SemGeo-Gen: Unsupervised Generation of Approximate Cross-Instance Semantic-Geometric Correspondences

Roy Amoyal **Shira Ifergane** **Oren Freifeld**

The Stein Faculty of Computer and Information Science,
Ben-Gurion University of the Negev, Israel

amoyalr@post.bgu.ac.il shiraif@post.bgu.ac.il orenfr@bgu.ac.il

Abstract

We introduce SemGeo-Gen, an unsupervised method for generating approximate cross-instance semantic-geometric supervision from weakly aligned 3D object collections. Our goal is not to recover exact point-level correspondences, which are often ambiguous across different instances and would require expensive, impractical manual supervision at scale. Instead, we automatically produce approximate correspondences that are semantically meaningful, geometrically consistent, and diverse enough to train modern correspondence models. Given only coarse category-level rotation alignment, SemGeo-Gen lifts multi-view DINOv2 features to 3D, aligns object instances using continuous piecewise-affine registration, and prunes candidate matches using semantic consistency. The resulting 3D correspondences can be projected into rendered views, yielding scalable 3D–3D, 3D–2D, and 2D–2D supervision without manual annotation. We validate the generated correspondences against sparse human annotations from KeypointNet, obtaining 93% PCK@0.10 and 97% PCK@0.15. More importantly, we show that using the generated correspondences for synthetic pretraining improves recent state-of-the-art semantic correspondence models on SPair-71k after fine-tuning on real data, with gains of up to 5 PCK@0.10 points. Together, these results indicate that approximate semantic-geometric supervision generated from 3D assets can improve real-image correspondence learning and serve as a scalable alternative to costly manual annotation. Project page: https://bgu-cs-vil.github.io/SemGeo-Gen/.

1 Introduction

Establishing correspondences across different object instances within the same category is a fundamental problem in computer vision [17, 18, 24, 29, 30, 37, 38, 39, 40]. It underlies semantic matching, category-level reasoning, and downstream tasks such as object manipulation, image alignment, and visual understanding [2, 3, 12, 13, 21, 22, 32, 36, 39, 40, 41]. Compared to same-instance matching, cross-instance matching is substantially harder: objects may differ in geometry, proportions, local structure, texture, and appearance, while sharing only partially overlapping semantic parts. This paper focuses on semantic-geometric correspondences, where matches should preserve both semantic identity and geometric role when such distinctions are meaningful.

Progress in this direction is limited by the scarcity of scalable correspondence supervision. Existing datasets [5, 6, 9, 18, 19, 24, 27, 30, 36, 38, 41, 42] rely heavily on manual annotation, which is costly, sparse, and difficult to scale across categories. AP-10K [38] required 13 trained annotators, while Human Correspondence Consensus [19] uses 80 professional annotators and highlights the ambiguity of semantic point annotation. This ambiguity is inherent: across different instances, exact point-level

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

correspondences are often not well-defined. We therefore ask whether approximate correspondences, if semantically meaningful, geometrically consistent, and sufficiently diverse, can provide effective supervision at scale.

We introduce SemGeo-Gen, an unsupervised pipeline for generating approximate cross-instance semantic-geometric correspondences from weakly aligned 3D object collections. Given rotation-aligned rigid objects, SemGeo-Gen lifts multi-view DINOv2 features into 3D, uses them to assess semantic compatibility, aligns object instances with continuous piecewise-affine registration, and prunes geometrically plausible but semantically inconsistent matches. The resulting 3D correspondences can be projected into rendered views, yielding 3D–3D, 3D–2D, and 2D–2D supervision without manual annotation; see Fig. 1. The construction also scales naturally with the number of available assets: a category with N objects yields up to $\binom{N}{2}$ cross-instance pairs, and each accepted pair can generate dense 3D correspondences as well as many rendered 2D–2D training pairs.

We evaluate SemGeo-Gen in two complementary ways. First, we use the generated correspondences as synthetic pretraining data for recent semantic correspondence models and then fine-tune them on real SPair-71k annotations [24], improving strong recent models without architectural modifications. Second, we compare our generated 3D correspondences to sparse human annotations from KeypointNet [36], obtaining 93% PCK@0.10, 97% PCK@0.15, and 99% PCK@0.20. These results indicate that automatically generated approximate correspondences, when constructed from 3D geometry and semantic features, can provide scalable supervision for cross-instance correspondence learning.

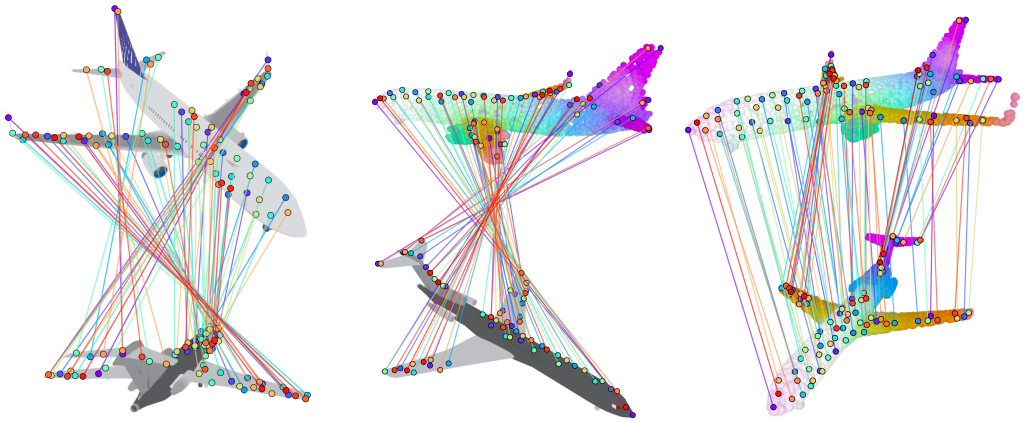
Our contributions are as follows:

1. We introduce SemGeo-Gen, a fully automatic and unsupervised pipeline for generating dense approximate cross-instance semantic-geometric correspondences from weakly aligned 3D object collections, without point, part, or correspondence supervision.
2. We combine lifted multi-view semantic features, deformable continuous piecewise-affine registration, and semantic correspondence pruning to generate matches constrained by both geometry and semantics.
3. We project the generated 3D correspondences into rendered views, enabling combinatorially scalable 3D–3D, 3D–2D, and 2D–2D supervision for training image-based correspondence models.
4. We show that the generated supervision improves recent semantic correspondence models on SPair-71k after real-data fine-tuning, yielding gains of up to 5 PCK points, while also agreeing well with sparse human KeypointNet annotations.

2 Related Work

2D and 3D Cross-Instance Correspondence Datasets. A wide range of datasets have been proposed for cross-instance correspondence learning, primarily in 2D. Early benchmarks such as PF-PASCAL [11] and SPair-71k [24] provide sparse keypoint annotations across object instances under significant viewpoint and scale variation, while CUB [30] provides fine-grained bird keypoints. AP-10K [38], although originally introduced for animal pose estimation, has been used by GeoAware-SC [40] to derive 2D semantic correspondence supervision from shared pose keypoints. Similarly, MARCO [5] derives semantic correspondences from large-scale pose estimation data [35], and Jamais Vu [23] extends SPair-71k with SPair-U to evaluate generalization to unseen keypoints. Despite their success, these datasets remain image-based and rely on human-annotated keypoints or pose supervision, providing predominantly sparse correspondence annotations.

In 3D, existing correspondence datasets, including SHREC [6, 27], parametric animal shape models such as SMAL [42], and deformation benchmarks [9], primarily focus on geometry and often target a single category or articulated shapes such as humans or animals. They also typically lack texture or appearance information, limiting their ability to support projected 2D–2D correspondences for training image-based correspondence models. More recently, DenseMatcher [41] introduced DenseCorr3D, a dataset of colored meshes with part-level annotations across 23 categories, but it does not provide scalable fine-grained point-to-point supervision. KeypointNet [36] provides the largest 3D cross-instance keypoint dataset for rigid objects, but it is limited to 16 categories with sparse annotations, averaging roughly 12 keypoints per object.



(a) 2D–2D (b) 3D–2D (c) 3D–3D

Figure 1: Approximate cross-instance semantic-geometric correspondences between two airplane instances across modalities. Left: 2D–2D correspondences between rendered views. Middle: 3D–2D correspondences between a 3D object and a rendered view. Right: 3D–3D correspondences between object surfaces. Colors in 3D visualize the lifted DINOv2 features reduced to three dimensions.

Overall, existing datasets rely on expensive and sparse human annotations, limiting scalability. In contrast, our work automatically generates large-scale approximate semantic-geometric correspondences directly between 3D object instances, enabling scalable supervision across diverse rigid object categories and naturally supporting 3D–3D, 3D–2D, and 2D–2D correspondence tasks.

2D Semantic-Geometric Correspondences. Semantic correspondence aims to identify analogous locations across different instances of the same category. However, existing work differs in how it treats the geometric specificity of the correspondence. A first line of work studies semantic-only correspondence, often formulated as cross-instance image alignment [3, 10, 13, 25]. In this setting, geometrically symmetric points may be treated as equivalent if they share the same semantic identity; for example, points on the left and right wings of an airplane may both be considered valid under flipped alignments. Accordingly, these methods often allow image flips or flipped keypoint annotations during evaluation on datasets such as SPair-71k and CUB [24, 30].

A second line of work follows the standard semantic correspondence protocol [4, 5, 12, 21, 22, 23, 31, 39, 40], typically evaluated on SPair-71k [24] without annotation flips. This setting requires models to resolve both semantic identity and geometric orientation when such distinctions are well defined, such as front/back, left/right, or top/bottom. However, for symmetric categories such as tables, bottles, or potted plants, geometric distinctions can be inherently ambiguous, and evaluation often depends on image-relative location rather than a unique semantic-geometric correspondence.

A third perspective, emphasized by GeoAware-SC [40], isolates cases of geometry-aware correspondence in which appearance similarity alone is insufficient and resolving the correct match requires reasoning about object orientation or relative part location. Their results show that existing methods often underperform on such cases, motivating our goal of generating richer semantic-geometric supervision at scale.

3D Semantic Keypoints and Correspondence Methods. Recent 3D correspondence methods increasingly combine appearance and geometry to obtain semantic matches across shapes. DenseMatcher [41] learns dense 3D semantic descriptors for textured meshes by projecting multi-view 2D foundation-model features onto mesh vertices, refining them with a 3D network, and computing correspondences with functional maps; its main focus is category-level manipulation from a limited set of annotated objects. UKPGAN [37] is an unsupervised 3D keypoint detector that predicts sparse salient keypoints for individual shapes, but does not directly establish cross-instance correspondences. Skeleton Merger [29] learns aligned sparse keypoints through skeleton-based reconstruction, yielding category-level keypoint alignment across instances. However, UKPGAN and Skeleton Merger focus on sparse keypoints, while DenseMatcher relies on costly part-level annotations and a limited set

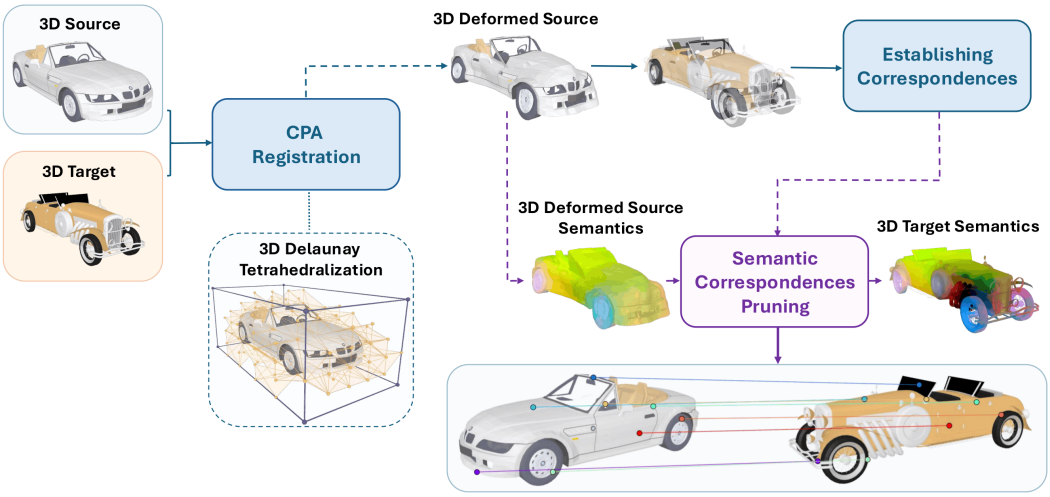


Figure 2: Overview of our correspondence generation pipeline.

of annotated objects. None provides scalable, annotation-free dense cross-instance correspondence supervision across diverse rigid-object categories.

3 Method

We propose a fully automatic pipeline for generating approximate 3D–3D cross-instance semantic-geometric correspondences between rigid objects, which can be projected into rendered views to obtain 3D–2D and 2D–2D correspondences. The pipeline's input is a collection of object pairs $\mathcal{P} = \{(\mathcal{S}_k, \mathcal{T}_k)\}_{k=1}^{K}$, where each source object $\mathcal{S}_k$ and target object $\mathcal{T}_k$ are approximately rotation-aligned. We normalize each object to the unit sphere. This assumption provides a weak category-level geometric prior, but does not require any point-level, part-level, or correspondence supervision.

For each object pair, our pipeline proceeds in three main stages. First, we extract per-point semantic descriptors by rendering each object from multiple views, computing dense DINOv2 features, and lifting and back-projecting them to 3D (§ 3.1). Second, we estimate a continuous piecewise-affine (CPA) deformation from the source to the target and extract candidate correspondences via nearest-neighbor assignment after alignment. Third, we prune semantically inconsistent matches, retaining approximate correspondences that are jointly constrained by deformable geometric alignment and multi-view semantic consistency.

In addition, we optionally apply a bidirectional semantic-coverage filter (§ 3.3) before registration to reject object pairs with insufficient semantic overlap. This step reduces the number of retained pairs, but improves the quality of the remaining pairs and the resulting correspondences. The pipeline is illustrated in Fig. 2 and summarized in Algorithm 1.

3.1 Object Representation and Feature Lifting

Each object $\mathcal{O}$ is represented as a point cloud with an associated semantic feature map:

$$\mathcal{O} = (\mathcal{X}, \mathcal{F}), \qquad \mathcal{X} = \{x_i\}_{i=1}^{N},\ x_i \in \mathbb{R}^3, \qquad \mathcal{F} = \{f_i\}_{i=1}^{N},\ f_i \in \mathbb{R}^C. \tag{1}$$

Here, x_i denotes a 3D point sampled from the object surface, and f_i denotes its corresponding semantic descriptor. To obtain these descriptors, we render each object from a set of calibrated virtual cameras, extract dense DINOv2 [26] features in image space, and lift them back to the visible 3D points using known camera projection and depth. We then aggregate multi-view observations by averaging to obtain a single semantic descriptor for each 3D point. For a source-target pair $(\mathcal{S}, \mathcal{T})$, we write

$$\mathcal{S} = (\mathcal{X}_S, \mathcal{F}_S), \qquad \mathcal{T} = (\mathcal{X}_T, \mathcal{F}_T), \tag{2}$$

where $\mathcal{X}_S = \{x_i^S\}_{i=1}^{N_S}$, $\mathcal{X}_T = \{x_j^T\}_{j=1}^{N_T}$, $\mathcal{F}_S = \{f_i^S\}_{i=1}^{N_S}$, and $\mathcal{F}_T = \{f_j^T\}_{j=1}^{N_T}$.

3.2 Semantic Similarity

We measure semantic compatibility using cosine similarity between DINOv2 [26] descriptors jointly reduced to 64 dimensions via Principal Component Analysis (PCA). Let z_i denote the reduced descriptor of point x_i. We define

$$s_{\text{sem}}(x_i, x_j) = \frac{z_i^\top z_j}{\|z_i\|_2 \|z_j\|_2}. \tag{3}$$

Two points are considered semantically compatible if

$$s_{\text{sem}}(x_i, x_j) > \tau_{\text{sem}}, \tag{4}$$

where $\tau_{\text{sem}} = 0.575$. In our implementation, this corresponds to a squared Euclidean distance threshold of 0.85 between L2-normalized descriptors, since $\|\hat{z}_i - \hat{z}_j\|_2^2 = 2(1 - s_{\text{sem}}(x_i, x_j))$, where $\hat{z} = z/\|z\|_2$.

3.3 Optional Semantic Coverage Pair Filtering

As an optional preprocessing stage, we apply bidirectional semantic-coverage filtering before registration.

For a source-target pair $(\mathcal{S}, \mathcal{T})$, we define the source-to-target semantic coverage as

$$\rho_{\text{sem}}(\mathcal{S} \to \mathcal{T}) = \frac{1}{|\mathcal{X}_S|} \sum_{x_i^S \in \mathcal{X}_S} \mathbf{1}\left[\max_{x_j^T \in \mathcal{X}_T} s_{\text{sem}}(x_i^S, x_j^T) > \tau_{\text{sem}}\right], \tag{5}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function, which equals 1 if the condition inside the brackets is true and 0 otherwise. This score measures the fraction of source points that have at least one semantically compatible point in the target. We also compute the same quantity in the reverse direction, $\rho_{\text{sem}}(\mathcal{T} \to \mathcal{S})$, and retain the object pair only if

$$\rho_{\text{sem}}(\mathcal{S} \to \mathcal{T}) \geq \gamma \quad \text{and} \quad \rho_{\text{sem}}(\mathcal{T} \to \mathcal{S}) \geq \gamma. \tag{6}$$

In our experiments, we use $\gamma = 0.6$.

3.4 Continuous Piecewise-Affine Registration

For each retained object pair $(\mathcal{S}, \mathcal{T})$, we estimate a non-rigid transformation that aligns the source point set $\mathcal{X}_S$ to the target $\mathcal{X}_T$. We model this transformation using a continuous piecewise-affine (CPA) mapping

$$T_{\text{CPA}} : \mathbb{R}^3 \to \mathbb{R}^3, \tag{7}$$

which produces a continuous, spatially coherent deformation field while allowing local shape variation across object instances. This transformation is parameterized by a regular control grid over the normalized 3D domain, which is decomposed into tetrahedral cells using Delaunay triangulation. Each grid vertex is associated with an optimized displacement vector, and we obtain the displacement of an arbitrary source point by barycentric interpolation from the vertices of the tetrahedron containing it. This yields a deformation field that is locally affine within each tetrahedron and continuous across neighboring cells. Unlike CPAB transformations [1, 8], which are obtained by integrating CPA velocity fields, here we use CPA mappings directly as transformations.

We estimate the transformation by optimizing a Chamfer distance loss [7] with smoothness regularization via gradient descent:

$$\mathcal{L}_{\text{CPA}} = \mathcal{L}_{\text{Chamfer}}\left(T_{\text{CPA}}(\mathcal{X}_S), \mathcal{X}_T\right) + \lambda_{\text{smooth}} \mathcal{R}(T_{\text{CPA}}), \tag{8}$$

where $\lambda_{\text{smooth}} = 0.3$ in all experiments, and

$$\mathcal{L}_{\text{Chamfer}}(\widehat{\mathcal{X}}_S, \mathcal{X}_T) = \frac{1}{|\widehat{\mathcal{X}}_S|} \sum_{\hat{x}_i^S \in \widehat{\mathcal{X}}_S} \min_{x_j^T \in \mathcal{X}_T} \left\|\hat{x}_i^S - x_j^T\right\|_2^2 + \frac{1}{|\mathcal{X}_T|} \sum_{x_j^T \in \mathcal{X}_T} \min_{\hat{x}_i^S \in \widehat{\mathcal{X}}_S} \left\|x_j^T - \hat{x}_i^S\right\|_2^2. \tag{9}$$

The first term aligns the deformed source to the target, while the second term encourages the target surface to be covered by the deformed source.

Algorithm 1: Automatic Approximate Semantic-Geometric Correspondence Generation

Input: Rotation-aligned and scale-normalized object pairs $\mathcal{P} = \{(\mathcal{S}_k, \mathcal{T}_k)\}_{k=1}^{K}$; lifted 3D DINOv2 features; semantic similarity threshold τ_{sem}; optional coverage threshold γ.
Output: Approximate semantic-geometric correspondences.
foreach $(\mathcal{S}, \mathcal{T}) \in \mathcal{P}$ **do**
  Let $(\mathcal{X}_S, \mathcal{F}_S)$ and $(\mathcal{X}_T, \mathcal{F}_T)$ be the source and target representations
  if *semantic coverage filtering is enabled* **then**
    if $\rho_{\text{sem}}(\mathcal{S} \to \mathcal{T}) < \gamma$ ***or*** $\rho_{\text{sem}}(\mathcal{T} \to \mathcal{S}) < \gamma$ **then**
      continue
  Estimate T_{CPA} by minimizing Eq. (8)
  Compute the deformed source $\widehat{\mathcal{X}}_S = T_{\text{CPA}}(\mathcal{X}_S)$
  Assign each $\hat{x}_i^S \in \widehat{\mathcal{X}}_S$ to its nearest neighbor in $\mathcal{X}_T$ using Eq. (12)
  Construct candidate correspondences $\mathcal{C}^0_{\mathcal{S},\mathcal{T}}$
  Prune correspondences using the semantic criterion in Eq. (14)
  Store the final correspondence set $\mathcal{C}_{\mathcal{S},\mathcal{T}}$
Return all processed object pairs that were not filtered out and their correspondence sets.

The smoothness term regularizes neighboring control-grid displacements:

$$\mathcal{R}(T_{\text{CPA}}) = \frac{1}{|\mathcal{E}|} \sum_{(v_a, v_b) \in \mathcal{E}} \|u_{v_a} - u_{v_b}\|_2^2, \tag{10}$$

where $\mathcal{E}$ denotes edges between neighboring vertices in the tetrahedral control grid, and $u_v \in \mathbb{R}^3$ denotes the displacement vector associated with vertex v. This regularization promotes locally coherent deformations while still allowing non-rigid cross-instance variation. After optimization, the transformed source point cloud is given by

$$\widehat{\mathcal{X}}_S = T_{\text{CPA}}(\mathcal{X}_S) = \{\hat{x}_i^S\}_{i=1}^{N_S}. \tag{11}$$

3.5 Approximate Correspondence Extraction

After CPA registration, the source and target are approximately aligned geometrically. We then assign each transformed source point to its nearest neighbor in the target point cloud. For each $\hat{x}_i^S \in \widehat{\mathcal{X}}_S$, the target index is

$$a(i) = \arg \min_{j \in \{1,\dots,N_T\}} \left\|\hat{x}_i^S - x_j^T\right\|_2. \tag{12}$$

This produces an initial set of candidate correspondences:

$$\mathcal{C}^0_{\mathcal{S},\mathcal{T}} = \left\{ \left(x_i^S, x_{a(i)}^T\right) \;\middle|\; i = 1, \dots, N_S \right\}. \tag{13}$$

Importantly, the assignment is computed using the deformed source point $\hat{x}_i^S$, while the stored correspondence is defined between the original source point x_i^S and the target point $x_{a(i)}^T$. These matches should therefore be interpreted as approximate cross-instance correspondences induced by deformation-based alignment, rather than exact point-level annotation.

3.6 Semantic Consistency Pruning

Geometric proximity alone may produce implausible matches in close regions with different semantics or regions with no meaningful semantic counterpart. We therefore apply a final semantic pruning step to retain only candidate matches that are also semantically compatible. A candidate correspondence $(x_i^S, x_{a(i)}^T) \in \mathcal{C}^0_{\mathcal{S},\mathcal{T}}$ is retained only if

$$s_{\text{sem}}\left(x_i^S, x_{a(i)}^T\right) > \tau_{\text{sem}}. \tag{14}$$

The final semantic-geometric correspondence set is

$$\mathcal{C}_{\mathcal{S},\mathcal{T}} = \left\{ \left(x_i^S, x_{a(i)}^T \right) \in \mathcal{C}_{\mathcal{S},\mathcal{T}}^0 \;\middle|\; s_{\text{sem}} \left(x_i^S, x_{a(i)}^T \right) > \tau_{\text{sem}} \right\}. \tag{15}$$

This step removes matches that are geometrically nearby after deformation but semantically inconsistent according to the lifted DINOv2 [26] features. The remaining set forms our final approximate semantic-geometric correspondences.

3.7 Projection to 2D Correspondences

The generated correspondences, established in 3D, can be directly projected into rendered views to produce aligned 3D–3D, 3D–2D, or 2D–2D supervision. Given calibrated rendering views with camera projections π_{c_S} and π_{c_T}, a 3D correspondence $(x_i^S, x_j^T) \in \mathcal{C}_{\mathcal{S},\mathcal{T}}$ induces the 2D projections

$$(u_i^S, u_j^T) = \left(\pi_{c_S}(x_i^S), \pi_{c_T}(x_j^T) \right), \tag{16}$$

where c_S and c_T denote the selected source and target camera views. Projecting one endpoint yields 3D–2D supervision, while projecting both endpoints yields 2D–2D supervision. In either case, we retain only correspondences whose projected endpoints are visible in their respective rendered views. By rendering multiple views of each object pair, we obtain large-scale automatically generated training data for image-based semantic correspondence models.

4 Experiments

We evaluate the proposed correspondence-generation pipeline in two complementary settings. First, we assess whether the generated correspondences can improve real-image semantic correspondence models when used as synthetic pretraining data (§ 4.2). Second, we directly evaluate the quality of the generated 3D correspondences against sparse human-annotated 3D ground truth(§ 4.3). All experiments were conducted on a single NVIDIA RTX A6000 GPU.

4.1 Experimental Data

Our method assumes collections of rigid 3D objects that are approximately rotation-aligned, providing a weak category-level geometric prior. In this work, we use two complementary 3D sources, **KeypointNet** [36] and **Objaverse-OA** [20], to validate the proposed pipeline. **KeypointNet** provides sparse human-annotated semantic keypoints on approximately aligned 3D object models. It has relatively few rigid categories, but many objects per category, making it suitable for evaluating how well the generated correspondences agree with human semantic annotations. These annotations are used only for evaluation, not for training or correspondence generation. **Objaverse-OA** provides far broader category and shape diversity, containing over 100 rigid categories after curation, but without correspondence annotations. Appendix A shows example correspondences generated by our method on several Objaverse-OA categories beyond those covered by existing correspondence datasets.

For downstream SPair-71k [24] experiments, we generate synthetic correspondences from nine rigid categories (aeroplane; bicycle; boat; bus; car; chair; motorbike; train; tvmonitor), excluding non-rigid and strongly symmetric objects. We evaluate two settings: fine-tuning on the same Rigid-9 categories, and fine-tuning on all 18 categories, additionally including Non-rigid-7 (bird; cat; cow; dog; horse; person; sheep) and Symmetric-2 (bottle; potted plant). Both settings use the same Rigid-9 synthetic pretraining data.

4.2 Results on SPair-71k

We evaluate the synthetic 2D correspondence supervision generated by our pipeline on SPair-71k [24], a standard benchmark for cross-instance semantic correspondence. Concretely, we generate 12K synthetic 2D training pairs per category from diverse 3D object instances sampled from Objaverse-OA [20]. To preserve category and shape diversity in the synthetic pretraining data, we avoid pair-level semantic filtering in this experiment. We focus on 9 rigid categories with well-defined 3D semantic-geometric structure: aeroplane; bicycle; boat; bus; car; chair; motorbike; train; tvmonitor. We report reproduced results for recent methods, including GeoAware-SC [40], Jamais Vu [23], and MARCO [5], alongside reported results for earlier methods in Table 1. For the state-of-the-art

Table 1: Category-wise PCK@0.10 results on selected SPair-71k categories with well-defined semantic-geometric correspondences. P.T. denotes pretraining.

Method	Training regime	aero	bike	boat	bus	car	chair	mbike	train	tv	Mean
Zero-shot / weakly supervised methods											
SD [28, 39]	Zero-shot	62.8	52.7	31.2	39.1	35.6	32.0	51.0	61.8	52.9	46.57
DINOv2 [26, 39]	Zero-shot	73.4	60.2	43.2	46.7	45.1	33.4	60.7	54.2	23.9	48.98
DINOv2+SD [39]	Zero-shot	73.8	61.0	40.2	47.4	44.1	41.5	61.7	63.5	52.4	53.96
SphericalMaps [22]	Weakly supervised (SPair-71k)	76.2	60.1	46.5	74.9	68.0	45.1	69.1	73.9	58.1	63.54
Supervised methods											
SCorrSan [14]	SPair-71k	57.1	40.3	38.1	57.8	47.1	25.2	45.3	77.7	69.7	50.92
CATS++ [4]	SPair-71k	60.6	46.9	41.6	64.9	50.4	29.2	50.9	80.9	74.9	55.59
DHF [21]	SPair-71k	74.0	61.0	40.7	70.0	74.4	38.5	66.6	87.4	60.3	63.66
DINO+SD (S) [39]	SPair-71k	84.7	67.5	64.5	85.7	82.0	57.0	75.9	93.6	70.5	75.71
Jamais Vu [23]	SPair-71k	83.6	70.1	64.1	91.4	83.2	61.4	78.4	95.5	80.1	78.64
Recent supervised state-of-the-art methods under different supervision settings											
GeoAware-SC [40]	SPair-71k	83.42	71.07	66.68	90.93	86.10	62.18	79.54	95.33	79.00	79.36
	SPair-71k + Ours	86.28	74.60	68.33	93.12	84.73	68.94	79.73	95.31	80.73	81.31
	Ours P.T. → SPair-71k	**89.38**	**76.11**	**71.15**	**93.80**	**86.74**	**73.80**	**82.33**	**95.67**	**84.59**	**83.73**
MARCO [5]	SPair-71k	92.16	76.76	76.53	93.17	88.57	82.49	82.03	93.62	91.72	86.34
	SPair-71k + Ours	92.26	77.49	73.74	92.59	84.86	82.71	79.78	93.22	89.72	85.15
	Ours P.T. → SPair-71k	**95.59**	**80.66**	**79.35**	**94.59**	**91.89**	**91.16**	**82.30**	**94.69**	**93.49**	**89.30**

GeoAware-SC and MARCO, we also evaluate the effect of our generated data in three training regimes: training on SPair-71k only; joint training with our projected correspondences; pretraining on our generated correspondences followed by fine-tuning on SPair-71k.

Since our data is generated from synthetic 3D objects, a synthetic-to-real domain gap remains, and naive joint training gives mixed results. In contrast, the standard pretrain-and-finetune regime consistently improves both evaluated models. GeoAware-SC [40] improves from 79.36 to 83.73 mean PCK (+4.37), and MARCO [5] improves from 86.34 to 89.30 (+2.96). These gains are obtained without any architectural modification, highlighting that even approximate semantic-geometric correspondences provide a useful geometric-semantic training signal that is easier to obtain than manual annotations. Additional qualitative examples in Fig. 3 further illustrate the improved semantic-geometric consistency obtained after pretraining on our generated data.

Beyond rigid categories. We also investigate whether synthetic pretraining on rigid categories transfers to non-rigid and symmetric objects. As Table 2 shows, pretraining GeoAware-SC on our Rigid-9 synthetic correspondences followed by fine-tuning on all 18 SPair-71k categories improves PCK@0.10 by 3.72 on rigid categories, 1.29 on non-rigid categories, and 2.39 overall, compared to real-only training. Notably, these gains extend to categories excluded from synthetic pretraining.

Table 2: **Generalization across all SPair-71k categories.** GeoAware-SC pretrained on Rigid-9 synthetic correspondences and fine-tuned on all 18 SPair-71k categories. PCK@0.10 (%); improvements over real-only training are shown in parentheses.

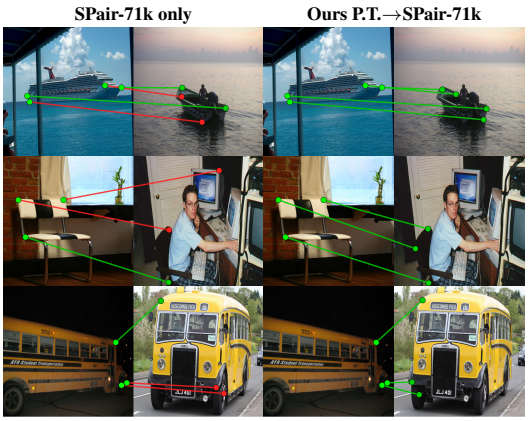
Training regime	Rigid-9	Non-rigid-7	Sym.-2	All-18
Real only	80.48	86.94	72.35	82.09
DenseMatcher [41] P.T. → Real	80.37	86.54	71.97	81.83
Ours P.T. → Real	**84.20 (+3.72)**	**88.23 (+1.29)**	**72.58 (+0.23)**	**84.48 (+2.39)**

We additionally compare against DenseMatcher [41] under the same training protocol, which performs similarly to real-only training (81.83 vs. 82.09 PCK overall). DenseMatcher combines semantic features with geometric refinement but requires costly part annotations, with only 5 animals, 5 chairs, and 6 vehicles in its released training data. Its functional-map matching may also struggle with large structural variations, consistent with the symmetry-mismatched and structurally inconsistent correspondences we observed on diverse Objaverse-OA instances. In contrast, SemGeo-Gen generates dense approximate semantic-geometric correspondences from scalable collections of rotation-aligned 3D models, without part or correspondence annotations.

Table 3: Evaluation of annotation quality on KeypointNet [36] correspondences.

Category	PCK@.05 ↑	PCK@.10 ↑	PCK@.15 ↑	PCK@.20 ↑
Airplane	0.79	0.94	0.98	0.99
Bathtub	0.70	0.88	0.96	0.99
Bed	0.68	0.91	0.96	0.99
Bottle	0.83	0.96	0.98	0.99
Cap	0.80	0.97	1.0	1.00
Car	0.78	0.95	0.99	1.00
Chair	0.77	0.96	0.99	0.99
Guitar	0.81	0.96	0.98	0.99
Helmet	0.55	0.84	0.94	0.98
Knife	0.83	0.93	0.96	0.98
Laptop	0.89	0.99	1.00	1.00
Motorcycle	0.75	0.93	0.98	0.99
Mug	0.73	0.96	0.99	1.00
Skateboard	0.86	0.97	0.99	1.00
Table	0.76	0.96	0.99	1.00
Vessel	0.45	0.71	0.87	0.94
Mean	0.75	0.93	0.97	0.99
Median	0.78	0.96	0.98	0.99

Figure 3: Qualitative SPair-71k results for MARCO [5]. Left: SPair-71k-only. Right: ours-pretrained → SPair-71k fine-tuned.



4.3 Evaluation of Annotation Quality

We evaluate our generated correspondences on KeypointNet [36], which provides human-annotated semantic keypoints and correspondences on approximately rotation-aligned 3D objects. We assess both **3D–3D** correspondence accuracy against human annotations and **3D–2D** accuracy after projection into image space.

3D–3D evaluation. For each category, we evaluate all model pairs within a subset, retaining pairs where at least 60% of each model's points have a DINOv2 match with cosine similarity above 0.575 (68% of pairs retained on average). Within retained pairs, we prune correspondences with endpoint feature similarity below 0.575. For each ground-truth query keypoint, we select the correspondence with the nearest source point, provided its distance is below 0.05 times the target bounding-box diagonal. Keypoints without a surviving correspondence are excluded from PCK, and their fraction is reflected in the GT-used rate. Predictions are correct if their target locations lie within α times the target bounding-box diagonal of the GT keypoint. We average results over three random subsets of 100 models per category (38 and 90 models for Cap and Helmet, respectively), as shown in Table 3. Our method achieves 93% PCK@0.10, 97% PCK@0.15, and 99% PCK@0.20, with a median GT-used rate of 57.6%. Reduced coverage reflects semantic pruning that avoids uncertain matches between structurally different instances. Detailed per-label results and evaluable keypoint counts appear in Appendix F. On the same pairs, we also outperform Back to 3D [34] in both 1-shot and 3-shot settings without requiring keypoint annotations (see § B.2).

3D–2D evaluation. We further evaluate our generated 3D correspondences in image space by projecting them onto randomly rendered views. As in our data generation, only correspondences whose target points are visible in the rendered image are retained. Across 16 categories, we achieve 94.2% PCK@0.10 in 2D, with an average GT-used rate of 45.3% of ground-truth keypoints in the rendered views. On the same visible correspondences, 3D PCK@0.10 reaches 95.1%, compared to 93% in the original 3D–3D evaluation, which includes all found keypoints regardless of visibility. The slight decrease may result from geometric inaccuracies in the approximate 3D correspondences, which can introduce projection errors or visibility inconsistencies in image space. Nevertheless, the results demonstrate that correspondence accuracy is largely preserved when transferred to 2D. The full evaluation protocol and per-category results are provided in Appendix E.

4.4 Ablations and Hyperparameter Sensitivity

We summarize the main findings of our ablation studies here and defer detailed quantitative analyses, per-category tables, and additional sensitivity experiments to Appendix C and Appendix F.

Hyperparameters. Performance is stable across a reasonable range of the CPA smoothness weight λ_{smooth} and semantic threshold τ_{sem}. As detailed in § C.2, τ_{sem} controls an accuracy–coverage trade-off, and $\tau_{\text{sem}} = 0.575$ provides a practical balance between retaining reliable correspondences

and preserving sufficient coverage. For the CPA smoothness weight, we observe only minor variation in KeypointNet [36] PCK across tested values, likely because the sparse annotations are mostly concentrated on salient corners and extremities. Qualitatively, however, $\lambda_{\text{smooth}} = 0.3$ better preserves the source object structure while still allowing sufficient local deformation to align with the target, producing more coherent dense correspondences in regions not captured by the sparse KeypointNet annotations. We therefore use $\lambda_{\text{smooth}} = 0.3$ and $\tau_{\text{sem}} = 0.575$ in all experiments.

Registration model. We compare continuous piecewise-affine (CPA) registration with a global affine transformation and thin-plate splines (TPS); see § C.1 for details. Affine performs substantially worse at all PCK thresholds, while TPS achieves accuracy closer to CPA. CPA nonetheless achieves comparable or slightly better accuracy at most evaluated thresholds and is approximately $3\times$ faster in our setup, likely because its tetrahedral piecewise-affine parameterization is local, whereas TPS relies on a global radial-basis deformation.

Semantic filtering. Semantic filtering consistently improves correspondence quality (see § C.2). Without correspondence-level or pair-level filtering, geometric registration alone achieves over 84% PCK@0.10. Adding correspondence-level semantic pruning improves performance to about 91% PCK@0.10, while pair-level filtering yields a further improvement to about 93% PCK@0.10 on the retained pairs. We additionally provide a detailed per-semantic-label analysis in Appendix F, following the evaluation protocol of [34], as well as a qualitative analysis of DINOv2-based semantic filtering and correspondence pruning in Appendix D.

Sensitivity to rotation misalignment. We evaluate robustness to imperfect canonical alignment. Mean PCK@0.10 decreases from 93% to 89% and 85% under 5° and 10° rotation perturbations, respectively, but drops substantially for larger rotations, confirming the importance of approximate alignment (see § C.3).

4.5 Discussion

Our results demonstrate that approximate semantic-geometric correspondences generated from rigid 3D objects provide effective supervision for cross-instance matching. KeypointNet evaluations validate our 3D–3D and projected 3D–2D correspondences, while SPair-71k results show improvements in real-image 2D–2D matching, including generalization to non-rigid categories. The scalability of our method lies in leveraging large-scale, approximately aligned 3D collections without manual correspondence annotations. While alignment remains a prerequisite, existing datasets such as Objaverse-OA [20] and advances in orientation estimation [33] and scalable canonicalization [15, 16] make aligned 3D assets increasingly available. Moreover, SemGeo-Gen scales combinatorially, generating correspondences for up to $\binom{N}{2}$ pairs from N aligned objects, with multiple rendered view pairs per pair. Together, these enable scalable semantic-geometric supervision without manual point-level annotations.

4.6 Limitations

Our approach has several limitations. First, the CPA deformation model cannot fully capture large cross-instance structural differences without supervision, potentially leading to imperfect alignments. Second, our semantic filtering relies on DINOv2 descriptors, although our modular pipeline can benefit from stronger future feature extractors. Third, our method assumes rigid, approximately aligned 3D objects and does not directly handle non-rigid pose variations; a promising extension is to cluster objects by pose and apply CPA within each cluster to accommodate residual variations. Finally, our synthetic 3D assets introduce a domain gap when transferring to real images, although pretraining followed by real-data fine-tuning proves effective. Growing collections of real, canonically aligned 3D assets could further reduce this gap.

5 Conclusion

We presented a scalable unsupervised pipeline for generating cross-instance semantic-geometric correspondences from 3D assets without manual annotation. Although approximate, the generated correspondences align well with human annotations and provide an effective pretraining signal for real-image semantic correspondence models. Our results demonstrate that scalable semantic-geometric data can complement manual annotations and improve cross-instance correspondence learning.

Acknowledgments and Disclosure of Funding

Roy Amoyal was partially supported by the Kreitman School of Advanced Graduate Studies. Shira Ifergane was partially supported by the VATAT National Excellence Scholarship for MSc Students in AI and Data Science.

References

[1] Amit Aflalo, Eran Treister, Chaim Baskin, and Oren Freifeld. Factorized gradients for scalable highly-expressive parametric diffeomorphisms. In *Advances in Neural Information Processing Systems*, 2026. 5

[2] Roy Amoyal, Oren Freifeld, and Chaim Baskin. Cross-Instance Gaussian Splatting Registration via Geometry-Aware Feature-Guided Alignment. In *CVPR*, 2026. 1

[3] 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 *European Conference on Computer Vision*, 2024. 1, 3

[4] Seokju Cho, Sunghwan Hong, and Seungryong Kim. CATs++: Boosting cost aggregation with convolutions and transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7174–7194, 2023. doi: 10.1109/TPAMI.2022.3218727. 3, 8

[5] Claudia Cuttano, Gabriele Trivigno, Carlo Masone, and Stefan Roth. MARCO: Navigating the unseen space of semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. 1, 2, 3, 7, 8, 9, 15, 20

[6] R. M. Dyke, C. Stride, Y.-K. Lai, P. L. Rosin, M. Aubry, A. Boyarski, A. M. Bronstein, M. M. Bronstein, D. Cremers, M. Fisher, T. Groueix, D. Guo, V. G. Kim, R. Kimmel, Z. Lähner, K. Li, O. Litany, T. Remez, E. Rodolà, B. C. Russell, Y. Sahillioglu, R. Slossberg, G. K. L. Tam, M. Vestner, Z. Wu, and J. Yang. SHREC'19: Shape correspondence with isometric and non-isometric deformations. In *Eurographics Workshop on 3D Object Retrieval*, pages 111–119. The Eurographics Association, 2019. doi: 10.2312/3dor.20191069. 1, 2

[7] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[8] Oren Freifeld, Søren Hauberg, Kayhan Batmanghelich, and John W. Fisher III. Transformations based on continuous piecewise-affine velocity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2496–2509, 2017. 5

[9] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. 3D-CODED: 3D correspondences by deep deformation. In *ECCV*, 2018. 1, 2

[10] Kamal Gupta, Varun Jampani, Carlos Esteves, Abhinav Shrivastava, Ameesh Makadia, Noah Snavely, and Abhishek Kar. ASIC: Aligning sparse in-the-wild image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 3

[11] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7):1711–1725, 2018. doi: 10.1109/TPAMI.2017.2724510. 2

[12] Eric Hedlin, Gopal Sharma, Shweta Mahajan, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Unsupervised semantic correspondence using stable diffusion. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023. 1, 3

[13] Omri Hirsch, Ron Shapira Weber, Shira Ifergane, and Oren Freifeld. FastJAM: a fast joint alignment model for images. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 1, 3

[14] Shuaiyi Huang, Luyu Yang, Bo He, Songyang Zhang, Xuming He, and Abhinav Shrivastava. Learning semantic correspondence with sparse annotations. In *European Conference on Computer Vision (ECCV)*, 2022. 8

[15] Li Jin, Yujie Wang, Wenzheng Chen, Qiyu Dai, Qingzhe Gao, Xueying Qin, and Baoquan Chen. One-shot 3D object canonicalization based on geometric and semantic consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16850–16859, 2025. 10

[16] Li Jin, Yuchen Yang, Weikai Chen, Yujie Wang, Dehao Hao, Tanghui Jia, Yingda Yin, Zeyu Hu, Runze Zhang, Keyang Luo, Li Yuan, Long Quan, Xin Wang, and Xueying Qin. CanoVerse: 3D object scalable canonicalization and dataset for generation and pose. In *European Conference on Computer Vision (ECCV)*, 2026. 10

[17] Yuanchen Ju, Kaizhe Hu, Guowei Zhang, Gu Zhang, Mingrun Jiang, and Huazhe Xu. Robo-ABC: Affordance generalization beyond categories via semantic correspondence for robot manipulation. In *European Conference on Computer Vision (ECCV)*, pages 222–239. Springer, 2024. 1

[18] Akshay Krishnan, Abhijit Kundu, Kevis-Kokitsi Maninis, James Hays, and Matthew Brown. OmniNOCS: A unified NOCS dataset and model for 3D lifting of 2D objects. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[19] Yujing Lou, Yang You, Chengkun Li, Zhoujun Cheng, Liangwei Li, Lizhuang Ma, Weiming Wang, and Cewu Lu. Human correspondence consensus for 3D object semantic understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1

[20] Yichong Lu, Yuzhuo Tian, Zijin Jiang, Yikun Zhao, Yuanbo Yang, Hao Ouyang, Haoji Hu, Huimin Yu, Yujun Shen, and Yiyi Liao. Orientation matters: Making 3D generative models orientation-aligned. In *Advances in Neural Information Processing Systems*, volume 38, 2025. 7, 10, 17

[21] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: Searching through time and space for semantic correspondence. In *Advances in Neural Information Processing Systems*, 2023. 1, 3, 8

[22] Octave Mariotti, Oisin Mac Aodha, and Hakan Bilen. Improving semantic correspondence with viewpoint-guided spherical maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19521–19530, June 2024. 1, 3, 8

[23] Octave Mariotti, Zhipeng Du, Yash Bhalgat, Oisin Mac Aodha, and Hakan Bilen. Jamais vu: Exposing the generalization gap in supervised semantic correspondence. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 2, 3, 7, 8

[24] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. SPair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint arXiv:1908.10543*, 2019. 1, 2, 3, 7, 15, 16, 17, 20

[25] Dolev Ofri-Amar, Michal Geyer, Yoni Kasten, and Tali Dekel. Neural congealing: Aligning images to a joint semantic atlas. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19403–19412, 2023. 3

[26] 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. *Transactions on Machine Learning Research*, 2024. 4, 5, 7, 8, 20, 22

[27] I. Pratikakis, M. A. Savelonas, F. Arnaoutoglou, G. Ioannakis, A. Koutsoudis, T. Theoharis, M. T. Tran, V. T. Nguyen, V. K. Pham, H. D. Nguyen, H. A. Le, B. H. Tran, Q. H. To, M. B. Truong, T. V. Phan, M. D. Nguyen, T. A. Than, K. N. C. Mac, M. N. Do, A. D. Duong, T. Furuya, R. Ohbuchi, M. Aono, S. Tashiro, D. Pickup, X. Sun, P. L. Rosin, and R. R. Martin. SHREC'16 track: Partial shape queries for 3D object retrieval. In *Eurographics Workshop on 3D Object Retrieval*, pages 79–88. The Eurographics Association, 2016. doi: 10.2312/3dor.20161091. 1, 2

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 8, 20

[29] Ruoxi Shi, Zhengrong Xue, Yang You, and Cewu Lu. Skeleton Merger: An unsupervised aligned keypoint detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 43–52, 2021. 1, 3

[30] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1, 2, 3

[31] Krispin Wandel and Hesheng Wang. SemAlign3D: Semantic correspondence between RGB-images through aligning 3D object-class representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3

[32] Xiaolong Wang. *Learning and Reasoning with Visual Correspondence in Time*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, September 2019. 1

[33] Zehan Wang, Ziang Zhang, Tianyu Pang, Chao Du, Hengshuang Zhao, and Zhou Zhao. Orient anything: Learning robust object orientation estimation from rendering 3D models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025. 10

[34] Thomas Wimmer, Peter Wonka, and Maks Ovsjanikov. Back to 3D: Few-shot 3D keypoint detection with back-projected 2D features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 9, 10, 21

[35] Lumin Xu, Sheng Jin, Wang Zeng, Wentao Liu, Chen Qian, Wanli Ouyang, Ping Luo, and Xiaogang Wang. Pose for everything: Towards category-agnostic pose estimation. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[36] Yang You, Yujing Lou, Chengkun Li, Zhoujun Cheng, Liangwei Li, Lizhuang Ma, Cewu Lu, and Weiming Wang. KeypointNet: A large-scale 3D keypoint dataset aggregated from numerous human annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13647–13656, 2020. 1, 2, 7, 9, 10, 17, 18, 19, 20, 21, 22, 27

[37] Yang You, Wenhai Liu, Yanjie Ze, Yong-Lu Li, Weiming Wang, and Cewu Lu. UKPGAN: A general self-supervised keypoint detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1, 3

[38] Hang Yu, Yufei Xu, Jing Zhang, Wei Zhao, Ziyu Guan, and Dacheng Tao. AP-10K: A benchmark for animal pose estimation in the wild. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 1, 2, 17

[39] 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 *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. 1, 3, 8

[40] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2, 3, 7, 8, 15, 16, 20

[41] Junzhe Zhu, Yuanchen Ju, Junyi Zhang, Muhan Wang, Zhecheng Yuan, Kaizhe Hu, and Huazhe Xu. DenseMatcher: Learning 3D semantic correspondence for category-level manipulation from a single demo. In *International Conference on Learning Representations (ICLR)*, 2025. 1, 2, 3, 8

[42] Silvia Zuffi, Angjoo Kanazawa, David W. Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5524–5532, 2017. doi: 10.1109/CVPR.2017.586. 1, 2

SemGeo-Gen: Unsupervised Generation of Approximate Cross-Instance Semantic-Geometric Correspondences

---

Supplementary Material

In this supplementary material, we provide additional qualitative results, extended ablations, and detailed per-label analysis to complement the main paper.

Contents

1. Appendix A - Qualitative Results on SPair-71k and KeypointNet.
2. Appendix B - Additional Quantitative Results.
3. Appendix C - Extended Ablation Study.
4. Appendix D - Qualitative Analysis of DINOv2 Based Semantic Filtering.
5. Appendix E - 3D–2D Correspondence Evaluation on KeypointNet.
6. Appendix F - Detailed PCK Results by Semantic Label and KeypointNet Category.

A Qualitative Results

A.1 SPair-71k Transfer Results

We provide qualitative SPair-71k [24] examples comparing models trained only on real SPair-71k annotations with the same models pretrained on our generated synthetic correspondences and then fine-tuned on SPair-71k. We show results for both MARCO [5] (see Fig. 4) and GeoAware-SC [40] (see Fig. 5).

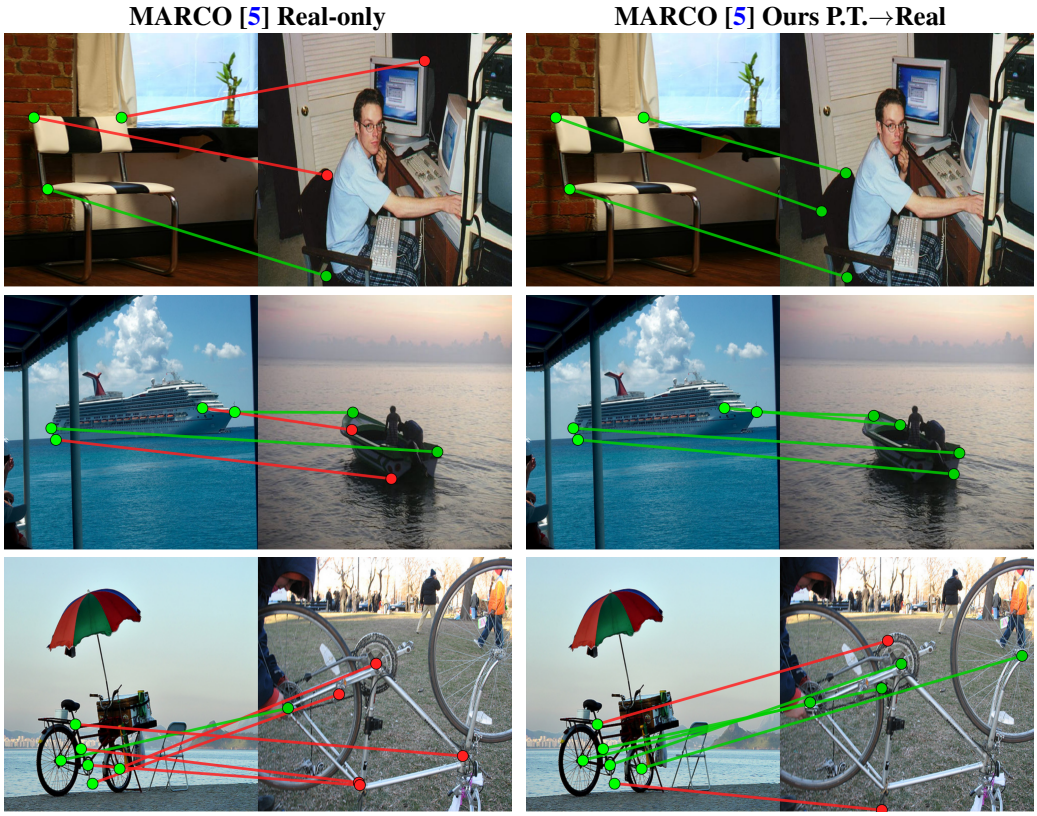


Figure 4: **Qualitative SPair-71k [24] results for MARCO [5].** Each image shows a source–target pair with predicted correspondences. We compare MARCO trained on SPair-71k against MARCO pretrained on our generated correspondences and then fine-tuned on SPair-71k.

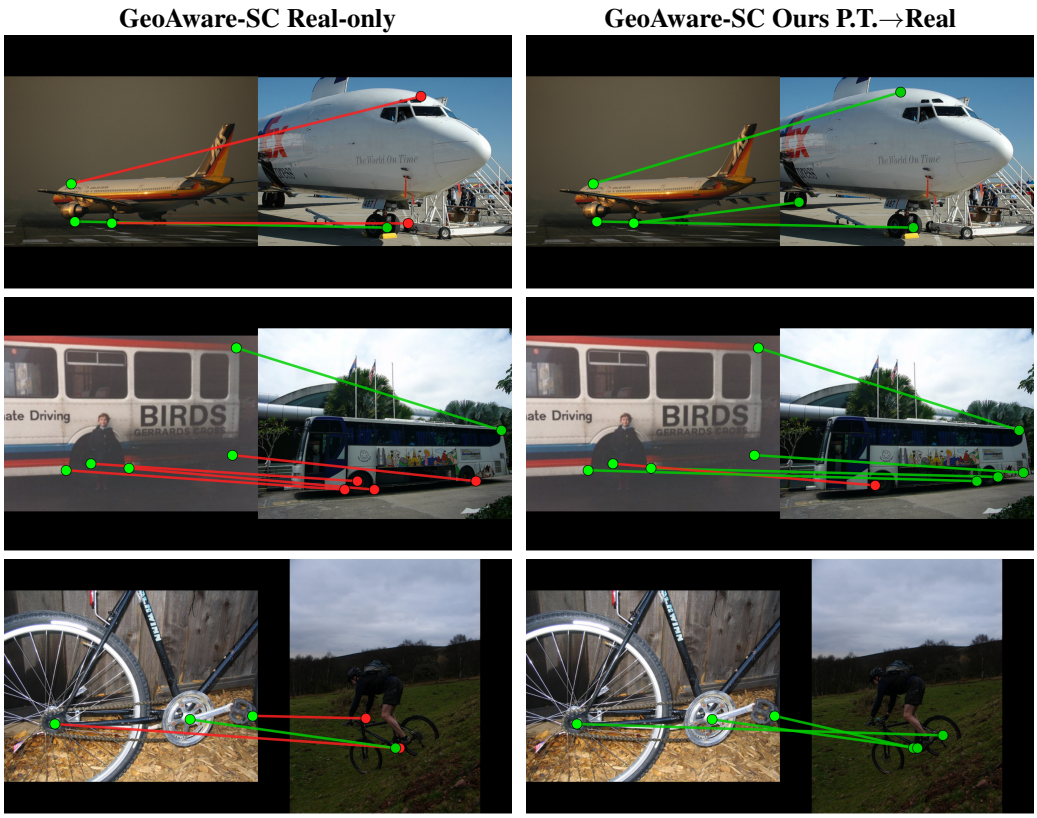


Figure 5: **Qualitative SPair-71k [24] results for GeoAware-SC [40].** Each image shows a source–target pair with predicted correspondences. We compare GeoAware-SC trained on SPair-71k against GeoAware-SC pretrained on our generated correspondences and then fine-tuned on SPair-71k.

A.2 Correspondence Examples on New Categories

We additionally show (Fig. 6) qualitative correspondence examples on rigid categories beyond those covered by existing correspondence benchmarks such as SPair-71k [24], KeypointNet [36], and AP-10K [38]. These examples demonstrate the ability of our method to establish approximate semantic-geometric correspondences on diverse categories from Objaverse-OA [20].

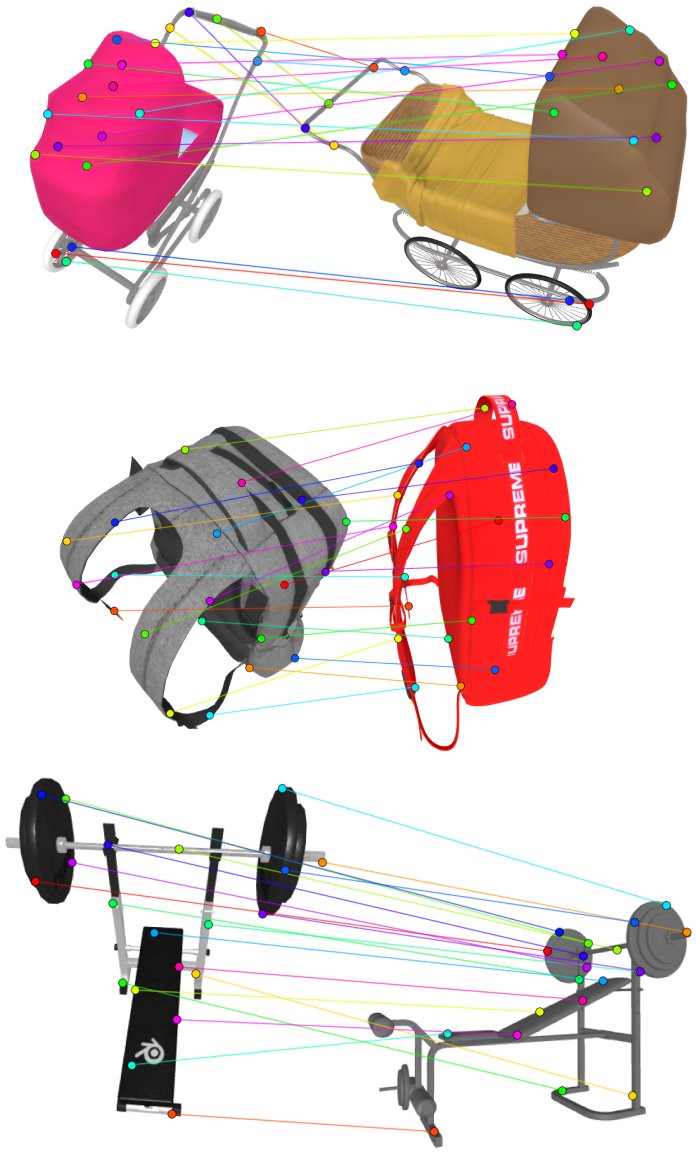
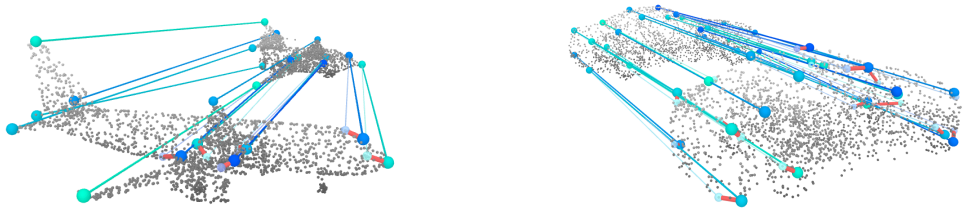


Figure 6: Qualitative correspondence examples on categories beyond existing correspondence benchmarks. Colors indicate sampled semantic-geometric correspondences established by our method.

A.3 Qualitative Results on KeypointNet

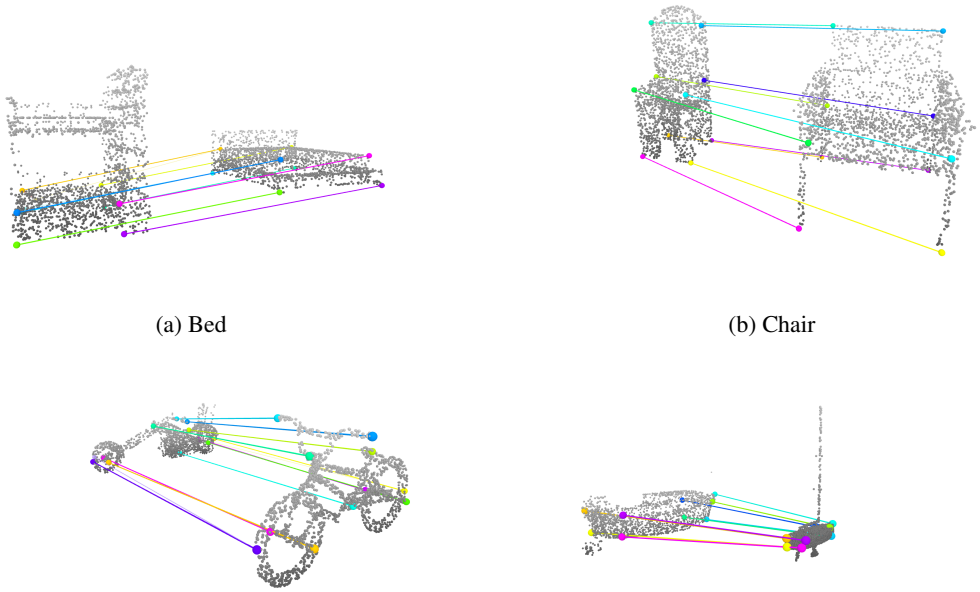
We provide qualitative correspondence results on KeypointNet [36], comparing our predicted correspondences against sparse human-annotated semantic keypoints. For each object pair, we use the human-annotated keypoints on the source object as queries and predict the corresponding points on the target object. As shown in Fig. 7, our predictions generally align well with the human annotations across different categories. Predicted correspondences are shown in lighter colors, while the human annotations are shown in darker colors. Red lines connect each prediction to its corresponding ground-truth annotation.

We further show examples with strong agreement in Fig. 8. The close overlap between our predictions and the human annotations indicates that our generated correspondences are highly consistent with the sparse semantic keypoints used for evaluation.



(a) Airplanes (b) Cars

Figure 7: **Qualitative correspondence results on KeypointNet [36]** We compare our predicted correspondences with human-annotated keypoints on the target object. Predicted correspondences are shown in lighter colors, while human annotations are shown in darker colors. Red lines connect each prediction to its corresponding ground-truth annotation, visualizing the localization difference between them.



(a) Bed

(b) Chair

(c) Motorcycle

(d) Vessel

Figure 8: **KeypointNet [36] examples with strong annotation agreement.** Predicted correspondences are shown in lighter colors, while human annotations are shown in darker colors. The strong overlap between the two indicates that our pipeline recovers correspondences that are highly consistent with human semantic annotations.

B Additional Quantitative Results

We provide additional quantitative evaluations on SPair-71k [24] and KeypointNet [36]. On SPair-71k, we examine different PCK thresholds and training strategies, highlighting the effectiveness of synthetic pretraining across architectures and feature representations. On KeypointNet, we compare our generated correspondences against the few-shot semantic keypoint detector Back to 3D.

B.1 Additional Results on SPair-71k

We provide additional evaluation results on SPair-71k [24] across multiple PCK thresholds and training regimes. While the main paper reports PCK@0.10 (see Table 1), here we analyze performance at stricter and looser thresholds, and compare four training strategies: *Synth-only*, *Both* (joint real and synthetic training), *Real-only*, and *Synth→Real* (pretraining on synthetic correspondences followed by fine-tuning on real data).

MARCO [5] results. Across most thresholds (PCK@0.05–0.20), we observed that the Synth→Real setting consistently outperforms Real-only training at practical thresholds. Improvements of approximately $+2.4$ to $+3.0$ PCK are observed for all PCK@0.05–0.20, and specifically at PCK@0.10, performance improves from 86.34 to 89.30 ($+2.96$), matching the results reported in Table 1.

However, at a very strict threshold (PCK@0.01), Synth→Real slightly underperforms Real-only (-2.17). However, this gap disappears at more relaxed thresholds, where Synth→Real consistently achieves the best performance.

We also observe that joint training (*Both*) underperforms Real-only, indicating that naive mixing of synthetic and real data is suboptimal, and that a two-stage training strategy is more effective.

GeoAware-SC [40] results. We observe the similiar qualitative trend for GeoAware-SC. Synth→Real consistently improves over Real-only training across all evaluated thresholds (including improvement at the very strict threshold of PCK@0.01), while Synth-only performs poorly in isolation. Joint training slightly outperforms Real-only, but remains inferior to synthetic pretraining followed by real-data fine-tuning, reinforcing that synthetic correspondences are most effective as a pretraining signal rather than as direct joint supervision.

At PCK@0.10, as we see in the results from the main paper (Table 1), synthetic pretraining improves performance from 79.36 to 83.73, corresponding to a gain of $+4.37$ PCK.

Cross-architecture consistency. Using the PCK@0.10 results from Table 1, we observe consistent improvements across architectures and feature choices:

Architecture	DINOv2 [26]	Stable Diffusion [28]	Real-only	Synth→Real	Gain
GeoAware-SC [40]	✓	✓	79.36	83.73	+4.37
MARCO [5]	✓	×	86.34	89.30	+2.96

GeoAware-SC [40] uses both DINOv2 [26] and Stable Diffusion [28] features, while MARCO [5] relies on DINOv2 features without Stable Diffusion. The gains across both settings suggest that our synthetic pretraining signal is not tied to a specific feature family, but provides useful semantic-geometric supervision across different recent correspondence models.

These consistent gains suggest that the benefit of synthetic pretraining is largely architecture-agnostic. Rather than improving a specific model design, the generated correspondences provide a strong pretraining signal that improves representation quality across different backbones and matching heads.

Summary. Overall, these results support three key observations: (i) synthetic correspondences alone are insufficient for high performance, (ii) naive joint training is suboptimal, and (iii) a two-stage Synth→Real strategy consistently yields the best results. This pattern holds across both architectures at practical evaluation thresholds, reinforcing the generality of the proposed data generation pipeline.

B.2 Comparison with Back to 3D on KeypointNet

We additionally compare against Back to 3D [34], a few-shot semantic keypoint detection method rather than a cross-instance correspondence method. Unlike SemGeo-Gen, it predicts predefined semantic keypoints using annotated source examples instead of generating dense correspondences without keypoint supervision. We evaluate its 1-shot and 3-shot settings on 100 object pairs per category (1,600 total). The annotated source keypoints define semantic template slots, and source-to-target associations are established through their shared KeypointNet labels. Predicted target locations are then evaluated using the same PCK thresholds as our correspondences.

As shown in Table 4, SemGeo-Gen achieves 86.5% PCK@0.10, outperforming Back to 3D's 1-shot (63.2%) and 3-shot (64.1%) settings, despite requiring no keypoint supervision.

Table 4: **Comparison with Back to 3D [34] on KeypointNet [36]** Mean PCK over 16 categories (100 pairs per category). Back to 3D uses few-shot keypoint supervision; SemGeo-Gen requires none.

Method	@0.05	@0.10	@0.15	@0.20
Back to 3D (1-shot)	0.438	0.632	0.725	0.777
Back to 3D (3-shot)	0.436	0.641	0.740	0.799
SemGeo-Gen	**0.648**	**0.865**	**0.945**	**0.977**

C Extended Ablation Study

We provide additional ablation studies examining the contributions of the registration model and semantic filtering, as well as the sensitivity of our method to imperfect canonical alignment. These experiments further characterize the accuracy, efficiency, and robustness of our method.

C.1 Registration Method Ablation

We ablate the deformation model used in the registration stage prior to correspondence extraction. All other components are kept fixed, including lifted DINOv2 [26] features, nearest-neighbor assignment, and the semantic threshold $\tau_{\text{sem}} = 0.575$. We compare our default continuous piecewise-affine (CPA) model with two alternatives: a global affine transformation and a thin-plate spline (TPS) deformation.

Affine. As a global transformation baseline, we use an affine transformation

$$f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{t},$$

where $\mathbf{A} \in \mathbb{R}^{3\times3}$ and $\mathbf{t} \in \mathbb{R}^3$. This model has 12 learnable parameters and is optimized with Adam using the same Chamfer objective. Since it cannot model non-rigid variation, it serves as a baseline for how much alignment can be explained by global pose normalization alone.

Continuous Piecewise-Affine (CPA). Our default model uses a $6 \times 6 \times 6$ control grid (216 vertices) over the normalized 3D domain, tetrahedralized via Delaunay triangulation. Each vertex has a learnable displacement, and each point is warped using barycentric interpolation within its enclosing tetrahedron. This yields a deformation that is locally affine and globally continuous. We enforce smoothness by penalizing squared differences between neighboring vertex displacements, with $\lambda_{\text{smooth}} = 0.3$. The model has 648 degrees of freedom.

Thin-Plate Spline (TPS). We compare with a classical smooth non-rigid deformation

$$f(\mathbf{x}) = \mathbf{a} + \mathbf{B}\mathbf{x} + \sum_{k=1}^{K} \mathbf{w}_k \, U(\|\mathbf{x} - \mathbf{c}_k\|), \qquad U(r) = -r,$$

where $\mathbf{B} \in \mathbb{R}^{3\times3}$, $\mathbf{a} \in \mathbb{R}^3$, and $\mathbf{w}_k \in \mathbb{R}^3$. We use $K = 216$ control points sampled from the source, resulting in a similar number of parameters to CPA (660 vs. 648). Smoothness is enforced with an ℓ_2 penalty on the radial weights.

Protocol. We evaluate all three methods on the 16 KeypointNet [36] categories. For each category, we sample 100 source-target pairs from random instances and run all methods on them. All hyperparameters are shared: 10,000 Adam iterations with early stopping on a loss plateau, a learning rate of 0.005, and no pair-level filtering. Performance is measured using PCK at multiple thresholds, where a prediction is correct if it falls within $\tau \cdot d$ of the ground-truth target keypoint, with d the target bounding-box diagonal.

Results. Table 5 reports mean PCK across all categories. CPA achieves the best average performance at four of the five PCK thresholds, while TPS is slightly better only at PCK@0.05. Overall, CPA and TPS obtain very similar accuracy, indicating that the choice of non-rigid basis has limited impact on correspondence quality.

In contrast, CPA is substantially more efficient. It is about $3.3\times$ faster per pair (1.13s vs. 3.71s), despite having a similar number of parameters. We also observe that CPA typically converges earlier. This difference arises from the underlying structure: CPA uses local barycentric interpolation within tetrahedra, while TPS requires dense kernel interactions between all points and control points at each iteration.

Compared with the affine baseline, CPA provides clear gains, especially at tighter PCK thresholds, highlighting the importance of non-rigid alignment for cross-instance correspondence.

Table 5: **Registration ablation on KeypointNet.** Mean PCK across all 16 categories. PCK thresholds define the allowed normalized distance from the ground-truth keypoint, measured as a fraction of the target bounding-box diagonal. DOF denotes the number of learnable parameters. Time reports the mean registration time per pair on a single A6000 GPU. Best results are shown in **bold**.

Method	DOF	Time (s/pair)	PCK@0.01	PCK@0.05	PCK@0.10	PCK@0.15	PCK@0.20
Affine	12	1.15	0.221	0.686	0.898	0.960	0.982
TPS ($K=216$)	660	3.71	0.226	**0.739**	0.919	0.966	0.983
CPA (ours)	648	**1.13**	**0.236**	0.729	**0.922**	**0.971**	**0.986**

Conclusion. This ablation supports two conclusions: (i) non-rigid deformation is essential for accurate alignment, as affine transformations perform significantly worse; and (ii) among non-rigid models, CPA offers a better accuracy-efficiency tradeoff, achieving similar performance to TPS while being significantly faster and more stable in practice.

C.2 Ablation on Semantic Filtering

We analyze the role of semantic filtering in our pipeline. Specifically, we study (i) the effect of removing semantic filtering entirely, (ii) the contribution of correspondence-level filtering alone, and (iii) the impact of the correspondence-level semantic threshold τ_{sem} when combined with pair-level filtering. All experiments are conducted on KeypointNet using the same evaluation protocol as in the main paper.

C.2.1 No Semantic Filtering

We first evaluate the pipeline without any semantic filtering. In this setting, all object pairs are retained, and correspondences are obtained directly from CPA registration followed by nearest-neighbor assignment.

Table 6 shows that this setting produces significantly lower accuracy. While geometric alignment alone can recover many plausible correspondences, it frequently matches inconsistent regions across instances, leading to degraded PCK, especially at stricter thresholds (e.g., PCK@0.05 and PCK@0.10). This highlights the limitation of relying solely on geometry for cross-instance correspondence and the need for a semantic pruning step.

C.2.2 Correspondence-Level Semantic Filtering Only

We next evaluate correspondence-level semantic filtering alone, without pair-level filtering. In this setting, all object pairs are retained, and individual correspondences are pruned using a fixed threshold $\tau_{\text{sem}} = 0.575$.

Table 7 shows that this step alone leads to a substantial improvement over the no-filtering baseline. In particular, mean PCK@0.10 improves from 0.84 to 0.91, confirming that a large portion of the error arises from semantically inconsistent matches that are geometrically plausible but incorrect.

C.2.3 Correspondence-Level Semantic Threshold

We finally analyze the effect of the semantic threshold τ_{sem} when combined with pair-level filtering, as used in the full pipeline.

This threshold controls an accuracy-coverage tradeoff. A stricter threshold keeps fewer but more reliable matches, while a looser threshold retains more matches but allows more semantically inconsistent correspondences. To quantify coverage, we report the *median GT-used rate* across categories. A ground-truth keypoint is counted as used if at least one surviving correspondence remains within 0.05 of the object bounding-box diagonal from the ground-truth location.

Importantly, this metric is conservative. KeypointNet annotations are sparse and typically located on salient corners or extreme parts, while our method produces dense correspondences across the object surface (see Fig. 1). As a result, semantically valid correspondences that do not fall within the 0.05 evaluation radius are still counted as missed.

Table 6: Evaluation of annotation quality on KeypointNet correspondences without filters. We report PCK at normalized distance thresholds of 5%, 10%, 15%, and 20% of the object scale, computed per category. (on 100 objects)

Category	PCK@0.05 ↑	PCK@0.10 ↑	PCK@0.15 ↑	PCK@0.20 ↑
Airplane	0.73	0.90	0.95	0.98
Bathtub	0.41	0.73	0.91	0.97
Bed	0.45	0.71	0.84	0.90
Bottle	0.65	0.83	0.92	0.98
Car	0.75	0.94	0.97	0.99
Chair	0.51	0.83	0.93	0.97
Guitar	0.78	0.94	0.98	0.99
Laptop	0.87	0.99	1.00	1.00
Motorcycle	0.69	0.88	0.95	0.99
Mug	0.70	0.95	0.99	1.00
Skateboard	0.70	0.87	0.96	0.99
Table	0.63	0.86	0.94	0.97
Vessel	0.38	0.66	0.82	0.89
Knife	0.52	0.68	0.82	0.91
Helmet	0.36	0.69	0.86	0.93
Cap	0.74	0.92	0.99	1.0
Mean	0.62	0.84	0.93	0.97
Median	0.67	0.86	0.94	0.98

Table 7: Ablation of only semantic filtering of correspondences, reporting PCK results per category. calculated over 100 models.

Category	PCK@0.05	PCK@0.10	PCK@0.15	PCK@0.20
Airplane	0.78	0.94	0.98	0.99
Bathtub	0.68	0.86	0.94	0.97
Bed	0.63	0.85	0.94	0.98
Bottle	0.80	0.95	0.97	0.99
Car	0.77	0.95	0.98	1.00
Chair	0.71	0.94	0.98	0.99
Guitar	0.80	0.96	0.98	0.99
Laptop	0.89	0.99	1.00	1.00
Table	0.74	0.95	0.99	1.00
Mug	0.70	0.95	0.99	1.00
Skateboard	0.86	0.97	0.99	1.00
Vessel	0.42	0.68	0.84	0.93
Motorcycle	0.75	0.93	0.98	0.99
Cap	0.80	0.96	1.00	**1.00**
Knife	0.86	0.93	0.97	0.98
Helmet	0.46	0.77	0.91	0.95
Mean	0.73	0.91	0.97	0.99
Median	0.76	0.94	0.98	0.99

Table 8 shows that the choice of threshold controls a clear tradeoff between accuracy and coverage. A strict threshold ($\tau_{\text{sem}} = 0.75$) achieves the highest accuracy at tight thresholds (75.8% PCK@0.05) but retains only 26.0% of ground-truth keypoints, while a loose threshold ($\tau_{\text{sem}} = 0.25$) retains 91.5% but degrades accuracy (69.9% PCK@0.05, 89.6% PCK@0.10). Our default ($\tau_{\text{sem}} = 0.575$) retains 57.6% of the keypoints with almost no loss in accuracy compared to the strict setting.

We therefore use $\tau_{\text{sem}} = 0.575$ as a practical middle point: it maintains similar accuracy to the stricter threshold while more than doubling the median GT-used rate (26.0% $\rightarrow$ 57.6%).

Table 8: **Ablation on the correspondence-level semantic threshold (with pair-level filtering).** Results are averaged over three random seeds and 16 KeypointNet categories. Median GT-used rate denotes the median fraction across categories of ground-truth keypoints that remain evaluable after correspondence-level pruning.

τ_{sem}	PCK@0.05 ↑	PCK@0.10 ↑	PCK@0.15 ↑	PCK@0.20 ↑	Median GT-used rate ↑
0.75	0.758	0.930	0.974	0.988	0.260
0.575	0.749	0.926	0.973	0.989	0.576
0.25	0.699	0.896	0.960	0.983	0.915

Discussion. Overall, these results show that semantic filtering is essential for removing geometrically plausible but semantically incorrect correspondences. Most of the improvement comes from correspondence-level filtering, while pair-level filtering provides an additional refinement by removing object pairs with insufficient semantic overlap.

C.3 Sensitivity to Rotation Misalignment

We evaluate the robustness of SemGeo-Gen to imperfect canonical alignment. For each of the 16 KeypointNet categories, we sample 100 object pairs and randomly select either the source or target object to perturb with a random-axis rotation of $0°$, $5°$, $10°$, $30°$, $60°$, or $90°$. We measure correspondence accuracy using the same KeypointNet evaluation protocol.

As shown in Table 9, mean PCK@0.10 decreases from 0.93 for aligned objects to 0.89 and 0.85 at $5°$ and $10°$, respectively. Performance drops substantially at $30°$ and beyond, confirming that approximate canonical alignment remains an important prerequisite for accurate correspondence generation.

Table 9: **Sensitivity to rotation misalignment.** Mean and median KeypointNet PCK under random-axis rotations applied to either the source or target.

PCK	Stat.	$0°$	$5°$	$10°$	$30°$	$60°$	$90°$
@0.05	Mean	.75	.71	.65	.32	.10	.04
	Median	.78	.75	.69	.25	.06	.02
@0.10	Mean	.93	.89	.85	.55	.20	.08
	Median	.94	.91	.88	.52	.13	.05
@0.15	Mean	.98	.94	.90	.66	.26	.12
	Median	.98	.95	.91	.64	.20	.07
@0.20	Mean	.99	.95	.91	.70	.30	.14
	Median	.99	.95	.91	.67	.26	.09

D Qualitative Analysis of DINOv2 Based Semantic Filtering

We provide a qualitative analysis of the DINOv2 based semantic filtering used in our pipeline, both at the pair level and at the correspondence level. As shown in Fig. 9, we visualize the lifted DINOv2 features after joint PCA to three dimensions, and mark points according to whether they have any sufficiently similar semantic counterpart in the paired object. Green points have at least one point in the paired object with cosine similarity above our threshold of 0.575, while red points have no such candidate. We use this semantic coverage to filter object pairs that do not share enough semantic structure. In addition, for pairs that survive pair-level filtering, red points are guaranteed to be removed by our semantic pruning after registration. This filtering is complementary to the correspondence-level pruning, which further removes geometrically matched points whose assigned matches are semantically inconsistent.

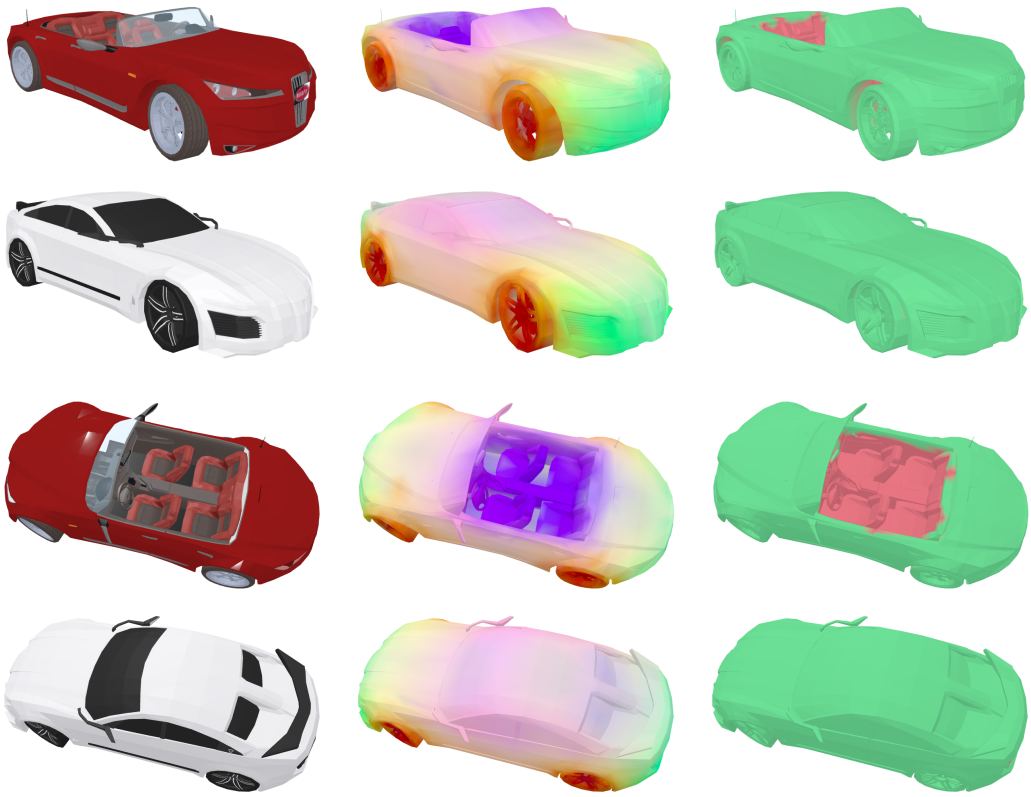


Figure 9: Qualitative analysis of DINOv2 based semantic coverage. Each row shows the original rendering, a PCA visualization of DINOv2 features computed jointly for the object pair, and the resulting semantic coverage mask. Green points have at least one semantically similar point in the paired object, while red points have none.

We further visualize sampled correspondences established by our method in Fig. 10. For visualization, we sample farthest points from the final correspondence set after semantic pruning. Each row shows a different view of the same object pair. As can be seen, semantically unmatched regions such as the exposed seats and steering wheel of the roofless car do not obtain correspondences. Similarly, the roof region of the white car is also removed, even though the CPA registration geometrically deforms these regions toward other object parts. These correspondences are filtered by our semantic correspondence pruning stage due to semantic inconsistency.

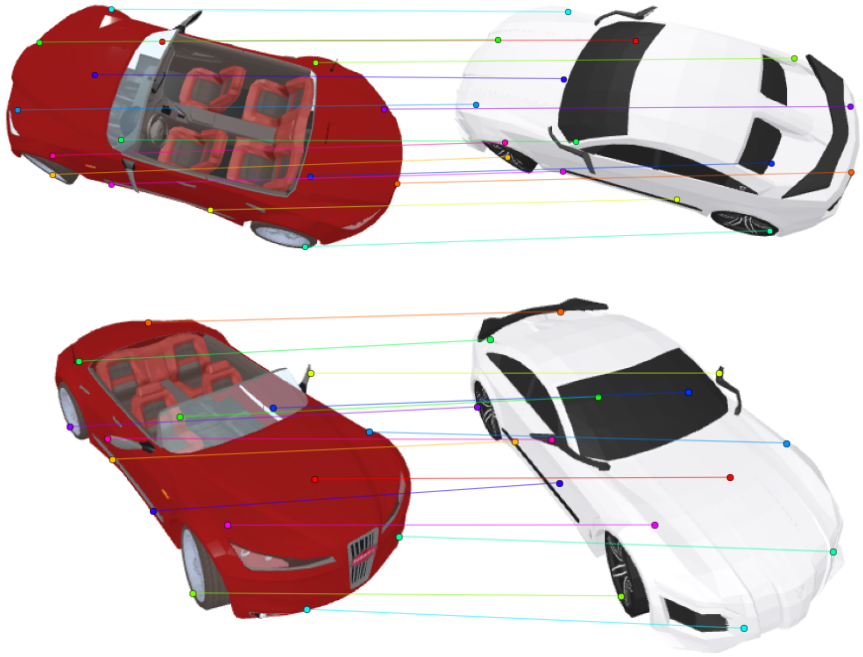


Figure 10: Sampled correspondences after semantic pruning. Each row shows a different view of the same object pair.

E 3D–2D Correspondence Evaluation on KeypointNet

We further evaluate the quality of the 3D–2D correspondences generated by SemGeo-Gen using KeypointNet [36]. Since our 3D–3D correspondences are approximate, geometric inaccuracies may introduce projection errors or visibility inconsistencies when transferred to image space. To assess their impact, we project the generated correspondences onto rendered target views and evaluate their accuracy against human-annotated semantic keypoints, accounting for visibility and occlusion.

Evaluation protocol. We follow the 3D–3D evaluation protocol described in the main paper, using the same 16 categories, model subsets, pair filtering, and generated correspondences. Each target model is rendered from four random viewpoints, with azimuth sampled from $[0°, 360°)$ and elevation from $[-10°, 70°]$.

We evaluate only ground-truth keypoints visible in the rendered target image, since occluded keypoints cannot provide valid image-space annotations. As in our data-generation pipeline, predicted correspondences are retained only if their target points are also visible. Visibility is determined by comparing each point's depth with the rendered depth map using a tolerance of 0.02 to account for small rasterization and surface discrepancies.

For each visible ground-truth keypoint, we select the correspondence with the nearest source point among those with visible targets, provided its distance is within 0.05 times the target 3D bounding-box diagonal. Keypoints without such a correspondence are considered not found.

Evaluation metrics. We report PCK on found keypoints. A prediction is considered correct at threshold α if

$$\|\mathbf{u}_{pred} - \mathbf{u}_{gt}\|_2 < \alpha d_{2D}, \tag{17}$$

where $\mathbf{u}_{pred}$ and $\mathbf{u}_{gt}$ denote the projected predicted and ground-truth target locations, and d_{2D} is the projected 2D bounding-box diagonal of the target object.

We additionally report coverage, defined as the fraction of visible ground-truth keypoints for which a valid correspondence is found (similar to the 3D-3D experiment). This measures coverage of the

Table 10: **3D–2D correspondence evaluation on KeypointNet.** PCK on found correspondences with visible ground-truth keypoints and visible predicted target points. Coverage denotes the fraction of visible ground-truth keypoints for which a valid correspondence is found. The last column reports the original 3D PCK@0.10 over all found keypoints.

Category	PCK@.05	PCK@.10	PCK@.15	PCK@.20	Coverage	3D PCK@.10
Airplane	0.79	0.94	0.98	0.99	0.73	0.94
Bathtub	0.81	0.95	0.99	1.00	0.26	0.88
Bed	0.81	0.96	0.99	1.00	0.45	0.91
Bottle	0.81	0.95	0.98	0.99	0.40	0.96
Cap	0.83	0.97	1.00	1.00	0.30	0.97
Car	0.76	0.95	0.98	1.00	0.88	0.95
Chair	0.80	0.97	0.99	1.00	0.31	0.96
Guitar	0.81	0.97	0.99	1.00	0.61	0.96
Helmet	0.57	0.85	0.96	0.99	0.20	0.84
Knife	0.79	0.92	0.96	0.98	0.32	0.93
Laptop	0.93	1.00	1.00	1.00	0.55	0.99
Motorcycle	0.70	0.90	0.97	0.99	0.74	0.93
Mug	0.79	0.97	0.99	1.00	0.59	0.96
Skateboard	0.84	0.97	0.99	1.00	0.42	0.97
Table	0.87	0.98	1.00	1.00	0.40	0.96
Vessel	0.70	0.86	0.93	0.97	0.08	0.71
Mean	**0.79**	**0.94**	**0.98**	**0.99**	**0.45**	**0.93**

sparse KeypointNet annotations rather than the fraction of dense correspondences retained by our data-generation pipeline.

Results. As shown in Table 10, SemGeo-Gen achieves 94.2% PCK@0.10 across 16 categories, compared to 92.7% in the original 3D–3D evaluation over all found keypoints. The higher overall 2D accuracy reflects the restriction to visible keypoints, as required for valid 3D–2D correspondences. When evaluated on the same retained keypoints, 3D PCK reaches 95.1%, compared to 94.2% in 2D. The slight decrease may arise from geometric inaccuracies in approximate correspondences, which can affect their projected locations and normalized image-space errors. Nevertheless, the small gap indicates that correspondence accuracy is largely preserved after projection into image space.

The average coverage is 45.3% of visible ground-truth keypoints, reflecting the combined effect of correspondence availability and visibility filtering. Results remain stable across independently sampled viewpoint sets, varying by at most ± 0.1 percentage points.

F Detailed PCK Results by Semantic Label and KeypointNet Category

We provide a detailed analysis of correspondence quality at the level of individual semantic keypoints. For each category, we evaluate one subset of up to 100 objects (38 for Cap and 90 for Helmet) and all induced source-target pairs, corresponding to up to $\binom{100}{2}$ object pairs per category, across 15 categories and 69,058 pairs overall, of which 46,849 pass the pair-level filter.

These results are computed using our full pipeline, including correspondence-level semantic filtering with $\tau_{\text{sem}} = 0.575$ and pair-level filtering. For each semantic label, we report results as correct/total, where *total* denotes the number of ground-truth keypoints that remain evaluable after filtering. A keypoint is considered evaluable if at least one predicted correspondence remains within 0.05 of the object bounding-box diagonal from the ground-truth location. We report raw correct/total counts rather than percentages to make the amount of retained evaluable annotations explicit for each semantic label.

KeypointNet annotations are sparse and typically located on salient corners or extreme parts, while our method produces dense correspondences across the object surface. Therefore, these per-label results should be interpreted as a conservative estimate of correspondence quality around the sparse annotated locations.

The full per-label results are reported in Tables 11-25.

Table 11: Per-keypoint PCK results for category `airplane`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	nose of the airplane	3805/3834	3829/3834	3834/3834
1	cockpit of the airplane	2768/3289	3113/3289	3205/3289
2	tail of the airplane	3059/3457	3437/3457	3455/3457
3	wingtip of the left wing of the airplane	3597/3790	3707/3790	3716/3790
4	wing root of the left wing of the airplane	3046/3192	3181/3192	3192/3192
5	wingtip of the right wing of the airplane	3606/3795	3709/3795	3719/3795
6	wing root of the right wing of the airplane	3042/3195	3182/3195	3193/3195
7	wingtip of the left horizontal stabilizer of the airplane	3569/3711	3615/3711	3682/3711
8	wingtip of the right horizontal stabilizer of the airplane	3487/3719	3534/3719	3683/3719
9	top of vertical stabilizer of the airplane	3146/3369	3307/3369	3368/3369
10	wingtip of the left diagonal stabilizer of the airplane	34/34	34/34	34/34
11	wingtip of the right diagonal stabilizer of the airplane	38/38	38/38	38/38
12	wing root of the left wing of the airplane	2923/3095	3063/3095	3090/3095
13	wing root of the right wing of the airplane	2920/3093	3056/3093	3091/3093
14	inner turbine of the left wing of the airplane	1402/1422	1421/1422	1422/1422
15	inner turbine of the right wing of the airplane	1400/1424	1423/1424	1424/1424
16	outer turbine of the left wing of the airplane	105/130	130/130	130/130
17	outer turbine of the right wing of the airplane	117/138	138/138	138/138
18	outer turbine of the right wing of the airplane	0/0	0/0	0/0
	Total	42064/44725	43917/44725	44414/44725

Table 12: Per-keypoint PCK results for category `bathtub`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	top outer corner of the bathtub	1340/1350	1346/1350	1350/1350
1	top outer corner of the bathtub	1300/1314	1310/1314	1314/1314
2	top outer corner of the bathtub	1277/1287	1285/1287	1287/1287
3	top outer corner of the bathtub	1234/1250	1248/1250	1250/1250
4	bottom outer corner of the bathtub	566/649	595/649	616/649
5	bottom outer corner of the bathtub	478/558	511/558	530/558
6	bottom outer corner of the bathtub	471/588	526/588	550/588
7	bottom outer corner of the bathtub	486/571	530/571	548/571
8	top inner corner of the bathtub	775/991	925/991	975/991
9	top inner corner of the bathtub	766/986	908/986	957/986
10	top inner corner of the bathtub	731/938	878/938	933/938
11	top inner corner of the bathtub	696/899	848/899	893/899
12	bottom inner corner of the bathtub	552/679	635/679	656/679
13	bottom inner corner of the bathtub	562/733	666/733	701/733
14	bottom inner corner of the bathtub	584/698	651/698	686/698
15	bottom inner corner of the bathtub	466/545	508/545	534/545
16	center of the short side upper edge of the bathtub	49/49	49/49	49/49
17	center of the short side upper edge of the bathtub	55/56	56/56	56/56
18	center of the short side lower edge of the bathtub	24/27	27/27	27/27
19	center of the short side lower edge of the bathtub	21/24	24/24	24/24
20	center of the short side upper edge of the bathtub	159/165	162/165	164/165
21	center of the short side upper edge of the bathtub	141/144	143/144	144/144
22	center of the short side lower edge of the bathtub	80/93	92/93	93/93
23	center of the short side lower edge of the bathtub	66/69	69/69	69/69
	Total	12879/14663	13992/14663	14406/14663

Table 13: Per-keypoint PCK results for category `bed`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	lower end of the front leg of the bed	989/1044	1032/1044	1042/1044
1	lower end of the headboard leg of the bed	880/965	924/965	960/965
2	lower end of the headboard leg of the bed	825/906	857/906	901/906
3	lower end of the front leg of the bed	937/988	979/988	984/988
4	foot corner of the mattress	606/663	640/663	648/663
5	head corner of the mattress	420/511	464/511	491/511
6	head corner of the mattress	416/492	461/492	475/492
7	foot corner of the mattress	521/566	550/566	553/566
8	foot corner of the upper mattress of the bunk bed	489/521	516/521	516/521
9	head corner of the upper mattress of the bunk bed	464/507	488/507	496/507
10	head corner of the upper mattress of the bunk bed	493/545	522/545	527/545
11	foot corner of the upper mattress of the bunk bed	480/510	507/510	510/510
12	upper corner of the bed frame of the bunk bed	567/568	568/568	568/568
13	upper corner of the bed frame of the bunk bed	515/553	525/553	552/553
14	upper corner of the bed frame of the bunk bed	527/556	537/556	555/556
15	upper corner of the bed frame of the bunk bed	548/554	554/554	554/554
16	upper corner of the footboard of the bed	0/0	0/0	0/0
17	upper corner of the headboard of the bed	79/84	83/84	84/84
18	upper corner of the headboard of the bed	86/87	87/87	87/87
19	upper corner of the footboard of the bed	0/0	0/0	0/0
	Total	9842/10620	10294/10620	10503/10620

Table 14: Per-keypoint PCK results for category `bottle`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	lower rim of the bottle	2836/2866	2860/2866	2864/2866
1	lower rim of the bottle	2845/2868	2866/2868	2866/2868
2	lower rim of the bottle	2693/2711	2707/2711	2711/2711
3	lower rim of the bottle	2829/2861	2857/2861	2860/2861
4	edge between the bottle body and the bottle neck	243/261	259/261	261/261
5	edge between the bottle body and the bottle neck	257/275	272/275	275/275
6	edge between the bottle body and the bottle neck	250/268	267/268	268/268
7	edge between the bottle body and the bottle neck	256/273	272/273	272/273
8	bottle cap center	1438/1453	1453/1453	1453/1453
9	edge of the bottle cap	2096/2135	2115/2135	2134/2135
10	edge of the bottle cap	2095/2135	2116/2135	2134/2135
11	edge of the bottle cap	2088/2133	2112/2133	2132/2133
12	edge of the bottle cap	2083/2123	2104/2123	2123/2123
13	lower edge of the bottle neck	856/1005	923/1005	968/1005
14	lower edge of the bottle neck	824/970	896/970	933/970
15	lower edge of the bottle neck	805/961	877/961	921/961
16	lower edge of the bottle neck	798/958	880/958	922/958
17	lower edge of the bottle neck	0/0	0/0	0/0
	Total	25292/26256	25836/26256	26097/26256

Table 15: Per-keypoint PCK results for category `cap`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	top button of the cap	528/558	557/558	558/558
1	front of the visor of the cap	558/558	558/558	558/558
2	corner of the visor of the cap	481/558	546/558	554/558
3	corner of the visor of the cap	484/558	540/558	555/558
4	center of the back of the cap	555/558	558/558	558/558
5	center of the visor attachment of the cap	493/502	502/502	502/502
	Total	3099/3292	3261/3292	3285/3292

Table 16: Per-keypoint PCK results for category `car`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	front right wheel of the car	3118/3118	3118/3118	3118/3118
1	rear right wheel of the car	3097/3100	3100/3100	3100/3100
2	rear left wheel of the car	3102/3113	3113/3113	3113/3113
3	front left wheel of the car	3107/3111	3111/3111	3111/3111
4	corner of the front bumper of the car	2890/2909	2909/2909	2909/2909
5	corner of the rear bumper of the car	3046/3067	3067/3067	3067/3067
6	corner of the rear bumper of the car	3068/3084	3084/3084	3084/3084
7	corner of the front bumper of the car	2919/2935	2935/2935	2935/2935
8	top corner of the windshield of the car	3044/3116	3112/3116	3115/3116
9	top corner of the windshield of the car	1919/2631	2283/2631	2521/2631
10	top corner of the rear window of the car	1976/2644	2360/2644	2558/2644
11	top corner of the rear window of the car	3053/3118	3112/3118	3118/3118
12	bottom corner of the windshield of the car	2817/3068	3032/3068	3064/3068
13	bottom corner of the windshield of the car	2892/3055	3032/3055	3053/3055
14	bottom corner of the rear window of the car	1830/2171	2037/2171	2117/2171
15	bottom corner of the rear window of the car	1821/2145	2049/2145	2119/2145
16	right headlight of the car	2857/2903	2902/2903	2903/2903
17	right taillight of the car	2907/2919	2919/2919	2919/2919
18	left taillight of the car	2906/2916	2916/2916	2916/2916
19	left headlight of the car	2849/2905	2904/2905	2905/2905
20	right side mirror of the car	2147/2231	2225/2231	2231/2231
21	left side mirror of the car	2158/2245	2236/2245	2245/2245
	Total	59523/62504	61556/62504	62221/62504

Table 17: Per-keypoint PCK results for category `chair`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	upper corner of the back of the chair	743/779	771/779	778/779
1	upper corner of the back of the chair	713/759	757/759	758/759
2	lower corner of the back of the chair	501/548	537/548	547/548
3	lower corner of the back of the chair	489/539	526/539	537/539
4	front corner of the seat of the chair	822/840	839/840	840/840
5	front corner of the seat of the chair	816/840	839/840	840/840
6	front corner of the armrest of the chair	109/111	110/111	111/111
7	back corner of the armrest of the chair	67/80	73/80	77/80
8	front corner of the armrest of the chair	105/109	109/109	109/109
9	back corner of the armrest of the chair	71/83	74/83	78/83
10	lower end of the front leg of the chair	1/3	1/3	1/3
11	lower end of the back leg of the chair	1/3	1/3	1/3
12	lower end of the front leg of the chair	1/3	1/3	1/3
13	lower end of the back leg of the chair	1/3	1/3	1/3
14	lower end of the leg of the chair	1/3	1/3	1/3
15	top of the chair cylinder	9/9	9/9	9/9
16	bottom of the chair cylinder	8/10	10/10	10/10
17	lower end of the front leg of the chair	1322/1352	1333/1352	1339/1352
18	lower end of the back leg of the chair	1135/1165	1156/1165	1160/1165
19	lower end of the back leg of the chair	1147/1174	1167/1174	1171/1174
20	lower end of the front leg of the chair	1308/1342	1332/1342	1336/1342
	Total	9370/9755	9647/9755	9705/9755

Table 18: Per-keypoint PCK results for category `guitar`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	head of the guitar	3885/3904	3904/3904	3904/3904
1	connection of the body and neck of the guitar	2690/2926	2883/2926	2916/2926
2	corner of the body of the guitar	1964/2039	2014/2039	2033/2039
3	corner of the body of the guitar	2046/2060	2057/2060	2060/2060
4	corner of the body of the guitar	2104/2135	2128/2135	2135/2135
5	corner of the body of the guitar	2038/2130	2112/2130	2130/2130
6	strap pin on the bottom of the body of the guitar	2015/2236	2102/2236	2148/2236
7	waist of the body of the guitar	1966/2115	2007/2115	2085/2115
8	waist of the body of the guitar	1814/1856	1855/1856	1856/1856
	Total	20522/21401	21062/21401	21267/21401

Table 19: Per-keypoint PCK results for category `helmet`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	top of the helmet	1765/1976	1949/1976	1974/1976
1	front of the helmet peak	1331/1976	1694/1976	1872/1976
2	back of the helmet peak	1451/1976	1789/1976	1915/1976
3	left side of the helmet peak	1300/1976	1738/1976	1895/1976
4	right side of the helmet peak	1343/1976	1762/1976	1912/1976
5	right bottom of the helmet	1296/1834	1592/1834	1697/1834
6	left bottom of the helmet	1351/1834	1635/1834	1722/1834
7	bottom of the visor of the helmet	1265/1654	1451/1654	1588/1654
8	bottom of the back of the helmet	1590/1766	1715/1766	1743/1766
	Total	12692/16968	15325/16968	16318/16968

Table 20: Per-keypoint PCK results for category `laptop`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	corner of the laptop body	2862/2870	2870/2870	2870/2870
1	hinge of the laptop	2334/2342	2342/2342	2342/2342
2	hint of the laptop	2193/2223	2222/2223	2223/2223
3	corner of the laptop body	2774/2780	2780/2780	2780/2780
4	corner of the laptop screen	2778/2791	2791/2791	2791/2791
5	corner of the laptop screen	2763/2770	2770/2770	2770/2770
	Total	15704/15776	15775/15776	15776/15776

Table 21: Per-keypoint PCK results for category `motorcycle`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	headlight of the motorcycle	2392/2875	2788/2875	2864/2875
1	end of the handlebar of the motorcycle	3005/3549	3320/3549	3478/3549
2	end of the handlebar of the motorcycle	3009/3545	3301/3545	3481/3545
3	seat of the motorcycle	3763/4092	4020/4092	4087/4092
4	axle of the front wheel of the motorcycle	4225/4267	4264/4267	4267/4267
5	axle of the back wheel of the motorcycle	3873/3914	3911/3914	3914/3914
6	bottom part of the frame of the motorcycle	4261/4398	4385/4398	4397/4398
7	axle of the front wheel of the motorcycle	4202/4247	4242/4247	4247/4247
8	axle of the back wheel of the motorcycle	3910/3958	3957/3958	3958/3958
9	front part of the front wheel of the motorcycle	4502/4511	4511/4511	4511/4511
10	back part of the back wheel of the motorcycle	3983/4019	4018/4019	4019/4019
11	top of the fuel tank of the motorcycle	2583/3002	2945/3002	3002/3002
12	front end of the front fender of the motorcycle	2578/2817	2800/2817	2815/2817
13	back end of the rear fender of the motorcycle	1239/2041	1654/2041	1928/2041
	Total	47525/51235	50116/51235	50968/51235

Table 22: Per-keypoint PCK results for category `mug`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	mug rim	1985/1996	1996/1996	1996/1996
1	mug rim	2038/2045	2044/2045	2045/2045
2	mug rim	2283/2307	2305/2307	2306/2307
3	mug rim	2274/2294	2293/2294	2294/2294
4	mug base rim	2718/2870	2843/2870	2868/2870
5	mug base rim	2733/2874	2845/2874	2874/2874
6	mug base rim	2642/2761	2735/2761	2760/2761
7	mug base rim	2655/2794	2780/2794	2794/2794
8	upper attachment point for the mug handle	2854/2897	2897/2897	2897/2897
9	center of the mug handle	2928/3179	3159/3179	3178/3179
10	lower attachment point for the mug handle	2597/2849	2807/2849	2841/2849
	Total	27707/28866	28704/28866	28853/28866

Table 23: Per-keypoint PCK results for category `skateboard`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	corner of the tail of the skateboard	1308/1409	1380/1409	1404/1409
1	corner of the tail of the skateboard	1314/1421	1386/1421	1415/1421
2	corner of the nose of the skateboard	1289/1445	1404/1445	1440/1445
3	corner of the nose of the skateboard	1233/1395	1337/1395	1392/1395
4	wheel of the skateboard	2680/2694	2692/2694	2693/2694
5	wheel of the skateboard	2606/2622	2615/2622	2619/2622
6	wheel of the skateboard	2613/2639	2634/2639	2638/2639
7	wheel of the skateboard	2591/2608	2604/2608	2608/2608
8	nose of the skateboard	2028/2030	2030/2030	2030/2030
9	tail of the skateboard	1814/1814	1814/1814	1814/1814
	Total	19476/20077	19896/20077	20053/20077

Table 24: Per-keypoint PCK results for category `table`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	corner of the table top	1466/1523	1515/1523	1521/1523
1	corner of the table top	1604/1654	1646/1654	1652/1654
2	corner of the table top	1640/1676	1675/1676	1675/1676
3	corner of the table top	1638/1681	1670/1681	1674/1681
4	end of the table leg	2419/2584	2547/2584	2574/2584
5	end of the table leg	2443/2583	2552/2583	2580/2583
6	end of the table leg	2420/2592	2557/2592	2586/2592
7	end of the table leg	2435/2583	2554/2583	2578/2583
8	end of the stretcher between the table legs	0/0	0/0	0/0
9	end of the stretcher between the table legs	0/0	0/0	0/0
10	end of the stretcher between the table legs	0/0	0/0	0/0
11	end of the stretcher between the table legs	0/0	0/0	0/0
	Total	16065/16876	16716/16876	16840/16876

Table 25: Per-keypoint PCK results for category `vessel`. Values shown as correct/total.

ID	Description	PCK@0.1	PCK@0.15	PCK@0.2
0	bow of a boat	1133/1142	1142/1142	1142/1142
1	lower part of the end of the bow of a boat (starboard)	396/668	553/668	634/668
2	upper part of the end of the bow of a boat (starboard)	179/292	245/292	276/292
3	upper part of the end of the bow of a boat (port)	213/323	277/323	313/323
4	lower part of the end of the bow of a boat (port)	445/717	611/717	684/717
5	lower corner of the boat stern (starboard	207/265	248/265	257/265
6	upper corner of the boat stern (starboard)	97/113	113/113	113/113
7	upper corner of the boat stern (port)	104/118	111/118	117/118
8	lower corner of the boat stern (port)	211/251	241/251	247/251
9	mast step of a sailboat	23/28	27/28	28/28
10	mast top of a sailboat	72/72	72/72	72/72
11	lower front corner of the deckhouse (starboard)	45/142	73/142	94/142
12	lower front corner of deckhouse (port)	36/125	66/125	85/125
13	lower aft corner of the deckhouse (starboard)	29/76	44/76	54/76
14	lower aft corner of the deckhouse (port)	45/108	67/108	82/108
15	upper front corner of the deckhouse (starboard)	11/26	18/26	22/26
16	upper front corner of the deckhouse (port)	12/20	16/20	18/20
17	upper aft corner of the deckhouse (starboard)	14/25	16/25	21/25
18	upper aft corner of the deckhouse (port)	15/24	17/24	22/24
19	foremost elevation point on the deck of the ship	51/58	54/58	57/58
20	lower corner of the windshield of the ship (starboard)	2/2	2/2	2/2
21	lower corner of the windshield of the ship (port)	2/2	2/2	2/2
22	upper corner of the windshield of the ship (starboard)	0/0	0/0	0/0
23	upper corner of the windshield of the ship (port)	0/1	1/1	1/1
	Total	3342/4598	4016/4598	4343/4598

Table 26: PCK averaged over seeds, semantic_threshold = 0.625

Category	PCK@0.05	PCK@0.10	PCK@0.15	PCK@0.20
airplane	0.80	0.94	0.98	0.99
bathtub	0.72	0.89	0.96	0.99
bed	0.69	0.91	0.97	0.99
bottle	0.82	0.96	0.99	0.99
cap	0.80	0.97	1.00	1.00
car	0.78	0.96	0.99	1.00
chair	0.78	0.97	0.99	0.99
guitar	0.81	0.96	0.98	0.99
helmet	0.56	0.84	0.95	0.98
knife	0.85	0.93	0.97	0.98
laptop	0.89	0.99	1.00	1.00
motorcycle	0.76	0.94	0.98	1.00
mug	0.73	0.96	0.99	1.00
skateboard	0.87	0.98	0.99	1.00
table	0.77	0.96	0.99	1.00
vessel	0.44	0.69	0.86	0.94
Mean	0.75	0.93	0.97	0.99
Median	0.78	0.96	0.98	0.99

Table 27: PCK averaged over seeds, semantic_threshold = 0.575

Category	PCK@0.05	PCK@0.10	PCK@0.15	PCK@0.20
airplane	0.79	0.94	0.98	0.99
bathtub	0.70	0.88	0.96	0.99
bed	0.68	0.91	0.96	0.99
bottle	0.83	0.96	0.98	0.99
cap	0.80	0.97	1.00	1.00
car	0.78	0.95	0.99	1.00
chair	0.77	0.96	0.99	0.99
guitar	0.81	0.96	0.98	0.99
helmet	0.55	0.84	0.94	0.98
knife	0.83	0.93	0.96	0.98
laptop	0.89	0.99	1.00	1.00
motorcycle	0.75	0.93	0.98	0.99
mug	0.73	0.96	0.99	1.00
skateboard	0.86	0.97	0.99	1.00
table	0.76	0.96	0.99	1.00
vessel	0.45	0.71	0.87	0.94
Mean	0.75	0.93	0.97	0.99
Median	0.78	0.96	0.98	0.99

Table 28: PCK averaged over seeds, semantic_threshold = 0.525

Category	PCK@0.05	PCK@0.10	PCK@0.15	PCK@0.20
airplane	0.78	0.94	0.98	0.99
bathtub	0.68	0.87	0.96	0.99
bed	0.67	0.90	0.96	0.98
bottle	0.82	0.95	0.98	0.99
cap	0.79	0.97	1.00	1.00
car	0.78	0.95	0.98	0.99
chair	0.75	0.96	0.99	0.99
guitar	0.80	0.96	0.98	0.99
helmet	0.53	0.83	0.94	0.98
knife	0.82	0.92	0.96	0.98
laptop	0.90	0.99	1.00	1.00
motorcycle	0.74	0.92	0.97	0.99
mug	0.74	0.96	0.99	1.00
skateboard	0.86	0.97	0.99	1.00
table	0.75	0.95	0.99	1.00
vessel	0.47	0.73	0.88	0.95
Mean	0.74	0.92	0.97	0.99
Median	0.76	0.95	0.98	0.99

Table 29: PCK averaged over seeds, semantic_threshold = 0.25

Category	PCK@0.05	PCK@0.10	PCK@0.15	PCK@0.20
airplane	0.76	0.92	0.97	0.99
bathtub	0.53	0.79	0.93	0.98
bed	0.62	0.86	0.93	0.97
bottle	0.75	0.91	0.97	0.99
cap	0.79	0.95	0.99	1.00
car	0.77	0.95	0.98	0.99
chair	0.68	0.93	0.98	0.99
guitar	0.78	0.95	0.98	0.99
helmet	0.45	0.77	0.91	0.96
knife	0.73	0.87	0.94	0.97
laptop	0.90	0.99	1.00	1.00
motorcycle	0.70	0.89	0.96	0.99
mug	0.74	0.96	0.99	1.00
skateboard	0.82	0.95	0.98	1.00
table	0.72	0.93	0.98	0.99
vessel	0.44	0.72	0.86	0.93
Mean	0.70	0.90	0.96	0.98
Median	0.74	0.92	0.98	0.99

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: The claims are consistent with the paper's method, experiments, and reported results.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Yes, the limitations are discussed in Section 4.5.

   Guidelines:

   - The answer [N/A] 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: [N/A]

Justification: The paper does not include theoretical results or proofs.

Guidelines:

- The answer [N/A] 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: The paper provides the main implementation, training, and evaluation details needed to reproduce the reported experimental results.

   Guidelines:

   - The answer [N/A] 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: The code, data, and reproduction instructions will be released upon acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the main training, evaluation, and implementation details necessary to understand the reported results.

   Guidelines:

   - The answer [N/A] 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: The paper does not report error bars or statistical significance analyses.

   Guidelines:

   - The answer [N/A] 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: The paper reports the compute resources required to reproduce the main experiments.

   Guidelines:

   - The answer [N/A] 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: The research complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [No]

    Justification: The paper focuses on a technical data generation pipeline and does not identify societal impacts specific to this task beyond those generally associated with standard computer vision methods.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The work does not release data or models with a high risk for misuse, such as pretrained language models, image generators, or scraped sensitive datasets.

    Guidelines:

    - The answer [N/A] 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: The original datasets, codebases, and assets used in the paper are properly credited. The corresponding licenses and terms of use will be included with the released code and data.

    Guidelines:

    - The answer [N/A] 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: The new assets introduced in the paper are documented and will be released together with the corresponding code and data.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve research with human subjects requiring IRB approval.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as part of the core research methodology.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.