Learning 3D Scene Analogies with Neural Contextual Scene Maps

Supplementary Material

A. Method Details

A.1. Contextual Descriptor Fields

Network Architecture Contextual descriptor fields described in Section 3.2 gather semantic and geometric information near query locations to summarize scene context information. The contextual descriptor field consists of 6 transformer encoder layers [12, 44], and each encoder layer contains multi-head attention models with 8 heads. For the semantic embedding, we use a learnable embedding of size 32, and for the distance embedding, we use a multi-layer perceptron (MLP) with a single hidden layer to produce an embedding of size 32. In addition, the descriptor fields operate on a lightweight scene presentation, where we sample 50 points per object in the scene point cloud using farthest point sampling [14].

Training To train descriptor fields, we build a dataset consisting of scene triplets for contrastive learning. For obtaining positive pairs, we replace each object in the source scene with a randomly sampled object from another scene sharing the same semantic labels. For negative pairs, we add translation noise sampled from the uniform distribution $\mathcal{U}(-0.5, 0.5)$ and z-axis rotation noise with rotation angles sampled from $\mathcal{U}(-90°, 90°)$. In addition, we uniformly sample 20^3 grid points from each object bounding box and an equal number of points sampled near the object surface as query points for training. For each object pair, we associate each grid point via nearest neighbor matching and surface point via Hungarian matching [24]. Then, during training, we minimize the contrastive learning objective (Equation 2) using the Adam [22] optimizer with a learning rate 10^{-4} and query point batch size 4 for 10^4 epochs.

A.2. Affine Map Estimation

As explained in Section 3.3, we estimate large, global transformations using affine maps. First, we combinatorially associate object pairs in scenes S_{tgt} and S_{ref} to extract initial sets of affine maps. Namely, the set of affine maps is created by associating N_{ortho} uniformly sampled transforms in $SO(2)$ with translations between object pairs $(o_{\text{tgt}}, o_{\text{ref}})$ with centroids $(\mathbf{c}_{\text{tgt}}, \mathbf{c}_{\text{ref}})$. This can be formally expressed as follows, $\{(A_{\text{init}}, b_{\text{init}})\} := \{(T_{\text{ortho}}, -T_{\text{ortho}}\mathbf{c}_{\text{tgt}} + \mathbf{c}_{\text{ref}})\}$, where $T_{\text{ortho}} \in SO(2)$. Here, we set N_{ortho} by combinatorially associating $N_{\text{rot}} = 4$ uniformly sampled rotations with $N_{\text{rfl}} = 2$ reflections along the x and y axes. From the initial set, we select K_{coarse} affine maps with the smallest cost specified in Equation 5.

Outlier Object Rejection Prior to gradient descent optimization, we perform a simple filtering procedure to remove outlier objects in the region of interest, i.e., objects that cannot be matched to the reference scene. For each selected affine map, we identify object instance matches between the region of interest and the reference scene. To elaborate, we create a distance matrix $D \in \mathbb{R}^{N_{\text{RoI}} \times N_{\text{Ref}}}$, where N_{RoI} and N_{Ref} are number of object instances in the region of interest and reference scene respectively. The $(i, j)^{\text{th}}$ entry of the distance matrix is initially set as the point cloud centroid distance between the i^{th} object in the region of interest after affine map warping and the j^{th} object in the reference scene. We then assign infinity values to matrix entries where the object semantic labels disagree. Finally, we perform Hungarian matching on D to find object instance matches, and identify objects in the region of interest as outliers if the distance matrix value to the matched object is over a threshold (which is set to 2.0 in all our experiments). After outlier removal, we keep affine maps with the largest number of inlier objects and optimize each affine map with gradient descent.

A.3. Local Displacement Map Estimation

Given the estimated set of affine maps, our method finds local displacement maps for fine-grained scene context alignments. To obtain local displacement maps, we first minimize Equation 6 with gradient descent, treating each local displacement $\delta \in \mathbb{R}^3$ as an independent vector. Then, we find the radial basis function weights w_k from Equation 4 to fit the optimized local displacements δ_{opt}. Here, the basis functions are fit by minimizing the following equation [7, 45],

$$\min_{w} \sum_{\delta_{\text{opt}}} \|d_w(\mathbf{x}; P_{\text{RoI}}) - \delta_{\text{opt}}\|^2 \tag{A.1}$$

$$+ \lambda \iint \sum_{i,j} \left(\frac{\partial^2 d_w}{\partial x_i \partial x_j}\right)^2 dx_i dx_j, \tag{A.2}$$

where $d_w(\mathbf{x}; P_{\text{RoI}}) = \sum_k w_k \phi(\|\mathbf{x} - \mathbf{p_k}\|)$ is the local displacement map. To ensure smooth maps are resilient to noise from outliers, we incorporate a regularization term (Equation A.2) with a weighting parameter λ set to 0.5.

B. Additional Experimental Results

B.1. Comparison with Additional Baselines

We compare our method with additional baselines using vision foundation models. Recall in Section 4.1 we design

Metric	Bijectivity PCP		Chamfer Acc.	
Threshold	0.25	0.50	0.15	0.20
Multi-view Semantic Corresp. (DeiT III [43])	0.05	0.09	0.24	0.45
Multi-view Semantic Corresp. (MAE [18])	0.04	0.08	0.23	0.45
Visual Feature Field (DeiT III [43])	0.31	0.34	0.58	0.67
Visual Feature Field (MAE [18])	0.50	0.56	0.70	**0.80**
Ours	**0.70**	**0.73**	**0.71**	0.76

Table B.1. 3D scene analogy comparison in manually collected scene pairs in 3D-FRONT [16]. We test baselines with additional vision foundation model features (DeiT III [43], MAE [18]).

Metric	Bijectivity PCP		Chamfer Acc.	
Threshold	0.25	0.50	0.15	0.20
Ours w/o Semantic Emb.	0.50	0.55	0.68	**0.81**
Ours w/o Distance Emb.	0.67	0.70	0.69	0.76
Ours	**0.70**	**0.73**	**0.71**	0.76

Table B.2. Ablation study of the semantic and distance embeddings on manually collected scene pairs from 3D-FRONT [16]. We report the metric values at varying thresholds.

the multi-view semantic correspondence [13, 29] and visual feature field [49] baselines that exploit DINOv2 [29] features for finding scene analogies. Here, we consider additional baselines using different vision foundation models, namely MAE [18] and DeiT III [43]. Table B.1 shows the accuracies of predicted scene analogies in the manually collected scene pairs in 3D-FRONT [16]. Our method constantly outperforms the newly added baselines, suggesting the effectiveness of the learned descriptor field features for scene context reasoning.

B.2. Ablation on Semantic and Distance Embeddings

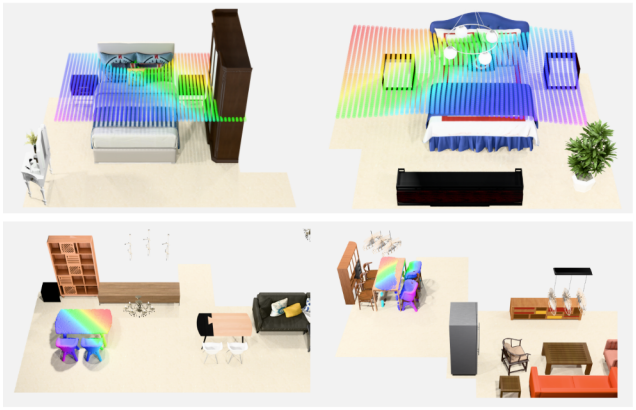
We conduct an additional ablation on using semantic and distance embeddings for scene analogy estimation. Recall in Section 3.2 our context descriptor fields aggregate semantics and distance information of keypoints near the query point and produces semantic and distance embeddings. Table B.2 shows the scene map accuracy measured from manually collected scenes in 3D-FRONT [16], where optimal performance is achieved when both are used as input. Omitting semantic embeddings results in a large performance drop, highlighting the importance of encoding nearby semantic information for effective scene analogy estimation.

B.3. Quantitative Evaluation of Sim2Real Map Estimation

We conduct a quantitative evaluation of Sim2Real map estimation using 102 manually collected Sim2Real scene pairs from 3D-FRONT [16] and ARKitScenes [6]. We compare our method against the 3D Point Feature Field baseline, which is the strongest performing baseline in Tables 1, 2. As shown in Table B.3, our method outperforms the 3D Point Feature Field baseline in Chamfer accuracy at threshold 0.15.

	3D Point Feature Field	Ours
Chamfer Acc.	0.55	**0.66**

Table B.3. Quantitative evaluation of 3D scene analogy estimation in manually collected Sim2Real scene pairs from 3D-FRONT [16] and ARKitScenes [6] We report the Chamfer Accuracy at threshold 0.15.



Target Reference

Figure B.1. Qualitative results of scene analogies found from our method on object groups with varying cardinalities. We show results both for near-surface and open-space points.

Cardinality	Identical	Different	Original
Chamfer Acc.	**0.76**	0.66	0.71

Table B.4. Quantitative evaluation of 3D scene analogy estimation in object groups with identical or different cardinalities. We report the Chamfer Accuracy at threshold 0.15 for manually collected scene pairs in 3D-FRONT [16]. Compared to the original metric reported in Table 1, our method shows consistent performance amidst object group cardinality variations.

B.4. Runtime Characteristics

We report the runtime for finding 3D scene analogies using neural contextual scene maps. As our method operates using sparse keypoints, affine and local displacement map estimation can quickly run on average 0.67s and 0.57s, respectively.

B.5. Robustness Evaluation on Object Groups with Different Cardinalities

Due to the outlier rejection explained in Section A.2 and Section 3.3, our method can robustly estimate scene analogies in scenarios where object group cardinalities differ. As shown in Figure B.1, our method can estimate scene analo-

Metric (Points Sampled per Object)	PCK (50)	PCK (100)	PCK (200)	PCK (400)
Scene Graph Matching	0.23	0.25	0.26	0.26
Multi-view Semantic Corresp.	0.09	0.09	0.10	0.10
Visual Feature Field	0.44	0.48	0.49	0.50
3D Point Feature Field	0.50	0.54	0.55	0.56
Ours	**0.68**	**0.73**	**0.75**	**0.76**

Table B.5. Quantitative comparison of scene analogies in the procedurally generated scene pairs from 3D-FRONT [16]. We measure percentage of correct points (PCP) at threshold 0.25 using varying number of points samples from the region of interest P_{RoI}. Compared to the PCK metric measured with 400 points sampled per object (which is mainly used for the experiments), our method performs stably amidst varying number of point samples.

gies for cases when (i) the RoI includes objects not present in the reference scene (Figure B.1 top) and (ii) the reference scene includes objects not present in the target scene (Figure B.1 bottom). We further report the accuracy of the estimated maps for scene pairs with object groups having identical / different cardinalities in Table B.4, where our method performs constantly in both cases.

B.6. Performance Analysis with Respect to the Number of RoI Points

As specified in Section C.4, we sample 400 points per each object in the RoI for estimating and evaluating scene analogies. In this section we evaluate map estimation performance with respect to the number of RoI points. As shown in Table B.5, our method constantly outperforms the baselines under RoI point variations. By holistically aligning descriptor fields using smooth maps, our method can attain robustness against individual point locations or point sampline rates and exhibit consistent performance.

B.7. Long Trajectory Transfer Comparison

In Figure B.2 and Figure B.3 we compare our method against the baselines in long trajectory transfer explained in Section 4.2. Recall to prevent collisions from directly applying scene maps on long trajectories, we proposed selectively mapping waypoints and interpolating the transferred waypoints via classical path planning (in our case the A* algorithm [17]). Note the scene maps are obtained by setting the objects near the waypoints as the region of interest. For baselines that only output object surface point matches (scene graph matching, multi-view semantic correspondence), we interpolate object surface matches from the baselines to open space using thin plate spline interpolation [7, 45] and find waypoint transfers. For cases where the A* algorithm fails to find a path due to inaccurate waypoint transfer, we directly use the interpolated map to transfer short trajectory fragments formed from the failed set of waypoints. A similar approach is taken for field alignment-based methods (visual feature field, 3D point feature field), while we skip the interpolation process as the output is already a continuous map.

As shown in Figures B.2 and B.3, our method can accurately place waypoints to the coherent location in the reference scene, resulting in long trajectory transfers respecting scene context. For example, our method can preserve the loop structure in `Scene 4` or the ribbon-like structure in `Scene 5` while placing all the waypoints at contextually similar locations. On the other hand, the baselines often fail to perform appropriate waypoint transfer, resulting in penetrations or misplacements of the transferred trajectory. Based on the descriptor field that distinguishes contextual information from geometry and semantics, our method can effectively handle waypoint transfers in various scenes.

B.8. Trajectory Transfer Using Multiple Regions of Interest

In this section we demonstrate the possibility of using our method for transferring trajectories by using multiple regions of interest. For long trajectories where a single scene analogy may be difficult to find, our method can instead transfer a sparse set of waypoints and use classical path planning [17] for interpolation. Given a target scene segmented into multiple RoIs as shown in Figure B.4, we set waypoints as sampled points in the input trajectory within each RoI. Note such coarse segmentations can be performed using scene graph clustering [21, 23] or vision language models [1, 10, 25].

We then find scene analogies for *multiple* RoIs and holistically align them. To account for symmetry ambiguities (e.g., table + 4 chair RoI in Figure B.4), for each RoI we have our method to output the top-5 maps with the smallest cost (Equation 6), which results in *combinations* of scene maps. Note we still apply the validity threshold ρ_{valid} explained in Section 3.3 to filter invalid mappings, which results in a relatively small number of mappings per RoI. Given a scene with N_{RoI} number of RoIs, this procedure results in at most $5^{N_{\text{RoI}}}$ possible *combinations* of mappings.

Next, we choose the optimal combination via a simple criterion based on isometry preservation [4, 30]. Here we take inspiration from prior works in 3D surface mapping [4, 30] that often impose isometry constraints such that the local geometric structure is preserved under non-rigid deformations. To elaborate, let $P_{\text{rand}} \in \mathbb{R}^{N_{\text{rand}} \times 3}$ be a set of randomly sampled points from the multiple regions of interest, and the distance matrix $D_{\text{rand}} \in \mathbb{R}^{N_{\text{rand}} \times N_{\text{rand}}}$ whose $(i,j)^{\text{th}}$ entry contains the euclidean distance between the i^{th} and j^{th} points. Similarly, for an arbitrary map combination, let $P_{\text{transform}}$ be the transformation result of P_{rand} under the map combination and $D_{\text{transform}}$ the distance matrix. The isometry cost is then defined as the Frobenius norm between the distance matrices, namely $\|D_{\text{rand}} - D_{\text{transform}}\|_F$. We aim

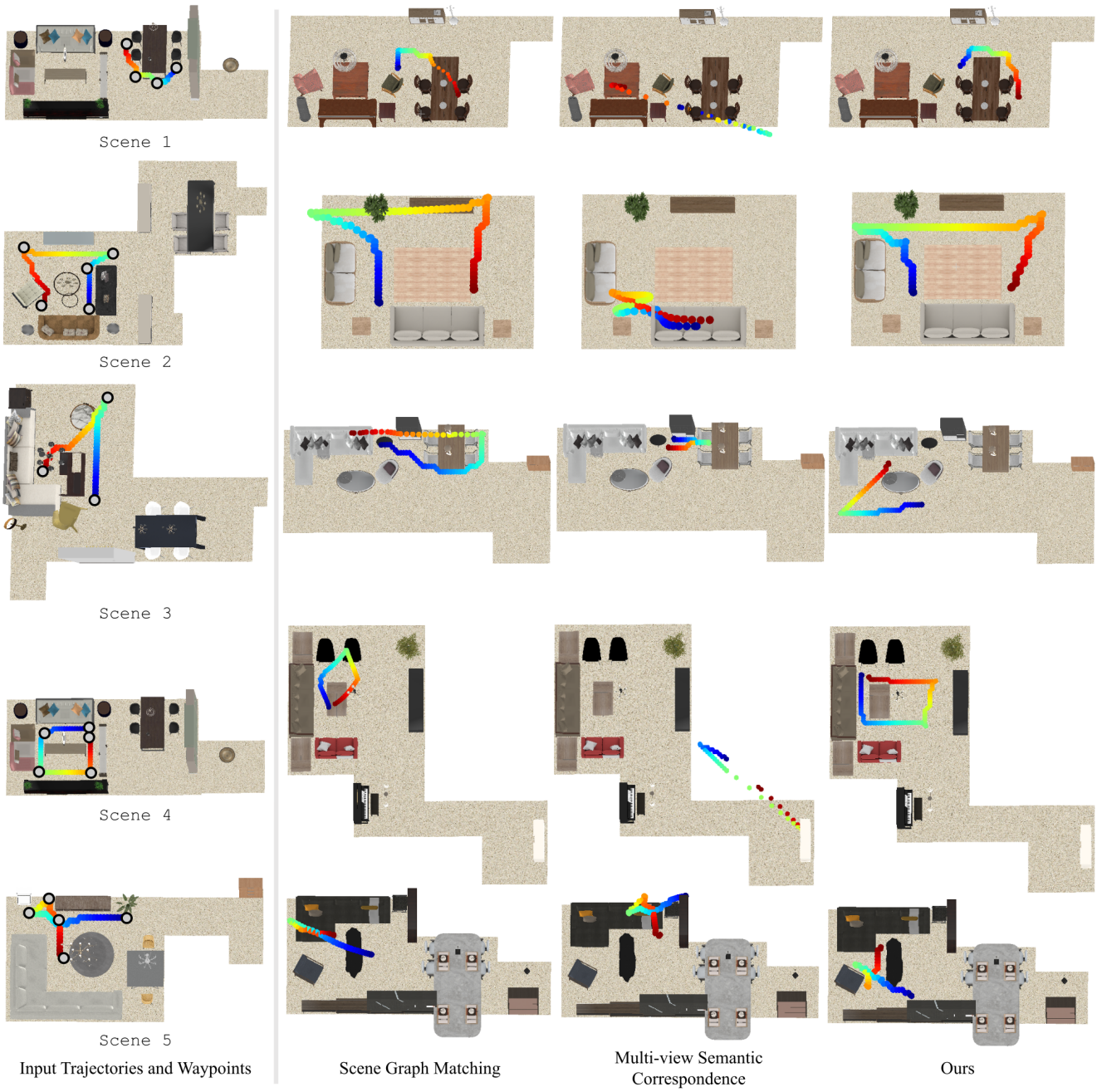


Figure B.2. Long trajectory transfer comparison against the scene graph matching and multi-view semantic correspondence baselines in 3D-FRONT [16].

to find the combination with a small isometry cost, where we employ a simple greedy approach. Given a randomly initialized combination, we sequentially update the map associated with each RoI to the one that produces a smaller isometry cost among the top-5 (or lower due to filtering) estimated maps. This process is repeated for a fixed number of iterations, and we use the final map combination to produce long trajectory transfers. While the search process is quite simple, we find this method to work well for scenes with a moderate number of RoI segments ($N_{\text{RoI}} < 5$).

Finally, we transfer each waypoint using the mapping found for the associated RoI, and interpolate between the transferred waypoints using classical path planning [17]. As shown in Figure B.4, the proposed method can align multiple scene analogies and produce a coherent long trajectory transfer spanning over the entire 3D scene. Nevertheless,

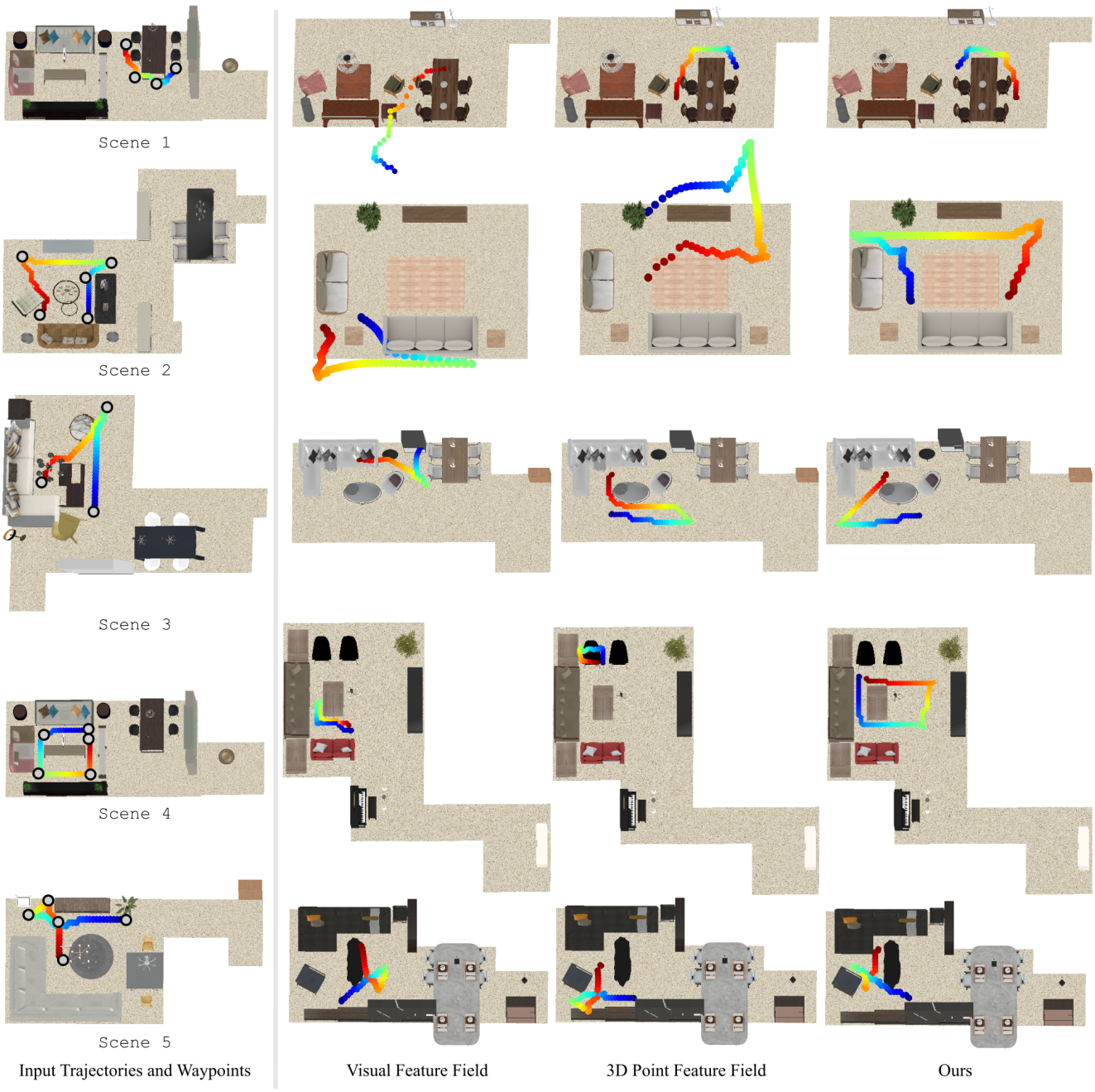


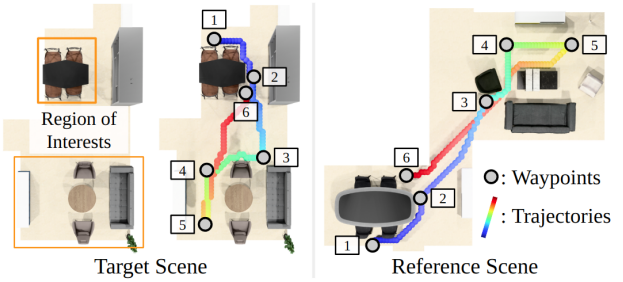
Figure B.3. Long trajectory transfer comparison against the field alignment-based baselines (visual feature field, 3D point feature field) in 3D-FRONT [16].

devising a more scalable and principled approach to align multiple scene maps originating from different RoIs is left as future work.

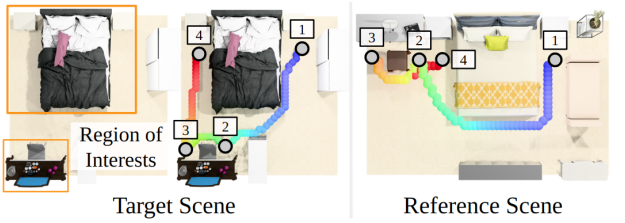
C. Experimental Setup Details

C.1. Baselines

In this section, we elaborate on the implementation details of the baselines compared against our method. As the 3D scene analogy task is new, we tailor existing 3D scene understanding pipelines to our task and introduce four baseline approaches capable of outputting dense scene maps.



(a) Living Room Scene Pairs



(b) Bedroom Scene Pairs

Figure B.4. Visualization of long trajectory transfer on 3D-FRONT [16] scene pairs using scene analogies from multiple regions of interest. We use the estimated maps to transfer waypoints, and apply traditional path planning [17] to obtain long trajectories spanning the entire 3D scene. We denote the waypoints as gray dots, and the estimated trajectories as color-coded spheres.

Figure C.5. Frontal view renderings of objects in 3D-FRONT [16], used for CLIP [35] and sentence embedding [36] feature extraction.

Scene Graph Matching The scene graph matching baseline builds a 3D scene graph [2] representing each object as nodes and finds affine transformations to align the graphs. First, we build scene graphs in a rule-based manner following Jia et al [20], where we use object bounding box intersections to determine scene graph edge types. Then, given a pair of 3D scene graphs $\mathcal{G}_{\text{tgt}}, \mathcal{G}_{\text{src}}$ for the target and reference scene, we list all subgraphs in $\mathcal{G}_{\text{ref}}$ and compare them against the subgraph containing the region of interest in $\mathcal{G}_{\text{tgt}}$. Here, we measure similarities between subgraphs using the Jaccard coefficient introduced by Wald et al [47]. After retrieving the closest subgraph in $\mathcal{G}_{\text{ref}}$ to the region of interest, we apply Hungarian matching [24, 52] between the subgraph nodes by using object semantic labels and adjacent edge labels as node features. Finally, similar to Sarkar et al. [39], we find an affine transformation from the node matches and deduce the final point-level alignment by performing iterative closest points (ICP) [3] separately for object point clouds associated with each node match.

Multi-view Semantic Correspondence The multi-view semantic correspondence baseline renders scenes at multiple views and operates based on 2D matches from vision foundation models [13]. To elaborate, we sample $N_{\text{render}} = 5$ views from virtual spheres encompassing S_{tgt} and S_{ref} [48, 54], and extract DINOv2 [29] features for each view. Then, we exhaustively match N_{render}^2 image pairs using the extracted features [13], and lift each 2D match to 3D via back-projection. Using the 3D matches, we obtain object-level matches by having each 3D match vote for an object pair. In this phase, for each object in the region of interest, the object in the reference scene with the largest amount of votes is selected. As the last step, we estimate affine transforms using the matched object centroids followed by iterative closest points (ICP) [3] to get the point-level alignments. Note that while it is possible to directly use the 3D matches and interpolate them to get point-level matches, we find the DINOv2 [29] descriptors to be quite noisy for obtaining fine-grained matches between object groups. Therefore, our baseline implementation mainly uses the features for object-level matching, which we empirically find to be more effective.

Visual Feature Field Instead of lifting 2D matches, the visual feature field baseline directly finds smooth scene maps by aligning vision foundation model features in 3D. The baseline first renders $N_{\text{render}} = 5$ views from virtual spheres encompassing the input scenes, and extracts DINOv2 [29] features. Next, the baseline computes multi-view aggregated features at each 3D keypoint in S_{tgt} and S_{ref}. Here, the method projects each 3D keypoint to the rendered views and extracts keypoint features via bilinear interpolation, and averages the N_{render} features. For an arbitrary query point, we compute features by using distance-weighted interpolation as in Wang et al. [41, 49]. In this step, we aggregate features by considering keypoints within radius r from the query point, where the radius values are set identical to our method. Finally, the baseline applies the coarse-to-fine map estimation from Section 3.3 to obtain scene analogies.

3D Point Feature Field Similar to the visual feature field baseline, the 3D point feature field interpolates keypoint features to obtain features at arbitrary locations, but uses 3D keypoint descriptors [11] instead of vision foundation models. For feature extraction, we use PointNet [31, 32] containing Vector Neuron layers [11] that is pre-trained on the ModelNet40 [50] dataset. Here we use the rotation-invariant embeddings obtained from the last layer of the Vector Neuron [11] encoder prior to max pooling. Given

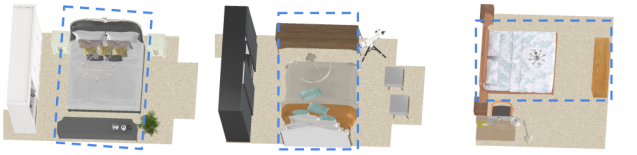


Figure C.6. Qualitative sample of scene pairs for evaluation. The blue box denotes the common object groups present both in the target and reference scenes.

the 3D keypoint features, we obtain features at arbitrary locations via distance-weighted interpolation [49], and align the keypoint-based fields using our coarse-to-fine estimation scheme.

C.2. Foundation Model Features for Ablation Study

In Section 4.1.2, we demonstrate scene analogy estimation using vision and language foundation model features, namely CLIP [35] and sentence embeddings [12, 36]. Here we elaborate on the details of the experiment. For CLIP feature extraction, we first render frontal views of 3D-FRONT [16] objects as shown in Figure C.5 and extract CLIP feature embeddings. The embeddings are then used in place of the semantic label embedding introduced in Section 3.2. For sentence embedding extraction, we first apply an off-the-shelf image captioning method on each of the object renderings in 3D-FRONT [16]. Then, we extract sentence embeddings for each of the image captions, and supply them as input to the descriptor fields in place of the semantic embeddings.

C.3. Scene Pair Preparation for Evaluation

We elaborate on the scene pair preparation process for evaluating scene analogies in Section 4.1. As shown in Figure C.6, we prepare two types of data, namely procedurally generated and manually collected scene pairs.

Procedurally Generated Scene Pairs Recall these scene pairs contain pseudo ground-truth annotations for evaluating point-level accuracy of scene analogy estimations. For each scene, we first randomly select an object and its k-nearest neighbors (where k is randomly sampled from $\{2, \ldots, 4\}$). The points sampled from the selected objects are used as the region of interest P_{RoI}. Then, for objects not selected, we either randomly remove them by a probability of 0.5 or apply pose perturbation. Here translation noise is sampled from the uniform distribution $\mathcal{U}(-0.05, 0.05)$ and rotation noise is obtained from the set of z-axis rotations with rotation angles sampled from $\mathcal{U}(-10°, 10°)$. Next, we randomly add N_{add} objects to open spaces in the scene (where N_{add} is sampled from $\mathcal{U}(2, 5)$). In this step, we retrieve the scene in the evaluation dataset with the closest object semantic label histogram, and select objects from that scene for addition. The objects are added by computing an occupancy grid map of the current scene and randomly choosing from collision-free locations [38]. Finally, we replace each object that has not been added or removed during the previous steps with a randomly selected object of the same semantic class, similar to training data generation explained in Section A.1. The resulting procedurally generated scenes contain realistic object placements while preserving meaningful object group structures for evaluation.

To compute pseudo ground-truth scene analogies for P_{RoI}, we uniformly sample points from the matching object group in the procedurally generated scene. Then, we apply Hungarian matching [24] between the two point sets, which yields an injective matching for each point in P_{RoI} to the sampled points in the generated scene. We use this matching result as the pseudo ground-truth for evaluation. Using the entire process, we generate 997 scene pairs for 3D-FRONT [16] and 549 scene pairs for ARKitScenes [6].

Manually Collected Scene Pairs In addition to the procedurally generated pairs, we manually collect scene pairs for evaluation. As obtaining point-level manual annotations is costly and possibly inaccurate, we only make group-level annotations for scene pairs. Specifically, for each scene pair sharing common object groups, we annotate the instance IDs of objects within the groups. We further annotate scene pairs not containing any common object groups, which we use for checking false positive scene analogies. We collect 120 scene pairs containing 20 pairs having no object group matches for both 3D-FRONT [16] and ARKitScenes [6].

C.4. Evaluation Metric Details

Percentage of Correct Points (PCP) and Bijectivity PCP The percentage of correct points (PCP) metric is measured for procedurally generated scene pairs having pseudo ground-truth annotations to evaluate point-level accuracy of scene maps, while the bijectivity PCP is a similar metric to measure whether the estimated maps are invertible. Both metrics are defined for points on the region of interest: namely, we sample 400 points from each object point cloud in the original scene using farthest point sampling (FPS) [14].

Chamfer Accuracy The Chamfer Accuracy metric evaluates the group-level accuracy of scene analogy predictions while penalizing false positive maps. Thus the metric additionally provides evaluation on the false positive rates of each method, i.e., whether the method falsely outputs mappings when the region of interest is unmatchable to the reference scene. In this section, we formally define the metric. We first define the Chamfer distance for a pair of point sets

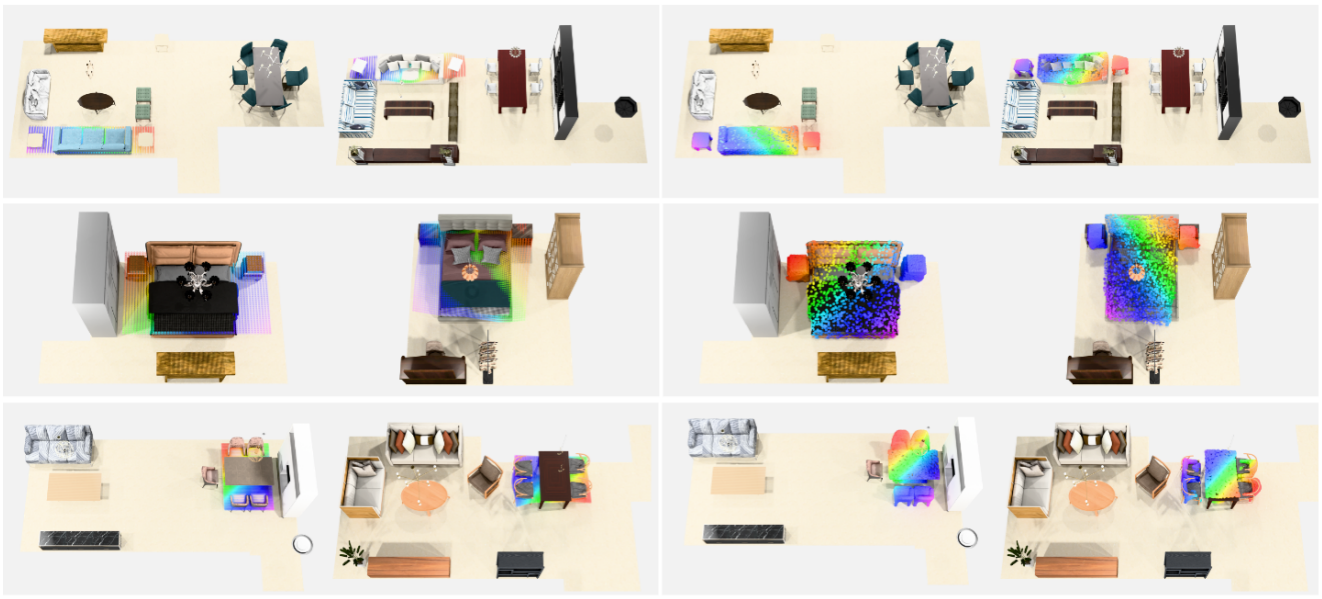


Figure C.7. Additional visualization of 3D scene analogies estimated in 3D-FRONT [16]. We show results both for near-surface and open-space points.

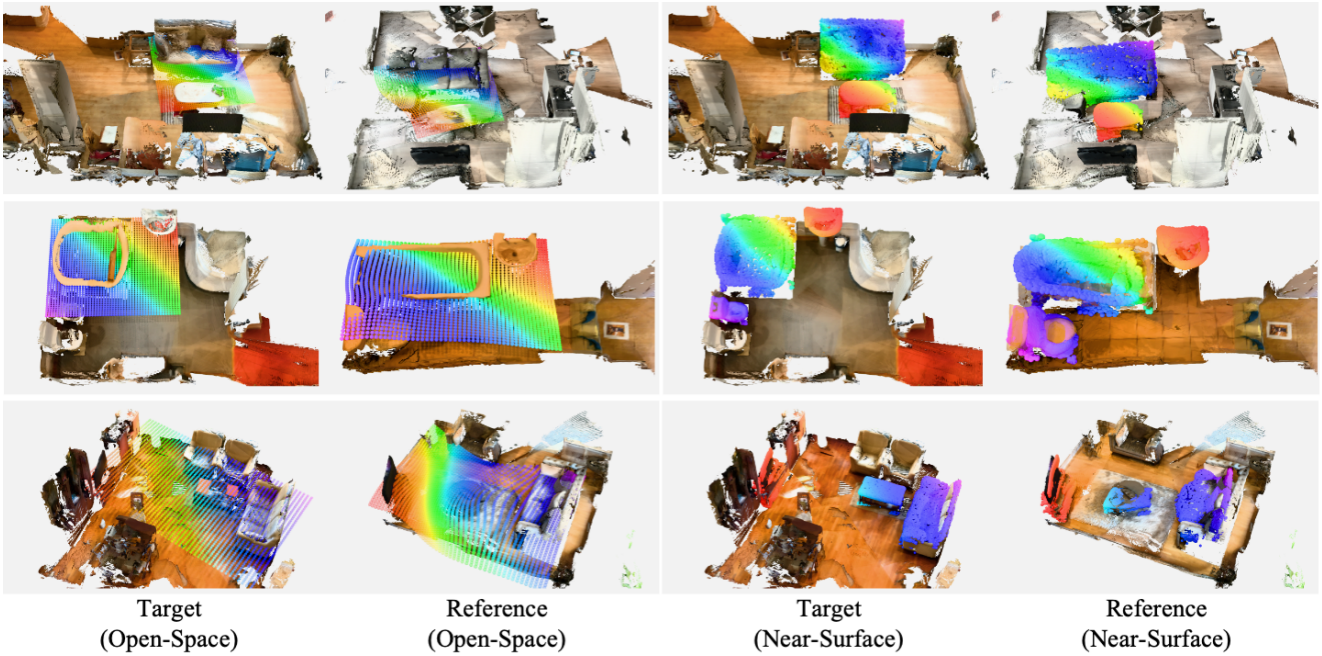


Figure C.8. Additional visualization of 3D scene analogies estimated in ARKitScenes [6]. We show results both for near-surface and open-space points.

$X, Y \in \mathbb{R}^3$ as follows

$$\text{CD}(X, Y) = \sum_{\mathbf{x} \in X} \min_{\mathbf{y} \in Y} \|\mathbf{x} - \mathbf{y}\|_2 + \sum_{\mathbf{y} \in Y} \min_{\mathbf{x} \in X} \|\mathbf{y} - \mathbf{x}\|_2. \tag{C.3}$$

Given the object set $\mathcal{O}_{\text{RoI}} = \{P_i^{\text{RoI}}\}$ in the region of interest, we perform nearest neighbor matching using object centroid locations to obtain corresponding objects in the reference scene $\mathcal{O}_{\text{match}} = \{P_i^{\text{match}}\}$. Recall that the region of

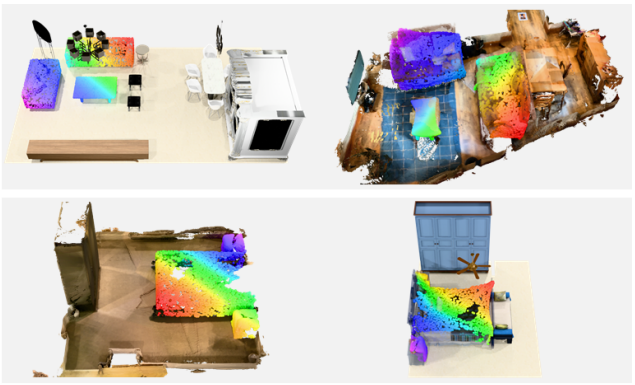


Figure C.9. Additional visualization of Sim2Real and Real2Sim scene analogies estimated between 3D-FRONT (Sim) and ARKitScenes (Real).

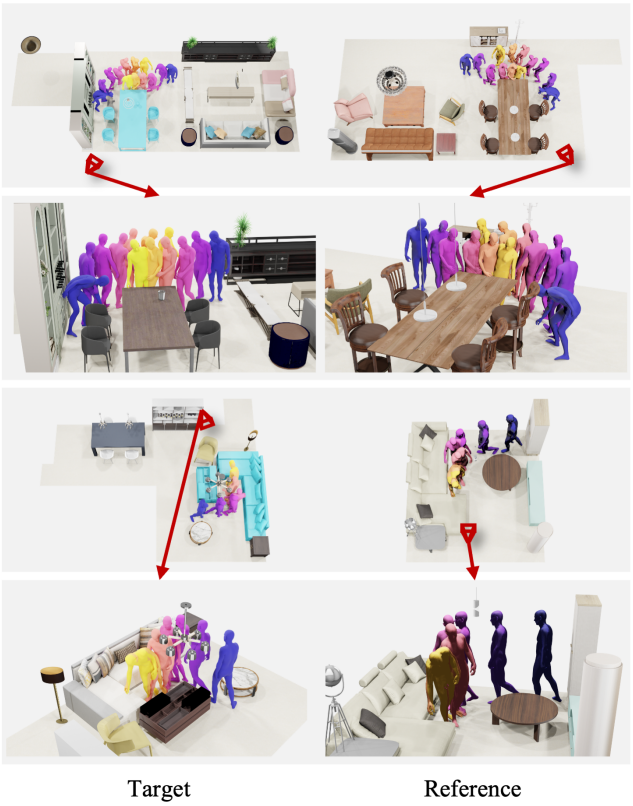


Figure C.10. Additional visualization of short trajectory transfer. We shade the region of interest used for estimating scene analogies in blue.

interest is defined as a union of object group points, namely $P_{\text{RoI}} = \bigcup_i P_i^{\text{RoI}}$. For scene pairs containing matchable regions, the Chamfer accuracy is then defined as follows,

Figure C.11. Additional visualization of object placement transfer. We shade the region of interest used for estimating scene analogies in blue.

$$\text{CA}(P_{\text{RoI}}) = \mathbb{1}[\frac{1}{|\mathcal{O}_{\text{RoI}}|}\sum_i \text{CD}(P_i^{\text{RoI}}, P_i^{\text{match}}) \leq \alpha], \quad \text{(C.4)}$$

where α is a threshold parameter. For scene pairs labeled as unmatchable, the Chamfer accuracy is set to 1 if no maps are produced, and 0 otherwise.

D. Limitations and Future Work

While our new task of finding 3D scene analogies holds practical applications for robotics and AR/VR, and our neural contextual scene maps can effectively find scene analogies, we acknowledge several limitations that invite further investigation in future work.

Handling Symmetries and Multi-modalities during Evaluation We observe reflection symmetries to exist quite often in object groups. For example, all object groups shown in Figure C.7 exhibit such symmetries. While these groups are symmetric *in isolation*, the ambiguities can be mitigated by leveraging the context from neighboring scene

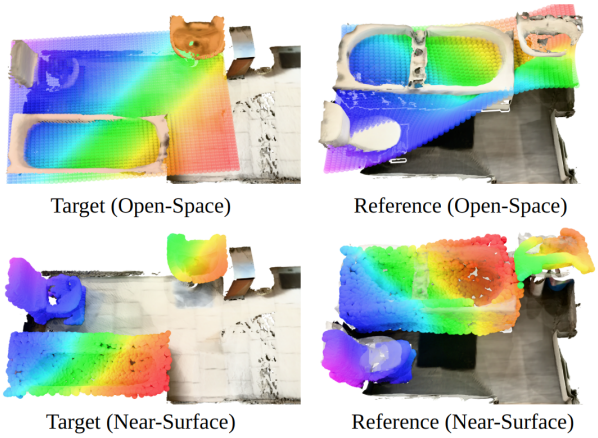


Figure D.12. Failure case of our method in scenes where the scene analogy cannot be initially approximated with affine maps, leaded to inaccurate estimations.

regions. To illustrate, the symmetric placements of the tables in the first row of Figure C.7 can be disambiguated by considering the neighboring objects: one table is next to another sofa, while the other table is next to a group of chairs. A similar argument can be made for the cabinet-and-bed group in the second row. Notice that our method correctly recognizes this contextual information and produces maps that respect nearby objects' information.

Nevertheless, there also exist object groups where such disambiguations are not effective: for the third row in Figure C.7, it is unclear from the neighboring scene contexts whether the currently estimated map is the only possible scene analogy. In this work we take a 'lenient' strategy for handling reflection symmetries: we additionally measure the PCP (percentage of correct points) metric for horizontal and vertical reflections, and report the smallest value. However, we believe symmetries can be better handled by additionally labeling the symmetry type of object groups, for example whether a group is reflection symmetric or can be disambiguated from nearby contexts, similar to how studies in object pose estimation [5, 46] evaluate symmetric objects. Obtaining additional labels and devising better symmetry-aware metrics are left as future work.

In addition to object-wise symmetries, ambiguities can arise in scenes containing higher-level symmetries, namely multiple similar object groups. While in most cases a single map can unambiguously match object groups, we acknowledge that there are scenarios where multiple scene analogies are detectable. For example, suppose one wants to find scene analogies between the target scene in Figure C.6 and a large room containing multiple bed-and-cabinet combinations. Although our neural contextual scene maps currently output a *single* mapping, it could be extended in such cases to output the top-K mappings as in Section B.8, which will lead to multiple scene analogy detections. Nevertheless, devising metrics and annotating scene pairs for multi-modal scene analogies is not straightforward, and thus is open to future work.

Infidelity of Affine Map Estimation Although the affine map estimation can effectively handle large, global transformations, we identified cases where the initial affine map estimation failed to find good solutions. These cases occur when scene analogies between two scenes cannot be approximated with an affine map. An example is shown in Figure D.12, where the relative locations of the toilet and bathtub are swapped, and thus affine maps are insufficient for aligning the scenes. Our method attempts to find an affine map that best aligns the two scenes, yet errors occur for regions near the sink (observe that the original points are incorrectly mapped to *flipped* regions in the reference scene). We expect a more flexible set of initializations, for example piece-wise affine transforms [15], can be used in place of the affine mapping procedure to solve such inaccuracies. Alternatively, finding multiple partial maps (e.g., separately mapping toilet-bathtub and toilet-sink groups for Figure D.12) and combinatorially aligning them as in Section B.8 could also be a feasible solution.

Scene Pair Generation for Training While training descriptor fields does not require densely labeled ground-truth data and descriptor fields can function without semantic labels during inference as demonstrated in Section 4.1.2, the training process still requires the generation of positive and negative scene pairs for contrastive learning [8, 9]. This process demands semantic and instance labels of 3D scenes, along with each object's pose. Although such information can be reliably extracted from modern 3D segmentation / pose estimation algorithms [33, 34, 40, 53], we posit our method to become more scalable if descriptor fields can be learned without exploiting any synthetic scene pairs. One interesting direction is to distill the knowledge of 3D scene generation methods [28, 51] trained on large amounts of indoor data for finding 3D scene analogies, similar to how image generation models [37] have been adapted to semantic correspondence tasks [42]. Finding more flexible learning strategies to train descriptor fields is left as future work.

Handling Various Notions of "Correct" Correspondences Inspired from prior works in semantic correspondence [19, 26, 27, 55], our work considers points having similar nearby object semantics and local geometry to be correct matches, and the descriptor fields are trained to support this notion of "correctness". We have demonstrated in Section 4.2 that this definition is useful for tasks such as trajectory transfer in robotics or object placement transfer in AR/VR. Nevertheless, we acknowledge that multiple

definitions of "correct" correspondences exist depending on the task. For example, one may want to find scene analogies based on other attributes such as affordance (e.g., mapping 'sittable' areas from one scene to another) or appearance (e.g., matching furniture groups with a specific style). Due to the modular design of our approach of separating descriptor extraction and map estimation based on classical optimization, our method can be flexibly modified to handle such definitions of correctness. Specifically, one may train new descriptor sets for different correctness definitions and subsequently apply the map estimation process that does not require training.

E. Additional Qualitative Results

We display additional qualitative results for 3D scene analogy estimation in 3D-FRONT [16] (Figure C.7), ARKitScenes [6] (Figure C.8), Sim2Real and Real2Sim (Figure C.9). Our method can produce accurate scene maps in all cases, due to the coarse-to-fine estimation framework which enhances robustness against input variations. We further show additional qualitative results for short trajectory transfer (Figure C.10) and object placement transfer (Figure C.11). The accurate scene analogy estimations can be effectively exploited for downstream tasks in robotics and AR/VR.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 3

[2] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R. Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 6

[3] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, PAMI-9 (5):698–700, 1987. 6

[4] Souhaib Attaiki and Maks Ovsjanikov. Understanding and improving features learned in deep functional maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[5] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Niessner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 10

[6] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Proceedings of the Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*, 2021. 2, 7, 8, 11

[7] F.L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 11(6):567–585, 1989. 1, 3

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. 10

[9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 10

[10] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Jen Dumas, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *ArXiv*, abs/1807.03748, 2024. 3

[11] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 6

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. 1, 7

[13] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 6

[14] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6:1305–15, 1997. 1, 7

[15] Oren Freifeld, Søren Hauberg, Kayhan Batmanghelich, and Jonn W. Fisher. Transformations based on continuous piecewise-affine velocity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12), 2017. 10

[16] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 4, 5, 6, 7, 8, 11

[17] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968. 3, 4, 6

[18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[19] Sangryul Jeon, Dongbo Min, Seungryong Kim, Jihwan Choe, and Kwanghoon Sohn. Guided semantic flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 10

[20] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *Proceedings of the European Conference on Computer Vision*, 2024. 6

[21] Seonji Kim, Dooyoung Kim, Jae-Eun Shin, and Woontack Woo. Object cluster registration of dissimilar rooms using geometric spatial affordance graph to generate shared virtual spaces. In *Proceedings of the IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, 2024. 3

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 1

[23] Sebastian Koch, Narunas Vaskevicius, Mirco Colosi, Pedro Hermosilla, and Timo Ropinski. Open3dsg: Open-vocabulary 3d scene graphs from point clouds with queryable objects and open-set relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[24] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52, 1955. 1, 6, 7

[25] Chengzu Li, Caiqi Zhang, Han Zhou, Nigel Collier, Anna Korhonen, and Ivan Vulić. TopViewRS: Vision-language models as top-view spatial reasoners. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1786–1807, Miami, Florida, USA, 2024. Association for Computational Linguistics. 3

[26] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. In *ICCV*, 2019. 10

[27] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *ArXiv*, abs/1908.10543, 2019. 10

[28] Başak Melis Öcal, Maxim Tatarchenko, Sezer Karaoglu, and Theo Gevers. Sceneteller: Language-to-3d scene generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 10

[29] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research (TMLR)*, 2023. 2, 6

[30] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics*, 31(4), 2012. 3

[31] C. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[32] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, Red Hook, NY, USA, 2017. Curran Associates Inc. 6

[33] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 10

[34] Xie Qian, Lai Yu-kun, Wu Jing, Wang Zhoutao, Zhang Yiming, Xu Kai, and Wang Jun. Mlcvnet: Multi-level context votenet for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 10

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021. 6, 7

[36] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 6, 7

[37] 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. 10

[38] André M. Santana, Kelson R.T. Aires, Rodrigo M.S. Veras, and Adelardo A.D. Medeiros. An approach for 2d visual occupancy grid map using monocular vision. *Electronic Notes in Theoretical Computer Science*, 281:175–191, 2011. 7

[39] Sayan Deb Sarkar, Ondrej Miksik, Marc Pollefeys, Daniel Barath, and Iro Armeni. Sgaligner : 3d scene alignment with

scene graphs. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 6

[40] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2023. 10

[41] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2023. 6

[42] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023. 10

[43] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 1

[45] Grace Wahba. Spline models for observational data. In *CBMS-NSF Regional Conference Series in Applied Mathematics*, 1990. 1, 3

[46] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. Rio: 3d object instance relocalization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 10

[47] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3d semantic scene graphs from 3d indoor reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[48] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Proceedings of the Conference on Neural Information Processing Systems*, 2021. 6

[49] Qianxu Wang, Haotong Zhang, Congyue Deng, Yang You, Hao Dong, Yixin Zhu, and Leonidas Guibas. SparseDFF: Sparse-view feature distillation for one-shot dexterous manipulation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 2, 6, 7

[50] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes . In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6

[51] Zhennan Wu, Yang Li, Han Yan, Taizhang Shang, Weixuan Sun, Senbo Wang, Ruikai Cui, Weizhe Liu, Hiroyuki Sato, Hongdong Li, and Pan Ji. Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. *ACM Transactions on Graphics (ToG)*, 43(4), 2024. 10

[52] Yaxu Xie, Alain Pagani, and Didier Stricker. Sg-pgm: Partial graph matching network with semantic geometric fusion for 3d scene graph alignment and its downstream tasks. *arXiv preprint arXiv:2403.19474*, 2024. 6

[53] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2023. 10

[54] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 6

[55] 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 (CVPR)*, 2024. 10