XSemanticFlow: Cross Object Semantic Alignment for Zero-shot Manipulation

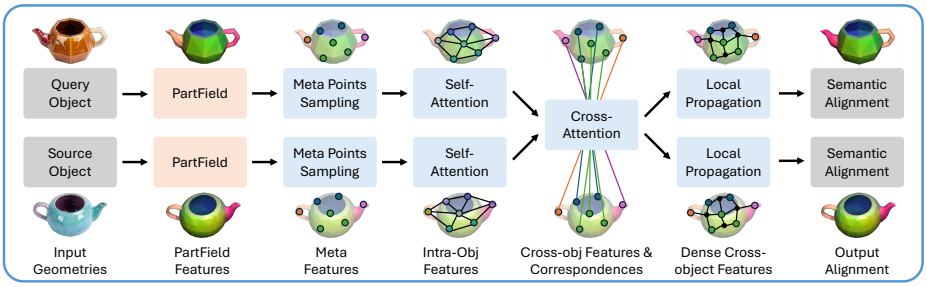
Junyu Nan[1,2], Brian Okorn[2], Kris Kitani[1], and Noam Eshed[2]

[1] Carnegie Mellon University, Pittsburgh PA 15222, USA
{jnan1,kmkitani}@andrew.cmu.edu
[2] Robotics and AI Institute, Cambridge 02142, USA
{jnan,neshed,bokorn}@rai-inst.com

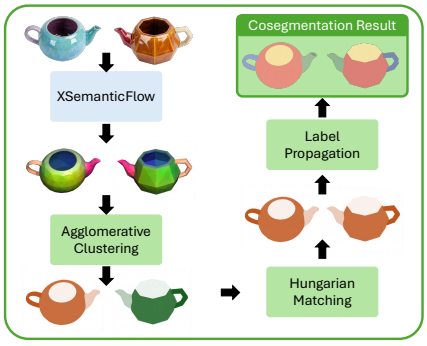
Abstract. Establishing reliable correspondences across diverse object instances is fundamental to robust 3D understanding and generalizable manipulation. However, existing methods are often inconsistent across geometries or vulnerable to non-canonicalized poses, making correspondence transfer unreliable. To address this, we propose XSemanticFlow, which learns correspondences from pure semantic features. Given per-object semantic feature fields, XSemanticFlow applies hierarchical intra-object self-attention and inter-object cross-attention to predict soft correspondence maps. By restricting cross-attention strictly to semantic tokens, XSemanticFlow avoids geometric overfitting, enabling it to predict generalizable correspondences across different topologies and produce aligned feature fields. This enables XSemanticFlow to learn effectively through self-supervised cross-pose alignment on large-scale unlabeled shapes, followed by supervised cross-instance finetuning on a cosegmentation dataset containing labeled object pairs from PartNet. We evaluate our method on both 3D understanding and manipulation tasks. For 3D understanding, XSemanticFlow improves self-segmentation with $SE(3)$ augmentations by +19.1 mIoU and +20.6 accuracy, and cross-instance cosegmentation by +10.9 mIoU and +12.5 accuracy over baselines, demonstrating stronger transform consistency and semantic alignment. For manipulation, XSemanticFlow provides a reliable alternative to recent video-generation-based pipelines for zero-shot manipulation. Rather than relying on synthesized visual plans, we directly transfer contact regions and end-effector trajectories from a single reference demonstration to novel instances, bypassing the brittle video generation failures common in contact-critical tasks.

1 Introduction

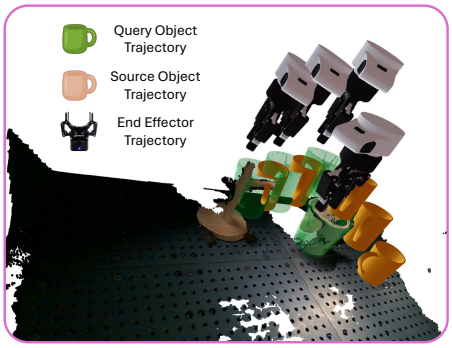
Establishing reliable correspondences across diverse 3D object instances is a foundational capability for both visual understanding and robotic manipulation. Given two objects that differ in shape, scale, and topology, a system that can reliably map every surface point on one to its functional counterpart on the other unlocks a wide range of downstream tasks: transferring part labels without per-instance annotation and remapping entire manipulation demonstrations onto



(a) XSemanticFlow to align different instances by semantics.



(b) Cosegmentation.



(c) Zero-shot Manipulation Remapping.

Fig. 1: XSemanticFlow for semantic correspondences and manipulation transfer. (a) Given a source and a query object, XSemanticFlow takes PartField [16] features and performs hierarchical self-/cross-attention in semantic space to produce aligned dense features and soft correspondences. (b) Cross-instance cosegmentation via label transfer using the aligned field. (c) Zero-shot demonstration remapping: correspondences transfer contact points and end-effector trajectories from a single reference demonstration to new object instances.

novel objects without retraining a policy [1, 20, 31, 32]. In this sense, correspondence prediction serves as a general-purpose primitive. Given accurate predicted correspondence field, many category-level perception and control problems reduce to simple correspondence lookups in an aligned feature space.

Recent advances in vision foundation models have made promising progress toward this goal. 2D foundation models [3, 4, 11, 19, 24, 25, 27] provide powerful open-vocabulary semantic descriptors that have been widely used for correspondence bootstrapping. In 3D, methods [16, 17, 26, 36] learn part-aware embeddings to capture semantics in 3D structures. However, these existing feature matching methods are often inconsistent across diverse 3D geometries and highly vulnerable to non-canonicalized object poses. Because these embeddings are not explicitly forced to align across instances, the same functional object part can lead to embeddings which occupy drastically different regions in the feature space on different objects, making semantic correspondence transfer unreliable.

To address this, we propose XSemanticFlow, which learns correspondences between object instances from purely semantic features. Given per-object seman-

tic feature fields, XSemanticFlow applies hierarchical intra-object self-attention to reason about local part structures, and inter-object cross-attention to predict soft correspondence maps between different geometries. The key design choice in our matching module is that it uses only semantic tokens, restricting the cross-attention layers from accessing absolute geometric coordinates. By strictly limiting cross-attention to semantic features, XSemanticFlow avoids geometric overfitting during large-scale pre-training. This formulation forces the correspondences to follow functional semantics rather than spatial patterns, enabling the model to predict generalizable soft correspondence maps across different topologies and produce aligned feature fields via weighted feature propagation.

To effectively train this architecture, we propose a three-stage training recipe using both large-scale unannotated shapes and labeled data. First, we fine-tune the backbone feature extractor for $SE(3)$ invariance. Next, we use large-scale unlabeled shapes from Objaverse-v1 [5] to perform self-supervised cross-pose alignment via random $SE(3)$ transforms and a spatial correspondence loss. Finally, we perform supervised cross-instance finetuning on PartNet-Coseg, a cosegmentation dataset containing labeled object pairs from PartNet [18]. Because exact point-to-point mappings are undefined across distinct geometries, this final stage drives cross-instance semantic alignment using canonical pose spatial targets and a triplet relative-consistency loss.

We evaluate our method on both 3D understanding and physical manipulation tasks. For 3D understanding, we measure $SE(3)$ consistency via self-segmentation on PartNet-Coseg, where XSemanticFlow improves mIoU by +19.1 points and Accuracy by +20.6 points over [16], demonstrating stronger transform-consistent representations. We also evaluate semantic consistency across instances through cosegmentation on PartNet-Coseg, where XSemanticFlow improves CoSeg-mIoU by +10.9 points and CoSeg-Acc by +12.5 points, showing the learned field aligns reliably across diverse geometries. For manipulation, XSemanticFlow enables zero-shot demonstration remapping across physical tasks, including threading a mug onto a rack, watering a plant, and precisely pouring granular blocks. By transferring contact regions and end-effector trajectories from a single reference demonstration to novel instances via contact-aware spatial alignment, we avoid the need for per-target demonstration video generation. This approach overcomes the common bottleneck of video generation quality in video-based zero-shot planning, resulting in higher execution success rates on contact-critical tasks compared to state-of-the-art zero-shot baselines [7,13]. The main contributions of this paper are as follows:

- We propose XSemanticFlow, which learns correspondences across instances using purely semantic hierarchical cross-attention to avoid geometric overfitting and generalize across topologies.
- We introduce a three-stage training recipe that combines foundational $SE(3)$ fine-tuning, self-supervised cross-pose alignment on unlabeled shapes, and supervised cross-instance alignment using canonical pose spatial targets.

- On self-segmentation, XSemanticFlow improves mIoU by +19.1 points and accuracy by +20.6 points over 3D segmentation baselines [16,26,36], demonstrating strong $SE(3)$ transform consistency.
- On cross-instance cosegmentation, XSemanticFlow improves mIoU by +10.9 points and accuracy by +12.5 points over baselines [16, 26, 36], enabling reliable label transfer for 3D understanding.
- We apply the learned field to zero-shot demonstration remapping, transferring a single reference manipulation trajectory to novel object instances. This contact-aware spatial alignment improves upon video-model-based zero-shot baselines [7, 13] by avoiding video generation failures.

2 Related Works

Correspondences through feature matching. Traditional 3D segmentation works [18, 21, 22, 34, 35, 38] target closed-vocabulary 3D understanding by predicting semantic part labels with supervised 3D segmentation networks. These methods primarily output discrete labels and typically require category-specific supervision, and therefore do not directly provide a reliable correspondence field that generalizes across unseen geometries and poses. Recently, advances in vision-language foundation models enables open-vocabulary semantic descriptors extractions for matching and alignment [3, 4, 11, 19, 24, 25, 27]. While these representations provide powerful semantics and enable correspondence bootstrapping in 2D, they do not explicitly enforce 3D consistencies. Features can drift under viewpoint, scale, or occlusion, and lifting 2D descriptors into 3D often produces fragmented, instance-dependent fields. PartField [16] improves 3D consistency of semantic featuring by distilling multi-view 2D foundation-model part proposals and available 3D part annotations into a continuous, queryable 3D feature field, optimized with a triplet-based contrastive objective that encourages points from the same proposed part to embed closely while separating others. Nevertheless, [16] does not explicitly enforce alignment of embeddings across different object instances or non-canonicalized poses, which limits its direct application in cross-object reasoning and manipulation. In contrast, our method learns to align two 3D feature fields directly; it propagates semantics across objects using semantic-only cross-attention and trains with cross-pose and cross-object objectives, yielding semantic correspondences that generalize across geometry and pose variations.

Zero-shot manipulation. Motivated by recent advances in large language models and vision-language models [15, 23, 30], there is growing interest in solving manipulation through zero-shot generalization. A prominent direction trains vision-language-action (VLA) policies [8, 10, 12, 29] on large collections of robot trajectories to directly map observations and language into actions. While powerful, the scalability of VLAs is often limited by the cost and coverage of high-quality physical interaction data compared to web-scale supervision. An alternative line leverages VLMs as high-level planners that generate symbolic programs or spatial plans rather than continuous motor commands [6, 9, 14, 37],

improving task decomposition and generalization but frequently struggling with fine-grained 3D localization and geometric reasoning required for precise contact.

Recent advances [7,13] improve zero-shot manipulation success rate by introducing explicit geometric grounding from generated videos. [13] extracts actionable 3D object flow from generated task-solving videos using off-the-shelf perception modules, then translates the flow into executable robot actions via grasp proposals and trajectory optimization. [7] extends video-based planning to long-horizon settings with a closed-loop VLM planner that monitors execution and triggers re-planning, while grounding low-level actions through tracked object keypoints and extracted human hand poses from generated rollouts to maintain stable execution under occlusion and depth noise. Complementary to these geometry-grounded video planning systems, our method introduces semantic reasoning and cross-instance alignment that better solves task configurations which are challenging for video generation. Given a single high-quality reference demonstration video at a reference configuration, we can use semantically aligned correspondences to remap robot execution trajectories to novel object instances and configurations.

3 Method

Given two 3D geometries $\mathcal{M}_A$ and $\mathcal{M}_B$ from the same semantic category, our goal is to learn correspondences that generalize across instances. We build upon the PartField backbone [16] and introduce XSemanticFlow, a hierarchical transformer that produces aligned feature fields by explicitly decoupling geometric coordinates from the correspondence prediction. The resulting dense field can then be directly leveraged towards downstream cosegmentation and zero-shot manipulation tasks.

3.1 Network Architecture

Given $\mathcal{M}_A$ and $\mathcal{M}_B$, PartField [16] triplane features are first extracted. This representation allows querying a semantic feature at any coordinate via bilinear interpolation. To process the geometry efficiently, XSemanticFlow predicts correspondences by transforming the unaligned backbone features into a shared subspace through a sequence of purely semantic operations. The network consists of the following modules, illustrated in Figure 1(a):

1. **Meta-Point Sampling and Local Feature Aggregation.** We first sample a set of representative surface meta-points. For each meta-point, we perform local attention over its K nearest neighbors in Euclidean space. However, to prevent geometric overfitting, the aggregation attention weights over these neighbors are computed using only their semantic features from [16]. Euclidean neighbors only define local neighborhood support within a single object, and coordinates never enter the inter-object cross-attention.

2. **Self-Attention for Intra-Object Aggregation.** To reason about intra-object structural dependencies, the aggregated meta-point tokens undergo standard multi-head self-attention. This operation is performed globally among all meta-points belonging to the same object. Consistent with our semantic-only design principle, this block applies no positional encoding, forcing the network to construct the object's internal layout purely through semantic relationships.
3. **Cross-Attention for Inter-Object Propagation.** To establish correspondence across different geometries, we apply cross-attention between the meta-points of the distinct objects. The inter-object attention weights are computed by matching the self-attended semantic queries of one object against the keys of the other. Because this mechanism relies solely on semantic similarity, it is completely blind to absolute geometric coordinates.
4. **Sparse Correspondence Prediction.** Beyond aligning features, the inter-object cross-attention block naturally yields a sparse meta-to-meta correspondence matrix $\mathcal{W}_{AB}$ (and $\mathcal{W}_{BA}$) from the attention weights. This weight matrix explicitly captures the soft assignments between the meta-points of the two objects and is used directly to compute the spatial correspondence loss during training. Degenerate many-to-one assignments in $\mathcal{W}_{AB}$ are discouraged jointly by $\mathcal{L}_{corr}$, which requires each query point to recover its spatial target, and the triplet loss, which preserves pairwise feature distances.
5. **Dense Feature Propagation.** Finally, the cross-updated meta-point semantics are propagated back to the dense mesh vertices. For each dense point, we identify its K-nearest meta-points in Euclidean space. We then perform a localized, semantic-only attention step where the dense point's semantic feature acts as the query, and the neighboring meta-points provide the keys and values. A final residual MLP decodes these attended features to refine the original backbone features into a fully aligned dense feature space. Because the resulting dense fields share a consistent semantic space, full-resolution point-to-point correspondences can be established downstream via simple nearest-neighbor feature matching.

3.2 Training Procedure

Learning generalizable semantic correspondences requires the predicted field to be stable under rigid pose changes of the same object and semantically consistent across different objects within a category. A major challenge in achieving this is the severe scarcity of fine-grained 3D part annotations [18], compared to large scale unlabeled 3D geometry data [5]. Furthermore, defining exact point-to-point ground truth mappings between structurally distinct instances is fundamentally ill-posed.

To overcome this data bottleneck, we optimize XSemanticFlow using a three-stage training pipeline designed to maximize the utility of both abundant unlabeled 3D assets and scarce labeled datasets. Our core intuition is that the general mechanism of robust correspondence can be learned entirely via self-supervision: by applying synthetic $SE(3)$ transformations to unlabeled shapes,

we generate infinite training pairs with perfect spatial ground truth. Because our cross-attention module is restricted to semantic tokens, this self-supervised phase prevents geometric memorization and forces the network to learn a generalizable matching function. Once this robust matching capability is established, we require only a small set of labeled cross-instance pairs to explicitly align the semantic feature spaces across different geometries without relying on strict point-to-point supervision.

To preserve part-level relational structures across transformations and instances, we introduce a *Triplet Relative Consistency Loss* applied during both self-supervised and supervised training stages. Rather than constraining absolute feature values, this objective aligns intra-object feature distributions before and after feature propagation to the same object going through an $SE(3)$ transform. For a sampled triplet of points (p_1, p_2, p_3) on an object, we compute their pairwise feature distances (e.g., cosine distance) in the raw feature space F, yielding a distance vector $d^F = [d^F_{12}, d^F_{13}, d^F_{23}]$. Features are ℓ_2-normalized before the dot product, so each similarity is bounded in $[-1, 1]$ and the softmax is numerically stable. We convert the distance vector into a probability distribution via softmax with temperature τ ($\tau = 0.01$), $P_F = \text{softmax}(d^F/\tau)$. We identically compute the distribution $P_{\hat{F}}$ using the propagated features $\hat{F}$ over transformed object. We then penalize the symmetric Kullback-Leibler divergence between these relational distributions across a set of randomly sampled triplets T:

$$\mathcal{L}_{triplet} = \frac{1}{|T|} \sum_{t \in T} \frac{1}{2} \Big(D_{KL}(P_{\hat{F}} \parallel P_F) + D_{KL}(P_F \parallel P_{\hat{F}}) \Big) \tag{1}$$

This ensures that the relative semantic distances between points on an object remain proportionally consistent before and after transformation.

The complete three stage training pipeline is as follows:

1. **Backbone $SE(3)$ Finetuning.** Before training the correspondence module, we first fine-tune the frozen PartField [16] backbone to achieve foundational $SE(3)$ invariance. This ensures that the base semantic features extracted from the input point clouds are robust to arbitrary rigid transformations before they are passed into the cross-attention layers.
2. **Self-Supervised Training on Objaverse.** In the second stage, we freeze the fine-tuned PartField [16] backbone and train the XSemanticFlow modules using self-supervision on Objaverse-v1 [5]. For each unannotated mesh, we generate two views by applying independent random $SE(3)$ transforms. For a sampled point set $\mathbf{Q}_A$, we compute its transformed counterpart $\mathbf{Q}_{A,gt}$ in the second view. We train the network using a spatial *Correspondence Loss*:
$$\mathcal{L}_{corr} = \left\| \text{bmm}(\mathcal{W}_{AB}, \mathbf{Q}_B) - \mathbf{Q}_{A,gt} \right\|^2. \tag{2}$$
Because the exact geometry is identical across the two views, we also apply a *Feature Loss* minimizing the absolute distance between the propagated features $\hat{F}$ and the ground-truth backbone features F:
$$\mathcal{L}_{feat} = \|\hat{F}_A - F_A\|^2. \tag{3}$$

Additionally, the *Triplet Relative Consistency Loss* ($\mathcal{L}_{triplet}$) is applied to ensure structural preservation within the identical geometry.

3. **Mixed Finetuning on Objaverse [5] and PartNet-Coseg.** To ensure the correspondences generalize across geometrically distinct instances, we fine-tune XSemanticFlow using a mixture of self-supervised pairs from Objaverse-v1 and supervised cross-object pairs from PartNet-Coseg. Crucially, for cross-object pairs, exact per-point geometric correspondence is undefined, meaning the direct feature $\mathcal{L}_{feat}$ cannot be used and is strictly masked out. Instead, we rely entirely on canonical-space $\mathcal{L}_{corr}$, with ground truth obtained by transforming query points into PartNet's shared canonical space, and the distribution-based $\mathcal{L}_{triplet}$ to drive cross-instance semantic alignment without forcing absolute feature values to match.

3.3 Downstream Application: Cosegmentation.

For 3D understanding, XSemanticFlow enables cross-instance cosegmentation as illustrated in Figure 1(b). To produce these discrete part labels, we first perform agglomerative clustering on the target object's features. Following [16], we incorporate a face adjacency matrix during this step to ensure that the resulting clusters strictly respect the physical manifold of the mesh. Final part labels are then globally assigned by matching the source object's labeled parts to the target object's clusters using the Hungarian algorithm, driven by the aligned feature similarities provided by our semantic correspondence fields.

3.4 Downstream Application: Zero-shot Demonstration Remapping.

For physical manipulation, XSemanticFlow enables zero-shot demonstration remapping as illustrated in Figure 1(c). By transferring contact regions and end-effector trajectories from a single reference demonstration to novel instances, we avoid the brittle video generation failures common in video-model-based zero-shot manipulation methods [7,13]. We utilize the semantic correspondences predicted by XSemanticFlow to remap and execute a contact-aware $SE(3)$ trajectory alignment. The remapping pipeline contains following steps:

1. **Scene Object Reconstruction.** We first identify and reconstruct both the manipulated object (e.g., a mug) and the stationary anchor object (e.g., a mug stand) in the demonstration and target scenes. We utilize open-vocabulary segmentation [2] and mesh reconstruction [28], followed by a pose estimator [33] to refine the 6-DoF mesh registration.
2. **Anchor Object Alignment.** To ground the demonstration in the target environment, we query XSemanticFlow to find the semantic correspondence between the demonstration and target anchor objects. This yields a relative $SE(3)$ transformation that structurally aligns the two scenes. Applying this transformation shifts and orients the entire base demonstration trajectory into the target scene's coordinate frame.

3. **Manipulated Object Rotational Alignment.** To ensure the manipulated object is oriented correctly for the task, we query XSemanticFlow between the source object at its goal state and the target object. This provides a correspondence field from which we derive a relative semantic rotation $R_{semantic}$ that aligns the target object's functional parts with the source's goal orientation.
4. **Contact Detection and Translation Refinement.** By tracking the source object throughout the demonstration, we identify the earliest *contact frame* where it approaches the anchor within a strict distance threshold. Using XSemanticFlow, we map these source contact vertices to their semantic equivalents on the target mesh. We apply the rotational offset $R_{semantic}$ to the target object first, and then compute a precise translation offset t_{delta} that perfectly aligns the centroid of the rotationally-aligned target contact region with the source's contact region in the physical scene.
5. **Grasp and Trajectory Warping.** Finally, the robot's execution trajectory is synthesized in two distinct phases, separated by a contact alignment point selected as the closest point in the demonstration trajectory to the starting position of the manipulated object in the target scene. In the *approach portion*, the trajectory smoothly interpolates to correct the combined $SE(3)$ offset—both the semantic rotation $R_{semantic}$ and the subsequent positional translation t_{delta}. In the *replay portion*, the robot follows the anchor-shifted and contact-aligned demonstration trajectory to complete the interaction. The initial grasp is correspondingly remapped to the semantically equivalent target vertices.

4 Experiments

We evaluate XSemanticFlow to answer two primary questions: (1) Does semantic-only cross-attention yield a feature field that is both robust to rigid transformations and numerically aligned across geometrically distinct objects? (2) Can these learned correspondences serve as a reliable geometric primitive for the zero-shot transfer of complex robotic manipulation tasks? To answer the first question, we evaluate same-instance $SE(3)$ consistency and cross-instance label transfer on the PartNet-Coseg dataset (Sec. 4.1). To address the second, we deploy XSemanticFlow in a zero-shot demonstration remapping pipeline, comparing its physical execution success against state-of-the-art video-based planning baselines (Sec. 4.2).

4.1 Semantic Correspondence for 3D Cosegmentation

In this section, we evaluate the quality of the predicted semantic-based correspondences to test whether the learned field is both transform-consistent and semantically aligned across distinct instances.

Dataset and Evaluation Protocols. We construct PartNet-Coseg from PartNet [18] by sampling instance pairs within the same semantic category that

Table 1: Semantic correspondence evaluation measured via self-segmentation and cosegmentation on PartNet-Coseg. We report per-point accuracy (%) and mean IoU (%) for same-instance self-segmentation under $SE(3)$ augmentations (PartNet-SE3) and cross-instance label transfer (PartNet-Coseg). The bottom block reports ablations on XSemanticFlow.

	PartNet-SE3 (Self-Segmentation)		PartNet-Coseg (CoSegmentation)	
Method	Acc ↑	mIoU ↑	Acc ↑	mIoU ↑
PartField [16]	55.55	43.41	61.23	52.43
SAMesh [26]	46.17	41.55	56.11	45.52
SAMPart3D [36]	50.15	40.80	58.92	48.40
XSemanticFlow (ours)	**76.18**	**62.47**	**73.75**	**63.32**
w/o rot-invariant	72.51	61.62	68.86	58.74
w/o PartNet-Coseg	72.27	61.03	71.59	61.81
+ geo features	75.10	61.86	71.69	62.21

share a matching set of part labels. We use the train split for supervision, and the validation and test splits for evaluation. We evaluate correspondence quality under two protocols: *same-object* $SE(3)$ *consistency* (PartNet-SE3) via self-segmentation, and *cross-object cosegmentation* (PartNet-Coseg) via label transfer between paired instances as discussed in Section 3.3. For both protocols, the input consists of a source point cloud with ground-truth part labels and a target point cloud. The output is the predicted part label assignment on the target. We evaluate these transferred predictions against the ground-truth target labels using two standard metrics: per-point accuracy, which measures the overall percentage of correctly classified points, and mean Intersection over Union (mIoU), which averages the overlap between predicted and ground-truth regions across all present part categories.

Baselines. We compare XSemanticFlow against PartField [16], SAMesh [26], and SAMPart3D [36]. For each method, we obtain 3D part segments on both instances from the baseline methods, and then compute a per-part feature by averaging the method's per-point/vertex descriptors within each segment. We then match parts across instances with the Hungarian algorithm using cosine distance between per-part features, and transfer source part labels to the matched target parts. Since no baseline directly emits dense cross-instance correspondences, this shared Hungarian protocol is applied uniformly to all methods, including ours, for fairness; a variant propagating labels through XSemanticFlow's own dense correspondences is reported in the supplementary material.

Implementation Details. For all shapes, we sample surface points uniformly from the mesh and construct a set of meta-points for feature propagation. All models are optimized using AdamW. Following the three-stage procedure detailed in Sec. 3, we first fine-tune the PartField [16] backbone for 50 epochs to learn $SE(3)$ invariance. Next, we freeze the backbone and train XSemanticFlow using self-supervision on Objaverse-v1 for 150 epochs. Finally, we fine-tune XSemanticFlow on a mixed batch of Objaverse-v1 and the PartNet-Coseg training split for 50 epochs.

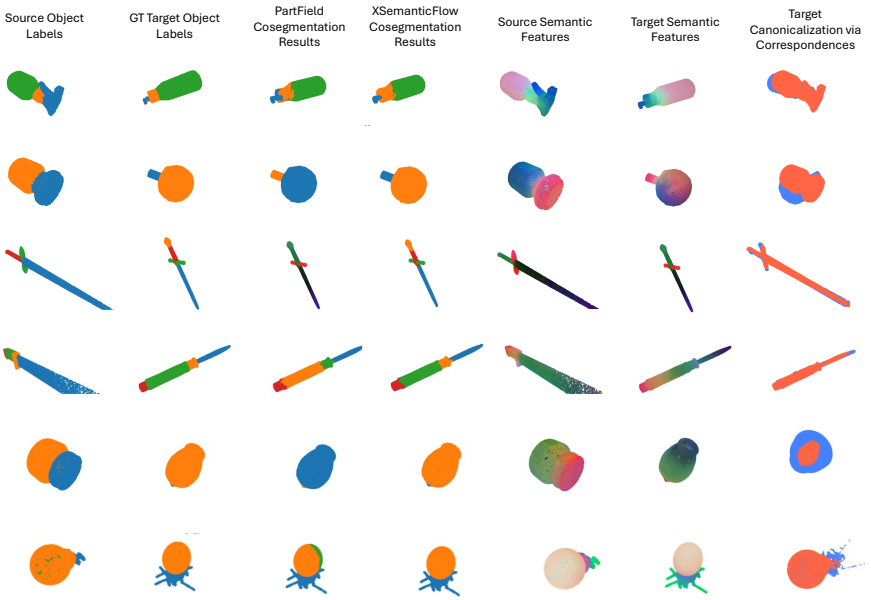


Fig. 2: Cross-instance cosegmentation via aligned semantic fields. Given a source instance with part labels, XSemanticFlow transfers labels to a geometrically distinct target via nearest-neighbor lookup in the aligned feature space.

Results and Analysis. Table 1 summarizes the performance across both protocols. On PartNet-SE3, XSemanticFlow yields a massive improvement over unaligned 3D semantic features from [16, 26, 36], increasing SelfSeg-mIoU from 43.41% to 62.47% (+19.06 points) and SelfSeg-Acc from 55.55% to 76.18% (+20.63 points). On PartNet-Coseg, XSemanticFlow improves CoSeg-mIoU from 52.43% to 63.32% (+10.89 points) and CoSeg-Acc from 61.23% to 73.75% (+12.52 points). Results from PartNet-SE3 suggests that raw PartField embeddings are highly sensitive to rigid transformations, whereas XSemanticFlow learns substantially more transform-consistent representations through semantic-only matching and spatial anchoring. Figure 2 visualizes this cross-instance label transfer using the learned correspondences, showing that our method is both quantitatively and qualitatively better than [16].

Ablations. We ablate key design choices that enable generalizable semantic correspondences, reporting mIoU on both PartNet-SE3 and PartNet-Coseg (Table 1, bottom block):

1. *w/o rotational invariant pre-training*: more rotationally robust PartField [16] features are both helpful for self-segmentation and for cross-instance semantic alignment.
2. *w/o* PartNet-Coseg *fine-tuning*: supervised cross-object alignment improves correspondence quality beyond self-supervised training alone, with its removal dropping cross-instance CoSeg-mIoU from 63.32 to 61.81.
3. *+ geometric features in cross-attention*: injecting positional and geometric signals into the attention block consistently degrades correspondence gen-

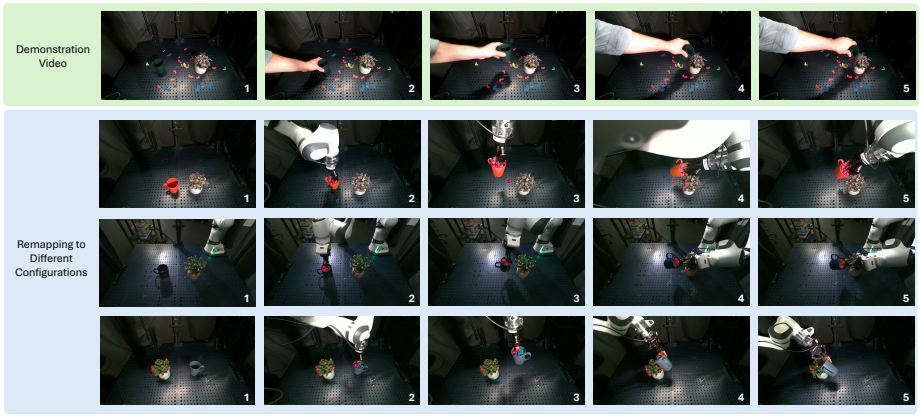


Fig. 3: Zero-shot trajectory remapping on Mug Hanging and Block Pouring tasks. (a) Experiments on *Mug Hanging* task. (b) Experiments on *Block Pouring* task; which is equivalent to *Water Plant* except the blocks as physical verification.

eralization across both tasks. The degradation is modest rather than catastrophic since the model still receives strong semantic features.

In summary, our ablation study validates the core architectural and training decisions of XSemanticFlow. The empirical results demonstrate that relying on purely semantic cross-attention prevents geometric overfitting, while rotation-invariant pre-training is crucial for handling arbitrary object poses. Together with supervised cross-instance fine-tuning, these components synergize to yield the most robust correspondence transfer across diverse geometries and topologies. We provide more ablations over training losses and self-supervised scaling in the supplement.

4.2 Zero-Shot Demonstration Remapping

To evaluate the functional utility of XSemanticFlow, we transfer a single successful manipulation demonstration on a source object to geometrically distinct target instances without retraining a policy or regenerating demonstration videos for each new object.

Baselines and Tasks. We compare against two state-of-the-art video-based zero-shot manipulation frameworks, NovaFlow [13] and NovaPlan [7], since they represent the strongest prior art that grounds robot actions from generated rollouts rather than requiring task-specific policy training. We evaluate zero-shot demonstration remapping on three tasks with increasing physical strictness:

1. *Mug Hanging*: thread a mug handle onto a rack, requiring precise contact geometry. [13] originally generated demonstration videos using both start and goal images, which violates the standard zero-shot setting. Our evaluation protocol instead enforces a stricter zero-shot constraint. We provide a single reference demonstration generated using both images; however, for all subsequent evaluation trials on novel instances, neither our method nor the baselines are given access to goal images.

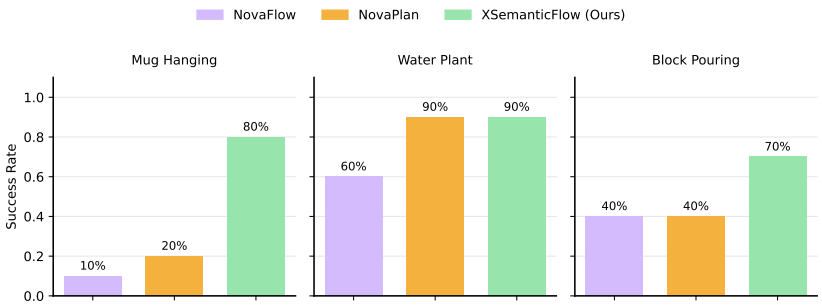


Fig. 4: Zero-Shot Demonstration Remapping Success Rates. We compare the physical execution success of XSemanticFlow against video-based zero-shot manipulation baselines (NovaPlan and NovaFlow) across three tasks. While baselines perform adequately on the lenient *Water Plant* metric, their performance drops significantly on tasks requiring precise spatial grounding. In *Mug Hanging* and the strict *Block Pouring* physical verification task, XSemanticFlow drastically outperforms the baselines by leveraging accurate contact-aware spatial alignment derived from our learned semantic correspondence field.

2. *Water Plant*: grasp a mug and tilt it over a plant. For hardware safety, no real liquid is used. Execution is counted as a success if the robot correctly positions and tilts the vessel directly over the target plant.
3. *Block Pouring*: a strict physical verification variant of *Water Plant* where the vessel is filled with granular blocks. For an execution to be counted as a success, at least one block must make physical contact with the plant.

Results and Analysis. On the *Water Plant* metric, all methods perform well, with NovaFlow [13] reaching 60% success and both NovaPlan [7] and XSemanticFlow reaching 90%. In contrast, *Block Pouring* is substantially harder because success requires physically grounded pouring as granular blocks must land in the pot. Small pose or contact errors in trajectory that pass *Water Plant* may cause failure in *Block Pouring*. Under this strict criterion, the baselines are vulnerable to the quality of generated videos. Both NovaFlow [13] and NovaPlan [7] achieve a 40% success rate while XSemanticFlow achieves 70% success rate. Similarly, in *Mug Hanging*, where video generation tends to fail without goal images, NovaFlow [13] and NovaPlan [7] drop to 10% and 20% success respectively, while XSemanticFlow transfers a single high-quality reference demonstration via correspondence-based remapping to achieve 80% success.

4.3 Failure Case Analysis

Figure 5 illustrates major failure modes of XSemanticFlow. Figure 5(a)(i) illustrates where feature alignment fails when instances within the same nominal class have different sets of semantic labels. Differences in sets of labels usually induce ambiguous or missing correspondences, and forcing one-to-one matching can lead to incorrect assignments or collapsed clusters. Figure 5(a)(ii) shows

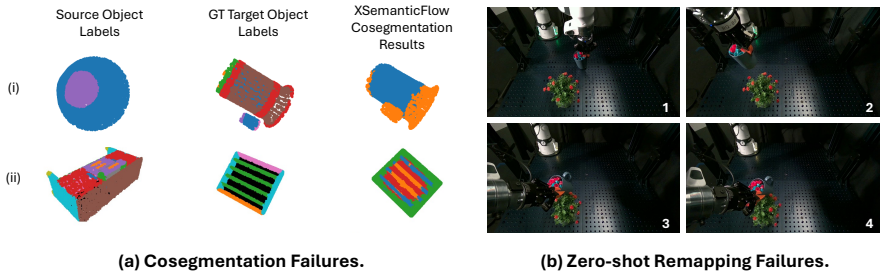


Fig. 5: Failure cases of XSemanticFlow. (a) Cosegmentation tends to fail on objects with mismatching label sets. (b) Trajectory remapping failure where the robot fails to perform the planned pouring action due to joint limit.

another failure of cosegmentation by XSemanticFlow due to geometric symmetries. In these cases, multiple regions are equally plausible matches, such as the vertical and horizontal panels on the shelf become indistinguishable after an $SE(3)$ transform. On the manipulation side, Figure 5(b) illustrates a failure where the robot cannot execute the full planned trajectory due to a collision with the camera pillar during the pouring action. XSemanticFlow currently does not support remapping with optimization against obstacles, and we leave it as future work. Also, since we target rigid objects, our pipeline collapses the dense correspondences into a single rigid $SE(3)$ transform; exploiting the full dense field for non-rigid manipulation is left to future work. Figure 3 illustrates the demonstration video and the execution snapshots by XSemanticFlow in different configurations.

5 Conclusion

We introduced XSemanticFlow, a method for learning generalizable dense correspondences from semantic features across 3D object instances. By explicitly decoupling geometric coordinates from cross-attention, XSemanticFlow prevents positional shortcuts and forces correspondence discovery to rely on latent semantic signatures. A three-stage training procedure—foundational $SE(3)$ backbone finetuning, self-supervised cross-pose training on large-scale unlabeled data, and mixed fine-tuning using canonical-pose spatial targets and a triplet consistency loss—yields dense features that are both pose-invariant and numerically consistent across geometrically diverse instances.

We validated the learned correspondences on two complementary downstream tasks. On cosegmentation (PartNet-Coseg), XSemanticFlow improves accuracy and mIoU over unaligned feature fields and coordinate-aware baselines. On zero-shot demonstration remapping—which subsumes both grasp transfer and trajectory warping—a single manipulation demonstration is successfully transferred to geometrically distinct instances within a category. Together, these results demonstrate that dense, semantically grounded correspondences serve as a general-purpose primitive for category-level 3D perception and manipulation.

References

1. Cai, E., Donca, O., Eisner, B., Held, D.: Non-rigid relative placement through 3d dense diffusion (2024), https://arxiv.org/abs/2410.19247
2. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., Lei, J., Ma, T., Guo, B., Kalla, A., Marks, M., Greer, J., Wang, M., Sun, P., Rädle, R., Afouras, T., Mavroudi, E., Xu, K., Wu, T.H., Zhou, Y., Momeni, L., Hazra, R., Ding, S., Vaze, S., Porcher, F., Li, F., Li, S., Kamath, A., Cheng, H.K., Dollár, P., Ravi, N., Saenko, K., Zhang, P., Feichtenhofer, C.: Sam 3: Segment anything with concepts (2025), https://arxiv.org/abs/2511.16719
3. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers (2021), https://arxiv.org/abs/2104.14294
4. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers (2024), https://arxiv.org/abs/2309.16588
5. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects (2022), https://arxiv.org/abs/2212.08051
6. Du, Y., Yang, M., Florence, P., Xia, F., Wahid, A., Ichter, B., Sermanet, P., Yu, T., Abbeel, P., Tenenbaum, J.B., Kaelbling, L., Zeng, A., Tompson, J.: Video language planning (2023), https://arxiv.org/abs/2310.10625
7. Fu, J., Nan, J., Sun, L., Li, H., Qian, J., Barry, J.L., Kitani, K., Konidaris, G.: Novaplan: Zero-shot long-horizon manipulation via closed-loop video language planning (2026), https://arxiv.org/abs/2602.20119
8. Ghosh, D., Walke, H.R., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., Luo, J., Tan, Y.L., Chen, L.Y., Vuong, Q., Xiao, T., Sanketi, P.R., Sadigh, D., Finn, C., Levine, S.: Octo: An Open-Source Generalist Robot Policy. In: Robotics: Science and Systems (RSS). vol. 20 (Jul 2024)
9. Huang, W., Wang, C., Li, Y., Zhang, R., Fei-Fei, L.: ReKep: Spatio-Temporal Reasoning of Relational Keypoint Constraints for Robotic Manipulation. In: Conference on Robot Learning (CoRL) (Jan 2025)
10. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Galliker, M.Y., Ghosh, D., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., LeBlanc, D., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Ren, A.Z., Shi, L.X., Smith, L., Springenberg, J.T., Stachowicz, K., Tanner, J., Vuong, Q., Walke, H., Walling, A., Wang, H., Yu, L., Zhilinsky, U.: $\pi_{0.5}$: a vision-language-action model with open-world generalization (2025), https://arxiv.org/abs/2504.16054
11. Jose, C., Moutakanni, T., Kang, D., Baldassarre, F., Darcet, T., Xu, H., Li, D., Szafraniec, M., Ramamonjisoa, M., Oquab, M., Siméoni, O., Vo, H.V., Labatut, P., Bojanowski, P.: Dinov2 meets text: A unified framework for image- and pixel-level vision-language alignment (2024)
12. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E.P., Sanketi, P.R., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: OpenVLA: An Open-Source Vision-Language-Action Model. In: Conference on Robot Learning (CoRL) (Jan 2025)
13. Li, H., Sun, L., Hu, Y., Ta, D., Barry, J., Konidaris, G., Fu, J.: Novaflow: Zero-shot manipulation via actionable flow from generated videos (2025), https://arxiv.org/abs/2510.08568

14. Liu, F., Fang, K., Abbeel, P., Levine, S.: MOKA: Open-Vocabulary Robotic Manipulation through Mark-Based Visual Prompting. In: First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024 (Apr 2024)
15. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning (2023), https://arxiv.org/abs/2304.08485
16. Liu, M., Uy, M.A., Xiang, D., Su, H., Fidler, S., Sharp, N., Gao, J.: Partfield: Learning 3d feature fields for part segmentation and beyond (2025), https://arxiv.org/abs/2504.11451
17. Liu, M., Zhu, Y., Cai, H., Han, S., Ling, Z., Porikli, F., Su, H.: Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models (2023), https://arxiv.org/abs/2212.01558
18. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding (2018), https://arxiv.org/abs/1812.02713
19. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.Y., Li, S.W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2024), https://arxiv.org/abs/2304.07193
20. Pan, C., Okorn, B., Zhang, H., Eisner, B., Held, D.: TAX-pose: Task-specific cross-pose estimation for robot manipulation. In: 6th Annual Conference on Robot Learning (2022), https://arxiv.org/abs/2211.09325
21. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation (2017), https://arxiv.org/abs/1612.00593
22. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space (2017), https://arxiv.org/abs/1706.02413
23. Qwen, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z.: Qwen2.5 Technical Report (Jan 2025). https://doi.org/10.48550/arXiv.2412.15115
24. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision (2021), https://arxiv.org/abs/2103.00020
25. Stracke, N., Baumann, S.A., Bauer, K., Fundel, F., Ommer, B.: Cleandift: Diffusion features without noise (2025), https://arxiv.org/abs/2412.03439
26. Tang, G., Zhao, W., Ford, L., Benhaim, D., Zhang, P.: Segment any mesh (2025), https://arxiv.org/abs/2408.13679
27. Tang, L., Jia, M., Wang, Q., Phoo, C.P., Hariharan, B.: Emergent correspondence from image diffusion (2023), https://arxiv.org/abs/2306.03881
28. Team, S.D., Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., Lin, A., Liu, J., Ma, Z., Sagar, A., Song, B., Wang, X., Yang, J., Zhang, B., Dollár, P., Gkioxari, G., Feiszli, M., Malik, J.: Sam 3d: 3dfy anything in images (2025), https://arxiv.org/abs/2511.16624
29. Team, T.L., Barreiros, J., Beaulieu, A., Bhat, A., Cory, R., Cousineau, E., Dai, H., Fang, C.H., Hashimoto, K., Irshad, M.Z., Itkina, M., Kuppuswamy, N., Lee, K.H., Liu, K., McConachie, D., McMahon, I., Nishimura, H., Phillips-Grafflin, C.,

Richter, C., Shah, P., Srinivasan, K., Wulfe, B., Xu, C., Zhang, M., Alspach, A., Angeles, M., Arora, K., Guizilini, V.C., Castro, A., Chen, D., Chu, T.S., Creasey, S., Curtis, S., Denitto, R., Dixon, E., Dusel, E., Ferreira, M., Goncalves, A., Gould, G., Guoy, D., Gupta, S., Han, X., Hatch, K., Hathaway, B., Henry, A., Hochsztein, H., Horgan, P., Iwase, S., Jackson, D., Karamcheti, S., Keh, S., Masterjohn, J., Mercat, J., Miller, P., Mitiguy, P., Nguyen, T., Nimmer, J., Noguchi, Y., Ong, R., Onol, A., Pfannenstiehl, O., Poyner, R., Rocha, L.P.M., Richardson, G., Rodriguez, C., Seale, D., Sherman, M., Smith-Jones, M., Tago, D., Tokmakov, P., Tran, M., Hoorick, B.V., Vasiljevic, I., Zakharov, S., Zolotas, M., Ambrus, R., Fetzer-Borelli, K., Burchfiel, B., Kress-Gazit, H., Feng, S., Ford, S., Tedrake, R.: A Careful Examination of Large Behavior Models for Multitask Dexterous Manipulation (Jul 2025). https://doi.org/10.48550/arXiv.2507.05331

30. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: LLaMA: Open and Efficient Foundation Language Models (Feb 2023). https://doi.org/10.48550/arXiv.2302.13971
31. Wang, J., Donca, O., Held, D.: Learning distributional demonstration spaces for task-specific cross-pose estimation (2024), https://arxiv.org/abs/2405.04609
32. Wen, B., Lian, W., Bekris, K.E., Schaal, S.: You only demonstrate once: Category-level manipulation from single visual demonstration. CoRR **abs/2201.12716** (2022), https://arxiv.org/abs/2201.12716
33. Wen, B., Yang, W., Kautz, J., Birchfield, S.: Foundationpose: Unified 6d pose estimation and tracking of novel objects (2024), https://arxiv.org/abs/2312.08344
34. Wu, X., Jiang, L., Wang, P.S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H.: Point transformer v3: Simpler, faster, stronger (2024), https://arxiv.org/abs/2312.10035
35. Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H.: Point transformer v2: Grouped vector attention and partition-based pooling (2022), https://arxiv.org/abs/2210.05666
36. Yang, Y., Huang, Y., Guo, Y.C., Lu, L., Wu, X., Lam, E.Y., Cao, Y.P., Liu, X.: Sampart3d: Segment any part in 3d objects (2024), https://arxiv.org/abs/2411.07184
37. Yin, G., Li, Y., Wang, Y., McConachie, D., Shah, P., Hashimoto, K., Zhang, H., Liu, K., Li, Y.: CodeDiffuser: Attention-Enhanced Diffusion Policy via VLM-Generated Code for Instruction Ambiguity. In: Robotics: Science and Systems (RSS) XXI. arXiv (Jun 2025). https://doi.org/10.48550/arXiv.2506.16652
38. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16259–16268 (October 2021)