# AREA3D: Active Reconstruction Agent with Unified Feed-Forward 3D Perception and Vision-Language Guidance

Supplementary Material

## 7. Dataset and Benchmark

In Sec. 4.1 of the main paper, we introduce our unified benchmark for active 3D reconstruction. Here we provide the complete details of the dataset configuration and scene construction. As illustrated in Fig. 6, we include eight scenes in total: four single-room scenes that capture diverse indoor layouts, and four tabletop scenes featuring object-centric setups with rich geometric details and occlusions. These scenes are used consistently across baselines and ablations, enabling fair comparison under the same camera budget.

## 8. Implementation Details

**Systematic Prompt for VLM.** In Sec. 3.3 we describe how the VLM output is fused with geometric uncertainty. Here we detail the concrete prompt used in practice. At the beginning of each episode, the agent collects $\mathcal{O}_0$ initial RGB views, and we query the VLM once with all $\mathcal{O}_0$ frames. For each image, the field of view is divided into a coarse grid (horizontal: left, center-left, center-right, right; vertical: top, middle, bottom), and the VLM is asked to return a ranked list of regions, each with a location, an uncertainty type (`OCCLUSION`, `GEOMETRIC`, `LIGHTING`, `BOUNDARY`, or `TEXTURE`), a priority level (`HIGH`, `MEDIUM`, `LOW`), and a short natural-language justification. The textual output is then parsed into per-pixel importance maps and lifted into a 3D, visibility-aware uncertainty field via 2D-to-3D unprojection.

For completeness, we show the instruction given to the VLM. We provide all $\mathcal{O}_0$ initial images together with the following text:

> You are an expert visual analyzer for active 3D reconstruction. You will be given several RGB images (initial observations) from the same scene. For each image, independently identify regions that require additional viewpoints for complete 3D reconstruction.
>
> **Coordinate system.** Divide each image into a $4 \times 3$ grid: horizontal positions are *left*, *center-left*, *center-right*, and *right*; vertical positions are *top*, *middle*, and *bottom*. Example locations include "left-top", "center-left-middle", "right-bottom", and "center-right-top".
>
> **Uncertainty categories (ranked by priority).**
>
> - **OCCLUSION (high):** hidden or blocked surfaces; regions behind furniture, walls, or large objects; back faces only visible from narrow viewing angles.
> - **GEOMETRIC (high):** thin structures, surfaces at grazing angles, complex curved shapes, reflective or transparent materials.
> - **LIGHTING (medium):** deep shadows, overexposed areas, strong highlights, blur, very low-contrast regions.
> - **BOUNDARY (medium):** objects cut by image borders, incomplete views, extreme tangential angles.
> - **TEXTURE (low):** textureless, repetitive, or very low-contrast regions.
>
> **Output format.** For each image, list 5–8 regions in decreasing order of importance. Each region should be summarized in one line with the following fields:
>
> - `REGION`: location using the grid notation (e.g., "center-left-middle").
> - `TYPE`: one of `OCCLUSION`, `GEOMETRIC`, `LIGHTING`, `BOUNDARY`, `TEXTURE`.
> - `PRIORITY`: `HIGH` (must observe), `MEDIUM` (should observe), or `LOW` (nice to observe).
> - `SIZE`: `small` ($<$ 10%), `medium` (10–25%), or `large` ($>$ 25%) of the image.
> - `REASON`: 1–2 sentences explaining why extra viewpoints are needed and what 3D information is currently missing.

**Parsing Uncertainty Regions.** In Sec. 3.3 of the main paper we define the 2D spatial weight map

$$W_i(u) = \sum_k \alpha_{\text{type}_k} \, \beta_{\text{prio}_k} \, M_k(u), \tag{1}$$

and the semantic modulation;

$$U_i^{\text{sem}}(u) = \text{Norm}\big(\sigma_i(u)\,[1 + \lambda\, W_i(u)]\big). \tag{2}$$

In our implementation, we employ a fixed weighting scheme that reflects the relative importance of different priorities and remains constant across all experiments.

We set the priority and size-dependent coefficients to fixed values summarized in Table 6; the same settings are used for all experiments. After aggregating over regions, $W_i(u)$ is normalized per image to $[0, 1]$.

**Frustum-based Uncertainty Decay.** In Sec. 3.4 of the main paper we state that, after committing a view, we multiplicatively reduce the fused uncertainty inside the corresponding frustum. Here we detail the decay rule used in our implementation.

Let $u_t(v)$ denote the fused 3D uncertainty at voxel center $v$ at step $t$. Given a committed camera pose $T_w^c$, we

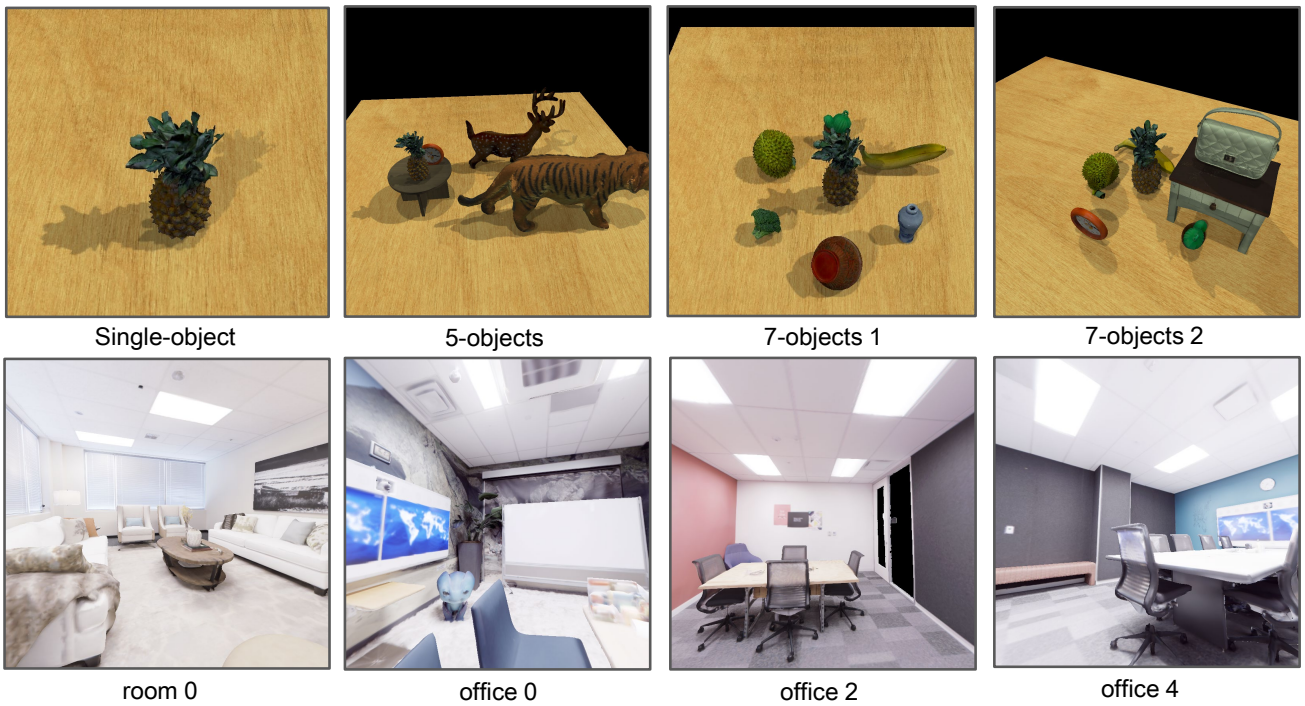

Figure 6. Four single-room scenes that capture diverse indoor layouts, and four tabletop scenes featuring object-centric setups with rich geometric details and occlusions

Table 6. Coefficients for VLM region priority, size, and modulation.

| Symbol | Meaning | Value |
|---|---|---|
| $\beta_{\text{HIGH}}$ | priority = HIGH | 3.0 |
| $\beta_{\text{MED}}$ | priority = MEDIUM | 1.5 |
| $\beta_{\text{LOW}}$ | priority = LOW | 0.5 |
| $s_{\text{small}}$ | size = small | 0.8 |
| $s_{\text{medium}}$ | size = medium | 1.0 |
| $s_{\text{large}}$ | size = large | 1.2 |
| $\lambda$ | modulation strength | 1.0 |

first determine the set of voxels whose centers fall inside the viewing frustum $\text{Frustum}(T_w^c)$, using the camera forward direction, a field-of-view threshold, and a depth range consistent with view rendering). The uncertainty is then updated by

$$u_{t+1}(v) = \begin{cases} (1-\eta)\, u_t(v), & v \in \text{Frustum}(T_w^c), \\ u_t(v), & \text{otherwise}, \end{cases} \tag{3}$$

i.e., all voxels inside the frustum are scaled by a constant decay factor while others remain unchanged.

The hyperparameters used for this frustum-based decay are summarized in Table 7 and are kept fixed for all experiments.

Table 7. Hyperparameters for frustum-based uncertainty decay.

| Symbol | Meaning | Value |
|---|---|---|
| $\eta$ | decay factor | 0.3 |
| FOV | field of view | $90^\circ$ |
| max_depth | maximum depth | 5 m |

## 9. More Quantitative Results

**Overall Aggregate Performance.** To summarize performance on our benchmark, we aggregate the per-scene PSNR, SSIM, and LPIPS reported in the main paper, separately for the object-level and scene-level configurations. Averaged over all object-level scenes, our policy attains 32.09 PSNR, 0.886 SSIM, and 0.102 LPIPS. On the scene-level benchmark, the corresponding averages are 32.40 PSNR, 0.897 SSIM, and 0.089 LPIPS. These aggregated scores provide a compact summary of our behavior on both parts of the benchmark and are consistent with the per-scene comparisons in the main paper, where our policy generally performs on par with or better than competing methods under a fixed view budget.

**Ablation on Global Initial Weight.** In Sec. 3.3 of the main paper we state that, to prevent the agent from being confined to the initially observed views, we assign a global initial uncertainty weight to all voxels. Here we describe the exact form used in implementation and compare it with

Table 8. Ablation study of the global initial weight on both object-level and scene-level benchmarks.

| Setting | PSNR↑ | SSIM↑ | LPIPS↓ |
|---|---|---|---|
| *Object-level* | | | |
| $\gamma = 0$ | 29.212 | 0.859 | 0.120 |
| $\gamma = 0.01$ (ours) | **29.661** | **0.870** | **0.111** |
| *Scene-level* | | | |
| $\gamma = 0$ | 27.845 | 0.837 | 0.153 |
| $\gamma = 0.005$ (ours) | **28.265** | **0.848** | **0.112** |

a variant that removes this term.

Let $\hat{U}(v)$ denote the fused 3D uncertainty projected from the 2D semantic-modulated field. Before view selection begins, each voxel is assigned a small additive initial weight

$$\tilde{U}(v) = \hat{U}(v) + \gamma, \tag{4}$$

where $\gamma$ is a constant offset that ensures non-zero uncertainty for voxels not covered by the initial observation set. This additive form is preserved throughout the reconstruction process and the global initial weight undergoes the same frustum-based decay as the other uncertainty components. In practice, we use different values for the two benchmarks: $\gamma = 0.01$ for the object-level setting and $\gamma = 0.005$ for the scene-level setting.

We compare two configurations: *(i)* a baseline without global initial weight, where $\gamma = 0$; and *(ii)* our default setting with a non-zero initial weight ($\gamma = 0.01$ for object-level, $\gamma = 0.005$ for scene-level), which preserves a minimal amount of residual uncertainty in unseen regions and encourages the policy to explore outside the initially observed frustum.

Quantitative results on both object-level and scene-level benchmarks are reported in Table 8. Using a non-zero initial weight consistently improves viewpoint coverage and leads to better long-range reconstruction quality.

## 10. More Visualization Results

Due to space constraints in the main paper, we only show three qualitative examples of novel view synthesis results obtained with 3D Gaussian Splatting under our active reconstruction policy. Here we provide additional visualizations covering both scene-level and object-level settings. Each row compares our method with baselines on the same target view, as illustrated in Fig. 7 and Fig. 8.

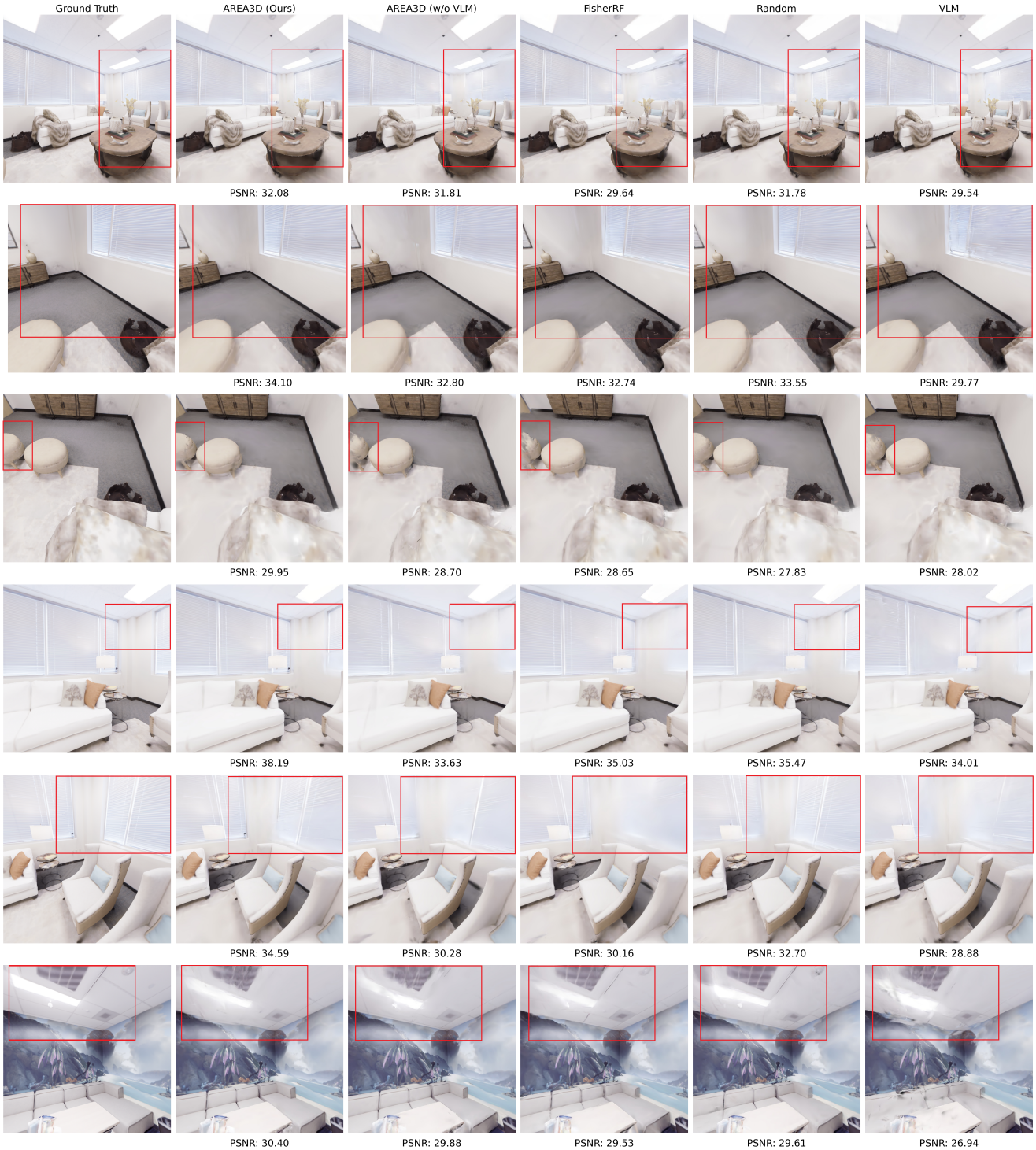


Figure 7. **Novel View Synthesis Results of different policies in scene-level.**

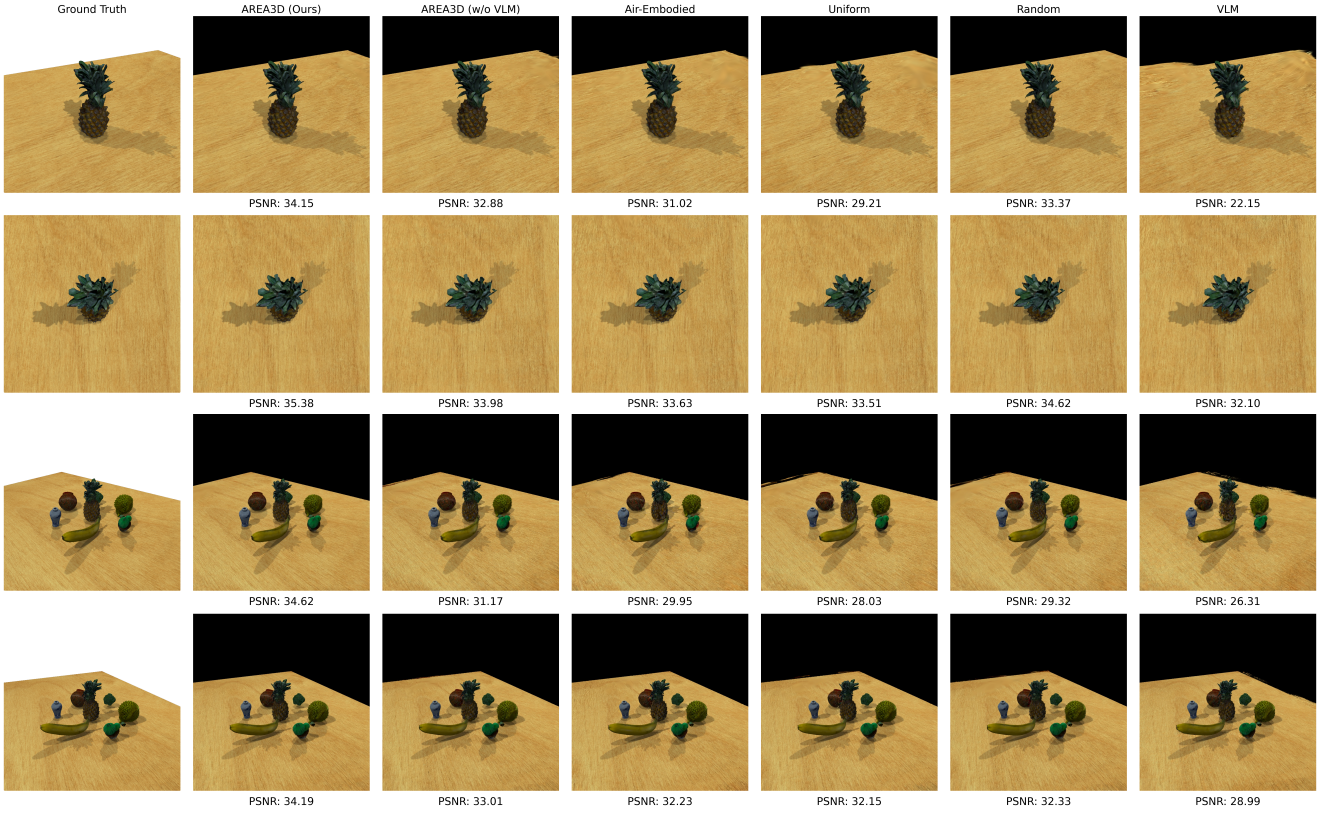


Figure 8. **Novel View Synthesis Results of different policies in object-level.**