
Jamais Vu: Exposing the Generalization Gap in Supervised Semantic Correspondence – Supplementary Material

Anonymous Author(s)

Affiliation

Address

email

A Additional Implementation Details

We base our model on Geo-SC [10], reusing all default hyperparameters that come with the official implementation¹, e.g., training for 2 epochs using AdamW [3] optimizer with 1.25×10^{-3} initial learning rate and 1.0×10^{-3} weight decay, coupled with one-cycle learning rate scheduler [8], with a batch size of 1. Every 5,000 iterations, models are evaluated on the validation split, and the best performing model is retained. For evaluation, unless stated otherwise, a soft-argmax window of size 15 is used.

Geo-SC specific hyperparameters are also untouched, with the contrastive $\mathcal{L}_{\text{sparse}}$ and dense $\mathcal{L}_{\text{dense}}$ objectives with gaussian noise being used, as well as feature maps dropout and pose-variant augmentation. We refer to the original publication and official implementation for in-depth description of these features.

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

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

B Ablations

Table A1: Average PCK@0.1 on SPair-71k validation set for different ablations.

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

We perform ablations of our designs in Table A1, and report results on the the SPair-71k [5] validation set which helps us chose the best performing model. It is not possible to ablate individual loss terms as they each have a distinct purpose without which the prototype cannot properly be learned: $\mathcal{L}_{\mathcal{P}}$ optimizes \mathcal{P} , $\mathcal{L}_{\mathcal{Z}}$ optimizes \mathcal{Z} , and $\mathcal{L}_{\text{geom}}$ provides a dense supervision signal, i.e., a loss for $\Phi(I, \mathbf{u})$ when \mathbf{u} is an arbitrary object pixel, i.e., not a keypoint.

We show that setting $\lambda_{\mathcal{Z}}$ to 1 or 0.1 both negatively affect performance. We believe this is due to the interaction between $\mathcal{L}_{\mathcal{Z}}$ and $\mathcal{L}_{\text{geom}}$, as a high $\lambda_{\mathcal{Z}}$ would push Φ to collapse towards defaulting to

¹<https://github.com/Junyi42/geoaware-sc>

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

Threshold	Spair-71k KAP			Spair-U KAP			Spair-71k PCK [†]			Spair-U PCK [†]			Spair-71k GA			AP-10K IS GA		
	0.01	0.05	0.10	0.01	0.05	0.10	0.01	0.05	0.10	0.01	0.05	0.10	0.01	0.05	0.10	0.01	0.05	0.10
\mathcal{K} SD [7][9]	38.2	47.5	53.0	43.4	51.6	58.5	6.3	34.4	44.4	3.3	27.7	45.4	4.2	28.3	43.3	1.3	15.3	31.0
DINOv2 [6][9]	37.8	47.1	52.8	43.3	52.4	60.6	6.7	34.4	46.0	3.7	32.1	52.1	3.6	26.3	43.4	2.3	25.6	47.0
DINOv2+SD [9]	38.4	49.6	55.9	43.7	54.2	62.8	8.1	41.0	52.8	4.7	36.6	56.6	4.8	32.7	50.8	2.4	26.1	48.1
SphericalMaps [4]	38.9	51.2	58.2	44.3	<u>55.4</u>	64.2	8.8	44.4	57.3	4.5	<u>37.8</u>	58.5	5.6	37.7	58.1	2.6	28.4	51.8
\mathcal{K} DINO+SD (S) [9]	39.1	55.4	64.1	43.8	54.7	63.9	13.0	59.7	72.0	3.6	35.5	57.2	10.2	53.6	69.7	2.8	32.1	56.6
Geo-SC [10]	39.8	59.3	67.8	43.8	54.2	62.8	20.0	69.8	78.3	<u>4.6</u>	35.1	54.9	17.2	65.6	78.0	3.7	33.6	<u>55.3</u>
Geo-SC [†] [10]	40.1	61.0	69.2	43.8	54.1	62.8	22.0	73.0	80.9	4.3	35.8	55.3	20.0	70.9	<u>82.3</u>	-	-	-
Ours	39.8	59.8	68.1	44.0	55.0	<u>64.5</u>	20.4	69.8	78.1	4.2	37.4	<u>60.3</u>	17.4	65.8	<u>77.7</u>	<u>3.5</u>	<u>33.5</u>	56.1
Ours [†]	<u>40.0</u>	<u>60.3</u>	<u>69.0</u>	<u>44.2</u>	56.0	66.0	<u>20.8</u>	<u>72.1</u>	<u>80.7</u>	4.5	41.3	64.2	<u>18.8</u>	<u>70.7</u>	82.4	-	-	-

32 predicting keypoint features \mathcal{Z} for most points, while a weight too low prevents correct prediction
 33 on the keypoints. We also test different neighbor sampling strategies for \mathcal{X}_b and \mathcal{X}_c , and show that
 34 sampling both spaces with either K-nearest neighbor or geodesic sampling is ineffective.

35 C Additional Results

36 C.1 Additional metrics

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

39 **PCK[†] [1]** PCK matches are counted correct even if the prediction lies closer to a keypoint that is
 40 not the target, which can lead to high scores when many points are grouped together, even though
 41 the system does not distinguish between them. The authors introduce PCK[†] which only considers a
 42 match correct if it lies within the threshold *and* its closest annotated point is the target.

43 **KAP [4]** PCK only considers matches when both ground-truth points are visible and does not penalize
 44 systems that predict strong similarities for points that do not correspond, for instance between the
 45 two opposite sides of a car. KAP reformulate the correspondence evaluation as a binary classification
 46 problem between the pixels that are close to the target and those those that are not. Crucially, it
 47 penalizes high predictions when a source keypoint is invisible in the target.

48 **Geo-aware subset (GA) [10]** Finally, [1],[10] and [4] noted that SC pipelines - especially unsuper-
 49 vised ones - often make mistakes because of repeated parts and object symmetries. [10] proposed
 50 evaluation on the *Geo-aware* subset of points only, e.g., the points for which there is a symmetric
 51 corresponding point.

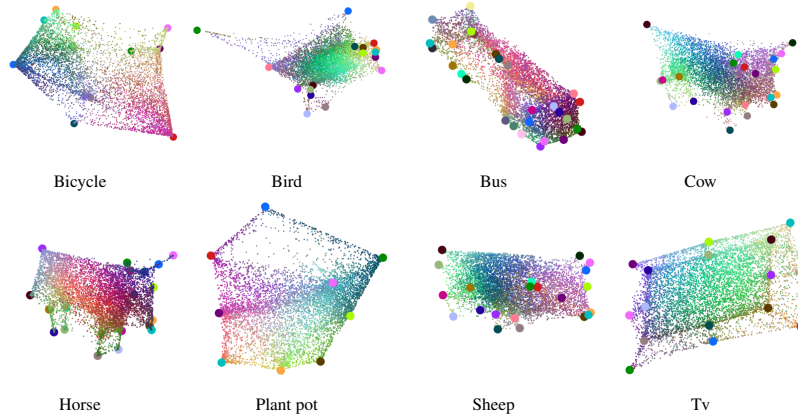


Figure A1: **Visualization of our learned canonical shapes.** Points are colored with PCA of the features.

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

C.2 Additional visualizations

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

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

D Additional SPair-U Details

We annotated images using the VGG Image Annotator [2]. We further post-processed the annotations into JSON files replicating the structure of SPair-71k annotations, i.e., per-image annotations and a list of testing pairs. This allows SPair-U to function as a drop-in replacement for SPair-71k evaluation in any semantic correspondence evaluation script. Note that it is designed to be a benchmark of unseen semantic points intended for evaluating the generalization ability of SC models, therefore does not come with a training or validation split. We present the full list of keypoint semantics of SPair-U in Table A3, per-category statistics in Table A4, and some keypoint visualization in Fig. A2.

Table A3: List of SPair-U keypoint semantics.

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

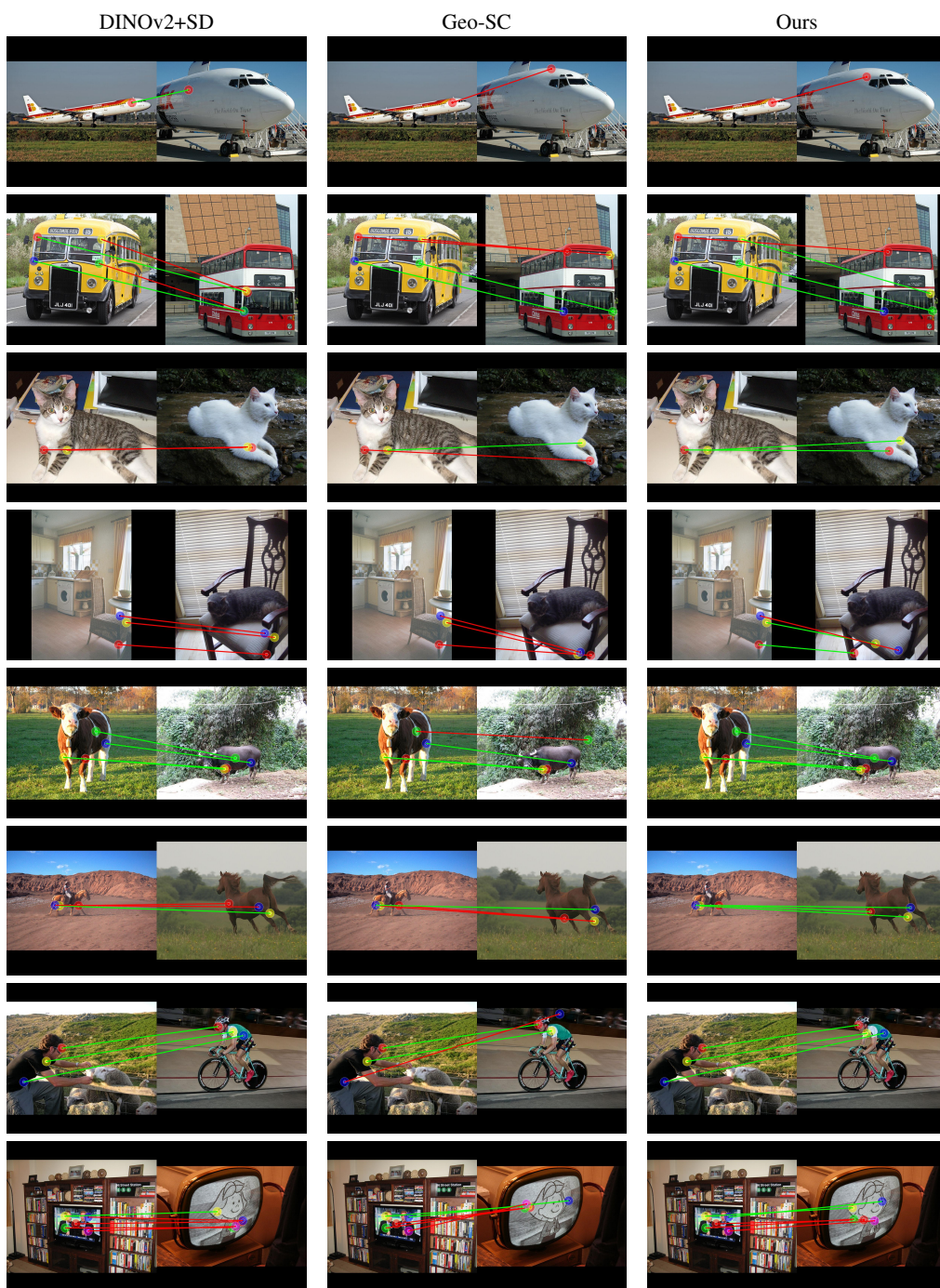


Figure A3: **Visualization matches for SPair-U.** Green lines are correct, red ones are incorrect.

77 References

- 78 [1] Mehmet Aygün and Oisin Mac Aodha. Demystifying unsupervised semantic correspondence estimation.
79 In *ECCV*, 2022.
- 80 [2] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In
81 *International Conference on Multimedia*, 2019.
- 82 [3] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv:1711.05101*, 2017.
- 83 [4] Octave Mariotti, Oisin Mac Aodha, and Hakan Bilen. Improving semantic correspondence with viewpoint-
84 guided spherical maps. In *CVPR*, 2024.
- 85 [5] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic
86 correspondence. *arXiv:1908.10543*, 2019.
- 87 [6] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre
88 Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual
89 features without supervision. *TMLR*, 2024.
- 90 [7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution
91 image synthesis with latent diffusion models. In *CVPR*, 2022.
- 92 [8] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large
93 learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*,
94 2019.
- 95 [9] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and
96 Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic
97 correspondence. In *NeurIPS*, 2023.
- 98 [10] Junyi Zhang, Charles Herrmann, Junhwa Hur, Eric Chen, Varun Jampani, Deqing Sun, and Ming-Hsuan
99 Yang. Telling left from right: Identifying geometry-aware semantic correspondence. In *CVPR*, 2024.