# GEAL: Generalizable 3D Affordance Learning with Cross-Modal Consistency

Supplementary Material

## Table of Contents



## A. Corrupt Data Benchmark

The robustness of models under real-world corruptions is a critical challenge in 3D point cloud analysis and 3D affordance learning [4, 5, 14, 17]. Unlike other 3D representations, point clouds often face various distortions caused by sensor inaccuracies, environmental complexities, and post-processing artifacts, which significantly impact downstream tasks [6, 7, 18]. For 3D affordance learning, ensuring robustness is paramount, as affordances are highly sensitive to object geometry and spatial details.

### A.1. Corruption & Severity Level Settings

To standardize evaluation, we introduce a taxonomy of **seven atomic corruption types** – *Scale*, *Jitter*, *Rotate*, *Drop Global*, *Drop Local*, *Add Global*, *Add Local* – each simulating distinct real-world perturbations. These atomic corruptions simplify complex scenarios into controllable factors, enabling systematic analysis across **five levels of severity**. By providing a unified framework for benchmarking, we facilitate consistent and comprehensive assessment of model robustness, setting the stage for more resilient 3D affordance learning methods.

Below, we detail the construction methodology for each corruption type:

- **Jitter**
  - *Description*: Adds Gaussian noise to perturb each point's X, Y, and Z coordinates.
  - *Mathematical Formulation*: For each point, a noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is added independently to X, Y, and Z.
  - *Severity Levels*: The standard deviation $\sigma$ varies as:

$$\sigma \in \{0.01, 0.02, 0.03, 0.04, 0.05\}.$$

- **Scale**
  - *Description*: Applies random scaling independently to the X, Y, and Z axes.
  - *Mathematical Formulation*: Each axis is scaled by a factor $s \sim \mathcal{U}\left(\frac{1}{S}, S\right)$, where $S$ determines the range of scaling.
  - *Severity Levels*: The range of $S$ is:

$$S \in \{1.6, 1.7, 1.8, 1.9, 2.0\}.$$

    After scaling, the point cloud is re-normalized to fit within a unit sphere.
- **Rotate**
  - *Description*: Introduces random rotation to the point cloud.
  - *Mathematical Formulation*: The rotation is specified by Euler angles $(\alpha, \beta, \gamma)$, where:

$$\alpha, \beta, \gamma \sim \mathcal{U}(-\theta, \theta).$$

  - *Severity Levels*: The angle range $\theta$ is:

$$\theta \in \{\pi/30, \pi/15, \pi/10, \pi/7.5, \pi/6\}.$$

    This approach does not guarantee uniform sampling in $\mathrm{SO}(3)$, but provides sufficient variation to simulate diverse rotations.
- **Drop Global**
  - *Description*: Randomly removes a percentage of points from the point cloud.
  - *Method*: Shuffle all points and drop the last $N \cdot \rho$ points, where $N = 2048$ is the total number of points.
  - *Severity Levels*: The proportion $\rho$ is:

$$\rho \in \{0.25, 0.375, 0.5, 0.675, 0.75\}.$$

- **Drop Local**
  - *Description*: Removes points in clusters around randomly selected local regions.

  - *Method*:
    1. Randomly choose the number of local regions $C \sim \mathcal{U}\{1, 8\}$.
    2. For each region $i$:
       * Randomly select a local center.
       * Assign a cluster size $N_i$.
       * Drop the $N_i$-nearest neighbor points to the center.
    3. Repeat for $C$ regions.
  - *Severity Levels*: The total number of points to drop $K$ is:
$$K \in \{100, 200, 300, 400, 500\}.$$

- **Add Global**
  - *Description*: Uniformly samples additional points inside a unit sphere and appends them to the point cloud. The added points are treated as noise and assigned a label of 0.
  - *Method*: Sample $K$ random points within a unit sphere.
  - *Severity Levels*: The total number of added points $K$ is:
$$K \in \{10, 20, 30, 40, 50\}.$$

- **Add Local**
  - *Description*: Adds clusters of points around randomly selected local regions. The added points are treated as noise and assigned a label of 0.
  - *Method*:
    1. Shuffle points and select $C \sim \mathcal{U}\{1, 8\}$ as the number of local centers.
    2. For each center $i$:
       * Define a cluster size $N_i$.
       * Generate neighboring points' coordinates from:
$$\mathcal{N}(\mu_i, \sigma_i^2 I),$$
         where $\mu_i$ is the $i$-th local center, and $\sigma_i \sim \mathcal{U}(0.075, 0.125)$.
    3. Append generated points to the cloud one cluster at a time.
  - *Severity Levels*: The total number of added points $K$ is:
$$K \in \{100, 200, 300, 400, 500\}.$$

## A.2. The PIAD-C Dataset

Our proposed PIAD-C dataset is constructed from the test set of the **Seen** partition in PIAD [22], specifically designed to evaluate the robustness of affordance detection models under various corruption scenarios. This dataset includes a total of $2,474$ object-affordance pairings, representing 17 affordance categories and 23 object categories, and with $1,012$ distinct clean object shapes. Comprehensive statistics, detailing object categories, their corresponding affordance categories, and the number of object-affordance pairings, are presented in Tab. A. We include additional visualization examples for the PIAD-C dataset in Fig. A.

Table A. Detailed statistics of the proposed **PIAD-C** dataset, showing the object categories, their corresponding affordance types, and the number of object-affordance pairings for each category.

| # | Object Category | Affordance Type | Data |
|---|---|---|---|
| 1 | **Earphone** | listen, grasp | 70 |
| 2 | **Bag** | contain, open, grasp, lift | 50 |
| 3 | **Chair** | move, support, sit | 587 |
| 4 | **Refrigerator** | contain, open | 53 |
| 5 | **Knife** | stab, cut, grasp | 138 |
| 6 | **Dishwasher** | contain, open | 39 |
| 7 | **Keyboard** | press | 25 |
| 8 | **Scissors** | stab, cut, grasp | 29 |
| 9 | **Table** | move, support | 194 |
| 10 | **StorageFurniture** | contain, open | 92 |
| 11 | **Bottle** | contain, wrap_grasp, open, grasp, pour | 273 |
| 12 | **Bowl** | contain, wrap-grasp, pour | 83 |
| 13 | **Microwave** | contain, open | 47 |
| 14 | **Display** | display | 52 |
| 15 | **TrashCan** | contain, open, pour | 69 |
| 16 | **Hat** | wear, grasp | 66 |
| 17 | **Clock** | display | 9 |
| 18 | **Door** | open, push | 47 |
| 19 | **Mug** | contain, wrap_grasp, grasp, pour | 126 |
| 20 | **Faucet** | open, grasp | 95 |
| 21 | **Vase** | contain, wrap-grasp, pour | 134 |
| 22 | **Laptop** | press, display | 112 |
| 23 | **Bed** | lay, support, sit | 84 |
| **Total** | **23 Categories** | **17 Affordance Types** | 2474 |

Table B. Detailed statistics of the proposed **LASO-C** dataset, showing the object categories, their corresponding affordance types, and the number of distinct objects for each category.

| # | Object Category | Affordance | Data |
|---|---|---|---|
| 1 | **Door** | open, push, pull | 35 |
| 2 | **Clock** | display | 34 |
| 3 | **Dishwasher** | open, contain | 20 |
| 4 | **Earphone** | listen, grasp | 28 |
| 5 | **Vase** | contain, pour, wrap-grasp | 167 |
| 6 | **Knife** | stab, grasp, cut | 59 |
| 7 | **Bowl** | contain, pour, wrap_grasp | 36 |
| 8 | **Bag** | open, contain, lift, grasp | 25 |
| 9 | **Faucet** | open, grasp | 80 |
| 10 | **Scissors** | stab, grasp, cut | 11 |
| 11 | **Display** | display | 58 |
| 12 | **Chair** | sit, support, move | 858 |
| 13 | **Bottle** | grasp, wrap_grasp, open, contain, pour | 122 |
| 14 | **Microwave** | open, contain | 23 |
| 15 | **StorageFurniture** | open, contain | 183 |
| 16 | **Refrigerator** | open, contain | 23 |
| 17 | **Mug** | contain, grasp, pour, wrap-grasp | 45 |
| 18 | **Keyboard** | press | 10 |
| 19 | **Table** | support, move | 431 |
| 20 | **Bed** | sit, support, lay | 36 |
| 21 | **Hat** | wear, grasp | 26 |
| 22 | **Laptop** | display, press | 55 |
| 23 | **TrashCan** | open, contain, pour | 51 |
| **Total** | **23 Categories** | **17 Affordance Types** | 2416 |

## A.3. The LASO-C Dataset

Our proposed LASO-C dataset is derived from the test set of the **Seen** partition in LASO [9], focusing on evaluating model robustness against point cloud corruptions. This dataset comprises $2,416$ object-affordance pairings, cover-

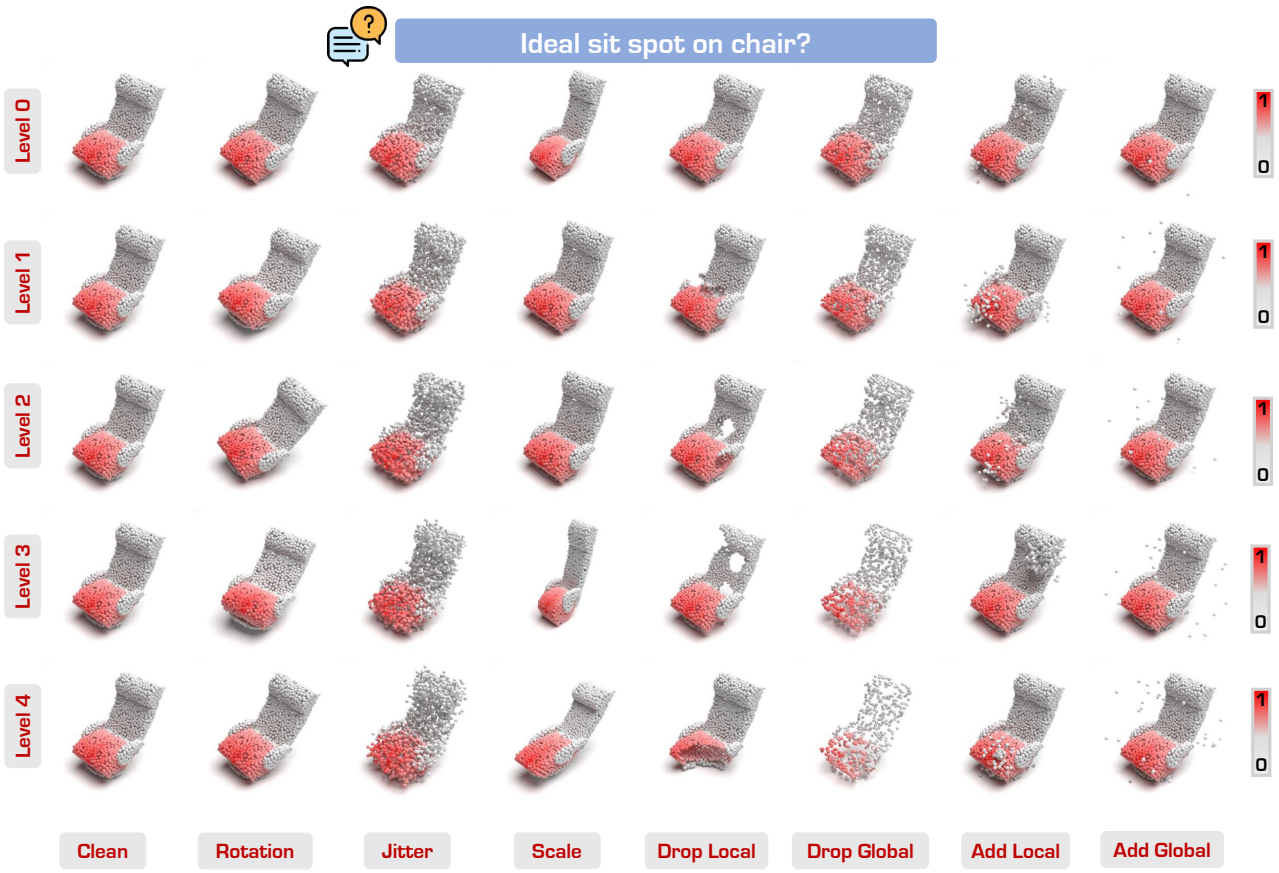


Figure A. Visualization examples of the PIAD-C dataset. We show 7 corruption types across 5 severity levels.

ing 17 affordance categories and 23 object categories, with a total of 1,035 distinct clean object shapes.

The comprehensive statistics, detailing object categories, their corresponding affordance categories, and the number of object-affordance pairings, are presented in Tab. B. We include additional visualization examples for the LASO-C dataset in Fig. B.

## B. Benchmark Configuration

In this section, we elaborate in more detail on the configurations and evaluations of the proposed robust 3D affordance learning benchmark.

### B.1. Datasets

We conduct experiments primarily on the **LASO**[9] and **PIAD**[22] datasets, both of which provide paired affordance and point cloud data for evaluating 3D affordance learning.

**LASO.** This dataset is a pioneering benchmark designed to enable language-guided affordance segmentation of 3D objects. It includes **19,751 point cloud-question pairs** across **8,434 unique object shapes**, spanning **23 object categories** and **17 affordance types**. Derived from **3D-AffordanceNet** [3], the dataset pairs 3D object point clouds with questions that were carefully crafted by human experts and augmented using **GPT-4**. This process incorporates principles of *contextual enrichment*, *concise phrasing*, and *structural diversity*, enhancing the linguistic variety and complexity of the dataset.

The LASO dataset introduces two distinct evaluation settings:

- **Seen Setting:** Models are trained and tested on overlapping object-affordance combinations, ensuring that both the object classes and affordance types in the training set are also present in the test set.
- **Unseen Setting:** This setting is designed to evaluate generalization capabilities. Certain object-affordance combinations (*e.g.*, "grasp-mug") are excluded during training but appear in testing. This setting challenges models to transfer affordance knowledge learned from seen combinations (*e.g.*, "grasp-bag") to novel combinations, promoting robust generalization.

These settings promote a comprehensive evaluation of

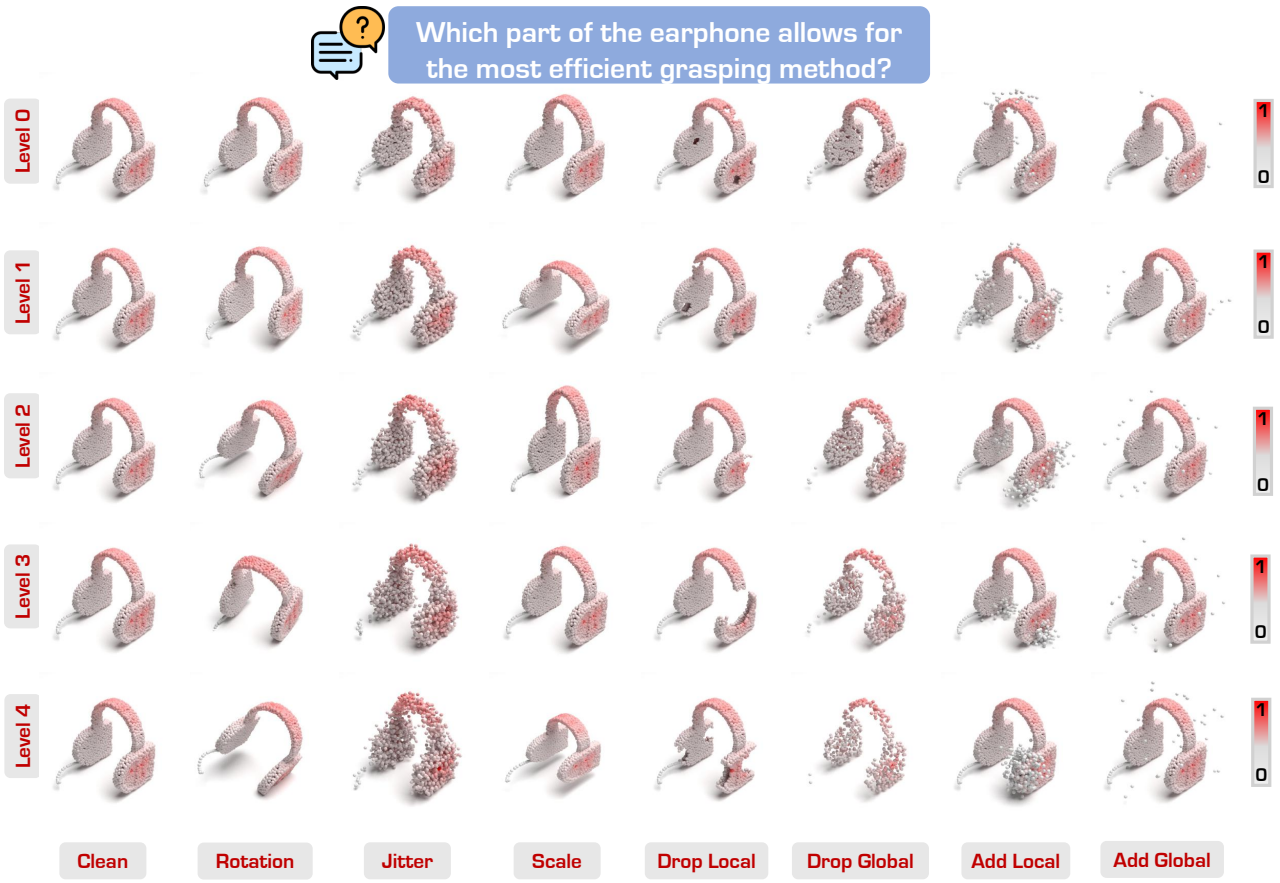


Figure B. Visualization examples of the LASO-C dataset. We show 7 corruption types across 5 severity levels.

models' abilities to generalize affordance knowledge to unseen object-affordance pairings, a critical aspect for real-world deployment. The dataset also emphasizes diverse affordance scales and shapes, presenting significant challenges for perception models. By addressing the semantic limitations of traditional visual-only 3D affordance datasets, LASO bridges the gap between 3D perception and natural language understanding, encouraging cross-modal learning. This integration fosters advancements in embodied AI, enabling tasks that require nuanced reasoning and action in real-world environments.

**PIAD.** The Point-Image Affordance Dataset (PIAD) [22] is specifically curated to advance the task of grounding 3D object affordances using 2D interactions. PIAD consists of $\mathbf{7,012}$ **point clouds** and $\mathbf{5,162}$ **images**, spanning **23 object classes** and **17 affordance categories**. Unlike other datasets, PIAD pairs point clouds with images that demonstrate corresponding affordances. For example, a point cloud of a "Chair" affords "Sit," and its paired image depicts a person sitting on a chair. These cross-modal pairings ensure consistency in affordance relationships while leveraging distinct modalities.

PIAD introduces two distinct evaluation settings:

- **Seen Setting:** In this setting, both objects and affordances in the training and testing sets are consistent. Point clouds and images of the same object categories and affordance types are included during training, allowing models to learn affordance relationships in a supervised manner. This standard evaluation setting enables benchmarking on familiar object-affordance combinations.
- **Unseen Setting:** The Unseen partition presents a more challenging evaluation by excluding certain object categories from the training set entirely. For instance, some object categories are entirely unseen during training. This partition tests the ability of methods to transfer affordance knowledge across completely novel object instances and contexts, simulating real-world scenarios where interaction data is sparse or varied.

Annotations in PIAD include detailed affordance labels for point clouds, represented as heatmaps indicating the likelihood of affordance at each point. Paired images are annotated with bounding boxes for interactive subjects and

objects, along with affordance category labels. This comprehensive annotation schema supports diverse affordance-learning paradigms and provides a robust benchmark for evaluating models in both Seen and Unseen scenarios.

Note that PIAD does not include language annotations. Since PIAD and LASO share the same object classes, affordance categories, and the same 58 affordance-object pairings, we reuse LASO's language annotations for PIAD. For each object and affordance category label in PIAD, we randomly select a question from LASO's question dataset corresponding to that affordance-object pairing.

### B.2. Evaluation Metrics

To comprehensively evaluate the performance of our method, we employ four widely used metrics: **AUC**, **aIoU**, **SIM**, and **MAE**. Each metric is designed to assess different aspects of affordance prediction, providing a robust and multi-faceted evaluation framework. Below, we detail the formulation and significance of each metric:

- **Area Under the ROC Curve (AUC)** [11]: AUC measures the model's ability to distinguish between regions of high and low affordance saliency on the point cloud. Specifically, the saliency map is treated as a binary classifier at various threshold levels, and a Receiver Operating Characteristic (ROC) curve is generated by plotting the true positive rate (TPR) against the false positive rate (FPR) at each threshold. AUC provides a single scalar value summarizing the overall performance, where higher values indicate better discrimination ability. It is particularly useful for comparing models' effectiveness in highlighting affordance-relevant regions.
- **Average Intersection over Union (aIoU)** [13]: IoU is a standard metric for comparing the similarity between two arbitrary regions—in this case, the predicted affordance region and the ground truth. It is defined as the size of the intersection between the two regions divided by the size of their union:

$$IoU = \frac{TP}{TP + FP + FN}, \tag{1}$$

  where $TP$, $FP$, and $FN$ denote true positives, false positives, and false negatives, respectively. The aIoU extends this metric to compute the average IoU across all categories and test samples, providing a quantitative measure of the overlap between predicted and labeled affordance regions. Higher values indicate better alignment between the prediction and the ground truth.
- **Similarity (SIM)** [15]: The SIM metric evaluates how closely the predicted affordance map matches the ground truth by summing the minimum values at each point. For normalized prediction and ground truth maps $P$ and $Q$, the similarity is calculated as:

$$SIM(P, Q) = \sum_i \min(P_i, Q_i), \tag{2}$$

  where the inputs are normalized such that $\sum_i P_i = \sum_i Q_i = 1$. SIM provides a measure of how well the model captures the relative affordance distribution across the point cloud. A higher similarity score reflects greater consistency between the predicted and true maps, making it a valuable metric for evaluating spatial prediction fidelity.
- **Mean Absolute Error (MAE)** [19]: MAE quantifies the average absolute difference between the predicted affordance values and the ground truth, offering a direct measure of prediction accuracy. For $n$ points in a point cloud, it is calculated as:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |e_i|, \tag{3}$$

  where $e_i$ is the point-wise error. MAE is particularly useful for evaluating overall prediction quality by penalizing larger deviations. Lower MAE values indicate better performance, as they reflect a smaller error margin between the predicted and ground truth affordance scores.

Together, these metrics provide a comprehensive framework to benchmark the performance of affordance prediction models. AUC evaluates ranking capability, aIoU measures spatial overlap, SIM assesses prediction similarity, and MAE quantifies overall prediction accuracy. By combining these complementary metrics, we ensure a holistic evaluation of model performance under diverse scenarios.

### B.3. Baselines

We evaluate our method against state-of-the-art approaches on both the PIAD and LASO datasets. Among these, LASO [9] is the closest to our method, as it also generates affordance scores based on textual cues. Additionally, we include 3D cross-modal baselines such as 3D-SPS [12], and image segmentation methods like ReferTrans [8] and RelA [10], which leverage vision-language models for cross-modal alignment. Results for these methods are referenced directly from the LASO paper.

On the PIAD dataset, we compare against IAGNet [22], a method that grounds 3D affordances by transferring knowledge from demonstration images into point clouds. Furthermore, this benchmark includes advanced image-point cloud cross-modal methods, including MBDF [16], PMF [23], FRCNN [21], ILN [2], PFusion [20], and XMF [1]. These baselines align image and point cloud features in various ways. Results for these baselines are taken from the IAGNet paper, except for LASO, which is retrained in the PIAD setting.

Below is a brief introduction to the baselines:

- **LASO** [9]: Generates affordance segmentation masks using textual-conditioned affordance queries, focusing on cross-modal alignment between text and 3D objects.

Table C. Computational costs comparison on LASO dataset.

| Method | Train(ms/it) | | Infer(ms/it) |
|---|---|---|---|
| | 2D | 3D | 3D |
| LASO [9] | - | 31.72 | 17.70 |
| **GEAL** | 32.04 | 40.44 | 18.89 |

- **IAGNet** [22]: Grounds 3D affordances by transferring knowledge from 2D demonstration images to point clouds, leveraging cross-modal affordance reasoning.
- **3D-SPS** [12]: A 3D visual grounding method that selects linguistic keypoints for affordance segmentation, adapted by removing its bounding box prediction module.
- **ReLA** [10]: Originally designed for image-based referring expression segmentation, it segments point clouds based on language expressions by adapting image region features to grouped point features.
- **ReferTrans** [8]: A transformer-based architecture for image-based expression segmentation, modified for point clouds by replacing the image backbone with a 3D backbone and focusing solely on mask prediction.
- **MBDF-Net (MBDF)** [16]: Employs an Adaptive Attention Fusion (AAF) module for cross-modal feature fusion, with modifications to exclude camera intrinsic parameters.
- **PMF** [23]: Uses a residual-based fusion model to combine image and point cloud features, incorporating convolution and attention, while omitting perspective projection.
- **FusionRCNN (FRCNN)** [21]: Fuses proposals extracted from images and point clouds through iterative self-attention and cross-attention mechanisms.
- **ImloveNet (ILN)** [2]: Projects image features into 3D space using a learnable mapping, and fuses these with point cloud features using an attention mechanism.
- **PointFusion (PFusion)** [20]: Performs dense fusion by combining global and point-wise features extracted separately from point clouds and images.
- **XMFnet (XMF)** [1]: Fuses localized features from point clouds and images using a combination of cross-attention and self-attention, originally designed for cross-modal point cloud completion.

# C. Additional Quantitative Results

In this section, we provide additional quantitative results, *i.e.*, the class-wise and corruption-wise evaluation metrics, and the computational cost comparison to demonstrate the effectiveness of our method.

## C.1. Complete Results on PIAD

The complete results of the comparative methods for all object categories in the **Seen** and **Unseen** partitions of the PIAD dataset [22] are provided in Tab. D and Tab. E, respectively.

## C.2. Complete Results on LASO

The complete results of the comparative methods for all object categories in the **Seen** and **Unseen** partitions of the LASO dataset [9] are provided in Tab. F and Tab. G, respectively.

## C.3. Computation Resources

As detailed in the main text, we first train the 2D branch, then train the 3D branch while inferring the 2D branch to propagate knowledge. During inference, only the 3D branch is used, avoiding slow 2D feature extraction and Gaussian initialization. Tab. C shows **GEAL** achieves comparable inference efficiency to LASO. While training time is not specifically optimized (as this is not the primary focus of our work or previous studies), it remains affordable (within one day totally). Experiments were conducted on an NVIDIA RTX A5000 24GB GPU and an AMD EPYC 32-core CPU. Efficiency is measured by average per-instance training/inference time.

# D. Additional Qualitative Results

In this section, we provide more qualitative results (visual examples) to demonstrate the effectiveness of our method.

## D.1. Additional Qualitative Results on PIAD-C

We include additional qualitative results of **GEAL** and LASO [9] on the PIAD-C dataset in Fig. C.

## D.2. Additional Qualitative Results on PIAD

We include additional qualitative results of **GEAL** and LASO [9] on the PIAD dataset in Fig. D.

Table D. The category-wise results for LASO [9] and GEAL (Ours) on the **Seen** partition of the **PIAD** dataset [22]. AUC and aIOU scores are reported in percentages (%).

| # | Category | LASO [9] | | | | GEAL (Ours) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | aIOU ↑ | AUC ↑ | SIM ↑ | MAE ↓ | aIOU ↑ | AUC ↑ | SIM ↑ | MAE ↓ |
| 1 | **Bag** | 23.4 | 83.3 | 0.567 | 0.090 | 24.0 | 85.1 | 0.588 | 0.088 |
| 2 | **Bed** | 21.1 | 87.3 | 0.587 | 0.097 | 22.7 | 88.1 | 0.595 | 0.091 |
| 3 | **Bowl** | 7.4 | 76.2 | 0.736 | 0.114 | 9.8 | 84.1 | 0.761 | 0.105 |
| 4 | **Clock** | 7.5 | 91.5 | 0.473 | 0.077 | 11.1 | 92.5 | 0.596 | 0.051 |
| 5 | **Dishwash** | 24.7 | 91.9 | 0.464 | 0.069 | 26.2 | 92.9 | 0.496 | 0.058 |
| 6 | **Display** | 32.5 | 91.5 | 0.719 | 0.083 | 37.7 | 91.3 | 0.726 | 0.104 |
| 7 | **Door** | 10.1 | 81.2 | 0.437 | 0.064 | 11.0 | 83.8 | 0.395 | 0.054 |
| 8 | **Earphone** | 18.8 | 85.9 | 0.615 | 0.094 | 21.6 | 87.6 | 0.654 | 0.086 |
| 9 | **Faucet** | 19.9 | 79.9 | 0.517 | 0.099 | 19.1 | 83.6 | 0.602 | 0.078 |
| 10 | **Hat** | 4.7 | 65.9 | 0.604 | 0.148 | 7.8 | 74.2 | 0.620 | 0.133 |
| 11 | **StorageFurniture** | 17.3 | 87.2 | 0.419 | 0.077 | 20.8 | 87.5 | 0.430 | 0.065 |
| 12 | **Keyboard** | 14.8 | 81.2 | 0.249 | 0.059 | 15.2 | 84.6 | 0.257 | 0.048 |
| 13 | **Knife** | 15.5 | 89.8 | 0.671 | 0.060 | 23.5 | 94.1 | 0.717 | 0.046 |
| 14 | **Laptop** | 29.2 | 94.1 | 0.566 | 0.072 | 31.2 | 94.2 | 0.575 | 0.069 |
| 15 | **Microwave** | 30.1 | 96.8 | 0.524 | 0.037 | 35.5 | 96.9 | 0.545 | 0.037 |
| 16 | **Mug** | 10.7 | 76.5 | 0.578 | 0.107 | 17.5 | 77.2 | 0.607 | 0.091 |
| 17 | **Refrigerator** | 23.2 | 87.1 | 0.473 | 0.070 | 24.7 | 89.6 | 0.460 | 0.070 |
| 18 | **Chair** | 27.5 | 88.1 | 0.649 | 0.094 | 28.5 | 89.0 | 0.652 | 0.066 |
| 19 | **Scissors** | 24.1 | 91.2 | 0.631 | 0.055 | 31.9 | 95.8 | 0.698 | 0.040 |
| 20 | **Table** | 10.1 | 78.2 | 0.627 | 0.129 | 11.4 | 79.1 | 0.639 | 0.135 |
| 21 | **TrashCan** | 11.9 | 67.4 | 0.323 | 0.143 | 16.2 | 68.8 | 0.385 | 0.146 |
| 22 | **Vase** | 10.3 | 72.0 | 0.608 | 0.120 | 12.5 | 72.4 | 0.612 | 0.116 |
| 23 | **Bottle** | 23.5 | 77.3 | 0.552 | 0.110 | 27.8 | 79.8 | 0.536 | 0.107 |

Table E. The category-wise results for LASO [9] and GEAL (Ours) on the **Unseen** partition of the **PIAD** dataset [22]. AUC and aIOU scores are reported in percentages (%).

| # | Category | LASO [9] | | | | GEAL (Ours) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | aIOU ↑ | AUC ↑ | SIM ↑ | MAE ↓ | aIOU ↑ | AUC ↑ | SIM ↑ | MAE ↓ |
| 1 | **Bed** | 12.0 | 78.0 | 0.469 | 0.126 | 12.8 | 78.4 | 0.473 | 0.120 |
| 2 | **Dishwasher** | 17.3 | 84.9 | 0.338 | 0.079 | 18.3 | 89.8 | 0.440 | 0.060 |
| 3 | **Laptop** | 4.5 | 65.4 | 0.192 | 0.122 | 6.3 | 74.5 | 0.201 | 0.100 |
| 4 | **Microwave** | 14.4 | 83.4 | 0.365 | 0.066 | 15.8 | 89.6 | 0.402 | 0.049 |
| 5 | **Scissors** | 3.2 | 66.5 | 0.310 | 0.107 | 3.7 | 69.8 | 0.333 | 0.123 |
| 6 | **Vase** | 5.2 | 58.1 | 0.455 | 0.140 | 6.4 | 54.9 | 0.466 | 0.127 |

Table F. The category-wise results for LASO [9] and GEAL (Ours) on the **Seen** partition of the **LASO** dataset [9]. AUC and aIOU scores are reported in percentages (%).

| # | Category | LASO [9] | | | | GEAL (Ours) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | aIOU ↑ | AUC ↑ | SIM ↑ | MAE ↓ | aIOU ↑ | AUC ↑ | SIM ↑ | MAE ↓ |
| 1 | **Bag** | 19.8 | 85.4 | 0.535 | 0.085 | 20.6 | 86.7 | 0.572 | 0.084 |
| 2 | **Bed** | 13.6 | 77.4 | 0.515 | 0.111 | 16.0 | 79.9 | 0.527 | 0.110 |
| 3 | **Bowl** | 8.6 | 81.3 | 0.777 | 0.102 | 12.2 | 87.4 | 0.807 | 0.102 |
| 4 | **Clock** | 6.2 | 84.2 | 0.461 | 0.064 | 9.8 | 84.8 | 0.485 | 0.062 |
| 5 | **Dishwash** | 29.6 | 94.1 | 0.472 | 0.070 | 28.5 | 89.9 | 0.505 | 0.068 |
| 6 | **Display** | 31.0 | 92.2 | 0.700 | 0.086 | 41.1 | 92.6 | 0.718 | 0.088 |
| 7 | **Door** | 12.3 | 82.3 | 0.311 | 0.060 | 15.7 | 83.8 | 0.368 | 0.058 |
| 8 | **Earphone** | 26.5 | 93.0 | 0.639 | 0.099 | 27.5 | 94.0 | 0.662 | 0.094 |
| 9 | **Faucet** | 14.2 | 78.9 | 0.498 | 0.089 | 18.3 | 84.3 | 0.589 | 0.087 |
| 10 | **Hat** | 3.6 | 67.0 | 0.538 | 0.152 | 9.3 | 72.7 | 0.560 | 0.148 |
| 11 | **StorageFurniture** | 19.2 | 88.6 | 0.437 | 0.067 | 24.7 | 89.3 | 0.481 | 0.066 |
| 12 | **Keyboard** | 12.0 | 89.0 | 0.227 | 0.055 | 12.9 | 87.9 | 0.232 | 0.039 |
| 13 | **Knife** | 14.8 | 91.3 | 0.642 | 0.064 | 22.9 | 93.2 | 0.657 | 0.063 |
| 14 | **Laptop** | 28.5 | 95.1 | 0.583 | 0.078 | 29.8 | 95.1 | 0.586 | 0.070 |
| 15 | **Microwave** | 27.2 | 96.1 | 0.440 | 0.042 | 31.8 | 92.8 | 0.464 | 0.038 |
| 16 | **Mug** | 13.3 | 78.1 | 0.547 | 0.098 | 21.7 | 87.6 | 0.635 | 0.076 |
| 17 | **Refrigerator** | 25.6 | 92.8 | 0.433 | 0.063 | 24.8 | 93.7 | 0.484 | 0.069 |
| 18 | **Chair** | 28.9 | 89.9 | 0.650 | 0.093 | 28.7 | 89.9 | 0.678 | 0.091 |
| 19 | **Scissors** | 17.5 | 95.4 | 0.661 | 0.053 | 24.9 | 95.9 | 0.684 | 0.045 |
| 20 | **Table** | 10.1 | 81.7 | 0.662 | 0.119 | 10.8 | 81.6 | 0.690 | 0.115 |
| 21 | **TrashCan** | 10.9 | 72.1 | 0.323 | 0.137 | 27.8 | 90.4 | 0.499 | 0.100 |
| 22 | **Vase** | 7.9 | 71.1 | 0.630 | 0.125 | 13.5 | 79.5 | 0.650 | 0.116 |
| 23 | **Bottle** | 20.4 | 81.2 | 0.553 | 0.114 | 28.7 | 81.9 | 0.570 | 0.116 |

Table G. The category-wise results for LASO [9] and GEAL (Ours) on the **Unseen** partition of the **LASO** dataset [9]. AUC and aIOU scores are reported in percentages (%).

| # | Category | LASO [9] | | | | GEAL (Ours) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | aIOU ↑ | AUC ↑ | SIM ↑ | MAE ↓ | aIOU ↑ | AUC ↑ | SIM ↑ | MAE ↓ |
| 1 | **Bag** | 20.7 | 89.1 | 0.513 | 0.089 | 22.1 | 91.0 | 0.522 | 0.086 |
| 2 | **Bed** | 12.2 | 80.6 | 0.553 | 0.115 | 13.6 | 81.4 | 0.563 | 0.113 |
| 3 | **Bowl** | 7.5 | 81.3 | 0.744 | 0.125 | 9.1 | 82.5 | 0.749 | 0.119 |
| 4 | **Clock** | 5.3 | 85.2 | 0.419 | 0.094 | 6.4 | 85.0 | 0.433 | 0.079 |
| 5 | **Dishwash** | 20.7 | 92.4 | 0.443 | 0.069 | 26.0 | 92.4 | 0.470 | 0.065 |
| 6 | **Display** | 23.4 | 86.6 | 0.512 | 0.112 | 25.0 | 87.6 | 0.526 | 0.112 |
| 7 | **Door** | 3.4 | 81.3 | 0.324 | 0.095 | 11.7 | 81.4 | 0.355 | 0.066 |
| 8 | **Earphone** | 9.5 | 76.8 | 0.454 | 0.130 | 20.8 | 93.5 | 0.639 | 0.091 |
| 9 | **Faucet** | 13.8 | 74.1 | 0.442 | 0.098 | 15.1 | 76.8 | 0.470 | 0.095 |
| 10 | **Hat** | 4.5 | 61.2 | 0.586 | 0.158 | 4.1 | 66.5 | 0.582 | 0.149 |
| 11 | **StorageFurniture** | 17.9 | 88.1 | 0.422 | 0.069 | 18.3 | 88.3 | 0.423 | 0.067 |
| 12 | **Keyboard** | 3.1 | 74.6 | 0.138 | 0.082 | 3.3 | 79.4 | 0.137 | 0.078 |
| 13 | **Knife** | 15.3 | 91.7 | 0.643 | 0.053 | 15.4 | 91.2 | 0.675 | 0.059 |
| 14 | **Laptop** | 8.7 | 79.7 | 0.334 | 0.096 | 29.3 | 95.6 | 0.610 | 0.064 |
| 15 | **Microwave** | 11.9 | 90.9 | 0.317 | 0.063 | 14.2 | 91.5 | 0.318 | 0.064 |
| 16 | **Mug** | 1.7 | 64.5 | 0.381 | 0.174 | 2.5 | 66.6 | 0.511 | 0.157 |
| 17 | **Refrigerator** | 20.1 | 87.2 | 0.378 | 0.066 | 21.0 | 89.4 | 0.390 | 0.065 |
| 18 | **Chair** | 25.2 | 87.4 | 0.642 | 0.098 | 26.0 | 89.4 | 0.624 | 0.094 |
| 19 | **Scissors** | 1.6 | 25.3 | 0.094 | 0.105 | 2.1 | 27.6 | 0.105 | 0.097 |
| 20 | **Table** | 7.5 | 70.4 | 0.604 | 0.135 | 7.8 | 72.1 | 0.620 | 0.129 |
| 21 | **TrashCan** | 2.6 | 63.1 | 0.191 | 0.124 | 7.4 | 71.0 | 0.293 | 0.125 |
| 22 | **Vase** | 6.4 | 56.4 | 0.466 | 0.148 | 7.6 | 67.0 | 0.614 | 0.140 |
| 23 | **Bottle** | 16.2 | 78.5 | 0.455 | 0.134 | 21.2 | 78.2 | 0.519 | 0.119 |

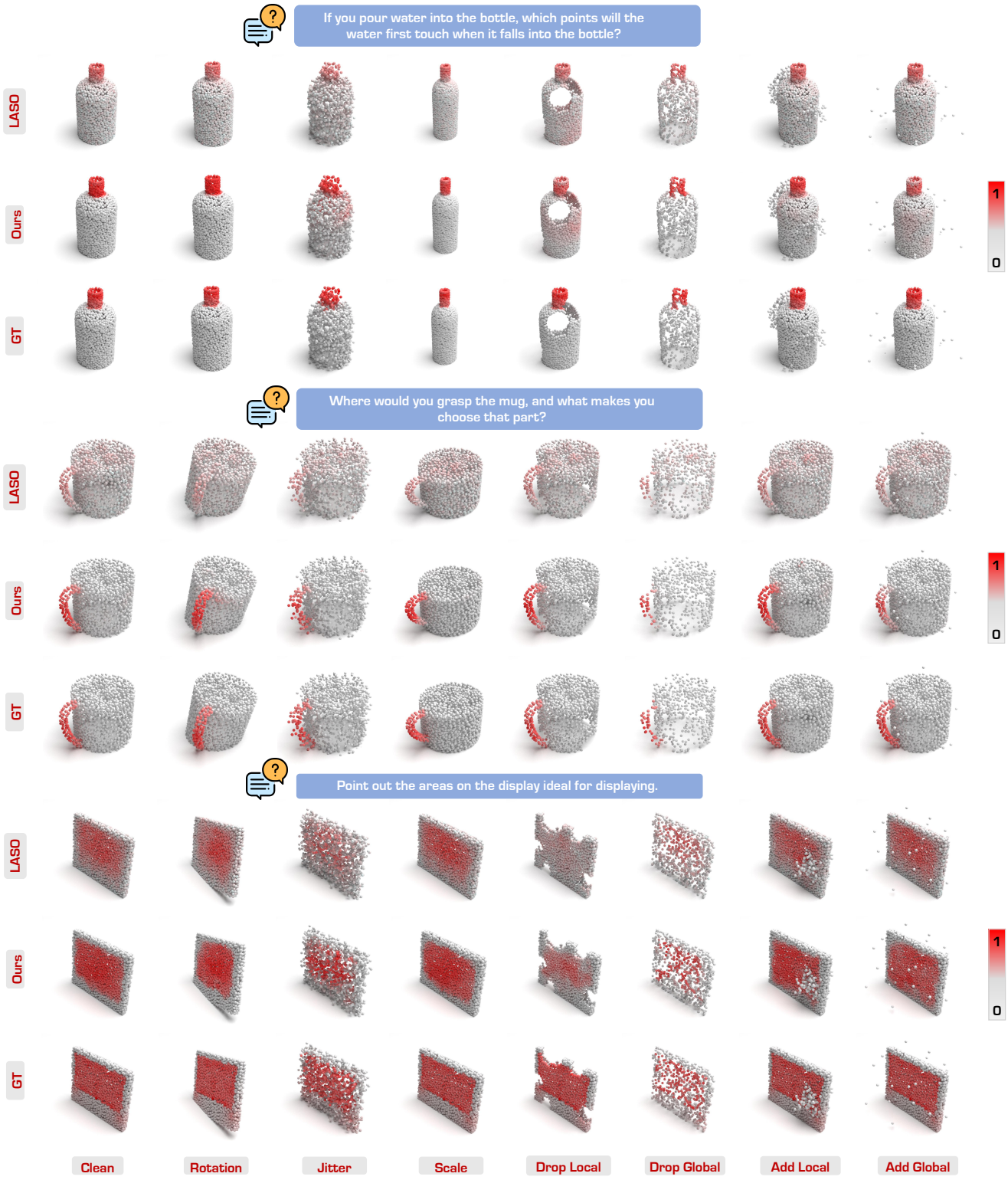


Figure C. Qualitative comparisons between **GEAL** and LASO [9] on the PIAD-C dataset, highlighting the superior robustness of our method on corrupted data.

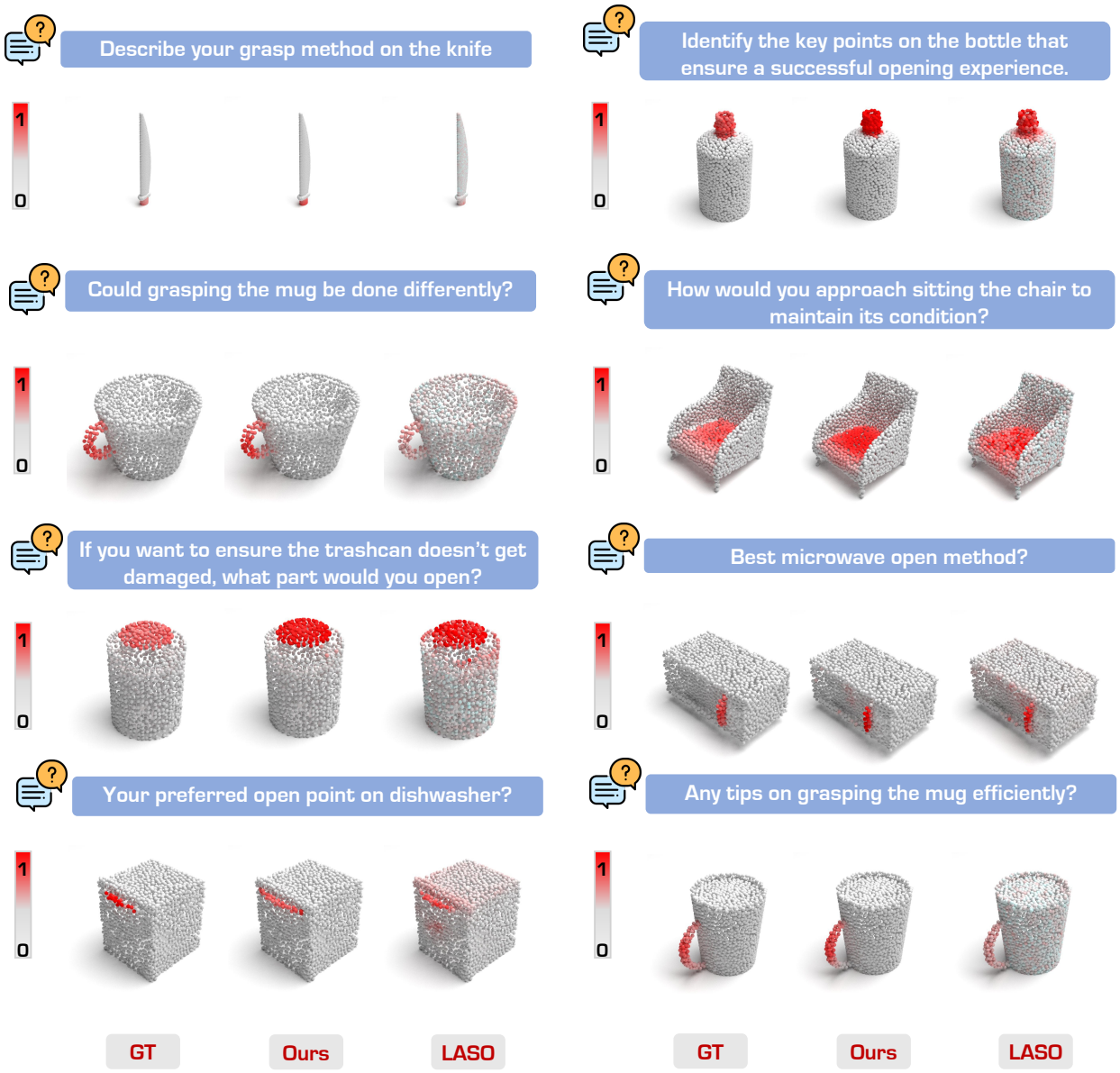


Figure D. Qualitative comparisons between **GEAL** and LASO [9] on the PIAD dataset.

## E. Broader Impact & Limitations

In this section, we discuss the societal impact, broader impact, and potential limitations of this work.

### E.1. Societal Impact

The proposed framework for 3D affordance learning has significant societal implications, enabling embodied intelligence for effective robot and AI interaction with surroundings. This advancement can enhance automated systems' efficiency and safety in fields like healthcare, elderly care, and disaster response, where understanding object affordances is critical. This technology also has the potential to empower individuals with disabilities by enabling assistive robots to perform tasks such as fetching, opening, or manipulating objects. Applications in education and augmented or virtual reality could transform learning and entertainment by offering immersive and interactive experiences.

### E.2. Broader Impact

Affordance learning can redefine robotics automation by improving autonomy and adaptability in industries. In manufacturing, it allows robots to handle diverse objects with minimal reprogramming, optimizing workflows and reducing human workload. In agriculture and environmental monitoring, affordance-aware systems can adapt to dynamic environments for precise operations. Integrating affordance grounding with augmented and virtual reality enables new possibilities in training, simulation, and interactive applications. This could drive innovations in medical training, such as AR-guided surgeries, and in gaming, offering intuitive and immersive user experiences through affordance-based interactions.

### E.3. Potential Limitations

Despite its advantages, the proposed framework may encounter certain limitations:

- **Limited Generalization for Internal Affordances:** The framework struggles to accurately perceive and generalize affordances associated with the internal properties of objects, such as the "contain" affordance of a bottle. This limitation arises because point cloud processing primarily focuses on an object's external surface, often neglecting internal structures. Furthermore, the scarcity of high-quality data representing internal affordances, hampers the system's ability to generalize on such affordances.
- **Ethical Concerns:** In applications such as surveillance or autonomous decision-making, the deployment of the framework introduces potential ethical concerns. Misuse of the technology could infringe on privacy or lead to a lack of accountability in critical decision-making scenarios, highlighting the importance of establishing robust ethical guidelines for its use.
- **Resource Intensity:** Training and deploying such sophisticated models demand significant computational resources, which can pose a challenge for smaller organizations or regions with limited access to advanced technology infrastructure. This barrier may restrict the broader adoption of the framework in resource-constrained environments.

## F. Public Resource Used

In this section, we acknowledge the use of the following public resources, during the course of this work:

- LASO[1] . . . . . Unknown
- IAGNet[2] . . . . . Unknown
- PointCloud-C[3] . . . . . Unknown
- OOAL[4] . . . . . MIT License
- DreamGaussian[5] . . . . . MIT License
- LangSplat[6] . . . . . Gaussian-Splatting License

[1] https://github.com/yl3800/LASO
[2] https://github.com/yyvhang/IAGNet
[3] https://github.com/ldkong1205/PointCloud-C
[4] https://github.com/Reagan1311/OOAL
[5] https://github.com/dreamgaussian/dreamgaussian
[6] https://github.com/minghanqin/LangSplat

## References


[1] Emanuele Aiello, Diego Valsesia, and Enrico Magli. Cross-modal learning for image-guided point cloud shape completion. In *Advances in Neural Information Processing Systems*, pages 37349–37362, 2022. 5, 6

[2] Honghua Chen, Zeyong Wei, Yabin Xu, Mingqiang Wei, and Jun Wang. Imlovenet: Misaligned image-supported registration network for low-overlap point cloud pairs. In *ACM SIGGRAPH Conference Proceedings*, pages 1–9, 2022. 5, 6

[3] Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3d affordancenet: A benchmark for visual object affordance understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2021. 3

[4] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 1

[5] Sihyeon Kim, Sanghyeok Lee, Dasol Hwang, Jaewon Lee, Seong Jae Hwang, and Hyunwoo J Kim. Point cloud augmentation with weighted local transformations. In *IEEE/CVF International Conference on Computer Vision*, pages 548–557, 2021. 1

[6] Lingdong Kong, Youquan Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robo3d: Towards robust and reliable 3d perception against corruptions. In *IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023. 1

[7] Dogyoon Lee, Jaeha Lee, Junhyeop Lee, Hyeongmin Lee, Minhyeok Lee, Sungmin Woo, and Sangyoun Lee. Regularization strategy for point cloud via rigidly mixed sample. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15900–15909, 2021. 1

[8] Muchen Li and Leonid Sigal. Referring transformer: A one-step approach to multi-task visual grounding. In *Advances in Neural Information Processing Systems*, pages 19652–19664, 2021. 5, 6

[9] Yicong Li, Na Zhao, Junbin Xiao, Chun Feng, Xiang Wang, and Tat-Seng Chua. Laso: Language-guided affordance segmentation on 3d object. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14251–14260, 2024. 2, 3, 5, 6, 7, 8, 9, 10

[10] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23592–23601, 2023. 5, 6

[11] Jorge M Lobo, Alberto Jiménez-Valverde, and Raimundo Real. Auc: a misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography*, 17(2):145–151, 2008. 5

[12] Junyu Luo, Jiahui Fu, Xianghao Kong, Chen Gao, Haibing Ren, Hao Shen, Huaxia Xia, and Si Liu. 3d-sps: Single-stage 3d visual grounding via referred point progressive selection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16454–16463, 2022. 5, 6

[13] Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International Symposium on Visual Computing*, pages 234–244, 2016. 5

[14] Jiawei Ren, Liang Pan, and Ziwei Liu. Benchmarking and analyzing point cloud classification under corruptions. In *International Conference on Machine Learning*, pages 18559–18575. PMLR, 2022. 1

[15] Michael J Swain and Dana H Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991. 5

[16] Xun Tan, Xingyu Chen, Guowei Zhang, Jishiyu Ding, and Xuguang Lan. Mbdf-net: Multi-branch deep fusion network for 3d object detection. In *International Workshop on Multimedia Computing for Urban Data*, pages 9–17, 2021. 5, 6

[17] Jie Wang, Lihe Ding, Tingfa Xu, Shaocong Dong, Xinli Xu, Long Bai, and Jianan Li. Sample-adaptive augmentation for point cloud recognition against real-world corruptions. In *IEEE/CVF International Conference on Computer Vision*, pages 14330–14339, 2023. 1

[18] Jie Wang, Tingfa Xu, Lihe Ding, and Jianan Li. Target-guided adversarial point cloud transformer towards recognition against real-world corruptions. *arXiv preprint arXiv:2411.00462*, 2024. 1

[19] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, 30(1):79–82, 2005. 5

[20] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018. 5, 6

[21] Xinli Xu, Shaocong Dong, Lihe Ding, Jie Wang, Tingfa Xu, and Jianan Li. Fusionrcnn: Lidar-camera fusion for two-stage 3d object detection. *Remote Sensing*, 15(7):1839, 2023. 5, 6

[22] Yuhang Yang, Wei Zhai, Hongchen Luo, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Grounding 3d object affordance from 2d interactions in images. In *IEEE/CVF International Conference on Computer Vision*, pages 10905–10915, 2023. 2, 3, 4, 5, 6, 7

[23] Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 16280–16290, 2021. 5, 6