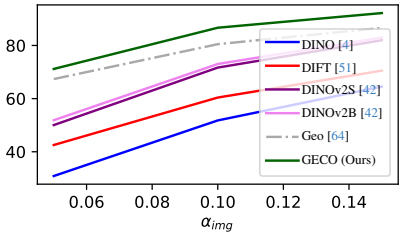


Figure 10. **PCK as a function of radius.** As the radius decreases, the set of correct matches declines. This figure illustrates the PCK metric for various radius values ($@\alpha_{img}$) on CUB [55], APK [64], and Spair [38]. Our method outperforms previous work on three out of four datasets while achieving a reduction in runtime and memory usage by two orders of magnitude.

Figure 11. **Qualitative results on the task of assignment to the bin (01-case) on APK [64] and CUB [55] (fifth row) .** We show that our method predicts small cosine similarities, if the actual match is occluded. This is learned by the bin and negative losses, which encourage the assignment to the bin instead of the symmetric counterpart. (First five rows) It is clearly visible, that the model knows the location of the ground truth correspondence, which could become visible by a small movement. The location of the symmetric counterpart is ignored. (Last three rows) We receive only low attention values for the whole image, including the symmetric counterpart, when the symmetric counterpart is occluded.

Figure 12. **Qualitative results on the task of assignment to the correct correspondence (11-case) on PFPascal [22] (First five rows), APK [64] (In the middle) and CUB [55] (Last row).** In the samples above the symmeric counterpart is visible in the target view. While previous work assigns attention on most of the image, our model is more confident in the predictions and only attends to the regions of interest.

Figure 13. **Qualitative results on the task of semantic correspondence estimation (10-case) on APK [64] and CUB [55] (Last row).** In the above samples only one semantically similar keypoint is visible in the target view. While previous work assigns attention on most of the image, our model is more confident in the predictions and only attends to the regions of interest.

8.2. Feature Space Segmentation

In Sec. 5.2, we revisit the procedure for evaluating the separability of the feature space by leveraging annotated parts and computing centroids for each. To complement the qualitative analysis, we now present a quantitative evaluation using standard segmentation metrics. Furthermore, we include normalized confusion matrices for another qualitative analysis to highlight the geometric consistency of the predicted segmentations.

Dataset We use the PascalParts [6] dataset, which offers detailed part annotations for a wide variety of object categories, enabling consistent and comprehensive evaluation across different classes. However, some of the 20 categories, such as *boat, table, and sofa*, have only a single part annotated. As a result, we exclude these categories from our evaluation.

Evaluation Similar to prior work [44][68] we use the **accuracy** and **mean Intersection over Union (mIoU)** as our primary evaluation metrics. To compute these metrics, let s_{ij} represent the number of patches from ground-truth part i that are predicted as part j by the model.

The mean Intersection over Union (mIoU) is defined as

$$\text{mIoU} = \frac{1}{N} \sum_i \frac{s_{ii}}{\sum_j s_{ij} + \sum_j s_{ji} - s_{ii}}, \tag{12}$$

where N is the number of parts. The accuracy is defined as:

$$\text{Acc} = \frac{\sum_i s_{ii}}{\sum_i \sum_j s_{ij}}. \tag{13}$$

This approach provides a per-category measure, enabling a detailed comparison of part segmentation performance across different parts. For an overall evaluation, we compute the average metric across all categories.

In addition, we report metrics specifically for the geometric subdivision, which includes parts divided into left and right counterparts, such as *legs, wings, eyes*. The geometric subdivision is presented both quantitatively and qualitatively through a confusion matrix. This focused analysis allows us to assess the model's ability to distinguish symmetric parts, which is critical for understanding its geometric reasoning capabilities.

Results In Tab. 7, we report the average mIoU and average accuracy across categories, evaluating performance on the segmentation task using the PascalParts [6] dataset.

First, evaluating all parts within each category on the left side of Tab. 7, offers insight into the model's capacity to differentiate between parts that are not necessarily symmetric, such as *head, body, and tail*. Although accuracy remains high across methods, mIoU scores decline due to sensitivity to class imbalance in the parts. Both fine-tuned geometry aware methods (last two rows) perform worse on non geometric parts, as expected. However, our method outperforms Geo [64], indicating better retention of non geometric information.

On the right side of Tab. 7, we report the metrics for the geometric parts only. As expected, both geometry-aware methods outperform all foundation models. The connection between the qualitative part segmentation results in Fig. 6 and the confusion matrix analysis is demonstrated in Fig. 14. Visual inspection aligns with the numbers and reveals that DINOv2-B struggles with symmetric distinctions (e.g., left vs. right). Additional examples of checkerboard patterns indicative of geometric confusion are shown in Fig. 15. Here, the matrices contain only geometrically confusable parts and are normalized to mitigate class imbalance.

	mean mIoU↑	mean Acc↑	mean mIoU↑ (geo)	mean Acc↑ (geo)
DINO [4]	31.5	87.9	44.1	95.9
DIFT$_{adm}$ [51]	31.5	87.9	48.0	96.4
DINOv2B [42]	**39.8**	**90.1**	57.6	97.3
Geo [64]	37.1	88.4	**62.8**	**97.6**
GECO (Ours)	<u>37.9</u>	<u>89.0</u>	**63.0**	**97.5**

Table 7. **Quantitative Evaluation of Segmentation Confusion Matrices on PascalParts [6].** Our features retain the segmentation performance of DINOv2 (two left columns), while additionally enabling the distinction between left and right parts (right two columns). The best scores are shown in **bold** and the second best are <u>underlined</u>. Methods with a performance difference of less than 0.5% are considered to be on par.

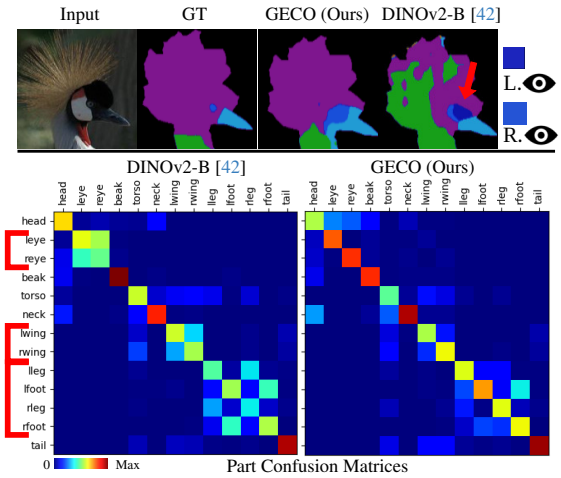


Figure 14. **Results on Symmetry. Top:** Our method reduces symmetric mismatches in part clustering. **Bottom:** We show the part confusion matrix (patches aggregated for each part across images), highlighting geometric confusion (red brackets).

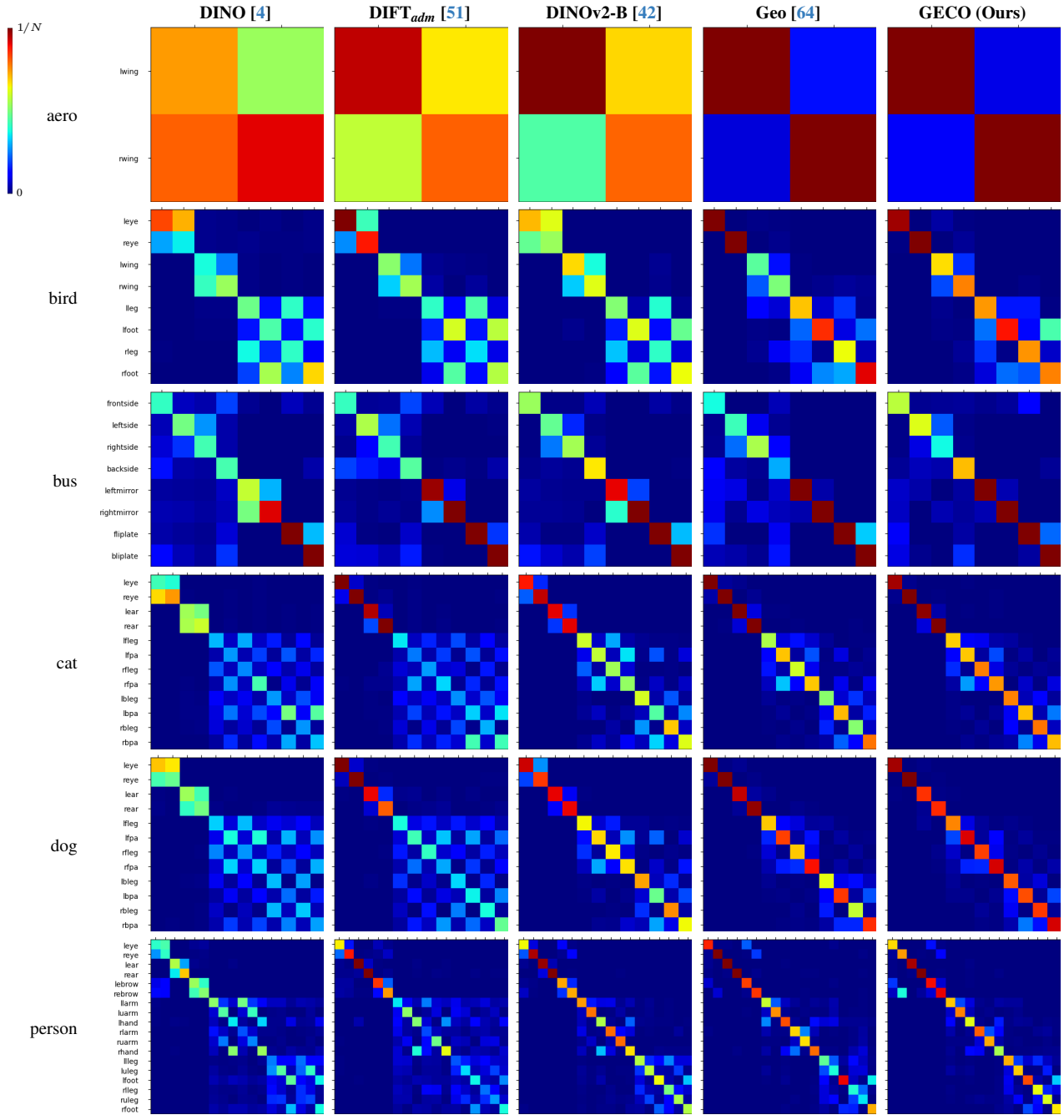


Figure 15. **Qualitative Evaluation of Segmentation Confusion Matrices on PascalParts [6].** We present the confusion matrix for the segmentation task on the PascalParts [6] dataset, showing every second category in alphabetical order. The matrix displays predicted parts on vertical and ground truth parts on the horizontal axis, plotting only parts per category with annotated symmetrical counterparts for visualization purposes. Diagonal entries indicate correct predictions, while off-diagonal entries show misclassifications. To address class imbalance, the matrix is column-normalized: each column is divided by the total number of patches for that ground truth part. With number of parts N, each ground truth part annotation is assigned a total mass of $1/N$, which is distributed across the predicted parts, such that the columns sum to $1/N$. The colour coding depends on the number of parts N.

8.3. 2D-3D Matching

Structure-from-Motion (SfM), Perspective-n-Point (PnP), and related methods that rely on rigid geometry are not typical applications of non-rigid intra-category feature learning, which remains effective even under substantial intra-class variations in shape and appearance. Nevertheless, we demonstrate that our features support downstream tasks such as inter-instance 2D-3D alignment, which is essential for pose estimation and geometric reasoning.

Dataset To obtain data that combines intra-class variability with rigid structure, we render two 3D animal meshes from the SMAL model [69]. We enhance their realism and intra-class diversity using ControlNet (see Fig. 16, Top), and j filter out hallucinated outputs.

Procedure We decorate all mesh vertices with our features by median-aggregating across views, using the ground-truth camera poses (see Fig. 16, Top). For 20 novel images, we compute 2D-3D correspondences (see Fig. 16, Bottom) using argmax matching to the mesh vertex with the highest feature similarity.

We report the mean geodesic distance between matched 2D keypoints (unprojected onto the mesh) p and ground-truth vertex p_{gt}:

$$\bar{d}_{\text{geo}} = \text{mean}\left(d_{\text{geo}}(p, p_{\text{gt}})\right), \tag{14}$$

where d_{geo} denotes the shortest path along the mesh surface between two vertices. In Fig. 16 (Bottom), black contours indicate regions of high geodesic error, where the matched 3D vertex is distant from the ground truth.

We further evaluate camera pose estimation based on these 2D-3D matches. A robust PnP algorithm inside RANSAC estimates the camera's rotation and translation relative to the 3D object. The quality of the estimated rotation R is assessed by comparing it to the ground-truth rotation R_{gt} using the median rotation error across all images:

$$\bar{e}_{\text{rot}} = \text{median}\left(\arccos\left(\frac{\text{trace}(R\text{gt}^{\top}R) - 1}{2}\right)\right). \tag{15}$$

Results As shown in Tab. 8, our GECO features outperform both Geo [64] and DINOv2-B [42] in terms of rotation error and geodesic distance. This confirms that our method enables effective intra-category 2D-3D matching for pose estimation and geometric reasoning.

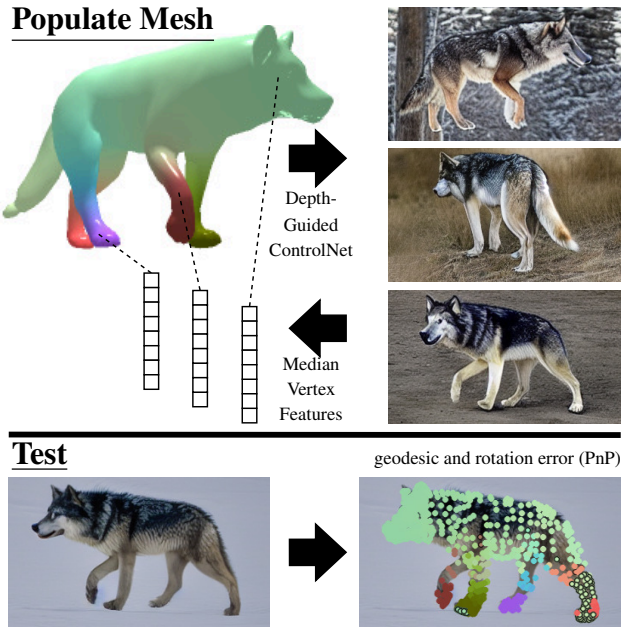


Figure 16. **Evaluation Protocol. Top:** A mesh from the SMAL model [69] is rendered to obtain depth images, which guide ControlNet-based synthesis of realistic 2D images. Hallucinated outputs are manually filtered. Mesh vertices are then populated with features by taking the median across the corresponding 2D views. **Bottom:** A novel view is processed, and 2D-3D correspondences are established via argmax matching to the mesh vertex with the highest feature similarity. Matches are color-coded according to surface color. Those exceeding a geodesic distance threshold d_{geo} to the ground-truth location are highlighted with a black contour.

	$\bar{e}_{\text{rot}}(^\circ) \downarrow$	$\bar{d}_{\text{geo}} \downarrow$
DINOv2-B [42]	14.8	0.34
Geo [64]	10.4	0.35
GECO (Ours)	**7.0**	**0.24**

Table 8. **Quantitative Evaluation of Viewpoint Reconstruction and 2D-3D Matching.** We assess the quality of 2D-3D correspondences using the median rotation error $\bar{e}_{\text{rot}}$ of the reconstructed view and mean geodesic distance $\bar{d}_{\text{geo}}$ between matched and ground-truth keypoints. Our method outperforms DINOv2-B [42] and Geo [64] on both metrics, demonstrating superior performance in pose estimation and geometric reasoning. Best scores are shown in **bold**.

References

[67] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chad Rolland, Laura Gustafson, Trevor Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[68] Gabriel L Oliveira, Abhinav Valada, Claas Bollen, Wolfram Burgard, and Thomas Brox. Deep learning for human part discovery in images. In *ICRA*, pages 1634–1641, 2016.

[69] Silvia Zuffi, Angjoo Kanazawa, David W. Jacobs, and Michael J. Black. 3D Menagerie: Modeling the 3D Shape and Pose of Animals. In *CVPR*, pages 5524–5532, 2017.

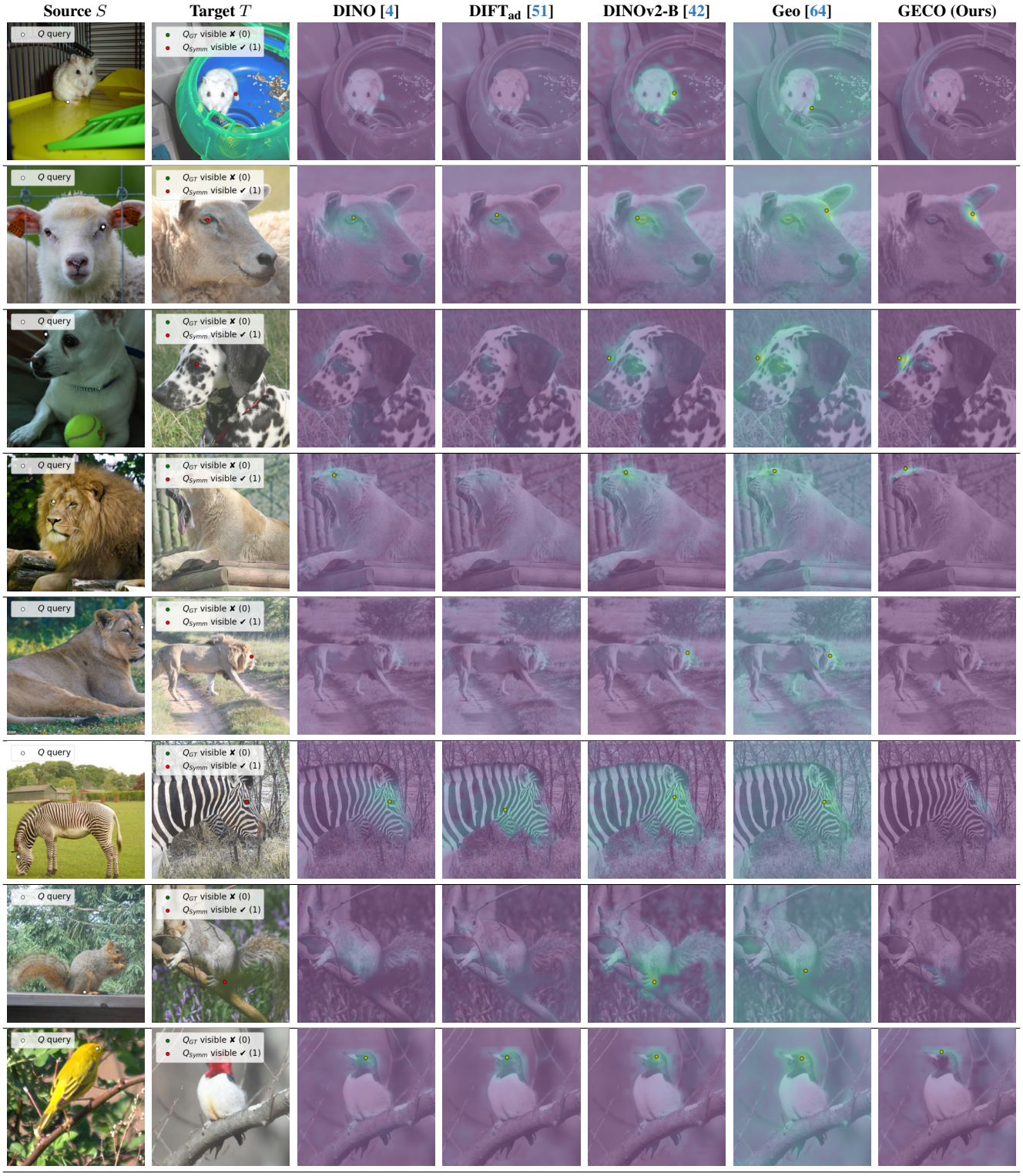


Figure 17. **Qualitative results on the task of assignment to the bin (01-case) on APK [64] and CUB [55] (Last row).**

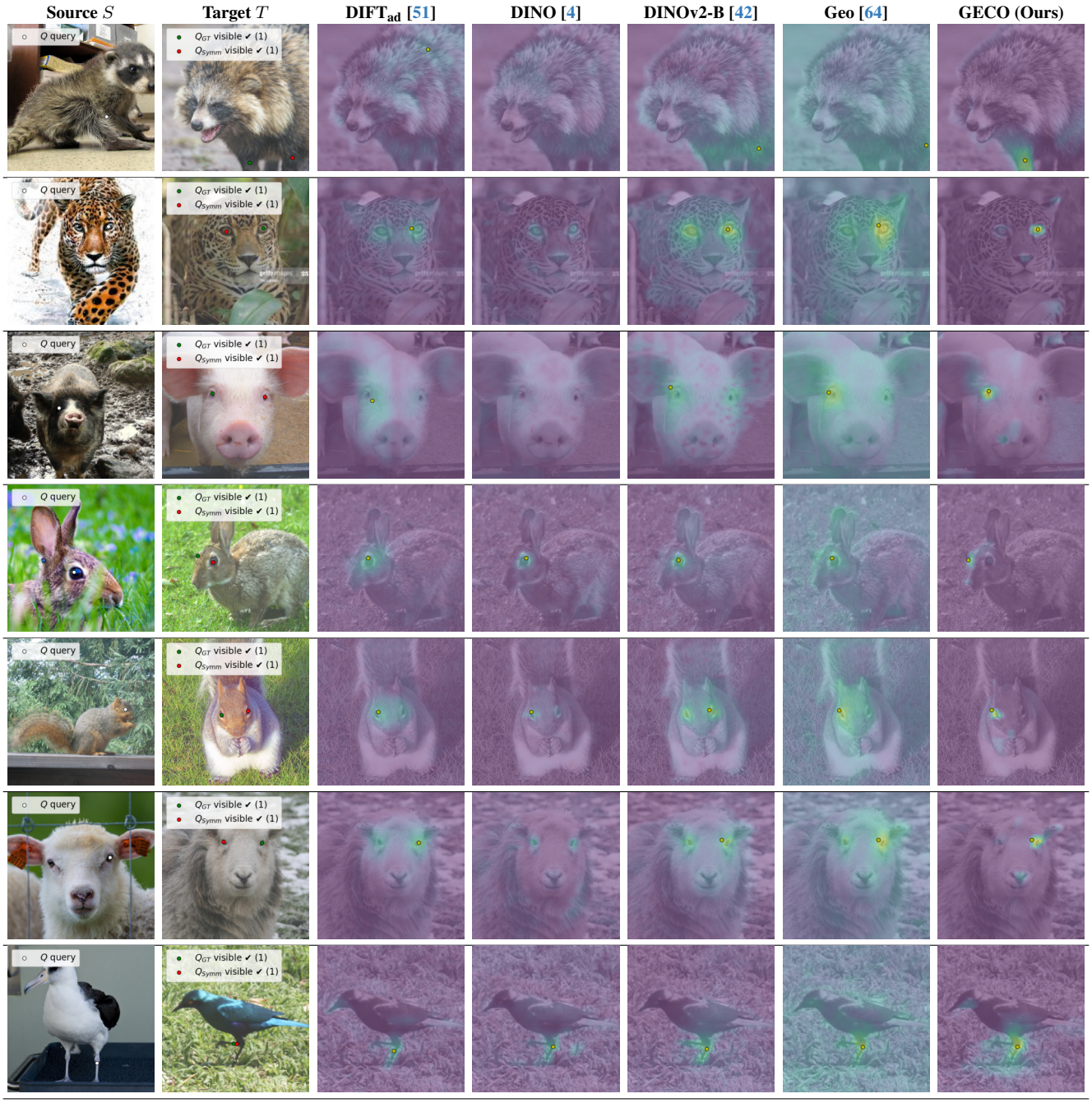


Figure 18. **Qualitative results Qualitative results on the task of assignment to the correct correspondence (11-case) on APK [64], and CUB [55] (Last row). Note that the keypoint pair in the last row would be excluded in the unambiguous TP subdivision.**

Figure 19. **Qualitative results Qualitative results on the task of semantic correspondence estimation (10-case) on PFPascal [22] (First row), APK [64], and CUB [55] (Last row).**