Believing is Seeing: Unobserved Object Detection using Generative Models

Supplementary Material

A. Object, Scene and Parameter Selection Methodology

In this Appendix, we detail the methodologies for scene selection, the filtering process for scenes analyzed, the criteria for selecting images and objects, and the approach for parameter calibration.

RealEstate10k random scene selection methodology. To systematically select diverse scenes, we sampled video frames in the RealEstate10k test dataset, which contains world-to-camera poses and pinhole camera intrinsics [87], built using the ORBSLAM pipeline (with an ambiguous scale) [42, 43]. Due to memory limitations, we downloaded 250 random scenes for analysis. RealEstate10k contains indoor and outdoor scenes but lacks metadata or labels, thus to interpret and filter each scene, our objective was to select representative viewpoints in each scene. We selected these frames, for every scene, at approximately equidistant [9] intervals in $\mathrm{SE}(3)$, between the two farthest camera-to-world poses for each scene. Using these selected images, per scene, we used the ChatGPT-4o API [46] to classify scenes as indoor or outdoor based on these 10 images per scene. For the indoor scenes, we further utilized the API to categorize them into four types: kitchen, study, bedroom, and living room. This automated labeling ensured that our subsequent analysis remained focused and consistent across the relevant indoor scenes.

NYU Depth V2 random scene selection methodology. For the NYU Depth V2 dataset [65], which contains meta-labels, this process was not necessary. From the entire dataset, we selected 50 random scenes that met the following criteria: living rooms, bedrooms, kitchens, and studies. These categories ensured that the selected scenes closely matched the diversity found in RealEstate10k, making the study more homogeneous compared to other available meta-labels in the dataset.

Filtering and selection of randomly selected scenes. We refined the selection of indoor scenes from both datasets based on the following criteria:

- *Scene Size*: Scenes containing fewer than 50 images were excluded to mitigate the risk of reconstruction sparsity.
- *Camera Movement*: To ensure sufficient variation in movement, a minimum threshold was applied to the geodesic distance between the two furthest camera poses, requiring it to exceed 0.01 units.
- *Semantic Content*: Scenes were filtered based on object detection using YOLOv8x [26, 53] on COCO object classes [34]. Specifically, scenes were excluded if fewer than 50% of frames contained objects detected with a confidence score ≥ 0.5.

Out of the multiple scenes that met these criteria in each dataset, we randomly selected 10 scenes from both datasets for our study. For the two datasets the selected scenes, target objects, and the input images are tabulated in Tab. 4.

End-to-end object selection pipeline. We deployed an end-to-end automated object selection pipeline on the randomly acquired scenes, logging the occurrence frequency of detected objects across the subsets of both datasets. From the combined scenes of NYU Depth V2 and RealEstate10k scenes, we selected the 20 most frequently detected object classes among the 80 classes in the COCO dataset [34]. For our analysis, we focused on 10 random object categories from these 20 classes that were present in at least one of the chosen scenes in the combined choices of scenes from both datasets. The COCO object categories analyzed include: *refrigerator, TV, bed, chair, sink, oven, book, laptop, couch, and door*. We used Yolov8x [26, 53] for its strong adaptability for images of various resolutions and superior generalizability. Note that all 2D/3D cases analyzed were manually inspected and verified against YOLOv8x annotations to ensure high quality ground truth annotations.

Context image-object pair selection for the study. For each scene, we select a single image based on specific criteria to ensure the target object is effectively represented within the scene. The selection process involves two criteria. First, an image is selected if the target object is absent in the 360×360 center-crop but present in the 360×640 full image. Alternatively, an image is chosen if the target object is not visible in the 360×640 frame but appears in more than 50% of the scene's frames at a confidence ≥ 0.5. This criterion accounts for objects that may be occluded or located entirely outside the camera's frustum while still present within the scene. These criteria result in a large set of image-object pairs across scenes, meeting at least one of the specified conditions. For the NYU Depth V2 dataset, the fraction of available cases is relatively higher, which is expected due to slower average sequential camera movements compared to the RealEstate10k dataset. From this filtered set, we select 10 random test image-object pairs per dataset. To avoid cross-fading or sudden movement artifacts, images are not chosen from the first or last 10 timestamps, where such transitions were qualitatively observed in some camera trajectories. For the RealEstate10k dataset, there are 31 scene-object pairs where the ground truth observa-

Table 4. Scene selected from RealEstate10k [87] and NYU Depth V2 [65] datasets.

RealEstate10k Dataset [87]		NYU Depth V2 Dataset [65]	
Scene ID	Image ID	Scene ID	Image ID
2e4013ea92d04301	119586133	Living Room 0004	1295148543.251260-1026494144
2bec33eeeab0bb9d	34768067	Kitchen 0040	1315269892.882236-1150326380
2e64a2d17f9a76f7	162629000	Living Room 0016	1300200232.988284-1300278508
2b625e92f2cf9de4	51384667	Living Room 0010	1295836465.564725-1670107084
2cb9869cb05a9a01	77786042	Living Room 0002	1294890229.045795-2653268294
3c64a373bc1c53bd	199767000	Bedroom 0025	1315330245.479316-1684325155
ff6d8ab35e042db5	142142000	Bedroom 0029	1315423943.586243-93796617
2bd7cee1fa9c8996	51133333	Kitchen 0024	1315441158.531288-3169924603
3de41ace235a3a13	49616000	Kitchen 0031	1315165725.285327-3895871610
2d6d5e82bda0611c	153253000	Study 0003	1300708629.505940-4057834691

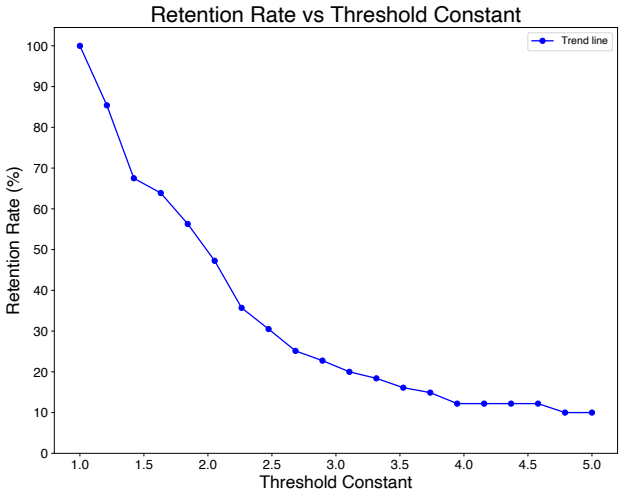


Figure 7. Threshold Multiplier λ vs the Retention Rate of points averaged across all predicted grids in 2D, 2.5D and 3D.

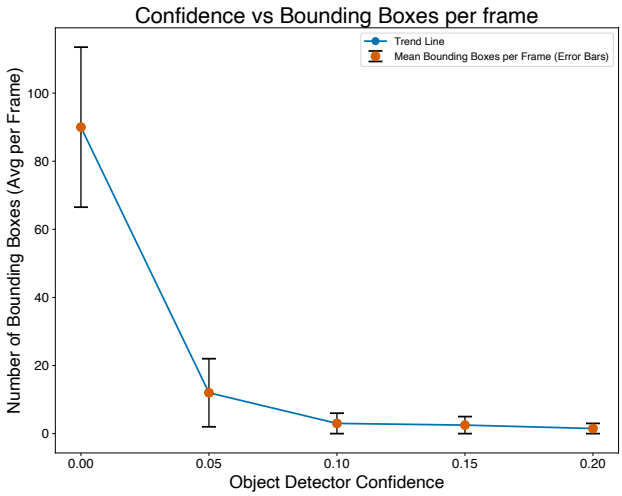


Figure 8. Calibration of Object Detector Confidence using Bounding Boxes per Frame.

tion captures a relevant COCO object in both 2D and 2.5D representations, with 37 cases observed in 3D. In contrast, the NYU Depth V2 dataset contains 19 scene-object pairs captured within the frustum in 2D and 2.5D, with 26 cases in 3D.

Parameter calibration methodology. As shown in Fig. 8, we calibrated the detector confidence by analyzing the average bounding boxes per frame across 100 randomly selected videos from the indoor RealEstate10k test set out of the 250 downloaded videos. We utilize the same heuristics across both the datasets. A threshold of 0.1 achieves a balanced trade-off between sensitivity and precision with stable variance. All other parameters, followed the YOLOv8x default parameters [26]. For the threshold of the metric, we use a multiplier as $\tau := \lambda/|\mathcal{X}| > 1/|\mathcal{X}|$. To calibrate this value, we use the average retention rates across all points in all predicted grids pre-normalization to post-normalization across the RealEstate10k dataset (2D and 3D inclusive) and plot it as a percentage. We see a breaking point at $\lambda \approx 1.4$, as shown in Fig. 7, which informs our choice of the heuristic.

B. 3D scene reconstruction details

In this appendix we outline the 3D scene reconstruction methodology for the two datasets.

RealEstate10k 3D reconstruction. We used *SIFT* [39] to extract up to 50,000 features per image across multi-

ple GPUs. These features were used for sequential matching [66], followed by point triangulation using the camera parameters provided by the RealEstate10k dataset [87]. This process yielded sufficient matches for patch-match stereo reconstruction[20, 61, 62], though the resulting 3D model remained relatively sparse [87].

NYU Depth V2 3D reconstruction. The NYU Depth V2 dataset posed significant challenges for reconstruction due to the absence of ground-truth poses. Using COLMAP [61] for scene mapping was computationally intensive, so we opted for GLOMAP [48], which significantly expedited the process. We calculated the camera intrinsics in normalized device coordinates (NDC), *i.e.*, clip space, for each reconstructed scene, using the COLMAP pinhole camera model, with the principal points positioned at the image midpoints. Camera extrinsics obtained from the reconstruction were subsequently utilized for downstream tasks. Next, we undistorted the images and applied patch-match stereo reconstruction to generate a dense 3D reconstruction.

Post-reconstruction processing. After dense reconstruction, statistical outlier filtering was applied to reduce noise and improve the quality of the reconstructed 3D point cloud. This process evaluates each point based on distances to its 20 nearest neighbors. Points deviating by more than twice the standard deviation from the mean distance are classified as outliers and are removed. We applied a depth clipping threshold of 10.0 distance units with respect to the input camera pose. Finally, each 3D reconstruction was transformed to align with the camera pose of a selected reference image in each scene. For every point cloud, given the bounding boxes, for the target objects, from the 2D images, we back-project to obtain the target 3D points with their confidences. We voxelize these points as required for the ground truth representation assigning the average non-zero confidence of the points within the voxel, to each voxel center. Details of the other COLMAP reconstruction parameters used will be released with the code.

C. Additional implementation details

In this appendix, we provide additional implementation details, especially on the sampling process and post-processing, filtering, pose selection process, and depth handling techniques employed in our study.

System Requirements. The experiments were conducted on a single machine equipped with an NVIDIA A6000 GPU, an AMD Ryzen Threadripper Pro 5995WX CPU, and 64GB of DDR5 RAM, running Ubuntu 22.04 LTS. From sampling to the final metrics summary, our end-to-end pipeline takes up to 38 minutes utilizing multiprocessing.

Implementation details for Δ. To identify the distribution peaks or modes, we applied spatial filtering within the Moore neighborhood for each grid element [18], using constant zero padding at the edges to detect local maxima. Multiple kernel sizes (3, 5, 7, and 10) were used to ensure robust peak detection across varying scales. To compute the nearest distances between these detected peaks, we employed a `KDTree` data structure.

DFM sample detection implementation. The outputs of the DFM [73] are of resolution 128×128. A key limitation of using an off-the-shelf detection model like Yolov8x is its reduced ability to detect all instances at lower resolutions than its training resolution. To address this, we resize the images from 128×128 to 512×512 for detection, compute the bounding boxes at the higher resolution, and then rescale the bounding box extents back to the original 128×128 resolution (to the nearest integer pixel value) for further processing.

DFM sampling and filtering details. DFM [73] inference was performed using autoregressive sampling with 3 time-steps per sample, generating 30 intermediate frames, with a temperature of 0.85 and a guidance scale of 2.5. To filter out intermediate frames potentially exhibiting noisy or featureless outputs due to unconditioned sampling, we implemented a three-phase filtering pipeline. First, noise was quantified using the Laplacian variance V_L, and samples were discarded if $V_L < 100$ in more than 80% of the frames [50]. Next, Structural Similarity Index (SSIM) was computed between consecutive frames, with videos rejected if SSIM > 0.9 in over 50% of the frames, indicating insufficient variation [76]. Finally, per-pixel average intensity L_i was tracked, and samples were discarded if the variation between consecutive frames remained below 50 units for over 50% of the frames, indicating static content or poor lighting. The inference time required for each sample per target camera pose constitutes a significant limitation of the DFM. Moreover, the filtration step, conducted after sample generation, often extended the overall process to several days to collect 75 samples per input image, with the duration varying between different input images. This computational overhead poses a significant constraint on scaling the sample size.

Choice of poses in $\mathrm{SE}(2)$ for DFM sampling. To balance scene diversity and computational efficiency for the DFM experiments, three target poses were selected for each context image. These poses, as illustrated in Fig. 9, (Pose 1 being the input frame, Fig. 2) were chosen based on their ability to cover the scene without blind spots, defined in terms of (x, y, θ) in Tab. 5.

Table 5. Details of DFM test poses, their coordinates, and spatial coverage in the scene.

Pose	Coordinates (x, y, θ)	Description
Pose 1	$(0, 0, 0°)$	Input pose set at the origin relative to the scene movement.
Pose 2	$(-2, 2, 90°)$	Covers the western (right) boundary and central zones of the scene.
Pose 3	$(2, 2, -90°)$	Focuses on the eastern (left) boundary, ensuring coverage of the left and central areas.
Pose 4	$(0, 5, 180°)$	Captures depth and provides a longitudinal perspective from the south.

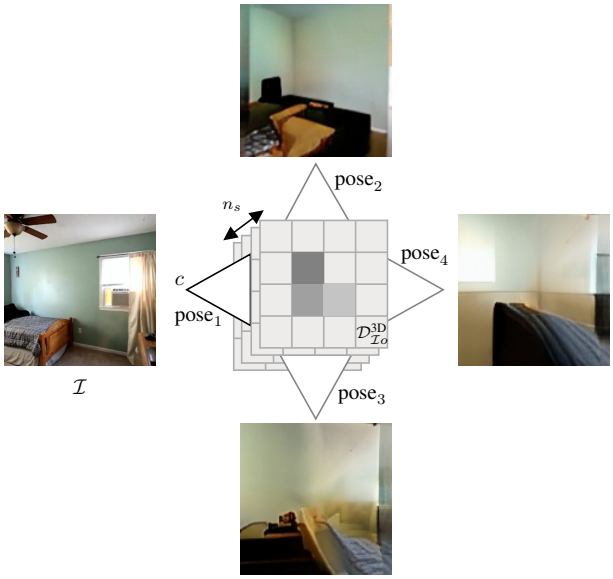


Figure 9. The 3D diffusion pipeline leverages four camera poses to generate up to n_s samples, given the input image $\mathcal{I}$.

(a) Pose 1 (b) Pose 2 (c) Pose 3 (d) Pose 4

Figure 10. DFM samples images from four different camera poses, including the input pose, with a resolution of 128×128 for each image. DFM [73] internally rescaled the original input image of 360×360.

Handling scene depth of the generated samples. For the 3D study, we utilized the aggregated point cloud from the samples from the corresponding poses, and correspondingly all valid 3D points in the ground-truth reconstruction, including occluded parts of the scene, voxelized into a grid of size $20 \times 20 \times 20$, such that the camera position is at the center of the grid. For the SDXL-based study, taking a similar approach as Chen et al. [11] to lift our representation to 2.5D, we generated metric depths up to the far plane of 10 units using DepthAnythingv2 [80, 81], selected for its superior cross-domain generalizability and compatibility with our known camera model. This approach provided depths within the frustum, ensuring scale consistency across the analyses, making sure that the camera centers are aligned to the voxel grid center for the 3D metric computations. For the RealEstate10k dataset experiments, we used the NDC camera intrinsics which is provided, and for the NYU Depth V2 dataset experiments, we used the NDC camera intrinsics that are obtained for each scene, derived from the reconstruction process. In our 2.5D study, ground truth representations and DFM-aggregated point clouds (with associated confidences) were culled within the camera's frustum determined by the camera's pose and intrinsic parameters. Occluded points were removed using Z-culling with a depth buffer, ensuring only the closest points to the camera's projection were retained [2]. These points were scaled and voxelized into a grid of size $10 \times 10 \times 10$ from the 3D grid. The voxel grid was indexed with the input camera positioned at a 5-unit offset from the grid center along the $+z$-axis [3]. The projected pixels and the associated confidences were used for 2D analysis, while the corresponding 3D points were incorporated into the 2.5D study for voxelization. Pixels without any projections are replaced with zero confidence.

SDXL generation details. For our experiments, we used the publicly available Stable Diffusion XL Inpaint-

[2] Efficient occlusion culling methods

[3] Camera Calibration and 3D reconstruction documentation

Table 6. VLM Prompts for Region-Wise Queries

Region	Prompt
Left	"If the image frame is extended by **140 pixels** to the **left**, is it likely that there would be a/an **[OBJECT]** there? **Answer strictly in: Yes/No.**"
Right	"If the image frame is extended by **140 pixels** to the **right**, is it likely that there would be a/an **[OBJECT]** there? **Answer strictly in: Yes/No.**"
Central	"Is it likely that there is a/an **[OBJECT]** within the frame of this image? **Answer strictly in: Yes/No.**"

ing pipeline from the Hugging Face Diffusers library [75]. Future changes to APIs and access to this model may affect reproducibility. During the 2D generation process with SDXL, we used a fixed set of seeds for all experiments. The input image is placed at the center of a 360×640 canvas with 140 pixels of masking on each side of the center 360×360 crop, by the image dimension of 360×140. An additional filter was applied to remove cases where *person(s)* were detected with a confidence score of ≥ 0.5. Both the image and mask are resized to 512×512, and was resized back to 360×640, to match the target dimension. In certain cases, the combination of seed and prompt led to the generation of NSFW content, which the model automatically rejected, returning a blank image for the outpainted regions. These samples were excluded from our analysis. To ensure fairness in aggregation, we resampled till we hit the target number for all sets in every prompt regime. We used 10 different prompts in three different regimes, (1) with object cues in text prompts, (2) without object cues in text prompts and (3) no text prompts. The prompts used were:

1. Extend this indoor scene naturally to the left and right of the given frame (with objects like [OBJECT]).
2. Extend this indoor scene on both sides (with the object: [OBJECT]).
3. Outpaint 140 pixels to the left and right of the room ensuring continuity (using the object: [OBJECT]).
4. Extend the indoor scene horizontally to reveal more of the indoor scene (including a/an [OBJECT]).
5. Add 140 pixels to both sides of the frame to expand the indoor scene (incorporating a/an [OBJECT]).
6. Widen the indoor scene on the left and right (ensuring the object: [OBJECT]).
7. Outpaint 140 pixels to the left and right of the given image horizontally to create a larger view of the indoor scene (featuring the object: [OBJECT]).
8. Expand the boundaries of the frame to the left and right (adding a/an [OBJECT]) to maintain continuity of this indoor scene image.
9. Extend this scene on both sides to display a broader perspective of the indoor scene (including a/an [OBJECT]).
10. Add to the left and right of the image, showcasing more of the indoor setting (with a/an [OBJECT]).

Parentheses indicate object cues, which are included in prompts with cues and omitted in prompts without cues. In the no-text-prompt regime, the prompt field is left empty. To ensure consistency in sampling for the ablation study, we generate 200 such prompts in total.

D. Detailed methodology for VLM sampling

This appendix provides additional details on how we implemented VLM query structuring and response processing for the study.

Query format for VLMs. To reduce the likelihood of incoherent outputs caused by uncontrolled generation for larger tokens, we restricted responses to binary "Yes" or "No", limiting all answers to a single token. This simplification mitigated errors stemming from the unpredictability of free-form responses, especially in tasks requiring spatial reasoning. Trials with spatial queries showed significant variability and low accuracy against ground truth, largely due to per-voxel granularity required for 2.5D and 3D studies. Hence, we defer these experiments to future work. We standardized queries to focus on object presence or relative positions in 2D, tabulated in Tab. 6.

Answer retrieval pipeline. To generate confidence values for each region-object pair, we calculated a granular score for each region (left, right, and within the frame) based on the fraction of "Yes" answers provided by the model, for that region. These scores were softmax normalized across each pixel to ensure consistency, with their sum equal to 1, and treated as model's normalized confidence. The normalized scores formed the basis of the 2D spatio-semantic distribution, $\mathcal{D}_{\mathcal{I}}^{2D}$, with dimensions 360×640, enabling a uniform comparison across models. We employed regular expressions (regex) parsing to automate the extraction of "Yes" or "No" answers from the model's responses. The binary response format minimized parsing errors during answer retrieval. For qualitative analysis, the normalized scores were further re-scaled using a log-scale adjustment to emphasize variations effectively, while maintaining parity with other analyses.

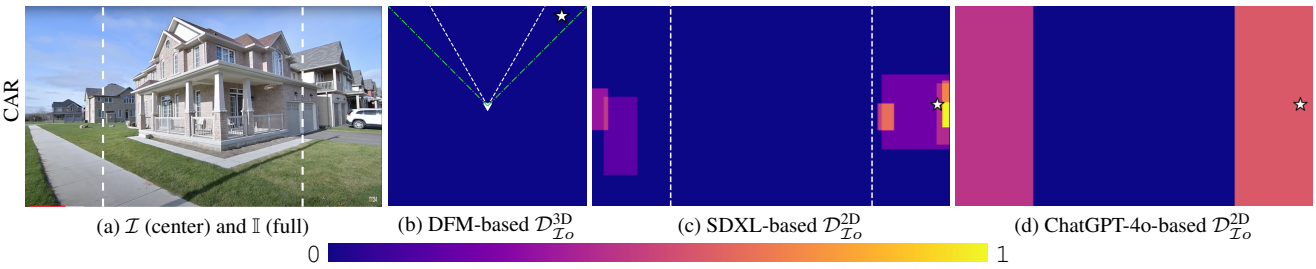


Figure 11. Results for an outdoor scene–object pair. The GT position is denoted by a white star.

Access dates and configurations. Visual-language models (VLMs) were queried using an automated pipeline between November 1st and 2nd, 2024. The latest official APIs were used for Claude 3.5 Sonnet [3] (`claude-3-5-sonnet-20241022`), ChatGPT-4o [46] (`chatgpt-4o-latest`), and Gemini 1.5 Pro [19] (`gemini-1.5-pro-latest`). LLaVa-34B-v1.6 (Nous-Hermes-2-34B) [37, 38], accessed via the Replicate API[4], is open source (unlike the other proprietary models) but was evaluated through APIs, as we could not perform local inference owing to model size.

E. Additional Results

DFM sample and pose ablations. In this section we provide all the complete ablation tables for individual datasets—NYU Depth V2 [65] and RealEstate10k [87]. We tabulate the results of the ablation of Diffusion with Forward Model samples [73] in 2D Tabs. 8 and 9, in 2.5D Tabs. 10 and 11 and in 3D Tabs. 12 and 13. We apply the same random seed to select 10 or 15 samples for random drop-outs, ensuring consistency in the sample selection process, retaining the points and the confidences from the input poses in all cases.

SDXL qualitative analysis. We also provide the breakdown for the analysis of the SDXL-based analysis in 2D Tabs. 14 and 15 and the analysis in 2.5D Tabs. 16 and 17. In Tab. 3, we tabulated the combined ablation study using the total number of valid samples in both datasets. We present the following metrics along with their means and standard deviations: normalized entropy ($\mathcal{H}$), normalized cross-entropy ($\mathcal{H}^{\times}$), normalized nearest neighbor distance as a percentage (Δ), 2D region-wise accuracy ($\mathcal{A}$), and false negative rates (FNR).

Additional qualitative results. In this section, we present additional qualitative results to supplement our analysis. Samples generated using the DFM are illustrated in Fig. 10, while those from SDXL are shown in Fig. 12. As expected, the DFM-based model exhibits poor qualitative performance on the NYU dataset due to its limited ability to generalize to out-of-distribution data [73]. The DFM-based model, as depicted in rows 6 and 7 of Fig. 13, frequently predicts flat distributions. While these distributions capture uncertainty and can localize the ground truth position, their higher uncertainty reflects in the normalization process followed by log-scaling, resulting in a wide probability mass spread across the distribution ($\mathcal{D}^{3D}_{\mathcal{I}o}$). For top-down heatmaps, we visualize the maximum normalized confidence value along the $+y$-axis for each column of the voxel grid. These visualizations may show spillover into regions without detected objects, which is an artifact of the softmax normalization and voxelization processes. In some 2D heatmaps, ground truth positions are omitted when the object is not visible in the ground truth image but is present in the scene. For SDXL-generated samples, we observe that semantic quality is significantly influenced by the guidance scale and the text prompt. When object cues are provided, both qualitative and quantitative performance improve noticeably. Conversely, generation without object cues or prompts yields minimal differences in quantitative results, as both scenarios result in limited detections. The qualitative results for SDXL analysis (column 3 in Figs. 5 and 13) are based on samples generated using prompts that include object cues. In certain SDXL-based heatmaps, spillovers into the central region may occur because the detector perceives the entire object, shared between the input image and the outpainted region, as part of the object.

Failure case analysis. In our current DFM experiments, 57% of failures arise from object prediction errors (non-occluded), while 43% result from detector false negatives in ambiguous contexts (*e.g.*, multiple doors in a corridor). No failures stem from the inability to predict occluded objects. SDXL-based methods with object prompts show no failures. In VLM experiments, all failures are direct prediction errors of the model.

Outdoor Scenes. Our framework is also applicable to outdoor scenes. We demonstrate this with an outdoor scene from the RealEstate10k dataset featuring the COCO object 'car' (Fig. 11, Tab. 7). The SDXL and VLM-based models perform well as expected while the DFM struggles in outdoor settings due to limited training on large-scale outdoor data, retraining which exceeds our computational resources.

[4] Replicate LLaVa-v1.6-34B model API documentation

Table 7. Results for the outdoor scene–object pair in Fig. 11.

Methods	2D Experiment				2.5D Experiment			3D Experiment		
	$\mathcal{H}^\times \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	$\mathcal{A} \uparrow$	$\mathcal{H}^\times \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	$\mathcal{H}^\times \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$
DFM (25 samples, 4 poses)	1.000	1.000	∞	0.670	0.998	1.000	∞	0.998	1.000	∞
SDXL (w. obj prompt)	0.891	0.771	0.000	0.670	0.788	0.962	0.000	–	–	–
ChatGPT-4o	0.880	0.879	0.000	0.670	–	–	–	–	–	–
Claude 3.5 Sonnet	0.880	0.879	0.000	0.670	–	–	–	–	–	–
Gemini 1.5 Pro	0.994	0.993	0.000	0.670	–	–	–	–	–	–
LLaVa-v1.6-34b	0.999	0.998	0.000	0.670	–	–	–	–	–	–

Table 8. 2D Metrics for RealEstate10k with lower number of poses and samples, used to study the trend.

Methods	$\mathcal{H}^\times \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	FNR $\downarrow$	$\mathcal{A} \uparrow$
DFM ($n_s = 25, k = 4$)	1.555 ± 1.460	0.774 ± 0.119	5.990 ± 39.470	0.032	0.747 ± 0.139
DFM ($n_s = 25, k = 3$)	1.877 ± 1.327	0.818 ± 0.112	18.136 ± 32.482	0.121	0.626 ± 0.107
DFM ($n_s = 25, k = 2$)	2.328 ± 1.041	0.902 ± 0.067	14.037 ± 42.005	0.868	0.581 ± 0.182
DFM ($n_s = 15, k = 4$)	1.736 ± 0.414	0.898 ± 0.117	15.334 ± 23.380	0.605	0.573 ± 0.198
DFM ($n_s = 10, k = 4$)	2.247 ± 1.206	0.923 ± 0.014	14.082 ± 29.083	0.711	0.550 ± 0.337

Table 9. 2D Metrics for NYU Depth V2 with lower number of poses and samples, used to study the trend.

Methods	$\mathcal{H}^\times \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	FNR $\downarrow$	$\mathcal{A} \uparrow$
DFM ($n_s = 25, k = 4$)	1.661 ± 1.336	0.696 ± 0.128	6.217 ± 36.112	0.0556	0.765 ± 0.236
DFM ($n_s = 25, k = 3$)	2.053 ± 0.351	0.929 ± 0.145	20.459 ± 35.216	0.173	0.589 ± 0.340
DFM ($n_s = 25, k = 2$)	2.691 ± 1.187	0.978 ± 0.072	15.871 ± 45.028	0.900	0.545 ± 0.190
DFM ($n_s = 15, k = 4$)	1.915 ± 0.497	0.969 ± 0.030	17.242 ± 25.764	0.558	0.623 ± 0.353
DFM ($n_s = 10, k = 4$)	2.498 ± 1.354	0.998 ± 0.012	15.936 ± 31.011	0.739	0.614 ± 0.356

Table 10. 2.5D Metrics for RealEstate10k with lower number of poses and samples, used to study the trend.

Methods	$\mathcal{H}^\times \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	FNR $\downarrow$
DFM ($n_s = 25, k = 4$)	1.955 ± 1.033	0.530 ± 0.441	5.128 ± 3.042	0.032
DFM ($n_s = 25, k = 3$)	2.018 ± 0.053	0.814 ± 0.080	12.889 ± 1.067	0.121
DFM ($n_s = 25, k = 2$)	2.586 ± 2.650	0.996 ± 0.003	13.461 ± 1.552	0.868
DFM ($n_s = 15, k = 4$)	2.113 ± 0.778	0.877 ± 0.067	9.018 ± 6.332	0.605
DFM ($n_s = 10, k = 4$)	2.498 ± 1.354	0.997 ± 0.002	18.018 ± 9.607	0.711

Table 11. 2.5D Metrics for NYU Depth V2 with lower number of poses and samples, used to study the trend.

Methods	$\mathcal{H}^\times \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	FNR $\downarrow$
DFM ($n_s = 25, k = 4$)	1.785 ± 2.101	0.610 ± 0.207	4.214 ± 2.803	0.0556
DFM ($n_s = 25, k = 3$)	2.067 ± 0.093	0.798 ± 0.089	11.442 ± 1.305	0.152
DFM ($n_s = 25, k = 2$)	2.523 ± 2.301	0.990 ± 0.005	12.786 ± 1.781	0.837
DFM ($n_s = 15, k = 4$)	2.184 ± 0.652	0.867 ± 0.079	8.336 ± 6.174	0.772
DFM ($n_s = 10, k = 4$)	2.451 ± 1.231	0.995 ± 0.003	17.124 ± 9.403	0.808

Table 12. 3D Ablation study of DFM with different configurations of n_s and k on the RealEstate10k dataset.

Methods	$\mathcal{H}^{\times} \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	FNR $\downarrow$
DFM ($n_s = 25, k = 4$)	2.471 ± 4.415	0.303 ± 0.202	9.190 ± 6.223	0.0174
DFM ($n_s = 25, k = 3$)	2.774 ± 7.210	0.412 ± 0.201	10.344 ± 7.004	0.0412
DFM ($n_s = 25, k = 2$)	4.102 ± 4.815	0.508 ± 0.210	12.001 ± 7.350	0.321
DFM ($n_s = 15, k = 4$)	3.718 ± 4.950	0.821 ± 0.202	11.210 ± 6.702	0.0645
DFM ($n_s = 10, k = 4$)	3.210 ± 3.550	0.965 ± 0.050	13.105 ± 7.219	0.184

Table 13. 3D Ablation study of DFM with different configurations of n_s and k on the NYU Depth V2 dataset.

Methods	$\mathcal{H}^{\times} \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	FNR $\downarrow$
DFM ($n_s = 25, k = 4$)	2.062 ± 3.915	0.541 ± 0.374	3.948 ± 5.801	0.000
DFM ($n_s = 25, k = 3$)	3.876 ± 6.235	0.768 ± 0.426	6.572 ± 6.245	0.138
DFM ($n_s = 25, k = 2$)	6.125 ± 7.325	0.983 ± 0.492	9.815 ± 7.412	0.453
DFM ($n_s = 15, k = 4$)	4.521 ± 5.813	0.912 ± 0.628	8.210 ± 6.925	0.084
DFM ($n_s = 10, k = 4$)	6.832 ± 3.750	0.965 ± 0.702	11.582 ± 7.625	0.288

Table 14. Comparative analysis of metrics across different *SDXL 2D analysis* for the RealEstate10k dataset.

Methods	$\mathcal{H}^{\times} \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	FNR $\downarrow$	$\mathcal{A} \uparrow$
SDXL w/ object cues	1.257 ± 2.033	0.848 ± 0.150	0.446 ± 25.490	0.039	0.918 ± 0.147
SDXL w/o object cues	1.888 ± 1.904	0.882 ± 0.115	6.274 ± 14.890	0.554	0.688 ± 0.272
SDXL w/o prompts	2.583 ± 4.612	0.887 ± 0.106	6.901 ± 12.663	0.589	0.773 ± 0.276

Table 15. Comparative analysis of metrics across different *SDXL 2D analysis* for the NYU Depth V2 dataset.

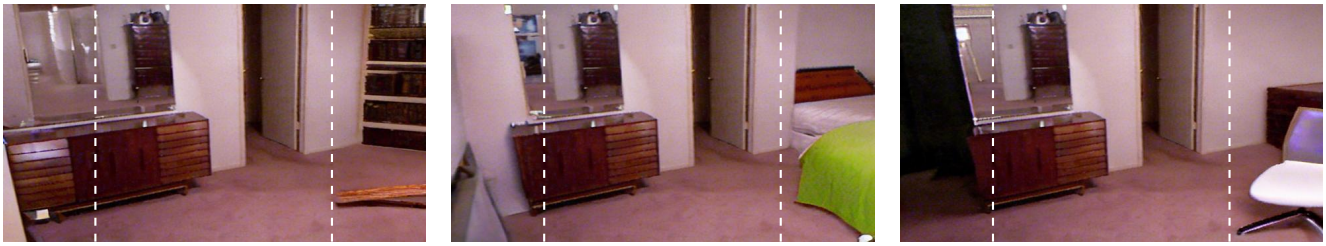
Methods	$\mathcal{H}^{\times} \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	FNR $\downarrow$	$\mathcal{A} \uparrow$
SDXL w/ object cues	1.223 ± 2.107	0.778 ± 0.212	0.510 ± 26.929	0.054	0.944 ± 0.142
SDXL w/o object cues	2.601 ± 2.906	0.891 ± 0.011	5.649 ± 15.050	0.378	0.761 ± 0.284
SDXL w/o prompts	3.555 ± 2.374	0.988 ± 0.001	5.798 ± 15.107	0.374	0.757 ± 0.285

Table 16. Comparative analysis of metrics across different SDXL *2.5D analysis* for the RealEstate10k dataset.

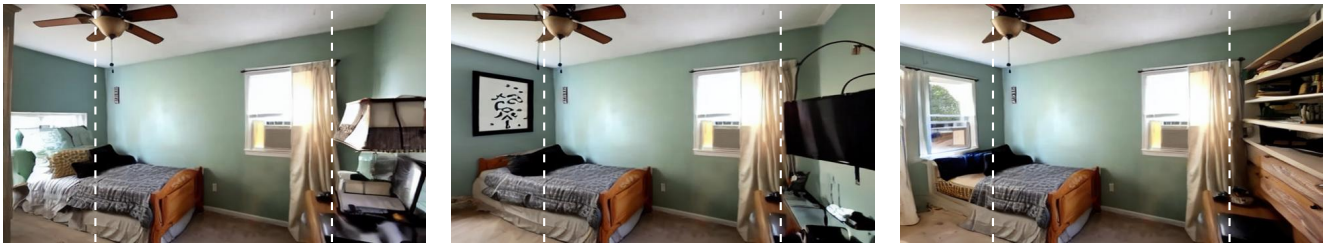
Methods	$\mathcal{H}^{\times} \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	FNR $\downarrow$
SDXL w/ object cues	1.752 ± 2.043	0.617 ± 0.305	6.277 ± 2.229	0.000
SDXL w/o object cues	1.833 ± 2.011	0.886 ± 0.104	18.144 ± 6.148	0.554
SDXL w/o prompts	1.874 ± 1.988	0.931 ± 0.016	18.533 ± 9.178	0.589

Table 17. Comparative analysis of metrics across different SDXL *2.5D analysis* for the NYU Depth V2 dataset.

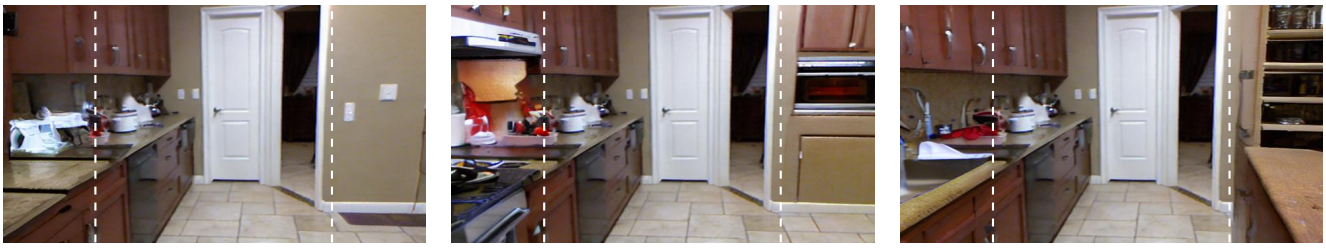
Methods	$\mathcal{H}^{\times} \downarrow$	$\mathcal{H} \downarrow$	$\Delta(\%) \downarrow$	FNR $\downarrow$
SDXL w/ object cues	1.533 ± 2.916	0.655 ± 0.323	5.245 ± 2.351	0.000
SDXL w/o object cues	2.612 ± 1.398	0.818 ± 0.102	12.637 ± 9.184	0.378
SDXL w/o prompts	2.807 ± 1.616	0.987 ± 0.010	16.998 ± 7.356	0.374



(a) No object prompt (b) With object prompt: bed (c) With object prompt: chair



(d) No object prompt (e) With object prompt: TV (f) With object prompt: laptop



(g) No object prompt (h) With object prompt: oven (i) With object prompt: sink

(j) No object prompt (k) With object prompt: book (l) With object prompt: TV

(m) No object prompt (n) With object prompt: oven (o) With object prompt: refrigerator

Figure 12. **Samples from SDXL.** We prompted SDXL with and without object cues. The image within the dotted lines is the input image.

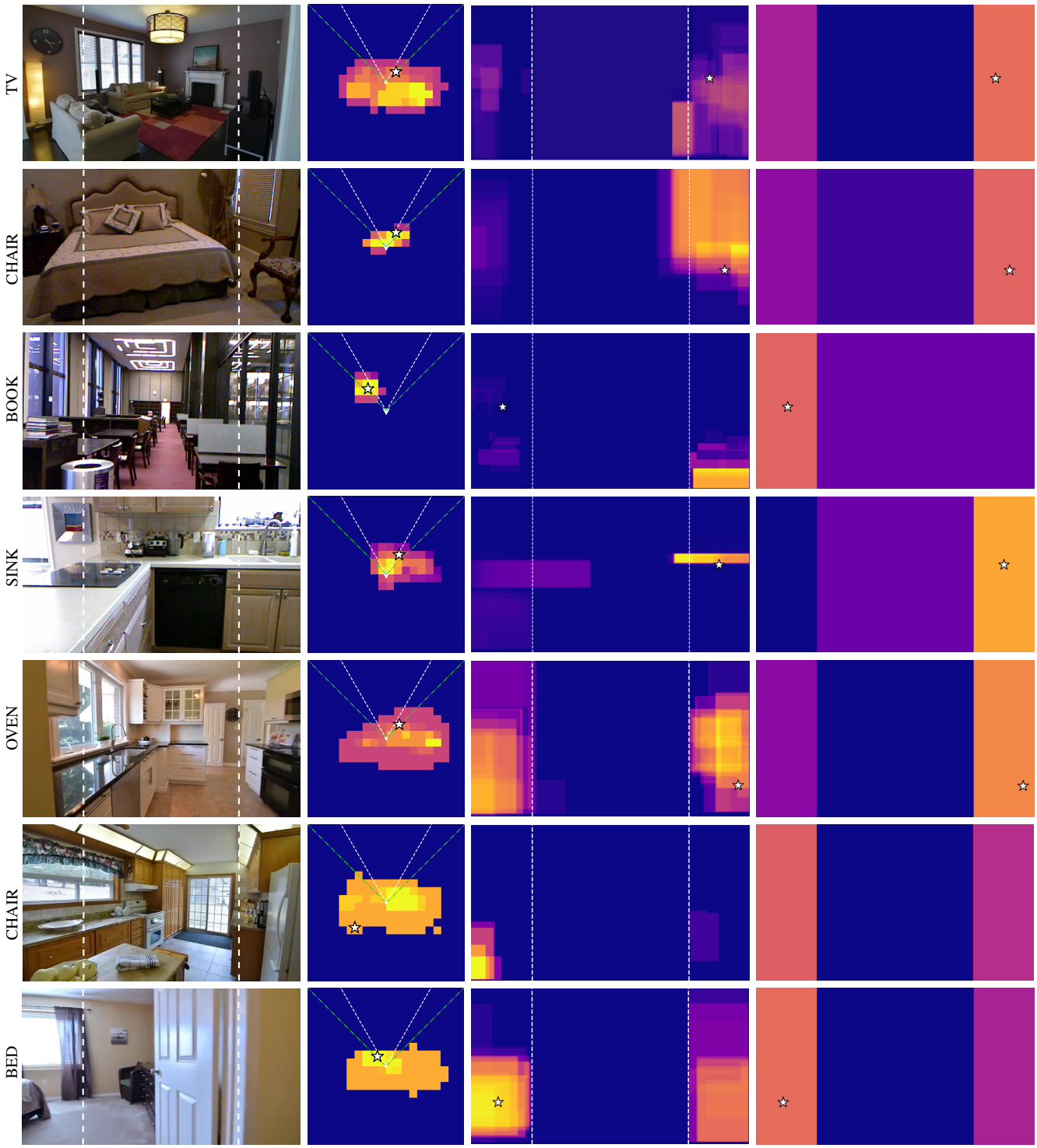


Figure 13. **Additional qualitative results.** We provide additional results for different objects and their corresponding heatmaps. White stars indicate ground truth position if applicable on the corresponding heatmap(s).