# *Jamais Vu*: Exposing the Generalization Gap in Supervised Semantic Correspondence

**Octave Mariotti**[1] **Zhipeng Du**[1] **Yash Bhalgat**[2] **Oisin Mac Aodha**[1] **Hakan Bilen**[1]

[1]University of Edinburgh [2]University of Oxford

https://github.com/VICO-UoE/JamaisVu

## Abstract

The goal of semantic correspondence (SC) estimation is to establish semantically meaningful matches across different instances of an object category. In this work, we illustrate how recent supervised SC methods generalize poorly beyond the annotated keypoints seen during training, thus effectively acting as keypoint detectors. To address this, we propose a new approach for learning dense correspondences by lifting 2D keypoints into a canonical 3D space using monocular depth estimation. Our method constructs a continuous canonical manifold that captures object geometry without requiring explicit 3D supervision or camera annotations. Additionally, we introduce SPair-U, an extension of SPair-71k with novel keypoint annotations, to better assess generalization. Experiments not only demonstrate that our model significantly outperforms supervised baselines on unseen keypoints, highlighting its effectiveness in learning robust correspondences, but that unsupervised baselines outperform supervised counterparts when evaluated across different datasets.

## 1 Introduction

Semantic correspondence (SC) estimation involves identifying semantically matching regions in images across different instances of the same object category. It remains a challenging problem, as it requires recovering fine-grained details while maintaining robustness against variations in object appearance, shape, and viewing conditions. Recent advances in large-scale vision models, particularly self-supervised transformers [6, 39] and generative diffusion models [41], have led to notable improvements in SC. When employed as backbones, these models have achieved over 20% gains in accuracy on the SPair-71k benchmark [34]. However, despite these advances, recent studies have highlighted that these powerful representations often struggle to disambiguate symmetric object parts due to their visual similarity [62, 32].

SC methods can be broadly categorized into two groups in terms of supervision: unsupervised models, which do not require correspondence annotations during training [1, 2, 61, 32], and supervised models [8, 18, 62], which are trained on manually annotated correspondences. As expected, supervised models generally achieve higher performances when using the same backbone and same training set as unsupervised models. However, a key limitation of current benchmarks is that evaluation is typically performed on the *same* set of keypoints used for training, potentially inflating perceived generalization. As illustrated in Fig. 1, the performance of supervised models drops significantly when evaluated on unseen keypoints, while unsupervised models maintain their performance.

Building dense correspondences are key to fine-grained object understanding and improving robustness in various recognition tasks, and essential to many applications including texture transfer [38] and robotic manipulation [49]. In this work, we examine the performance of state-of-the-art SC

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

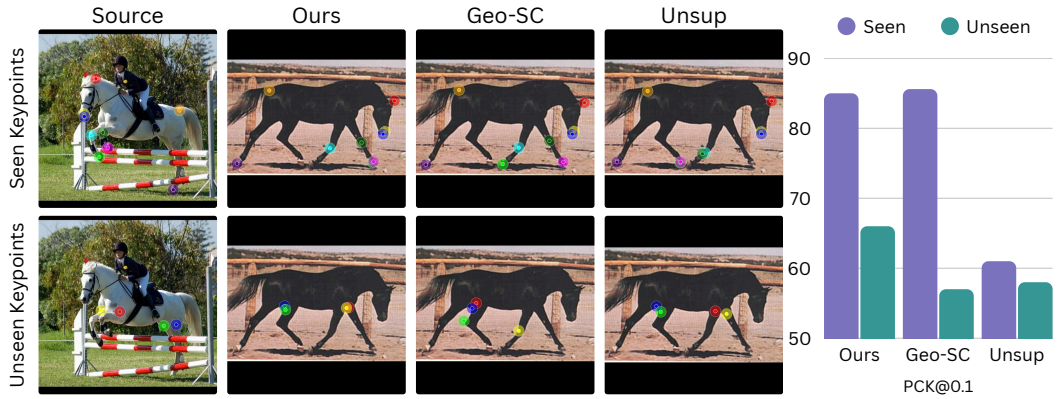



Figure 1: **Illustration of the generalization gap on unseen keypoints.** (Left) Top row: when evaluated on known keypoints, both our model and Geo-SC [62] perform well, while the unsupervised DINOv2+SD [61] struggles to correctly disambiguate the legs of the horse. Bottom row: when presented with keypoints unseen at training time, both our model and DINO+SD predict noisy but reasonable correspondence, while Geo-SC predictions noticeably degrade. (Right) Even though it obtains strong performance on known keypoints, Geo-SC performs worse than its unsupervised counterpart on our new benchmark of unseen keypoints. In comparison, our model still achieves competitive results.

models when evaluated on points that lie outside the set of annotated training keypoints. Under these conditions, we observe that supervised pipelines often underperform their unsupervised counterparts, effectively reducing their function to that of 'sparse keypoint detectors'. We attribute this limitation to two main factors: (i) the sparsity of supervision, which typically focuses on a limited number of keypoints and (ii) the lack of evaluation on unseen points, which tends to favor models that bias their predictions toward the nearest seen annotations.

We argue that an ideal SC method should be capable of matching arbitrary points, akin to the objectives in classical dense correspondence tasks [26]. To move beyond sparse supervision, we propose a learning framework that predicts dense feature maps and supervises them using geometry from an off-the-shelf depth estimator, thereby enabling training on richer and more spatially diverse cues. Some unsupervised SC methods [46, 32] have leveraged 3D geometry to learn dense correspondences through mapping object pixels to a spherical coordinate system where each coordinate corresponds to a different characteristic point of the object. However, this approach requires estimating the object shape and viewpoint from a collection of 2D images, which limits the applicability of such models to synthetically generated datasets [46] and they can require additional camera viewpoint supervision [32].

We propose a new approach that leverages existing 2D keypoint annotations and estimated 3D geometry to learn dense correspondences. We build on the idea of learning a canonical representation of the object category, which is invariant to the object instance, viewpoint, and pose. We achieve this by lifting the 2D keypoints to 3D using a monocular depth model, aligning them with a set of canonical keypoints that are shared across all instances of the object category. Finally, by interpolating between them, we learn a continuous canonical manifold, that captures the underlying 3D shape of the object and incorporates geometric constraints into learning more effective and general feature representations. We also introduce a new dataset for SC estimation, SPair-U, which extends the original SPair-71k test annotations with a set of new keypoints, allowing us to evaluate the generalization of SC models on unseen keypoints. We show that supervised SC models trained on the original SPair-71k dataset typically fail to generalize well to unseen keypoints, while our method is able to learn a more general representation that can be applied to unseen keypoints.

## 2 Related Work

**Supervised methods.** Supervised approaches rely on the availability of datasets with annotated keypoints such as CUB [51], PF-PASCAL [15], and SPair-71k [34] to learn corresponding points across instances of the same object class depicted in different images. This is typically using contrastive objectives minimizing distance between features coming from the same keypoints while

pushing other features away [16, 44, 61, 62]. A more computationally intensive option is to compute dense 4D correlations maps between each source and target locations [8, 33, 18, 22]. To obtain stronger descriptors, it is also common to aggregate features from multiple network layers to form hypercolumns [35, 1, 62]. Current state-of-the-art supervised methods forego training from scratch and instead typically use a large pretrained vision model as a backbone, the most popular options being DINOv2 [39] and Stable Diffusion [41]. While effective at matching instances of keypoints of the same type that have been observed during training, in our experiments we demonstrate that current supervised methods have a tendency to overfit to the set of keypoints observed during training and struggle to generalize to previously unseen keypoints (see Fig. 1 for an example).

There have also been recent attempts to utilize the expressive power of the representations encoded in large multi-modal models for detecting sets of keypoints. Few-shot methods require supervision in the form of a support set at inference time [28, 29, 17]. Zero-shot methods forego the need for such supervision, but instead require that keypoints should be described via natural language prompts [63, 60]. Describing common keypoints (e.g., 'the left eye') can be easily done via language, but how to best describe less salient points via text is not so clear. There have also been attempts to develop models that can take different various modalities (i.e., text and or keypoints) as input [30]. While promising, these methods make use of large multi-modal models and need large quantities of keypoint supervision data, spanning many diverse keypoints and categories, for pretraining.

**Unsupervised / weakly-supervised methods.** Methods that do not use correspondence supervision during training range from unsupervised approaches using general-purpose backbones [1, 61], very weakly supervised methods that only assume an curated training set without labels, e.g., images of a single category [46, 45, 2], zero-shot methods that only use test-time information about the relationships between keypoints [38, 62], dense methods that directly impose structure on the correspondence field [7], and weakly supervised methods that use extra labels like segmentation masks or camera pose [20, 32, 4]. Earlier unsupervised methods typically use self-supervised objectives that make use of synthetic deformations/augmentations of the same image [46] or by using cycle consistencies [45, 47, 48] to provide pseudo ground truth correspondence. Later it was observed that large pretrained vision models naturally posses features that are very strong for SC, despite not being trained on this task explicitly. As a result, more recent unsupervised methods tend to not train their own backbones from scratch and instead explore ways to aggregate [1, 61], or align [14, 32] these features across images.

**Geometry-aware methods.** Inspired by the classic correspondence setting in vision that relies on geometric constraints to match the same 3D locations across views [10, 42], utilizing geometry cues is an effective way to learn SC. For SC, the underlying assumption is that different instances from the same object category share a similar spatial structure. Flow and rigidity constraints are often used in tracking [53, 58] and unsupervised SC [25, 14, 4]. Recent studies have shown that ambiguities caused by symmetric objects are a major source of errors in SC [62, 32]. One potential way to mitigate these errors is to develop 3D-aware methods. Initially, this has been explored by building correspondence across images by matching points along the surface of objects to 3D meshes [64, 24, 37, 56, 11], but the requirement for meshes greatly limits the applicability of these approaches. More recently, methods have been proposed to learn 3D shape from image collections [36, 57, 3]. However, these methods require solving multiple problems at once, i.e., estimating object shape, deformations, camera pose, and are therefore limited to specific types of object categories and tend to break easily when applied to more complex shapes. Recent advances in monocular depth [40, 5, 21, 59, 13, 54] and geometry prediction [55] allows for reliable geometry estimation from a single image, which can be leveraged for imparting 3D-awareness in into SC methods.

Concurrent work [52] also proposes to geometrically align images in 3D using depth maps to build category prototypes. It is designed around a test-time alignment of the 3D prototype to the test image. This requires knowing the test category beforehand, having build a dedicated prototype for it, having a segmentation mask for the test instance, and solving a computationally expensive alignment problem, limiting its applicability to severely constrained scenarios where the category has been seen during training and throughput it not an issue. In comparison, while our model shares a similar concept of building 3D category prototype, our goal is to design a generalizable SC pipeline by training a correspondence head on top of a backbone. This allows our model to compute correspondence in a single feedforward pass and generalize to new categories all the while not requiring knowledge of the test category or segmentation.

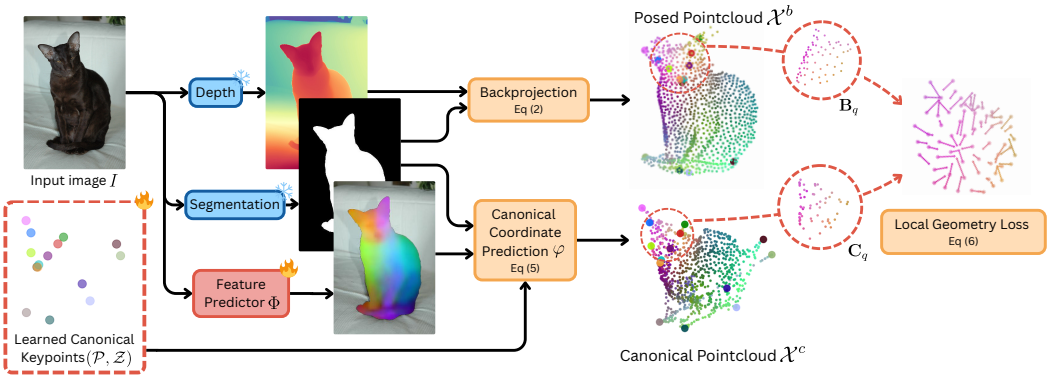



Figure 2: **Overview of our approach.** We extract segmentation masks and depths maps from training images and backproject object points to produce the posed point clouds $\mathcal{X}^b$. We predict dense features with $\Phi$ and match them against our jointly learned sparse category prototype $(\mathcal{P}, \mathcal{Z})$ to produce the canonical point clouds $\mathcal{X}^c$. The local geometric alignment between the two provides supervision for updating $\Phi$.

# 3 Method

## 3.1 Overview

Let $I \in \mathbb{R}^{H \times W \times 3}$ be an RGB image depicting an object, defined over the image domain $\Lambda \in \mathbb{R}^2$, a lattice of size $H \times W$. Our objective is to learn a function $\Phi(I, \boldsymbol{u}) \rightarrow \boldsymbol{w}$, which maps each pixel coordinate $\boldsymbol{u} \in \Lambda$ to a descriptor $\boldsymbol{w} \in \mathbb{R}^M$. The descriptor $\boldsymbol{w}$ should be semantically consistent, be meaningfully aligned across different images of objects from the same category, and be invariant to changes in pose and shape. Once $\Phi$ is learned, SC between two images $I$ and $I'$, depicting the same object category, can be established by finding, for a pixel $\boldsymbol{u}$ in image $I$, the most similar pixel $\boldsymbol{u}'$ in the other image $I'$. This is done by querying nearest-neighbor matching in descriptor space according to distance $d$, typically the cosine distance, i.e., $d(\boldsymbol{a}, \boldsymbol{b}) = 1 - \langle \boldsymbol{a}, \boldsymbol{b} \rangle / (||\boldsymbol{a}||_2 ||\boldsymbol{b}||_2)$:

$$\boldsymbol{u}' = \arg\min_{\boldsymbol{v}} d(\Phi(I, \boldsymbol{u}), \Phi(I', \boldsymbol{v})). \tag{1}$$

In the standard supervised SC task, we are given a training set, $\{(I^{(1)}, \mathcal{K}^{(1)}), \ldots, (I^{(N)}, \mathcal{K}^{(N)})\}$ where each image $I^{(n)}$ is annotated with a sparse set of semantic keypoints $\mathcal{K} = \{\boldsymbol{k}_1, \ldots, \boldsymbol{k}_{|\mathcal{K}|}\}$ where $\boldsymbol{k} \in \Lambda$. A common strategy, adopted in recent works [18, 8, 9, 31, 61, 62], is to learn a descriptor function $\Phi(I, \boldsymbol{u})$ that produces a local descriptor $\boldsymbol{w}$ for each pixel $\boldsymbol{u}$ in $I$ such that the descriptors of corresponding keypoints in paired images are close in feature space. While this sparse keypoint supervision helps the model learn semantically meaningful descriptors for the annotated keypoints, it does not guarantee generalization to unlabeled regions of the object.

A promising direction to address this limitation is to incorporate 3D geometry by assigning each pixel a coordinate in an object centric reference frame. Prior works [46, 32] explores this idea by projecting object surfaces onto a spherical coordinate system, with each coordinate on the sphere corresponding to a different characteristic point of the object. However, this necessitates inferring both object shape and the viewpoint from a collection of 2D images, a highly ill-posed problem, which requires generating pairs through synthetic warps resulting unrealistic shapes [46] or relies on viewpoint supervision which can be challenging to predict automatically. In the next section we show how to combine sparse keypoints annotations with 3D geometry cues to learn dense and semantically consistent descriptors for every pixel in an image. An overview of our approach is shown in Fig. 2.

## 3.2 Canonical Representation Learning

Similar to [46, 32], we aim to learn a 3D *canonical* representation for each object category, along with a function $\varphi(I, \boldsymbol{u}) \rightarrow \boldsymbol{x} \in \mathbb{R}^3$ that maps a pixel $\boldsymbol{u}$ in image $I$ to its 3D coordinates in the canonical object-centric coordinate system. Unlike the spherical representations in prior work [46, 32], we do not impose any topological constraints on the canonical representation (e.g., enforcing a spherical surface). We parameterized the canonical representation by a set of 3D keypoints $\mathcal{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{|\mathcal{K}|}\}$, where each $\boldsymbol{p}_i$ corresponds to a labeled keypoint $\boldsymbol{k}_i$ in $I$. A crucial aspect of this parametrization is

that unlike the labeled image keypoints $\mathcal{K}$, $\mathcal{P}$ is shared across all instances of the category, and is invariant to object instance, viewpoint, and pose, ensuring the canonicity of the representation.

To compute $\mathcal{P}$, we first estimate the 3D coordinates of each keypoint $\boldsymbol{k}$ in image $I$ using a monocular depth model $\Psi(I, \boldsymbol{k}) \rightarrow \mathbb{R}^+$ and then backproject it using estimated camera intrinsics $\mathbf{A} \in \mathbb{R}^{3\times3}$ as follows:

$$\bar{\boldsymbol{k}} = \Psi(I, \boldsymbol{k})\mathbf{A}^{-1}\left[k_x, k_y, 1\right]^\top, \tag{2}$$

where $\boldsymbol{k} = (k_x, k_y)$ and $\bar{\boldsymbol{k}} \in \mathbb{R}^3$ represents the 3D 'posed' coordinate. We denote the set of backprojected coordinates as $\bar{\mathcal{K}} = \{\bar{\boldsymbol{k}}_1, \ldots, \bar{\boldsymbol{k}}_{|\mathcal{K}|}\}$.

To align $\bar{\mathcal{K}}$ with $\mathcal{P}$, we compute a rigid transformation, comprising rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{T} \in \mathbb{R}^{3\times1}$, and scale $s \in \mathbb{R}^+$ such that $\mathbf{M} = s[\mathbf{R}|\mathbf{T}] \in \mathbb{R}^{3\times4}$. We optimize the canonical keypoints $\mathcal{P}$ by minimizing their alignment error across the training set by solving a generalized Procrustes problem:

$$\min_{\boldsymbol{p}_i} \sum_{n=1}^{N} ||\boldsymbol{p}_i - \hat{\boldsymbol{M}}^{(n)}\bar{\boldsymbol{k}}_i^{(n)}||_1 \quad \text{where} \quad \hat{\boldsymbol{M}}^{(n)} = \arg\min_{\boldsymbol{M}} \sum_{i=1}^{|\mathcal{K}|} ||\boldsymbol{p}_i - \boldsymbol{M}\bar{\boldsymbol{k}}_i^{(n)}||_2. \tag{3}$$

We use the Kabsch-Umeyama algorithm [19, 50] to compute the optimal transformation $\hat{\boldsymbol{M}}^{(n)}$ between the canonical and posed coordinates keypoints. To prevent the degenerate solution where $s = 0$ leads to a global collapse, we modify the procedure to constrain $s \geq 1$. This ensures objects cannot shrink in size, effectively resizing them to the size of the largest object in the training set. Even though alignments rely on only a sparse subset of visible keypoints per-image, i.e., occluded and out-of-frame points are not considered, we find this sufficient to recover a globally consistent arrangement for $\mathcal{P}$.

Next we associate each canonical keypoint in $\boldsymbol{p}$ with a learnable descriptor $\boldsymbol{z}$, forming a set $\mathcal{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{|\mathcal{K}|}\}$. We learn these jointly with $\Phi$, using a cross-entropy loss over cosine similarities between extracted and canonical descriptors:

$$\min_{\Phi,\mathcal{Z}} -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{|\mathcal{K}|} \log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \Phi(I^{(n)}, \boldsymbol{k}_i^{(n)})/\tau)}{\sum_{j=1}^{|\mathcal{K}|}\exp(\text{sim}(\boldsymbol{z}_j, \Phi(I^{(n)}, \boldsymbol{k}_i^{(n)})/\tau)}, \tag{4}$$

with cosine similarity $\text{sim}(\cdot,\cdot)$ and learned temperature parameter $\tau$. This objective encourages $\Phi$ to produce distinctive and semantically consistent features across object instances for each keypoint $\boldsymbol{k}$.

### 3.3 Dense Geometric Alignment

So far, we have only modeled object geometry at the sparse level via $\mathcal{P}$. We now extend this to dense correspondence by defining $\varphi(I, \boldsymbol{u})$, a function that maps every pixel $\boldsymbol{u}$ in $I$ to a coordinate in the canonical space. We compute it as an attention-weighted sum over canonical keypoints:

$$\varphi(I, \boldsymbol{u}) = \sum_{i=1}^{|\mathcal{K}|} \frac{\exp(\text{sim}(\boldsymbol{z}_i, \Phi(I, \boldsymbol{u}))/\tau)}{\sum_{j=1}^{|\mathcal{K}|}\exp(\text{sim}(\boldsymbol{z}_j, \Phi(I, \boldsymbol{u}))/\tau)}\boldsymbol{p}_i. \tag{5}$$

This is equivalent to computing descriptors via softmax attention over $\mathcal{Z}$, using $\Phi(I, \boldsymbol{u})$ as queries, $\boldsymbol{z}$ as keys, and $\boldsymbol{p}$ as values. For labeled keypoints $\boldsymbol{k}_l$, minimizing Eq. (4) ensures $\varphi(I, \boldsymbol{k}_l) = \boldsymbol{p}_l$.

For each training image $I$, we can now estimate the dense canonical coordinates $\mathcal{X}^c$ over its pixels via Eq. (5), and the posed coordinates $\mathcal{X}^b$ via depth backprojection using Eq. (2). In practice, $\mathcal{X}^c$ and $\mathcal{X}^b$ only consist of object points that are selected using an object segmentation mask. We aim to align these two representations so that $\mathcal{X}^c$ properly reflects the object geometry. However, our annotations are only sparse, thus we cannot directly supervise $\varphi(I, \boldsymbol{u})$ for arbitrary coordinates $\boldsymbol{u}$. Instead, we make the assumption that even though the posed and canonical shape are different, they should be *locally* similar. We encourage the geometric alignment between a small neighborhood of points sampled in the posed space, and their corresponding locations in the canonical space.

For a given point $\mathbf{q} \in \mathcal{X}^c$, we sample its $k$ nearest neighbors to obtain $\mathbf{C}_q$ in the canonical space, and the corresponding coordinates in the posed space $\mathbf{B}_q$, and minimize the alignment error between

two sets. We also sample neighbors of a given point $r \in \mathcal{X}^b$ in the posed space, denoted as $\mathbf{B}_r$, and compute the loss in the other direction:

$$\min_{\Phi,\mathcal{Z}} \frac{1}{N} \sum_{n=1}^{N} ||\mathbf{C}_q^{(n)} - \mathbf{M}_{c2b}^{(n)} \mathbf{B}_q^{(n)}||_1 + ||\mathbf{B}_r^{(n)} - \mathbf{M}_{b2c}^{(n)} \mathbf{C}_r^{(n)}||_1, \tag{6}$$

where $\mathbf{M}_{c2b}$ and $\mathbf{M}_{b2c}$ are the rigid transformations between the canonical and posed coordinates, computed using the Kabsch-Umeyama algorithm at each iteration as in Eq. (3).

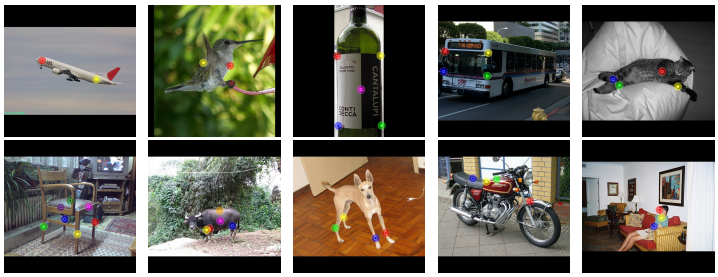
In the canonical space, neighbors are selected using a standard k-nearest neighbors strategy. However, this approach can be unreliable in the posed space due to object deformations. For instance, in the case of a person eating, the hand might be close to the face in 3D space, and thus points belonging to the face might mistakenly be selected as neighbors of the hand. Instead, we use a pseudo-geodesic sampling strategy that samples points along the surface of the object. Starting from a seed point, we iteratively grow the neighborhood by selecting the next point with the shortest distance to the current set, effectively approximating surface-based proximity rather than raw spatial closeness. Pseudocode is provided in Alg. 1.

We jointly optimize the descriptor learning loss in Eq. (4) and the geometric consistency loss in Eq. (6) to learn $\Phi$ and $\mathcal{P}$. While these objectives suffice to learn a SC model, in practice we build our implementation on Geo-SC [62] and optimize its parameters jointly over the sum of our objective and the original one. At inference time, rather than simply querying nearest-neighbor in the descriptor space predicted by $\Phi$, we make use of the soft-argmax window matching strategy proposed in [62]. Unlike $\varphi$, which relies on category-specific canonical coordinate set $\mathcal{P}$ and descriptors $\mathcal{Z}$, $\Phi$ can be applied to previously unseen object categories directly.

## 4 SPair-U: A Benchmark for Evaluating Unseen Keypoints

As illustrated in Fig. 1, the performance of the state-of-the-art supervised SC methods [62, 61] degrades significantly when queried on keypoints that are not part of their training sets. We posit that this is caused by models only learning strong representations for these specific points, while largely ignoring the remaining pixels. We would like to assess the performance of SC methods when evaluated on keypoints that were previously unseen (i.e., not in the labeled set) at training time. A possible solution is to use an existing dataset while splitting the annotations into two mutually exclusive sets of keypoints, seen and unseen, between training and evaluation. However, this strategy would reduce the supervision available, and require retraining previous techniques for evaluation.

Instead, we introduce a new evaluation benchmark, **SPair-U**, by labeling additional keypoints from the SPair-71k dataset [34]. We added at least four new points for each of the 18 categories found in SPair-71k. For animals, we focused on additional joints on the limbs, and for vehicles we added semantic parts that were not already labeled, e.g., windshield or fenders. Boats, bottles, potted plants, and tv monitors keypoints are not semantic *per se* in SPair-71k, but are rather spread around on the outline of the object. Thus, we added midway points between those already defined. In total, we add 1,272 new individual test keypoint annotations resulting in 19,990 new keypoint pairs spread across 8,254 image pairs. We illustrate some of these new annotations in Fig. 3, and the full list with more details can be found in Table A3. As shown in Fig. 1, current supervised methods tend to predict locations of keypoints seen during when queried on the new SPair-U points.



| SPair-U Dataset Statistics | |
|---|---|
| # of image pairs | 8,254 |
| # of new keypoints | 1,272 |
| # of keypoint pairs | 19,990 |
| # of images without new points | 53 |
| # of occurrences for least frequent kp | 2 |
| # of occurrences for most frequent kp | 27 |
| average # of occurrences | 16 |
| average # of new points per-image | 2.6 |

Figure 3: **Example keypoint annotations from our new SPair-U evaluation dataset.** It utilizes the same images as the SPair-71k dataset [34], but adds additional keypoints not present in SPair-71k. This enables benchmarking of SC methods on the existing keypoints along with our new ones. On the right we summarize the main statistics of our new dataset.

Table 1: **Results on standard evaluation keypoints for SPair-71k**. Per-image PCK@$0.1_{bbox}$ scores are reported. In this table and the following: All models use the soft matching strategy described in [62] except those followed by *. Models with a dagger† benefit from AP-10K pretraining. Models in the $\mathcal{K}$ category use keypoint supervision, while $\not{\mathcal{K}}$ do not. Best results are **bolded** and second best are underlined.

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\not{\mathcal{K}}$ | SD [41][61] | 62.8 | 52.7 | 80.6 | 31.2 | 43.4 | 39.1 | 35.6 | 76.0 | 32.0 | 67.6 | 50.9 | 59.7 | 51.0 | 47.3 | 48.6 | 43.8 | 61.8 | 52.9 | 52.0 |
| | DINOv2 [39][61] | 73.4 | 60.2 | 88.8 | 43.2 | 41.1 | 46.7 | 45.1 | 75.0 | 33.4 | 69.8 | 66.1 | 69.6 | 60.7 | 66.6 | 30.7 | 61.3 | 54.2 | 23.9 | 55.3 |
| | DINOv2+SD [61] | 73.8 | 61.0 | 89.6 | 40.2 | 52.5 | 47.4 | 44.1 | 81.1 | 41.5 | 76.8 | 64.8 | 70.5 | 61.7 | 66.3 | 54.3 | 62.7 | 63.5 | 52.4 | 61.1 |
| | SphericalMaps [32] | 76.2 | 60.1 | 90.0 | 46.5 | 53.0 | 74.9 | 68.0 | 83.8 | 45.1 | 81.7 | 67.6 | 75.4 | 69.1 | 58.9 | 50.0 | 67.5 | 73.9 | 58.1 | 66.1 |
| $\mathcal{K}$ | SCorrSan* [18] | 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++* [9] | 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* [31] | 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 |
| | DINO+SD (S) [61] | 84.7 | 67.5 | 93.2 | 64.5 | 59.2 | 85.7 | 82.0 | 89.8 | 57.0 | 89.3 | 76.2 | 80.8 | 75.9 | 80.2 | 64.7 | 71.2 | 93.6 | 70.5 | 76.5 |
| | Geo-SC [62] | 86.6 | 70.7 | 95.8 | 69.2 | 64.8 | 94.5 | 90.6 | 91.0 | 67.1 | 91.8 | 86.1 | 86.3 | 79.3 | 87.9 | 80.8 | 82.1 | 96.6 | 83.4 | 83.2 |
| | Geo-SC† [62] | 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** |
| | Ours | 86.8 | 72.6 | 95.3 | 70.7 | 64.8 | 94.6 | 90.3 | 89.4 | 70.7 | 94.1 | 84.8 | 83.0 | 80.5 | 87.0 | 79.1 | 77.5 | 95.8 | 82.8 | 82.9 |
| | Ours† | **92.2** | **76.3** | 96.5 | **72.0** | 68.1 | **95.0** | **90.8** | **93.1** | **75.1** | 94.2 | **91.2** | 86.0 | **82.1** | **91.7** | 80.0 | 81.2 | 95.8 | 84.0 | 85.4 |

Table 2: **Results on unseen keypoints on our SPair-U benchmark.** Per-image PCK@$0.1_{bbox}$ scores on unseen keypoints are reported.

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\not{\mathcal{K}}$ | SD [41][61] | 73.2 | 71.8 | 48.8 | 37.7 | 43.0 | 55.1 | 47.2 | 25.4 | 35.9 | 60.4 | 46.2 | 41.6 | 59.9 | 53.1 | 57.8 | 36.1 | 50.6 | 19.5 | 47.4 |
| | DINOv2 [39][61] | 88.2 | 75.6 | **79.0** | 52.9 | 39.8 | 54.1 | 60.0 | 43.9 | 34.8 | 67.2 | 64.6 | 53.6 | **75.8** | 79.1 | 37.8 | 45.6 | 53.3 | 8.4 | 54.9 |
| | DINOv2+SD [61] | 88.0 | **80.4** | 72.3 | 48.2 | **47.9** | 62.3 | 61.5 | 44.8 | 45.0 | 73.0 | 64.7 | 58.2 | 75.5 | **80.0** | 62.7 | 46.1 | **55.9** | 16.9 | 59.4 |
| | SphericalMaps [32] | **90.2** | 76.8 | 71.7 | 55.6 | 44.6 | 89.5 | **81.7** | 50.8 | 46.4 | 71.2 | 70.4 | 62.9 | 65.4 | 68.2 | 56.1 | 45.9 | 51.6 | 26.9 | 61.0 |
| $\mathcal{K}$ | SCorrSan* [18] | 56.9 | 26.9 | 23.0 | 37.6 | 31.4 | 52.8 | 41.7 | 16.6 | 15.4 | 21.0 | 47.1 | 17.8 | 27.3 | 48.1 | 47.8 | 20.1 | 28.0 | **34.2** | 32.7 |
| | CATS++* [9] | 69.9 | 43.8 | 14.0 | 47.1 | 31.9 | 69.5 | 47.0 | 11.7 | 24.4 | 15.1 | 47.9 | 25.8 | 32.0 | 54.3 | 51.6 | 17.5 | 27.9 | 22.8 | 35.9 |
| | DHF* [31] | 71.4 | 58.1 | 39.1 | 35.8 | 44.7 | 74.0 | 40.2 | 33.5 | 27.4 | 52.0 | 50.4 | 41.6 | 56.5 | 51.6 | 41.6 | 30.0 | 42.5 | 14.5 | 43.3 |
| | DINO+SD (S) [61] | 81.5 | 73.6 | 57.1 | 63.4 | 35.8 | 85.7 | 67.7 | **64.3** | 39.3 | 67.9 | **86.8** | **79.5** | 60.9 | 70.1 | 55.8 | **57.8** | 42.7 | 12.6 | 60.0 |
| | Geo-SC [62] | 80.9 | 71.4 | 51.8 | 65.3 | 36.9 | 91.0 | 70.8 | 55.7 | 36.9 | 55.7 | 79.2 | 53.7 | 66.5 | 62.3 | 61.1 | 39.0 | 39.0 | 17.4 | 56.9 |
| | Geo-SC† [62] | 74.6 | 70.6 | 55.5 | 65.1 | 36.4 | 85.1 | 72.3 | 50.1 | 40.1 | 60.6 | 85.3 | 65.7 | 52.9 | 61.9 | 66.6 | 41.8 | 36.6 | 13.8 | 57.1 |
| | Ours | 80.3 | 74.5 | 70.6 | 67.1 | 40.2 | **92.9** | 72.7 | 53.8 | 45.8 | 68.5 | 75.3 | 62.0 | 67.8 | 65.4 | 68.1 | 45.4 | 47.9 | 30.5 | 62.4 |
| | Ours† | 81.1 | 73.2 | 72.0 | **67.5** | 35.2 | 92.1 | 75.5 | 61.2 | **51.4** | **74.3** | **86.8** | 78.8 | 70.9 | 68.9 | **72.6** | 54.7 | 44.8 | 32.2 | **66.1** |

# 5 Experimental Results

## 5.1 Implementation Details

Our 3D prototype approach is complementary to existing semantic correspondence architectures, thus we can add it as an additional objective on top of established models. We base our experiments on Geo-SC [62], strictly following their provided hyperparameters, e.g., learning rate, batch size, optimizer, scheduler, and epoch count, simply adding our additional loss terms and jointly optimizing $\mathcal{P}$ and $\mathcal{Z}$ alongside Geo-SC's feature extractor $\Phi$. We also preserve the contrastive $\mathcal{L}_{\text{sparse}}$ and dense $\mathcal{L}_{\text{dense}}$ objectives with gaussian noise, as well as feature maps dropout and pose-variant augmentation. We refer to the original publication and official implementation for in-depth description of these features. While $\mathcal{P}$ and $\mathcal{Z}$ are category-specific, a single $\Phi$ is trained on the full dataset, allowing generalization to new categories. Furthermore, gradients coming from Eq. (6) are not backpropagated to $\mathcal{P}$, meaning it is only optimized using Eq. (3).

Our complete loss term is $\mathcal{L}_{\text{sparse}} + \mathcal{L}_{\text{dense}} + \mathcal{L}_{\mathcal{P}} + 0.3 \times \mathcal{L}_{\mathcal{Z}} + \mathcal{L}_{\text{geom}}$, where $\mathcal{L}_{\mathcal{P}}$, $\mathcal{L}_{\mathcal{Z}}$, and $\mathcal{L}_{\text{geom}}$ correspond to Eq. (3), Eq. (4) and Eq. (6) respectively. The justification for setting the weight $\lambda_{\mathcal{Z}} = 0.3$ is provided in Appendix B.

In practice, $\Phi$ consists of additional bottleneck layers trained on top of frozen DINOv2 and SD backbones. We extract depth maps and camera intrinsics using MoGe [54], and use Segment Anything [23] to obtain segmentation masks. Importantly, these are only used during training. During evaluation, matches are computed only from the predictions of $\Phi$. Additional implementation details can be found in Appendix A.

## 5.2 Quantitative Results

**Seen – SPair-71k.** We first compare our models to other SC approaches on the SPair-71k benchmark [34], which contains images from 18 categories. We use the standard PCK@$0.1_{bbox}$ which considers a match to be correct if its prediction lies within distance $0.1 \times \max(h, w)$ of the ground truth location, where $(h, w)$ is the height and width of the target object bounding box. Typically, supervised models report *per-image* PCK, i.e., the average score of each image per-category, while unsupervised ones use *per point* PCK, i.e., the average number of correct matches per-category. In order to properly compare results between the two families, we recompute *per-image* PCK for all models, which results in a small drop for the unsupervised models.

Table 3: **Cross-benchmark evaluation on held-out datasets.** Scores are reported using PCK with different thresholds. Here, only keypoint supervision from Spair-71k is used for supervised models.

| | | Spair-71k | | | Spair-U | | | AP-10K IS | | | AP-10K CS | | | AP-10K CF | | | PF-PASCAL | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PCK threshold | 0.01 | 0.05 | 0.10 | 0.01 | 0.05 | 0.10 | 0.01 | 0.05 | 0.10 | 0.01 | 0.05 | 0.10 | 0.01 | 0.05 | 0.10 | 0.05 | 0.10 | 0.15 |
| ✗ | SD [41][61] | 6.3 | 37.3 | 52.0 | 3.3 | 28.0 | 47.4 | 8.4 | 36.9 | 52.5 | 6.6 | 32.6 | 47.9 | 4.3 | 24.6 | 37.6 | 66.1 | 80.0 | 85.3 |
| | DINOv2 [39][61] | 6.8 | 38.0 | 55.3 | 3.7 | 32.4 | 54.9 | 10.5 | 44.8 | 63.6 | 8.8 | 41.7 | 61.6 | 7.7 | 34.7 | 52.0 | 62.4 | 78.2 | 83.5 |
| | DINOv2+SD [61] | 8.2 | 44.2 | 61.1 | **4.7** | 37.0 | 59.4 | 11.7 | 47.4 | 65.8 | 10.0 | 44.0 | 63.5 | 7.7 | 35.4 | 52.4 | 72.5 | 85.6 | 90.3 |
| | SphericalMaps [32] | 8.2 | 47.7 | 66.1 | 4.5 | **38.2** | 61.0 | 12.5 | 48.5 | 66.7 | 10.6 | 44.9 | 63.6 | 8.0 | 35.7 | 52.1 | 74.6 | **88.9** | **93.2** |
| ✓ | DINO+SD (S) [61] | 13.0 | 61.6 | 76.5 | 3.6 | 35.9 | 59.3 | 15.1 | 54.3 | **71.7** | 13.6 | 51.1 | 68.7 | 11.0 | 44.0 | 60.4 | 74.5 | 87.4 | 91.1 |
| | Geo-SC [62] | 20.0 | **72.2** | **83.2** | 4.6 | 35.5 | 56.9 | **16.6** | **55.8** | 70.5 | **15.2** | 52.4 | 67.7 | **11.9** | 45.9 | 59.6 | 75.3 | 87.0 | 90.7 |
| | Ours | **20.5** | 72.1 | 82.9 | 4.2 | 37.8 | **62.4** | 16.5 | **55.8** | 71.3 | 15.1 | **53.0** | **69.0** | 11.2 | **46.1** | **61.1** | **75.8** | 87.5 | 91.2 |

Results on seen keypoints in Table 1 show that our model ranks competitively against other approaches, with a marginal $0.2\%$ performance drop on average against its backbone Geo-SC [62]. Per-category results show small improvements in nine of the categories, the highest one being $3.6\%$ on bus, and small drop on the other nine, the largest being $2.4\%$ on bottle. Overall, the differences are minor, illustrating that adding our extra objective does not interfere with the original model.

**Unseen – SPair-U.** To evaluate a model's ability to generalize to unseen semantic points, we assess its performance on our new SPair-U keypoints using per-image PCK@$0.1_{\text{bbox}}$ for a like-for-like comparison. We exclude the Test-time Adaptive Pose Alignment from [62] since it requires prior knowledge of keypoint semantics to relabel flipped keypoints, which contradicts the assumption that evaluation keypoints are unknown.

As shown in Table 2, results on SPair-U reveal a stark contrast between supervised and unsupervised models. While unsupervised models see only a modest performance drop, likely due to increased task difficulty, supervised models experience a significant decline. Many of the pre-existing approaches are outperformed by the unsupervised DINO+SD baseline [61] and they are consistently beaten by the weakly-supervised Spherical Maps [32]. Notably, in eight categories, the best-performing model Has not seen keypoints during training, suggesting that supervised approaches behave more like keypoint regressors and fail to generalize to novel correspondences.

Our method also shows some performance degradation on SPair-U, but the drop is smaller than that of its backbone Geo-SC. It achieves the highest overall performance, improving upon the best prior supervised model by $6.1\%$, indicating stronger generalization to unseen keypoints. Nevertheless, the substantial gap between results on SPair-71k and SPair-U underscores a broader limitation: despite recent progress, most models struggle to move beyond sparse keypoint supervision toward robust, general semantic correspondence.

**Cross-benchmark evaluation.** We further evaluate our model on four benchmarks: SPair-71k, SPair-U, AP-10k [62], and PF-PASCAL [15]. While most supervised SC methods train separate models for each benchmark, this setup encourages overfitting to the benchmark and is impractical for real-world use. Instead, we advocate for evaluating generalization by training a single model on one dataset and testing it across multiple benchmarks. We choose SPair-71k for training due to its balanced mix of object and animal categories, making it suitable for generalization. To ensure fairness, we exclude models pretrained on AP-10K and standardize evaluation using the windowed soft-argmax protocol from [62].

As shown in Table 3, while Geo-SC [62] achieves the best performance on the standard SPair-71k test set, it underperforms on all other benchmarks, highlighting its limited generalization. Notably, even a simple supervised DINO+SD [61] baseline outperforms Geo-SC at the standard $0.10$ threshold when using the same soft window matching strategy. This stands in sharp contrast to the findings in [62] where Geo-SC consistently outperforms its baseline by $10\%$ on the three AP-10K benchmarks, when both models are trained on AP-10K, indicating potential overfitting to that dataset.

Consistent with earlier observations, a clear pattern emerges: models trained without keypoint supervision maintain stable rankings and performance gaps across datasets, whereas supervised models cluster more tightly in performance when evaluated out of their training distribution, revealing weaker cross-set generalization.

### 5.3 Qualitative Results

In Fig. 4, we visualize PCA projections of object features produced by our model, Geo-SC, and the unsupervised DINO+SD. Our model produces descriptors that vary smoothly over the object surface while uniquely identifying each point. In comparison, the predictions of Geo-SC are noisier, with

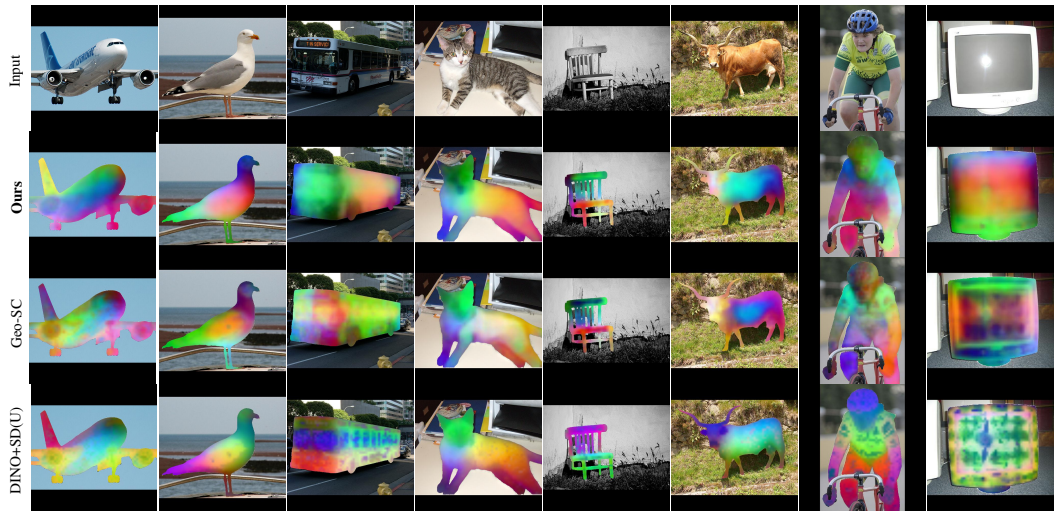



Figure 4: **PCA visualization of the feature maps from different models.** Note that PCA is computed on object features only. The inclusion of geometric constraints during training results in fewer high frequency artifacts in the predicted feature maps for our approach.


Figure 5: **Visualization of keypoint matches for randomly selected object points.** On each source image (left) we randomly sample points on the object of interest and compute their match on the target (right). Colored lines are used as a way to distinguish the points.

sudden discontinuities (e.g., bus) and uniform descriptors on regions that have no keypoints (e.g., the body of the cat and the cow). Meanwhile, the unsupervised features fail to separate repeated parts (e.g., plane engines) and produces noisy features in textureless areas (e.g., tv).

In Fig. 5 we further qualitatively evaluate our model's ability to generalize to unseen points. We randomly sample points on the source object and compute their matches on the target. Compared with Geo-SC, our approach exhibits better robustness against matching outside the target object, as well as better spatial awareness of points (e.g., Geo-SC matching points from the bottom of the source bus to the roof of the target), leading to higher matching quality. Further examples, along with comparisons to the unsupervised DINOv2+SD backbone, are provided in Fig. A4.

Following [52], we visualize our learned canonical shapes in Fig. 6 by collecting predicted canonical coordinates of multiple objects in order to overlap their partial point clouds over the training data. We observe that the spatial organization of $\mathcal{P}$, i.e., the large bold points, captures the general shape of the category, and the predicted coordinates densely span the object surface. Interestingly, our parametrization of the canonical shape Eq. (5) forces predicted coordinates to lie within the convex hull of $\mathcal{P}$, which explains the incomplete wheel on the motorbike. We also observe that very few points are mapped towards the end of the train, which we attribute to the varying length of trains across instances and the bias toward frontal viewpoints. Note that contrary to [52], these are simply visualizations and are not used for inference, meaning this limitation is unlikely to significantly affect performance.

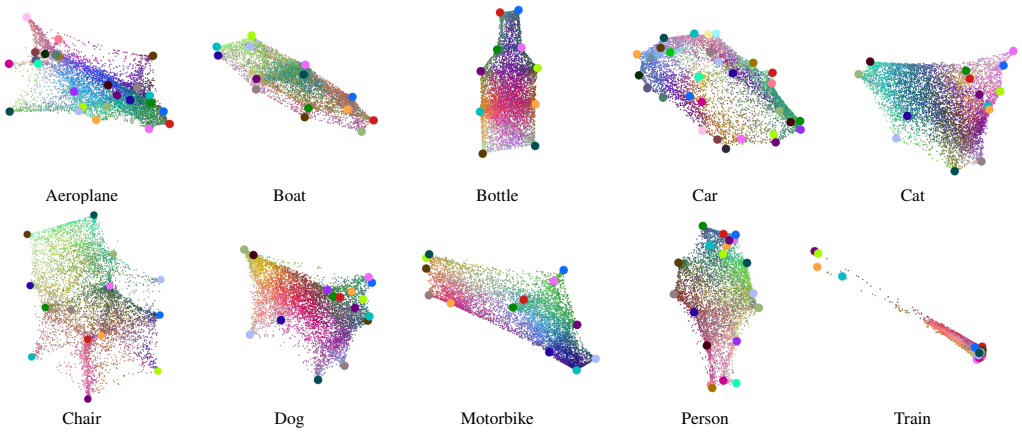



Figure 6: **Visualization of our learned canonical shapes.** Large points correspond to $\mathcal{P}$, each being attributed a distinctive color for visualization. Small points are predicted canonical coordinates of objects, colored with PCA of the features predicted by $\varphi$.

## 6 Limitations

While augmenting the test set of SPair-71k with new keypoints enables us to evaluate existing techniques with their provided models and code, our proposed benchmark SPair-U inherits some drawbacks from SPair-71k. In particular, the test set is small, consisting of only 481 images but with over 8,000 test pairs, and the categories are restricted to only common objects and animals. Furthermore, some categories were already labeled with a high number of keypoints where it is potentially easier to detect the newly added ones by relating them to the existing ones. While our findings related to generalization issues in supervised SC techniques remain valid, a larger, higher-quality held-out set of images and keypoints would be beneficial for more extensive evaluation.

Compared to prior supervised methods [18, 8, 31, 62], our approach incorporates additional supervision in the form of depth maps and segmentation masks, similar to [52], although in our case, they are only used during training. Furthermore, unlike [32], which relies on camera viewpoint annotations that off-the-shelf models cannot reliably provide, we obtain all additional signals using existing pretrained models.

In Eq. (3), we assume there exist a *global* rigid transformation between the posed keypoints and their canonical counterpart, which in practice is not the case especially for deformable objects. However, the sole purpose of this step is to optimize $\mathcal{P}$ into a coarse spatial organization of 3D keypoints (e.g., making sure that the left hand keypoint generally sits opposite of the right one) in order to allow the computation of *local* geometric alignment in Eq. (6). We show in Fig. 6 and Section 5.2 that despite this coarse assumption, our method is able to recover a reasonable 3D structure and performs well on deformable objects.

Finally, our assumption that geometry is a good proxy for semantics breaks down for complex object categories with diverse spatial part configurations. For example, cabinets may have different numbers of doors that open in various directions, leading to inconsistent placement of features like handles. Finally, we do not foresee any negative social impacts of our work.

## 7 Conclusion

We addressed the challenge of estimating semantic correspondences across different image instances of the same object category. Although recent supervised methods perform well on keypoints seen during training, we show that they often struggle to generalize to unseen keypoints. To overcome this, we introduced a new approach that incorporates geometric constraints during training by learning a continuous canonical manifold specific to each category. Our method outperforms both supervised and unsupervised baselines, as demonstrated on SPair-U – a new dataset we introduce with additional keypoint annotations for the widely used SPair-71k benchmark.

**Acknowledgements.** HB was supported by the EPSRC Visual AI grant EP/T028572/1.

## References


[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ECCV Workshops*, 2022.

[2] Mehmet Aygün and Oisin Mac Aodha. Demystifying unsupervised semantic correspondence estimation. In *ECCV*, 2022.

[3] Mehmet Aygün and Oisin Mac Aodha. SAOR: Single-View Articulated Object Reconstruction. In *CVPR*, 2024.

[4] Nir Barel, Ron Shapira Weber, Nir Mualem, Shahaf E Finder, and Oren Freifeld. Spacejam: a lightweight and regularization-free method for fast joint alignment of images. In *ECCV*, 2024.

[5] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[6] 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.

[7] Xinle Cheng, Congyue Deng, Adam W Harley, Yixin Zhu, and Leonidas Guibas. Zero-shot image feature consensus with deep functional maps. In *ECCV*, 2024.

[8] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Cats: Cost aggregation transformers for visual correspondence. *NeurIPS*, 2021.

[9] Seokju Cho, Sunghwan Hong, and Seungryong Kim. Cats++: Boosting cost aggregation with convolutions and transformers. *PAMI*, 2022.

[10] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *PAMI*, 2007.

[11] Niladri Shekhar Dutt, Sanjeev Muralikrishnan, and Niloy J Mitra. Diffusion 3d features (diff3f): Decorating untextured shapes with distilled semantic features. In *CVPR*, 2024.

[12] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In *International Conference on Multimedia*, 2019.

[13] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *ECCV*, 2024.

[14] 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.

[15] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow: Semantic correspondences from object proposals. *PAMI*, 2017.

[16] Kai Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Scnet: Learning semantic correspondence. In *ICCV*, 2017.

[17] Or Hirschorn and Shai Avidan. A graph-based approach for category-agnostic pose estimation. In *ECCV*, 2024.

[18] Shuaiyi Huang, Luyu Yang, Bo He, Songyang Zhang, Xuming He, and Abhinav Shrivastava. Learning semantic correspondence with sparse annotations. In *ECCV*, 2022.

[19] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 1976.

[20] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *CVPR*, 2016.

[21] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024.

[22] Seungwook Kim, Juhong Min, and Minsu Cho. Transformatcher: Match-to-match attention for semantic correspondence. In *CVPR*, 2022.

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023.

[24] Nilesh Kulkarni, Abhinav Gupta, and Shubham Tulsiani. Canonical surface mapping via geometric cycle consistency. In *ICCV*, 2019.

[25] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Semi-local affine parts for object recognition. In *BMVC*, 2004.

[26] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 2010.

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

[28] Changsheng Lu and Piotr Koniusz. Few-shot keypoint detection with uncertainty learning for unseen species. In *CVPR*, 2022.

[29] Changsheng Lu and Piotr Koniusz. Detect any keypoints: An efficient light-weight few-shot keypoint detector. In *AAAI*, 2024.

[30] Changsheng Lu, Zheyuan Liu, and Piotr Koniusz. Openkd: Opening prompt diversity for zero-and few-shot keypoint detection. In *ECCV*, 2024.

[31] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. *NeurIPS*, 2023.

[32] Octave Mariotti, Oisin Mac Aodha, and Hakan Bilen. Improving semantic correspondence with viewpoint-guided spherical maps. In *CVPR*, 2024.

[33] Juhong Min and Minsu Cho. Convolutional hough matching networks. In *CVPR*, 2021.

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

[35] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Learning to compose hypercolumns for visual correspondence. In *ECCV*, 2020.

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

[37] Natalia Neverova, David Novotny, Marc Szafraniec, Vasil Khalidov, Patrick Labatut, and Andrea Vedaldi. Continuous surface embeddings. In *NeurIPS*, 2020.

[38] Dolev Ofri-Amar, Michal Geyer, Yoni Kasten, and Tali Dekel. Neural congealing: Aligning images to a joint semantic atlas. In *CVPR*, 2023.

[39] 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. *TMLR*, 2024.

[40] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *PAMI*, 2020.

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

[42] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[43] 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*, 2019.

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

[45] James Thewlis, Samuel Albanie, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of landmarks by descriptor vector exchange. In *ICCV*, 2019.

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

[47] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Warp consistency for unsupervised learning of dense correspondences. In *ICCV*, 2021.

[48] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Probabilistic warp consistency for weakly-supervised semantic correspondences. In *CVPR*, 2022.

[49] Nikolaos Tsagkas, Jack Rome, Subramanian Ramamoorthy, Oisin Mac Aodha, and Chris Xiaoxuan Lu. Click to grasp: Zero-shot precise manipulation via visual diffusion descriptors. In *IROS*, 2024.

[50] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *PAMI*, 1991.

[51] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset, 2011.

[52] Krispin Wandel and Hesheng Wang. Semalign3d: Semantic correspondence between rgb-images through aligning 3d object-class representations. In *CVPR*, 2025.

[53] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *ICCV*, 2023.

[54] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *CVPR*, 2025.

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

[56] Thomas Wimmer, Peter Wonka, and Maks Ovsjanikov. Back to 3d: Few-shot 3d keypoint detection with back-projected 2d features. In *CVPR*, 2024.

[57] Shangzhe Wu, Ruining Li, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. MagicPony: Learning articulated 3d animals in the wild. In *ICCV*, 2023.

[58] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *CVPR*, 2024.

[59] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *NeurIPS*, 2024.

[60] Hao Zhang, Lumin Xu, Shenqi Lai, Wenqi Shao, Nanning Zheng, Ping Luo, Yu Qiao, and Kaipeng Zhang. Open-vocabulary animal keypoint detection with semantic-feature matching. *IJCV*, 2024.

[61] 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.

[62] 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*, 2024.

[63] Xu Zhang, Wen Wang, Zhe Chen, Yufei Xu, Jing Zhang, and Dacheng Tao. Clamp: Prompt-based contrastive learning for connecting language and animal pose. In *CVPR*, 2023.

[64] Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros. Learning dense correspondence via 3d-guided cycle consistency. In *CVPR*, 2016.

# Appendix

## A Additional Implementation Details

**General implementation.** We base our model on Geo-SC [62], reusing all default hyperparameters that come with the official implementation[1], e.g., training for 2 epochs using AdamW [27] optimizer with $1.25 \times 10^{-3}$ initial learning rate and $1.0 \times 10^{-3}$ weight decay, coupled with one-cycle learning rate scheduler [43], with a batch size of 1. Every $5,000$ iterations, models are evaluated on the validation split, and the best performing model is retained. For evaluation, unless stated otherwise, a soft-argmax window of size 15 is used.

Experiments were performed on a single NVIDIA RTX 6000 Ada Generation, using pre-extracted DINOv2 and SD feature maps, depth maps, and segmentation masks. Training a model on SPair-71k consumes roughly 4.3GB of VRAM over 8 hours, representing an increased memory cost over Geo-SC's 2.9GB, mainly due to the many $\mathcal{X}_b$ and $\mathcal{X}_c$ we sample, and a doubling of runtime from roughly 4 hours. At inference time however, there is no impact as we estimate matches using features predicted with $\Phi$ in the exact same way Geo-SC does.

**Point cloud sampling.** When backprojecting image points using Eq. (2), we obtain a point cloud whose size depends on the number of visible object pixels. This can cause an imbalance of the samples in the final loss terms, with larger objects contributing more. Furthermore, these point clouds have a very specific grid-like structure inherited from the bitmap format of images, which is dense on surfaces parallel to the image plane and gets sparser as the angle increases. Therefore, we first subsample each training point cloud to a size of $k = 1024$ points using farthest point sampling to obtain a fixed number of well-distributed samples.

When computing the local geometry loss $\mathcal{L}_{\text{geom}}$, we use neighborhoods of size $k' = 64$. An ablation study of this parameter is provided in Table A1c. We also provide in Alg. 1 the pseudocode for the sampling strategy used to obtain $\mathbf{B}_r$ in the posed space.

**Input:** Point cloud $\mathcal{X}$, seed point $p$, number of neighbors $k$
**Output:** neighbor set $B$
`// Start with the seed`
$B \leftarrow \{p\}$;
**while** $|B| < k$ **do**
  `// Filter out already selected points`
  $\mathcal{X}' \leftarrow \mathcal{X} \setminus B$;
  **for** $x$ *in* $\mathcal{X}$**'** **do**
    `// Compute the distance to the closest point in` $B$
    $D_x \leftarrow \min_{y \in B} \|x - y\|_2$;
  **end**
  `// Add the point with minimal distance to` $B$
  $B \leftarrow B \cup \{\arg\min_{x \in \mathcal{X}'} D_x\}$;
**end**
**return** $B$;

**Algorithm 1:** Pseudo-geodesic sampling

## B Ablations

**General ablations.** We perform ablations of our designs in Table A1, and report results on the the SPair-71k [34] validation set which helps us chose the best performing model. It is not possible to ablate individual loss terms as they each have a distinct purpose without which the prototype cannot properly be learned: $\mathcal{L}_{\mathcal{P}}$ optimizes $\mathcal{P}$, $\mathcal{L}_{\mathcal{Z}}$ optimizes $\mathcal{Z}$, and $\mathcal{L}_{\text{geom}}$ provides a dense supervision signal, i.e., a loss for $\Phi(I, \boldsymbol{u})$ when $\boldsymbol{u}$ is an arbitrary object pixel, i.e., not a keypoint. We can however examine the different effect of our and Geo-SC's specific loss terms on both SPair-71k and SPair-U, by comparing a simple supervised approach using a DINOv2 and SD backbone trained only using $\mathcal{L}_{\text{sparse}}$, adding only the Geo-SC specific losses, adding only our canonical prototype losses, and

[1] https://github.com/Junyi42/geoaware-sc

Table A1: **Results of different ablations.**

(a) Average PCK@0.1 on SPair-71k test set and SPair-U for different models.

| DINOv2+SD | Geo-SC losses | Canonical prototype | Spair-71k | Spair-U |
|---|---|---|---|---|
| ✓ | | | 76.5 | 60.0 |
| ✓ | ✓ | | 85.6 | 57.1 |
| ✓ | | ✓ | 75.9 | 66.0 |
| ✓ | ✓ | ✓ | 85.4 | 66.1 |

(b) Average PCK@0.1 on SPair-71k validation set for general ablations.

| Ablation | PCK |
|---|---|
| $\lambda_{\mathcal{Z}} = 1$ | 85.9 |
| $\lambda_{\mathcal{Z}} = 0.1$ | 86.1 |
| K-nn sampling | 85.9 |
| Geodesic sampling | 86.2 |
| Full model | 86.5 |

(c) Average PCK@0.1 on SPair-71k validation set for different neighborhood size.

| Neighborhood size | PCK |
|---|---|
| 4 | 85.5 |
| 8 | 86.1 |
| 16 | 85.9 |
| 32 | 86.5 |
| 64 | 86.5 |
| 128 | 86.6 |
| 256 | 86.1 |

(d) Average PCK@0.1 on SPair-71k validation set for different rigidity constraints.

| Selected points | PCK |
|---|---|
| 3 | 86.6 |
| 4 | 86.4 |
| 5 | 86.1 |
| 6 | 86.4 |
| 7 | 86.5 |
| 8 | 86.0 |
| all | 86.5 |

our proposed model that combines them. Results shown in Table A1a show a clear pattern, i.e., adding our canonical prototype loss results in a very small drop in performance on seen keypoints (i.e., SPair-71k), which we attribute to the models having less capacity to fully overfit the training keypoint supervision, but endows them with the ability to generalize much better to unseen points (i.e., SPair-U). Conversely, the Geo-SC losses allow models to perform really well on seen keypoints but its effect on generalization ranges from harmful to null (with vs. without Geo-SC losses). These results also demonstrate that our contributions do not require Geo-SC to work, as they also boost performance of the supervised baseline on unseen keypoints (with vs. without Canonical prototype).

We show that setting $\lambda_{\mathcal{Z}}$ to $1$ or $0.1$ both negatively affect performance. We believe this is due to the interaction between $\mathcal{L}_{\mathcal{Z}}$ and $\mathcal{L}_{\text{geom}}$, as a high $\lambda_{\mathcal{Z}}$ would push $\Phi$ to collapse towards defaulting to predicting keypoint features $\mathcal{Z}$ for most points, while a weight too low prevents correct prediction on the keypoints. We also test different neighbor sampling strategies for $\mathcal{X}_b$ and $\mathcal{X}_c$, and show that sampling both spaces with either K-nearest neighbor or geodesic sampling is ineffective.

**Neighborhood size in $\mathbf{C}_q$ and $\mathbf{B}_r$.** We experiment with different neighborhood sizes when computing $\mathcal{L}_{\text{geom}}$ and valite the different models on the SPair-71k validation set in Table A1c. Results show little to no effect of the neighborhood size, which is consistent with our previous finding that our losses do not improve performance on semantics points that are present in the training set.

**Number of points in Eq. (3)** To compute $\mathcal{L}_{\mathcal{P}}$, we compute a global rigid transformation between the posed and canonical keypoints, which our qualitative analysis in Fig. 6 and Fig. A1 shows to be reasonable despite being a coarse simplification of the problem. In order to evaluate its impact quantitatively, we evaluate altered version of this procedure where we learn $\mathcal{P}$ not by globally aligning all keypoints but only a randomly sampled local subset of them. Results in Table A1d show that only considering a local neighborhood does not impact the validation performance of our model.

# C Additional Results

## C.1 Additional Metrics

Multiple recent works pointed out issues with evaluating using PCK, and proposed additional evaluation metrics to address its limitations.

**PCK$^\dagger$ [2]** PCK matches are counted correct even if the prediction lies closer to a keypoint that is not the target, which can lead to high scores when many points are grouped together, even though the system does not distinguish between them. The authors introduce PCK$^\dagger$ which only considers a match correct if it lies within the threshold *and* its closest annotated point is the target.

Table A2: **Evaluation under robust metrics.** All metrics use *per-image* averaging, and all models use window soft-argmax. All models are trained on SPair-71k, and models with a double dagger‡ benefit from AP-10K pretraining. Models in the $\mathcal{K}$ category use keypoint supervision, while $\not{\mathcal{K}}$ do not. Best results are **bolded** and second best are underlined.

| | | Spair-71k KAP | | | Spair-U KAP | | | Spair-71k PCK† | | | Spair-U PCK† | | | Spair-71k GA | | | AP-10K IS GA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Threshold | 0.01 | 0.05 | 0.10 | 0.01 | 0.05 | 0.10 | 0.01 | 0.05 | 0.10 | 0.01 | 0.05 | 0.10 | 0.01 | 0.05 | 0.10 | 0.01 | 0.05 | 0.10 |
| $\not{\mathcal{K}}$ | SD [41][61] | 38.2 | 47.5 | 53.0 | 43.4 | 51.6 | 58.5 | 6.3 | 34.4 | 44.4 | 3.3 | 27.7 | 45.4 | 4.2 | 28.3 | 43.3 | 1.3 | 15.3 | 31.0 |
| | DINOv2 [39][61] | 37.8 | 47.1 | 52.8 | 43.3 | 52.4 | 60.6 | 6.7 | 34.4 | 46.0 | 3.7 | 32.1 | 52.1 | 3.6 | 26.3 | 43.4 | 2.3 | 25.6 | 47.0 |
| | DINOv2+SD [61] | 38.4 | 49.6 | 55.9 | 43.7 | 54.2 | 62.8 | 8.1 | 41.0 | 52.8 | **4.7** | 36.6 | 56.6 | 4.8 | 32.7 | 50.8 | 2.4 | 26.1 | 48.1 |
| | SphericalMaps [32] | 38.9 | 51.2 | 58.2 | **44.3** | 55.4 | 64.2 | 8.8 | 44.4 | 57.3 | 4.5 | 37.8 | 58.5 | 5.6 | 37.7 | 58.1 | 2.6 | 28.4 | 51.8 |
| $\mathcal{K}$ | DINO+SD (S) [61] | 39.1 | 55.4 | 64.1 | 43.8 | 54.7 | 63.9 | 13.0 | 59.7 | 72.0 | 3.6 | 35.5 | 57.2 | 10.2 | 53.6 | 69.7 | 2.8 | 32.1 | 56.6 |
| | Geo-SC [62] | 39.8 | 59.3 | 67.8 | 43.8 | 54.2 | 62.8 | 20.0 | 69.8 | 78.3 | 4.6 | 35.1 | 54.9 | 17.2 | 65.6 | 78.0 | **3.7** | **33.6** | 55.3 |
| | Geo-SC‡ [62] | **40.1** | **61.0** | **69.2** | 43.8 | 54.1 | 62.8 | **22.0** | **73.0** | **80.9** | 4.3 | 35.8 | 55.3 | **20.0** | **70.9** | 82.3 | - | - | - |
| | Ours | 39.8 | 59.8 | 68.1 | 44.0 | 55.0 | 64.5 | 20.4 | 69.8 | 78.1 | 4.2 | 37.4 | 60.3 | 17.4 | 65.8 | 77.7 | 3.5 | 33.5 | **56.1** |
| | Ours‡ | 40.0 | 60.3 | 69.0 | 44.2 | **56.0** | **66.0** | 20.8 | 72.1 | 80.7 | 4.5 | **41.3** | **64.2** | 18.8 | 70.7 | **82.4** | - | - | - |

**KAP [32]** PCK only considers matches when both ground-truth points are visible and does not penalize systems that predict strong similarities for points that do not correspond, for instance between the two opposite sides of a car. KAP reformulates the correspondence evaluation as a binary classification problem between the pixels that are close to the target and those those that are not. Crucially, it penalizes high predictions when a source keypoint is invisible in the target.

**Geo-aware subset (GA) [62]** Finally, [2],[62] and [32] noted that SC pipelines - especially unsupervised ones - often make mistakes because of repeated parts and object symmetries. [62] proposed evaluation on the *Geo-aware* subset of points only, e.g., the points for which there is a symmetric corresponding point.

Results in Table A2 confirm the patterns observed in Section 5.2. For all metrics, supervised models performances drop back down to unsupervised-level or worse when evaluated outside their training labels. Interestingly, KAP scores do not widely vary between supervised and unsupervised models, indicating that supervised models are still likely to predict strong similarity between points when none exists.

### C.2 Additional Visualizations

We visualize more canonical surfaces in Fig. A1. While the shapes are sensible, we observe some limitations in adequately modeling categories with extreme deformations like birds: points belonging to the wings are predicted close to the body when they are folded, and away when they are spread. However, this is consistent with SPair-71k labeling, where the tips are only labeled when the wings are spread.

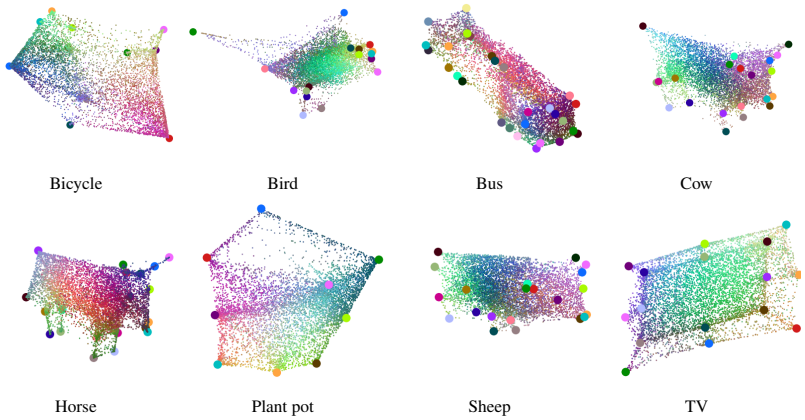



Figure A1: **Visualization of extra canonical shapes.** Large points correspond to $\mathcal{P}$, each being attributed a distinctive color for visualization. Small points are predicted canonical coordinates of objects, colored with PCA of the features predicted by $\varphi$.

We also show some predictions for the unsupervised DINOv2+SD, Geo-SC, and our model on SPair-U in Fig. A3. We observe some interesting failure cases: on the aeroplane, the unsupervised model correctly matches the door, while both supervised models incorrectly predict a training keypoint. In two occasions, Geo-SC predicts points outside of the object when queried on points that are far from training annotations (cow and person). Finally, two very challenging cases are shown with the chair and the tv, illustrating that generic semantic correspondence is still a particularly challenging task.

# D Additional SPair-U Details

We annotated images using the VGG Image Annotator [12]. We further post-processed the annotations into JSON files replicating the structure of SPair-71k annotations, i.e., per-image annotations and a list of testing pairs. This allows SPair-U to function as a drop-in replacement for SPair-71k evaluation in any semantic correspondence evaluation script. Note that it is designed to be a benchmark of unseen semantic points intended for evaluating the generalization ability of SC models, therefore does not come with a training or validation split. We present the full list of keypoint semantics of SPair-U in Table A3, per-category statistics in Table A4, and some keypoint visualization in Fig. A2.

Table A3: **List of SPair-U keypoint semantics.**

| | |
|---|---|
| Aeroplane | front-left, front-right, rear-left, rear-right doors |
| Bicycle | top and bottom of head tube; front brake; rear brake |
| Bird | center of back, chest; left wing wrist; right wing wrist |
| Boat | midpoint of the bow; front-left, front-right, rear-left, rear-right side midpoints |
| Bottle | center and corner points of label |
| Bus | top-left, top-right, bottom-left, bottom-right corners of windshield |
| Car | front-left, front-right, rear-left, rear-right top of the wheel arches |
| Cat | front-left, front-right, rear-left, rear-right hocks |
| Chair | leg midpoints; seat edge midpoints; seat center |
| Cow | left and right shoulder joints; left and right hip joints; left and right centers of the body; middle of back |
| Dog | front-left, front-right, rear-left, rear-right hocks |
| Horse | left and right shoulder joints; left and right hip joints |
| Motorbike | front fender midpoint; seat front edge, seat rear edge; engine compartment center |
| Person | forehead center; navel; neck base; left hip joint, right hip joint |
| Plant Pot | center of pot; midpoints of edges; midpoints of rim |
| Sheep | left and right shoulder joints; left and right hip joints |
| Train | locomotive rear top-left, top-right, bottom-left, bottom-right corners |
| Tv | center point; top-left, top-right, bottom-left, bottom-right quadrant centers |

Table A4: **Per-category statistics for our SPair-U benchmark.**

| | | | | | | | | | | | | | | | | | | | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Image count | 27 | 26 | 27 | 27 | 30 | 27 | 25 | 25 | 26 | 26 | 25 | 25 | 27 | 26 | 30 | 27 | 28 | 27 | 27 |
| Number of pairs | 254 | 576 | 480 | 666 | 338 | 304 | 300 | 510 | 552 | 466 | 488 | 420 | 536 | 488 | 744 | 218 | 314 | 600 | 458 |
| Count of new semantic labels | 4 | 4 | 4 | 5 | 5 | 4 | 4 | 4 | 9 | 7 | 4 | 4 | 4 | 5 | 5 | 4 | 4 | 5 | 4.7 |
| Total labeled points | 39 | 74 | 37 | 74 | 71 | 66 | 44 | 64 | 138 | 81 | 72 | 60 | 67 | 62 | 118 | 39 | 50 | 116 | 70.7 |
| Average number of visible points | 1.4 | 2.9 | 1.4 | 2.7 | 2.4 | 2.4 | 1.8 | 2.6 | 5.3 | 3.1 | 2.9 | 2.4 | 2.5 | 2.4 | 3.9 | 1.4 | 1.8 | 4.3 | 2.6 |
| Number of zero-kp images | 3 | 1 | 2 | 0 | 11 | 9 | 0 | 1 | 1 | 2 | 2 | 3 | 2 | 0 | 2 | 10 | 2 | 2 | 2.9 |
| Min keypoint occurrence | 8 | 14 | 2 | 11 | 13 | 17 | 10 | 14 | 11 | 8 | 17 | 15 | 16 | 9 | 22 | 9 | 12 | 23 | 12.8 |
| Avg keypoint occurrence | 10.8 | 19.5 | 10.3 | 15.8 | 15.2 | 17.5 | 12.0 | 17.0 | 16.3 | 12.6 | 19.0 | 16.0 | 17.8 | 13.4 | 24.6 | 10.8 | 13.5 | 24.2 | 15.9 |
| Max keypoint occurrence | 13 | 25 | 20 | 23 | 19 | 18 | 14 | 21 | 21 | 19 | 20 | 17 | 20 | 21 | 27 | 13 | 15 | 25 | 19.5 |
| Avg kp per pair | 1.4 | 2.4 | 1.2 | 1.7 | 2.9 | 3.4 | 1.5 | 1.9 | 3.8 | 2.0 | 2.5 | 2.0 | 2.0 | 1.7 | 3.6 | 1.6 | 1.9 | 4.3 | 2.3 |

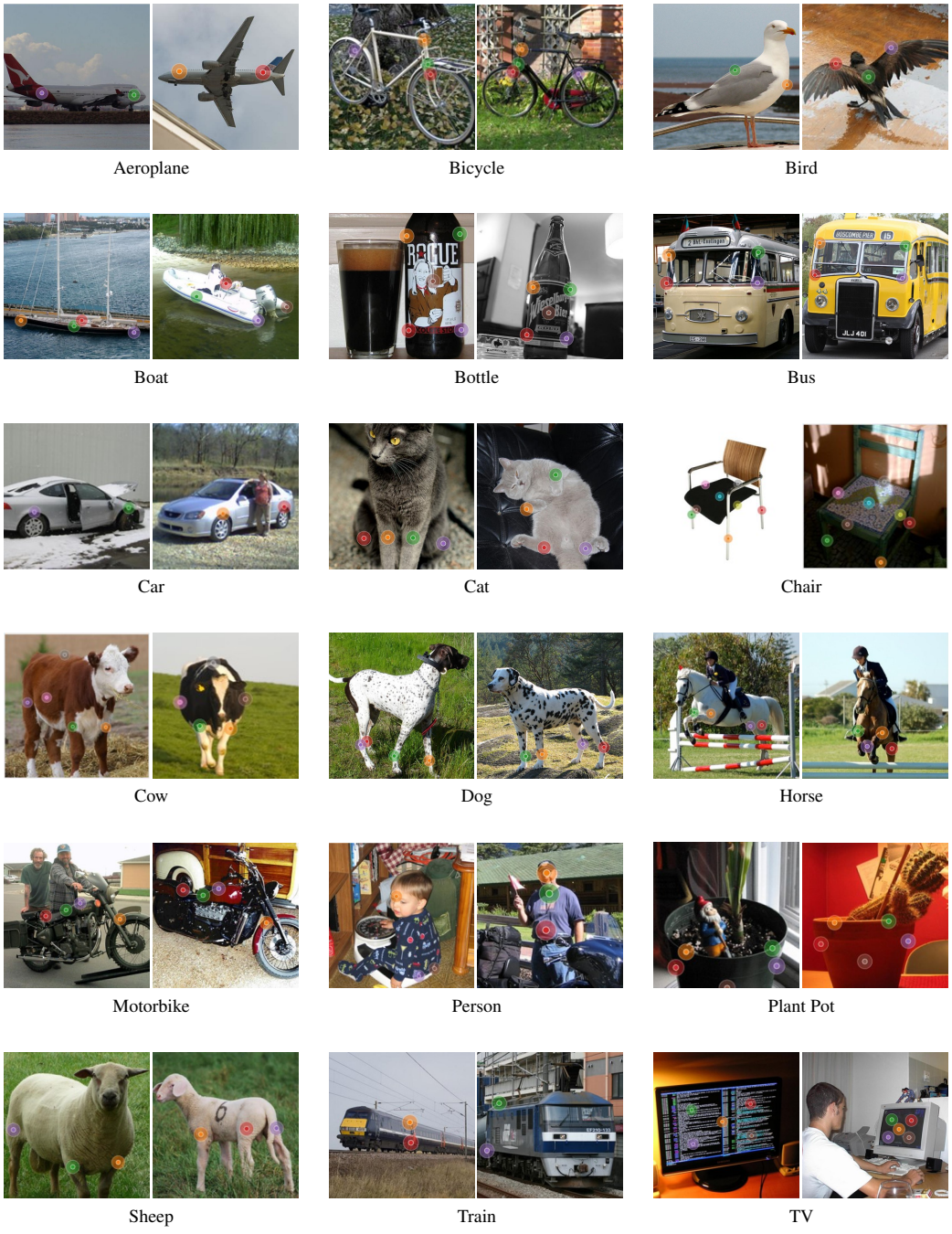



Figure A2: **Visualization of keypoint annotations from SPair-U.** Colors represent keypoint IDs.

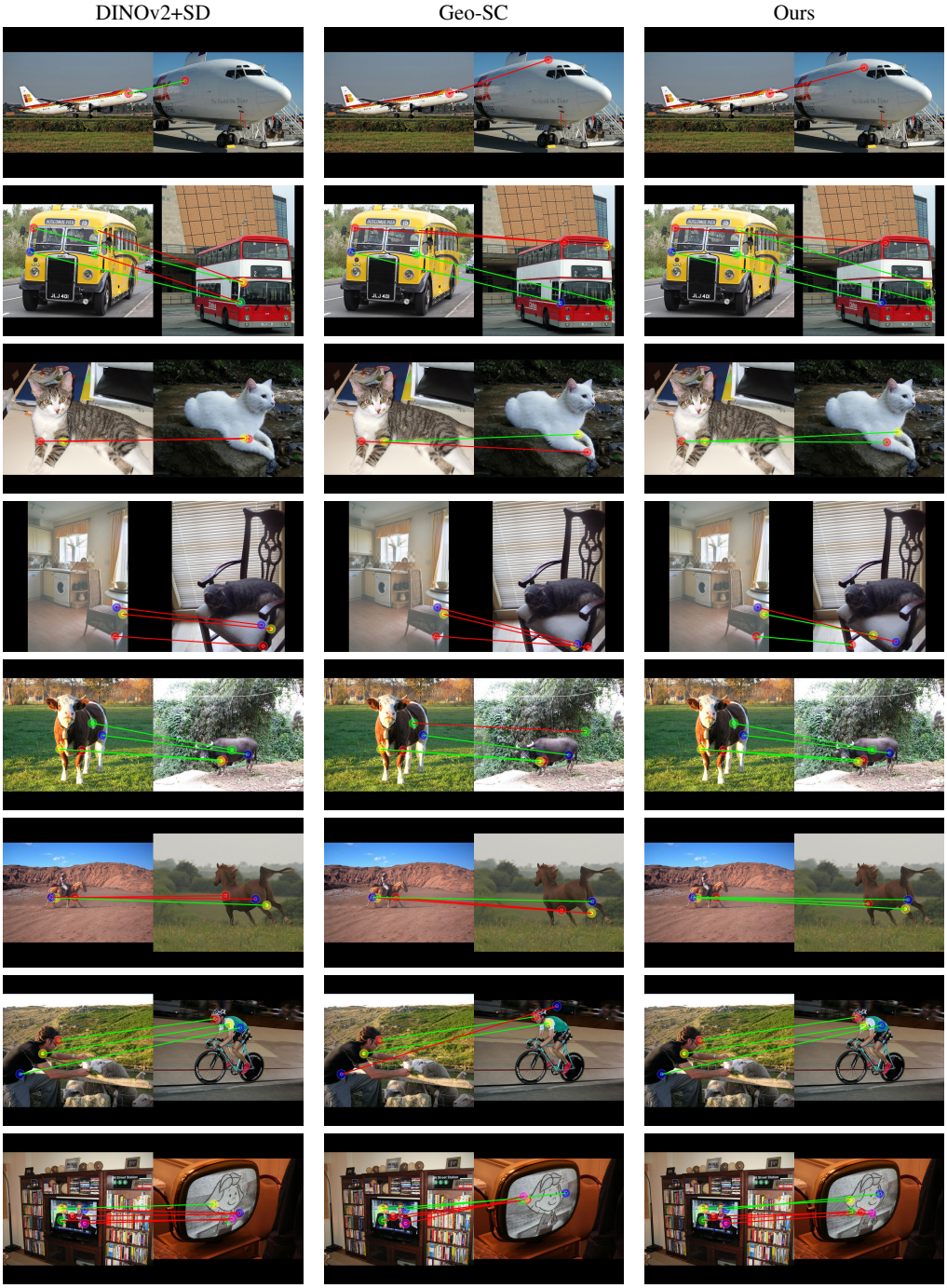



Figure A3: **Visualization matches for SPair-U.** Green lines are correct, red ones are incorrect.

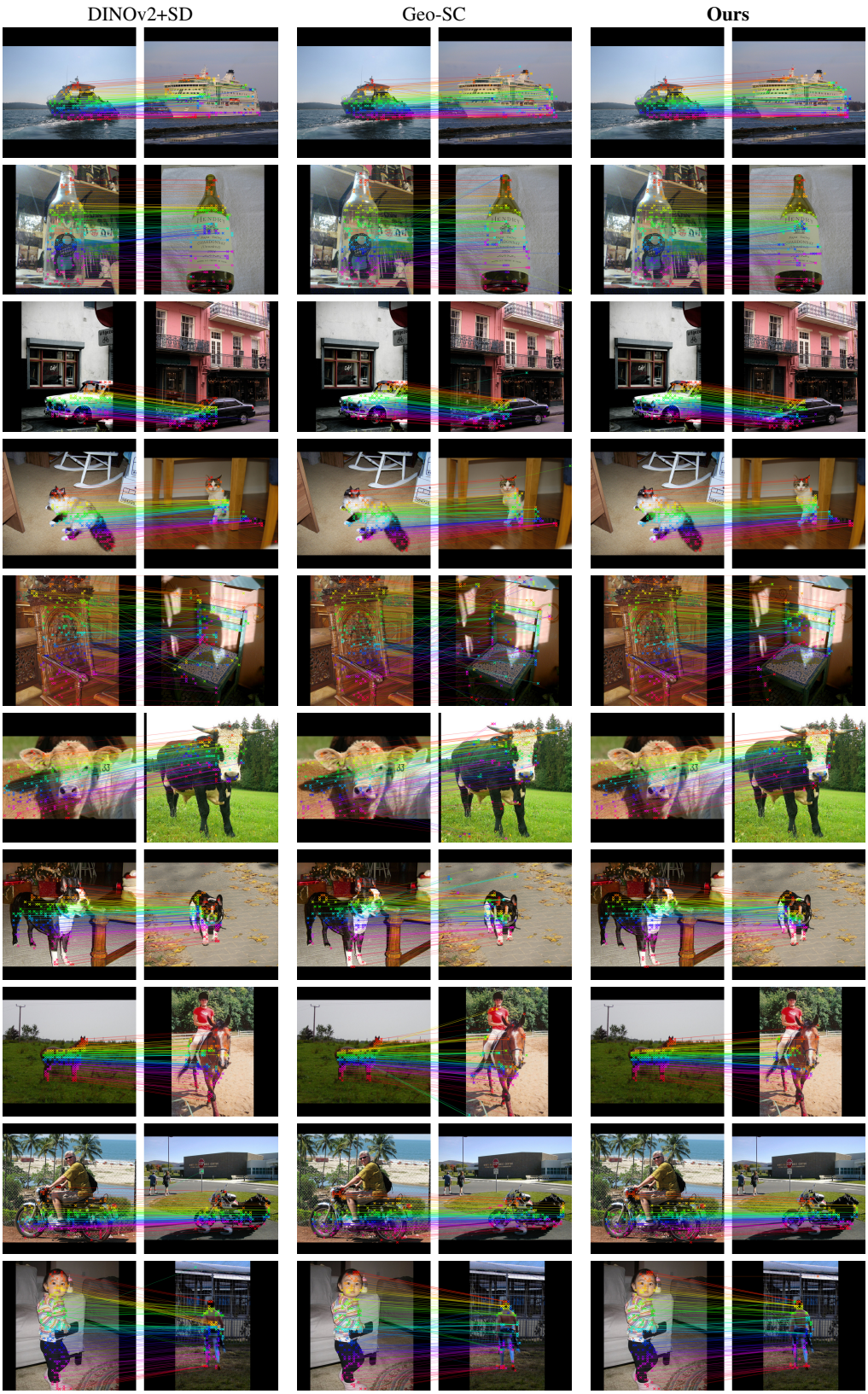



Figure A4: **Visualization matches for randomly selected object points.** Colors are provided as a way to distinguish the points.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We illustrate the lack of generalization of supervised SC models by introducing a novel benchmark of unseen points, while showing qualitatively and quantitatively that our proposed approach outperforms prior works in generalization settings

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss our limitations in a Section 6, where we highlight shortcomings in our proposed benchmark and discuss the key limitations associated the assumptions made when designing our model.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provided detailed implementation details in the supplementary material. Code for reproducing our results will be provided upon publication.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We plan to release both code for our model and experiments as well as our new SPair-U benchmark upon acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Detailed implementation details and training settings are provided in the supplementary material. We plan to release the code detailing training and evaluation settings upon acceptance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Retraining all methods evaluated on the multiple benchmarks we include in our experiments would incur a computational cost beyond our means. Instead, we report performance using the standard protocol used in the semantic correspondence literature, where results are averaged over many categories and many images.

   Guidelines:

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

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information related to computational resources used in our supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the code of ethics and confirm our paper conforms to it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: We briefly discuss broader impacts at the end of our limitations section. We do not anticipate any negative societal consequences stemming from our method. However, like any learning-based methods our approach is subject to potential limitations stemming from biases in the training data and there could be certain negative consequences in specific high stakes use cases if the outputs of the model are used without being validated first.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: As discussed in the previous answer, we do not anticipate any risks associated with the use of our method or SPair-U dataset.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite and discuss the relevant datasets we use for evaluation in Section 5.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We use the same data format for SPair-U as used in SPair-71k to ensure that it is a drop in replacement for users. We describe the data annotation process in Section 4.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our research does not involve crowdsourcing nor research with human subjects. The authors themselves were responsible for annotating the correspondences in SPair-U.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our research does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The topic of this paper does not overlap with LLMs. Beyond conventional spell checkers, no LLMs were used in the writing of text for this paper. We did use LLMs to assist with visualizing the results.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.