Language-Driven 6-DoF Grasp Detection Using Negative Prompt Guidance

Toan Nguyen[1], Minh Nhat Vu[2,3,*], Baoru Huang[4], An Vuong[1], Quan Vuong[5], Ngan Le[6], Thieu Vo[7], and Anh Nguyen[8]

[1] FPT Software AI Center, Vietnam
[2] TU Wien, Austria
[3] AIT GmbH, Austria, *Corresponding author
[4] Imperial College London, United Kingdom
[5] Physical Intelligence, United States
[6] University of Arkansas, United States
[7] Ton Duc Thang University, Vietnam
[8] University of Liverpool, United Kingdom

Abstract. 6-DoF grasp detection has been a fundamental and challenging problem in robotic vision. While previous works have focused on ensuring grasp stability, they often do not consider human intention conveyed through natural language, hindering effective collaboration between robots and users in complex 3D environments. In this paper, we present a new approach for language-driven 6-DoF grasp detection in cluttered point clouds. We first introduce Grasp-Anything-6D, a large-scale dataset for the language-driven 6-DoF grasp detection task with 1M point cloud scenes and more than 200M language-associated 3D grasp poses. We further introduce a novel diffusion model that incorporates a new negative prompt guidance learning strategy. The proposed negative prompt strategy directs the detection process toward the desired object while steering away from unwanted ones given the language input. Our method enables an end-to-end framework where humans can command the robot to grasp desired objects in a cluttered scene using natural language. Intensive experimental results show the effectiveness of our method in both benchmarking experiments and real-world scenarios, surpassing other baselines. In addition, we demonstrate the practicality of our approach in real-world robotic applications. Our project is available at https://airvlab.github.io/grasp-anything.

Keywords: Language-Driven 6-DoF Grasp Detection, Diffusion Models

1 Introduction

Grasp detection stands as a foundational and enduring challenge in the field of robotics and computer vision [9, 28]. This task involves identifying a suitable configuration for the robotic hand that stably grasps the objects, facilitating the effective manipulation capability in the robot's operating environment. Traditional grasp detection methods have predominantly focused on ensuring the

stability of the detected grasp pose, while often neglecting the human intention. This limitation underscores a large gap between current approaches and real-world user-specified requirements [81]. The integration of human intention conveyed through natural language, is therefore crucial to help robots perform complex tasks more flexibly. This enables users to communicate task specifications more intuitively and comprehensively to the intelligent robot, facilitating a more effective human-robot collaboration.

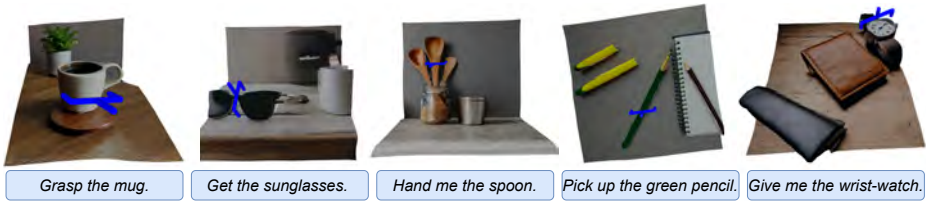

Fig. 1: We tackle the task of language-driven 6-DoF grasp detection in cluttered 3D point cloud scenes.

In recent years, thanks to advancements in large language models [11, 12, 54] and large vision-language models [33, 35, 59], there has been a surge of interest in language-driven robotics research [6, 7, 14, 47, 60, 78, 97]. This research field focuses on developing intelligent robots that can understand and respond to human linguistic commands. For example, SayCan [7] and PaLM-E [11] are robotic language models designed to provide instructions for robots operating in real-world environments. Trained on large-scale data, RT-1 [6] and RT-2 [97] are robotic systems capable of performing low-level actions in response to natural language commands. While significant progress has been made in the field, it is noteworthy that only a few works have addressed the task of language-driven grasp detection [46, 60, 72, 73, 75, 81]. Furthermore, these methods still exhibit considerable shortcomings. Particularly, while the authors in [46, 72] solely focus on single-object scenarios, the works in [73, 75, 81] restrict grasp detection to 2D configurations. These limitations prevent the robot from capturing the complexity of real-world 3D and multi-object scenarios. In this research, we address these limitations by training a new system that detects language-driven 6-DoF grasp poses, with a focus on grasping objects within diverse and complex scenes represented as 3D point clouds.

We first introduce a new dataset, namely **Grasp-Anything-6D**, as a large-scale dataset for language-driven 6-DoF grasp detection in 3D point clouds. Our dataset builds upon the Grasp-Anything dataset [81] and incorporates a state-of-the-art depth estimation method [5] to support 2D to 3D projection, and manual correction to ensure the dataset quality. Specifically, Grasp-Anything-6D provides one million (1M) 3D point cloud scenes with comprehensive object grasping prompts and dense 6-DoF grasp pose annotations. With its extensive volume, our dataset enables the capability of 6-DoF grasp detection using language instructions directly from the point cloud. Empirical demonstrations show that our dataset successfully facilitates grasp detection in diverse and complex scenes, both in vision-based experiments and real-world robotic settings.

With the new dataset in hand, we propose a new diffusion model to address the challenging problem of language-driven 6-DoF grasp detection called **LGrasp6D**. We opt for diffusion models due to their recent impressive results in various generation tasks [24, 50, 68], including image synthesis [13, 49], video generation [55, 87], and point cloud generation [39, 44]. However, the application of diffusion models to grasp detection remains under-explored [46, 76]. Unlike previous works that mostly focus on language-driven grasp detection in 2D image [73, 75, 81] or in 3D point cloud with single object [46, 72], our work proposes a new diffusion model for language-driven 6-DoF grasp detection in cluttered 3D point cloud environments. In practice, language-driven 6-DoF grasp detection is a fine-grained task driven by the language, e.g., *"Grasp the blue cup"* and *"Grasp the black cup"* are for two different objects in the scene. Therefore, we introduce a new negative prompt guidance learning strategy to tackle this fine-grained nature. The main motivation of this strategy is to learn a negative prompt embedding that can encapsulate the notion of other undesired objects in the scene. When being applied in the generation process, the learned negative prompt embedding explicitly guides the grasp pose toward the desired object while avoiding unwanted ones. Our LGrasp6D method is an end-to-end pipeline that enables humans to command the robot to grasp desired objects in a cluttered scene using a natural language prompt. Figure 1 illustrates examples of our language-driven grasp detection in 3D point clouds. To summarize, our contributions are three-fold:

- We propose Grasp-Anything-6D, a large-scale dataset for language-driven 6-DoF grasp detection in 3D point clouds.
- We propose a new diffusion model that learns and applies negative prompt guidance, significantly enhancing the grasp detection process.
- We demonstrate that our dataset and the proposed method outperform other approaches and enable successful real-world robotic manipulation.

2 Related Works

Robot Grasp Detection. Several works for robot grasp detection addressed the task on 2D images [25, 32, 61, 94]. Thanks to recent advancements in 3D perception [26, 48, 56, 57], 6-DoF grasp detection in 3D point clouds is gaining increasing interest in both computer vision and robotics communities. In general, two main lines of approaches have been employed for this problem. The first line [22, 36, 42, 43] involves sampling various grasp candidates across the input point cloud, followed by validation using a grasp evaluator network. The primary drawback of methods in the first line lies in their inefficiency in terms of speed, attributed to their multi-stage structure. In contrast, the second line of research detects the grasp poses in an end-to-end manner [20, 48, 58, 83], achieving a more favorable balance in terms of the time-accuracy tradeoff. For instance, Qin *et al.* [58] presented a novel gripper contact model and a single-shot neural network to predict amodal grasp proposals, while Wang *et al.* [83] proposed the concept

of graspness to detect the scene graspable areas. However, most of the existing 6-DoF grasp detection methods do not take into account language as the input. In this work, we follow the end-to-end approach. Our method integrates language instructions into the grasp detection process, ensuring that the detected grasp pose is aligned with the user-specified requirements.

Language-Guided Robotic Manipulation. Amidst the remarkable strides of large language models [8, 12, 54] and large vision-language models [33, 35, 59], several recent works have harnessed language semantics for multiple tasks of robot manipulation [6, 21, 47, 62, 97]. For instance, the authors in [21] presented a framework that learns meaningful skills from language-based expert demonstrations. Nguyen *et al.* [47] utilized language to detect open-vocabulary affordance for 3D point cloud objects. More recently, the authors in [97] proposed a family of models that learn generalizable and semantically aware policies derived from fine-tuning large vision-language models trained on web-scale data. Besides, the task of language-guided grasp detection is also under active exploration. However, approaches in this research direction present several limitations. Specifically, the works in [46, 72] only addressed single-object scenarios. The authors in [73, 75, 80, 81] exclusively detected 2D rectangle grasp poses. More recently, the method in [60] required multiple viewpoints of the scene to build the language field, which is not always obtainable. In contrast to these works, our method is capable of detecting language-driven 6-DoF grasp poses in cluttered single-view point cloud scenes, making it well-suitable for real-world robotic applications.

Diffusion Probabilistic Models. Diffusion models are a class of neural generative models, based on the stochastic diffusion process in Thermodynamics [67]. In this setting, a sample from the data distribution is gradually noised by the forward diffusion process. Then, a neural network learns the reverse process to gradually denoise the sample. First introduced by [67], diffusion models have been further simplified and accelerated [24, 68], and improved significantly [3, 50, 70, 86]. In recent years, many works have explored applying diffusion models for various generation problems, such as image synthesis [13, 89], scene synthesis [31, 82], and human motion generation [74, 85]. In robotics, diffusion models have also been applied to many problems ranging from policy learning [10, 93], task and motion planning [38, 76] to robot design [84]. However, few works have adopted diffusion models for the task of grasp detection [46, 76]. Notably, none of them consider the task of language-driven grasping in 3D cluttered point clouds. To address this challenging task, we propose a novel diffusion model that incorporates a new negative prompt guidance learning approach. This strategy assists in guiding the generation process toward the desired grasp distributions while steering away from unwanted ones. The effectiveness of our proposed approach is demonstrated through comprehensive experiments.

3 The Grasp-Anything-6D Dataset

Our Grasp-Anything-6D dataset is built upon the Grasp-Anything dataset [81]. Leveraging foundation models [53, 64], Grasp-Anything is a large-scale dataset for

2D language-driven grasp detection. This dataset consists of 1M RGB images and ≈3M objects, substantially surpassing prior datasets in diversity and volume. To bring the problem from 2D to 3D, we first leverage the state-of-the-art depth estimation method ZoeDepth [5] to estimate the depth map given the input RGB images of Grasp-Anything. Subsequently, we perform projection and manual verification to ensure the quality of our dataset.

3D Scenes and 6-DoF Grasps Construction. For a given 2D scene in the Grasp-Anything dataset [81], we first employ ZoeDepth [5] to get the depth map for the image and establish the 3D point cloud scene with the camera model assumption of a 55-degree field of view and central principal point. We select the field of view of 55 degrees because it leads to 3D scenes representing real object scales. Next, to bring a 2D grasp configuration to 3D, we first infer the 3D position using the center of the 2D rectangle grasp in the image. Since in the Grasp-Anything dataset, the position of the 2D grasp may not necessarily be integers, we employ bilinear interpolation to calculate its corresponding 3D position by considering the 3D coordinates of neighboring pixels. The position determines the translation part of the grasp representation. For the rotation part, we utilize the angle of the 2D rectangle grasp and map it to 3D to rotate the 6-DoF grasp pose accordingly. The width of the 6-DoF grasp is derived from the width of the 2D grasp. Adhering to the Robotiq 2F-140 gripper specifications [63], we establish the maximum grasp width as 202.1 mm, and discard any grasps exceeding this threshold. The overview of our 3D scenes and 6-DoF grasps construction process is illustrated in Figure 2. We maintain the same scene description and grasping prompts as in the Grasp-Anything dataset. Additionally, we infer the 3D masks on the point cloud scene for every object in the grasp list using the corresponding segmentation masks in 2D.

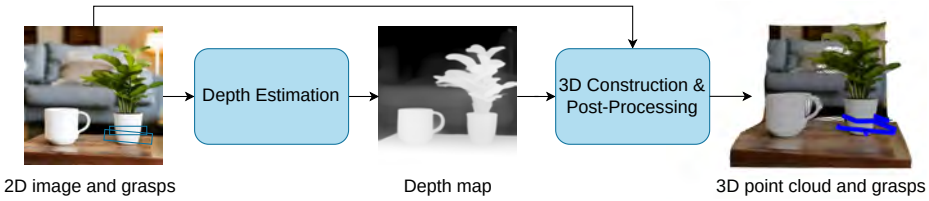

Fig. 2: Overview of Grasp-Anything-6D dataset construction pipeline.

Post-Processing. After converting the 2D scenes and grasps to 3D, we manually check for the collision of the 6-DoF grippers and point cloud scenes, as well as whether the grippers can stably grasp the objects. These problems may occur since the depth estimation network [5] may not always bring good results. Concretely, we remove the grasp poses that collide with the point cloud scene and those whose closing volume between the fingers does not intersect the object determined by its 3D mask. As a result, our Grasp-Anything-6D dataset consists of 1M point cloud scenes, with comprehensive grasping prompts, and 200M corresponding dense and high-quality 6-DoF grasp poses.

4 Grasp Detection using Negative Prompt Guidance

4.1 Motivation

Diffusion models have recently shown remarkable performance across various generation tasks. This makes it a promising choice for our problem, where grasp detection can be viewed as a generation process conditioned on both the point cloud scene and the language prompt. The main contribution of our diffusion model is a novel negative prompt guidance learning strategy. This is motivated by the notion that generating a grasp for a specific object can benefit significantly from guidance away from unwanted objects in the scene. Our LGrasp6D leverages this by integrating learning the negative prompt embedding into the training process alongside the conventional denoising objective. Our target for the negative prompt embedding is to capture the notion of other undesired objects in the scene. The learned negative prompt guidance is then applied in the sampling to assist the grasp detection process.

4.2 Language-Driven 6-DoF Grasp Detection

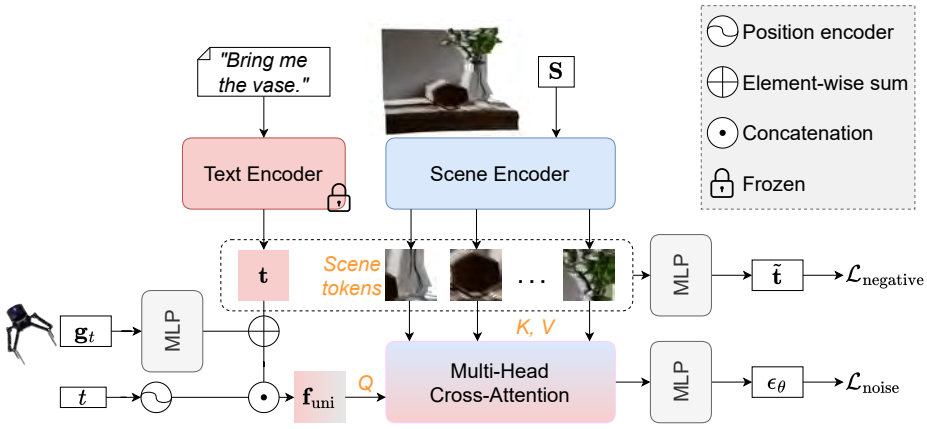

Fig. 3: Overview of our denoising network. In addition to predicting the noise, our denoising network is trained to learn the negative prompt embedding, which is supervised by the text embeddings associated with other unwanted objects in the same scene.

Forward Process. We use the $\mathfrak{se}(3)$ Lie algebra [65] to represent the translation and rotation of our grasp poses. We use the $\mathfrak{se}(3)$ representation since it allows us to conveniently perform the operators of addition and multiplication by a scalar required by our forward and reverse diffusion processes. The grasp pose is then represented as the concatenation of $\mathfrak{se}(3)$ vector and the grasp width. Note that one can easily convert between the $\mathfrak{se}(3)$ and 4×4 transformation matrix representation using the logarithm map and exponential map [52]. Given a target grasp pose $\mathbf{g}_0$ in the training dataset, in the forward process, we obtain a sequence of perturbed grasp poses by gradually adding to it small amounts

of Gaussian noise in T steps. The noise step sizes are specified by a predefined variance schedule $\{\beta_t \in (0,1)\}_{t=1}^{T}$. The forward process is formulated as:

$$q\left(\mathbf{g}_t|\mathbf{g}_{t-1}\right) = \mathcal{N}\left(\mathbf{g}_t; \sqrt{1-\beta_t}\mathbf{g}_{t-1}, \beta_t\mathbf{I}\right). \tag{1}$$

The perturbed pose at any arbitrary time step t can be obtained by:

$$\mathbf{g}_t = \sqrt{\bar{\alpha}_t}\mathbf{g}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{2}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_t$ with $\alpha_t = 1-\beta_t$ and $\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$. When $T \to \infty$, $\mathbf{g}_T$ is equivalent to $\mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$ [24].

Denoising Network. Our denoising network approximates the added noise described in the forward process by incorporating both the conditions of the point cloud scene and the textual prompt specifying the target object. Additionally, our network learns a vector representation serving as a negative prompt guidance. In our framework, this representation is guided by the available textual prompts associated with other objects within the scene. The details of our denoising network are shown in Figure 3.

The denoising network first encodes the grasp pose $\mathbf{g}_t$ at a specific time step t using a grasp encoder MLP. The scene encoder encodes the point cloud scene $\mathbf{S}$ to n_s scene embedding tokens. In our framework, we use PointNet++ [57] as the underlying architecture for the scene encoder. For the textual prompt, we employ a pretrained text encoder to get a text embedding $\mathbf{t}$. We use sinusoidal positional embedding [24] to embed the time step t to a high-dimensional vector. Afterward, we form the unified representation $\mathbf{f}_{\text{uni}}$ of the grasp pose, the textual prompt, and the time step. In concrete, we concatenate the time embedding with the element-wise sum of the grasp embedding and the text embedding. Subsequently, we adopt the multi-head cross-attention mechanism to capture the intricate relationships among input components. Specifically, the query for the cross-attention is the unified feature $\mathbf{f}_{\text{uni}}$ while the n_s scene tokens serve as keys and values. The output of the cross-attention module is then fed to an MLP to obtain the predicted noise $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \mathbf{t}, t\right)$. We supervise the noise prediction by optimizing the simplified objective function as described in [24]:

$$\mathcal{L}_{\text{noise}} = \mathbb{E}_{\boldsymbol{\epsilon}, \mathbf{g}_0, \mathbf{S}, \mathbf{t}, t}\left[\left\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \mathbf{t}, t\right) - \boldsymbol{\epsilon}\right\|^2\right]. \tag{3}$$

Negative Prompt Learning. Along with estimating the noise, the denoising network also produces the negative prompt embedding $\tilde{\mathbf{t}}$. We subtract the text embedding $\mathbf{t}$ from the scene tokens, compute the average over n_s resulting vectors, and then pass the output through an MLP to get $\tilde{\mathbf{t}}$. Our purpose for $\tilde{\mathbf{t}}$ is that it can encapsulate the notion of other objects in the same scene. Hence, our objective is to minimize the distance between $\tilde{\mathbf{t}}$ and the negative text embeddings which are text embeddings corresponding to other objects. Specifically, we define the loss function for the learning of negative prompt embedding as:

$$\mathcal{L}_{\text{negative}} = D\left(\tilde{\mathbf{t}}, \bar{\mathbf{T}} = \{\bar{\mathbf{t}}_i\}_{i=1}^{m}\right) = \min_{i=1}^{m}\left\|\tilde{\mathbf{t}} - \bar{\mathbf{t}}_i\right\|_2^2, \tag{4}$$

where $D(\cdot)$ denotes the distance function, $\bar{\mathbf{T}} = \{\bar{\mathbf{t}}_i\}_{i=1}^{m}$ is the set of m negative text embeddings. In training, we simultaneously optimize both the denoising loss $\mathcal{L}_{\text{noise}}$ and the loss for negative prompt embedding learning $\mathcal{L}_{\text{negative}}$.

Reverse Process with Negative Prompt Guidance. Different from conventional diffusion models, our reverse diffusion process utilizes the negative prompt embedding learned during the training to guide the grasp pose toward the desired object while avoiding unwanted ones. Our generation process can be formulated as a conditional distribution $p\left(\mathbf{g}|\mathbf{S}, \mathbf{t}, \neg\tilde{\mathbf{t}}\right)$. The negation sign of $\tilde{\mathbf{t}}$ indicates that we aim to sample the grasp pose with the absence of the $\tilde{\mathbf{t}}$ prompt condition. We begin with the following proposition:

Proposition 1. *The conditional distribution* $p\left(\mathbf{g}|\mathbf{S}, \mathbf{t}, \neg\tilde{\mathbf{t}}\right)$ *can be factorized as*

$$p\left(\mathbf{g}|\mathbf{S}, \mathbf{t}, \neg\tilde{\mathbf{t}}\right) \propto p\left(\mathbf{g}|\mathbf{S}\right) \frac{p\left(\mathbf{g}|\mathbf{t}, \mathbf{S}\right)}{p\left(\mathbf{g}|\tilde{\mathbf{t}}, \mathbf{S}\right)}. \tag{5}$$

Proof. See Supplementary Material.

With Equation 5, alongside detecting grasps conditioning on the scene and the user-specified prompt via $p\left(\mathbf{g}|\mathbf{S}\right)$ and $p\left(\mathbf{g}|\mathbf{t}, \mathbf{S}\right)$, we can now seamlessly incorporate the negative prompt guidance into our reverse process via $p\left(\mathbf{g}|\tilde{\mathbf{t}}, \mathbf{S}\right)$.

Remark 1. Liu *et al.* [37] demonstrated how diffusion models can be composed based on their connection to energy-based models [15]. We recall this relationship in detail in our Supplementary. Consequently, following the expression in [37], we can formulate our compositional denoising step in the reverse process as:

$$\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \mathbf{t}, \neg\tilde{\mathbf{t}}, t\right) = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \varnothing, t\right) + w\left(\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \mathbf{t}, t\right) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \tilde{\mathbf{t}}, t\right)\right). \tag{6}$$

In Equation 6, $p\left(\mathbf{g}|\mathbf{S}\right)$, $p\left(\mathbf{g}|\mathbf{t}, \mathbf{S}\right)$ and $p\left(\mathbf{g}|\tilde{\mathbf{t}}, \mathbf{S}\right)$ are parameterized by $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \varnothing, t\right)$, $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \mathbf{t}, t\right)$ and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \tilde{\mathbf{t}}, t\right)$ respectively. $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \tilde{\mathbf{t}}, t\right)$ is the output of the denoising network when the learned negative prompt embedding $\tilde{\mathbf{t}}$ is plugged in as the text embedding. w is a hyperparameter that controls the strength of the negative guidance. $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \varnothing, t\right)$ is the predicted noise when the text condition is discarded. In training, we learn $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \varnothing, t\right)$ by randomly masking out the text embedding with a predefined probability p_{mask}. Given the denoising step defined in Equation 6, we can now sample grasps from Gaussian noise by applying the reverse process from timestep T back to 0 using the following formulation:

$$\mathbf{g}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{g}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}\left(\mathbf{g}_t, \mathbf{S}, \mathbf{t}, \neg\tilde{\mathbf{t}}, t\right)\right) + \sqrt{\beta_t}\mathbf{z}, \tag{7}$$

where $\mathbf{z} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$ if the time step $t > 1$, else $\mathbf{z} = \mathbf{0}$.

4.3 Training and Sampling

We define the overall loss function for training as $\mathcal{L} = 0.9\mathcal{L}_{\text{noise}} + 0.1\mathcal{L}_{\text{negative}}$. We utilize the pretrained CLIP ViT-B/32 text encoder [59] for our text encoder and

freeze it during training. We set the number of timesteps to $T = 200$, and set the forward diffusion variances to increase linearly from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$. The probability of masking out the text embedding is set to $p_{\text{mask}} = 0.1$. The whole network is trained over 200 epochs on a cluster of 8 A100 GPUs with a batch size of 128. We use Adam optimizer [27] with the learning rate 10^{-3} and the weight decay 10^{-4}. In sampling, we set the negative guidance scale to $w = 0.2$. To obtain a favorable inference speed, we pre-compute the scene tokens, the text embedding $\mathbf{t}$, and the negative prompt embedding $\tilde{\mathbf{t}}$ since they are independent of the timestep. This precomputation substantially reduces the detection time, making our method feasible for practical implementation on real robots.

5 Experiments

In this section, we evaluate the effectiveness of our LGrasp6D trained on the Grasp-Anything-6D dataset via several vision-based and real robot experiments.

5.1 Language-Driven 6-DoF Grasp Detection Results

Baselines. We evaluate our method against generative approaches for 6-DoF grasp detection, which are 6-DoF GraspNet [42], SE(3)-DF [76], and 3DAP-Net [46]. We adapt the frameworks of these baselines to integrate textual input into the detection process. To ensure a fair comparison, we utilize the CLIP ViT-B/32 [59] as the text encoder for all methods. We also include our method without utilizing negative prompt guidance (denotes as Ours w.o. NPG) as an additional baseline for comparison. Detailed implementation information for all baselines is available in our Supplementary Material.

Setup. We train all baselines on 80% scenes of the Grasp-Anything-6D dataset and evaluate them on the remaining 20%. For each pair of point cloud scene-textual prompts, we detect 64 grasp poses for evaluation. To benchmark the methods' detection capabilities, we use three metrics, which are the coverage rate [42], earth mover's distance [76], and collision-free rate [95]. The coverage rate (CR) [42] measures how well the space of ground-truth grasps is covered by the detected grasps. The earth mover's distance (EMD) [76] evaluates the dissimilarity between the distributions of ground-truth grasps and the detected ones. Finally, the collision-free rate (CFR) [95] assesses the occurrence of collisions between the gripper of the detected grasps and the scene. The final results for all metrics are averaged across all scene-text prompt pairs. Since latency is a critical factor for any robotics applications, we additionally benchmark the inference speeds of all methods using the inference time in seconds (IT). Specifically, for each baseline, we calculate its inference time for detecting 1000 grasp poses across 1000 different scene-text pairs and take the average result.

Quantitative Results. Table 1 shows the results of language-driven 6-DoF grasp detection on our Grasp-Anything-6D dataset. The outcomes indicate the advantages of our methods, even without negative prompt guidance, over other baselines. Our complete method consistently achieves the highest scores across

all three metrics for grasp detection capability. It significantly surpasses the second-best method, which is our framework without negative prompt guidance, with large margins of 0.1235 on CR, 0.2249 on EMD, and 0.0370 on CFR. This highlights the effectiveness of our proposed negative prompt guidance learning. Regarding latency, our methods achieve competitive IT scores compared to other diffusion model-based methods (SE(3)-DF and 3DAPNet). Although 6-DoF GraspNet achieves the best IT, it is important to note that this is a variational autoencoder-based method requiring only a single decoding step, and its results on the remaining metrics are poor.

Baseline	**CR↑**	**EMD↓**	**CFR↑**	**IT↓**
6-DoF GraspNet [42]	0.3802	0.8035	0.6900	**0.4216**
SE(3)-DF [76]	0.4290	0.7565	0.7325	1.7233
3DAPNet [46]	0.4777	0.7381	0.7213	3.4274
LGrasp6D (ours) w.o. NPG	0.5459	0.6262	0.7336	1.4328
LGrasp6D (ours)	**0.6694**	**0.4013**	**0.7706**	1.4832

Table 1: Results on Grasp-Anything-6D dataset.

Qualitative Results. We present the qualitative results of all baselines in detecting language-driven grasps in Figure 4. Point cloud scenes are selected from our Grasp-Anything-6D dataset. The results indicate that LGrasp6D exhibits a significantly stronger capability in detecting language-driven grasp poses compared to the others. Specifically, our method excels at focusing on the desired objects, whereas other methods often get distracted by undesired ones. More qualitative results are provided in our Supplementary Material.

Accelerating Detection. While latency is critical for robot applications, diffusion models are notorious for their low inference speed [69]. Despite our method achieving a competitive inference speed, as shown in Table 1, we continue to seek even faster models with comparable performance. Hence, we benchmark our LGrasp6D employing the fast reversion technique of denoising diffusion implicit models (DDIM) [68], with numbers of sampling steps of 200 (the original one), 100, 50, 20, and 10. The results are shown in Table 2. We can observe that decreases in the sampling step lead to decreases in performance. However, all the variants still outperform other baselines in Table 1. Regarding the inference time, these accelerated models obtain significantly better inference speed compared to the original one. The variant with 50 steps already surpasses the 6-DoF

Baseline	**CR↑**	**EMD↓**	**CFR↑**	**IT↓**
LGrasp6D - 10 steps	0.5611	0.5273	0.7368	**0.0726**
LGrasp6D - 20 steps	0.6425	0.4300	0.7580	0.1464
LGrasp6D - 50 steps	0.6439	0.4254	0.7639	0.3991
LGrasp6D - 100 steps	0.6522	0.4110	0.7633	0.8427
LGrasp6D - 200 steps	**0.6694**	**0.4013**	**0.7706**	1.4832

Table 2: DDIM accelerating results.

Baseline	**CR↑**	**EMD↓**	**CFR↑**
6-DoF GraspNet [42]	0.3498	0.8501	0.6927
SE(3)-DF [76]	0.3892	0.7622	0.7205
3DAPNet [46]	0.4491	0.7434	0.7092
LGrasp6D (ours) w.o. NPG	0.5208	0.6422	0.7385
LGrasp6D (ours)	**0.6420**	**0.4197**	**0.7683**

Table 3: Cross-dataset results.

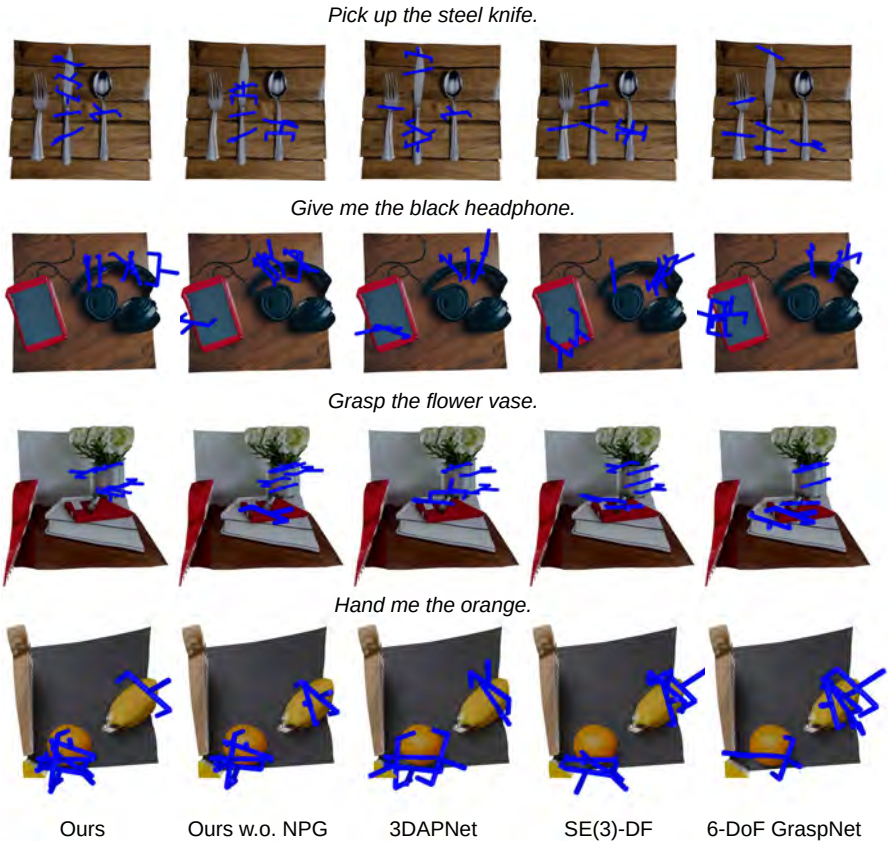

Fig. 4: Language-driven 6-DoF grasp detection qualitative results.

GraspNet method (0.3991 seconds compared to 0.4216 seconds). Although the variant with 10 steps achieves the best detection speed, it is not recommended as its detection performance is severely compromised.

5.2 Generalization Analysis

Cross-Dataset Transferability. Given the extensive scale and diversity of our Grasp-Anything-6D dataset, we expect that our proposed method, trained on this dataset, will exhibit strong generalization capabilities when tested on a distinct dataset. Specifically, we evaluate the language-driven grasp detection performance of models trained on Grasp-Anything-6D using the Contact-GraspNet dataset [71]. This dataset comprises point cloud scenes of cluttered tabletops synthesized using objects and 6-DoF grasps from [19] and a random camera view. We utilize the object category names as textual prompts for language-driven grasping. The findings showcased in Table 3 exhibit a comparable trend to those observed in the Grasp-Anything-6D dataset. Our method continues to outperform its counterparts across all three metrics, with the version lacking negative prompt guidance following behind. Furthermore, the slight performance decrease

on the new dataset is noteworthy. They underscore the efficacy of our dataset, as models trained on it demonstrate strong generalizability.

Grasp Detection in the Wild. Figure 5 illustrates results of our method in point cloud scenes captured from diverse real-world environments, such as working desks, bathrooms, and kitchens. As we can observe, the detected grasp poses exhibit satisfactory quality. This indicates that despite being trained on synthetic data, our approach effectively generalizes to real-world environments.

Fig. 5: In the wild language-driven 6-DoF grasp detection results.

5.3 Negative Prompt Guidance Analysis

We offer a more intuitive understanding of how negative prompt guidance influences the grasp detection results. Specifically, we ultimately sample 1000 grasp poses for each object in a given point cloud scene for both cases: our framework with negative prompt guidance and the one without it. We then employ t-SNE [77] to visualize all grasp poses on a 2D plane. The results are depicted in Figure 6, where grasp data points of the same color are detected for the same object. We can observe that negative prompt guidance significantly assists our method in discriminatively detecting grasp poses for different objects. Conversely, without negative prompt guidance, detecting grasp poses for one object is seriously confused by other ones. This further highlights the effectiveness of our proposed approach. More comparison results can be viewed in Figure 8.

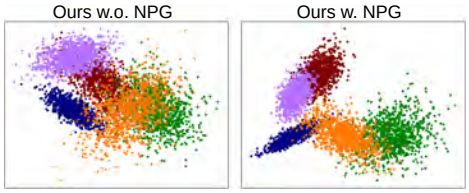

Fig. 6: Negative prompt guidance analysis.

Fig. 7: Failure cases.

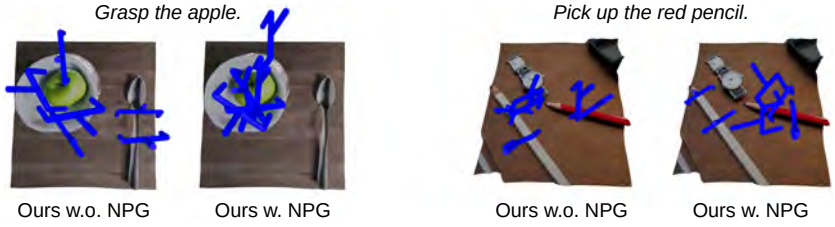

Fig. 8: Comparisons between models with and without negative prompt guidance.

5.4 Robotics Experiment

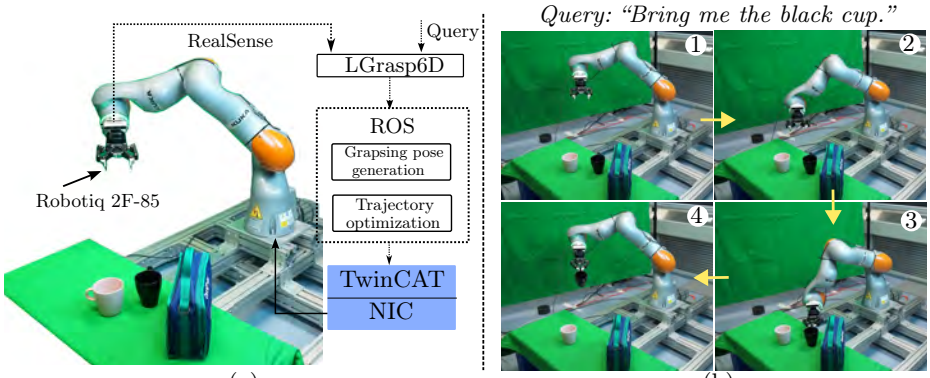

Fig. 9: (a) Experiment setup. (b) Example of the execution of a grasping task.

Baseline	Input Modality	Single	Cluttered
GG-CNN [41] + CLIP [59]	RGB-D	0.10	0.07
CLIPORT [66]	RGB-D	0.27	0.30
Det-Seg-Refine [1] + CLIP [59]	RGB-D	0.30	0.23
GR-ConvNet [30] + CLIP [59]	RGB-D	0.33	0.30
CLIP-Fusion [90]	RGB-D	0.40	0.40
LGD [80]	RGB-D	**0.43**	**0.42**
6-DoF GraspNet [42]	Point clouds	0.31	0.27
SE(3)-DF [76]	Point clouds	0.35	0.34
3DAPNet [46]	Point clouds	0.36	0.34
LGrasp6D (ours) w.o. NPG	Point clouds	0.38	0.36
LGrasp6D (ours)	Point clouds	**0.43**	**0.42**

Table 4: Robotic language-driven grasp detection results.

Setup. In Figure 9, we present the robotic experiment conducted on a KUKA robot. The success rate is used for evaluation. Using an Intel RealSense D435i depth camera, the detected 6-DoF grasp poses are mapped to robot's 6-DoF end-effector poses using transformation matrices obtained via hand-eye calibration [45]. The trajectory planner and the computed torque controller [4, 79] are employed for the grasp execution. We use two computers for the experiment. The first computer executes the real-time control software Beckhoff TwinCAT of 8 kHz update frequency, while the second one utilizes the Robot Operating System (ROS) for the camera and the Robotiq 2F-85 gripper. Using EtherCAT protocol, PC1 communicates with the robot via a network interface card (NIC). The inference process is performed on PC2, utilizing an NVIDIA RTX 3080 graphic card. Our assessment encompasses both single-object and cluttered scenarios, involving a diverse set of real-world daily objects. To ensure the reliability, we repeat each experiment for all methods a total of 45 times.

Baselines. Besides the baselines utilized in previous experiments, we additionally compare LGrasp6D with language-supported versions of state-of-the-art

2D grasp detectors, including GR-CNN [30], Det-Seg-Refine [1], GG-CNN [41], CLIPORT [66], CLIP-Fusion [90], and LGD [80]. In all cases, we employ the pretrained CLIP ViT-B/32 [59] as the text encoder. The implementation details of all baselines can be found in our Supplementary Material.

Results. Our method, incorporating negative prompt guidance, demonstrates better performance compared to other baselines in Table 4. Additionally, even though LGrasp6D is trained on Grasp-Anything-6D, a synthesis dataset exclusively created by foundation models, it still yields commendable results when applied to real-world objects.

6 Discussion

Despite promising results, it is important to acknowledge that our method still has limitations, as illustrated in Figure 7. The left case depicts an example of grasping the wrong object, while the middle one illustrates a detected grasp colliding with an object. The final case shows our method detecting a grasp that mis-targets the desired object. These underscore the challenges in language-driven 6-DoF grasping, indicating its need for further investigation.

For future research, we aim to enhance the performance by incorporating more advanced techniques to capture the intricate correlation among input modalities. In addition, our work can be extended to address language-driven 6-DoF grasping at both the part-level and task-level. For instance, instead of object-specific prompts like *"Grasp the knife"*, one can provide more detailed prompts such as *"Grasp the knife by its handle"* or *"Grasp the knife for cutting"*. Furthermore, it would be beneficial to extend our approach to accommodate different types of robot end-effectors to enhance the flexibility and adaptability of our framework. Lastly, integrating learning language-driven 6-DoF grasp detection with robotic control could create a more effective end-to-end pipeline, connecting human instructions directly to low-level robot actions.

7 Conclusion

We address the task of language-driven 6-DoF grasp detection in cluttered point clouds. In particular, we have presented the Grasp-Anything-6D dataset as a large-scale dataset for the task with 1M point cloud scenes. We have introduced a novel LGrasp6D diffusion model incorporating the new concept of negative prompt guidance learning. Our proposed negative prompt guidance assists in tackling the fine-grained challenge of the language-driven grasp detection task, directing the detection process toward the desired object by steering away from undesired ones. Empirical results demonstrate the superiority of our method over other baselines in various settings. Furthermore, extensive experiments validate the efficacy of our approach in real-world environments and robotic applications.

A Theoretical Findings

A.1 Proof of Proposition 1

Proof. We have the following derivation:

$$
\begin{aligned}
p\left(\mathbf{g}|\mathbf{S},\mathbf{t},\neg\tilde{\mathbf{t}}\right) &= \frac{p\left(\mathbf{g},\mathbf{S},\mathbf{t},\neg\tilde{\mathbf{t}}\right)}{p\left(\mathbf{S},\mathbf{t},\neg\tilde{\mathbf{t}}\right)} \\
&\propto p\left(\mathbf{g},\mathbf{S},\mathbf{t},\neg\tilde{\mathbf{t}}\right) && p\left(\mathbf{S},\mathbf{t},\tilde{\mathbf{t}}\right) \text{ is a constant} \\
&= p\left(\neg\tilde{\mathbf{t}}|\mathbf{g},\mathbf{t},\mathbf{S}\right) p\left(\mathbf{g},\mathbf{t},\mathbf{S}\right) \\
&= p\left(\neg\tilde{\mathbf{t}}|\mathbf{g}\right) p\left(\mathbf{g},\mathbf{t},\mathbf{S}\right) && \tilde{\mathbf{t}},\ \mathbf{t},\ \mathbf{S} \text{ are independent} \\
&= p\left(\neg\tilde{\mathbf{t}}|\mathbf{g}\right) p\left(\mathbf{t},\mathbf{S}|\mathbf{g}\right) p\left(\mathbf{g}\right) \\
&= p\left(\neg\tilde{\mathbf{t}}|\mathbf{g}\right) p\left(\mathbf{t},\mathbf{S}|\mathbf{g}\right) \frac{p\left(\mathbf{g}|\mathbf{S}\right) p\left(\mathbf{S}\right)}{p\left(\mathbf{S}|\mathbf{g}\right)} && \text{Using Bayes' Theorem} \\
&\propto p\left(\mathbf{g}|\mathbf{S}\right) p\left(\mathbf{t},\mathbf{S}|\mathbf{g}\right) \frac{p\left(\neg\tilde{\mathbf{t}}|\mathbf{g}\right)}{p\left(\mathbf{S}|\mathbf{g}\right)} && p\left(\mathbf{S}\right) \text{ is a constant} \\
&= p\left(\mathbf{g}|\mathbf{S}\right) p\left(\mathbf{t},\mathbf{S}|\mathbf{g}\right) \frac{p\left(\mathbf{g},\neg\tilde{\mathbf{t}}\right)}{p\left(\mathbf{g}\right) p\left(\mathbf{S}|\mathbf{g}\right)} \\
&= p\left(\mathbf{g}|\mathbf{S}\right) p\left(\mathbf{t},\mathbf{S}|\mathbf{g}\right) \frac{p\left(\mathbf{g}\right) - p\left(\mathbf{g},\tilde{\mathbf{t}}\right)}{p\left(\mathbf{g}\right) p\left(\mathbf{S}|\mathbf{g}\right)} \\
&= p\left(\mathbf{g}|\mathbf{S}\right) p\left(\mathbf{t},\mathbf{S}|\mathbf{g}\right) \frac{1 - p\left(\tilde{\mathbf{t}}|\mathbf{g}\right)}{p\left(\mathbf{S}|\mathbf{g}\right)} \\
&\propto p\left(\mathbf{g}|\mathbf{S}\right) p\left(\mathbf{t},\mathbf{S}|\mathbf{g}\right) \frac{1}{p\left(\mathbf{S}|\mathbf{g}\right) p\left(\tilde{\mathbf{t}}|\mathbf{g}\right)} \\
&= p\left(\mathbf{g}|\mathbf{S}\right) p\left(\mathbf{t},\mathbf{S}|\mathbf{g}\right) \frac{p\left(\mathbf{g}\right)}{p\left(\mathbf{S}|\mathbf{g}\right) p\left(\mathbf{g},\tilde{\mathbf{t}}\right)} \\
&= p\left(\mathbf{g}|\mathbf{S}\right) p\left(\mathbf{t},\mathbf{S}|\mathbf{g}\right) \frac{p\left(\mathbf{g}\right)}{p\left(\mathbf{S}|\mathbf{g},\tilde{\mathbf{t}}\right) p\left(\mathbf{g},\tilde{\mathbf{t}}\right)} && \tilde{\mathbf{t}},\ \mathbf{t},\ \mathbf{S} \text{ are independent} \\
&= p\left(\mathbf{g}|\mathbf{S}\right) p\left(\mathbf{t},\mathbf{S}|\mathbf{g}\right) \frac{p\left(\mathbf{g}\right)}{p\left(\mathbf{S},\mathbf{g},\tilde{\mathbf{t}}\right)} \\
&= p\left(\mathbf{g}|\mathbf{S}\right) \frac{p\left(\mathbf{t},\mathbf{S}|\mathbf{g}\right)}{p\left(\tilde{\mathbf{t}},\mathbf{S}|\mathbf{g}\right)} \\
&= p\left(\mathbf{g}|\mathbf{S}\right) \frac{p\left(\mathbf{g}|\mathbf{t},\mathbf{S}\right) p\left(\mathbf{t},\mathbf{S}\right)}{p\left(\mathbf{g}|\tilde{\mathbf{t}},\mathbf{S}\right) p\left(\tilde{\mathbf{t}},\mathbf{S}\right)} && \text{Using Bayes' Theorem} \\
&\propto p\left(\mathbf{g}|\mathbf{S}\right) \frac{p\left(\mathbf{g}|\mathbf{t},\mathbf{S}\right)}{p\left(\mathbf{g}|\tilde{\mathbf{t}},\mathbf{S}\right)} && p\left(\mathbf{t},\mathbf{S}\right),\ p\left(\tilde{\mathbf{t}},\mathbf{S}\right) \text{ are constants}
\end{aligned}
$$

The assumption of independence between $\tilde{\mathbf{t}}$, $\mathbf{t}$, and $\mathbf{S}$ reflects general real-world scenarios where human language prompts can be arbitrary and are not necessarily dependent on the scene. Proposition 1 is now proved. ■

A.2 Connection between Diffusion and Energy-Based Models

The connection between diffusion and energy-based models is not restricted to our problem. We will recall this connection in the general context of any generation task.

Diffusion Models. Denoising diffusion probabilistic models (DDPMs) construct a forward diffusion process by gradually adding Gaussian noise to the ground truth sample $\mathbf{x}_0$ through T timesteps. A neural network then learns to revert this noise perturbation process. Both the forward and the reverse processes are modeled as Markov chains:

$$q\left(\mathbf{x}_{0:T}\right) = q(\mathbf{x}_0) \prod_{t=1}^{T} q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right), \quad p_\theta\left(\mathbf{x}_{T:0}\right) = p\left(\mathbf{x}_T\right) \prod_{t=T}^{1} p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right), \tag{8}$$

where $q\left(\mathbf{x}_0\right)$ is the ground truth data distribution and $p\left(\mathbf{x}_T\right)$ is a standard Gaussian prior $\mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$.

In the reverse process, each step is parameterized by a Gaussian distribution with mean $\boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right)$ and covariance matrix $\tilde{\beta}_t\mathbf{I}$, where $\tilde{\beta}_t = \beta_t \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}$. Following the simplification in [24], we can keep the covariance fixed and formulate the reverse distribution as:

$$p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right) = \mathcal{N}\left(\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right), \beta_t\mathbf{I}\right). \tag{9}$$

Subsequently, an individual step in sampling can be performed by:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right)\right) + \sqrt{\beta_t}\mathbf{z}, \tag{10}$$

where $\mathbf{z} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$ if the time step $t > 1$, else $\mathbf{z} = \mathbf{0}$.

Energy-Based Models. Energy-Based Models (EBMs) [16, 17, 23, 51] are a family of generative models in which the data distribution is modeled by an unnormalized probability density. Given a sample $\mathbf{x} \in \mathbb{R}^D$, its probability density is defined as:

$$p_\theta\left(\mathbf{x}\right) \propto e^{-E_\theta(\mathbf{x})}, \tag{11}$$

where the energy function $E_\theta\left(\mathbf{x}\right) : \mathbb{R}^D \to \mathbb{R}$ is a learnable neural network. Langevin dynamics [17] is then used to sample from the unnormalized probability distribution to iteratively refine the generated sample $\mathbf{x}$:

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}} E_\theta\left(\mathbf{x}_{t-1}\right) + \sqrt{\lambda}\mathbf{z}, \tag{12}$$

where λ is the predefined step size and $\mathbf{z} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$.

The sampling procedure used by diffusion models in Equation 10 is functionally similar to the sampling procedure used by EBMs in Equation 12. In both settings, samples are iteratively refined starting from Gaussian noise, with a small amount of noise removed at each iterative step. At a timestep t, in

DDPMs, samples are updated using a learned denoising network $\boldsymbol{\epsilon}(\mathbf{x}_t, t)$, while in EBMs, samples are updated via the gradient of the energy function $\nabla_{\mathbf{x}} E_\theta(\mathbf{x}_t) \propto \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}_t)$. Thus, we can view a DDPM as an implicitly parameterized EBM and apply similar composition techniques for EBMs as in [15] for DDPMs. More details about compositional DDPMs can be referred to in [37].

B Remark on Related Works

Diffusion Models in Robotics. Recent years have witnessed diffusion models being applied to several robotic tasks. For instance, in policy learning, diffusion models have been employ for multi-task robotic manipulation [88], long-horizon skill planning [40], or cross-embodiment skill discovery [91]. Besides, the ability of diffusion models to generate realistic videos over a long horizon has enabled new applications in the context of robotics [2,18,29]. For example, Du *et al.* [18] proposed to learn universal planning strategy via text-to-video generation. In robot development, diffusion models have been leveraged for manipulator construction [92] or soft robot co-design [84]. Although diffusion models have also been explored for the task of grasp detection [46, 76], none of them address the task of detecting language-driven 6-DoF grasp poses in 3D cluttered scenes.

Language-Driven Grasp Detection. Language-driven grasp detection has emerged as an active research domain in recent years. Previous works have primarily focused on addressing this task using 2D images [72, 75, 80, 81, 90]. For instance, the authors in [73] presented a method that combines object grounding and task grounding to tackle the task of task-oriented grasp detection, while Xu *et al.* [90] proposed to jointly model vision, language, and action for grasping in clutter. Despite achieving promising results, these approaches are limited in their ability to handle complex 3D environments. To overcome this limitation, recent research has explored language-driven grasp detection in 3D data. In particular, Nguyen *et al.* [46] addressed the task of affordance-guided grasp detection for 3D point cloud objects, while Tang *et al.* [72] leveraged knowledge from large language models for task-oriented grasping. However, these methods are designed for single-object scenarios, limiting their applicability in cluttered settings. In contrast, our method is capable of detecting language-driven 6-DoF grasp poses in cluttered point cloud scenes.

C Dataset Statistics

Table 5 shows our dataset statistics and comparisons to other 6-DoF grasp datasets.

D Implementation Details

D.1 Grasp Detection Methods for 3D Point Clouds

– Our LGrasp6D: The text embedding $\mathbf{t}$ produced by the pretrained CLIP ViT-B/32 and the negative prompt embedding $\tilde{\mathbf{t}}$ are 512-dimensional (512-

Dataset	Text?	#objects	#grasps	#scenes	Cluttered?	Data type	Annotation
GraspNet-1B [20]	✗	88	∼1.2B	97K	✓	Real	Analysis
6-DoF GraspNet [42]	✗	206	∼7M	206	✗	Sim.	Sim.
ACRONYM [19]	✗	8872	∼17.7M	-	✗	Sim.	Sim.
Ours	✓	∼3M	∼200M	1M	✓	Synth.	Analysis

Table 5: Dataset statistics.

D). We employ a PointNet++ [48] architecture for our scene encoder. The number of points per scene is 8192. The scene encoder extracts $n_S = 128$ scene tokens of 256-D. We employ 4 heads for the multi-head cross-attention block, with the output of 512-D. The timestep t is encoded by a sinusoidal positional encoder to obtain a 64-D vector. To speed up the training process, we freeze the scene encoder after the first 100 epochs.

- 6-DoF GraspNet: We modified the model to integrate the text embedding derived from the CLIP text encoder [59] into both the encoder and decoder of the variational autoencoder. Since our dataset does not include negative grasp poses, we refrained from employing additional refinement steps. This is also to ensure a fair comparison with other methods. The remaining architecture, hyperparameters, and training loss are inherited from the original work.
- SE(3)-DF [76]: We append the text embedding extracted by the CLIP text encoder [59] to the input of the feature encoder. As the signed distance function is not available for our 3D point clouds, we exclude the signed distance function learning objective from the framework. The remaining architecture, hyperparameters, and training loss are retained from the original work.
- 3DAPNet [46]: 3DAPNet jointly addresses the tasks of language-guided affordance detection and pose detection. To adapt this method to our problem, we remove the affordance learning objective from the original framework. The remaining architecture, hyperparameters, and training loss are inherited from the original work.

D.2 Grasp Detection Methods for Images

Methods in this section are used in our robotic experiment in Section 5.2 of our main paper. They are trained on the RGB-D images to predict rectangle grasp poses inherited from Grasp-Anything [81]. Specifically, each grasp pose is represented by $(g_x, g_y, g_w, g_h, g_\theta)$, where (g_x, g_y) is the center of the rectangle, (g_w, g_h) are the width and height of the rectangle and g_θ is the grasp angle.

- Language-supported versions of GG-CNN [41], Det-Seg-Refine [1], and GR-ConvNet [30]: We slightly modify these baselines by adding a component to fuse the input image and text prompt. Specifically, we utilize the CLIP text encoder [59] to extract the text embedding. Additionally, we employ the ALBEF architecture presented in [34] to fuse the text embedding and the

visual features. The remaining training loss and architecture are inherited from the original works.
- CLIPORT [66]: The original CLIPORT framework learns a policy π, which is not directly applicable to our setting. Therefore, we modify its architecture's final layers by adding an MLP to output the rectangle grasp pose.
- CLIP-Fusion [90]: We follow the cross-attention module in CLIP-Fusion. The final MLP in the architecture is modified to output five parameters of the rectangle grasp pose.
- LGD [80]: We report results from the original paper.

E Ablation Studies

Negative Guidance Scale. Recall that the negative guidance scale w plays an important role in controlling the strength of the negative guidance in the sampling process. We conduct an ablation study of the effect of the change in w on the grasp detection performance. Table 6 demonstrates that values of $w = 0.2$ (used in experiments in the main paper) and $w = 0.5$ yield the best results, whereas excessively small or large values of w detrimentally affect performance.

w	CR↑	EMD↓	CFR↑
0.1	0.6573	0.4183	0.7629
0.2	**0.6649**	<u>0.4013</u>	**0.7706**
0.5	<u>0.6607</u>	**0.4005**	<u>0.7698</u>
1.0	0.6531	0.4310	0.7622
2.0	0.6372	0.4521	0.7563

Table 6: Grasp detection performance with varying negative guidance scale.

Loss Function. Table 7 shows the performances when using varying ratios of $\mathcal{L}_{\text{negative}}$ (called ζ) and $\mathcal{L}_{\text{noise}}$ (which is $1-\zeta$). The results indicate that setting ζ to 0.1 or 0.2 yields strong accuracy, while either too high (0.4) or low (0.05) values significantly hurt the performance.

ζ	CR↑	EMD↓	CFR↑
0.05	0.6237	0.4500	0.7420
0.1	**0.6733**	**0.4029**	<u>0.7754</u>
0.2	<u>0.6664</u>	<u>0.4093</u>	**0.7812**
0.4	0.5833	0.5298	0.7326

Table 7: Loss function analysis.

Backbone Variation. We conduct an ablation study on two different scene encoder backbone, i.e., PointNet++ [57] and Point Transformer [96], and two different pretrained text encoders, i.e., CLIP ViT-B/32 [59] and BERT [12]. The number of parameters and results of all variants are shown in Table 8. We observe that in general, PointNet++ performs better than Point Transformer, and CLIP performs better than BERT. Variants using Point Transformer run significantly slower than those using PointNet++ due to the larger and more complicated architecture. Particularly, the combination of Point Transformer and CLIP obtains a competitive grasp detection performance compared to that of PointNet++ and CLIP; however, its inference time is considerably higher. This pattern is also observed when comparing CLIP and BERT text encoders. The gap in grasp detection performance between variants utilizing the CLIP ViT-B/32 text encoder and those employing BERT is substantial, highlighting CLIP's superiority in semantic language-vision understanding.

Scene Encoder	Text Encoder	CR↑	EMD↓	CFR↑	IT↓
Point Transformer [96] (23M)	BERT [12] (110M)	0.6428	0.4597	0.7583	2.0137
Point Transformer [96] (23M)	CLIP [59] (63M)	<u>0.6591</u>	<u>0.4167</u>	**0.7725**	1.9755
PointNet++ [57] (2M)	BERT [12] (110M)	0.6430	0.4225	0.7622	<u>1.5449</u>
PointNet++ [57] (2M)	CLIP [59] (63M)	**0.6649**	**0.4013**	<u>0.7706</u>	**1.4832**

Table 8: Scene encoder and text encoder backbone variation.

F Robotic Experiments

We show 20 real-world daily objects used in robotic experiments in Figure 10. The sequences of actions when the KUKA robot grasps objects are presented in Figure 11. Figure 12 further shows the detection result of our LGrasp6D on point clouds captured by a RealSense camera mounted on the robot. The robotic experiments demonstrate that although our method is trained on a synthetic Grasp-Anything-6D dataset, it can generalize to detect grasp poses in real-world scenarios. More illustrations can be found in our Demonstration Video.

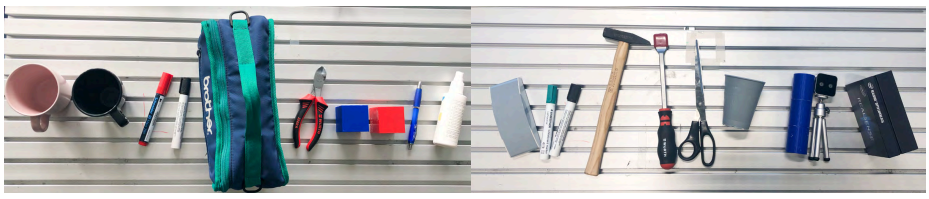

Fig. 10: Set of 20 objects used in the robotic experiments.

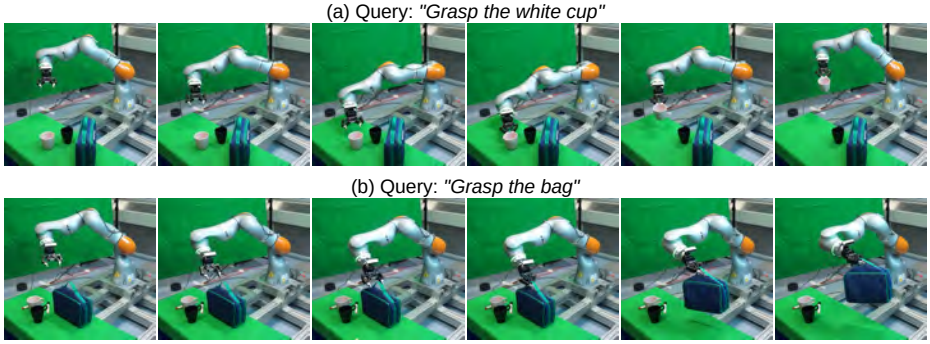

Fig. 11: Snapshots of two example robotic experiments.

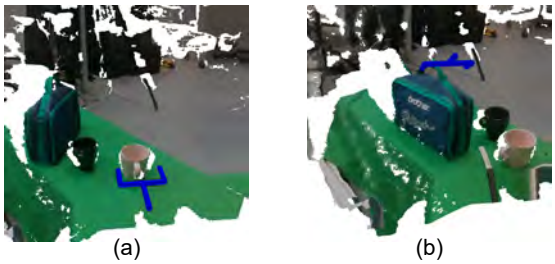

Fig. 12: Detection results in robotic experiments. Point clouds are captured from a RealSense camera with experiments in Figure 11.

G Additional Qualitative Results

Figure 13 shows more qualitative results to demonstrate the effectiveness of our method in detecting grasp poses for different objects.

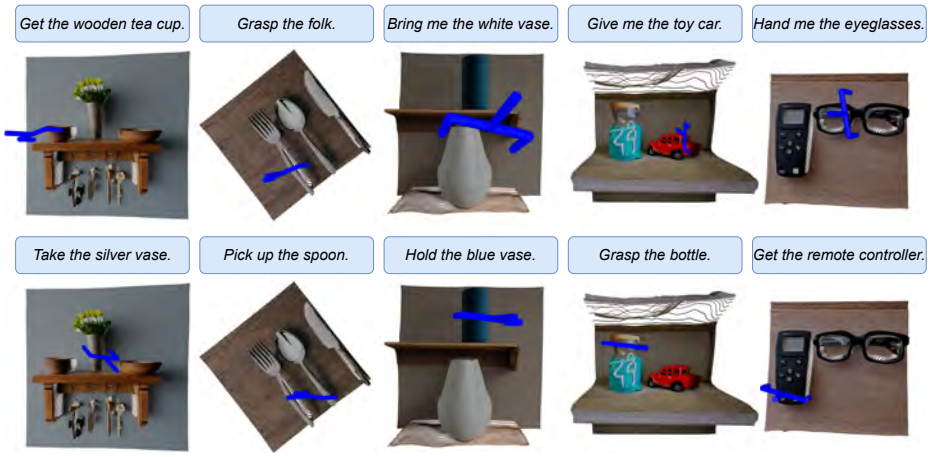

Fig. 13: Additional qualitative results.

References

1. Stefan Ainetter and Friedrich Fraundorfer. End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb. In *ICRA*, 2021.
2. Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. *NeurIPS*, 2024.
3. Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 2021.
4. Florian Beck, Minh Nhat Vu, Christian Hartl-Nesic, and Andreas Kugi. Singularity avoidance with application to online trajectory optimization for serial manipulators. *IFAC-PapersOnLine*, 2023.
5. Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.
6. Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. In *RSS*, 2023.
7. Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *CoRL*, 2023.
8. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020.
9. Shehan Caldera, Alexander Rassau, and Douglas Chai. Review of deep learning methods in robotic grasp detection. *Multimodal Technologies and Interaction*, 2018.
10. Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *RSS*, 2023.
11. Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *JMLR*, 2023.
12. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.
13. Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021.
14. Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. In *ICML*, 2023.
15. Yilun Du, Shuang Li, and Igor Mordatch. Compositional visual generation with energy based models. *NeurIPS*, 2020.
16. Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy-based models. In *ICML*, 2021.
17. Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *NeurIPS*, 2019.
18. Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *NeurIPS*, 2024.

19. Clemens Eppner, Arsalan Mousavian, and Dieter Fox. Acronym: A large-scale grasp dataset based on simulation. In *ICRA*, 2021.
20. Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *CVPR*, 2020.
21. Divyansh Garg, Skanda Vaidyanath, Kuno Kim, Jiaming Song, and Stefano Ermon. Lisa: Learning interpretable skill abstractions from language. *NeurIPS*, 2022.
22. Minghao Gou, Hao-Shu Fang, Zhanda Zhu, Sheng Xu, Chenxi Wang, and Cewu Lu. Rgb matters: Learning 7-dof grasp poses on monocular rgbd images. In *ICRA*, 2021.
23. Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, and Richard Zemel. Learning the stein discrepancy for training and evaluating energy-based models without sampling. In *ICML*, 2020.
24. Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020.
25. Yun Jiang, Stephen Moseson, and Ashutosh Saxena. Efficient grasping from rgbd images: Learning using a new rectangle representation. In *ICRA*, 2011.
26. Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *ICCV*, 2023.
27. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
28. Kilian Kleeberger, Richard Bormann, Werner Kraus, and Marco F Huber. A survey on learning-based robotic grasping. *Current Robotics Reports*, 2020.
29. Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B. Tenenbaum. Learning to act from actionless videos through dense correspondences. In *ICLR*, 2024.
30. Sulabh Kumra, Shirin Joshi, and Ferat Sahin. Antipodal robotic grasping using generative residual convolutional neural network. In *IROS*, 2020.
31. Seoyoung Lee and Joonseok Lee. Posediff: Pose-conditioned multimodal diffusion model for unbounded scene synthesis from sparse inputs. In *WACV*, 2024.
32. Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *IJRR*, 2015.
33. Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.
34. Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *NeurIPS*, 2021.
35. Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022.
36. 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 *ICRA*, 2019.
37. Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *ECCV*, 2022.
38. Weiyu Liu, Tucker Hermans, Sonia Chernova, and Chris Paxton. Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects. In *CoRL Workshop*, 2022.
39. Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 2021.
40. Utkarsh Aashu Mishra, Shangjie Xue, Yongxin Chen, and Danfei Xu. Generative skill chaining: Long-horizon skill planning with diffusion models. In *CoRL*, 2023.

41. Douglas Morrison, Peter Corke, and Jürgen Leitner. Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. In *RSS*, 2018.
42. Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *ICCV*, 2019.
43. Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Chris Paxton, and Dieter Fox. 6-dof grasping for target-driven object manipulation in clutter. In *ICRA*, 2020.
44. George Kiyohiro Nakayama, Mikaela Angelina Uy, Jiahui Huang, Shi-Min Hu, Ke Li, and Leonidas Guibas. Difffacto: Controllable part-based 3d point cloud generation with cross diffusion. In *ICCV*, 2023.
45. Huy Nguyen and Quang-Cuong Pham. On the **Covariance** of $\mathbf{X}$ in $\mathbf{AX} = \mathbf{XB}$. *T-RO*, 2018.
46. Toan Nguyen, Minh Nhat Vu, Baoru Huang, Tuan Van Vo, Vy Truong, Ngan Le, Thieu Vo, Bac Le, and Anh Nguyen. Language-conditioned affordance-pose detection in 3d point clouds. In *ICRA*, 2024.
47. Toan Nguyen, Minh Nhat Vu, An Vuong, Dzung Nguyen, Thieu Vo, Ngan Le, and Anh Nguyen. Open-vocabulary affordance detection in 3d point clouds. In *IROS*, 2023.
48. Peiyuan Ni, Wenguang Zhang, Xiaoxiao Zhu, and Qixin Cao. Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds. In *ICRA*, 2020.
49. Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.
50. Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021.
51. Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *AAAI*, 2020.
52. Arkadij L Onishchik and Ernest B Vinberg. *Lie groups and algebraic groups*. Springer Science & Business Media, 2012.
53. OpenAI. Introducing chatgpt. *OpenAI Blog*, 2022.
54. OpenAI. GPT-4 technical report. *CoRR*, 2023.
55. OpenAI. Video generation models as world simulators. *OpenAI Technical Report*, 2024.
56. Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.
57. Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017.
58. 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 *CoRL*, 2020.
59. Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
60. Adam Rashid, Satvik Sharma, Chung Min Kim, Justin Kerr, Lawrence Yunliang Chen, Angjoo Kanazawa, and Ken Goldberg. Language embedded radiance fields for zero-shot task-oriented grasping. In *CoRL*, 2023.
61. Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *ICRA*, 2015.
62. Allen Z Ren, Bharat Govil, Tsung-Yen Yang, Karthik R Narasimhan, and Anirudha Majumdar. Leveraging language for accelerated learning of tool manipulation. In *CoRL*, 2023.

63. Robotiq. Robotiq 2f-140. *2F-85 and 2F-140 Grippers*, 2018.
64. Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.
65. Hans Samelson. *Notes on Lie algebras*. Springer Science & Business Media, 2012.
66. Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *CoRL*, 2022.
67. Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.
68. Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.
69. Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.
70. Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *NeurIPS*, 2021.
71. Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *ICRA*, 2021.
72. Chao Tang, Dehao Huang, Wenqi Ge, Weiyu Liu, and Hong Zhang. Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping. *RA-L*, 2023.
73. Chao Tang, Dehao Huang, Lingxiao Meng, Weiyu Liu, and Hong Zhang. Task-oriented grasp prediction with visual-language inputs. *IROS*, 2023.
74. Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. In *ICLR*, 2022.
75. Georgios Tziafas, XU Yucheng, Arushi Goel, Mohammadreza Kasaei, Zhibin Li, and Hamidreza Kasaei. Language-guided robot grasping: Clip-based referring grasp synthesis in clutter. In *CoRL*, 2023.
76. Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *ICRA*, 2023.
77. Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008.
78. Tuan Van Vo, Minh Nhat Vu, Baoru Huang, Toan Nguyen, Ngan Le, Thieu Vo, and Anh Nguyen. Open-vocabulary affordance detection using knowledge distillation and text-point correlation. In *ICRA*, 2024.
79. Minh Nhat Vu, Florian Beck, Michael Schwegel, Christian Hartl-Nesic, Anh Nguyen, and Andreas Kugi. Machine learning-based framework for optimally solving the analytical inverse kinematics for redundant manipulators. *Mechatronics*, 2023.
80. An Dinh Vuong, Minh Nhat Vu, Baoru Huang, Nghia Nguyen, Hieu Le, Thieu Vo, and Anh Nguyen. Language-driven grasp detection. In *CVPR*, 2024.
81. An Dinh Vuong, Minh Nhat Vu, Hieu Le, Baoru Huang, Binh Huynh, Thieu Vo, Andreas Kugi, and Anh Nguyen. Grasp-anything: Large-scale grasp dataset from foundation models. In *ICRA*, 2024.
82. An Dinh Vuong, Minh Nhat Vu, Toan Nguyen, Baoru Huang, Dzung Nguyen, Thieu Vo, and Anh Nguyen. Language-driven scene synthesis using multi-conditional diffusion model. *NeurIPS*, 2024.
83. Chenxi Wang, Hao-Shu Fang, Minghao Gou, Hongjie Fang, Jin Gao, and Cewu Lu. Graspness discovery in clutters for fast and accurate grasp detection. In *ICCV*, 2021.

84. Tsun-Hsuan Johnson Wang, Juntian Zheng, Pingchuan Ma, Yilun Du, Byungchul Kim, Andrew Spielberg, Josh Tenenbaum, Chuang Gan, and Daniela Rus. Diffusebot: Breeding soft robots with physics-augmented generative diffusion models. *NeurIPS*, 2024.
85. Yin Wang, Zhiying Leng, Frederick WB Li, Shun-Cheng Wu, and Xiaohui Liang. Fg-t2m: Fine-grained text-driven human motion generation via diffusion model. In *ICCV*, 2023.
86. Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, Mingyuan Zhou, et al. Patch diffusion: Faster and more data-efficient training of diffusion models. *NeurIPS*, 2024.
87. Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, 2023.
88. Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, and Katerina Fragkiadaki. Unifying diffusion models with action detection transformers for multi-task robotic manipulation. In *CoRL*, 2023.
89. Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *CVPR*, 2023.
90. Kechun Xu, Shuqi Zhao, Zhongxiang Zhou, Zizhang Li, Huaijin Pi, Yifeng Zhu, Yue Wang, and Rong Xiong. A joint modeling of vision-language-action for target-oriented grasping in clutter. In *ICRA*, 2023.
91. Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery. In *CoRL*, 2023.
92. Xiaomeng Xu, Huy Ha, and Shuran Song. Dynamics-guided diffusion model for robot manipulator design. *arXiv preprint arXiv:2402.15038*, 2024.
93. Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *CoRL*, 2023.
94. Hanbo Zhang, Xuguang Lan, Site Bai, Xinwen Zhou, Zhiqiang Tian, and Nanning Zheng. Roi-based robotic grasp detection for object overlapping scenes. In *IROS*, 2019.
95. Binglei Zhao, Hanbo Zhang, Xuguang Lan, Haoyu Wang, Zhiqiang Tian, and Nanning Zheng. Regnet: Region-based grasp network for end-to-end grasp detection in point clouds. In *ICRA*, 2021.
96. Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, 2021.
97. Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *CoRL*, 2023.