Active Perception for Grasp Detection via Neural Graspness Field

Haoxiang Ma[1] **Modi Shi**[1] **Boyang Gao**[2] **Di Huang**[1]*
[1]State Key Laboratory of Complex and Critical Software Environment,
School of Computer Science and Engineering, Beihang University, Beijing, China
[2]Geometry Robotics
{mahaoxiang822,modishi,dhuang}@buaa.edu.cn, boyang.gao@geometryrobot.com

Abstract

This paper tackles the challenge of active perception for robotic grasp detection in cluttered environments. Incomplete 3D geometry information can negatively affect the performance of learning-based grasp detection methods, and scanning the scene from multiple views introduces significant time costs. To achieve reliable grasping performance with efficient camera movement, we propose an active grasp detection framework based on the Neural Graspness Field (NGF), which models the scene incrementally and facilitates next-best-view planning. Constructed in real-time as the camera moves, the NGF effectively models the grasp distribution in 3D space by rendering graspness predictions from each view. For next-best-view planning, we aim to reduce the uncertainty of the NGF through a graspness inconsistency-guided policy, selecting views based on discrepancies between NGF outputs and a pre-trained graspness network. Additionally, we present a neural graspness sampling method that decodes graspness values from the NGF to improve grasp pose detection results. Extensive experiments on the GraspNet-1Billion benchmark demonstrate significant performance improvements compared to previous works. Real-world experiments show that our method achieves a superior trade-off between grasping performance and time costs. Code is available at `https://github.com/mahaoxiang822/ActiveNGF`.

1 Introduction

Learning-based robotic grasp synthesis [24] has been explored to enable the manipulation of various objects across different grippers, sensors, and scenarios. The completeness and accuracy of the scene representation significantly influence the performance of these methods, as the geometric ambiguity can confuse the synthesis of grasp poses. To address this issue, previous works [4, 8, 20] have employed multi-view information to reconstruct different 3D representations. However, as the robot needs to move when observing from different views, scanning the entire scene to achieve complete coverage incurs a substantial time cost and is challenging to apply in real-world environments.

To enhance the efficiency of multi-view perception for robotic grasping, previous works have introduced active perception methods to select the Next-Best-View (NBV). Part of the methods [13, 7, 5] apply active 3D reconstruction techniques, treating grasp detection as a secondary task. However, active reconstruction tends to select viewpoints that cover more unobserved space, which may not be optimal for grasp detection, as different regions have varying relevance to grasping. Consequently, active perception strategies based on 3D reconstruction have limited performance in grasp detection tasks. Recently, ACE-NBV [32] combines grasp detection and NBV planning by specially designing

*Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

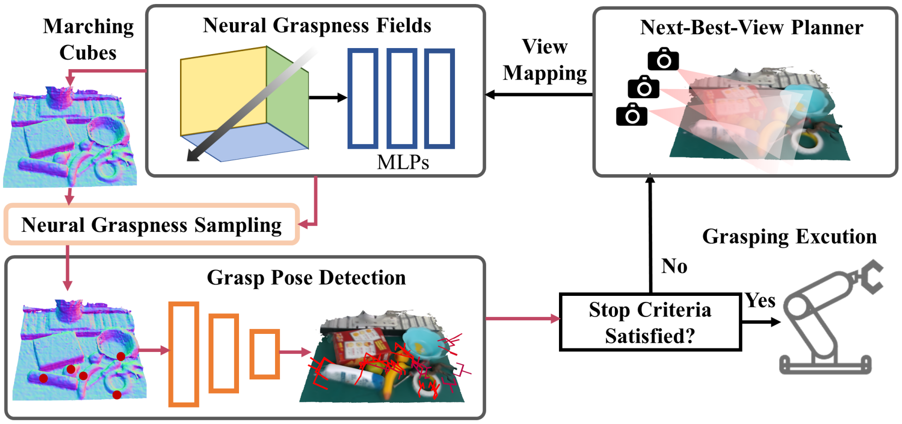


Figure 1: Overview of the active grasp detection system. The RGB, depth and predicted graspness from a new view are mapped to the Neural Graspness Field (NGF) by rendering loss. After each mapping step, the scene geometry is exported from the neural representation using the marching-cubes algorithm [19] and the candidate positions for grasp synthesis are sampled by neural graspness sampling. If the maximum perception step is reached or a specific result condition is satisfied, *e.g.*, a sufficient number of high-quality grasps are detected, the robot arm is employed to execute the detected grasps. Otherwise, the Next-Best-View (NBV) planner is employed to sample candidate views and select the view with the largest information gain for robot movement.

a grasp detection network to predict the grasp affordance of candidate views. The viewpoint with the highest predicted grasp quality is then selected for reconstruction to efficiently obtain feasible grasps. However, this approach has two main limitations. First, it suppresses the perception of views that have low grasp quality but provide grasp-related geometric information, which can lead the method to fall into local optima. Second, directly predicting information gain through a specially designed grasp detection network can be limited by the network's generalization ability, potentially resulting in performance degradation when applied to novel environments.

To address the aforementioned issues, inspired by the development of the neural representation and radiance field [21, 14], we propose to employ a Neural Graspness Field (NGF) to model the scene grasp distribution and build the active grasp detection system based on it. The graspness of a position measures the sum of feasible grasps in its pose space. The NGF is optimized online during the camera movement by back-propagating the rendering loss of the view graspness. The online-training scheme employed for NGF modeling makes it easier to transfer to unseen scenarios. Moreover, distilling multi-view information into 3D space using the neural representation enforces the multi-view consistency, making the NGF less susceptible to the depth noise and view sparsity compared to directly predicting the grasp distribution on a reconstructed 3D representation [33]. With the neural representation of the scene grasp distribution, the active perception problem can be defined as minimizing the error of the scene grasp distribution modeled by NGF. This is achieved by strategically selecting views that can bring the largest information gain for the NGF after mapping, thereby incrementally refining the modeled grasp distribution towards an optimal state.

In this paper, we propose an active perception method for the robotic grasp detection consisting of two components: neural graspness field mapping and graspness inconsistency-guided next-best-view planning. For neural graspness field mapping, we extend a NeRF-based real-time mapping system [12] to render view graspness by adding a separate network branch. The view graspness is predicted by a pre-trained graspness network from the corresponding depth image. For NBV planning, we provide a graspness inconsistency-guided strategy targeting minimizing the inconsistency between the current NGF and the ground-truth scene grasp distribution. For a given view, its information gain is described as the inconsistency between the view graspness rendered from the NGF and the pseudo label predicted from the rendered depth image. Furthermore, we propose an inference strategy based on our active grasp detection framework, which decodes the graspness score from the NGF to generate grasp samples instead of predicting from explicit 3D geometries. The contributions of this paper can be summarized as follows:

- We propose an active grasp detection framework via neural graspness field to model the scene grasp distribution online during camera movement.

- We adopt a graspness inconsistency-guided strategy for next-best-view planning, which targets on reducing the uncertainty of the neural graspness field.
- A neural graspness sampling inference strategy is proposed to enhance the performance of the grasp detection framework.

2 Related Work

2.1 Grasp Detection

Grasp detection aims to generate feasible and diverse gripper poses for given objects. Early methods mainly studied the theoretical framework for robotic grasping by analyzing the contacts between the gripper and object models. Recently, to achieve grasping of unseen objects, data-driven grasp detection methods have been extensively studied. Some works [9, 34, 1] investigate grasp detection in planar space, simplifying the problem by directly locating rotated grasp rectangles on images. To generate more diverse grasps that support complex downstream manipulation tasks, 6-DoF grasp detection has been proposed to predict grasping in $SE(3)$ space. [30] samples gripper poses on point clouds using geometric priors and scores these samples using a Convolutional Neural Network (CNN). [18] follows the sample-and-score framework but introduces PointNet [27] to achieve better scoring performance. Some recent works adopt end-to-end networks for 6-DoF grasp detection to achieve real-time inference. [23] employs a variational autoencoder for grasp generation given the object point clouds, and [28] proposes a single-shot grasp proposal network to enhance the efficiency of grasp detection. [10] densely annotates grasp labels in clutter to construct a large benchmark and provides an end-to-end 6-DoF grasp detection baseline trained with the large amount of annotations. Following this, [31] defines graspness to represent the grasp distribution in a scene and proposes a graspness discovery method. Although the aforementioned methods have achieved good results, using single-view point clouds as input leads to a lack of geometric information due to occlusions and limited field of view, which affects the grasping performance on some objects. To solve this problem, [4, 20] utilize Truncated Signed Distance Functions (TSDFs) to map multi-view frames before generating grasps, and [8] employs a generalizable neural radiance field to reconstruct the scene from cameras set around the scene, which supports the grasp detection of transparent and specular objects. However, these methods rely on scanning the scene from all possible views or several views surrounding the scene, resulting in excessive robot execution time for moving the camera. The large time cost reduces the usability of grasp detection methods in real-world scenarios.

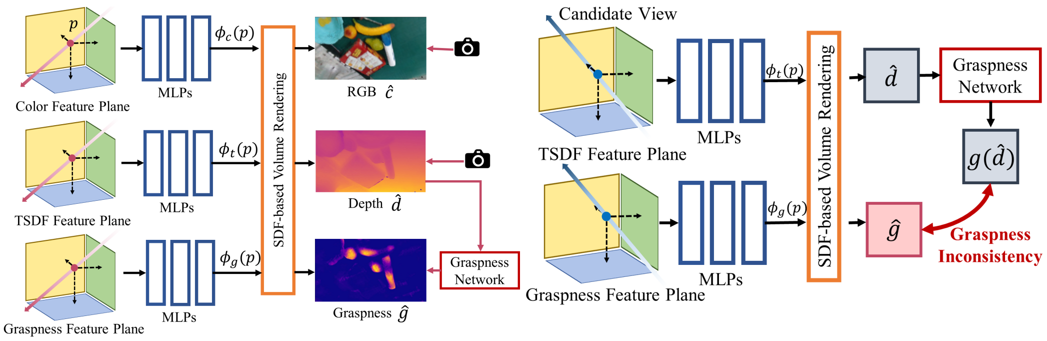
2.2 Active perception for Grasp Detection

Active perception [3] aims to develop an agent that knows why it wishes to sense, and then chooses what to perceive, and determines how, when and where to achieve that perception. Planning the next-best-view is a commonly used method to realize an active perception system. In terms of grasp detection, active perception is introduced to determine the camera views that can achieve a trade-off between the grasp performance and time cost. There are two main lines of work in this area: those that treat grasp detection as a secondary task of 3D reconstruction and those that directly incorporate grasp detection into the view planning process. Some works [13, 2, 7, 5] plan the view sequence based on the 3D reconstruction metric of the grasp-relevant region, where the grasp detection is treated as a secondary task. [13] models the uncertainty of occluded voxels as a mixture of Gaussians and utilizes trajectory optimization to generate the view sequence. [2] provides an active vision approach to maximize surface reconstruction quality near the contact point region, and [7] adopts reinforcement learning for view planning with an object mask-guided reward function. To achieve close-loop control, [5] incorporates a grasp detection network to continuously predict grasps after each view mapping. However, employing grasp detection as a secondary task of 3D reconstruction overlooks the internal relation between grasp synthesis and scene reconstruction, which can lead to sub-optimal results. Another line of works [11, 22, 32] directly incorporates grasp detection into the view planning process. [11] maps grasp detection performance as a function of viewpoint for each object but struggles with novel objects. [22] uses the entropy of the network prediction to determine the next-best-view in a top-down grasping setting and recently, [32] proposes an affordance-driven policy based on an implicit grasp detection network to generate grasp affordance for unseen views. However, these approaches that utilize the output of grasp detection networks for view planning rely on the specific design of the network and are easily influenced by the generalization ability of the grasp detection network, making it challenging to apply these methods to diverse real-world scenarios. In contrast to these approaches, our method utilizes view grasp distribution, which is independent of specific network designs, and constructs the NGF online. This enables our method to easily adapt to diverse real-world scenarios at test time.

3 Method

3.1 System Overview

The objective of grasp detection in cluttered scenes is to generate diverse feasible grasps for each object in the scene. Given a robotic arm with a mounted depth camera, active grasp detection aims to find a camera movement policy that can achieve high-quality results within a maximum of T time steps. An overview of our active grasp detection system is provided in Fig. 1.



(a) Neural graspness Field Mapping

(b) Graspness Inconsistency-guided NBV Planning

Figure 2: The pipeline of the proposed mapping and NBV planning methods.

3.2 Neural Graspness Field Mapping

Inspired by neural feature fields [15] and semantic NeRF [33], we incorporate a graspness field to represent the scene grasp distribution. The NGF employs a separate branch besides the appearance and geometry to render grasp distribution information from multiple views. The graspness score proposed in [31] measures the graspable landscape in cluttered scenes given a position p. With L grasp candidates $G_p = \{g_k^p | k = 1, ..., L\}$ sampled in its configuration space, *i.e.*, approach direction, gripper depth and in-plane rotation, the ground-truth graspness score $\widetilde{g}_p$ is defined as:

$$\widetilde{g}_p = \frac{\sum_{k=1}^{L} \mathbf{1}(q_k > t) \cdot \mathbf{1}(c_k)}{|G_p|} \tag{1}$$

where q_k is the grasp quality score computed from force closure analysis, c_k indicates the collision state of the gripper in clutter and $\mathbf{1}$ is the indicator function. For an observed view, the NGF aims to map the view appearance, depth and graspness, as shown in Fig. 2 (a). The NGF is composed of two parts: axis-aligned feature planes and SDF-based volume rendering. The axis-aligned feature planes store learned features at different resolutions, which are queried and interpolated based on the 3D positions of the sampled points. For points p sampled in a ray, the corresponding features are queried from the feature planes and decoded by MLPs to get raw color $\phi_c(p)$, raw Truncated Signed Distance Field (TSDF) $\phi_t(p)$ and raw graspness $\phi_g(p)$. To convert raw TSDF values to volume densities, the SDF-based volume rendering from StyleSDF [25]:

$$\sigma(p) = \beta \cdot Sigmoid(-\beta \cdot \phi_t(p)) \tag{2}$$

where β is a learnable parameter. With the volume density, the color, depth and graspness of each ray r can be computed as:

$$\hat{c}(r) = -\sum_{n=1}^{N} w_n \phi_c(p_n) \quad \text{and} \quad \hat{d}(r) = -\sum_{n=1}^{N} \cdot z_n \quad \text{and} \quad \hat{g}(r) = -\sum_{n=1}^{N} w_n \phi_g(p_n) \tag{3}$$

where z_n is the ray depth of p_n and weight w_n is formulated as:

$$w_n = exp(-\sum_{n-1}^{k=1} \sigma(p_k))(1 - exp(-\sigma(p_n))) \tag{4}$$

For a given view, it is challenging to directly obtain the corresponding ground-truth graspness during online mapping. Therefore, we use a pre-trained graspness network that takes the depth image as input and predicts the view graspness to render the NGF. By rendering from different views, the graspness field can reduce the graspness noise caused by single-view depth and render the graspness distribution from any given view, which can be used to measure the grasp-correlated information gain.

3.3 Graspness Inconsistency-guided Next-Best-View Planning

Given a sequence of viewpoints $s = (v_1, v_2, ..., v_N)$ and sequence set S, the NBV planning problem can be defined as:

$$s^* = \underset{s \in S}{argmax} \sum_{n=1}^{N} I(v_n) \tag{5}$$

where N is the maximum step of robot movement and $I(v_n)$ represents the information gain of viewpoint v_n. The selection of an informative view is influenced by the scene representation and the definition of information gain. We consider the NGF obtained from observing the entire scene as the ground-truth scene grasp distribution and define the information gain of a view as the improvement in the NGF prediction by mapping this view. Given an unseen candidate view, the NGF can estimate the view graspness based on the existing observations, and the improvement is expressed as the inconsistency between the ground-truth graspness $g(d)$ predicted from the real depth and the graspness $\hat{g}$ rendered by the current graspness field.

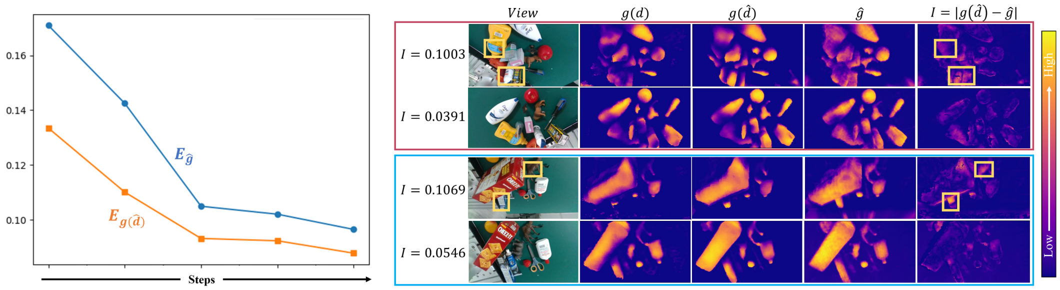
However, for the NBV problem, obtaining the real depth image of a candidate view is not possible. To address this, our inconsistency-guided NBV policy adopts the pseudo-label paradigm by substituting the ground-truth grasp distribution $g(d)$ with the pseudo-graspness $g(\hat{d})$, which is widely employed in other semi-supervised and active learning vision tasks [17, 29]. As shown in Fig. 2 (b), we leverage the NGF's ability to render a pseudo-depth image $\hat{d}$ of the candidate view by volume rendering the TSDF values and then pass this pseudo-depth image through the graspness prediction network to obtain the pseudo-graspness $g(\hat{d})$. The pseudo-graspness is used to calculate the information gain of the candidate view, which is defined as:

$$I(v) = |\sum_{r \in v} \hat{g}(r) - g(\hat{d})| \tag{6}$$

where $g(r)$ represents the rendered graspness of sampled ray and the summation symbol represents the rendered graspness of the whole view.

It should be noted that the effectiveness of our NBV policy relies on the premise that the $g(\hat{d})$ predicted by the graspness network is closer to the ground-truth $g(d)$ compared to $\hat{g}$. The smaller error in $g(\hat{d})$ can be attributed to two reasons: First, the robot-mounted camera moves continuously in small steps, resulting in minimal differences in the observed geometric information between views, leading to insignificant errors in the rendered depth $\hat{d}$. Second, the graspness prediction network, trained on a large dataset of real point clouds, inherently provides robustness to depth noise. We visualize the rendered graspness error $E_{\hat{g}} = |g(d) - \hat{g}|$ and the pseudo-graspness error $E_{g(\hat{d})} = |g(d) - g(\hat{d})|$ in Fig. 3 (a), where $E_{\hat{g}}$ is significantly larger than $E_{g(\hat{d})}$ and the difference decreases with more steps.

The proposed pseudo-graspness information gain can incorporate the view grasp distribution prior into the planning process, which is encoded by the pre-trained graspness network. Thus the NBV system can select the view containing the most graspness information that has not been distilled from the pre-trained network to the NGF. We visualize the ground-truth graspness $g(d)$, pseudo-graspness $g(\hat{d})$, rendered graspness $\hat{g}$ and the information gain I in different views of the neural graspness field in Fig. 3 (b). For different views, the pseudo-graspness predicted from the rendered depth image can approximately represent the ground-truth $g(d)$ but the accuracy of the rendered graspness $\hat{g}$ varies, which introduces different information gains.



(a)

(b)

Figure 3: (a) The pseudo-graspness error $E_{g(\hat{d})}$ and rendered graspness error $E_{\hat{g}}$ of initial steps. (b) Visualization of the pseudo-graspness, rendered graspness and the corresponding information gain of different views.

3.4 Neural Graspness Sampling

For the grasp detection method, predicting the grasp distribution in clutter and sampling positions for grasp synthesis is an important part. Previous methods usually employ different encoders to predict grasp distribution from explicit 3D representations but the incomplete and noisy geometry information can lead to inaccurate grasp distribution. To achieve more precise grasp sampling, in addition to using the NGF for active perception, we propose an inference strategy based on sampling from the NGF. Given the positions p sampled from the reconstructed surface, the graspness of the position can be decoded from the NGF. During inference, we replace the graspness sampled from the neural representation with the graspness predicted by the grasp detection network and utilize Furthest Point Sampling (FPS) on the positions larger than a threshold T to get positions for grasp synthesis, which is formulated as:

$$Samples = FPS(p\left\{\phi_g(p) >= T\right\}) \tag{7}$$

where ϕ_g is the graspness branch of the NGF.

4 Experiments

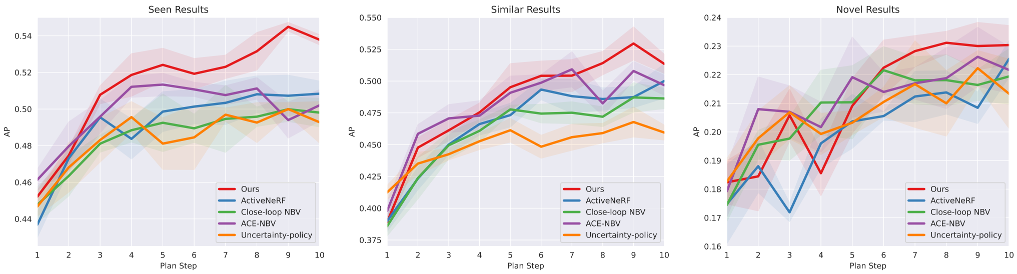
4.1 Experimental Setup

Simulation Setup We construct a simulation active grasp benchmark based on the GraspNet-1Billion benchmark [10], which consists of 100 scenes for training and 90 scenes for testing. The test set is divided into seen, similar, and novel sets based on the included objects. Each scene is captured from 256 views using Intel RealSense and Kinect cameras. We conduct all the experiments with the data captured by the Realsense camera. We set the pre-collected 256 views as the perception space for NBV planning. Since moving the camera is a continuous process, moving it over long distances would waste the information captured during the movement. Therefore, we sample the candidate views from the current view with a relatively small step size. In our experiments, we set the step size to 10cm and set maximum step to 10. For evaluation, we follow the metric used in the GraspNet-1Billion benchmark, which simulate grasps with friction μ ranging from 0.2 to 1.2 with interval $\delta\mu = 0.2$. Following [20], we sample 5 grasps for each object in the scene to calculate the average precision **AP**. The training and evaluation of the simulation experiments are conducted on a single NVIDIA V100 GPU.

Baselines We compared the following baseline methods to validate the effectiveness of our proposed method. The baselines can be divided into two categories: NBV for robotic grasping [5, 32] and NBV planning based on NeRF [26, 16]. Close-loop NBV [5] utilizes ray casting to calculate the number of unobserved voxels of objects, which drives exploration targeting occluded object parts. ACE-NBV [32] incorporates grasp affordance prediction into NBV planning and selects the view with the largest grasp affordance as the next-best-view. ActiveNeRF [26] proposes an plug-in uncertainty estimation method for NeRF based on Bayesian estimation and Uncertainty-policy [16] computes the entropy of the weight distribution of each ray as the uncertainty.

Implementation Details For the mapping of NGF, the first view is trained for 100 iterations and following views are trained for 50 iterations. For each ray, 32 points are sampled for stratified sampling and 8 points for importance sampling. Only the coarse planes in [12] are employed for mapping. For NBV planning, we downsample the original image to $1/8$ to sample rays for graspness rendering to speed up the computation of view information gain. We utilize the first-stage network of GSNet [31] which predicts the graspness score for point-clouds as the graspness network in this paper. For the grasp detection network used for inference, we adopt the baseline method from [20] which uses the reconstructed scene geometry as input. For each scene, 1024 points are sampled from the NGF for grasp pose synthesis.

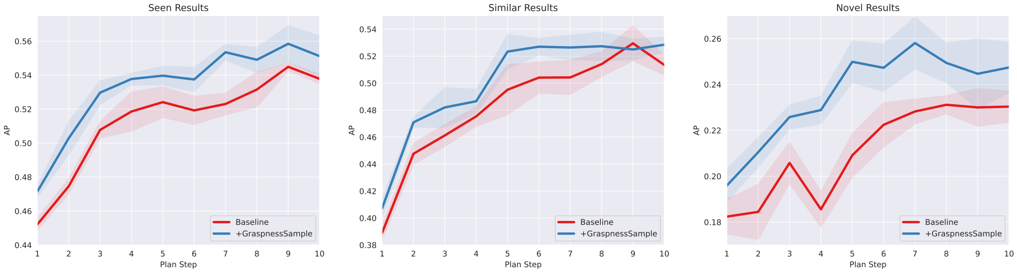
4.2 Simulation Experiments



(a) Seen (b) Similar (c) Novel

Figure 4: Comparison on different NBV policies based on the proposed NGF.

Comparison on different NBV policies To validate the effectiveness of the proposed NGF and the graspness inconsistency-guided NBV policy based on it, we re-implement other view planning policies on the same ESLAM mapping framework [12] which the NGF is built on. As shown in Fig. 4, our pseudo-graspness guided policy achieves superior performance after the first several views on seen, similar, and novel sets. Compared to ActiveNeRF [26] and Uncertainty-policy [16], our method selects views with more grasp distribution discrepancy instead of geometry or appearance ambiguity in the neural representation, which improves the results. Compared to policies targeting grasp detection [5, 32], our method is specially designed to reduce the uncertainty of the NGF by distilling the prior knowledge of a pre-trained network, thus achieving superior performance with more views. It should be noted that ACE-NBV [32] can achieve comparable results to ours in the initial steps while showing little improvement as more views are added. This is because the affordance-based policy only selects views with more feasible grasps but does not consider optimizing the scene grasp representation throughout the entire planning process.



(a) Seen (b) Similar (c) Novel

Figure 5: The comparison of grasp detection result generated with the graspness predicted from 3D geometry and sampled from NGF.

Effectiveness of Neural Graspness Sampling We apply the neural graspness sampling during the inference of the grasp detection network to validate its effectiveness. The results are shown in Fig. 5. The sampling strategy improves the grasp detection results on seen, similar and novel objects at each step. Constructing a NGF through online multi-view rendering, compared to directly predicting the grasp distribution from 3D geometric information using the network, can reduce the errors in the scene grasp distribution caused by incomplete geometric information. Furthermore, since the optimization is performed online for each scene, it demonstrates better robustness compared to direct network prediction and thus the results on the novel set improve significantly.

Methods	Seen			Similar			Novel		
	AP	$\mathbf{AP}_{0.8}$	$\mathbf{AP}_{0.4}$	AP	$\mathbf{AP}_{0.8}$	$\mathbf{AP}_{0.4}$	AP	$\mathbf{AP}_{0.8}$	$\mathbf{AP}_{0.4}$
Close-loop [5]	43.84	53.95	34.18	42.17	51.51	34.02	19.54	23.96	9.49
ACE-NBV [32]	46.74	56.17	38.13	46.14	55.42	38.86	21.76	26.89	12.16
Ours	55.12	65.07	48.88	52.85	62.63	46.49	24.74	30.21	12.00
All views	63.75	73.30	58.38	61.54	71.17	55.94	24.89	30.18	13.95

Table 1: Overall results compared to the state-of-the-art active grasp detection methods.

Overall Performance We compare the overall performance after 10 views of our method with previous active grasp detection methods [5, 32] on the GraspNet-1Billion benchmark, as shown in Table 1. We employ the same grasp detection network trained on the GraspNet-1Billion benchmark for all these methods. Our active grasp detection method improves the performance by 8.38%, 6.71%, 2.98% on the seen, similar and novel sets compared to ACE-NBV [32] for grasp detection in clutter, demonstrating the effectiveness of the proposed method. All views represents a complete reconstruction using all 256 views, serving as an upper-bound reference for active perception methods.

4.3 Real-world Experiments

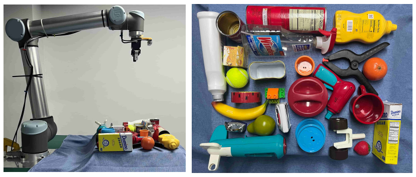


Figure 7: The robot setup of real-world experiments and the objects used for grasping.

Model	Success Rate (%)
Close-loop [5]	70.67 (53/75)
ACE-NBV [32]	62.67 (47/75)
Ours	74.67 (56/75)

Table 2: Results of the real-world grasping experiments.

We conduct real-world experiments of the proposed active grasp detection method on a 6-DoF UR-10 robot arm with a mounted RealSense D435i depth camera. The robot setup and objects used for experiments are shown in Fig. 7. We select 25 objects from the YCB dataset [6] with various sizes and shapes for grasp detection. In the experiments, each cluttered scene is composed of 5 objects and we place these objects in different poses to evaluate each scene for 3 times. We employ the grasp success rate as the metric. As shown in Table 2, our method achieves 12.00% and 4.00% improvement on success rate compared to ACE-NBV [32] and Close-loop NBV [5], respectively.

Overall	NBV Planning	Mapping	Grasp Detection	Robot Execution
3.44s	1.00s (29.07%)	0.45s (13.08%)	0.23s (6.69%)	1.76s (51.16%)

Table 3: Runtime analysis of the proposed method.

Runtime Analysis We provide a runtime analysis of the proposed active perception system, as shown in Table 3. The analysis is performed on a workstation with a single NVIDIA 3090 GPU and an AMD Ryzen 5 2600 six-core processor. The average execution time for each step is 3.44 seconds, with the robot execution accounting for approximately 50% of the total time. In the active grasp detection system, the majority of the time is consumed on NBV planning, while updating the NGF (mapping) and grasp detection take a relatively small proportion of the time. By investing some time in NBV planning, we achieve a trade-off between the performance and time cost compared to scanning the entire scene.

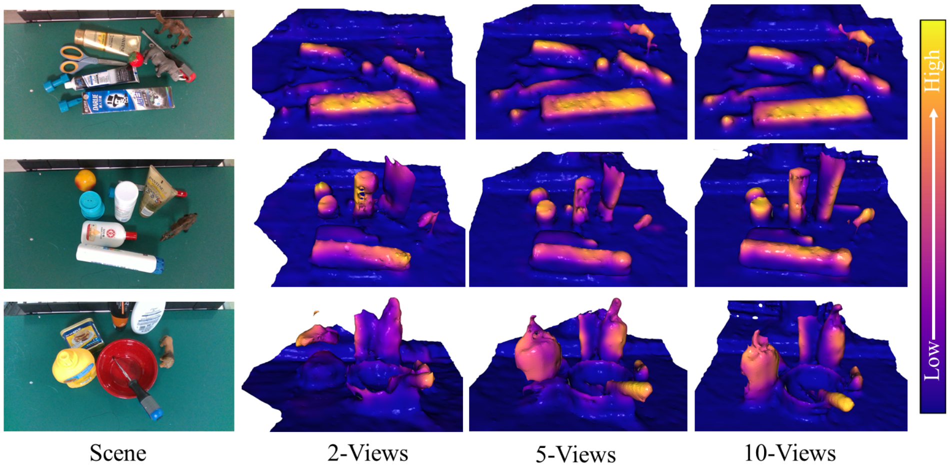


Figure 8: Visualization of the geometry and graspness extracted from NGF in different planning steps.

4.4 Visualization of the Neural Graspness Field

Fig. 8 visualizes the NGF with different perception views, where the yellow region represents a higher grasp probability. It can be observed that the NGF can not only reconstruct the 3D geometry of the scene but also jointly model the graspness. With approximately 5 views planned using active grasp detection, the NGF can effectively model the grasp distribution of objects. As more steps are taken, the details of the geometry and grasp distribution of the scene can be incrementally refined.

4.5 Visualization of the Planned Camera Trajectories

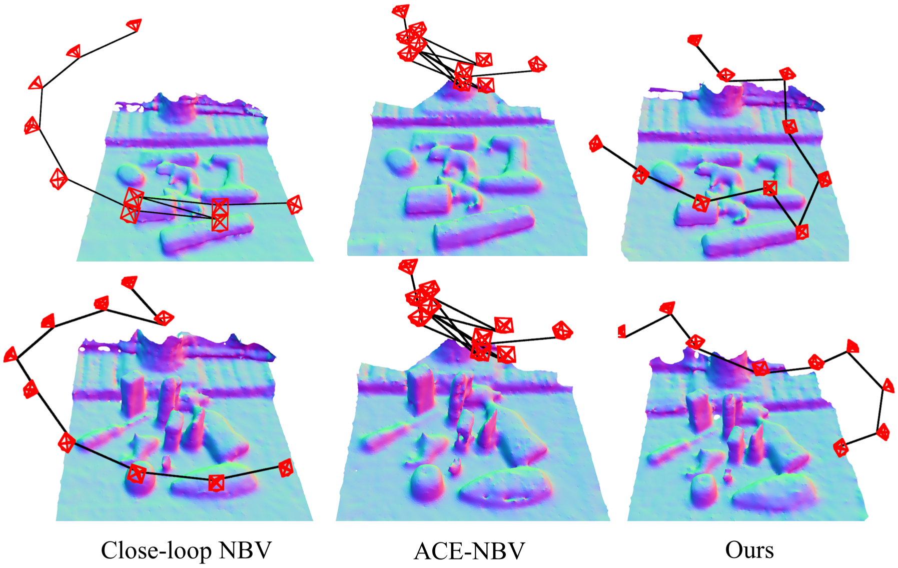


Figure 9: Visualization of the camera trajectories generated from different active grasp detection methods.

Fig. 9 illustrates the view trajectories obtained by different view planning strategies. The close-loop NBV [5] approach, which employs unobserved space as the metric, guides the camera view path to scan regions with minimal overlap with the currently observed areas, aiming to maximize scene coverage. However, this method does not prioritize the graspable regions of the objects. In contrast, ACE-NBV [32] incorporates a grasp detection network to guide view planning by selecting views with the highest grasp affordance. Nevertheless, this approach tends to repeatedly scan a limited region, potentially leading to sub-optimal local results. Compared to these methods, our proposed approach efficiently scans the grasp-correlated regions of the scene while ensuring comprehensive scene reconstruction.

5 Limitations

Our approach has two limitations. First, the time cost of NBV planning is positively correlated with the number of candidate views. When more refined view sampling is required, the planning time increases. Since our information gain computation is differentiable, this issue may be alleviated by sparse view sampling combined with pose optimization for the selected views. Second, our method cannot handle dynamic scene changes. Although efficient for static scenes, when grasping fails, *e.g.*, objects fall out of the gripper or change pose without being grasped, the robot must re-execute the perception process. Incorporating techniques used in dynamic radiance fields could potentially address this problem.

6 Conclusion

In this paper, we propose an active perception method for grasp detection by introducing the neural graspness field, which models the grasp distribution of a scene. By rendering the graspness predicted from a pre-trained network for each view, the NGF can be optimized online and reduce the noise of graspness in each view. Based on which, we introduce a graspness inconsistency-guided NBV policy to select the view with the largest inconsistency between the rendered graspness and pseudo-graspness label. Furthermore, we introduce neural graspness sampling to decode the grasp distribution from the neural representation, which benefits the position sampling of grasp pose synthesis. The experiments conducted on the simulation and real-world settings demonstrate the effectiveness of the proposed active grasp detection method.

Acknowledgment

This work is partly supported by the National Natural Science Foundation of China (62022011), the Research Program of State Key Laboratory of Software Development Environment (SKLSDE-2023ZX-14), and the Fundamental Research Funds for the Central Universities.

References

[1] Stefan Ainetter and Friedrich Fraundorfer. End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from RGB. In *IEEE International Conference on Robotics and Automation*, 2021.

[2] Ermano Arruda, Jeremy L. Wyatt, and Marek Sewer Kopicki. Active vision for dexterous grasping of novel objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.

[3] Ruzena Bajcsy, Yiannis Aloimonos, and John K. Tsotsos. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, 2018.

[4] Michel Breyer, Jen Jen Chung, Lionel Ott, Roland Siegwart, and Juan I. Nieto. Volumetric grasping network: Real-time 6 DOF grasp detection in clutter. In *Conference on Robot Learning*, 2020.

[5] Michel Breyer, Lionel Ott, Roland Siegwart, and Jen Jen Chung. Closed-loop next-best-view planning for target-driven grasping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.

[6] Berk Çalli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha S. Srinivasa, Pieter Abbeel, and Aaron M. Dollar. Yale-cmu-berkeley dataset for robotic manipulation research. *International Journal of Robotics Research*, 36(3):261–268, 2017.

[7] Xiangyu Chen, Zelin Ye, Jiankai Sun, Yuda Fan, Fang Hu, Chenxi Wang, and Cewu Lu. Transferable active grasping and real embodied dataset. In *IEEE International Conference on Robotics and Automation*, 2020.

[8] Qiyu Dai, Yan Zhu, Yiran Geng, Ciyu Ruan, Jiazhao Zhang, and He Wang. Graspnerf: Multiview-based 6-dof grasp detection for transparent and specular objects using generalizable nerf. In *IEEE International Conference on Robotics and Automation*, 2023.

[9] Amaury Depierre, Emmanuel Dellandréa, and Liming Chen. Scoring graspability based on grasp regression for better grasp prediction. In *IEEE International Conference on Robotics and Automation*, 2021.

[10] Haoshu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[11] Marcus Gualtieri and Robert Platt Jr. Viewpoint selection for grasp detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[12] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. ESLAM: efficient dense SLAM system based on hybrid representation of signed distance fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[13] Gregory Kahn, Peter Sujan, Sachin Patil, Shaunak D. Bopardikar, Julian Ryde, Kenneth Y. Goldberg, and Pieter Abbeel. Active exploration using trajectory optimization for robotic grasping in the presence of occlusions. In *IEEE International Conference on Robotics and Automation*, 2015.

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

[15] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. In *Advances in Neural Information Processing Systems*, 2022.

[16] Soomin Lee, Le Chen, Jiahao Wang, Alexander Liniger, Suryansh Kumar, and Fisher Yu. Uncertainty guided policy for active robotic 3d reconstruction using neural radiance fields. *IEEE Robotics Automation Letters*, 7(4):12070–12077, 2022.

[17] Hengduo Li, Zuxuan Wu, Abhinav Shrivastava, and Larry S. Davis. Rethinking pseudo labels for semi-supervised object detection. In *AAAI Conference on Artificial Intelligence*, 2022.

[18] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. Pointnetgpd: Detecting grasp configurations from point sets. In *IEEE International Conference on Robotics and Automation*, 2019.

[19] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Annual Conference on Computer Graphics and Interactive Techniques*, 1987.

[20] Haoxiang Ma, Modi Shi, Boyang Gao, and Di Huang. Generalizing 6-dof grasp detection via domain prior knowledge. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[21] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *European Conference on Computer Vision*, 2020.

[22] Douglas Morrison, Peter Corke, and Jürgen Leitner. Multi-view picking: Next-best-view reaching for improved grasping in clutter. In *IEEE International Conference on Robotics and Automation*, 2019.

[23] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[24] Rhys Newbury, Morris Gu, Lachlan Chumbley, Arsalan Mousavian, Clemens Eppner, Jürgen Leitner, Jeannette Bohg, Antonio Morales, Tamim Asfour, Danica Kragic, Dieter Fox, and Akansel Cosgun. Deep learning approaches to grasp synthesis: A review. *IEEE Transactions on Robotics*, 39(5):3994–4015, 2023.

[25] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[26] Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. Activenerf: Learning where to see with uncertainty estimation. In *European Conference on Computer Vision*, 2022.

[27] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[28] Yuzhe Qin, Rui Chen, Hao Zhu, Meng Song, Jing Xu, and Hao Su. S4G: amodal single-view single-shot SE(3) grasp detection in cluttered scenes. In *Conference on Robot Learning*, 2019.

[29] Liangchen Song, Yonghao Xu, Lefei Zhang, Bo Du, Qian Zhang, and Xinggang Wang. Learning from synthetic images via active pseudo-labeling. *IEEE Transactions on Image Processing*, 29:6452–6465, 2020.

[30] Andreas ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt Jr. Grasp pose detection in point clouds. *International Journal of Robotics Research*, 36(13-14):1455–1473, 2017.

[31] Chenxi Wang, Haoshu Fang, Minghao Gou, Hongjie Fang, Jin Gao, and Cewu Lu. Graspness discovery in clutters for fast and accurate grasp detection. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[32] Xuechao Zhang, Dong Wang, Sun Han, Weichuang Li, Bin Zhao, Zhigang Wang, Xiaoming Duan, Chongrong Fang, Xuelong Li, and Jianping He. Affordance-driven next-best-view planning for robotic grasping. In *Conference on Robot Learning*, 2023.

[33] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J. Davison. In-place scene labelling and understanding with implicit scene representation. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[34] Xinwen Zhou, Xuguang Lan, Hanbo Zhang, Zhiqiang Tian, Yang Zhang, and Nanning Zheng. Fully convolutional grasp detection network with oriented anchor box. In *IEEE International Conference on Intelligent Robots and Systems*, 2018.

A Appendix

Methods	Seen			Similar			Novel		
	AP	**AP**$_{0.8}$	**AP**$_{0.4}$	**AP**	**AP**$_{0.8}$	**AP**$_{0.4}$	**AP**	**AP**$_{0.8}$	**AP**$_{0.4}$
Close-loop [5]	40.07	48.06	32.05	34.74	42.32	27.78	8.68	10.61	2.93
ACE-NBV [32]	44.74	54.23	36.32	37.72	45.69	31.65	13.56	16.51	7.46
Ours	52.35	61.64	45.86	44.50	51.76	39.82	13.94	18.02	5.97
All views	61.35	70.45	55.76	55.12	62.07	49.85	19.54	23.75	9.89

Table 4: Kinect results compared to the state-of-the-art active grasp detection methods.

Overall performance on the kinect camera We compare the overall performance of kinect camera on the GraspNet-1Billion benchmark, as shown in Table 4. Our active grasp detection method improves the performance by 7.61%, 6.78%, 0.38% on the seen, similar and novel sets compared to ACE-NBV [32] for grasp detection in clutter.

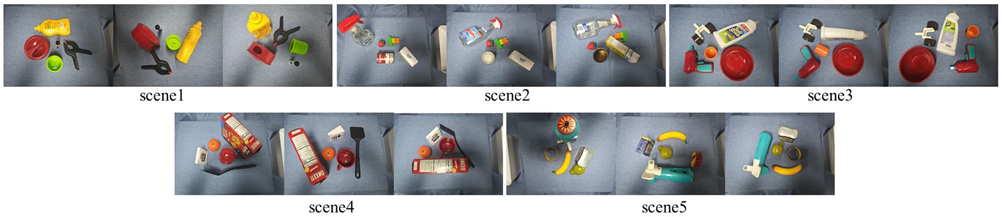


Figure 10: Object setting of the real-world experiment.

Methods	Scene1			Scene2			Scene3			Scene4			Scene5			Success Rate
	1	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	
Close-loop [5]	4	4	3	3	5	3	4	4	3	4	3	3	4	3	3	53/75
ACE-NBV [32]	4	2	3	4	3	3	3	3	4	3	2	2	4	4	3	47/75
Ours	5	3	4	4	3	4	3	3	4	4	3	4	5	4	3	56/75

Table 5: Detailed results for each scene in real-world experiments.

Details of the real-world experiment The scene setting for real-world experiment is shown in Fig. 10. In total, we constructed 5 scenes, each containing 5 objects. For each scene, we repeated the experiment 3 times by changing the poses of the objects within the scene. The number of the success attempts for each scene is provided in Table 5.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction in this paper reflect the paper's contribution accurately.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We discuss the limitations of our paper in Section 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Our paper doesn't contain theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide necessary information to reproduce the experimental results, including hyper-parameters and the related papers.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We used a public dataset in the simulation experiments, and the related code will be open sourced after the paper is accepted.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the necessary training and test details of the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide the standard deviation in the simulation experiments by inference 5 times on each scene.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the information about the computer resources for simulation and real-world experiments.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper is a purely academic work and has no societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

Justification: This paper doesn't release any data or models that have a high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The assets used in this paper are explicitly mentioned and properly respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.