A Appendix

A.1 Hyperparameter Calibration

We calibrate the two key hyperparameters, the confidence threshold and uncertainty weight, using a held-out subset of 5k COCO images. Matching (image–correct label) and non-matching (image–incorrect label) pairs are used to compute CLIP similarities, with distributions estimated via kernel density estimation (KDE). As shown in Figure 6, the distributions are well-separated (Cohen's $d = 5.06$).

The confidence threshold is set to the mean similarity of matching pairs ($\mu_+ = 0.311$, rounded to 0.32), and the uncertainty weight is set to the intersection of the distributions (rounded to 0.2), corresponding to the Bayes-optimal decision boundary.

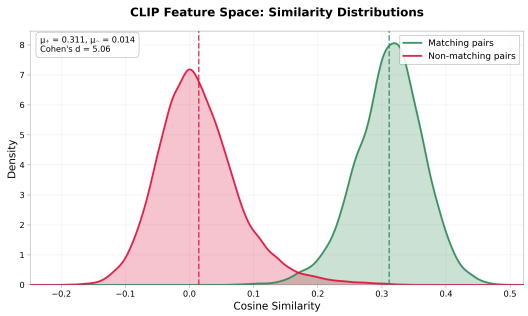

Figure 6: **CLIP Feature Space: Similarity Distributions.** The plot shows the kernel density estimates (KDE) of cosine similarity distributions for matching (image–correct label) and non-matching (image–incorrect label) pairs within the CLIP feature space, using a held-out subset of COCO. The green distribution represents matching pairs, and its mean ($\mu_+ = 0.311$) is used as the **confidence threshold** (dashed green line), rounded to 0.32. This ensures that accepted labels correspond to in-distribution confidence levels. The red distribution represents non-matching pairs. The intersection point of the two distributions, which corresponds to the Bayes-optimal decision boundary, is used to set the **uncertainty weight**, rounded to 0.2.

A.2 Hyperparameters

We present the hyperparameters used in our experiments at each step of the workflow. All images are 1920x1080 to preserve high-level detail.

DetectAndSegment YOLOE (Wang et al., 2025) in prompt-free mode is used to generate object bounding boxes. Detections with fewer than 5,000 pixels or confidence below 0.15 are discarded. Features are extracted using OpenCLIP (Cherti et al., 2023) ViT-H-14 with the "laion2b_s32b_b79k" weights, and segmentation is performed with Segment Anything 2 (Ravi et al., 2024).

MergeObservations In single-object experiments, all detections are assumed to correspond to the same object and are merged. In multi-object experiments, observations are merged based on semantic (visual and textual) and spatial similarity. Visual semantic similarity is computed as the average CLIP features of detections (excluding removed ones from the RefineAndPropose step) with a threshold of 0.6. Textual similarity is the cosine similarity of CLIP-encoded current object labels, thresholded at 0.25. Spatial similarity is measured via point cloud overlap, with a threshold of 0.1 to allow minimal overlap. Objects exceeding all three thresholds are merged.

RefineAndPropose For both experiments, we set the maximum number of inner-loop iterations between data collection steps to 3, the confidence threshold to 0.32, the uncertainty weight to 0.2, and the number of polygon faces in the spatial partitioning algorithm to 8.

A.3 Evaluation

A.3.1 Setting the Success Threshold

As LADR is fully open-vocabulary, direct comparison with ground-truth labels is insufficient: the LLM may propose synonyms, which should be accepted. Since CLIP is sensitive to lexical variations, we use a Sentence Transformer (Reimers & Gurevych, 2019) to evaluate label equivalence. The final similarity for each prediction is the maximum of its similarity to the class name or description. To convert similarities into success rates, we construct a small set of synonym and non-synonym pairs, compute their similarities in the Sentence Transformer feature space, and visualize the distributions using kernel density estimation (KDE). The results show clear separation: while matching pairs can occasionally fall below 0.5, non-matching pairs never exceed 0.5. Based on this, we adopt 0.5 as the default threshold for evaluating label correctness.

To provide a more nuanced view, we also evaluate success rates at multiple thresholds:

- **0.3:** Almost all word pairs are detected as synonyms, including weakly related or contextually distant ones.
- **0.5:** Serves as a baseline, capturing meaningful synonyms while avoiding unrelated pairs.
- **0.7:** Mostly multi-word phrases with strong semantic alignment; loosely related pairs are excluded.
- **0.9:** Captures nearly identical or identical pairs, useful for exact matches.

By reporting success rates at these thresholds, we provide a more detailed picture of the model's behavior across varying levels of semantic similarity, from broad synonym detection to nearly exact matches.

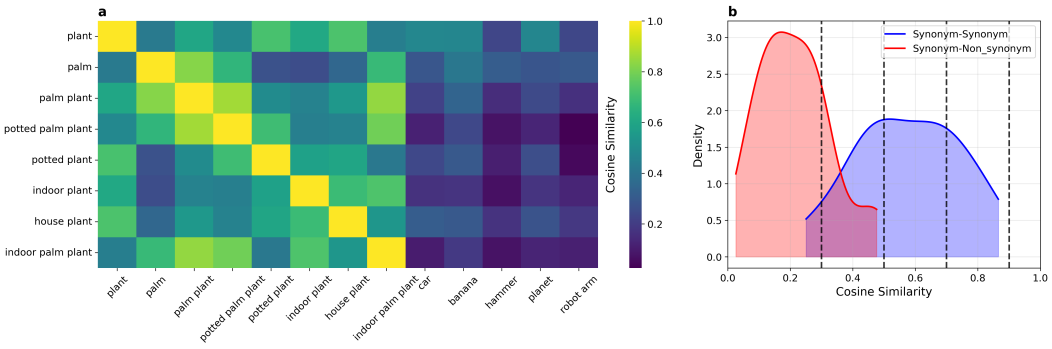

Figure 7: **Synonym Distance Analysis (a)** A cosine similarity heatmap for the word 'plant' and a set of related terms. The diagonal entries show high similarity for synonyms and near-synonyms (e.g., 'plant', 'potted plant', 'indoor plant'). Non-synonyms (e.g., 'car', 'banana', 'planet') exhibit low similarity. **(b)** Kernel density estimate (KDE) plots of cosine similarity distributions for synonym (blue) and non-synonym (red) pairs. The distributions show a clear separation, with a default threshold of 0.5 effectively distinguishing between the two. The dashed lines indicate various thresholds (0.3, 0.5, 0.7, and 0.9) used to evaluate the model's performance at different levels of semantic similarity, from broad synonym detection to near-exact matches.

A.3.2 Matching Detections to Ground Truth

To evaluate multi-object detections, we assign each ground-truth object to the best-matching prediction based on a semantic-spatial similarity score, computed as a weighted combination of label similarity and spatial overlap. Only matches with similarity above 0.1 are considered; lower values count as unsuccessful detections. Among eligible matches, the final assignment uses a bias-adjusted aggregation with $\text{phys\_bias} = 0.2$ to select the best match.

A.4 DATASETS

A.4.1 SINGLE-OBJECT DATASET

The single-object dataset comprises five instances for each of the five selected object classes from the OmniObjects3D dataset (Wu et al., 2023). These instances are used to generate multi-view image sequences, with representative examples shown in Figure 8.

Figure 8: Examples of five object instances from the single-object experiments.

A.4.2 MULTI-OBJECT DATASET

The multi-object dataset consists of custom 3D scenes created in NVIDIA Isaac Sim and manually labeled by the authors. To demonstrate the flexibility of our approach, we designed diverse environments using simulator-provided asset packs. The included room types are:

- **SimpleRoom:** open indoor spaces with a mix of miscellaneous objects,
- **Residential:** home-like settings with rug, chairs, and decorative items,
- **Commercial:** office area with a counter, a coffee-table and a storage unit,
- **Industrial:** warehouse-inspired space with shelving, crates, and utility equipment,
- **Vegetation:** outdoor theme featuring plants, trees, and garden elements.

Each scene contains multiple objects of interest, with dense arrangements to test robustness under occlusions, see Figure 9.

Figure 9: Five room scenes used in the multi-object experiments

A.4.3 EXPERIMENT CONFIGURATION

In the single-object experiments, we average over 75 detections (5 classes $\times$ 5 instances $\times$ 3 seeds), starting with two initial views and allowing a budget of five additional views. In the multi-object setting, we average over 300 detections (5 scenes $\times$ 10 objects $\times$ 3 exploration policies $\times$ 2 seeds). At each position, eight new images are captured, beginning from a single initial position with a budget limit of three additional positions.

A.5 Single-Object Experiment Results

A.5.1 Detailed Results

Algorithm	Class Sim	Desc Sim	Best Sim	Avg Sim	Succ@0.3	Succ@0.5	Succ@0.7	Succ@0.9	Avg Tokens
YOLO	0.41 ± 0.25	0.31 ± 0.22	0.43 ± 0.26	0.36 ± 0.23	0.60	0.31	0.20	0.04	0
CLIP	0.51 ± 0.29	0.35 ± 0.20	0.52 ± 0.28	0.43 ± 0.23	0.64	0.39	0.31	0.16	0
LLM-Label	0.49 ± 0.28	0.38 ± 0.21	0.50 ± 0.28	0.42 ± 0.24	0.65	0.47	0.27	0.07	237
LLM-Angle	0.68 ± 0.31	0.59 ± 0.21	0.74 ± 0.27	0.64 ± 0.23	0.91	0.79	0.67	0.40	1575
LLM-Tiled	0.72 ± 0.27	0.62 ± 0.17	0.78 ± 0.23	0.67 ± 0.19	0.97	0.85	0.69	0.40	1008
LLM-Random	0.66 ± 0.26	0.62 ± 0.17	0.73 ± 0.21	0.64 ± 0.19	0.96	0.85	0.63	0.23	2182
LLM-Sampling	0.71 ± 0.30	0.63 ± 0.17	0.79 ± 0.23	0.67 ± 0.20	0.95	0.91	0.72	0.43	16115
LLM-Polygon	0.73 ± 0.24	0.66 ± 0.14	0.80 ± 0.17	0.69 ± 0.15	1.00	0.99	0.72	0.35	22176

Table 2: Detailed evaluation results for different algorithms. Similarity metrics are reported as mean ± standard deviation, followed by success rates at various thresholds and average LLM tokens used.

A.5.2 Sample Results

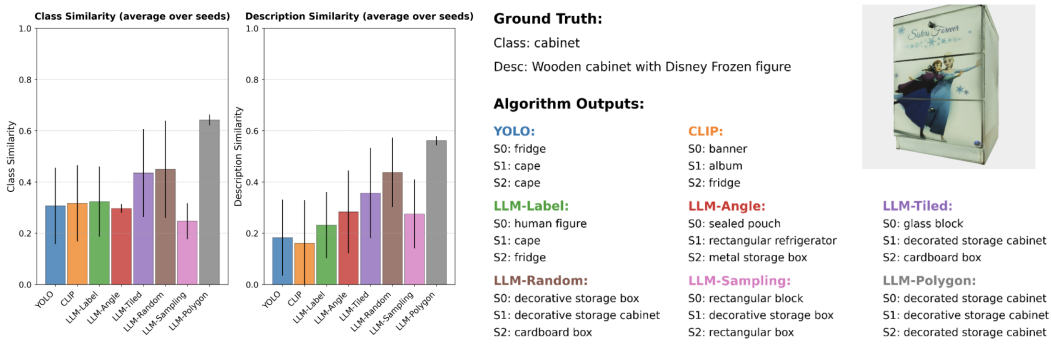

Figure 10: A per-object example showing algorithm performance. The bar charts on the left present class and description similarity, averaged over the seeds, while the right provides a qualitative example for an object in the ‘cabinet’ category from the single-object dataset. This example highlights that the generic ‘cabinet’ label is not sufficiently descriptive for this particular object.

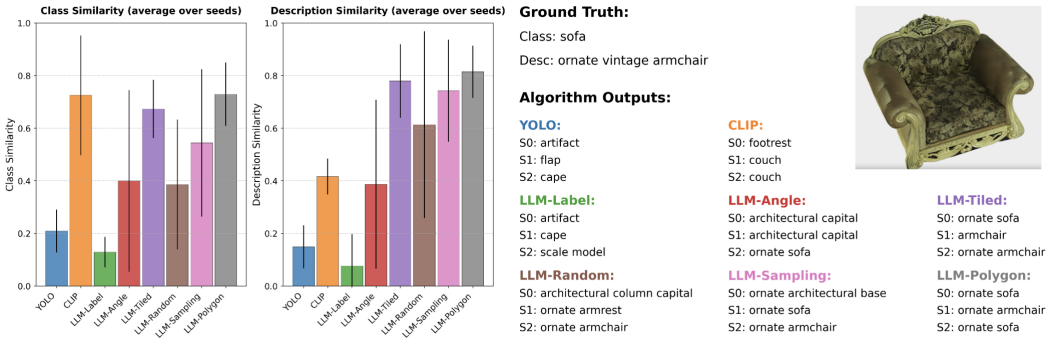

Figure 11: A per-object example showing algorithm performance. The bar charts on the left present class and description similarity, averaged over the seeds, while the right provides a qualitative example for an object in the ”sofa” category in the single-object dataset.

A.6 Multi-Object Experiment Results

A.6.1 Detailed Results

Algorithm	Class Sim	Desc Sim	Best Sim	Avg Sim	Succ@0.3	Succ@0.5	Succ@0.7	Succ@0.9	Avg Tokens
YOLO	0.45 ± 0.24	0.34 ± 0.18	0.46 ± 0.24	0.39 ± 0.21	0.76	0.27	0.16	0.09	0
CLIP	0.51 ± 0.28	0.40 ± 0.23	0.52 ± 0.28	0.45 ± 0.24	0.77	0.40	0.28	0.16	0
LLM-Label	0.48 ± 0.26	0.38 ± 0.20	0.49 ± 0.25	0.43 ± 0.22	0.79	0.33	0.21	0.12	2965
LLM-Angle	0.56 ± 0.28	0.54 ± 0.28	0.63 ± 0.30	0.55 ± 0.26	0.82	0.62	0.47	0.26	8350
LLM-Tiled	0.57 ± 0.27	0.56 ± 0.28	0.64 ± 0.30	0.57 ± 0.26	0.82	0.62	0.47	0.25	6412
LLM-Random	0.56 ± 0.27	0.57 ± 0.28	0.64 ± 0.29	0.57 ± 0.26	0.80	0.63	0.48	0.26	12496
LLM-Sampling	0.59 ± 0.28	0.56 ± 0.28	0.65 ± 0.29	0.58 ± 0.26	0.84	0.64	0.50	0.29	14278
LLM-Polygon	0.59 ± 0.26	0.61 ± 0.28	0.67 ± 0.28	0.60 ± 0.25	0.86	0.69	0.55	0.27	17633

Table 3: Detailed evaluation results for different algorithms. Similarity metrics are reported as mean ± standard deviation, followed by success rates at various thresholds and average LLM tokens used.

A.6.2 Sample Results

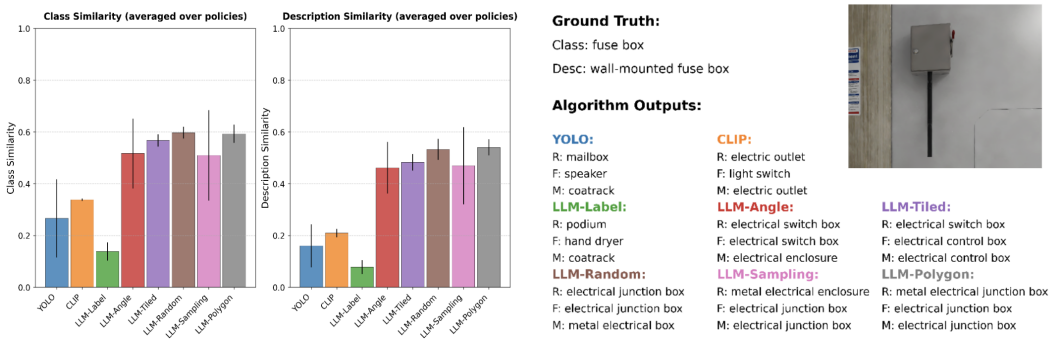

Figure 12: A per-object example showing algorithm performance. The bar charts on the left present class and description similarity, averaged over the exploration policies, while the right provides a qualitative example for an object with ”fuse box” as the ground truth label in the multi-object dataset.

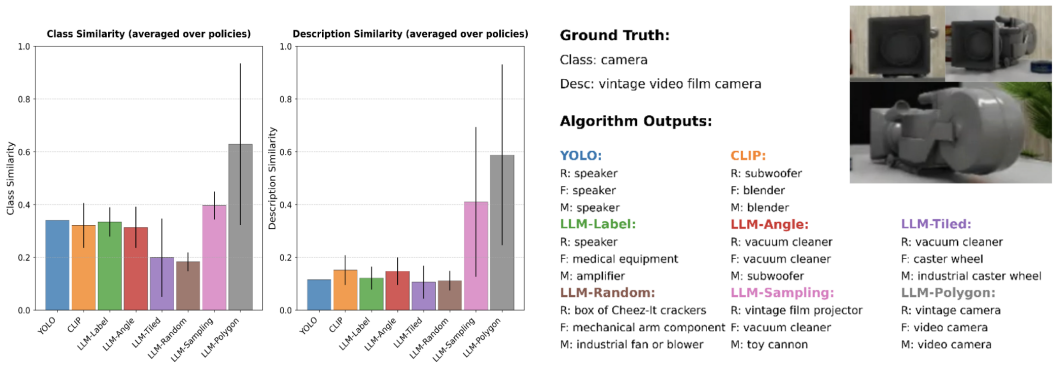

Figure 13: A per-object example showing algorithm performance. The bar charts on the left present class and description similarity, averaged over the exploration policies, while the right provides a qualitative example for an object with ”camera” as the ground truth label in the multi-object dataset.

A.7 LLM IMAGERY INPUT DATA

We provide examples of the LLM-Angle, and LLM-Tile in Figure 14.

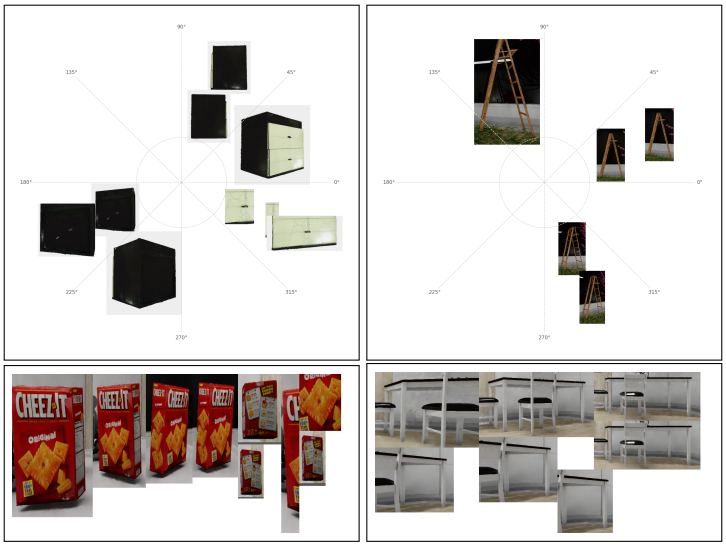

Figure 14: Top: input for LLM-Angle. Bottom: Input for LLM-Tile.

A.8 LLM PROMPT

Example of the prompt provided to the LLM for object labeling. We provide the prompt used for LLM-Random. The prompts for other LADR algorithms are largely similar, with a few differences: neither LLM-Sampling nor LLM-Polygon requests a confidence score or the more descriptive view, and LLM-Polygon also does not request the next-best-view suggestion.

```
You will receive two images of the same object taken from (different) viewpoints, along with
↪  the angles (in degrees) from which they were captured. Analyze both images together
↪  considering their angles and return a single JSON object with these fields:

confident: true or false, indicating whether you are fully confident in the objects class
↪  based on the two views.
label: a brief class name of the object.
description: a clear, detailed description of the object for CLIP encoding. Focus on visually
↪  distinctive features (shape, material, color, texture, patterns) observable in at least
↪  one image.
next_best_angle: an integer in the range [-180, 180] suggesting the single most informative
↪  angle for revealing any ambiguous or missing features.
more_descriptive: either "left" or "right", indicating which image shows features most
↪  representative of the labeled class.
explanation: a short rationale covering:
    - why you set confident to true or false;
    - how you chose label and description;
    - why the proposed next_best_angle will improve clarity;
    - why the chosen image (\left" or \right") is more descriptive.

Guidelines:

Focus on the Main Object.
Each image is a crop around the objects bounding box, and the object fills most of the frame.
↪  Ignore background elements or smaller occluded items.

Combine Both Views and Angles.
Use both images and their provided angles to form a complete understanding. One view may
↪  reveal overall shape, while the other shows texture or details. Identify any remaining
↪  ambiguity or blind spots when choosing your next_best_angle.

Avoid Misidentifying from Partial Views.
If one image shows only a fragment (e.g., a handle), defer to the other image for overall
↪  class identification. Do not let a partial segment mislead your label.
```

```
Highlight Distinctive Features.
Describe only the most visually salient characteristics clearly visible in at least one image.
↪  Write in plain, factual language similar to alt-text or OpenCLIP-style captions.

Assess Confidence.
Set confident to true only if both images clearly support the same object class. If you
↪  suspect the label might change from another viewpoint or if one view is ambiguous, set
↪  confident to false and propose a next_best_angle that would resolve that ambiguity.

Determine \More Descriptive" View.
Compare the two images (left vs. right). Whichever one shows features most representative of
↪  the labeled classwhether by revealing overall shape, distinctive markings, or full
↪  extentshould be marked in more_descriptive. If both show equal detail, choose the one
↪  closest to the objects canonical appearance.

Next-Best-View Proposal.
Recommend a single integer angle in [-180, 180] that would most improve clarity of class or
↪  reveal missing features. Base your suggestion on the two given angles. For example, if the
↪  provided images are at 45 (left) and 60 (right), proposing 0 might reveal the front;
↪  proposing 90 might reveal the opposite side.

Be Precise and Concise.
Write factually. Avoid speculation beyond what the two views suggest. Do not use generic class
↪  labels unsupported by the images.

Output Format
Return exactly one JSON object, for example:
{
  "confident": false,
  "label": "ceramic vase",
  "description": "a rounded ceramic vase with a narrow neck and blue floral patterns on a
  ↪  white background",
  "next_best_angle": 0,
  "more_descriptive": "right",
  "explanation": "The right image clearly shows the floral pattern and vase shape, but the
  ↪  left image only reveals the neck. Because the base is not visible from either 30 or 45,
  ↪  a 0 angle would show the full body and confirm the class."
}

Ensure that your JSON is valid, that all fields are present with the correct types, and that
↪  your response is accurate, well-structured, and concise.
Return only the raw JSON object.
```