

Informative Object-centric Next Best View for Object-aware 3D Gaussian Splatting in Cluttered Scenes

Seunghoon Jeong¹, Eunho Lee¹, Jeongyun Kim², and Ayoung Kim^{2*}

Abstract—In cluttered scenes with inevitable occlusions and incomplete observations, selecting informative viewpoints is essential for building a reliable representation. In this context, 3D Gaussian Splatting (3DGS) offers a distinct advantage, as it can explicitly guide the selection of subsequent viewpoints and then refine the representation with new observations. However, existing approaches rely solely on geometric cues, neglect manipulation-relevant semantics, and tend to prioritize exploitation over exploration. To tackle these limitations, we introduce an instance-aware Next Best View (NBV) policy that prioritizes underexplored regions by leveraging object features. Specifically, our object-aware 3DGS distills instance-level information into one-hot object vectors, which are used to compute confidence-weighted information gain that guides the identification of regions associated with erroneous and uncertain Gaussians. Furthermore, our method can be easily adapted to an object-centric NBV, which focuses view selection on a target object, thereby improving reconstruction robustness to object placement. Experiments demonstrate that our NBV policy reduces depth error by up to 77.14% on the synthetic dataset and 34.10% on the real-world GraspNet dataset compared to baselines. Moreover, compared to targeting the entire scene, performing NBV on a specific object yields an additional reduction of 25.60% in depth error for that object. We further validate the effectiveness of our approach through real-world robotic manipulation tasks.

I. INTRODUCTION

Accurate perception of the operating environment including the geometry of objects is essential for successful robotic manipulation. While many prior works [1, 2] estimate grasping points from a single view depth image, real-world cluttered scenes often contain stacked or irregularly distributed objects, where severe occlusions occur. In such cases, predefined viewpoints may either lack necessary information or include redundant observations, making them insufficient and ineffective for reliable manipulation [3, 4]. This motivates the need for a robot to *actively* move and select informative viewpoints based on the scene context or

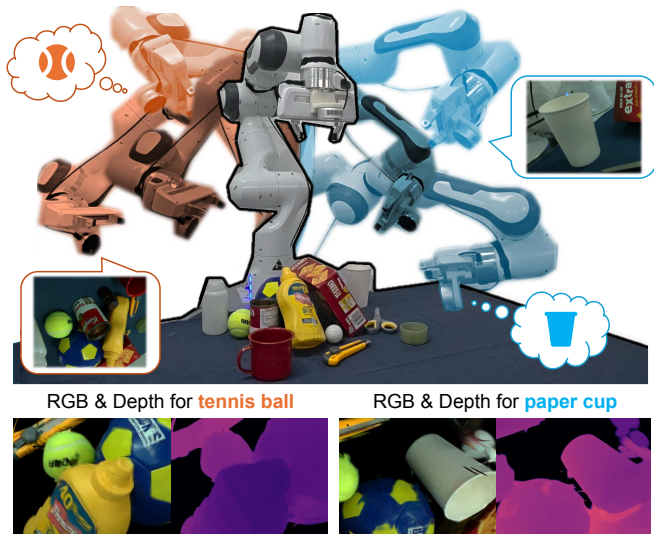


Fig. 1: Object-centric NBV for manipulation in cluttered scenes. Instead of reconstructing the entire scene, we prioritize task-relevant regions surrounding the target object (tennis ball (red) or cup (blue)). Using these selected views with our object-aware 3DGS yields improved novel view synthesis and depth reconstruction, as shown below.

the target object and incrementally refine its understanding of the scene from the newly captured perception.

Identifying a minimal set of training views that preserves reconstruction quality frames the problem of view selection and Next Best View (NBV) planning. Voxel-based methods [5, 6] estimate volumetric completeness via per-voxel occupancy, but require high memory at finer resolution and do not account for photometric quality. Bayes' Rays [7] addresses this limitation by parameterizing perturbations to Neural Radiance Fields (NeRF) [8] parameters to quantify uncertainty, yet its slow rendering speed makes it unsuitable for robotic tasks.

Compared to these approaches, methods [9–11] based on 3D Gaussian Splatting (3DGS) [12] offer the advantages of using an explicit representation, which enables per-Gaussian uncertainty estimation and faster rendering. By leveraging the Jacobian of Gaussian parameters with respect to the rendering output, these methods avoid heuristic modelings and select views in a direct and efficient manner. However, since well-observed Gaussians with low uncertainty still dominate the rendering contribution, these NBV strategies tend to focus on exploitation rather than exploration. Moreover, these methods are designed for reconstructing the entire scene,

[†]This work was partly supported by Hyundai Motor Company and Kia, and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) No.2022-0-00480, Development of Training and Inference Methods for Goal-Oriented Artificial Intelligence Agents and the Korea government(MSIT) [NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)].

¹S. Jeong and E. Lee are with the Interdisciplinary Program in Artificial Intelligence, SNU, Seoul, S. Korea [shoon0602, eunho1124]@snu.ac.kr

²J. Kim and A. Kim are with the Dept. of Mechanical Engineering, SNU, Seoul, S. Korea [jeongyunkim, ayoungk]@snu.ac.kr

making them less robust for manipulation tasks targeting objects that are occluded, small, or far from the center. In such cluttered environments, robots often require object-centric NBV planning that can focus on a specific object.

To tackle these limitations, we propose an informative, object-centric NBV policy integrated with object-aware 3DGS. Our key insight is to inject instance-aware confidence score into uncertainty quantification so that information gain and view selection prioritize uncertain, task-relevant regions. This enables NBV for both whole scene coverage and target-conditioned inspection with better novel view synthesis and depth reconstruction as shown in Fig. 1. Our object-aware 3DGS distills object instance masks from image foundation models [13–15] into a one-hot object vector, reconstructs scenes in fewer iterations than optimization using only RGB images and yields explicit per-Gaussian object features that guide NBV. We validate our approach through substantial improvements in both novel view synthesis and depth estimation, as well as successful deployment in real-world robotic arm experiments. Our main contributions are as follows:

- **Instance-aware Information Gain for Next Best View:** By weighting each Gaussian’s information gain with a confidence score from object features, our NBV planner prioritizes poorly observed yet task-relevant regions, encouraging exploration.
- **Object-aware 3D Gaussian Splatting and Fast Scene Refinement:** We introduce an object-aware 3DGS that reconstructs and segments cluttered scenes in just a few iterations with low training time, by fusing incoming RGB images and object instance masks through a one-hot object vector.
- **Object-centric Reconstruction and Manipulation:** Beyond whole-scene modeling, our system can focus NBV around user-specified objects and produces optimal 6-DoF grasp poses in cluttered scenes.

II. RELATED WORK

A. View Selection and Next Best View

View selection and NBV have received significant attention as approaches for identifying the most informative next viewpoint and selecting an optimal set of input images that effectively summarize the scene content. Previous approaches in radiance field-based NBV have included methods that treat uncertainty directly as a learnable parameter [16] or exploit heuristic information gain functions [17]. However, these methods fail to fully capture epistemic uncertainty arising from insufficient knowledge.

Some approaches [7, 9–11], on the other hand, have proposed quantifying uncertainty using the Hessian matrix of the negative log-likelihood and successfully applied to both NeRF and 3DGS. Specifically, FisherRF [9] directly measured uncertainty by leveraging the Jacobian of rendering outputs with respect to the Gaussian primitives in 3DGS. Subsequently, POp-GS [10] introduced concepts from optimal experimental design [18] into information gain calculations, partially accounting for parameter correlations.

Next Best Sense [11] further expanded its applicability by incorporating depth information into the information gain.

However, conventional information gain-based NBV methods often display a strong bias toward regions that have already been sufficiently observed, while leaving underexplored areas inadequately covered. This imbalance between exploration and exploitation limits their effectiveness in completing the scene understanding. We address this by incorporating per-Gaussian confidence weighting to guide viewpoint selection more evenly.

B. Feature Embedded 3D Representation for Manipulation

Recent studies [19, 20] have demonstrated that applying feature distillation to volumetric rendering can jointly capture semantic and geometric information, making it highly suitable for robotic manipulation. This idea has also been extended to 3DGS, where several works [21, 22] have customized the CUDA rasterizer to incorporate features from various language-based models such as CLIP [23], SAM [14], and GroundingDINO [13]. GraspSplats [24] leverages masks obtained from MobileSAM [25] and their CLIP embeddings to design a hierarchical feature representation, enabling zero-shot object manipulation. However, directly injecting such features into Gaussians is challenging because their high dimensionality complicates optimization, slows processing, and makes it difficult to distinguish among semantically similar instances.

Accordingly, research has also emerged focusing on faster reconstruction using simpler features. Gaussian Grouping [26] introduced identity encodings to cluster Gaussians belonging to the same object, while ObjectGS [27] and TRAN-D [28] segmented and tracked objects using Grounded SAM [15] and then optimized a one-hot vector for each object. Similarly, we employ a one-hot vector as supervision for objects, and we further extend its use by leveraging it as a confidence measure integrated into NBV planning. This also serves as the cornerstone of our object-centric reconstruction.

III. METHOD

In this section, we present our proposed NBV pipeline for cluttered scenes, consisting of three components: object-aware 3D Gaussian Splatting for scene reconstruction, an improved uncertainty and information gain quantification for exploring unseen regions, and an object-centric view selection method for target-focused robotic manipulation.

A. Object-aware 3D Gaussian Splatting

1) *One-hot Object Vector for Instance Segmentation:* With recent advances in 2D segmentation, obtaining object masks has become practical using image foundation models. Here, we describe how to integrate a given object mask into the 3DGS framework by adding a lightweight feature to each Gaussian. Assume the scene contains n distinct objects, with object IDs assigned from 1 to n and the background labeled as 0. Each pixel in the mask is represented as a one-hot vector of dimension $n + 1$, where each channel corresponds to a specific object class or the background.

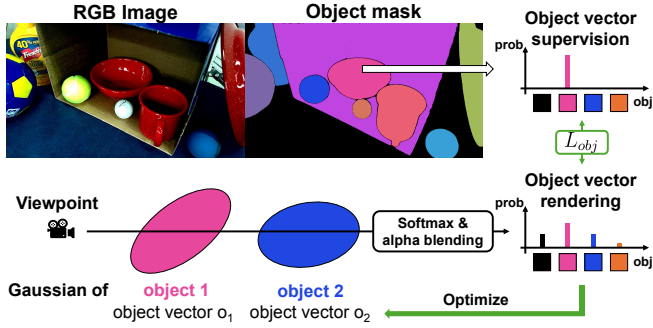


Fig. 2: Visualization of the rendering and learning process of the one-hot object vector. Each Gaussian stores logits for $n + 1$ classes, including n object categories and the background. The logits are passed through a softmax function and alpha-blended in the same manner as RGB in 3DGS. After normalization, this yields the per-pixel class probability along each ray. By supervising these probabilities with one-hot vectors obtained from instance masks, the logits for each Gaussian can be optimized.

$$O' = \sum_i \text{softmax}(o_i) \alpha_i T_i, \quad T_i = \prod_{j=1}^{i-1} (1 - \alpha_j). \quad (1)$$

3DGS represents a 3D scene as a set of Gaussian ellipsoids, each parameterized by a center position $\mu \in \mathbb{R}^3$, scale $s \in \mathbb{R}^3$, rotation $r \in \mathbb{R}^4$, opacity $\sigma \in \mathbb{R}$, and spherical harmonic coefficients $c \in \mathbb{R}^{3(d+1)^2}$ where d is the degree of spherical harmonic. We assign an additional object feature vector $o_i \in \mathbb{R}^{n+1}$ to each Gaussian and optimize it by comparing the rendered results against the ground-truth object masks as shown in Fig. 2. We apply a softmax to each o_i , then perform alpha-weighted blending across Gaussians to obtain a per-pixel object vector.

$$O = O' + [1, 0, \dots, 0] \cdot (1 - \mathbf{1}^\top O') \quad (2)$$

Since the blended object vector often sums to less than 1 due to incomplete coverage, we add the residual value to the background channel to ensure that it sums to 1. This lets each value O_k be interpreted as the probability of the pixel belonging to object class k . We supervise the object features with the L1 loss and the Dice loss [29]:

$$L_{obj} = (1 - \lambda_{Dice}) \times L_{1,obj} + \lambda_{Dice} \times L_{Dice}, \quad (3)$$

and the softmax of the supervised object feature yields the per-Gaussian object probability $p_i = \text{softmax}(o_i)$.

Segmentation masks primarily remove background in 3DGS. Assigning a default color to background pixels suppresses the opacity of background Gaussians, markedly reducing both the total number of Gaussians and floaters. In addition, we prune any Gaussian whose maximum object class probability $\max_k(p_{i,k})$ falls below a threshold δ_{obj} near the uniform prior $1/(n+1)$, thereby eliminating unobservable or irrelevant regions.

2) *Fast Scene Refinement with New Images*: Our reconstruction begins with few initial views with predefined camera pose. Given m views, we run optimization for $100m$ iterations and then proceed with the NBV selection, as described in Section 3.B, based on the current reconstruction.

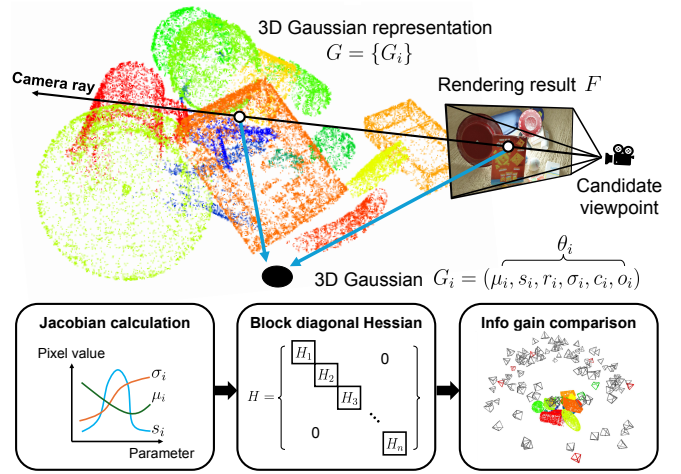


Fig. 3: Illustration of the overall NBV system. Given an object-segmented 3D Gaussian map G and candidate camera pose, the CUDA rasterizer computes the Jacobian of the rendering with respect to Gaussian parameters. Correlations among parameters of each Gaussian are captured in a block diagonal Hessian, which is then used to estimate information gain. By comparing gains across candidates, the most informative view is selected, without requiring GT images of candidate views.

To enable faster optimization during the early stages, we set the spherical harmonic (SH) degree $d = 0$ until the final view is added. Additionally, to avoid the influence of local minima or overfitting effects from previous reconstructions, we reinitialize the Gaussians with new random points at each step. The overall loss function is based on the standard 3DGS loss, with an additional term introduced for supervising the object vector associated with each Gaussian.

$$L_{overall} = (1 - \lambda)L_1 + \lambda L_{SSIM} + \lambda_{obj} L_{obj} \quad (4)$$

Once the designated number of views has been collected, the final reconstruction can be achieved in 3,000 iterations, despite raising the SH degree back to 3 because we integrate object segmentation into the optimization. Even when considering all iterations performed before the full view set is acquired, the total iterations remains under 10,000, compared to 30,000 iterations in the standard 3DGS setup.

B. Next Best View Policy with Uncertainty Quantization

1) *Information Gain of the New Image*: Information gain $IG(T, c)$ with training view set T and candidate c can be calculated by comparing quantified uncertainty value with and without the new image. Candidate which maximizes the information gain among the candidate view set C includes to the training view set:

$$IG(T, c) = f(T \cup c) - f(T), \quad c = \underset{c \in C}{\operatorname{argmax}} IG(T, c). \quad (5)$$

Following POP-GS [10], we assume i.i.d. Gaussian pixel noise and adopt a maximum-likelihood formulation as shown in Fig. 3. Maximizing the likelihood is equivalent to minimizing the least-squares objective $\frac{1}{2} \|e\|^2$ for the 3DGS model h , where e denotes the per-pixel residual between the observation and the rendering. Linearizing $h(\theta)$ around

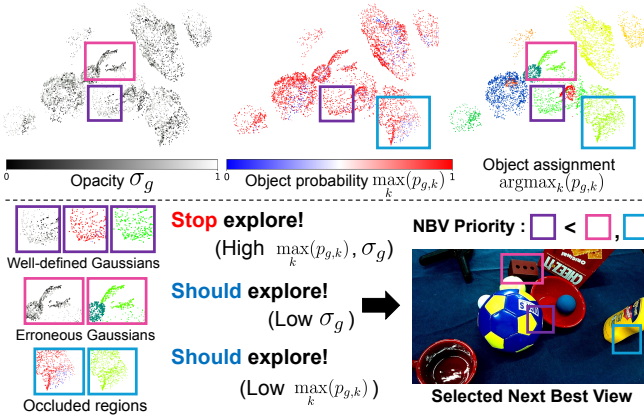


Fig. 4: Illustration of object confidence scores for exploration. The top row shows the 3D representation of a captured scene trained with 6 views, visualized as Gaussian opacity, object probability, and object index. Rectangles of the same color highlight different cases: purple indicates well-optimized regions that require no further observation, while pink and blue denote areas either unseen or poorly localized, thus requiring exploration. We leverage the intuition that regions needing additional views exhibit low opacity and low object probability, and incorporate this insight into our NBV formulation.

the current estimate yields the Gauss–Newton approximation with the Hessian $H \approx J^T J$, where J is the Jacobian of the rendering with respect to Gaussian parameters. We use H^{-1} as a proxy for epistemic uncertainty and score candidate views by their expected reduction of this covariance while preserving fidelity on the training views.

Several choices exist for f such as the trace or determinant of the Hessian, but we adopt the log-determinant $\log(\det(H))$ for numerical stability and computational convenience. Given a view set V , the matrix H is computed as the sum of $J^T J$ over all views in V :

$$f(V) = \log(\det(H_V)) = \log(\det(\sum_{v \in V} J_v^T J_v)). \quad (6)$$

For n Gaussians each with l parameters, the full Information matrix H_V has a dimensionality of $nl \times nl$, which results in a substantial computational burden when computing its determinant. To alleviate this, we consider only the intra-Gaussian correlations among parameters, approximating H_V as a block diagonal matrix. This allows us to compute the determinant as the product of per-Gaussian blocks, reducing the computational complexity from $O(l^3 n^3)$ to $O(l^3 n)$. This block-diagonal approximation is implemented by modifying the original CUDA rasterizer in 3DGS.

2) *Confidence-weighted Information Gain for Exploration*: Gaussians corresponding to regions that require further observation through additional views tend to have low contributions in the training views. As a result, they generally exhibit small gradients, low opacity σ_i , and uniform object probability p_i that are not strongly biased toward a single object. Consequently, in Hessian-based NBV formulations, the Jacobians of well-observed primitives with low uncertainty still dominate the overall information, leading to a misalignment between information gain and true exploration needs. To address this issue, we propose a confidence-

weighted information matrix that incorporates confidence of each Gaussian, prioritizing under-observed regions.

$$c_g = \max_k (p_{g,k})^{-\alpha_{obj}} \sigma_g^{-\alpha_{opa}} \quad (7)$$

$$C_g = \text{diag}(c_g, \dots, c_g) \in R^{l \times l} \quad (8)$$

We define confidence without introducing any additional components, instead leveraging the previously discussed insights on opacity and the object vector. For example, as illustrated in Fig. 4, the well-observed part of the surface typically exhibits both high opacity and a maximum object probability close to 1. On the other hand, Gaussians that are poorly fitted or occluded tend to exhibit low opacity and object probability. To encourage more exploration of low-confidence Gaussians and reduce focus on those already well observed, we scale each Jacobian by a factor inversely proportional to both the opacity and the object probability during information gain computation:

$$H := J^T C J, \quad C = \text{diag}(C_g). \quad (9)$$

3) *Information Gain from Multiple Outputs*: A single 3DGS map can render RGB images, depth maps, and object masks from the same Gaussians but with different parameter subsets. All outputs use position, scale, rotation, and opacity, while RGB additionally uses SH coefficients, and masks use the object vector. Although information gain can be aggregated across outputs, scale differences across parameters and outputs make it difficult to balance their contributions.

$$\tilde{I}G_f(T, c) = \frac{IG_f(T, c)}{\frac{1}{|T|} \sum_{t \in T} IG_f(T, t)} \quad f \in \{rgb, d, o\} \quad (10)$$

To compensate, we normalize each output’s information gain by the mean information gain over the training views, which calibrates modality scales so that a typical training view has unit gain. This per-output normalization mitigates imbalance and allows RGB, depth, and mask contributions to be integrated more fairly and effectively in NBV:

$$\tilde{I}G(T, c) = \tilde{I}G_{rgb}(T, c) + \beta_d \tilde{I}G_d(T, c) + \beta_o \tilde{I}G_o(T, c) \quad (11)$$

C. Object-centric Next Best View for Manipulation

When a robot is tasked with performing a specific operation, it is often unnecessary to reconstruct the entire scene. If a particular object or region is of interest, prioritizing the reconstruction of that area is more practical and efficient. To enable this targeted reconstruction, we apply a simple modification to the previously defined confidence measure. Instead of using the maximum value of the object vector for each Gaussian, we directly use the value corresponding to the target object’s for confidence:

$$c_{g,obj} = \begin{cases} p_{g,obj}^{-\alpha_{obj}} \sigma_g^{-\alpha_{opa}} & \text{if } \text{argmax}_k(p_{g,k}) = obj \\ 0 & \text{else} \end{cases}, \quad (12)$$

and this allows the system to focus on the Gaussians most relevant to the specified object.

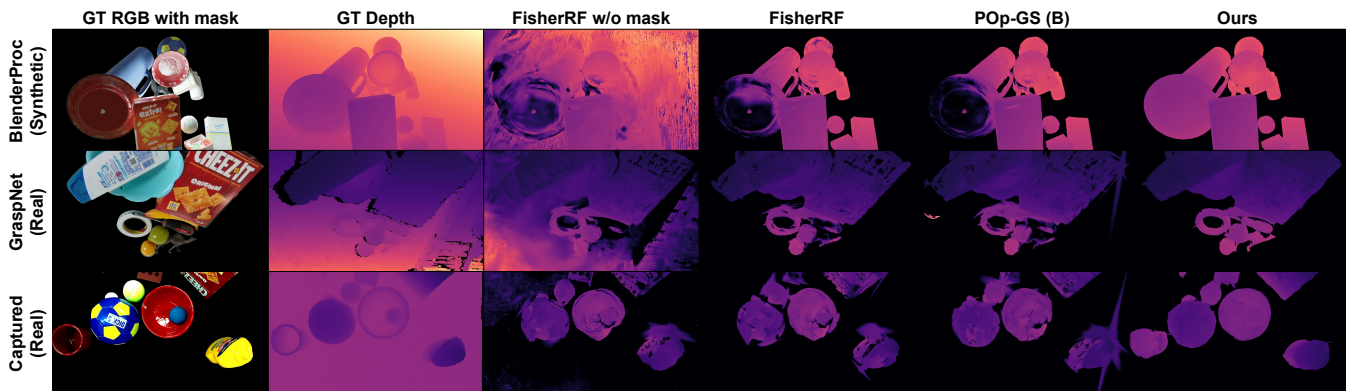


Fig. 5: Qualitative results of NBV system and 3D reconstruction for whole scene. Looking at the dish or the snack box, we can see that in the baseline, since training was performed only with RGB without instance information, textures and text appear in the depth results. In contrast, with our method such artifacts are absent, and the views selected through our NBV capture scene information from all directions, enabling more accurate overall depth reconstruction.

TABLE I: Results of NBV evaluation for whole scene reconstruction in synthetic and GraspNet dataset. Best results are highlighted in **bold**, and second-best results are underlined. The “All” row, which uses all candidate views, is excluded from this comparison.

Methods	BlenderProc Synthetic Dataset (10 Scenes)								GraspNet Real Dataset (10 Scenes)							
	W/o Obj mask				With Obj mask				W/o Obj mask				With Obj mask			
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	D-MAE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	D-MAE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	D-MAE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	D-MAE \downarrow
All	23.87	0.7756	0.3249	0.0390	29.95	0.9646	0.0601	0.0552	22.49	0.8035	0.3885	0.0645	23.15	0.8759	0.1590	0.0664
Spiral	17.75	0.5726	0.4100	<u>0.0630</u>	25.69	0.8836	0.1047	0.0694	19.85	0.7326	0.4164	0.0749	21.69	<u>0.8414</u>	0.1605	<u>0.0739</u>
Random	17.95	0.5568	0.4347	0.0701	24.91	0.9079	0.0978	0.0716	18.55	0.6990	0.4427	0.0801	20.44	0.7997	0.1830	0.0786
FPS	18.42	0.5808	0.4271	0.0670	24.47	0.9026	0.1020	0.0733	19.22	0.7098	0.4404	0.0822	21.25	0.8346	0.1706	0.0786
FisherRF [9]	18.64	0.5753	0.4335	0.0703	25.34	0.9253	0.0873	0.0695	18.82	0.7046	0.4358	0.0805	21.22	0.8328	0.1715	0.0768
POp-T [10]	19.07	0.6046	0.4093	0.0648	26.54	0.9366	0.0783	0.0651	19.05	0.7047	0.4435	0.0801	21.10	0.8310	0.1733	0.0768
POp-D [10]	19.04	0.5973	0.4139	0.0675	26.64	0.9387	<u>0.0770</u>	0.0641	18.99	0.7046	0.4427	0.0801	21.16	0.8313	0.1731	0.0770
POp-B [10]	19.04	0.5973	0.4139	0.0675	<u>26.69</u>	<u>0.9393</u>	0.0766	0.0636	18.99	0.7043	0.4413	0.0803	21.12	0.8315	0.1731	0.0768
Ours	-	-	-	-	26.90	0.9408	0.0773	0.0144	-	-	-	-	<u>21.47</u>	0.8430	<u>0.1653</u>	0.0487

TABLE II: Results of NBV evaluation for whole scene reconstruction in captured scenes. Best results are highlighted in **bold**, and second-best results are underlined. The “All” row, which uses all candidate views, is excluded from this comparison.

Methods	Captured Scene - Grounded SAM2 Mask (2 scenes)								Captured Scene - Manual SAM Mask (2 scenes)							
	W/o Obj mask				With Obj mask				W/o Obj mask				With Obj mask			
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	D-MAE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	D-MAE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	D-MAE \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	D-MAE \downarrow
All	17.04	0.6439	0.5401	0.0985	17.23	0.8333	0.2053	0.1169	17.85	0.7049	0.5145	0.0561	18.27	0.7839	0.2936	0.0494
Spiral	15.51	0.5706	0.5566	<u>0.1066</u>	<u>16.66</u>	<u>0.8072</u>	<u>0.2152</u>	0.1128	14.56	0.5424	0.5714	0.0845	15.49	0.6554	0.3511	0.0792
Random	14.42	0.5318	0.5759	0.1178	16.22	0.7668	0.2315	0.1228	15.11	0.5807	0.5578	0.0841	15.57	0.6699	0.3397	0.0779
FPS	14.67	0.5404	0.5736	0.1239	16.37	0.7787	0.2271	0.1336	14.23	0.5471	0.5775	0.0811	15.26	0.6554	0.3543	<u>0.0637</u>
FisherRF [9]	14.28	0.5127	0.5852	0.1283	15.60	0.7677	0.2405	0.1135	14.91	0.5739	0.5646	0.0857	15.34	0.6697	0.3472	0.0761
POp-T [10]	14.70	0.5255	0.5728	0.1265	16.16	0.7883	0.2271	0.1291	14.96	0.5630	0.5638	0.0803	15.44	0.6837	0.3412	0.0766
POp-D [10]	14.93	0.5319	0.5678	0.1224	16.28	0.7894	0.2264	0.1188	15.18	0.5783	0.5596	0.0864	<u>15.65</u>	<u>0.6927</u>	<u>0.3350</u>	0.0773
POp-B [10]	14.50	0.5251	0.5761	0.1264	16.29	0.7902	0.2263	0.1214	15.20	0.5808	0.5618	0.0823	15.58	0.6895	0.3385	0.0747
Ours	-	-	-	-	16.74	0.8148	0.2064	0.0527	-	-	-	-	16.71	0.7280	0.3102	0.0461

After performing reconstruction using the views selected through NBV, we obtain depth images from all views for actual manipulation. Since the selected views are expected to contain high-information content, it is more effective to extract candidate grasping points from the depth maps of these views and select the most confident one among them.

IV. EXPERIMENT

In this section, we evaluate the effectiveness of our view selection for both reconstruction and manipulation. First, we assess how closely the reconstructed scene obtained from the selected views matches the ground-truth geometry. Next, we evaluate the quality of reconstruction for a specific object of interest to determine whether the selected views effectively

focus on that target. Finally, we apply our method to real-world robotic experiments.

A. Next Best View for Whole Scene Reconstruction

1) *Datasets and Metrics:* We evaluate our method on both real and synthetic datasets. For the real-world data, we use 10 scenes from the GraspNet [1] training set and 4 robot-captured scenes of BOP objects. GraspNet provides ground-truth object masks, while the captured scenes we use pseudo ground-truth masks generated with manual SAM or Grounded SAM2 [15] with text prompts. For the synthetic data, we construct 10 scenes using BlenderProc [30] with BOP objects introducing heavy occlusions, leaning, and stacking among objects. Candidate views are generated on a sphere centered at the scene centroid, by combining ran-

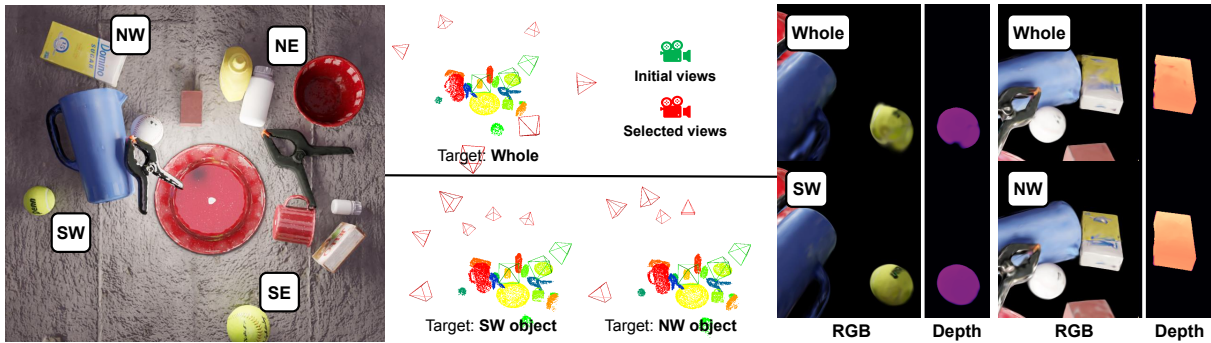


Fig. 6: **Left**: Cluttered scene with objects far from the center, labeled using cardinal directions (NW, NE, SW, SE). **Center**: Views selected by NBV (red camera frustum) targeting the entire scene versus object-centric NBV focusing on the SW and NW objects. While targeting the entire scene yields a near-uniform distribution of viewpoints, specifying a particular object (the SW or NW object) results in viewpoints clustering on the western side, from which the target object is best observed. **Right**: Reconstructions trained with these selected views yield improved RGB and depth renderings compared with whole scene NBV.

domly sampled viewpoints with spiral-arranged viewpoints that are evenly spaced in longitude at the same latitude.

The primary evaluation metrics for reconstruction are PSNR, SSIM, and LPIPS, computed from the rendered RGB images. To further assess geometric accuracy, we also compare the reconstructed depth MAE. Since our method prunes Gaussians corresponding to background regions, depth comparisons are performed only on pixels that fall within the objects. All results are averaged over three runs.

2) *Baseline and Implementation Details*: For heuristic baselines, we use random sampling, evenly spaced spiral trajectories, and farthest point sampling. For information-based NBV, we evaluate FisherRF [9] and the T-/D-optimality criteria from POP-GS [10], including a block-diagonal variant (B) that accounts for parameter correlations. (POP-T/D/B in table) Each baseline is tested with and without segmentation masks, which are used only for background removal without incorporating any additional features.

Reconstruction follows the original 3DGS pipeline for initialization, rasterization, and densification, except for the object vector supervision, which is unique to our method. We set $\lambda_{Dice} = 0.5$, $\lambda_{obj} = 0.1$, $\delta_{obj} = 0.1$ for the reconstruction loss. For NBV, confidence weights are $\alpha_{obj} = 0.3$ and $\alpha_{opa} = 0.3$. These values are selected through grid search and fixed for all scenes in experiments. Information gain weights are $\beta_d = 10$ and $\beta_o = 1$ following [11].

3) *Evaluation and Analysis*: Table. I and Table. II show the results of reconstructing the entire scene by adding 6 views to an initial 4 views. In both the synthetic and real datasets, our method outperforms other information-based approaches. While methods mostly benefit from using object masks to remove the background, our approach achieves a particularly significant reduction in depth error thanks to the incorporation and optimization of object features, which effectively encodes information from object mask. Moreover, even for the captured scenes in Table. II, where pseudo ground-truth object masks are used instead of actual ground-truth, our method continues to demonstrate strong performance.

The ‘‘All’’ row, which uses all candidate views for training, can be regarded as the upper bound for reconstruction within

TABLE III: Ablation study for design choices.

	BlenderProc Dataset		GraspNet Dataset		Captured Scenes	
	PSNR \uparrow	D-MAE \downarrow	PSNR \uparrow	D-MAE \downarrow	PSNR \uparrow	D-MAE \downarrow
FisherRF with mask	25.34	0.0695	21.23	0.0768	15.25	0.0913
+ Object vector	25.60	0.0189	21.44	0.0486	16.05	0.0583
+ Confidence score	26.84	0.0143	21.39	0.0488	16.29	0.0521
+ IG scaling (Ours)	26.92	0.0144	21.37	0.0491	16.34	0.0507
NBS (Depth) [11]	26.00	0.0675	21.21	0.0765	15.62	0.0936
NBS (RGB + D) [11]	25.83	0.0675	21.16	0.0763	15.17	0.0896

the scene. The spiral selection strategy intuitively produces a set of views that provides a well-balanced coverage of the scene and indeed outperforms other heuristic selection methods and information-based baselines in terms of reconstruction quality. However, because the spiral pattern does not account for scene-specific characteristics such as objects located deep inside, so it fails to adapt to individual scene layouts. This limitation becomes more pronounced in the synthetic dataset, which is designed to be more dynamic, compared to the simpler GraspNet dataset where training and test view distributions are more similar.

Fig. 5 provides a qualitative comparison of the results. In the outputs from other methods, the text printed on the boxes appears with a different depth from the surface it is on, making the text distinguishable in the depth image. This occurs because optimization is performed solely on RGB rendering without considering object-level consistency, often leading to local minima in sparse-view settings. In contrast, our method benefits from instance-aware supervision, preventing 3D Gaussians from falling into such local minima and ensuring that they are properly aligned with the surfaces of the boxes. Regarding next best view selection, our approach captures not only the top surfaces and larger objects but also occluded regions and smaller objects, demonstrating a balanced exploration capability and better reconstruction.

4) *Ablation Study*: Table. III provides an in-depth analysis of each design choice within the proposed method. Incorporating one-hot object vector yields more accurate reconstruction without requiring additional views. Compared to FisherRF results obtained with background-removed images, the improvement is particularly pronounced in depth accuracy which decreases significantly, highlighting the necessity of semantic information for reliable robotic manipulation.

Furthermore, we observe that scaling with confidence scores derived from the one-hot object vector and opacity, combined with the information gain from training views, is also effective. While Next Best Sense [11] (NBS in table) heuristically balances RGB and depth contributions by defining $IG(T, c) = IG_{rgb}(T, c) + 10IG_d(T, c)$, our approach adaptively determines these weights based on the training views, leading to improved robustness.

TABLE IV: Object-centric reconstruction results

Scene	Target	Depth MAE ↓					
		Corner objs				Center objs	Total objs
		NW	NE	SW	SE		
Syn.	Whole	0.0180	0.0365	0.0130	0.0215	0.0162	0.0193
	Corner	0.0132	0.0124	0.0114	0.0166	0.0149	0.0205
	Center	0.0182	0.0503	0.0151	0.0286	0.0148	0.0225
Real	Whole	0.0621	0.2632	0.0695	0.0777	0.0334	0.0517
	Corner	0.0490	0.1963	0.0624	0.0619	0.0313	0.0540
	Center	0.1000	0.1816	0.0683	0.0647	0.0304	0.0535

B. Next Best View for Object-centric Reconstruction

Our method has the capability to select views that are specifically tailored to a designated target object. To clearly validate this effect, we select four corner objects in each scene that are spatially distant from one another as shown in Fig. 6. This choice is intentional, as performing object-centric reconstruction on objects nearby center would likely result in overlapping visibility across most candidate views.

For comparison, we consider three setups: (1) using the entire scene as the reconstruction target, (2) targeting non-selected center objects, and (3) individually targeting each of the four distant corner objects. This comparison highlights how our object-wise NBV strategy adapts to different reconstruction goals and spatial distributions.

As shown in Table. IV and Fig. 6, when NBV planning is targeted for a specific object rather than the entire scene, the selected viewpoints are concentrated on the corresponding object, leading to improved depth accuracy for the object.

In manipulation tasks, where the target object and its surrounding regions are substantially more critical than distant areas, this strategy not only prevents unnecessary time spent on scanning irrelevant regions but also provides a decisive advantage in both efficiency and accuracy. Consequently, it represents a highly promising direction for achieving robust and task-oriented robotic manipulation.

C. Analysis on Convergence Speed of Reconstruction

Fig. 7 illustrates how the error with respect to the ground-truth depth decreases as the number of selected views increases, comparing our method with the baselines. Only the final number of selected views is varied, ranging from 5 to 14, and the corresponding depth MAE is measured.

As shown in the graph, our method yields better initial results and ultimately achieves higher PSNR and lower depth MAE than all baselines as views increase. By leveraging an appropriate termination criterion or empirical tuning, the

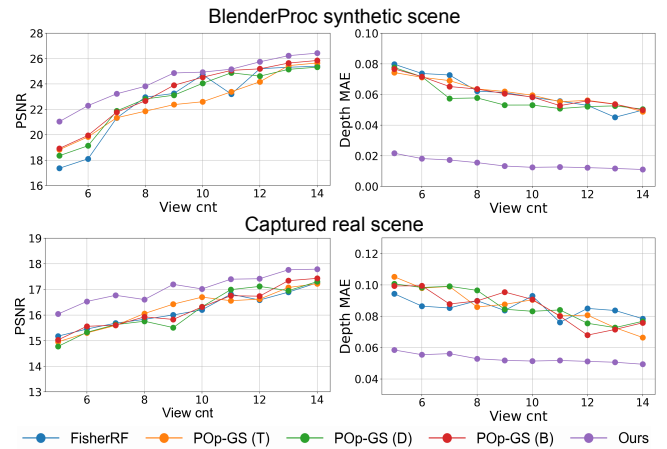


Fig. 7: PSNR and Depth MAE as the number of views increases from 5 to 15 in both synthetic and real scenes.

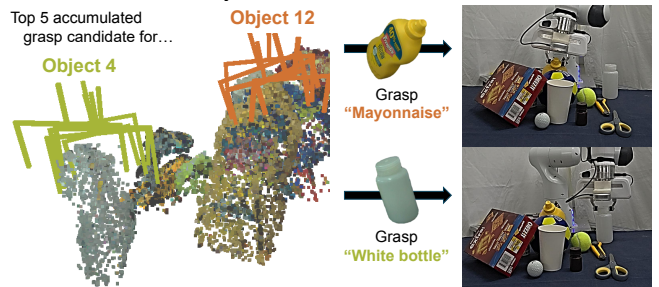


Fig. 8: Left shows the top-5 aggregated grasp candidates from multi-view EconomicGrasp predictions, grouped by object. Target object can be selected using CLIP embedding similarity between a text prompt and object masks. Right shows execution snapshots of successful grasps on two target objects.

scanning process can be terminated earlier while still maintaining reconstruction quality, thereby significantly reducing the time required for robotic operation.

D. Extension to Real-world Manipulation

We apply our method to real-world robotic manipulation. To obtain grasping point, we employ EconomicGrasp [2], and the robot platform consists of a Franka Emika Panda arm equipped with an Intel RealSense L515 camera.

As shown in Fig. 8, our NBV system successfully selects informative viewpoints and also returns reliable grasping points from multiple views. This approach is more robust than simply obtaining the grasping point from a top-down view. Furthermore, when a language prompt specifying a target object is given, we use CLIP to compute the cosine similarity between the prompt embedding and the embeddings of the object masks. The object with the highest similarity is selected, and the NBV planning is performed with a focus on that object. This demonstrates that our object-wise reconstruction framework is well suited for integration with open-vocabulary vision-language models.

In real-world scenarios, such as bin picking or retrieving objects from the containers, the reasonable viewpoints are often physically constrained. We recommend watching the supplementary video showcasing examples where our method is applied to such restricted environments.

V. CONCLUSION

We presented an instance-aware 3DGS-based, confidence-aware NBV policy that injects object masks into the 3DGS pipeline, prioritizes underexplored regions, and extends naturally to object-centric NBV. By balancing the contributions of RGB, depth, and mask rendering, the selected views form a compact, informative set, which also yields clearer and more reliable grasping points in cluttered scenes.

A key limitation is the reliance on object masks. Although we observed good robustness with pseudo ground-truth on challenging real scenes, inaccuracies in the masks can propagate to both reconstruction and NBV decisions. Future work will focus on online refinement of object mask via 3DGS or 3D foundation models as new views are acquired. We also plan to generate candidate viewpoints actively and to adopt principled termination criteria to enable early stopping.

REFERENCES

- [1] H.-S. Fang, C. Wang, M. Gou, and C. Lu, “Graspnet-1billion: A large-scale benchmark for general object grasping,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2020, pp. 11 444–11 453.
- [2] X.-M. Wu, J.-F. Cai, J.-J. Jiang, D. Zheng, Y.-L. Wei, and W.-S. Zheng, “An economic framework for 6-dof grasp detection,” in *Proc. European Conf. on Comput. Vision*. Springer, 2024, pp. 357–375.
- [3] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, “6-dof grasping for target-driven object manipulation in clutter,” in *Proc. IEEE Intl. Conf. on Robot. and Automat.* IEEE, 2020, pp. 6232–6238.
- [4] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic *et al.*, “Deep learning approaches to grasp synthesis: A review,” *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3994–4015, 2023.
- [5] X. Zhang, D. Wang, S. Han, W. Li, B. Zhao, Z. Wang, X. Duan, C. Fang, X. Li, and J. He, “Affordance-driven next-best-view planning for robotic grasping,” in *Conference on Robot Learning*. PMLR, 2023, pp. 2849–2862.
- [6] L. Jin, X. Zhong, Y. Pan, J. Behley, C. Stachniss, and M. Popović, “Activegs: Active scene reconstruction using gaussian splatting,” *IEEE Robot. and Automat. Lett.*, 2025.
- [7] L. Goli, C. Reading, S. Sellán, A. Jacobson, and A. Tagliasacchi, “Bayes’ rays: Uncertainty quantification for neural radiance fields,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2024, pp. 20 061–20 070.
- [8] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [9] W. Jiang, B. Lei, and K. Daniilidis, “Fisherrf: Active view selection and mapping with radiance fields using fisher information,” in *Proc. European Conf. on Comput. Vision*. Springer, 2024, pp. 422–440.
- [10] J. Wilson, M. Almeida, S. Mahajan, M. Labrie, M. Ghaffari, O. Ghasemalizadeh, M. Sun, C.-H. Kuo, and A. Sen, “Popgs: Next best view in 3d-gaussian splatting with p-optimality,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2025, pp. 3646–3655.
- [11] M. Strong, B. Lei, A. Swann, W. Jiang, K. Daniilidis, and M. Kennedy III, “Next best sense: Guiding vision and touch with fisherrf for 3d gaussian splatting,” in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2025.
- [12] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.
- [13] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *et al.*, “Grounding dino: Marrying dino with grounded pre-training for open-set object detection,” in *Proc. European Conf. on Comput. Vision*. Springer, 2024, pp. 38–55.
- [14] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2023, pp. 4015–4026.
- [15] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan *et al.*, “Grounded sam: Assembling open-world models for diverse visual tasks,” *arXiv preprint*, 2024.
- [16] Z. Feng, H. Zhan, Z. Chen, Q. Yan, X. Xu, C. Cai, B. Li, Q. Zhu, and Y. Xu, “Naruto: Neural active reconstruction from uncertain target observations,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2024, pp. 21 572–21 583.
- [17] L. Chen, H. Zhan, K. Chen, X. Xu, Q. Yan, C. Cai, and Y. Xu, “Activegamer: Active gaussian mapping through efficient rendering,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2025, pp. 16 486–16 497.
- [18] J. Kiefer, “General equivalence theory for optimum designs (approximate theory),” *The Annals of Statistics*, pp. 849–879, 1974.
- [19] S. Kobayashi, E. Matsumoto, and V. Sitzmann, “Decomposing nerf for editing via feature field distillation,” *Advances in Neural Information Processing Sys. Conf.*, vol. 35, pp. 23 311–23 330, 2022.
- [20] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik, “Lerf: Language embedded radiance fields,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2023, pp. 19 729–19 739.
- [21] R.-Z. Qiu, G. Yang, W. Zeng, and X. Wang, “Language-driven physics-based scene synthesis and editing via feature splatting,” in *Proc. European Conf. on Comput. Vision*, 2024.
- [22] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, “Langsplat: 3d language gaussian splatting,” in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2024, pp. 20 051–20 060.
- [23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *Proc. Intl. Conf. on Machine Learning*. PMLR, 2021, pp. 8748–8763.
- [24] M. Ji, R.-Z. Qiu, X. Zou, and X. Wang, “Graspsplats: Efficient manipulation with 3d feature splatting,” in *Conference on Robot Learning*. PMLR, 2025, pp. 1443–1460.
- [25] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong, “Faster segment anything: Towards lightweight sam for mobile applications,” *arXiv preprint*, 2023.
- [26] M. Ye, M. Danelljan, F. Yu, and L. Ke, “Gaussian grouping: Segment and edit anything in 3d scenes,” in *Proc. European Conf. on Comput. Vision*. Springer, 2024, pp. 162–179.
- [27] R. Zhu, M. Yu, L. Xu, L. Jiang, Y. Li, T. Zhang, J. Pang, and B. Dai, “Objectgs: Object-aware scene reconstruction and scene understanding via gaussian splatting,” in *Proc. IEEE Intl. Conf. on Comput. Vision*, 2025.
- [28] J. Kim, S. Jeong, G. Kim, M.-H. Jeon, E. Jun, and A. Kim, “2d gaussian splatting-based sparse-view transparent object depth reconstruction via physics simulation for scene update,” in *Proc. IEEE Intl. Conf. on Comput. Vision*, 2025.
- [29] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, “Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations,” in *International Workshop on Deep Learning in Medical Image Analysis*. Springer, 2017, pp. 240–248.
- [30] M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam, “Blender-proc,” *arXiv preprint*, 2019.