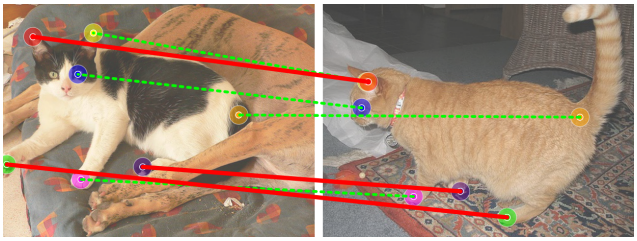
Telling Left from Right: Identifying Geometry-Aware Semantic Correspondence

Junyi Zhang[1] Charles Herrmann[2] Junhwa Hur[2] Eric Chen[3]
Varun Jampani[4] Deqing Sun[2] Ming-Hsuan Yang[2,5]

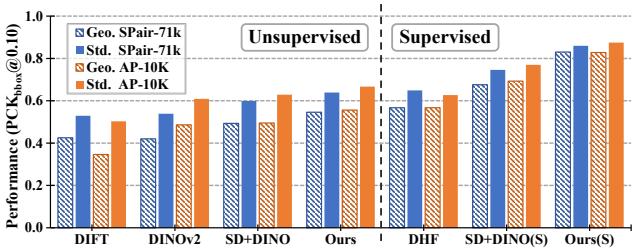
[1]Shanghai Jiao Tong University [2]Google Research [3]UIUC [4]Stability AI [5]UC Merced

Abstract

*While pre-trained large-scale vision models have shown significant promise for semantic correspondence, their features often struggle to grasp the geometry and orientation of instances. This paper identifies the importance of being geometry-aware for semantic correspondence and reveals a limitation of the features of current foundation models under simple post-processing. We show that incorporating this information can markedly enhance semantic correspondence performance with simple but effective solutions in both zero-shot and supervised settings. We also construct a new challenging benchmark for semantic correspondence built from an existing animal pose estimation dataset, for both pre-training validating models. Our method achieves a PCK@0.10 score of **64.2** (zero-shot) and **85.6** (supervised) on the challenging SPair-71k dataset, outperforming the state-of-the-art by 4.3p and 11.0p absolute gains, respectively. Our code and datasets will be publicly available at: https://telling-left-from-right.github.io.*



(a) The state-of-the-art method [51] fails at matching keypoints with geometric ambiguity, or, "telling left from right" (red solid lines).



(b) **The performance gap between geometry-aware set (Geo.) and standard set (Std.) of state-of-the-art methods.** The geometry-aware set accounts for **59.6%** and **45.7%** of the total keypoint pairs on SPair-71k [27] and AP-10K [50], respectively.

Figure 1. Illustration of geometry-aware correspondence.

1. Introduction

Since the advent of high fidelity text-to-image (T2I) generative models [34, 35] and large vision foundation models [31], there has been significant interest in understanding both what these models are learning and what they are not. Numerous works show that these models have powerful feature embeddings that can be used for many computer vision tasks including depth estimation [31, 52], semantic segmentation [41, 47], and semantic correspondences [2, 8, 12, 15, 26, 30, 51]. While many works have shown their strengths, less analysis has been done on their weaknesses; in particular, what do these features struggle with?

We propose using semantic correspondence as a promising test bed. Semantic correspondence, the establishment of pixel-level matches between two images with semantically similar objects, is an important Computer Vision problem with a variety of downstream applications, *e.g.*, image editing [8, 29, 30, 51] and style transfer [7, 20]. It also has many difficult challenges, *e.g.*, from large intra-class variations to different backgrounds, lighting, or viewpoints.

Despite these challenges, the large foundation model features currently achieve state-of-the-art performance [51]. However, a closer examination shows that this performance is inconsistent across all challenges. In particular, we find that these foundation model's features significantly underperform on "geometry-aware"[1] semantic correspondences: correspondences which share semantic properties but have different relations to the overall geometry of the object, *e.g.*, the "left" paw vs. the "right" paw as shown in Fig. 1a. Motivated by this, we conduct an in-depth analysis of these correspondences (Fig. 1b). We find that surprisingly,

[1]We use the term geometry in the loose sense and do not refer to 3D geometric properties, such as shape and surface normal.

such cases account for a significant portion of the benchmark datasets (nearly 60% in SPair71k), and state-of-the-art methods with the deep features perform considerably worse on this challenging subset (up to 30% worse, Fig. 1b).

Are these problems an innate failing of these features, or can they be alleviated through better post-processing? Based on the above observations, we propose several methods that resolve the geometric ambiguity during matching. First, we introduce a test-time viewpoint alignment strategy that approximately aligns viewpoints of instances to make the problem easier. Then we train a lightweight post-processing module that improves geometric awareness of features from visual foundation models [31, 34], by using a soft-argmax based dense training objective with given annotated sparse keypoints. We further introduce a pose-variant augmentation strategy as well as a window soft-argmax module. These not only significantly improve performance on standard benchmarks by 15% while costing only 0.32% of extra runtime.

For more advanced analysis, we create a new benchmark dataset using existing annotations from the AP-10K [50] animal pose estimation dataset. Compared to the largest existing benchmark [27], our new benchmark dataset includes 5 times more training pairs and is the first benchmark to evaluate cross-species and cross-families semantic correspondence. We also demonstrate that this benchmark can serve as a valuable pre-training resource for improving geometry-aware semantic correspondence.

To summarize, we make the following contributions:

- We identify the problem of geometry-aware semantic correspondence and show that pre-trained features of foundation models (SD [34] and DINOv2 [31]) struggle with geometric information.
- We propose to improve geometric awareness of the features in both unsupervised and supervised manners.
- We introduce a large-scale and challenging benchmark, AP-10K, for both training and evaluation.
- Our method boosts the overall performance on multiple benchmark datasets, especially on the geometry-aware correspondence subset. It achieves an 85.6 PCK@0.10 score on SPair-71k, outperforming the state-of-the-art method by more than 15%.

2. Related Work

Semantic correspondence. Conventional approaches to semantic correspondence estimation follow a common pipeline that consists of i) feature extraction [1, 6, 11, 23, 25, 37]), ii) cost volume computation [4, 13, 19, 28], and iii) matching field regression [18, 42–45]. To handle challenging intra-class variations between images, previous work have presented various approaches such as matching uniqueness prior [22], parameterized spatial prior [14, 17, 32, 33, 36], or end-to-end regression [4, 5, 14, 21, 43]. However, due to the limited capacity of their features or the usage of strong spatial prior, they still exhibit difficulties handling challenging intra-class variations such as large pose changes or non-rigid deformation.

Figure 2. Generated samples from SD-2-1 with the prompt (left) "A cat holding up its *left front paw*" and (right) "A car with the *right front door* open". SD has difficulty generating images that require understanding the intrinsic geometry of instances.

Recently, visual foundation models (*e.g.* DINO [3, 31] and SD [34]) demonstrate that their pretrained features, learned by self-supervised learning or generative tasks [12, 26, 39, 51], can serve as powerful descriptors for semantic matching by surpassing prior arts specifically designed for semantic matching. Yet, we reveal that such features still show limitations [8] in comprehending the intrinsic geometry of instances (*e.g.*, Fig. 2) and formally investigate this issue, termed "geometry-aware" semantic correspondence.

Benchmark datasets. Recent advances in semantic correspondence have continuously revealed limitations of existing benchmark datasets. For example, widely used datasets (PF-Pascal [9], PF-Willow [10], CUB-200-2011 [46], and TSS [40]) provide image pairs with only limited viewpoints or pose variations, making it hard to evaluate methods on handling large object viewpoint changes. The CUB dataset [46] provides images of a single object class, bird, only. SPair-71k [27] introduces a more challenging benchmark dataset that consists of 1,800 images across 18 object categories with substantial intra-class variations. While existing methods [5, 26, 51] have low performance on the SPair-71k dataset, the dataset is still small-scale. To address these shortcomings, we introduce a new, large-scale, and challenging benchmark using the animal pose estimation dataset, AP-10K [50]. This new benchmark further facilitates comprehensive evaluations of geometric awareness and provides annotations for training models.

3. Geometric Awareness of Deep Features

In this section, we first provide the clear problem definition of "geometry-aware semantic correspondence" as challenging cases of semantic correspondence, which requires an understanding of relations of similar semantic parts. Then we provide comprehensive analyses on the performance of pretrained features of foundation models on the problem and what geometric information those features possess.

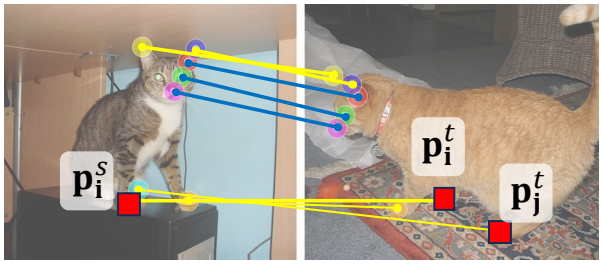


Figure 3. Annotations of geometry-aware semantic correspondence (yellow) and standard semantic correspondence (blue).

3.1. Geometry-Aware Semantic Correspondence

We define geometry-aware semantic correspondence as a challenging case of semantic correspondence, where there exist geometry-ambiguous matching cases, and thus it requires an understanding of instances' orientations or geometry. Fig. 1a illustrates the exemplar cases that require proper understanding of both semantic parts (*i.e.* paws) and their associations (*i.e.* left paw or right paw) with the orientation of instances.

As a formal definition, for each instance category, we first cluster keypoints into subgroups $\mathcal{G}_{parts}$ by their semantic parts. Each subgroup $\mathcal{G}_{parts}$ consists of a set of keypoints $\mathbf{p}_{(parts,\ index)}$ that fall into the same subgroup but position in different part locations according to their orientations. For the *cat* category as an example, the subgroups are $\mathcal{G}_{parts}$ with $parts$ = {ears, paws, ... }, and $\mathcal{G}_{\text{paws}} = \{\mathbf{p}_{(\text{paws, front left})}, \mathbf{p}_{(\text{paws, front right})}, \mathbf{p}_{(\text{paws, rear left})}, \mathbf{p}_{(\text{paws, rear right})}\}$.

Then, give a source $\mathbf{I}^s$ and a target image $\mathbf{I}^t$ that contains the same/similar instance category with their keypoint correspondence annotations, the correspondence $\langle \mathbf{p}_\mathbf{i}^s, \mathbf{p}_\mathbf{i}^t \rangle$ is considered as a "geometry-aware" correspondence if the two conditions are met. First, two keypoints $\mathbf{p}_\mathbf{i}^s$ and $\mathbf{p}_\mathbf{i}^t$ belong to the same subgroup, $\mathbf{p}_\mathbf{i}^s \in \mathcal{G}_{part}^s$ and $\mathbf{p}_\mathbf{i}^t \in \mathcal{G}_{part}^t$. Second, there are other visible keypoint(s) belonging to same subgroup in the target image, $\exists\, \mathbf{j} \neq \mathbf{i}$ s.t. $\mathbf{p}_\mathbf{j}^t \in \mathcal{G}_{part}^t$. As illustrated in Fig. 3, the front right paw ($\mathbf{p}_\mathbf{i}^s$) of the cat in the source image has several semantically similar correspondences, such as ($\mathbf{p}_\mathbf{j}^t$) and ($\mathbf{p}_\mathbf{i}^t$), which requires proper understanding of geometry to find the correct match.

3.2. Evaluation on the Geometry-aware Subset

We evaluate the state-of-the-art methods on geometry-aware semantic correspondence to see if their features are geometry-aware and how well they perform on this challenging task. From the challenging SPair-71k [27] datasets, we first cluster keypoint subgroups $\mathcal{G}_{parts}$ for each category and gather geometry-aware correspondence cases as the "geometry-aware subset". Surprisingly such cases account for a substantial portion, 82.4% of total image pairs and 59.6% of matching keypoints, of the dataset. (Please refer to Supplementary for more details.)

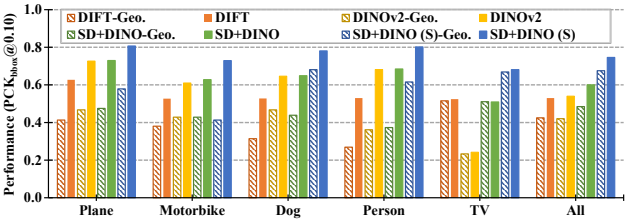


Figure 4. **Per-category evaluation of state-of-the-art methods on SPair-71k geometry-aware subset (Geo.) and standard set.** While the geometry-aware subset accounts for 60% of the total matching keypoints, we observe a substantial performance gap between the two sets for all the methods.

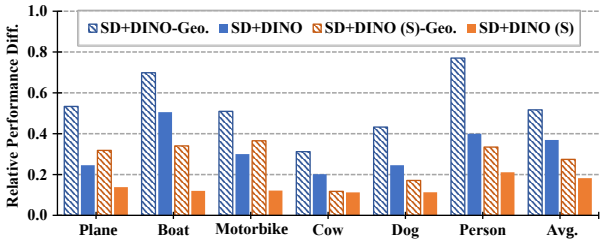


Figure 5. **Evaluation of the sensitivity to pose variations.** The y-axis shows the normalized difference between the best and the worst performance among 5 different azimuth-variation subsets. We report the results of the unsupervised and supervised methods on both the geometry-aware (Geo.) and standard set. The larger the value, the more sensitive the performance is to pose variation.

Fig. 4 shows the performance of the state-of-the-arts on the subset (in both zero-shot and supervised (S)). For all methods, there exists a substantial performance gap between the geometry-aware subset and the standard set, around 20% for zero-shot methods and still 10% for supervised methods. This reveals the weakness of current methods in matching keypoints where the geometry ambiguity is involved and the limitation on geometric awareness.

3.3. Sensitivity to Pose Variation

For certain categories, however, where the pose variation of the pair images is small (*e.g.*, potted plant and TV in SPair-71k), performance gaps on both the standard and geometry-aware sets are nearly marginal. This suggests that the pose variation is one of the key factors that affect the accuracy of geometry-aware correspondence. To delve deeper into it, we divide image pairs in SPair-71k into 5 subsets, based on their annotated azimuth differences, ranging from 0 (identical poses) to 4 (completely opposing directions). For each category, we then again evaluate the performance on these 5 subsets, $\mathcal{A} = \{\mathbf{a_0}, \mathbf{a_1}, \ldots, \mathbf{a_4}\}$ and define the normalized relative difference, $\mathbf{d} = \frac{\max(\mathcal{A}) - \min(\mathcal{A})}{\max(\mathcal{A})}$, which measures the sensitivity to pose variations. As shown in Fig. 5, the performance on the geometry-aware subset is more sensitive to the pose variation than the standard set across all categories, indicating that the pose variation affects the performance on

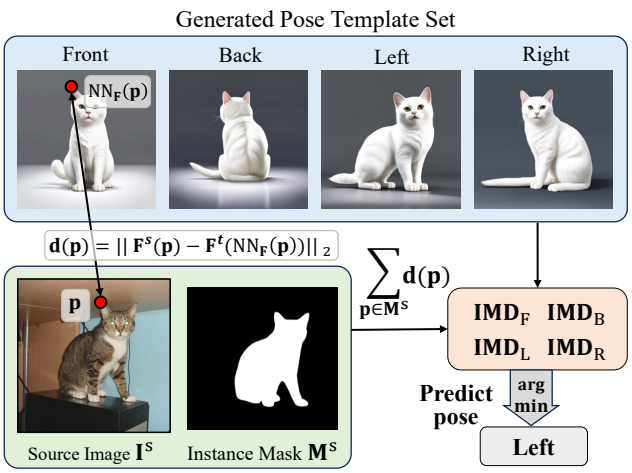


Figure 6. **Rough pose prediction with feature distance.** By computing the instance matching distance (IMD) of the source image to the generated pose templates, we can utilize the feature maps to predict rough pose and evaluate the global pose awareness of current deep features. We only show one template set for brevity.

geometry-aware semantic correspondence.

3.4. Global Pose Awareness of Deep Features

We further analyze if deep features are aware of high-level pose (or viewpoint) information of an instance in an image. We explore this pose awareness by a template-matching approach in the feature space.

Instance matching distance (IMD). We introduce this metric to examine pose prediction accuracy. Given a source $\mathbf{I}^s$ and target image $\mathbf{I}^t$, their normalized feature maps $\mathbf{F}^s$ and $\mathbf{F}^t$, and a source instance mask $\mathbf{M}^s$, we define the IMD metric as:

$$\text{IMD}(\mathbf{I}^s, \mathbf{I}^t, \mathbf{M}^s) = \sum_{\mathbf{p} \in \mathbf{M}^s} \|\mathbf{F}^s(\mathbf{p}) - \text{NN}(\mathbf{F}^s(\mathbf{p}), \mathbf{F}^t)\|_2, \quad (1)$$

where $\mathbf{p}$ denotes a pixel within the source instance mask, $\mathbf{F}^s(\mathbf{p})$ is the feature vector at $\mathbf{p}$, and $\text{NN}(\mathbf{F}^s(\mathbf{p}), \mathbf{F}^t)$ represents the nearest-neighboring feature vector in the target feature map. IMD measures the similarity of two images via the average feature distance of corresponding pixels.

Pose prediction via IMD. With the IMD metric, we can evaluate the global pose awareness of features from existing methods via pose prediction. We start by generating multiple pose template sets (in Fig. 6). We then compute the IMD between the input and template images for each set and predict the pose whose IMD is the smallest. A collective vote across all sets determines the final pose estimate.

We manually annotated 100 cat images from SPair-71k with pose labels {left, right, front, and back} and evaluate the pose prediction performance of the following deep features: DINOv2 [31], SD [34], and fused SD+DINO [51]. Due to some ambiguous cases for annotations, we also report the performance of binary classification into {left, right} or {front, back}. As in Tab. 1, DINOv2 struggles with left/right (L/R) distinction [8] but excels in front/back (F/B) prediction; SD performs well in both distinguishing L/R and F/B; and SD+DINO surpass both in all cases, achieving near-perfect results. This suggests that the deep features are aware of global pose information.

Table 1. **Zero-shot rough pose prediction result with IDM (Eq. (1)).** We report the accuracy of predicting left or right (L/R), front or back (F/B), the former two cases (L/R or F/B), and one of the four directions (L/R/F/B).

Feature	L/R	F/B	L/R or F/B	L/R/F/B
DINOv2	63.8	100.0	75.0	51.0
SD	95.7	96.8	96.0	78.0
SD+DINO	98.6	100.0	99.0	84.0

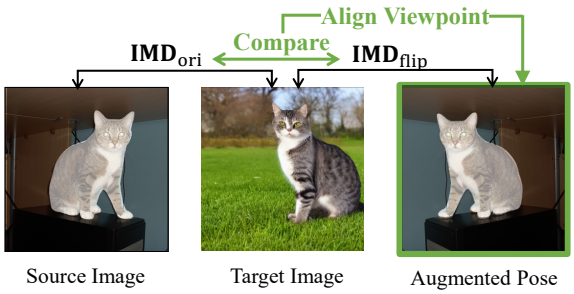


Figure 7. **Adaptive pose alignment.** By comparing the matching distance between the target image and augments of the source image, we can reduce the pose variation of pair images at test time for better correspondence.

4. Improving Geo-Aware Correspondence

We propose several techniques that improve geometric awareness during matching, in both zero-shot and supervised settings. We first introduce an adaptive pose alignment strategy that runs at test time without any training involved. Then, we further introduce a post-processing module with various training strategies that can improve the geometry awareness of deep features.

4.1. Test-time Adaptive Pose Alignment

In Sec. 3.3, we find that pose variations can largely affect the performance of geometry-aware semantic correspondence. To address this, we introduce a very simple test-time pose alignment strategy that utilizes the global pose information inherent in deep features (Sec. 3.4) and improves correspondence accuracy.

As in Fig. 7, we first augment the source image by using a set of pose-variant augmentations (*e.g.*, flip, rotations *etc.*), calculate the IMD (Eq. (1)) between the augmented source images and the target image, and choose the optimal pose with the minimum IMD distance. As in Fig. 9,

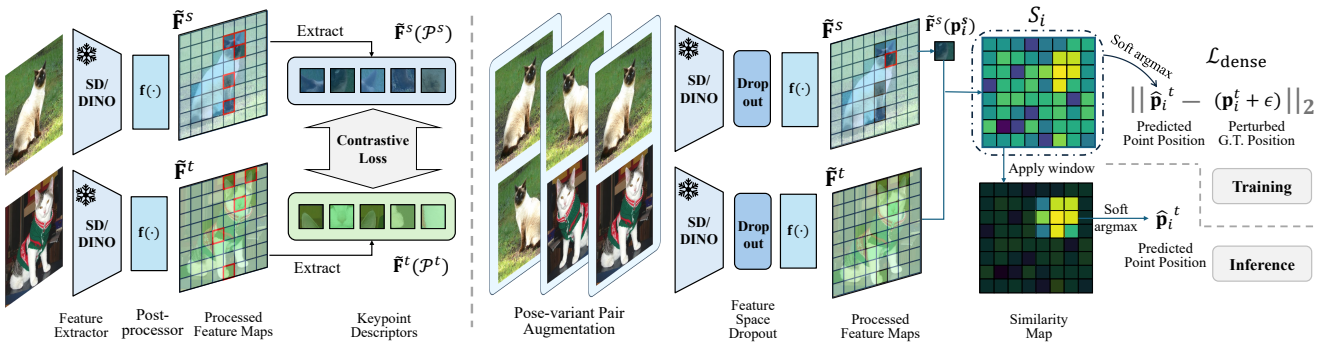


Figure 8. (Left) previous supervised methods [26, 51] with a sparse training objective. (Right) an overview of our supervised method. Only the lightweight post-processor is updated during training. Both the pair augmentation and feature space Dropout are for training only.

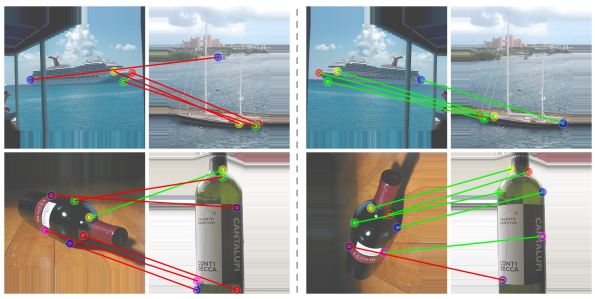


Figure 9. (Left) original image pairs. (Right) image pairs with the test-time aligned pose. The reduced pose variation improves the correspondence accuracy.

this simple pose alignment can drastically improve the correspondence accuracy in a test-time, unsupervised manner.

4.2. Dense Training Objective

Let $\mathbf{F}$ represent the raw feature map and $\mathrm{f}(\cdot)$ be the post-processing model that outputs the refined feature map $\tilde{\mathbf{F}} = \mathrm{f}(\mathbf{F})$, illustrated in Fig. 8. Given a set of annotated keypoint pairs from source images $\mathcal{P}^s = \{\mathbf{p}_1^s, \mathbf{p}_2^s, \ldots, \mathbf{p}_n^s\}$ and target images $\mathcal{P}^t = \{\mathbf{p}_1^t, \mathbf{p}_2^t, \ldots, \mathbf{p}_n^t\}$, previous works [26, 51] adopt a CLIP-style symmetric contrastive loss $\mathrm{CL}(\cdot, \cdot)$ to train the post-processing model:

$$\mathcal{L}_{\text{sparse}} = \mathrm{CL}(\tilde{\mathbf{F}}^s(\mathcal{P}^s), \tilde{\mathbf{F}}^t(\mathcal{P}^t)), \tag{2}$$

where $\tilde{\mathbf{F}}^s$ and $\tilde{\mathbf{F}}^t$ are the post-processed source and target features. However, the loss is applied only to features with sparsely annotated keypoints, which potentially neglects additional informative features.

Instead, we employ the soft-argmax operator [16, 19, 48] so that gradients calculated from sparse annotations can be back-propagated to all spatial locations. Specifically, we compute the similarity map $S_i = \tilde{\mathbf{F}}^s(\mathbf{p}_i^s)^T \tilde{\mathbf{F}}^t$ between the normalized query keypoint descriptor $\tilde{\mathbf{F}}^s(\mathbf{p}_i^s)$ and the target feature map $\tilde{\mathbf{F}}^t$. Then, we take soft-argmax over the similarity map to get the predicted position $\hat{\mathbf{p}}_i^t = \mathrm{SoftArgmax}(S_i)$. The L2 norm penalizes the distance between the predicted position and the target position $\hat{\mathbf{p}}_i^t$:

$$\mathcal{L}_{\text{dense}} = \sum_i \|\hat{\mathbf{p}}_i^t - (\mathbf{p}_i^t + \epsilon)\|_2, \tag{3}$$

To prevent overfitting, we also apply Dropout at the input feature map $\mathbf{F}$ and Gaussian noise ϵ that perturbs the ground truth keypoint positions $\mathbf{p}_i^t$. We empirically find that combining the two objectives achieves better performance; thus our final training objective is $\mathcal{L} = \mathcal{L}_{\text{dense}} + \mathcal{L}_{\text{sparse}}$.

4.3. Pose-variant Augmentation

Standard data augmentation schemes (*e.g.*, random scaling, cropping, color jittering, *etc.*) have been generally used to augment the limited number of annotated data. However, such standard augmentations show two shortcomings in naively adopting them to our approach. Diverse augmentations on input images require our model to process the feature map of each augmented image using visual foundation models, which linearly increases the computational cost along with the number of augmentation schemes used. Besides, such augmentations (*e.g.*, cropping, scaling, or photometric augmentations) do not augment images with different poses or viewpoints, which might not bring additional effective supervision signals for geometric awareness.

Instead, we introduce a set of pose-varying augmentation schemes tailored to our approach, which needs to process only one feature map from a single additional augmented image (horizontal flipped) yet can utilize the feature in multiple ways. The motivation is that the deep features are aware of the global pose; thus, the processed feature map of the flipped image can add an additional signal; compared to simply flipping the feature map. We introduce the following three augmentation settings: 1) *double flip*: flipped source image and flipped target image; 2) *single flip*: flipped source image and original target image; and 3) *self flip*: source image and flipped source image. For setting 2 and 3, keypoint annotations are correspondingly flipped to preserve the inherent geometric concept, *e.g.*, the left paw in the flipped image should be the right paw of the original image. The keypoint flipping in setting 3 also ensures that the model

Table 2. **Evaluation on SPair-71k.** Per-class and average PCK@0.10 on test split. The methods are categorized into two types: supervised (S) and unsupervised (U). $*$: fine-tuned backbone. †: index is used to flip source keypoints at test time. We report *per point* PCK result for the (U) methods, following [8, 30], and *per image* result for the (S) methods, following [5, 14, 21, 22]. The highest PCK are highlighted in **bold**, while the second highest are underlined. Both our zero-shot and supervised methods outperform prior arts across all categories.

	Method	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dog	Horse	Motor	Person	Plant	Sheep	Train	TV	**All**
U	NeuCongeal† [30]	-	29.1	-	-	-	-	-	53.3	-	-	35.2	-	-	-	-	-	-	-	-
	ASIC [8]	57.9	25.2	68.1	24.7	35.4	28.4	30.9	54.8	21.6	45.0	47.2	39.9	26.2	48.8	14.5	24.5	49.0	24.6	36.9
	DINOv2+NN [31, 51]	72.7	62.0	85.2	41.3	40.4	52.3	51.5	71.1	36.2	67.1	64.6	67.6	61.0	68.2	30.7	62.0	54.3	24.2	55.6
	DIFT [39]	63.5	54.5	80.8	34.5	46.2	52.7	48.3	77.7	39.0	76.0	54.9	61.3	53.3	46.0	57.8	57.1	71.1	63.4	57.7
	SD+DINO [51]	73.0	64.1	86.4	40.7	52.9	55.0	53.8	78.6	45.5	77.3	64.7	69.7	63.3	69.2	58.4	67.6	66.2	53.5	64.0
	Ours-Zero-Shot†	**77.5**	**65.4**	**88.4**	**44.1**	**59.8**	**65.4**	**60.4**	**81.4**	**53.2**	**80.6**	**67.7**	**72.3**	**65.1**	**70.6**	**58.6**	**70.0**	**83.0**	**54.5**	**68.5**
S	SCOT [22]	34.9	20.7	63.8	21.1	43.5	27.3	21.3	63.1	20.0	42.9	42.5	31.1	29.8	35.0	27.7	24.4	48.4	40.8	35.6
	PMNC* [21]	54.1	35.9	74.9	36.5	42.1	48.8	40.0	72.6	21.1	67.6	58.1	50.5	40.1	54.1	43.3	35.7	74.5	59.9	50.4
	SCorrSAN* [14]	57.1	40.3	78.3	38.1	51.8	57.8	47.1	67.9	25.2	71.3	63.9	49.3	45.3	49.8	48.8	40.3	77.7	69.7	55.3
	CATs++* [5]	60.6	46.9	82.5	41.6	56.8	64.9	50.4	72.8	29.2	75.8	65.4	62.5	50.9	56.1	54.8	48.2	80.9	74.9	59.8
	DHF [26]	74.0	61.0	87.2	40.7	47.8	70.0	74.4	80.9	38.5	76.1	60.9	66.8	66.6	70.3	58.0	54.3	87.4	60.3	64.9
	SD+DINO (S) [51]	81.2	66.9	91.6	61.4	57.4	85.3	83.1	90.8	54.5	88.5	75.1	80.2	71.9	77.9	60.7	68.9	92.4	65.8	74.6
	Ours	87.0	73.7	95.4	69.0	66.1	91.6	86.9	90.7	68.6	93.6	85.2	84.6	78.7	86.9	79.7	79.0	**96.9**	84.3	82.9
	Ours (Adapt. Pose)†	87.6	74.1	95.5	70.1	66.7	**92.0**	87.4	91.4	68.0	93.2	85.5	84.7	79.9	87.8	79.9	78.9	**96.9**	84.8	83.2
	Ours (AP-10K P.T.)	**92.0**	**76.1**	**97.2**	**70.4**	**70.5**	91.4	**89.7**	**92.7**	**73.4**	**95.0**	**90.5**	**87.7**	**81.8**	**91.6**	**82.3**	**83.4**	96.5	**85.3**	**85.6**

learns to discern concepts rather than simply matching keypoints based on appearances.

4.4. Window Soft Argmax

At test time, current methods [26, 51] use the argmax operation on the similarity map to infer correspondence. However, it shows two major limitations: argmax is limited to discrete pixel coordinates without sub-pixel reasoning, and it does not incorporate any spatial context with neighboring pixels when determining correspondence. One could use soft-argmax at time time too, but our study shows in Tab. 5 that it does not improve the performance on all metrics, probably due to its nature of incorporating similarities from all pixels with possible noisy response.

To complement the weaknesses of both, we propose a *window soft argmax* technique. First, we determine the target center location using the argmax operation and apply soft-argmax on the pre-defined window, as illustrated in Fig. 8. This hybrid approach naturally enables sub-pixel reasoning but also prevents it from being affected by any noisy response in the similarity map. Tab. 5 shows that the usage of window soft argmax substantially improves the correspondence performance on all metrics.

5. Experimental Results

Implementation details. We follow [51] to resize the input image to 960^2 and 840^2 to extract the SD and DINOv2 features, respectively, yielding a feature map at a resolution of 60×60. The post-processor on top of the fused features is four bottleneck layers [11] with 5M parameters in total. The model is trained with the AdamW optimizer [24] of weight decay rate 0.001 and the one-cycle scheduler [38] of 1.25×10^{-3} upper bound learning rate and 0.3 percentage for the increasing cycle. We train all our models on one NVIDIA RTX3090 GPU. Refer to Supp. for more details.

Datasets. We evaluate our methods on two widely-used benchmarks, namely PF-Pascal and SPair-71k, and our new proposed benchmark. *PF-Pascal* [9] consists of 2941 training, 308 validation, and 299 testing image pairs with similar viewpoints and instance pose. The images span across 20 categories of objects. *SPair-71k* [27] is a more challenging and larger-scale dataset with $53,340$ training pairs, $5,384$ validation pairs, and $12,234$ testing pairs across 18 categories, with large intra-class variation.

AP-10K benchmark. To further validate and improve our method in an in-the-wild setting, we build a new large-scale, challenging semantic correspondence benchmark with an existing animal pose estimation dataset. The AP-10K dataset [50] consists of 10,015 images across 23 families and 54 species. All the images share the same keypoint annotation of 17 keypoints. After manually filtering images with multiple instances and images with less than three visible keypoints, we construct a benchmark with 261k training, 17k validation, and 36k testing image pairs. The validation and testing image pairs span three settings: the main intra-species set, the cross-species set, and the cross-family set. It is 5× larger than the largest existing benchmark [27] and the first benchmark to evaluate cross-class semantic correspondence. Please refer to Supp. for more details.

Evaluation metrics. We follow the common practice and use the Percentage of Correct Keypoints (PCK) [49] to evaluate the correspondence accuracy. The PCK is computed

Table 3. **Evaluation on SPair-71k, AP-10K, and PF-Pascal datasets at different PCK levels.** We report the performance of the AP-10K intra-species (I.S.), cross-species (C.S.), and cross-family (C.F.) test sets. $*$: fine-tuned backbone. †: index is used to flip source keypoints at test time. We report the *per image* PCK results. The highest PCK among each category is highlighted in bold, while the second highest is underlined. Both our zero-shot and supervised methods outperform all previous methods significantly.

		SPair-71k			AP-10K-I.S.			AP-10K-C.S.			AP-10K-C.F.			PF-Pascal		
	Method	0.01	0.05	0.10	0.01	0.05	0.10	0.01	0.05	0.10	0.01	0.05	0.10	0.05	0.10	0.15
U	DINOv2+NN [31, 51]	6.3	38.4	53.9	6.4	41.0	60.9	5.3	37.0	57.3	4.4	29.4	47.4	63.0	79.2	85.1
	DIFT [39]	7.2	39.7	52.9	6.2	34.8	50.3	5.1	30.8	46.0	3.7	22.4	35.0	66.0	81.1	87.2
	SD+DINO [51]	7.9	44.7	59.9	7.6	43.5	62.9	6.4	39.7	59.3	5.2	30.8	48.3	**72.7**	**82.7**	**91.6**
	Ours-Zero-Shot†	**8.9**	**48.7**	**64.2**	**8.1**	**47.4**	**66.7**	**6.7**	**42.7**	**62.4**	**5.4**	**33.0**	**50.8**	72.5	82.6	91.5
S	SCorrSAN* [14]	3.6	36.3	55.3	-	-	-	-	-	-	-	-	-	81.5	93.3	96.6
	CATs++* [5]	4.3	40.7	59.8	-	-	-	-	-	-	-	-	-	84.9	93.8	96.8
	DHF [26]	8.7	50.2	64.9	8.0	45.8	62.7	6.8	42.4	60.0	5.0	32.7	47.8	78.0	90.4	94.1
	SD+DINO (S) [51]	9.6	57.7	74.6	9.9	57.0	77.0	8.8	53.9	74.0	6.9	46.2	65.8	80.9	93.6	96.9
	Ours	21.6	72.6	82.9	23.1	73.0	87.5	**21.7**	70.2	85.8	**18.4**	63.1	78.4	85.5	95.1	97.4
	Ours (Adapt. Pose)†	21.7	72.8	83.2	**23.2**	**73.2**	**87.7**	**21.7**	**70.3**	**85.9**	18.3	**63.2**	**78.5**	85.3	95.0	97.4
	Ours (AP-10K P.T.)	**22.0**	**75.3**	**85.6**	-	-	-	-	-	-	-	-	-	**85.9**	**95.7**	**98.0**

Table 4. **Evaluation on the geometry-aware subset.** We report the results on both SPair-71k and AP-10K intra-species test sets across three PCK levels. The best performances are **bold**.

		SPair-71k			AP-10K-I.S.		
	Method	0.01	0.05	0.10	0.01	0.05	0.10
U	DINOv2+NN [31, 51]	3.4	28.2	42.0	2.1	26.8	48.6
	DIFT [39]	4.6	30.0	42.5	1.8	18.9	34.6
	SD+DINO [51]	5.3	34.5	49.3	2.5	28.0	49.5
	Ours-Zero-Shot†	**6.3**	**39.6**	**55.0**	**3.2**	**33.8**	**55.6**
S	SCorrSAN* [14]	2.8	30.0	49.4	-	-	-
	CATs++* [5]	3.2	33.1	53.0	-	-	-
	DHF [26]	6.8	42.1	56.7	2.5	30.0	50.7
	SD+DINO (S) [51]	7.5	50.3	67.6	4.0	43.7	69.3
	Ours	18.2	66.0	77.4	10.4	64.8	82.8
	Ours (Adapt. Pose)†	18.3	66.3	78.0	**10.5**	**65.0**	**83.2**
	Ours (AP-10K P.T.)	**20.1**	**71.0**	**82.3**	-	-	-

within a threshold of $\alpha \cdot max(h, w)$ where α is a positive decimal (*e.g.*, 0.10) and (h, w) denotes the dimensions of the bounding box of an instance in SPair-71k and AP-10K, and the dimensions of the images in PF-Pascal, respectively.

5.1. Quantitative Analysis

Overall semantic correspondence. The per-category evaluation results, presented in Tab. 2, demonstrate the efficacy of our methods. Our *zero-shot* approach, despite its simplicity, achieves considerable gains over SD+DINO, highlighting the significance of pose alignment in semantic correspondence. In the *supervised* category, our methods outperform existing works across all 18 categories, registering a substantial improvement of **11.0p** (from 74.6 to 85.6). Notably, pre-training on the AP-10K dataset contributes a gain of 2.7p, underscoring the untapped potential of animal pose datasets in this domain.

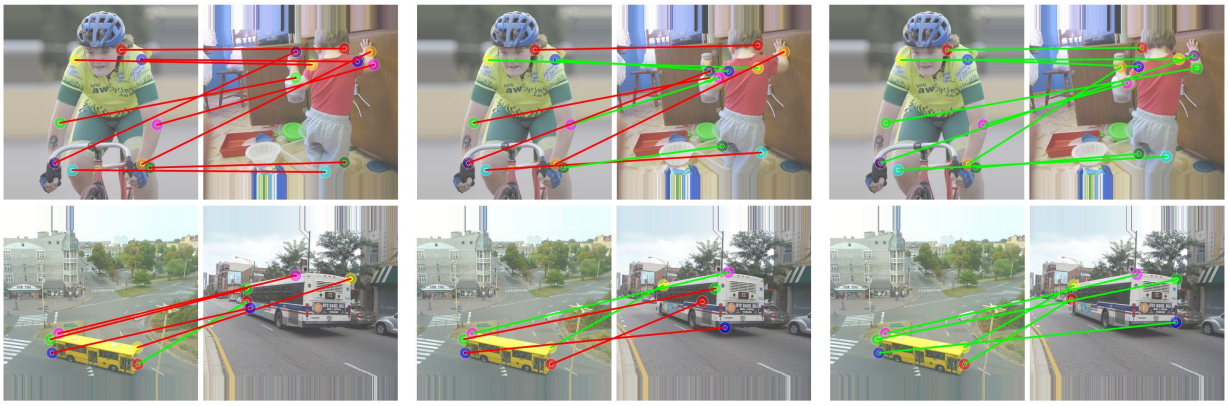
Further comparisons across different datasets and three PCK levels are in Tab. 3. Our methods exhibit significant improvements across most metrics, particularly with notable gains in the more strict thresholds (e.g., PCK@0.05 and PCK@0.01), especially considering that SD+DINO uses the same raw feature maps as our model. Despite the methods being trained only on AP-10K intra-species sets, the robust performance on cross-species and cross-family test sets showcases the generalizability of our approach.

Table 5. **Ablation study on SPair-71k.** We report the PCK@α_{bbox} results for both standard set (Std.) and geometry-aware set (Geo.). The best performances are **bold**. Our default method is underlined.

		SPair-71k (Std.)			SPair-71k (Geo.)		
	Model Variants	0.01	0.05	0.10	0.01	0.05	0.10
	Baseline	9.6	57.7	74.6	7.5	50.3	67.6
+	Dense Training Objective	13.0	65.2	78.3	11.1	58.8	71.9
+	Pose-variant Augmentation	13.8	66.7	80.0	11.4	60.5	73.9
+	Perturbation & Dropout	15.1	69.3	81.3	13.5	63.3	75.4
	Soft Argmax Inference	20.5	69.6	81.0	16.9	61.9	75.0
+	Window Soft Argmax (5)	**22.3**	72.1	82.0	**19.8**	66.0	76.5
+	Window Soft Argmax (9)	22.0	**72.7**	82.5	19.2	**66.3**	77.1
+	Window Soft Argmax (15)	21.6	72.6	**82.9**	18.2	66.0	**77.4**

Geometry-aware semantic correspondence. Our methods achieve even more significant improvements in the geometry-aware subset, as reported in Tab. 4. We reduce the relative gap from 9.38% (SD+DINO (S)) to 3.86% on the SPair-71k PCK@0.10 metric. Notably, the proposed adaptive viewpoint alignment brings more substantial gain on the geometry-aware subset for both zero-shot and supervised settings, suggesting its effectiveness in improving the geometric ambiguity by mitigating the pose variation. Besides, pre-training on the AP-10K dataset brings even a gain of 4.3p on the geo-aware subset.

Ablation studies. We perform further ablation studies in Tab. 5. Each element of our designs brings about mod-



SD+DINO SD+DINO (S) **Ours**

Figure 10. **Qualitative comparison.** Green lines indicate correct matches and red incorrect. Our method can build geometrically correct semantic correspondence even at extreme view variation, while both versions of SD+DINO struggle with geometric ambiguity (*e.g.*, ear and hands in the person example, corners in the bus example). Please refer to Supp. for more results.

Source Image SD+DINO SD+DINO (S) **Ours**

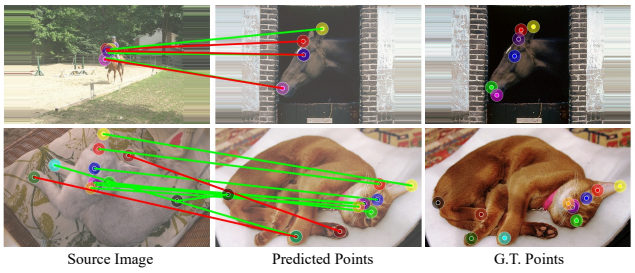
Figure 11. **Visualization of the similarity map.** For the red query point, SD+DINO tends to find appearance-similar points (wooden desk, floor); SD+DINO (S) returns a noisy similarity map due to the query point being out of supervision. Our methods can locate both semantically and geometrically correct points. The keypoint supervision of the "chair" category is in blue, though these images are not in the training set.

erate improvements. The dense training objective, pose-variant augmentation, and window soft argmax notably enhance results in the geometry-aware subset, while ground truth perturbation and feature map Dropout improve the overall correspondence (as shown in the similar gain on both sets). Regarding the window soft argmax, varying window sizes have different effects across three thresholds. We set the window size to 15×15 for optimal balance. Please refer to Supp. for more details.

5.2. Qualitative Analysis

We qualitatively compare our methods against both zero-shot and supervised versions of SD+DINO. As shown in Fig. 10, our approach significantly enhances semantic correspondence under the extreme view-variation cases. While additional supervision in SD+DINO does aid in keypoint localization to some extent, both versions of SD+DINO struggle with geometric ambiguity.

We further investigate cases where the query point lacks meaning and without direct supervision. As the visualization of the similarity map shown in the Fig. 11, SD+DINO highlights the regions with similar appearance (wooden materials) but fails to locate the chair; SD+DINO (S) generates noisy similarity maps when the query point is out of supervision, due to the sparse training objective; Our method locates the points both semantically and geometrically correct. Notably, all methods utilize the same raw feature maps, and our approach employs the same feature post-processor as SD+DINO (S). Despite this, the notable improvements in our method further underscore the efficacy of our design.



Source Image Predicted Points G.T. Points

Figure 12. **Limitations.** Top: small instance. Bottom: scenarios combining both large pose variation and severe deformation.

Limitations. As shown in Fig. 12 (top), small instances may be challenging for our method due to the resolution limits of raw feature maps. Our method may fail on extreme pose variations with severe deformation (see Fig. 12, bottom). Future work may address these complex scenarios by advanced reasoning mechanisms.

6. Conclusion

We have identified the problem of geometry ambiguity in semantic correspondence and introduced simple and effective techniques to improve current methods in both the zero-shot and supervised settings. We have also developed a new benchmark to train and validate existing methods. Extensive experiments demonstrate that our method not only significantly improves the overall semantic correspondence but also narrows the gap between the geometry-aware sub-set and the standard set, thereby benefiting various downstream tasks and providing another angle to understand the internal representation of foundation models.

References

[1] Kfir Aberman, Jing Liao, Mingyi Shi, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Neural best-buddies: Sparse cross-domain correspondence. *ACM TOG*, 37(4):1–14, 2018. 2

[2] 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. 1

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 2

[4] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Cats: Cost aggregation transformers for visual correspondence. In *NeurIPS*, pages 9011–9023, 2021. 2

[5] Seokju Cho, Sunghwan Hong, and Seungryong Kim. Cats++: Boosting cost aggregation with convolutions and transformers. *IEEE TPAMI*, 2022. 2, 6, 7, 16

[6] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893. Ieee, 2005. 2

[7] Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 1

[8] Kamal Gupta, Varun Jampani, Carlos Esteves, Abhinav Shrivastava, Ameesh Makadia, Noah Snavely, and Abhishek Kar. Asic: Aligning sparse in-the-wild image collections. In *ICCV*, 2023. 1, 2, 4, 6

[9] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow. In *CVPR*, pages 3475–3484, 2016. 2, 6

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

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2, 6

[12] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised semantic correspondence using stable diffusion. In *NeurIPS*, 2023. 1, 2

[13] Sunghwan Hong, Seokju Cho, Jisu Nam, Stephen Lin, and Seungryong Kim. Cost aggregation with 4D convolutional swin transformer for few-shot segmentation. In *ECCV*, pages 108–126. Springer, 2022. 2

[14] Shuaiyi Huang, Luyu Yang, Bo He, Songyang Zhang, Xuming He, and Abhinav Shrivastava. Learning semantic correspondence with sparse annotations. In *ECCV*, pages 267–284. Springer, 2022. 2, 6, 7, 16

[15] 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. 1

[16] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, pages 66–75, 2017. 5

[17] Seungryong Kim, Stephen Lin, Sang Ryul Jeon, Dongbo Min, and Kwanghoon Sohn. Recurrent transformer networks for semantic correspondence. In *NeurIPS*, 2018. 2

[18] Seungryong Kim, Dongbo Min, Somi Jeong, Sunok Kim, Sangryul Jeon, and Kwanghoon Sohn. Semantic attribute matching networks. In *CVPR*, pages 12339–12348, 2019. 2

[19] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. SFNet: Learning object-aware semantic correspondence. In *CVPR*, pages 2278–2287, 2019. 2, 5, 14, 15

[20] Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *CVPR*, pages 5801–5810, 2020. 1

[21] Jae Yong Lee, Joseph DeGol, Victor Fragoso, and Sudipta N. Sinha. Patchmatch-based neighborhood consensus for semantic correspondence. In *CVPR*, pages 13153–13163, 2021. 2, 6

[22] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *CVPR*, pages 4463–4472, 2020. 2, 6

[23] Jonathan L. Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *NeurIPS*, 2014. 2

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[25] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157. Ieee, 1999. 2

[26] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *NeurIPS*, 2023. 1, 2, 5, 6, 7, 16

[27] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv prepreint arXiv:1908.10543*, 2019. 1, 2, 3, 6, 12

[28] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Learning to compose hypercolumns for visual correspondence. In *ECCV*, pages 346–363. Springer, 2020. 2

[29] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. DragonDiffusion: Enabling drag-

style manipulation on diffusion models. *arXiv preprint arXiv:2307.02421*, 2023. 1

[30] Dolev Ofri-Amar, Michal Geyer, Yoni Kasten, and Tali Dekel. Neural congealing: Aligning images to a joint semantic atlas. In *CVPR*, pages 19403–19412, 2023. 1, 6

[31] 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, 4, 6, 7

[32] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, pages 6148–6157, 2017. 2

[33] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *CVPR*, pages 6917–6925, 2018. 2

[34] 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, 4

[35] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, pages 36479–36494, 2022. 1

[36] Paul Hongsuck Seo, Jongmin Lee, Deunsol Jung, Bohyung Han, and Minsu Cho. Attentive semantic alignment with offset-aware correlation kernels. In *ECCV*, pages 349–364, 2018. 2

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[38] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, pages 369–386. SPIE, 2019. 6

[39] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. In *NeurIPS*, 2023. 2, 6, 7

[40] Tatsunori Taniai, Sudipta N. Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *CVPR*, pages 4246–4255, 2016. 2

[41] Junjiao Tian, Lavisha Aggarwal, Andrea Colaco, Zsolt Kira, and Mar Gonzalez-Franco. Diffuse, attend, and segment: Unsupervised zero-shot segmentation using stable diffusion. *arXiv preprint arXiv:2308.12469*, 2023. 1

[42] Prune Truong, Martin Danelljan, Luc V. Gool, and Radu Timofte. Gocor: Bringing globally optimized correspondence volumes into your neural network. In *NeurIPS*, pages 14278–14290, 2020. 2

[43] Prune Truong, Martin Danelljan, and Radu Timofte. Glu-net: Global-local universal network for dense flow and correspondences. In *CVPR*, pages 6258–6268, 2020. 2

[44] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *CVPR*, pages 5714–5724, 2021.

[45] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Warp consistency for unsupervised learning of dense correspondences. In *ICCV*, pages 10346–10356, 2021. 2

[46] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD birds-200-2011 dataset. 2011. 2

[47] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, pages 2955–2966, 2023. 1, 11

[48] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. *NeurIPS*, 32, 2019. 5

[49] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE TPAMI*, 35(12):2878–2890, 2012. 6

[50] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. Ap-10k: A benchmark for animal pose estimation in the wild. In *NeurIPS*, 2021. 1, 2, 6

[51] 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. In *NeurIPS*, 2023. 1, 2, 4, 5, 6, 7, 11, 14, 15, 16

[52] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *ICCV*, 2023. 1

Telling Left from Right: Identifying Geometry-Aware Semantic Correspondence

Supplementary Material

Contents

A. Further Implementation Details

Feature extraction. The extraction of SD and DINOv2 features is conducted in a manner similar to that described in Zhang *et al*. [51]. Specifically, the SD features are extracted from SD-1-5's UNet decoder layer 2, 5, and 8 at timestep 50 with an implicit captioner, and the DINOv2 features are extracted from the token facet of the 11th layer.

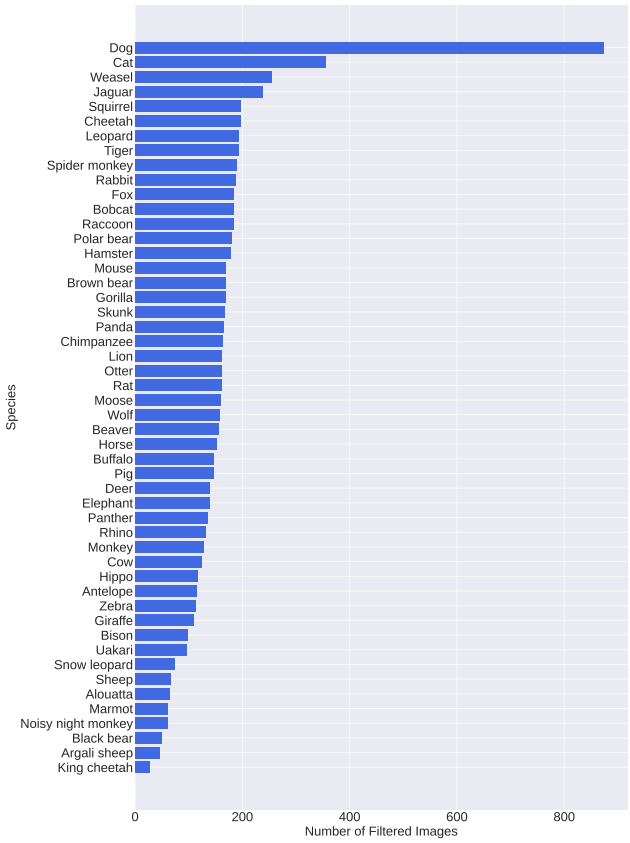


Figure 13. **Distribution of the filtered images across different species.** Note that only 50 species have annotated images.

Adaptive viewpoint alignment. For adaptive viewpoint (or pose) alignment in Sec. 4.1 in the main paper, we utilize segmentation masks from ODISE [47] to calculate the Instance Matching Distance (IMD). Considering the imbalanced viewpoint distribution in the images, "horizontal flip" is employed as the primary viewpoint augmentation for all categories. Specifically for the bottle category, to accommodate its unique viewpoint variations, we further apply rotations of +90°, 180°, and -90° as additional augmented viewpoints.

Pose-variant augmentation. In terms of pose-variant augmentation, we compute all the pair augmentations in a single batch and assign weights of 1 for both *single-flip* and *double-flip*, and a weight of 0.25 for the *self-flip*. Note that pose-variant augmentation is not applied during training on the PF-Pascal dataset due to all image pairs in this dataset are of similar pose.

Training. Our model is trained for 100k steps (equivalent to

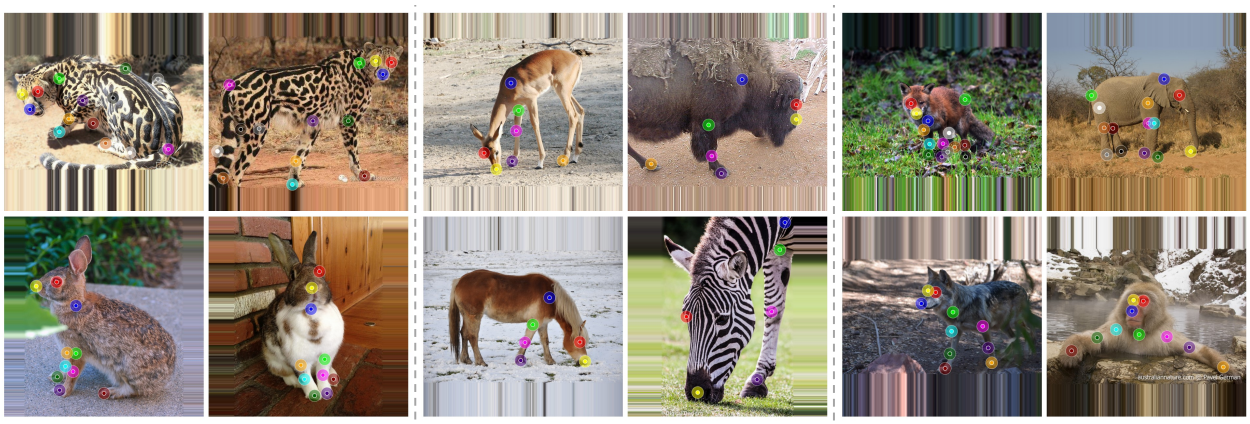


Figure 14. **Sample image pairs of AP-10K benchmark** including intra species, cross species, and cross family.

2 epochs) on the SPair-71k dataset, and 250k steps on AP-10K (equivalent to 1 epoch) and PF-Pascal (equivalent to 85 epochs), with a mini-batch size of 1. For a faster training, we pre-extract features from the visual foundation models offline and only train the post-processor online. This strategy significantly reduces the training duration, allowing it to be completed within just a few hours on a single GPU.

B. Benchmarking AP-10K Dataset for Semantic Correspondence

Image filtering. To start with, we exclude images with fewer than three visible keypoints or with multiple instances of the target category, to make the dataset less ambiguous for semantic matching.

Train/validation/test sets. After the filtering, there exists an imbalance in the number of images per species within the AP-10K dataset, as illustrated in Fig. 13. To ensure a balanced evaluation across different species, we uniformly sample an equivalent number of images for validation and test sets across all species — specifically, $N_{\text{val}} = 20$ for validation and $N_{\text{test}} = 30$ for testing, in line with the protocol established by SPair-71k [27]. The remaining images constitute the training set. It is important to note that for these three species, king cheetah, argali sheep, and black bear, whose numbers of images after the filtering are below 50, we earmark these as a hold-out set without including them in the training set. Thereby, it can also provide a measure for evaluating the generalization capability of semantic correspondence methods.

Intra-species image pair sampling. For each species, we construct all possible image matching pairs within each validation and test set (*i.e.*, $\binom{N_{\text{val}}}{2}$ and $\binom{N_{\text{test}}}{2}$) that are established in the previous step. On the other hand, the training set exhibits a more significant variance in the number of images; to circumvent the unbalanced distribution that arises from quadratic pairing growth, we limit the pairing to a maximum of either $50 \times N_{\text{train}}$ or $\binom{N_{\text{train}}}{2}$ pairs, whichever is fewer. Considering that the AP-10K dataset was not initially curated for the task of semantic correspondence, we apply an additional filtration criterion to the image pairs, retaining only those with a minimum of three mutual visible keypoints. This results in a total number of 260,950 training, 8816 validation, and 20,630 testing image pairs.

Cross-species and cross-family image pair sampling. We also include correspondence matching pairs across different species and families. For all 11 families with multiple species, we sample $\binom{N_{\text{val}}}{1} \cdot \binom{N_{\text{val}}}{1}$ validation pairs and $\binom{N_{\text{test}}}{1} \cdot \binom{N_{\text{test}}}{1}$ testing pairs for each family. For the cross-family setting, among all the $\binom{21}{2}$ combination of the total of 21 families, we only sample N_{val} validation and N_{test} testing pairs to save compute. A filtering process based on the mutually visible keypoints is also applied, yielding a total number of 4300 and 4200 validation pairs, alongside 9619 and 6300 testing pairs for cross-species and cross-family correspondence, respectively. Please refer to Fig. 14 for sample image pairs.

C. Details on Geo-Aware Correspondence

Keypoint subgroups. We list the keypoint subgroups of each category in Tab. 6. We exclude very few parts (nostril, eyes, *etc*.) that are close to each other and thus cannot be easily distinguished by existing metrics. We suggest that an improved metric (*e.g.*, a keypoint can be only regarded as a prediction to its nearest ground truth point) can make up this issue.

Per-category proportion. We show the average proportion of the geometry-aware subset with respect to both image pairs and keypoint pairs for each category in Fig. 15. For

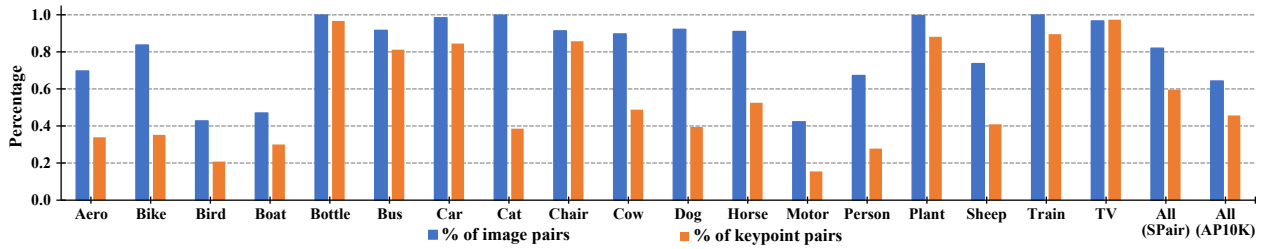


Figure 15. **Proportion of the geometry-aware subset with respect to image pair and keypoint pair.** We show the per-category results of SPair-71k as well as the average results of SPair-71k and AP-10K intra-species set.

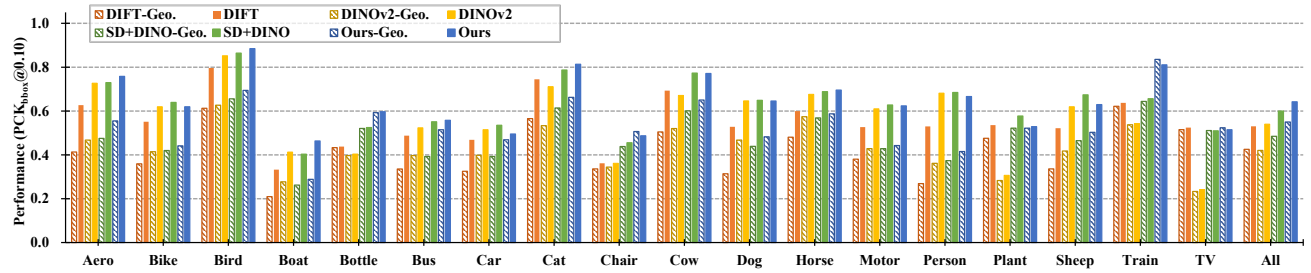
Table 6. **Semantically similar keypoint subgroups.** We list the keypoint subgroups for categories from both SPair-71k and AP-10K. The number in the bracket indicates the number of keypoints in each subgroup. The annotation in the index version will also be released.

Dataset	Category	Subgroups
SPair-71k	Aeroplane	$\mathcal{G}_{\text{landing\_gear}}$ (2), $\mathcal{G}_{\text{engine\_front}}$ (2), $\mathcal{G}_{\text{wing\_end}}$ (2), $\mathcal{G}_{\text{engine\_back}}$ (2), $\mathcal{G}_{\text{wing\_foot\_front}}$ (2), $\mathcal{G}_{\text{wing\_foot\_back}}$ (2), $\mathcal{G}_{\text{tailplane\_end}}$ (2), $\mathcal{G}_{\text{tailplane\_foot\_front}}$ (2), $\mathcal{G}_{\text{tailplane\_foot\_back}}$ (2)
	Bicycle	$\mathcal{G}_{\text{handle}}$ (2), $\mathcal{G}_{\text{seat\_back\_end}}$ (2), $\mathcal{G}_{\text{pedal}}$ (2)
	Bird	$\mathcal{G}_{\text{wing\_end}}$ (2), $\mathcal{G}_{\text{foot}}$ (2), $\mathcal{G}_{\text{knee}}$ (2), $\mathcal{G}_{\text{hip}}$ (2)
	Boat	$\mathcal{G}_{\text{upper\_front}}$ (2), $\mathcal{G}_{\text{upper\_side}}$ (2), $\mathcal{G}_{\text{upper\_back}}$ (2), $\mathcal{G}_{\text{lower\_front}}$ (2), $\mathcal{G}_{\text{lower\_side}}$ (2), $\mathcal{G}_{\text{lower\_back}}$ (2)
	Bottle	$\mathcal{G}_{\text{cap}}$ (2), $\mathcal{G}_{\text{neck}}$ (2), $\mathcal{G}_{\text{shoulder}}$ (2), $\mathcal{G}_{\text{body}}$ (2), $\mathcal{G}_{\text{base}}$ (2)
	Bus	$\mathcal{G}_{\text{rearview\_mirror}}$ (2), $\mathcal{G}_{\text{light}}$ (2), $\mathcal{G}_{\text{licence\_plate}}$ (2), $\mathcal{G}_{\text{front\_fender}}$ (4), $\mathcal{G}_{\text{wheel}}$ (4), $\mathcal{G}_{\text{rear\_fender}}$ (4), $\mathcal{G}_{\text{window\_top\_corner}}$ (4), $\mathcal{G}_{\text{window\_bottom\_corner}}$ (4)
	Car	$\mathcal{G}_{\text{rearview\_mirror}}$ (2), $\mathcal{G}_{\text{light}}$ (2), $\mathcal{G}_{\text{licence\_plate}}$ (2), $\mathcal{G}_{\text{brand\_logo}}$ (2), $\mathcal{G}_{\text{rear\_fender}}$ (4), $\mathcal{G}_{\text{wheel}}$ (4), $\mathcal{G}_{\text{front\_fender}}$ (4), $\mathcal{G}_{\text{window\_bottom\_corner}}$ (4), $\mathcal{G}_{\text{window\_top\_corner}}$ (4)
	Cat	$\mathcal{G}_{\text{ear}}$ (2), $\mathcal{G}_{\text{paw}}$ (4)
	Chair	$\mathcal{G}_{\text{cushion\_front}}$ (2), $\mathcal{G}_{\text{cushion\_back}}$ (2), $\mathcal{G}_{\text{leg}}$ (4), $\mathcal{G}_{\text{backrest\_top}}$ (2), $\mathcal{G}_{\text{armrest\_front}}$ (2), $\mathcal{G}_{\text{armrest\_back}}$ (2)
	Cow	$\mathcal{G}_{\text{ear}}$ (2), $\mathcal{G}_{\text{hoof}}$ (4), $\mathcal{G}_{\text{knee}}$ (4), $\mathcal{G}_{\text{horn}}$ (2)
	Dog	$\mathcal{G}_{\text{ear}}$ (2), $\mathcal{G}_{\text{paw}}$ (4)
	Horse	$\mathcal{G}_{\text{ear}}$ (2), $\mathcal{G}_{\text{hoof}}$ (4), $\mathcal{G}_{\text{knee}}$ (4)
	Motorbike	$\mathcal{G}_{\text{rearview\_mirror}}$ (2), $\mathcal{G}_{\text{handle}}$ (2)
	Person	$\mathcal{G}_{\text{shoulder}}$ (2), $\mathcal{G}_{\text{elbow}}$ (2), $\mathcal{G}_{\text{wrist}}$ (2), $\mathcal{G}_{\text{knee}}$ (2), $\mathcal{G}_{\text{ankle}}$ (2), $\mathcal{G}_{\text{foot}}$ (2)
	Pottedplant	$\mathcal{G}_{\text{top}}$ (4), $\mathcal{G}_{\text{side\_wall}}$ (2), $\mathcal{G}_{\text{bottom}}$ (2)
	Sheep	$\mathcal{G}_{\text{ear}}$ (2), $\mathcal{G}_{\text{hoof}}$ (4), $\mathcal{G}_{\text{knee}}$ (4), $\mathcal{G}_{\text{horn}}$ (2)
	Train	$\mathcal{G}_{\text{front\_top}}$ (2), $\mathcal{G}_{\text{front\_bottom}}$ (2), $\mathcal{G}_{\text{back\_top}}$ (2), $\mathcal{G}_{\text{back\_bottom}}$ (2), $\mathcal{G}_{\text{window\_top\_outer\_corner}}$ (2), $\mathcal{G}_{\text{window\_bottom\_outer\_corner}}$ (2), $\mathcal{G}_{\text{window\_top\_inner\_corner}}$ (2), $\mathcal{G}_{\text{window\_bottom\_inner\_corner}}$ (2), $\mathcal{G}_{\text{front\_light}}$ (2)
	Tvmonitor	$\mathcal{G}_{\text{outer\_corner}}$ (4), $\mathcal{G}_{\text{outer\_side}}$ (4), $\mathcal{G}_{\text{inner\_corner}}$ (4), $\mathcal{G}_{\text{inner\_side}}$ (4)
AP-10K	All	$\mathcal{G}_{\text{shoulder}}$ (2), $\mathcal{G}_{\text{foot}}$ (4), $\mathcal{G}_{\text{knee}}$ (4), $\mathcal{G}_{\text{hip}}$ (2)

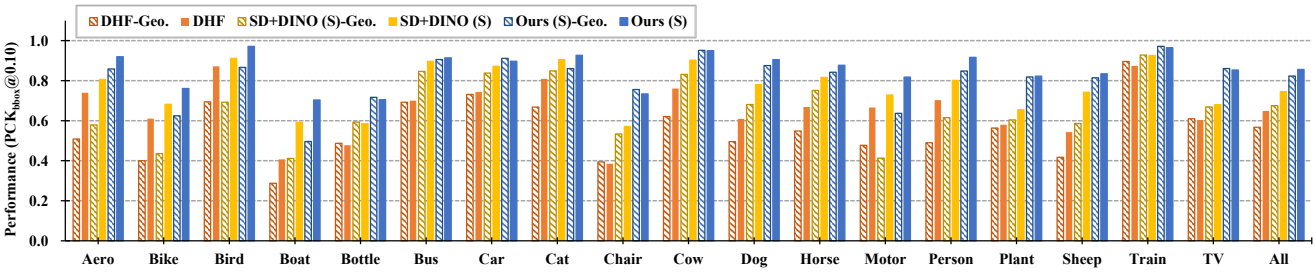
most of the categories, the geometry-aware subset accounts for a considerable fraction of all pairs.

Notably, due to the unbalanced pose distribution exhibited in specific categories of the SPair-71k (*e.g.* bottles, potted plants, TVs, and trains) where image pairs often share similar poses, almost all keypoint subgroups in these categories are mutually visible, which results in proportions to be near 100%. In contrast, the AP-10K dataset, comprised solely of animal images, does not exhibit this imbalance.

Per-category performance. In Fig. 16a and Fig. 16b, we provide detailed per-category performance for both unsupervised and supervised state-of-the-art methods on the geometry-aware subset and the standard set. These figures provide an expanded view of Fig. 4 from the main paper.



(a) **Performance of the unsupervised methods.**



(b) **Performance of the supervised methods.**

Figure 16. **Per-category performance of the state-of-the-art methods and ours (blue).** We report both the geometry-aware subset (Geo.) and the standard set on SPair-71k. Our methods consistently outperform previous arts across all categories.

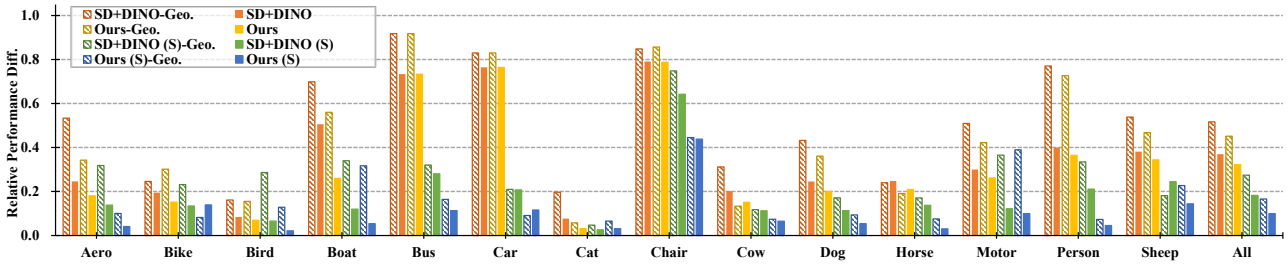


Figure 17. **Per-category evaluation of the sensitivity to pose variations.** Both our zero-shot (yellow) and supervised methods (blue) considerably improve the robustness to pose variations on both the geometry-aware set (Geo., hashed bar) and the standard set (solid bar) compared to the state-of-the-art methods [51]. We exclude categories that only have one azimuth-variation subset.

Regardless of the method or category, performance on the geometry-aware subset consistently lags behind that of the standard set.

Additionally, in Fig. 17, we offer a per-category analysis of pose variation sensitivity. The results for both unsupervised and supervised variants of SD+DINO [51] are presented, comparing their performance on both the geometry-aware and standard sets. This analysis serves as an extended version of Fig. 5 from the main paper. The findings clearly show that sensitivity to pose variation is considerably higher in the geometry-aware subset across all categories and methodologies.

D. Additional Analysis

D.1. Detailed Performance on Geo-Aware Subset

We provide the per-category performance on the geometry-aware subset in Fig. 16 as well as the pose-sensitivity analysis of our methods in Fig. 17.

D.2. Detailed Analysis on Window Soft Argmax

Performance in accordance with window size. We evaluate the effect of soft-argmax's window size on the performance at different PCK thresholds. As depicted in Fig. 18, the performance across all PCK levels initially improves and then declines as the window size increases from 0 (hard argmax) to 60 (soft argmax). Notably, the peak PCK values for 0.01, 0.05, and 0.1 are observed at window sizes of 5, 11, and 17, respectively. We opt for a window size of 15 to achieve an optimal balance in performance.

Comparison with Gaussian kernel soft argmax. Previous work [19] also explored a trade-off solution between hard and soft argmax by applying a Gaussian kernel on the feature map, centered at the hard argmax position.

We also search different σ values for the Gaussian kernel

Table 7. **Effect of window soft argmax on zero-shot semantic correspondence performance.** We report the PCK@α_{bbox} results on both the standard set (Std.) and geometry-aware set (Geo.) of SPair-71k. The best performances are **bold**.

		SPair-71k (Std.)			SPair-71k (Geo.)		
Method Variants	Inference Strategy	0.01	0.05	0.10	0.01	0.05	0.10
SD+DINO [51]	Argmax Inference (Default)	7.9	44.7	59.9	5.3	34.5	49.3
	Soft Argmax Inference	6.4	36.5	53.7	6.4	36.5	53.7
	Window Soft Argmax (3)	**10.0**	45.9	60.1	**6.7**	35.5	49.6
	Window Soft Argmax (5)	9.9	**46.3**	60.5	6.6	**35.8**	50.1
	Window Soft Argmax (11)	8.7	45.3	**61.3**	5.5	34.3	**51.1**
Ours-zero-shot	Argmax Inference (Default)	8.9	48.7	64.2	6.3	39.6	55.0
	Soft Argmax Inference	7.6	40.7	58.4	4.1	29.0	48.2
	Window Soft Argmax (3)	**11.2**	49.7	64.3	**8.3**	40.8	55.4
	Window Soft Argmax (5)	11.1	**50.1**	61.8	8.1	**41.1**	56.0
	Window Soft Argmax (11)	9.9	49.1	**65.4**	6.9	39.5	**56.8**

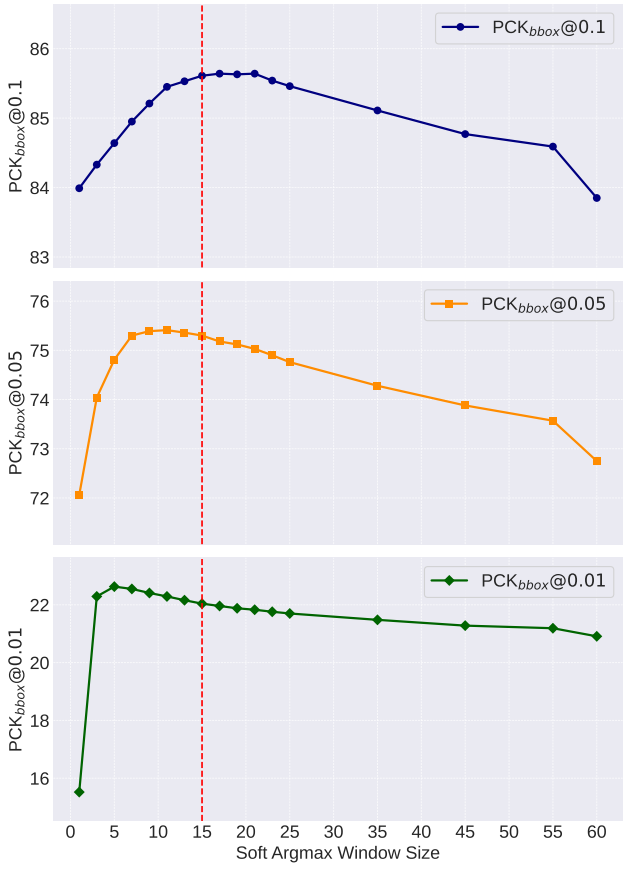


Figure 18. **Performance of different PCK levels vs. soft argmax window size.** We test the performance on the SPair-71k dataset and set the window size as 15 for optimal balance.

to achieve the best performance across different PCK levels. We then compare our window soft argmax with the kernel soft argmax in Tab. 8 on different peak PCK levels and the default value as reported in [19]. Our window soft argmax consistently outperforms kernel argmax across all settings, suggesting the superiority of our approach. We hypothesize that this is because when using the argmax-centered Gaussian kernel to scale the similarity map, it makes the similarity map biased to argmax locations, while our method treats the window region with the same scale.

Table 8. **Comparison with Gaussian kernel soft argmax on SPair-71k.** Default and peak values for each PCK level are reported for both methods, with the best results **bolded**.

Setting	Method	PCK@0.01	PCK@0.05	PCK@0.10
Default	Kernel	19.7	73.5	84.3
	Window	22.0	75.3	85.6
Best PCK@0.01	Kernel	22.4	75.0	84.9
	Window	**22.6**	75.3	85.0
Best PCK@0.05	Kernel	22.3	75.3	85.3
	Window	22.3	**75.4**	85.5
Best PCK@0.10	Kernel	21.9	75.1	85.5
	Window	22.0	75.2	**85.7**

Table 9. **Effect of applying window soft argmax during training.** We train all the post-processors on SPair-71k for one epoch and from scratch. The best results are **bolded**.

Setting	PCK@0.01	PCK@0.05	PCK@0.10
Window Soft Argmax (7)	21.4	69.0	79.0
Window Soft Argmax (15)	21.8	69.5	80.1
Window Soft Argmax (22)	**22.1**	70.3	81.2
Soft Argmax	20.4	**70.8**	**82.1**

Training with window soft argmax. We also experiment if applying the window soft argmax during training is beneficial. As shown in Tab. 9, applying window soft argmax during training hurts the PCK performances with the loose thresholds, while helping the stricter threshold

Table 10. **Generalizability test with training on PF-PASCAL.** We test the generalizability of our method by training the model on the PF-PASCAL dataset and testing on the SPair-71k and AP-10K intra-species (I.S.) test set. The best results are **bold**.

	SPair-71k			AP-10K-I.S.		
Method	0.01	0.05	0.10	0.01	0.05	0.10
SCorrSAN [14]	1.5	18.4	32.7	-	-	-
CATs++ [5]	2.1	19.7	32.0	-	-	-
DHF [26]	4.6	30.1	41.8	7.3	37.0	49.1
SD+DINO (S) [51]	**5.3**	34.1	46.9	8.2	43.4	59.2
Ours	**5.3**	**37.1**	**54.3**	**10.1**	**44.0**	**62.5**

Table 11. **Generalizability test with training on SPair-71k.** We test the generalizability of our method by training the model on the SPair-71k dataset and testing on the PF-PASCAL and AP-10K intra-species (I.S.) test set. The best results are **bold**.

	PF-PASCAL			AP-10K-I.S.		
Method	0.05	0.10	0.15	0.01	0.05	0.10
SCorrSAN [14]	54.5	71.2	78.8	-	-	-
CATs++ [5]	54.8	68.7	76.1	-	-	-
DHF [26]	64.2	77.8	84.0	9.3	42.0	55.2
SD+DINO (S) [51]	68.9	81.7	87.2	9.7	50.4	65.9
Ours	**74.0**	**85.3**	**89.7**	**16.5**	**56.7**	**70.2**

(*i.e.*, PCK@0.01). Our hypothesis is that applying windows during training helps the model focus on the local region but overlook global information.

D.3. Discussion on Generalizability

As shown in the main paper, we validate the generalizability of our method by training on AP-10K intra-species set and testing on cross-species and cross-family subsets. Here, we extend this analysis with additional tests:

Training on PF-PASCAL and testing on other datasets. We evaluate the generalizability of our method by training it on PF-PASCAL and then testing it on SPair-71k and AP-10K intra-species test sets (see Tab. 10). While previous studies [5, 14] have noted a potential performance decrease due to models' overfitting to the limited distribution of pose variation in PF-PASCAL, our method consistently outperforms across different datasets and PCK thresholds, demonstrating its robustness.

Training on SPair-71k and testing on AP-10K and PF-PASCAL. In a similar vein, we trained our model on the SPair-71k dataset and evaluated its performance on PF-PASCAL and AP-10K intra-species test sets (see Tab. 11). The findings mirrored those from Tab. 10, with our approach achieving the best results across all datasets and PCK metrics, confirming its generalizability again.

E. Additional Results

E.1. Window Soft Argmax for Zero-Shot Semantic Correspondence

In the main paper, our zero-shot method only applies adaptive pose alignment, however, we could also employ the window soft argmax at test time as it doesn't require additional supervision.

As shown in Tab. 7, the window soft argmax consistently improves the zero-shot results for both the SD+DINO and our zero-shot method, on both the geometry-aware subset and standard set, outperforming either the argmax or soft argmax. This further demonstrates the effectiveness of our method.

E.2. Qualitative Results on AP-10K

We show the qualitative comparison of our supervised methods with both unsupervised and supervised versions of SD+DINO [51] on AP-10K intra-species (Fig. 19), cross-species (Fig. 20), and cross-family (Fig. 21) subset.

E.3. Additional Qualitative Results on SPair-71k

In Fig. 22 and Fig. 23, we show the qualitative comparison of our supervised methods with both the unsupervised and supervised versions of SD+DINO [51] on SPair-71k dataset. Our method establishes correct correspondence for challenging cases that previous works cannot handle.

Figure 19. **Qualitative comparison on the AP-10K intra-species set.**

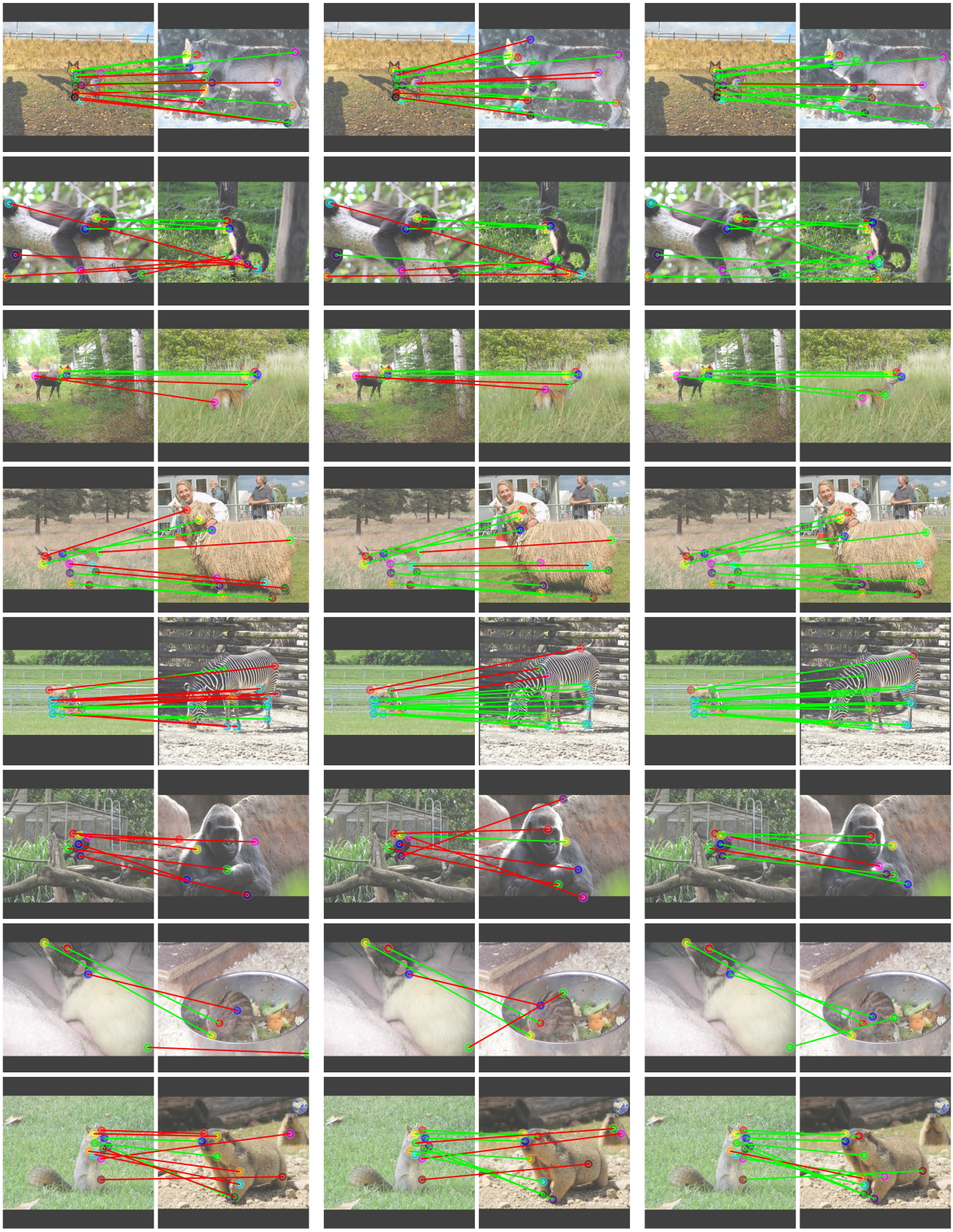
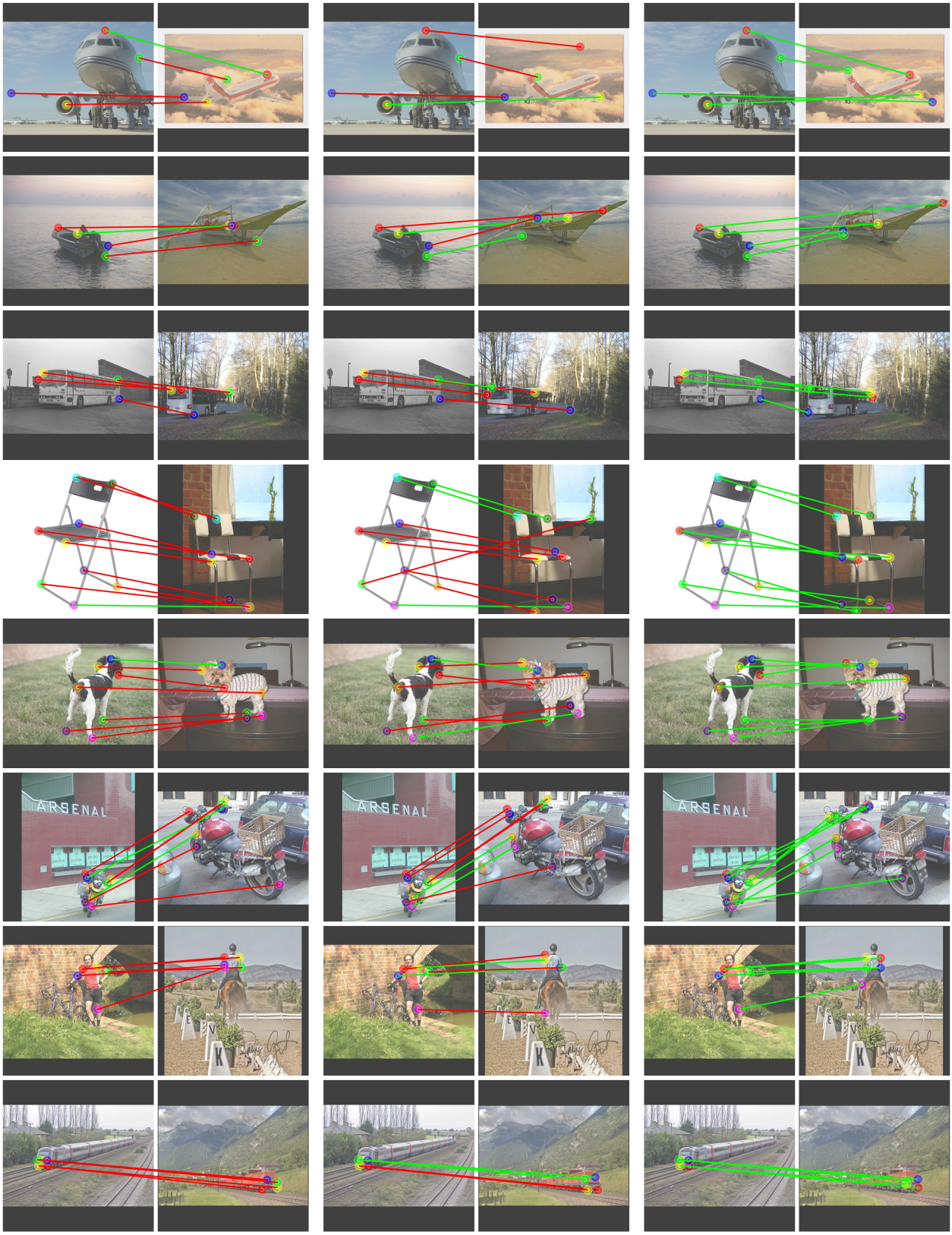


Figure 20. **Qualitative comparison on the AP-10K cross-species set.**

Figure 21. **Qualitative comparison on the AP-10K cross-family set.**



SD+DINO SD+DINO (S) **Ours**

Figure 22. **Qualitative comparison on the SPair-71k.** Our method shines even in cases with large viewpoint variations.

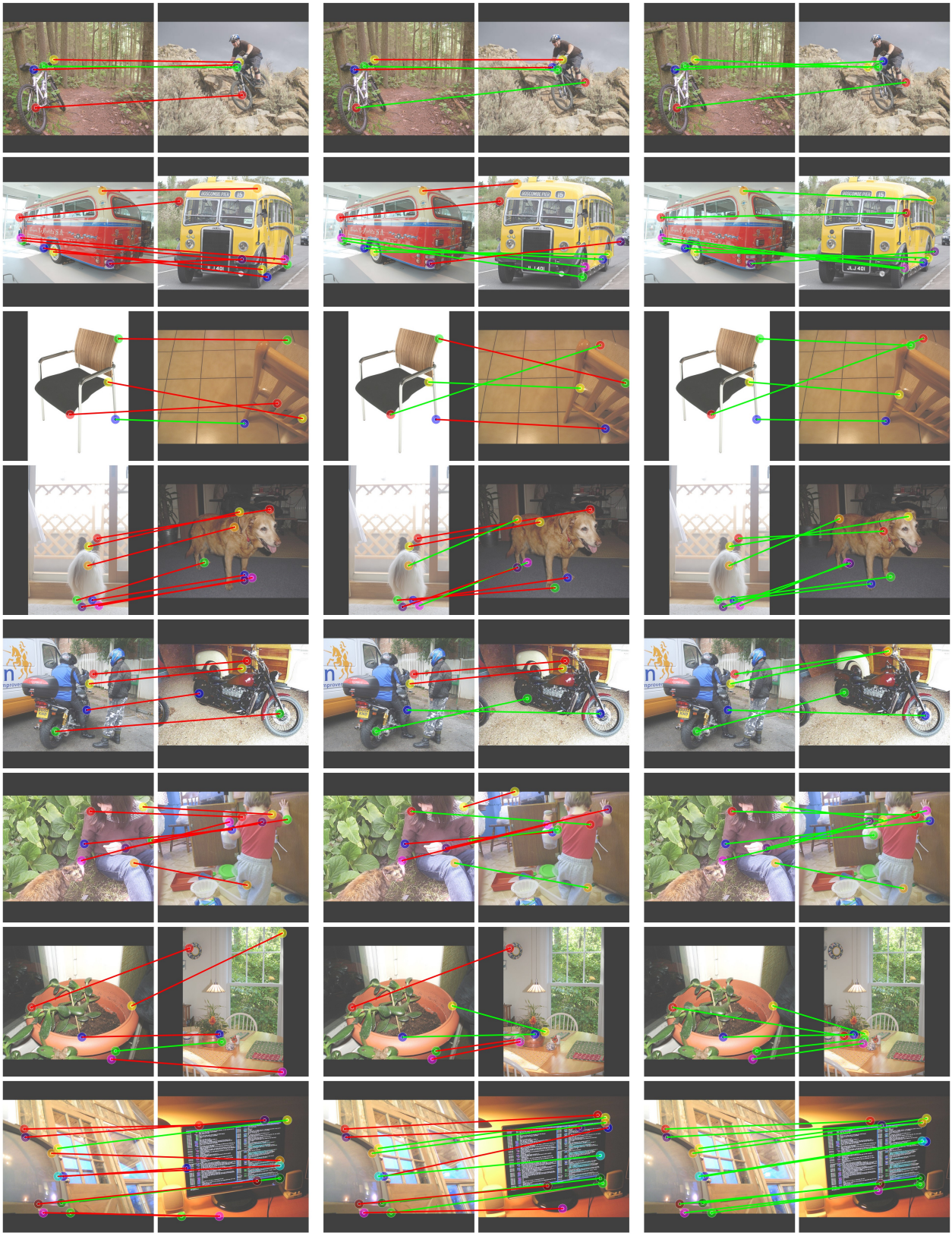


Figure 23. **Qualitative comparison on the SPair-71k.** Our method shines even in cases with large viewpoint variations.