GECO: Geometrically Consistent Embedding with Lightspeed Inference

Regine Hartwig Dominik Muhle Riccardo Marin Daniel Cremers
TU Munich Munich Center for Machine Learning
{regine.hartwig, dominik.muhle, riccardo.marin, cremers}@tum.de

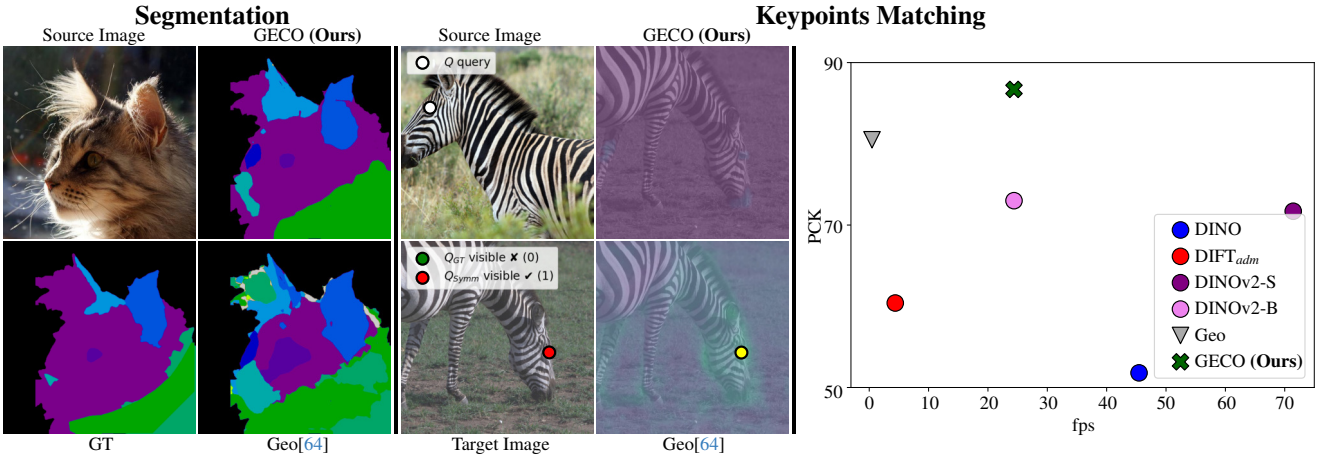

Figure 1. We propose **GECO**, an optimal-transport-based approach for learning geometrically consistent visual features. Characterizing geometric properties, like distinguishing left/right eyes or front/back legs, remains challenging even for sophisticated methods. For instance, in the keypoint transfer example, showing a zebra, Geo [64] confuses the eyes, while our features have low similarity. Our method learns robust feature representations, achieving high accuracy while remaining lightweight and efficient at inference time (plot on the right), enabling real-time applications at 30fps. Our feature embedding exhibits a continuous and structured understanding of objects, making it highly effective for image segmentation (left) and precise keypoint matching, even in challenging scenarios involving occlusions (middle, with attention maps overlaid on the target images).

Abstract

Recent advances in feature learning have shown that self-supervised vision foundation models can capture semantic correspondences but often lack awareness of underlying 3D geometry. GECO addresses this gap by producing geometrically coherent features that semantically distinguish parts based on geometry (e.g., left/right eyes, front/back legs). We propose a training framework based on optimal transport, enabling supervision beyond keypoints, even under occlusions and disocclusions. With a lightweight architecture, GECO runs at 30 fps, 98.2% faster than prior methods, while achieving state-of-the-art performance on PFPascal, APK, and CUB, improving PCK by 6.0%, 6.2%, and 4.1%, respectively. Finally, we show that PCK alone is insufficient to capture geometric quality and introduce new metrics and insights for more geometry-aware feature learning.
https://reginehartwig.github.io/publications/geco/

1. Introduction

Vision Foundation models either trained only on images [14, 42] or text-image pairs [47] produce flexible features that can be applied across various tasks such as image generation, style transfer, and correspondence estimation. While these models perform well across a range of applications, they are constrained by their limited ability to distinguish between geometric properties [16]. For instance, they may struggle to differentiate between left and right eyes or the legs of a chair, issues that have been explored in the context of the Janus problem [16, 64]. Since they are trained using consistency between self-augmented images, including flipping, they often do not differentiate between symmetric parts, and are even encouraged to learn features that are invariant to such transformations. These models often encounter challenges like ambiguous features for semantically similar regions and confusion due to occlusions. Such mismatches can lead to severe artifacts in practical downstream applications, such as a category template reconstruction pipeline incorrectly reconstructing 5-legged

animals or chairs [39]. More advanced approaches, such as Geo [64], enhance SD+DINO features [65] to address left-right ambiguities. However, despite its advancements, this method is often too slow, taking several seconds for a single prediction, making it impractical and difficult to scale.

Our work proposes **GE**ometrically **CO**nsistent embeddings, an efficient and robust approach to address this problem without compromising efficiency. A key limitation of previous methods is their reliance on argmax-based assignments during training, which inherently assumes the keypoint to be visible in both views and fails to account for (dis-)occlusions. Our intuition is that it therefore does not provide a strong supervision signal for the model to learn meaningful features. Specifically, it overlooks the fact that we are dealing with a partial assignment problem. For instance, in cases of occlusion, annotated information is ignored during training, even though it could help the model learn the correct bin assignment. Furthermore, the supervision signal is sparse: only a few annotated points are used, while the rest of the image contributes nothing to the training process. Although soft-argmax assignments with Gaussian perturbations of the position have been proposed to mitigate this issue, they tend to introduce blurring and ultimately still depend on an argmax operation, which is not a natural fit for the underlying assignment problem. Instead, to account for the partial assignments and differentiability, we introduce a novel *soft assignment loss* that leverages optimal transport on top of vision foundation models, leading to a nuanced and discriminative feature learning process. This loss function provides strong supervision feedback, allowing our module to learn distinctive features that effectively differentiate symmetric keypoints. The optimal transport loss is based on the Sinkhorn algorithm [11], which enables a differentiable soft assignment formulation for backpropagation.

Importantly, unlike previous approaches [48, 50], our method does not rely on cross-attention between image pairs, the marginal distributions of the optimal transport module are created and enforced differently (see supplementary), and the module operates without trainable parameters. As a result, our model is a *feature encoder rather than a feature matcher*. GECO focuses on robust representation learning rather than direct correspondence estimation between two images.

Concretely, we employ a pre-trained model and refine it through LoRA adaptation [26]. Leveraging *DINOv2* ensures computational efficiency, enabling more extensive data augmentation compared to prior methods and further enhancing the robustness of learned features.

Our method outperforms the state of the art while being **faster and smaller by two orders of magnitude w.r.t. the closest competitor** (40 ms vs 2127ms; 332MB vs 9GB). We perform an extensive evaluation, demonstrating our superiority on the classical PCK@0.1 metric on CUB, APK, and PFPascal datasets [22, 55, 64]. We also extend our analysis to complement the information provided by the PCK metric, which we found might not be indicative of some prediction modes. Thanks to this, we demonstrate that our better performance directly derives from a better geometrical understanding, which also leads to a more focused and calibrated prediction, distributing similarity on the correct area and assigning occluded parts to the bin (see Fig. 1).

In summary, our contributions are:

1. We propose a novel loss function and a lightweight architecture for image representation learning, leveraging optimal transport-based soft assignment;
2. Our formulation leads to geometrically-aware features, enabling state-of-the-art performance on correspondence estimation while being significantly more efficient. It surpasses geometrically-aware competitors on multiple datasets while *reducing computation time by 98%*, maintaining the speed of the lightweight backbone.
3. We conduct an extensive analysis of the common PCK metric and complement it with object part segmentation evaluation on PascalParts. Our method effectively separates parts, indicating that it learns meaningful, dense feature representations.

Our implementation will be instrumental for researchers interested in downstream tasks of geometry-aware encoders.

2. Related Work

Deep features Self-supervised and unsupervised methods have gained popularity [3–5, 17, 20, 23, 24, 42], but if trained on image-level objectives [3–5, 20, 24] often fail to capture fine spatial details [1, 54]. Patch-based methods like DINOv2 [42], inspired by [66], learn effective feature representations for tasks like clustering and matching. Diffusion models [14, 47] have become powerful generative models, with (Clean-)DIFT [49, 51] enabling dense feature extraction for vision tasks. While DINOv1, DINOv2 [4, 42], and DIFT [51] work well for zero-shot tasks [1, 19, 21, 25, 54], including semantic correspondence finding, they often struggle with geometric awareness, especially left-right distinctions [7, 16], potentially due to image-flipping augmentations [42]. Stable Diffusion (SD) features [14] encode more geometry than DINOv1 [4], but still lack left-right distinction, exposing limitations in spatial understanding, which can cause problems in 3D reconstruction [31, 35, 39].

Geometry-aware matching Several methods adress keypoint matching for rigid objects, such as SuperPoint [13] and SuperGlue [48], which use homography supervision. LOFTR [50] improves upon this with dense optimal transport matching. These approaches rely on (cross-)attention

and learnable optimal transport layers for sparse labels. Recent methods like DUST3R, MAST3R, Fast3R [33, 57, 62] assume rigid motion, limiting their applicability to more complex cases. Our method, GECO, focuses on learning features rather than matching and handles deformable objects and complex geometries. Deformable objects complicate training, as constraints (*e.g.* epipolar) are not available. Improving matching of deformable objects, CATs++ [9] uses attention layers, LCorrSan [27] estimates dense flow via correlation layers, [8] uses functional maps, and [34, 53] optimize dense probabilities. These methods focus on correspondence estimation. Our approach prioritizes feature learning, not matching.

Geometry-aware representation Recent work [56, 58, 60, 63] uses rigid inter-instance supervision for learning features with geometric awareness. Others focus on geometry-aware deformable intra-instance representation learning: Using 3D supervision, concurrent work [18, 32, 45] achieve good results on the geometric-aware matching task and a similar speedup as our method. Unlike our focus on speed and efficiency, [61] targets matching performance using additional correspondence supervision and Diffusion Features. However, the need for extra signals and lack of real-time capability limit its practical applicability. Earlier works like AnchorNet [40] use category-level supervision for geometry-aware features. DHF [36] fine-tunes diffusion features for semantic matching with a sparse contrastive loss on keypoints. Unsup [52] and Sphere [37] map to spherical coordinates with category and viewpoint supervision, while [15] also uses the spherical geometric prior without supervision. Completely unsupervised, SCOPS [28] enforces equivariance for co-part segmentation, while [10] uses contrastive learning for part discovery. We consider Geo [64] the most closely related work to ours. It uses a sparse contrastive loss with keypoint annotations and a soft argmax operator for feature enhancement. However, its reliance on SD features results in slower inference, category annotation dependency, and larger model size. Additionally, the extra parameters introduced by their head are significant, increasing the potential for overfitting.

Training with sparse correspondences To leverage sparse correspondence during training, a common approach applies an argmax operator [36, 64] with a sparse contrastive loss [29, 46]. While effective in some domains, this method, when used with keypoint annotations [36], provides a learning signal to only a small subset of annotated patches visible in both images, resulting in suboptimal feature representations [64]. We leverage optimal transport (OT) [11, 12] not as a matching module, but to define a structured soft assignment loss. Specifically, we employ the Sinkhorn algorithm [11, 12, 43], a differentiable OT solver that produces dense supervision across all image patches [2, 30, 59] by minimizing the cost to transport mass between distributions. Unlike prior work [48, 50], our formulation requires no additional trainable parameters, resulting in more broadly applicablel and task-agnostic features.

Our positioning Our method builds on a lightweight and efficient DINOv2-B backbone, fine-tuned using LoRA [26] to improve both computational efficiency and memory usage. To address the core challenges of sparse annotations and (dis-)occlusions in loss design, we introduce a soft optimal transport layer. This layer delivers a learning signal to all patches, effectively utilizing the full set of available annotations and enabling the learning of robust and discriminative feature representations.

3. Background

Argmax matching Considering a source image S and target one T, both equipped with a set of patches represented by their indices $\mathcal{I} := \{i_1, ..., i_l\}$ and $\mathcal{J} := \{j_1, ..., j_m\}$ and their d-dimensional features $\mathbf{X}^s \in \mathbb{R}^{l \times d}$ and $\mathbf{X}^t \in \mathbb{R}^{m \times d}$ respectively. Given a query location i, the corresponding $\widehat{j}$ on the target can be received by:

$$\widehat{j} = \arg\max_j \left\langle \mathbf{X}_i^s, \mathbf{X}_j^t \right\rangle . \tag{1}$$

While simple, argmax matching suffers from several drawbacks. It introduces ambiguity, as multiple geometrically distinct source locations can map to the same target point (e.g., left and right eyes both matching to one eye). It does not account for occlusions, often resulting in incorrect assignments with low similarity. Moreover, it imposes hard, one-hot assignments that assume equal mass per patch across images, limiting its ability to handle scale differences or partial matches. Crucially, it ignores the global structure of the assignment, focusing only on local maxima, which leads to sparse gradients and hinders effective training. We instead formulate matching as an optimal transport problem. To handle occlusions, we introduce a dustbin entry in the assignment matrix, and employ a soft, iterative solver [11, 12] that provides meaningful gradients for all features, improving training stability and convergence.

Optimal Transport formulation Suppose we have pairs of patches $(i, j) \in \mathcal{I} \times \mathcal{J}$. We solve for an assignment between two images [42, 48] by using the cosine similarity of the features as score matrix $\mathbf{C}$ with

$$\mathbf{C}_{i,j} = \left\langle \mathbf{X}_i^s, \mathbf{X}_j^t \right\rangle \in [-1, 1]. \tag{2}$$

In order to model occlusions properly, we augment the vectors with a dustbin at $l' = l + 1$ and $m' = m + 1$, which gets assigned a threshold parameter $z \in \mathbb{R}$:

$$\mathbf{C}_{i_{l'}, i_{m'}} = \mathbf{C}_{i_{l'}, j} = \mathbf{C}_{i, i_{m'}} = z \quad \forall i, j \in \mathcal{I} \times \mathcal{J}. \tag{3}$$

We then solve for an assignment matrix $\mathbf{P} \in U(\mathbf{a}, \mathbf{b})$, where

$$U(\mathbf{a}, \mathbf{b}) := \left\{ \mathbf{P} \in \mathbb{R}_+^{i_{l'} \times i_{m'}} | \mathbf{P}\mathbf{1}_{i_{m'}} = \mathbf{a}, \mathbf{P}^T \mathbf{1}_{i_{l'}} = \mathbf{b} \right\}, \tag{4}$$

$\mathbf{a} \in \Sigma_{i_{l'}}$, and $\mathbf{b} \in \Sigma_{i_{m'}}$, i.e. the marginalizations of P (see supplementary) are in the respective simplex $\Sigma_n := \{\mathbf{x} \in \mathbb{R}_+^n : \mathbf{x}^T \mathbf{1}_n = 1\}$. The assignment can be obtained by solving the OT problem

$$\widehat{\mathbf{P}} = \arg \max_{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})} \langle \mathbf{P}, \mathbf{C} \rangle . \tag{5}$$

To provide gradients for all input features during training, we use a regularized OT, which yields a smooth and differentiable assignment

$$\widehat{\mathbf{P}}^{\lambda} = \arg \max_{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})} \langle \mathbf{P}, \mathbf{C} \rangle + \lambda H(\mathbf{P}), \tag{6}$$

with $H(\mathbf{P}) = -\sum_{i,j} \mathbf{P}_{ij} \log \mathbf{P}_{ij}$ being the entropy of the assignment matrix. For $\lambda > 0$ the entropy regularization promotes smoother, less sparse assignments, resulting in a differentiable soft assignment $\widehat{\mathbf{P}}^{\lambda}$ that provides a gradient signal for all input features. Furthermore, changing the problem to an unbalanced OT problem, using additionally the KL-divergence as a regularizer [44], helps in the case of unbalanced marginals $\mathbf{a}$ and $\mathbf{b}$, meaning, that the amount of mass that needs to be assigned is not equal for both marginals

$$\begin{aligned} \widehat{\mathbf{P}}^{\lambda,\alpha,\beta} = \arg & \max_{\mathbf{P} \in \mathbb{R}_+^{i_{l'} \times i_{m'}}} \langle \mathbf{P}, \mathbf{C} \rangle + \lambda H(\mathbf{P}) \\ & - \alpha \, \mathrm{KL}\left(\mathbf{P}\mathbf{1}_{i_{m'}} || \mathbf{a}\right) - \beta \, \mathrm{KL}\left(\mathbf{P}^T \mathbf{1}_{i_{l'}} || \mathbf{b}\right) . \end{aligned} \tag{7}$$

In general, for formulating the OT problem, we need to know the distributions of $\mathbf{a}$ and $\mathbf{b}$. However, using the unbalanced OT problem, we can relax the requirement of knowing the marginals, which is crucial for our application, as we do not have access to the ground truth marginals, but only estimates.

4. Method

We now describe our approach, including the feature learning module and the differentiable OT layer without learnable parameters, which processes the cosine similarity of patches. Finally, we introduce a loss function that encourages higher output probabilities for positive and bin pairs and lower probabilities for negative pairs.

Architecture Our architecture is simple, efficient, and interpretable (see Fig. 2). We use a pre-trained DINOv2-B model [42], kept frozen during training, and adapt it with a LoRA adapter [26] of rank 10.

Differentiable OT layer We compute the cosine similarity of all features from S and T, including the background, obtaining a cosine similarity matrix. We augment this matrix with an additional row and column, setting a bin threshold of $z = 0.3$, and obtaining the score Matrix $\mathbf{C}$.

Next, we pass $\mathbf{C}$ to a differentiable optimal transport (OT) layer, which computes the optimal transport plan between the features of S and T. Unlike previous works [48, 50], which enforce hard constraints on the boundaries $\mathbf{a}$ and $\mathbf{b}$, we adopt a KL-regularized soft assignment (Eq. (7)), allowing greater flexibility. Additionally, the OT layer has no learnable parameters and contributes to the loss function rather than being part of the model itself.

The marginals $\mathbf{a}$ and $\mathbf{b}$ of the multivariate probability matrix $\widehat{\mathbf{P}}$ are estimated using mask annotation and a method to determine the amount of the visible mass of the shape. Details on this are provided in the supplementary material. The Sinkhorn-Knopp algorithm runs for 10 iterations, producing a probability matrix $\widehat{\mathbf{P}}^{\lambda,\alpha,\beta}$ with the same dimensions as the score matrix $\mathbf{C}$. The hyperparameters of the OT layer are $\lambda = 0.1$, $\alpha = 10$, and $\beta = 10$.

Binary cross entropy loss As we do not have access to the ground truth matrix $\mathbf{P}$, which assigns all features of S to all features of T, but only to the sets of positive, negative, and bin correspondences $\mathcal{M}^+, \mathcal{M}^-, \mathcal{M}^0 \subset \mathcal{I} \times \mathcal{J}$, we formulate our loss only on these sets. Using a binary cross-entropy loss, we train the model to predict the correct values at some sparse entries of the assignment matrix $\mathbf{P}$ by

$$\mathcal{L} = - \sum_{(i,j) \in \mathcal{M}^+ \cup \mathcal{M}^0} \log \widehat{\mathbf{P}}_{i,j}^{\lambda,\alpha,\beta} - \sum_{(i,j) \in \mathcal{M}^-} \log(1 - \widehat{\mathbf{P}}_{i,j}^{\lambda,\alpha,\beta}). \tag{8}$$

We additionally add matches between foreground and background features to the set $\mathcal{M}^-$, which are not part of the ground truth, to enforce the model to learn to distinguish between the two.

5. Experiments

In this section we extensively evaluate our method compared to baselines, analyzing their capability of understanding geometry. First, in Sec. 5.1, we utilize the established PCK metric on different datasets, where we show that we are on par and better than the previous state-of-the-art. We also provide a detailed analysis of the limitations of PCK, and propose new metrics to inspect geometric knowledge of the features. In Sec. 5.2, we show that our method, using centroid clustering, also performs well on the pixel-level classification task without additional training, relying on a simple supervised classifier. This indicates that our approach effectively learns meaningful features.

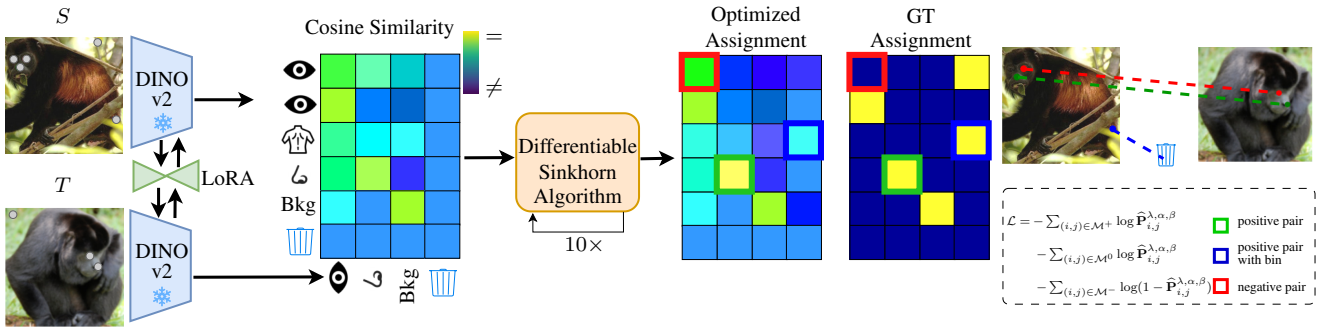

Figure 2. **Architecture and Training.** At training time, our method takes a pair of images composed of a source S and targets one T, and obtains features through a DINOv2 [42] frozen model complemented with a LoRA adapter [26]. The features are used to compute the matching by cosine similarity associations between all patches. For visualization purposes, we show the matrix for the landmarks (grey dots). We pass the obtained matrix to a differentiable Optimal Transport Layer that, in a few iterations, obtains the predicted assignment. We compare this with the ground truth, and we use this as a supervision signal for our method. Our training objective also provides supervision signals on negative keypoint pairs and, thanks to the "bin" category, on keypoints that do not even have a visible counterpart in one of the two images.

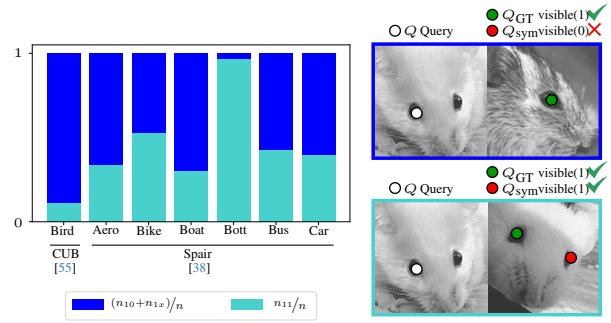

Figure 3. **PGCK Dataset imbalance**. We report keypoint pairs grouping under our PGCK subdivision. n_{11} counts only the pairs for which a geometric mismatch is possible. Other keypoint pairs (having no geom. counterpart/ with occluded geom. counterpart) are counted in $(n_{10} + n_{1x})$. Due to the high category imbalance, geometric error modes impact the overall PCK differently.

5.1. Analysis with (and of) PCK

Percentage of correct keypoints (PCK) The percentage of correct keypoints (PCK) is a widely used metric for evaluating the performance of correspondence estimation methods. We take an image pair, where keypoint matches are annotated, and evaluate how many query points Q are correctly reprojected into the second view (see Fig. 5). A reprojection is labeled as correct if the distance ϵ between predicted and annotated location Q_{GT} is less than a threshold α_{img} of the image size or α_{bbox} of the bounding box size. Following the evaluation protocol of previous work [51], we use argmax matching for building assignments and reprojection only in one direction. In this work, we consider the PCK$\texttt{point}@\alpha_{img}$ version of it, where the percentage of correct keypoint reprojections per point is $\widehat{n}/n$, the fraction of the sum of correctly reprojected points $\widehat{n}$ across all image pairs to all annotated point pairs n. It is interesting to note that PCK considers only keypoint pairs in which the query keypoint is visible in the target image. Hence, it does not evaluate a model's performance when the reprojection is not visible in the target image, but its symmetric counterpart is (see Fig. 5, first four rows). To address this, we also report performance using qualitative results.

Percentage of geometry-aware correct keypoints (PGCK) Although PCK has played an important role in evaluating matching methods, we believe it is insufficient to depict the methods' understanding of geometry. We propose to break down the proportion between the correct reprojections $\widehat{n}$ and the total number of keypoints n into different sets. Specifically, we separate the evaluation of query points that have a visible symmetric counterpart in the target image (n_{11}, an example in Fig. 3, second row) from those that have a symmetric counterpart but are occluded in the target, and those for which a symmetric counterpart does not exist (n_{10} and n_{1x} respectively; example of the first in Fig. 3, first row). PCK can then be divided in:

$$\text{PCK}\texttt{point} = \frac{\widehat{n}}{n} \tag{9}$$

$$= \frac{\widehat{n}_{10} + \widehat{n}_{11} + \widehat{n}_{1x}}{n_{10} + n_{11} + n_{1x}} \tag{10}$$

$$= \underbrace{\frac{\widehat{n}_{10}}{n_{10}}\frac{n_{10}}{n}}_{\substack{Q_{\text{GT}}\checkmark \\ Q_{\text{Symm}}\times}} + \underbrace{\frac{\widehat{n}_{11}}{n_{11}}\frac{n_{11}}{n}}_{\substack{Q_{\text{GT}}\checkmark \\ Q_{\text{Symm}}\checkmark}} + \underbrace{\frac{\widehat{n}_{1x}}{n_{1x}}\frac{n_{1x}}{n}}_{\substack{Q_{\text{GT}}\checkmark \\ Q_{\text{Symm}}\text{-}}}, \tag{11}$$

PGCK

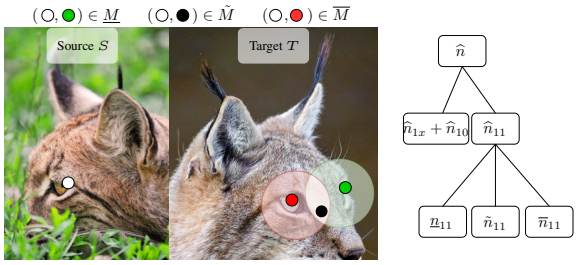

Figure 4. **Geometric Ambiguity of PGCK.** In the case of symmetric correspondences, the PCK metric does not account for ambiguous assignments of true positive correspondences. The Green dot around the annotated match will collect an assignment that results in a true positive match, the Red dot collects symmetric mismatches. In cases where the symmetric counterpart keypoint is sufficiently close, the circles overlap. These points can not be unambiguously assigned to either keypoint. The PCK metric still counts them as true positives. Points falling into the ambiguous set area are collected in $\tilde{M}$ and make up around 50% of the prediction for all investigated methods.

where the number of keypoint pairs $n = n_{10}+n_{11}+n_{1x}$ and number of correct reprojections $\widehat{n} = \widehat{n}_{10} + \widehat{n}_{11} + \widehat{n}_{1x}$. Although PCK comprehends all these quantities, we see that evaluation on n_{10} and n_{1x} is less informative than those on set n_{11}, where actually the method could get confused by the presence of a symmetric element. For evaluating the geometric reasoning, we are interested only in the number $\widehat{n}_{11}$, where the model correctly matches the query keypoint when the symmetric counterpart is also visible in the target image. We can see its relevance by measuring the distribution of the annotated point pairs for the CUB and SPair datasets in the three categories. We report the counting in Fig. 3. In the CUB dataset, $^{n_{1x}}/_{n} = 78\%$ of the keypoint pairs have no symmetric counterpart. For the remaining 22% of the keypoint pairs, in only 11% of the keypoint pairs, the symmetric counterpart is also visible in the second view. The SPair dataset has a high variation between the categories, with the highest value for the bottle category (97%) and the lowest for the bird category (23%). In the following, we report both evaluations using PCK metric and our proposed split. We call such division *Percentage of geometry-aware correct keypoints* (PGCK).

Geometric Ambiguity Although $^{\widehat{n}_{11}}/_{n_{11}}$ show informativeness about geometric knowledge of models, to complete our analysis, we also highlight a special set that is worth further investigation. Specifically, the set $\tilde{M}$ with $|\tilde{M}| = \tilde{n}_{11}$ contains point pairs where the predicted location is close (with a distance smaller than the defined radius) to Q_{GT} and Q_{Symm} as shown in Fig. 4.

Therefore, success in these cases does not directly measure geometric knowledge, as it "cheats" the metric by predicting points in the middle of the two. To compensate for this, it is possible to derive two further measures. First, the **unambiguous true positives U-TP** $\underline{n}_{11}/n_{11}$, which measures the cardinality $|\underline{M}| = \underline{n}_{11}$ of the subset, which contains correctly matched keypoint pairs, where Q_{Symm} is far enough away from the predicted position. Second, the **false correspondences** $\overline{n}_{11}/n_{11}$, which measures the cardinality $|\overline{M}| = \overline{n}_{11}$ of the subset, which contains wrongly matched keypoint pairs, where Q_{GT} is far enough away from Q_{Symm} to be considered as a completely wrong match. As we will see later, the high number of keypoint pairs $\tilde{n}_{11}$ indicates that the commonly used radius $\alpha_{img} = 0.1$ is too big for the PCK metric. A further analysis of how the subsets behave for other values of α_{img} is provided in the supplementary material.

	Unambiguous Correct Pred. $\frac{\underline{n}_{11}}{n_{11}} \uparrow$	ambiguous $\frac{\tilde{n}_{11}}{n_{11}}$	Unambiguous Wrong Pred. $\frac{\overline{n}_{11}}{n_{11}} \downarrow$
DINO [4]	17.0	38.8	16.0
DIFT$_{\text{ad}}$ [51]	25.2	43.8	8.4
DINOv2-S [42]	25.1	50.9	13.8
DINOv2-B [42]	24.5	51.8	16.0
Geo [64]	36.2	53.1	2.9
GECO (Ours)	**40.0**	53.2	**2.3**

Table 1. **Geometric ambiguity** Analysis of PGCK on APK [64]. We outperform previous work in the unambiguous geom. correct matching (left) by 3.8 %, while our method disregards more of the unambiguously wrong pairs (right). The best scores are highlighted in **bold**.

5.1.1. Evaluation

Data In this experiment, we evaluate our method on datasets with pairwise keypoint annotations. Specifically, we use the CUB dataset [55], selecting 10,000 image pairs of at random, as well as SPair [38], PFPascal [22], and APK [64], which provide predefined image pairs. Since PF-Pascal lacks symmetric counterpart annotations, we assess only the standard PCK metric on this dataset.

Results The quantitative PCK analysis on PFPascal, APK, and Spair is shown in Tab. 2. Our method surpasses previous state-of-the-art by 6.0% on PFPascal, 6.2% on APK, and 4.1% on CUB, while being nearly two orders of magnitude faster. Notably, Geo, using DINOv2-B and Stable Diffusion features, generalizes less well on CUB than DINOv2-S. In fact, the only dataset where Geo outperforms our method is Spair, indicating that is not a general purpose model. Detailed PCK results for PFPascal, APK, and Spair appear in the supplementary material. We provide an in-depth APK evaluation in Tab. 1, where our method improves across all n_{11} splits and consistently outperforms Geo in three of four cases (see supplementary).

	PFPascal [22]		APK [64]		Spair [38]		CUB [55]	
	PCK↑	time[ms]↓	PCK↑	time[ms]↓	PCK↑	time[ms]↓	PCK↑	time[ms]↓
DINO [4]	65.3	26	51.8	22	48.4	25	75.6	26
DIFT_{ad} [51]	72.5	221	60.4	228	59.3	222	84.2	222
DINOv2-S [42]	85.5	**14**	71.7	**14**	66.3	**15**	92.4	**17**
DINOv2-B [42]	86.0	45	73.0	40	67.1	38	**92.8**	43
Geo [64]	86.1	2141	80.5	2127	**90.1**	2159	88.4	2274
Sphere [37]	88.5	2152	75.2	2144	74.5	2164	92.1	2151
GECO (Ours)	**92.1**	45	**86.7**	40	85.2	38	**92.5**	43

Table 2. **Quantitative evaluation of PCK on PFPascal [22] (Left), APK [64] (Middle), Spair [38] (Middle), and CUB [55] (Right).** We report the PCK@$\alpha = 0.1$ for four different datasets on the test split. Our method outperforms competitors in three out of four datasets by 6.0% on PFPascal, 6.2% on APK, and 4.1% on CUB, while $\sim$2 orders of magnitude faster. In contrast, does not generalize well to CUB, where it lags behind DINOv2-S. Both Geo and our method are trained on PFPascal [22], APK [64], and Spair [38], while Sphere is trained solely on Spair. The best scores are highlighted in **bold**. Methods are considered to be on par if their performance difference is less than 0.5%. We mark **CUB [55]** in blue to highlight that it has not been seen at training time.

5.2. Feature Space Segmentation

The PCK metric evaluates correspondence accuracy based on a limited set of keypoints. While effective at those points, it does not assess the dense feature space, which is crucial since methods are trained on these annotations. We propose evaluating the feature space by partitioning it into semantically meaningful parts using only Euclidean distance to part-specific representation vectors. This indicates whether the learned features capture meaningful, consistent structures within the image.

5.2.1. Evaluation

Centroid Clustering To analyze the feature-space structure, we compute centroids for each annotated object part, assessing whether the model can separate parts using only Euclidean distance to these centroids. We first gather sets of feature vectors corresponding to each annotated part and then compute their median as the part representations. On the test set, each patch is assigned to the nearest centroid based on Euclidean distance, evaluating the model's ability to discriminate parts purely from the learned features.

Data We evaluate dense features using part annotations from PascalParts [6], which offer consistent, category-specific labels for assessment.

Results Our learned feature representations reliably distinguish semantically similar parts with distinct geometric properties. Figure 6 shows qualitative examples where our method accurately separates challenging regions such as left/right eyes, wings, and ears. In contrast, the foundation model often exhibits artifacts, marked by red arrows and circles in the figure. Geo [64], which uses Gaussian sampling around keypoints during training, tends to assign overly broad regions to the eyes. Quantitative and qualitative confusion matrix analyses in the supplementary highlight geometric ambiguities in foundation models. Our method achieves geometrical awareness comparable to Geo [64] and shows greater consistency on non-geometric parts.

5.3. Runtime

Procedure We measure inference time by running a forward pass (excluding image loading) on 1,000 images per dataset using an RTX A4000 GPU, averaging the results. All models are evaluated with a batch size of 1.

Results Our timing results can be found in Tab. 2. For DINOv2 our measurements are consistent with those reported in the NVIDIA NCG catalog [41], falling within the same order of magnitude. The addition of our "light-speed" adapter introduces minimal overhead, contributing less than 0.5 ms to DINOv2-B's baseline runtime of approximately 40 ms. Importantly, the performance of the original DINOv2 on the geometric matching task is not as strong. On the other hand, Geo's [64] geometrically aware features, which rely on diffusion models, result in much longer inference times exceeding 2 seconds. Furthermore, in contrast to Geo [64], our method benefits from lower memory requirements, enabling efficient batch processing at inference time, which would further amplify the speed performance gap, making our method significantly faster and more scalable for practical applications.

6. Conclusion

We present a fast and efficient representation learning method based on optimal transport loss, achieving state-of-the-art PCK performance and improved geometric understanding. Our structured analysis highlights underexplored aspects of feature learning using PCK subdivisions and centroid clustering. While our method is lightweight and generalizes well, it relies on sparse keypoint supervision and is currently limited to categories with available annotations.

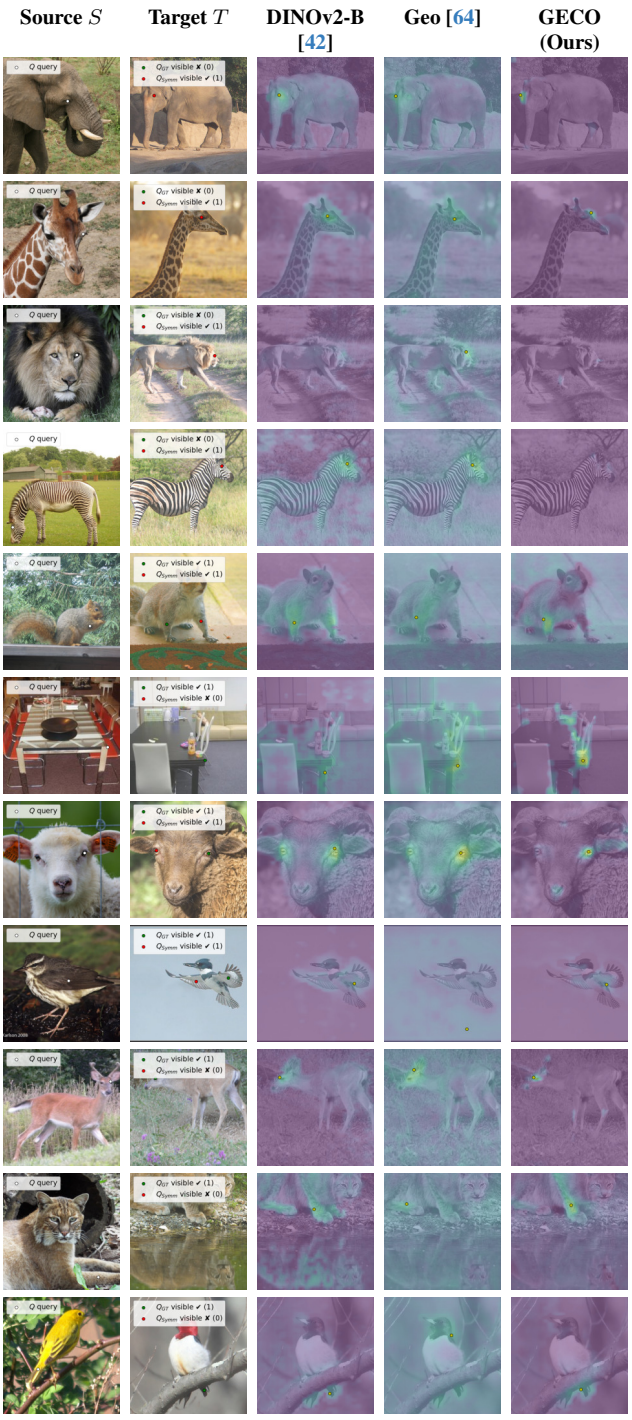

Figure 5. **Qualitative results on correspondence estimation for APK [64], PFPascal [22], and CUB [55].** (Top two rows) The model accurately locates the ground truth correspondence, becoming visible with slight movement, while ignoring symmetric counterparts. (Third, Fourth row) When the symmetric counterpart is occluded, attention is uniformly low across the image. (Bottom row) CUB samples confirm that our method preserves the pretrained foundation model's generalization.

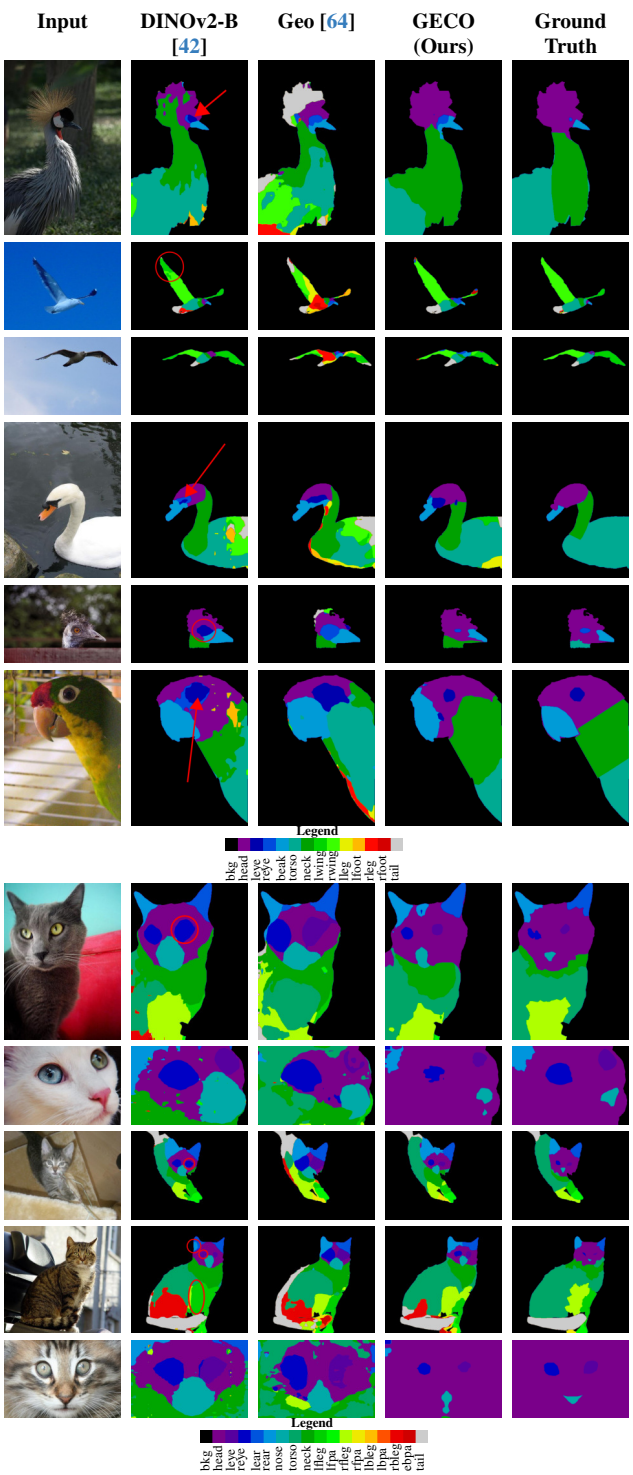

Figure 6. **Clustering of the feature space based on Euclidean distance to a part representation vector for PascalParts [6].** Our learned representation effectively separates even challenging parts, such as left and right eyes, wings, and ears, while also being similarly time and memory efficient as the DINOv2v2-B backbone and much more time efficient than Geo.

Acknowledgements: We thank the anonymous reviewers for their valuable feedback. This work was supported by the ERC Advanced Grant SIMULACRON, cby the Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV) through the AuSeSol-AI project (grant 67KI21007A), and by the TUM Georg Nemetschek Institute Artificial Intelligence for the Built World (GNI) through the AICC project.

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021. 2

[2] Nicolas Bonneel and Julie Digne. A survey of optimal transport for computer graphics and computer vision. In *Computer Graphics Forum*, pages 439–460. Wiley Online Library, 2023. 3

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018. 2

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2, 6, 7

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 2

[6] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, pages 1971–1978, 2014. 7, 8

[7] Yue Chen, Xingyu Chen, Anpei Chen, Gerard Pons-Moll, and Yuliang Xiu. Feat2gs: Probing visual foundation models with gaussian splatting. In *CVPR*, pages 6348–6361, 2025. 2

[8] Xinle Cheng, Congyue Deng, Adam Harley, Yixin Zhu, and Leonidas Guibas. Zero-Shot Image Feature Consensus with Deep Functional Maps, 2024. 3

[9] Seokju Cho, Sunghwan Hong, and Seungryong Kim. Cats++: Boosting cost aggregation with convolutions and transformers. *IEEE TPAMI*, 45(6):7174–7194, 2022. 3

[10] Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised part discovery from contrastive reconstruction. *NeurIPS*, 34:28104–28118, 2021. 3

[11] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *NeurIPS*, 2013. 2, 3

[12] Marco Cuturi and Charlotte Bunne. Tutorial on optimal transport in learning, control, and dynamical systems. https://icml.cc/virtual/2023/tutorial/21559, 2023. 2024-11-12. 3

[13] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR workshops*, pages 224–236, 2018. 2

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021. 1, 2

[15] Olaf Dünkel, Thomas Wimmer, Christian Theobalt, Christian Rupprecht, and Adam Kortylewski. Do it yourself: Learning semantic correspondence from pseudo-labels. *arXiv preprint arXiv:2506.05312*, 2025. 3

[16] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *CVPR*, pages 21795–21806, 2024. 1, 2

[17] Michael Fuest, Pingchuan Ma, Ming Gui, Johannes Schusterbauer, Vincent Tao Hu, and Bjorn Ommer. Diffusion models and representation learning: A survey. *arXiv preprint arXiv:2407.00783*, 2024. 2

[18] Frank Fundel, Johannes Schusterbauer, Vincent Tao Hu, and Björn Ommer. Distillation of diffusion features for semantic correspondence. In *WACV*, pages 6762–6774, 2025. 3

[19] Walter Goodwin, Sagar Vaze, Ioannis Havoutis, and Ingmar Posner. Zero-shot category-level object pose estimation. In *ECCV*, pages 516–532. Springer Nature Switzerland, 2022. 2

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33:21271–21284, 2020. 2

[21] Kamal Gupta, Varun Jampani, Carlos Esteves, Abhinav Shrivastava, Ameesh Makadia, Noah Snavely, and Abhishek Kar. Asic: Aligning sparse in-the-wild image collections. In *CVPR*, pages 4134–4145, 2023. 2

[22] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE TPAMI*, 40(7):1711–1725, 2017. 2, 6, 7, 8

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 2

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 2

[25] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised semantic correspondence using stable diffusion. *NeurIPS*, 36:8266–8279, 2023. 2

[26] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 2, 3, 4, 5

[27] Shuaiyi Huang, Luyu Yang, Bo He, Songyang Zhang, Xuming He, and Abhinav Shrivastava. Learning semantic correspondence with sparse annotations. In *ECCV*, pages 267–284, 2022. 3

[28] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. Scops:

Self-supervised co-part segmentation. In *CVPR*, pages 869–878, 2019. 3

[29] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, et al. Openclip. Zenodo, 2021. 3

[30] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *CVPR*, pages 17658–17668, 2024. 3

[31] Mohammad Nasir Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–8, 2022. 2

[32] Dominik Kloepfer, João F Henriques, and Dylan Campbell. Loco: Learning 3d location-consistent image features with a memory-efficient ranking loss. *NeurIPS*, 37:124391–124419, 2024. 3

[33] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, pages 71–91. Springer, 2024. 3

[34] Xin Li, Deng-Ping Fan, Fan Yang, Ao Luo, Hong Cheng, and Zicheng Liu. Probabilistic model distillation for semantic correspondence. In *CVPR*, pages 7505–7514, 2021. 3

[35] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, pages 9298–9309, 2023. 2

[36] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *NeurIPS*, 36, 2024. 3

[37] Octave Mariotti, Oisin Mac Aodha, and Hakan Bilen. Improving semantic correspondence with viewpoint-guided spherical maps. In *CVPR*, pages 19521–19530, 2024. 3, 7

[38] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543*, 2019. 5, 6, 7

[39] Tom Monnier, Matthew Fisher, Alexei A Efros, and Mathieu Aubry. Share with thy neighbors: Single-view reconstruction by cross-instance consistency. In *ECCV*, pages 285–303, 2022. 2

[40] David Novotny, Diane Larlus, and Andrea Vedaldi. Anchornet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *CVPR*, pages 5277–5286, 2017. 3

[41] Nvidia. Ngc catalog 2023 tutorial 21559. https://catalog.ngc.nvidia.com/orgs/nvaie/models/imagenet_nv_dinov2, 2023. 2024-11-12. 7

[42] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[43] Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *CVPR*, pages 460–467, 2009. 3

[44] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 4

[45] Qiyang Qian, Hansheng Chen, Masayoshi Tomizuka, Kurt Keutzer, Qianqian Wang, and Chenfeng Xu. Bridging viewpoint gaps: Geometric reasoning boosts semantic correspondence. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11579–11589, 2025. 3

[46] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, and G. Krueger. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 3

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2

[48] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020. 2, 3, 4

[49] Nick Stracke, Stefan Andreas Baumann, Kolja Bauer, Frank Fundel, and Bjorn Ommer. CleanDIFT: Diffusion Features without Noise. In *CVPR*, pages 117–127, 2025. 2

[50] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *CVPR*, pages 8922–8931, 2021. 2, 3, 4

[51] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *NeurIPS*, 36:1363–1389, 2023. 2, 5, 6, 7

[52] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. *NeurIPS*, 30, 2017. 3

[53] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Probabilistic warp consistency for weakly-supervised semantic correspondences. In *CVPR*, pages 8708–8718, 2022. 3

[54] Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *CVPR*, pages 10748–10757, 2022. 2

[55] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset, 2011. 2, 5, 6, 7, 8

[56] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, pages 5294–5306, 2025. 3

[57] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, pages 20697–20709, 2024. 3

[58] Yuanbo Xiangli, Ruojin Cai, Hanyu Chen, Jeffrey Byrne, and Noah Snavely. Doppelgangers++: Improved Visual Disambiguation with Geometric 3D Features. In *CVPR*, pages 27166–27175, 2025. 3

[59] Jiankai Xing, Fujun Luan, Ling-Qi Yan, Xuejun Hu, Houde Qian, and Kun Xu. Differentiable rendering using rgbxy derivatives and optimal transport. *ACM Transactions on Graphics (TOG)*, 41(6):1–13, 2022. 3

[60] Chenfeng Xu, Huan Ling, Sanja Fidler, and Or Litany. 3difftection: 3d object detection with geometry-aware diffusion features. In *CVPR*, pages 10617–10627, 2024. 3

[61] Fei Xue, Sven Elflein, Laura Leal-Taixe, and Qunjie Zhou. MATCHA: Towards Matching Anything. In *CVPR*, pages 27081–27091, 2025. 3

[62] Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *CVPR*, pages 21924–21935, 2025. 3

[63] Yuanwen Yue, Anurag Das, Francis Engelmann, Siyu Tang, and Jan Eric Lenssen. Improving 2d feature representations by 3d-aware fine-tuning. In *ECCV*, pages 57–74, 2024. 3

[64] Junyi Zhang, Charles Herrmann, Junhwa Hur, Eric Chen, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. Telling left from right: Identifying geometry-aware semantic correspondence. In *CVPR*, pages 3076–3085, 2024. 1, 2, 3, 6, 7, 8

[65] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *NeurIPS*, 36, 2024. 2

[66] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 2