Shape-of-You: Fused Gromov-Wasserstein Optimal Transport for Semantic Correspondence in-the-Wild

Supplementary Material

Before presenting the extended analyses, we briefly outline the structure of this supplementary material, which provides further insights to complement the main paper.

- **Sec. A: Additional experiments.**
  - Sec. A.1 examines the effect of different 2D feature backbones, comparing DINOv2 [1] and fused DINOv2+SD [6] in terms of pseudo-label quality and final correspondence accuracy.
  - Sec. A.2 analyzes key design choices of the pseudo-label generator, including the number of anchors K, the feature–geometry trade-off α, the KL regularization strength ρ in UOT, and the choice of 3D foundation backbone [4, 5].
  - Sec. A.3 investigates training-time hyperparameters, focusing on the soft-target mixing weight β and its synergy with pseudo-label quantity (top-k) and cycle-consistency.
  - Sec. A.4 provides a systematic evaluation across various challenging conditions (e.g., viewpoint, occlusion, pose) and an in-depth per-category analysis.
- **Sec. B: Failure cases.** Typical failure cases of the pseudo-label generator are summarized and discussed.
- **Sec. C: Implementation details.** We provide low-level implementation details and the full set of hyperparameters used in all experiments.
- **Sec. D: Algorithm.** We present PyTorch-style pseudocode describing the complete pseudo-label generation pipeline.
- **Sec. E: Additional visualizations.** Additional qualitative visualizations on SPair-71k are provided to complement the quantitative results in the main paper.

A. Additional experiments

We provide additional experiments and analyses to complement the main paper. This section is organized into four parts: **(a)** Sec. A.1 studies how our method behaves under different 2D foundation feature backbones, comparing DINOv2 and fused DINOv2+SD features in terms of both pseudo-label quality and final correspondence accuracy; **(b)** Sec. A.2 analyzes the sensitivity of our pseudo-label generator to key design choices, including the number of anchors K, the feature–geometry trade-off α, the KL regularization strength ρ in UOT, and the choice of 3D foundation backbone; **(c)** Sec. A.3 investigates training hyperparameter sensitivity, focusing on the soft-target mixing weight β in our loss; and **(d)** Sec. A.4 provides a systematic evaluation across various challenging conditions and an in-depth per-category analysis.

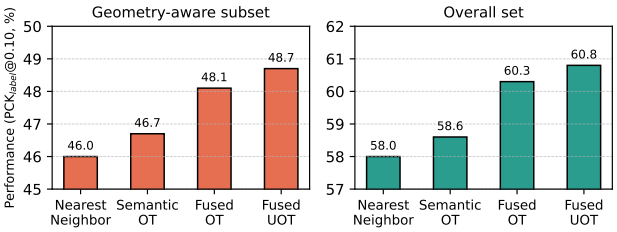


Figure 1. **Pseudo-label analysis with DINOv2 backbone.** Performance **(PCK$_{label}$@0.1)** on the *geometry-aware subset* (left) and *overall set* (right). Using only DINOv2 features, we compare four pseudo-label generation strategies: Nearest Neighbor, Semantic OT, Fused OT, and Fused UOT. Both the geometry-aware and overall subsets show a consistent, monotonic improvement across methods, with clear gains from incorporating semantic OT and our geometry-aware matching. This confirms that our pseudo-labeling remains effective even with a weaker backbone.

A.1. Backbone

Pseudo-labels with DINOv2 backbone. As in the main paper, the PCK$_{label}$ metric evaluates only target keypoints within the SAM mask. We note that evaluating the DINOv2+SD baseline across all ground-truth keypoints yields identical improvement trends (NN 62.4% $\rightarrow$ Semantic OT 63.4% $\rightarrow$ Fused OT 64.6% $\rightarrow$ Fused UOT 64.9%) demonstrating robustness beyond the masked regions. To verify our conclusions from Fig. 5 (in main) are not tied to this backbone choice we repeat the evaluation using *only* DINOv2 features. Fig. 1 reports the PCK$_{label}$@0.1 results. On the geometry-aware subset, performance improves from 46.0% (Nearest Neighbor) to 46.7% (Semantic OT), 48.1% (Fused OT) and 48.7% (Fused UOT), a +2.7%p gain over the NN baseline. A similar trend occurs on the overall set, with performance increasing from 58.0% (NN) to 58.6% (Semantic OT), 60.3% (Fused OT) and 60.8% (Fused UOT), a +2.8%p total gain. Although the absolute PCK$_{label}$ values are lower than with the DINOv2+SD backbone, the *relative* improvements from incorporating geometric consistency remain consistent. This confirms our pseudo-labeling is complementary to the visual backbone and provides benefits with different feature representations.

Evaluation across different feature backbones. Tab. 1 further compares the final correspondence accuracy when training our model using pseudo-labels generated from the

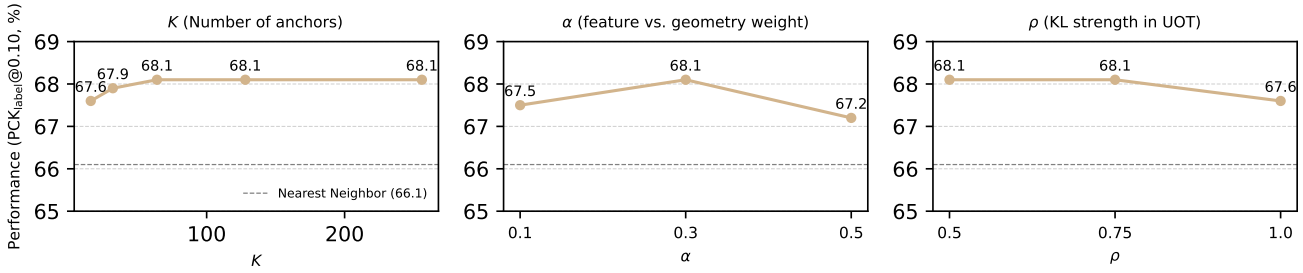


Figure 2. **Pseudo-label hyperparameter ablation on SPair-71k (PCK$_{\text{label}}$@0.1).** We study three key hyperparameters of our pseudo-label generator: **(Left)** number of anchors K, **(Middle)** feature–geometry trade-off α, and **(Right)** KL regularization strength ρ in UOT. In all cases, the performance is measured as PCK$_{\text{label}}$@0.1 on the SPair-71k test set.

fused DINOv2+SD backbone, while evaluating the trained model under two different backbone choices: (1) DINOv2-only features and (2) DINOv2+SD features. Note that these adapter evaluations were conducted prior to the integration of the relaxed cycle-consistency filtering discussed in the main paper. Therefore, the reported PCK scores isolate the impact of the backbone and the base FGW pseudo-labels without the additional cycle-consistency boost. Across both settings, our method consistently improves PCK at all thresholds, confirming that the geometric priors introduced by our matching remain beneficial regardless of the underlying feature strength. With the weaker DINOv2 backbone, our approach yields substantial gains over the baseline (+1.2, +1.9, +4.8 at PCK@0.01/0.05/0.1 respectively), demonstrating that our pseudo-labels help compensate for limited semantic discriminability in the features. When moving to the stronger DINOv2+SD backbone, the absolute accuracy increases—as expected from higher-quality visual features—yet our method continues to provide additional improvements (+0.2, +0.7, +4.0). Interestingly, the relative gain at the higher threshold (PCK@0.1) remains particularly large in both cases, suggesting that the geometric consistency enforced by our FGW matching systematically enhances mid-to-coarse correspondence quality. Overall, these results reinforce that our improvements do not rely on a specific backbone and that SoY complements both standard visual features and diffusion priors. The model consistently benefits from pseudo-labels generated with DINOv2+SD, even when the evaluation backbone differs, indicating that our geometric alignment signal generalizes robustly across feature extractors.

A.2. Pseudo-label sensitivity

Number of anchors K. Fig. 2 (left) shows the effect of varying the number of anchors $K \in \{16, 32, 64, 128, 256\}$ used in the anchor-based FGW linearization. PCK$_{\text{label}}$@0.1 improves from 67.6% at K=16 to 68.1% at K=64, and then saturates (68.1% for K=64, 128, 256). This indicates that our method is not overly sensitive to the exact choice of K once a moderate number of anchors is available, and that K=64 provides a good trade-off between robustness and computational cost.

Method	PCK@0.01	PCK@0.05	PCK@0.1
DINOv2 Backbone			
DINOv2 [1]	6.4	40.2	55.7
Ours (DINOv2)	7.6 (+1.2)	42.1 (+1.9)	60.5 (+4.8)
DINOv2 + SD Backbone			
DINOv2 + SD [6]	8.8	48.3	63.5
Ours (DINOv2 + SD)	9.0 (+0.2)	49.0 (+0.7)	67.5 (+4.0)

Table 1. Comparison of PCK scores (PCK@0.01, 0.05, 0.1) with different feature backbones. All models are trained using pseudo-labels generated from DINOv2+SD without relaxed cycle-consistency. Our method improves over both baselines with gains shown in red.

3D backbone	PCK$_{\text{label}}$@0.1
VGGT [4]	**68.1**
DUSt3R [5]	67.1

Table 2. Effect of the 3D foundation backbone on pseudo-label quality on SPair-71k. We report PCK$_{\text{label}}$@0.1 for our pseudo-labels when using either VGGT or DUSt3R to obtain 3D structure.

Feature–geometry weight α. Fig. 2 (middle) studies the fusion weight $\alpha \in \{0.1, 0.3, 0.5\}$ between feature similarity and geometric cost in the fused OT objective. We observe a clear peak at α=0.3 (68.1%), whereas both a too feature-dominated setting (α=0.1, 67.5%) and a too geometry-dominated setting (α=0.5, 67.2%) lead to lower PCK$_{\text{label}}$. This confirms that balancing semantic and geometric cues is important, and our default choice α=0.3 lies near the optimum.

KL strength ρ in UOT. Fig. 2 (right) analyzes the KL regularization strength $\rho \in \{0.5, 0.75, 1.0\}$ in the unbalanced OT formulation. PCK$_{\text{label}}$@0.1 remains high and stable for ρ=0.5 and ρ=0.75 (both 68.1%), but slightly decreases at ρ=1.0 (67.6%). This suggests that overly strong KL regularization, which enforces the marginals too strictly, can harm pseudo-label quality, while moderate relaxation yields more robust correspondences. We therefore use ρ=0.75 in all our experiments.

β	PCK@0.01	PCK@0.05	PCK@0.1
0.25	**9.3**	**49.4**	67.2
0.50 (default)	9.0	49.0	**67.5**
0.75	6.4	44.9	65.2

Table 3. Sensitivity of the soft-target weight β in the training loss on SPair-71k after 20 epochs. Evaluations here are conducted without relaxed cycle-consistency. Within the range $\beta \in [0.25, 0.50]$, performance varies only mildly, whereas a larger value β=0.75 noticeably degrades accuracy.

3D foundation backbone. Finally, we study the impact of the underlying 3D foundation model used to obtain geometric structure. In all our main experiments, we adopt VGGT as the 3D backbone, which yields $\text{PCK}_{\text{label}}$@0.1 of 68.1% for our pseudo-labels. Tab. 2 compares this setting with an alternative 3D foundation model, DUSt3R [5]. DUSt3R is originally designed for multi-view 3D reconstruction and correspondence, where several views of the *same* scene are jointly encoded to recover accurate geometry. In our semantic correspondence setting, however, the source and target images typically depict different scenes or object instances, so each image is effectively processed in a single-view regime. As also noted in the VGGT paper [4], DUSt3R's reconstruction quality degrades noticeably when only a single RGB view is available. Consistent with this observation, replacing VGGT with DUSt3R leads to a modest drop in pseudo-label quality from 68.1% to 67.1% $\text{PCK}_{\text{label}}$@0.1. Nevertheless, even with DUSt3R our method still improves over the nearest-neighbor baseline, indicating that the FGW formulation can still exploit 3D cues as long as the backbone provides reasonably stable structure.

A.3. Training hyperparameter sensitivity

Soft-target weight β. We first ablate the soft-target weight β used in our training loss while keeping all other settings fixed. Note that to strictly isolate the effect of β, this evaluation does not employ the relaxed cycle-consistency filtering. Tab. 3 reports PCK scores on SPair-71k after 20 epochs for $\beta \in \{0.25, 0.50, 0.75\}$. Within the range $\beta \in [0.25, 0.50]$, the performance is fairly stable: PCK@0.10 changes only slightly (67.2% vs. 67.5%) and the differences at stricter thresholds (PCK@0.01/0.05) are within 0.4 points. In contrast, a larger value β=0.75 substantially degrades performance (6.4/44.9/65.2 at PCK@0.01/0.05/0.10), indicating that over-emphasizing noisy soft targets can be harmful.

We also observe that when training is extended to roughly 50 epochs, the default setting β=0.50 yields a slight further improvement (from 9.0/49.0/67.5 to 9.3/49.5/67.7 at PCK@0.01/0.05/0.10). This suggests that a moderate soft-target weight may require a few more epochs to fully exploit the denoising effect of the soft supervision whereas a smaller value β=0.25 reaches its best performance earlier but tends to plateau. Overall, these results indicate that our method is reasonably robust to the choice of β as long as it lies in a moderate range (e.g., 0.25–0.50) while too large values giving excessive emphasis to noisy labels (*e.g.*, β=0.75) should be avoided.

Setting (hard labels, β=0)	Top-k candidates				Ours (β=0.5, k=3)
	k=1	k=3	k=5	k=10	
w/o relaxed c.c.	66.8	66.8	66.7	66.2	-
w/ relaxed c.c.	67.1	67.4	67.4	66.8	**67.9**

Table 4. Synergy between pseudo-label quantity (top-k) and soft-target loss on SPair-71k (PCK@0.1). Filtering multiple candidates with relaxed cycle-consistency improves performance, and combining them with our soft-target loss yields the optimal result.

Synergy with top-k and cycle-consistency. To further investigate the relationship between pseudo-label quantity and our proposed training components, Tab. 4 analyzes the impact of retrieving multiple matches (top-k). When training with hard labels (β=0) without cycle-consistency, increasing the number of candidates k introduces excessive noise causing accuracy to drop (66.8% $\rightarrow$ 66.2% for k=10). However, applying relaxed cycle-consistency effectively filters this noise allowing the model to benefit from the richer candidate pool and peaking at k=3, 5 (67.4%). Finally, combining this expanded cycle-consistent candidate set with our soft-target loss (β=0.5) achieves the optimal performance of **67.9%**, demonstrating the complementary benefits of geometric filtering and soft supervision.

A.4. Systematic evaluation & per-category analysis

Systematic evaluation. To provide a deeper understanding of our method under challenging in-the-wild conditions, we systematically evaluate pseudo-label quality categorizing the test set by azimuth difference occlusion level and pose. Tab. 5 summarizes these results. Our approach yields consistent improvements across most variations demonstrating strong robustness against severe occlusions (+2.5%p) and challenging cross-pose alignments (+1.6%p). The notable gain in frontal same-pose pairs (+8.2%p) further confirms that our geometric lifting effectively refines ambiguous 2D semantic matches.

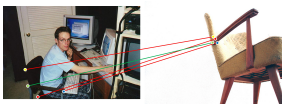
Analysis of specific challenges (car and boat). While our framework improves pseudo-label quality across 17 out of 18 categories, we provide an in-depth analysis of specific cases where distinct challenges remain. For the *car* category, degradation primarily occurs at mid-range viewpoints (45°–135°). This stems from severe semantic aliasing between symmetric parts such as front and rear windshields. When these visually similar structures are incorrectly matched as initial anchors they often maintain structurally plausible geometric relationships. Consequently the

	(a) Azimuth Analysis (All | Car (C))										(b) Occlusion			(c) Pose				
Variable	0°	45°	90°	135°	180°	0°(C)	45°(C)	90°(C)	135°(C)	180°(C)	None	Part.	Heavy	F→F	L→L	R→R	Un.	Cross
NN	65.3	65.2	58.9	57.5	57.3	88.1	65.0	39.4	23.1	15.3	63.8	58.8	59.4	62.4	74.8	76.7	57.7	67.3
Ours	69.9	67.9	59.7	58.3	58.6	89.7	61.9	33.6	20.9	17.1	66.4	61.4	61.9	70.6	75.9	78.7	60.8	68.9
Δ	+4.6	+2.5	+0.8	+0.7	+1.3	+1.6	-3.1	-5.8	-2.2	+1.8	+2.6	+2.6	+2.5	+8.2	+1.1	+2.0	+3.1	+1.6

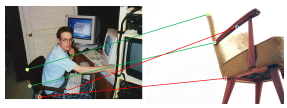
Table 5. Systematic evaluation of pseudo-label quality ($PCK_{label}@0.1$) across various challenging conditions. Our framework provides consistent gains in most scenarios particularly under extreme pose variations and heavy occlusions while also revealing specific challenges in mid-range azimuths for symmetric objects.

(a) NN (broken structure)

(b) Ours (broken structure)



(c) NN (noisy features)



(d) Ours (noisy features)

Figure 3. **Failure modes of pseudo-label generation.** Top: failure when the object breaks into disconnected parts. Bottom: failure when the features used for pseudo labels are highly noisy and all correspondences collapse to a local region.

linearized GW cost propagates these initial errors rather than correcting them highlighting a fundamental limitation when severe 2D ambiguity aligns with 3D structural symmetry.

For the *boat* category, incorporating our relaxed cycle-consistency successfully improves the final accuracy beyond the zero-shot baseline. However, we hypothesize that soft targets may over-smooth the geometric signal. In failure cases, the trained model tends to match similar local features rather than preserving the global geometric structure. In both cases, we leave more detailed investigation for future work.

B. Failure cases

Pseudo-label failure modes. Fig. 3 summarizes two typical failure modes of our pseudo-labels. In the first row (broken structure), nearest-neighbor (NN) matching produces an incorrect correspondence along the bicycle frame, whereas our method corrects the violet keypoint by enforcing global geometric consistency. However, when the object is physically broken into disconnected parts (the detached pedal), the green keypoint has no structurally consistent counterpart, so our FGW refinement can no longer rely on geometry and therefore fails to update the NN pseudo-label. In the second row (noisy features), the DINOv2+SD features used for pseudo-label generation are themselves highly ambiguous, causing multiple points to collapse onto a small spurious region on the chair. Since both the semantic cost and anchor selection are driven by these noisy similarities, our refinement is also misled and cannot recover the correct correspondences. Taken together, these examples highlight that our pseudo-label generator still depends on reasonably coherent 3D structure and sufficiently informative 2D features to produce reliable matches. In rare cases, such incorrectly generated pseudo-labels may provide slightly inconsistent supervision during training and can mildly bias the learned matcher, suggesting an interesting direction for making our framework more robust to pseudo-label noise in future work.

Group	Parameter	Value
Semantic UOT	KL penalty ρ	0.75 (UOT marginal relaxation)
	Entropy ε	1 (implicit Sinkhorn regularization)
	Mass distribution	Uniform over valid patches
	Cost C_{sem}	$1 - \text{cosine\_sim}(f_i, f_j)$
FGW Fusion	Fusion weight α	0.3 (semantic vs. geometric balance)
	Distance metric	3D Euclidean distance in lifted space
	Normalization	Both costs normalized before fusion
FGW Refinement	Anchor count K	64 anchors per iteration
	Iterations T	5 refinement steps
	Cycle-consistency δ	Quantile threshold $q = 0.01$
	Anchor mass	Uniform: $\hat{\pi}_{i'j'} = 1/K$
	Linearized cost	$C_{\text{geo}}(i,j) = \frac{1}{K}\sum \lvert D_A(i, a_A) - D_B(j, a_B)\rvert$
Training	Soft-target mixing β	0.5
	Temperature τ	Learnable
	Dense-loss noise ϵ	Gaussian noise for regularization
	Optimizer	AdamW
	Optimizer args	lr = 5e-3, weight_decay = 1e-3
	LR scheduler	OneCycleLR
	Scheduler steps	total_steps = 2e+5
3D Lifting	Backbone 3D model	VGGT (pretrained)
	Patch grid	60×60
	Interpolation	Bilinear interpolation of 3D maps
	Intra-structure	Distance matrices D_A and D_B from 3D points

Table 6. **Hyperparameters used in the Shape-of-You (SoY) framework.** Values reflect the unified configuration used across all experiments.

C. Implementation details

We summarize all hyperparameters in Tab. 6. Semantic UOT uses a KL penalty $\rho = 0.75$, entropy regularization $\varepsilon = 1$, uniform patch masses, and a cosine-based cost C_{sem}. For FGW fusion, we use the corrected semantic–geometric balance of $\alpha = 0.3$, and normalize both semantic and geometric costs before combining them.

Anchor-based refinement runs for $T = 5$ iterations with $K = 64$ mutual anchors per iteration. The cycle-consistency tolerance δ is determined by a data-driven quantile threshold $q = 0.01$ over the 3D cycle-error dis-

tribution and serves mainly to filter out clear outliers. Final anchors are ranked by a combined score favoring both high transport confidence and low geometric distortion, making the method robust to the exact choice of q.

Training uses soft-target mixing $\beta = 0.5$, a learnable temperature τ, Gaussian noise in the dense loss, and an AdamW optimizer with a OneCycleLR schedule as specified in Tab. 6. For 3D lifting, we employ a pretrained VGGT backbone, lift images to a 60×60 grid via bilinear interpolation, and construct intra-structure distance matrices D_A and D_B from the resulting 3D points for use in the FGW term.

D. Algorithm

To make our pseudo-label generator easy to reproduce, we provide PyTorch-style pseudocode for the full FGW pipeline in Alg. 1. The code explicitly shows (i) the initial semantic UOT matching, (ii) the construction of 3D distance matrices, (iii) the anchor-based linearization of the FGW structural term, and (iv) the iterative re-solving of unbalanced OT with the fused semantic–geometric cost. This low-level view complements the high-level description in the main paper and clarifies how each component of SoY is implemented in practice.

E. Additional visualization

In this section, we present additional qualitative comparisons on SPair-71k [3] across all object categories. Figures 4 and 5 visualize dense correspondences produced by DistillDIFT [2], DINOv2+SD [6], and our method, where correct and incorrect matches are highlighted in green and red, respectively. Across a variety of categories, SoY tends to produce sharper and more globally consistent correspondences under large viewpoint, scale, and appearance changes, qualitatively complementing the quantitative improvements reported in the main paper.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 1, 2

[2] Frank Fundel, Johannes Schusterbauer, Vincent Tao Hu, and Björn Ommer. Distillation of diffusion features for semantic correspondence. In *IEEE Winter Conference on Applications of Computer Vision*, 2025. 5, 7, 8

[3] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv prepreint arXiv:1908.10543*, 2019. 5, 7, 8

[4] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2025. 1, 2, 3

[5] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2, 3

[6] 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*, 2023. 1, 2, 5, 7, 8

Algorithm 1 PyTorch-style pseudocode for our pseudo-label generation.

```
# Inputs:
#   F_A, F_B: feature maps of image A and B
#   V_A, V_B: 3D points (vertices) of image A and B
#   T:  number of refinement iterations
#   K:  number of anchors
#   alpha: feature-vs-geometry trade-off
#   rho:   KL strength in UOT
#   iters: #Sinkhorn iterations
# Output:
#   pi_T: final transport plan

def unbalanced_sinkhorn(C, rho, iters):
    # 1) convert cost to log-kernel (soft affinity)
    Z = -C / rho                  # log K = - cost / rho
    m, n = C.shape

    # uniform marginals: mu_i = 1/m, nu_j = 1/n
    log_mu = torch.full((m,), -math.log(m))  # log(1/m)
    log_nu = torch.full((n,), -math.log(n))  # log(1/n)

    u = torch.zeros_like(log_mu)
    v = torch.zeros_like(log_nu)

    for _ in range(iters):
        # log-domain row / col updates (unbalanced)
        u = rho * (log_mu - torch.logsumexp(Z + v[None, :], dim=-1))
        v = rho * (log_nu - torch.logsumexp(Z + u[:, None], dim=-2))

    log_pi = Z + u[:, None] + v[None, :]
    pi = torch.exp(log_pi)        # final UOT plan
    return pi

# ---------- Stage 1: initial semantic matching ----------
# semantic cost from cosine similarity
C_sem = 1.0 - F_A @ F_B.T           # semantic cost matrix
pi = unbalanced_sinkhorn(C_sem, rho=rho, iters=iters)

# ---------- Pre-compute 3D distance matrices ----------
D_A = pairwise_dist(V_A)            # ||V_A[i] - V_A[j]||_2
D_B = pairwise_dist(V_B)            # ||V_B[i] - V_B[j]||_2

# ---------- Stage 2: iterative FGW refinement ----------
for t in range(1, T + 1):
    # 1) select 3D cycle-consistent mutual anchors
    anchors = select_anchors(pi, V_A, V_B, k=K)

    # 2) build geometric cost from anchors
    C_geo = torch.zeros_like(C_sem)
    for a_s, a_t in anchors:
        # distances to anchor on each shape
        dist_A = D_A[:, a_s][:, None].expand_as(C_geo)
        dist_B = D_B[:, a_t][None, :].expand_as(C_geo)
        # cycle-consistent structure cost
        C_geo += (dist_A - dist_B).abs()

    # 3) fuse normalized semantic & geometric costs
    C_sem_n = normalize(C_sem)      # scaling [0, 1]
    C_geo_n = normalize(C_geo)
    C_total = (1.0 - alpha) * C_sem_n + alpha * C_geo_n

    # 4) re-solve unbalanced OT with fused cost
    pi = unbalanced_sinkhorn(C_total, rho=rho, iters=iters)

pi_T = pi
```

Figure 4. Visual comparison of semantic correspondences on SPair-71k [3] across DistillDIFT [2], DINOv2+SD [6], and our approach. Correct and incorrect matches are indicated by green lines and red lines, respectively.

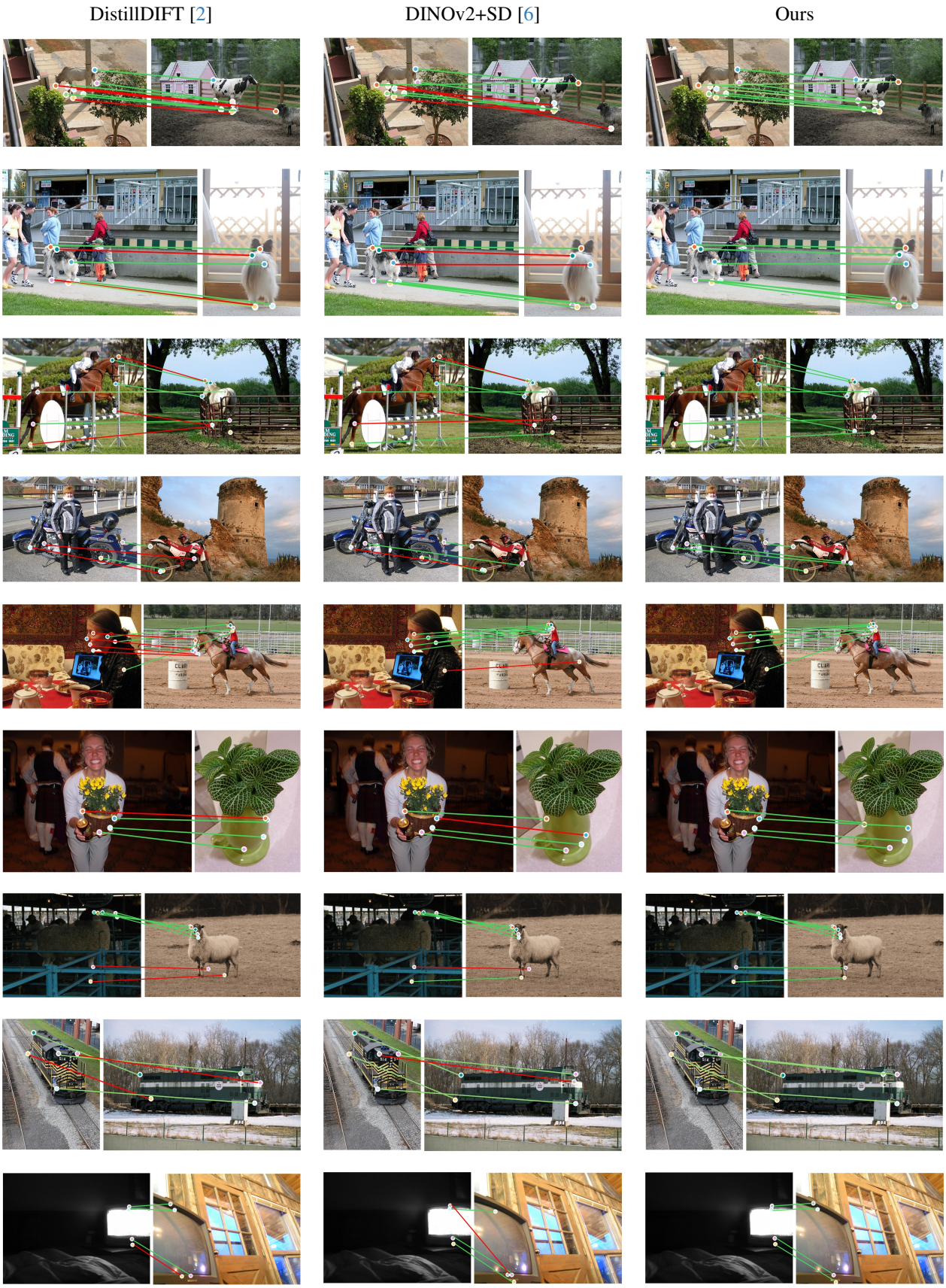


Figure 5. Visual comparison of semantic correspondences on SPair-71k [3] across DistillDIFT [2], DINOv2+SD [6], and our approach. Correct and incorrect matches are indicated by green lines and red lines, respectively.