IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

GraspControl: Text-Sketch Instruction as an Interface for Controllable Grasp Synthesis

Xiaopeng Wen[1], Songtao Tian[1],Yi Sun[1]* *Member, IEEE*

***Abstract*—Large vision-language models have been shown to perform complex tasks. However, aligning language instructions with object visual information to enable general inference for robotic grasping poses a significant challenge. To tackle this issue, we introduce GraspControl, a method that leverages grasp language instructions and sketches of objects to control the generation of grasps. Initially, we construct a dataset that augments language instructions with position and orientation information of grasps, and visual information with sketches of the gripper and target objects. Subsequently, we develop a model capable of generating 2D grasp sketches given grasp language and 2D object sketches as input prompts, thereby bridging the gap between the linguistic and visual representations of the object to be grasped. These generated 2D grasp sketches serve as an innovative input modality for grasp synthesis, directing the creation of 3D object models and corresponding 3D grasp poses through a 3D reconstruction module. Furthermore, we incorporate a multi-modal attention loss to ensure the consistency between high-level semantic grasp features and intricate low-level visual features, with a particular emphasis on the grasping area of the object. We evaluate the capabilities of our grasp approach through extensive experiments in both simulated and real-world robotic scenarios. The experimental results confirm that our method can execute grasps in complex environments.**

***Index Terms*—Grasp generation, language instructions, editing sketches, human intent.**

I. INTRODUCTION

GRASP synthesis [1] in complex environments is a prerequisite for intelligent robots to manipulate objects. To achieve successful grasps, previous vision-based techniques rely on either model-based optimization methods or data-driven deep learning methods [2]–[8]. As different tasks may require completely different grasps for the same object, such as different grasps generated for the same knife to complete different "handing" and "cutting" task, these techniques still struggle to accurately generate grasps that execute specific tasks in the absence of task-oriented semantic instruction. This drives the development from solely vision based grasp to Visual-Language combined Grasp (VLG).

VLG [9]–[15] aims to exploit the advantages of both language and visual information to generate grasps suitable for post-grasp manipulation tasks. However, directly combining these two types of information is difficult to achieve successful grasps, as the alignment between language and image prompts is not a trivial task. Existing large language datasets are limited in terms of spatial representations for grasp such as the instruction of approaching direction for a grasping task, making it difficult for robots to identify how an object can be grasped for multiple tasks. The image features extracted from RGB images or point clouds is not clearly associated with the parts of the object to be grasped. Consequently, current state-of-the-art visual language models trained on typical vision-language tasks cannot directly solve robotic grasping tasks according to human intentions.

To address this challenge, we delve into the intricate realm of fine-grained grasp synthesis by utilizing Text-Sketch instructions as an interface for diverse grasp synthesis, coined as GraspControl. Regarding text prompts for grasping tasks, we have enriched the descriptions to include the position and orientation of the grasp. To provide a more nuanced level of fine-grained expression for the object being grasped, we extract a 2D sketch of the object. As depicted in Fig. 1, we introduce a diffusion model [16] that leverages both grasp text prompts and object sketches to generate 2D grasp sketches (i.e., sketches with a grasp), which act as an innovative input query modality for grasp synthesis, akin to image retrieval using sketches. Both the grasp text prompts and object sketches contribute to align task-level semantic features with object-level visual features, enabling the generated grasp sketches, guided by text, accurately correspond to the grasping positions of the objects. Furthermore, we employ these 2D grasp sketches to direct the reconstruction of 3D objects and their corresponding grasps in 3D space. From the generated grasps, we extract the grasp 6D poses via a registration module, facilitating subsequent robot grasping.

The main contributions of this work are as follows:

- We propose a grasp synthesis method that exploits text prompts containing the position and orientation of grasps, in conjunction with sketches of object contours, to guide the grasp synthesis according to human intentions.
- We generate grasp sketches using a grasp-oriented vision-language model, which is guided by both language instructions and object sketches to align the generated grasps with the objects.
- We utilize these grasp sketches as grasp prompts to guide the 3D reconstruction of objects and hand grasps, which in turn provide the grasp poses for subsequent robot grasping.

Extensive simulated and real-world experiments demonstrate the effectiveness of our GraspControl method. Compared

Manuscript received 15 July 2025; accepted 27 October 2025. This letter was recommended for publication by Editor Júlia Borràs Sol upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Natural Science Foundation of China under Grants 62373075 and U1708263. (Xiaopeng Wen and Songtao Tian contributed equally to the work.) (Corresponding authors: Yi Sun.)

The authors are with the School of Information and Communication Engineering, Dalian University of Technology, Dalian, 116024, China. pengpeng@mail.dlut.edu.cn,duttiansongt@163.com, lslwf@dlut.edu.cn

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

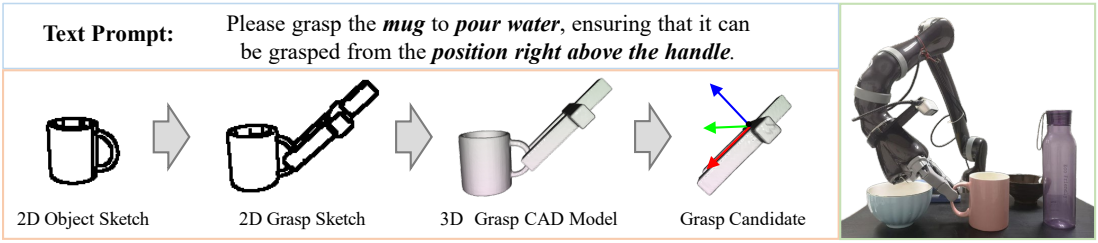


Fig. 1. Inference pipeline of GraspControl.

to state-of-the-art grasping algorithms, this method achieves a high quality of diverse grasps.

II. RELATED WORK

Visual-based Grasping: Robot grasp is a fundamental problem in robotics. Currently, most methods for generating grasps on objects are visual-based grasping. For example, 6-DOF GraspNet [4] builds a variational autoencoder (VAE) to sample grasp candidates effectively. Fang et al. [5] construct a large-scale grasp pose dataset named GraspNet-1Billion, using a 3D grasp detection network to learn grasp parameters and achieve end-to-end robust grasping. As grasp pose annotation is time-consuming, TransGrasp [7] addresses this by transferring poses from a single instance to an entire category via intra-class shape correspondence. Although the grasp poses generated by these visual-based grasping methods satisfy the force closure [17] principle, they do not consider the need for different grasp strategies for the same object in different tasks. This limitation restricts the applicability of these methods in various scenarios. Our GraspControl enables effective human-robot interaction through language instructions and aligns them with vision models, allowing the robot to grasp the corresponding object parts according to the task.

Visual-language Grasping: This approach aims to ensure that the grasp generated is compatible with the requirements of the subsequent manipulation tasks. To achieve task-oriented grasp generation, methods such as [9]–[11], [15], employ approaches for constructing semantic knowledge bases. Murali et al. [11] propose GCNGrasp, which uses the semantic knowledge of objects and tasks encoded in a knowledge graph to guide task-oriented grasping. GraspCLIP [12] combines a language model and RGB images to generate grasping boxes at specified parts of the object. GraspGPT [13] and FoundationGrasp [19] leverage the open-ended knowledge from foundation models to learn generalizable task-oriented grasp skills. 3DAPNet [14] detects affordance regions from the shapes of objects and uses a language-guided diffusion model to generate 6-DoF poses for robotic grasping tasks. Sim-Grasp [20] proposes Sim-GraspNet, which generates grasp poses from point clouds and incorporates a language model to improve object manipulation performance in cluttered scenes.

However, direct concatenation of VLM's low-level pixel features and LLM's high-level language features may lead to mismatches, making it difficult to effectively align grasp language instructions with the spatial information of objects. In this paper, we expand upon language instructions and employ structurally simple sketches to better align visual and language features, thereby enhancing the model's ability to comprehend grasping tasks, such as performing grasps in specified orientations at particular parts of an object.

III. METHOD

Our goal is to align task-level semantic features with object-level visual features to guide the grasp synthesis. However, directly combining these two types of information is challenging, as achieving successful grasps requires a non-trivial alignment between language and image prompts [21]. To address this issue, we delve into the intricate realm of fine-grained grasp synthesis by utilizing Text-Sketch instructions as an interface for diverse grasp synthesis. Specifically, we first collect a multi-modal grasp dataset. Based on the dataset, we develop a grasp-oriented vision-language model to generate grasp sketches. We then employ these grasp sketches to guide the reconstruction of 3D objects and their corresponding 3D grasps. Finally, from the generated grasps, we extract the 6D grasp poses intended for the robot's gripper to execute. Our overall architecture is shown in Fig. 2. We will discuss each component in detail.

A. Dataset for Visual-Language Grasp

To successfully implement Text-Sketch guided grasp, our initial step involves establishing a dataset including text and sketch. This dataset is based on the existing ShapeNet [22], and it is enriched by integrating textual descriptions of the grasping process, sketches that delineate the structural outlines of the targeted objects, and grasp sketches with the grasp on these objects.

Collection of object sketches. We choose object CAD models from ShapeNet, which include everyday household items like mugs and bowls, as exemplified in the first column of Tab. I. The subsequent row in Tab. I delineates the quantity of instances for each respective category. We augment the data of the object CAD models through the application of shape deformations, thereby augmenting the number of object instances. Subsequently, we render images from various viewpoints using the object CAD models and utilize the Canny edge detector [18] to precisely extract edges from these rendered images. This process generates corresponding 2D object sketches.

Text description for grasp. We design 5 grasp templates of text prompt, each containing object, grasping task, grasping position and orientation to facilitate the grasp synthesis according to human intentions. As shown in Tab. II, these key

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

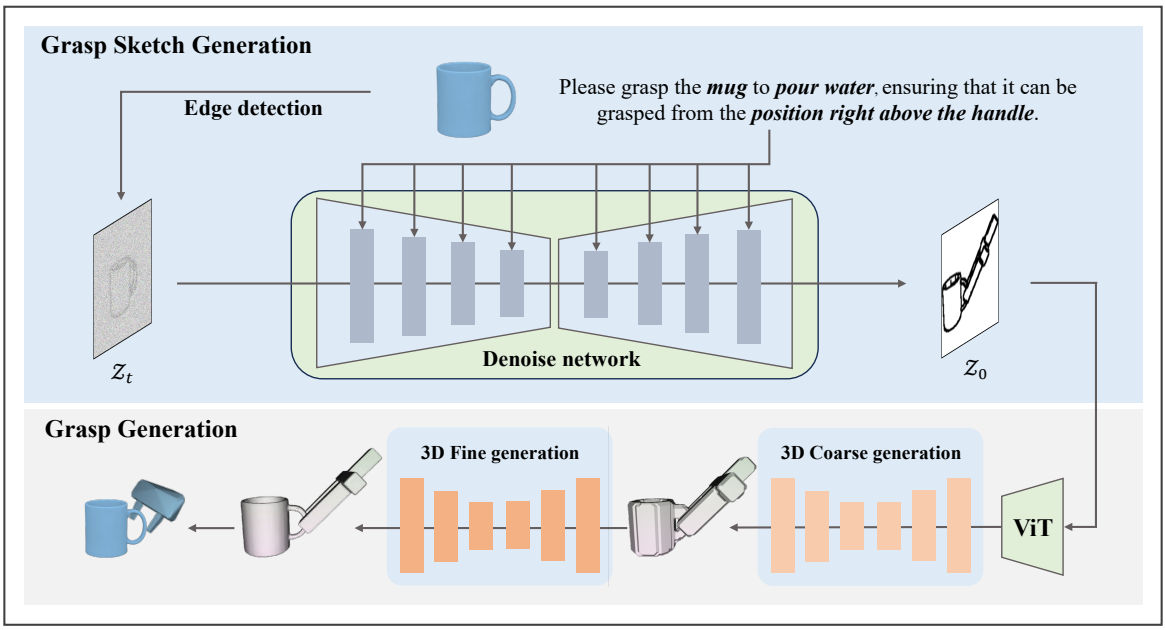


Fig. 2. The overall architecture of our GraspControl framework. In the **Grasp Sketch Generation**: We process the image using **Edge Detection** [18] to obtain an object sketch. The object sketch and grasp text prompt guide a fine-tuned 2D diffusion model to generate the corresponding grasp sketch. In the **Grasp Generation**: The grasp sketch guides a Coarse-to-Fine two-stage 3D diffusion model evolving from a rough model shell to a fully refined grasp cad model. The grasp pose extracted from the grasp cad model is applied to robotic grasping.

TABLE I
THE DATASET STATISTICS INCLUDE OBJECT CATEGORIES, THE NUMBER OF OBJECTS, AND THE NUMBER OF DEFORMATIONS.

Obj	Num	Scale	Task	Position	Orientation
Mug	160	×5	pour water, wrap-grasp, grasp	wall, handle, body	left, upper-left, upper, front, back, right, upper-right
Bottle	460	×2	wrap-grasp, open	body, opening	left, upper-left, upper, right, upper-right
Bowl	130	×10	pour water, grasp, handing	wall	left, front-left, front, front-right, back, right, back-right
Knife	330	×2	cutting, handing, stab	handle, body	left, upper-left, lower-left, right, upper-right, lower-right

TABLE II
FIVE LANGUAGE INSTRUCTION TEMPLATES. THE ORDER OF $< position >$ AND $< orientation >$ CAN BE INTERCHANGED IN LINE 5.

Typec	Template
1	Ensure that the $< object >$ is grasped from the $< position >$ in the $< orientation >$ to perform the $< task >$.
2	To carry out the $< task >$, please grasp the $< object >$ from the $< position >$ at the specified $< orientation >$.
3	For the $< task >$, ensure the $< object >$ can be picked up from the $< position >$ within that $< orientation >$.
4	To execute the $< task >$, secure the $< object >$ by grasping it from the $< position >$ in the given $< orientation >$.
5	Please grasp the $< object >$ to $< task >$, ensuring that it can be grasped from the $< position >$ $< orientation >$.

words are respectively represented as $< object >$, $< task >$, $< position >$ and $< orientation >$. Here is an example based on the 5th template: 'Please grasp the *mug* to *pour water*, ensuring that it can be grasped from the position *right above the handle*'. The grasping task information is derived from prior work [23], [24], the position is defined according to the object's geometric structure, and the orientation is determined by the grasping viewpoint. The text prompts, enriched with specific grasp positions and orientations, combined with sketches outlining the contours of the objects, will collaboratively steer the creation of grasp sketches, embedding the grasp configurations applied to the respective objects.

Grasp sketches. We annotate grasps using the IsaacGym [25] simulation platform, ensuring that each instance contains 1,000 successful grasps $(R, T, width)$. Where $R \in \mathbb{R}^{3\times3}$ denotes the grasp rotation matrix, $T \in \mathbb{R}^3$ denotes the grasp position, and $width$ denotes the grasp width. To establish a clear spatial relationship between an object and its intended grasp, we design a gripper CAD model and seamlessly integrated it with the object CAD model, producing a grasp CAD model. Using the grasp CAD model, we can obtain a grasp sketch similar to an object sketch,thereby facilitating a visual understanding of the grasp. These generated grasp sketches will play a role as grasp prompts to guide the 3D reconstruction of objects and hand grasps. Overall, the dataset covers 3680 instances across four object categories, including grasp text instructions, object sketch, grasp sketches, and 3D grasp CAD models. In addition, we collected four

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

unseen categories from ShapeNet with augmentation: Pen (378), Phone (500), Remote (500), and Headphone (360), which are used for simulation testing of cross-category and zero-shot generalization.

B. Grasp Sketch Generation

The purpose of this section is to develop a model that is capable of generating 2D grasping sketches trained on the aforementioned dataset. These grasping sketches serve as a guide for the subsequent generation of 3D grasp akin to the process of image search based on sketches. Since aligning abstract language instructions with complex visual information poses challenges, we enrich the grasp descriptions by incorporating the position and orientation of the grasp, and offer a more nuanced and fine-grained representation of the object through 2D sketches. Consequently, we introduce a diffusion model [26], which utilizes both grasp text prompts and object sketches to generate 2D grasp sketches.

During the training process, given an input image z_0, the diffusion model gradually adds noise to the image, generating noisy images z_t, where t represents the number of noise addition steps. Given a set of conditions, grasp text prompts c_T, and object sketches c_I, the diffusion model learns a denoising network ϵ_θ to predict the noise added to z_t. c_T is performed using the cross-attention, while c_I is first concatenated with z_t and then performed using the self-attention. We minimize the loss function:

$$\mathcal{L}_{z_0} = \mathbb{E}_{z,c_I,c_T,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c_I, c_T)\|^2\right] \quad (1)$$

Although the training process enables the model to learn conditional image generation based on grasp text prompts and object sketches, it does not always guarantee that the generated results are accurately focused on grasp-relevant regions or clearly represent the geometric structure of the object. To address this limitation, inspired by [27], we introduce two attention-guided loss functions during inference: the text attention loss $\mathcal{L}_T$ and the image attention loss $\mathcal{L}_I$. These losses are applied to the cross-attention and self-attention maps within the diffusion model, respectively, to enhance the model's attention to the semantic and structural task-relevant regions during the generation process. Specifically, we define a set of attention supervision loss functions based on the key characters extracted from the grasp text prompts and the corner points along the object sketch contours, which are semantically and structurally aligned with the grasping task. During the denoising process, we dynamically refine z_t by maximizing the attention weights associated with these supervision targets in their respective attention maps. This process effectively guides the model to focus on local graspable regions and object contours without modifying the parameters of the diffusion model itself.

Text Attention Loss: We consider key characters in the grasp text prompt as a set $S = \{s_1, \ldots, s_i\}$, where key character $s \in S$ describe the object, location, task or orientation as "<>" is shown in Tab. II. To assign high attention value to s to refine the grasp positions and orientations in the generated grasp sketch, we extract the 16 × 16 cross attention layers from the diffusion model for Grasp Sketch Generation and obtain the cross-attention maps. Based on the index value of s in the grasp text prompt, we query the corresponding map $A_T^{(s)} \in \mathbb{R}^{16\times16}$ and attempt to maximize it. We perform the same operation for each key character in S and then sum them to obtain the text attention loss $\mathcal{L}_T$:

$$\mathcal{L}_T = \sum_{s\in S} \max(1 - A_T^{(s)}) \quad (2)$$

Image Attention Loss: We consider corner points that represent the sharp turns or significant curvature changes along the object contour in the object sketch as a set $H = \{h_1, \ldots, h_j\}$. Similar to the $\mathcal{L}_T$, we query the corresponding self-attention map $A_I^{(h)} \in \mathbb{R}^{16\times16}$ and attempt to maximize it. We perform the same operation for each corner point in H and then sum them to obtain the image attention loss $\mathcal{L}_I$:

$$\mathcal{L}_I = \sum_{h\in H} \max(1 - A_I^{(h)}) \quad (3)$$

We set the weight parameter λ to derive the multi-modal attention loss $\mathcal{L}_{multi}$:

$$\mathcal{L}_{multi} = \lambda \cdot \mathcal{L}_T + (1 - \lambda) \cdot \mathcal{L}_I \quad (4)$$

Based on $\mathcal{L}_{multi}$, we update z_t inversely:

$$z_t' \leftarrow z_t - \alpha_t \cdot \nabla_{z_t} \mathcal{L}_{multi} \quad (5)$$

Where α_t is the update factor, The denoising process proceeds through discrete steps $t = T, T-1, \ldots, 0$ with $T = 50$. To ensure stability and control during inference, we apply the refinement only at selected steps $t = 0, 10, 20$. At each selected denoising step, we update z_t using the $\mathcal{L}_{multi}$ and then perform another forward pass with the refined latent variable z_t' to obtain z_{t-1}. This gradual adjustment prevents deviation from the learned data distribution while promoting sharper object contours and more semantically accurate 2D grasp sketches.

C. Grasp Generation

Our goal is to generate 3D models guided by 2D grasp sketches which intuitively express the relativity between the gripper and the object. We employ a signed distance function (SDF) for the 3D representation. Due to the cubic complexity of memory storage and computational costs, generating high-resolution and full-grid discrete SDF is challenging. As a result, we introduce a Coarse-to-Fine generation framework of a 3D diffusion model [28] which is shown in Fig. 2.

In the Coarse stage, under the guidance of the grasp sketch, we generate a 64^3 voxel grid to approximate the shell of the 3D model. Specifically, we first create a surface-occupancy function o in 64^3 resolution:

$$o(g) = \begin{cases} 1, & (f(g) \le |\delta|) \\ 0, & (f(g) > |\delta|) \end{cases} \quad (6)$$

Where g represents a point in the 3D grid, δ is the distance threshold, and $f(\cdot)$ is the signed distance function.

We convert the 3D model's SDF into a 64^3 voxel grid x_0 through surface-occupancy function o. Then, the 3D diffusion model gradually adds noise to the x_0, generating x_t. Given

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

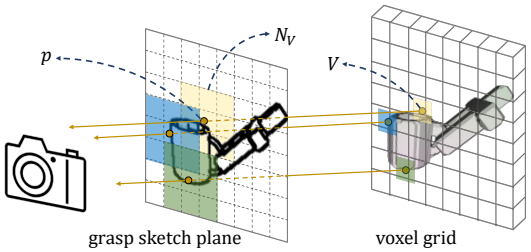


Fig. 3. The projection correspondence between the voxel grid and the grasp sketch plane. Where p is a projected coordinate, N_V is the grasp sketch patch feature at p (in yellow color), and V is the voxel grid feature.

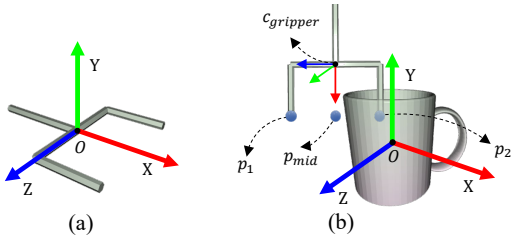


Fig. 4. Grasp pose representation and annotations on object. (a) two-finger gripper model with freely movable finger joints; (b) grasp representation, where $c_{gripper}$ is the centroid of the gripper model, p_1 and p_2 are two points at the end of the gripper, p_{mid} is the midpoint between p_1 and p_2.

a grasp sketch c_s, the 3D diffusion model learns a denoising network u_θ to predict the noise added to the x_t. To align the grasp sketch c_s with 3D voxel grid, we first project the voxel grid feature V onto the grasp sketch plane to obtain the projected coordinate p, as shown in Fig. 3. Then, we select neighboring grasp sketch patche feature N_V to interact with V. We use cross-attention to model feature interaction between the voxel feature f_V at V and the grasp sketch feature f_I in N_V:

$$Q = f_V W^Q, K = f_I W^K, V = f_I W^V \tag{7}$$

Finally, we minimize the loss function:

$$\mathcal{L}_{x_0} = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1),t}\left[\left\|u_\theta\left(x_t, c_s, t\right) - x_0\right\|_2^2\right] \tag{8}$$

In the Fine stage, we use a similar denoising network as in the Coarse stage to generate fine-grained discrete SDF values inside the 64^3 voxel grid.

Based on the 3D generative framework, we use the grasp sketch as the condition to separately supervise the training of the object model and the grasp model. We focus on extracting keypoints on the two-finger gripper's mesh to compute the grasp candidate shown in Fig. 4(a). We capture the local geometric features of the object model and the grasp model through FPFH [29], and further use RANSAC [30] and ICP [31] for registration to obtain the gripper model. Fig. 4(b) illustrates that, we compute the translation T using its centroid $c_{gripper}$ and select the centroids p_1 and p_2 of the two subsets farthest from $c_{gripper}$ as grasp points. The $width$ corresponds to the distance between p_1 and p_2. The rotation matrix R is constructed as an orthonormal basis derived from the gripper point cloud, where the grasping axis is defined by the vector from p_1 to p_2 and the approach direction by the vector from the centroid $c_{gripper}$ to p_{mid}.

IV. Experiments

A. Baseline and Metric

We compare the performance of GraspControl with other grasp generation methods. Among them, Contact-GN [6], TransGrasp, and GraspLDM [32] are representative visual-based grasping methods, while GraspGPT [13], 3DAPNet [14], and Sim-Grasp [20] are representative visual–language grasping methods, all of which employ grippers and object samples similar to ours. To ensure fairness, these methods generate 50 grasp candidates and select the final grasp pose as the one with the highest score.

We use the grasp success rate as the evaluation metric, defined as the ratio of successfully grasped samples to the total number of test samples. A grasp is deemed successful if the object remains stably held for at least five seconds after being grasped. To further compare task-level grasp quality with visual–language methods, we introduce task-level evaluation metrics. In simulation experiments, following previous work [12], we adopt a Task-Match metric, where a predicted grasp g is regarded as correct if two conditions hold: (1) the angle difference between g and a ground-truth grasp pose $\hat{g}$ is less than $30°$, and (2) the Jaccard index $J(g,\hat{g}) = \frac{|g\cap\hat{g}|}{|g\cup\hat{g}|}$ exceeds 0.25. In real-robot experiments, we further measure a Task-Success metric, defined as the proportion of grasps that remain stably held and simultaneously accomplish the language-specified manipulation.

B. Implementation Details

All models are implemented in PyTorch and trained on a server equipped with an AMD EPYC 7543 processor and a 48 GB NVIDIA A40 GPU. In the Grasp Sketch Generation stage, the resolution of the object sketches is set to 256 × 256, with a batch size of 64, a learning rate of 0.0001, and 100 training epochs. In the Grasp Generation stage, following [28], the Coarse stage is trained with a batch size of 8, a learning rate of 0.0002, and 100 training epochs. The Fine stage is trained with a batch size of 16, a learning rate of 0.0001, and 200 training epochs. The number of denoising steps is set to 50.

C. Simulation Experiment

We conducted simulation experiments in IsaacGym using a calibrated Franka Panda manipulator. For the seen categories, 90% of the collected data were used for training and 10% for testing. To evaluate zero-shot generalization, we further assessed grasping performance on the unseen categories. As shown in Fig. 5, GraspControl is able to reliably grasp various objects. The quantitative results are shown in Tab. III (where Rem. and Head. denote Remote and Headphone, respectively). GraspControl achieves an average grasp success rate of 83.87%, outperforming several advanced methods, including visual–language methods such as 3DAPNet and Sim-Grasp as well as visual-based methods such as Contact-GN

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE III
THE GRASP PERFORMANCE OF GRASPCONTROL AND OTHER ADVANCED METHODS IN A SIMULATION ENVIRONMENT (%).

Method	Seen Categories				Unseen Categories				Average	Task-Match
	Mug	Bottle	Bowl	Knife	Pen	Phone	Rem.	Head.		
Contact-GN [6]	78.19	82.39	73.85	83.27	70.90	82.20	79.20	65.83	76.98	-
TransGrasp [7]	87.36	89.45	78.94	86.15	72.75	79.00	76.80	59.72	78.77	-
GraspLDM [32]	80.83	84.57	82.02	87.23	74.34	82.40	80.40	61.11	79.11	-
GraspGPT [13]	82.36	83.15	81.25	83.09	73.81	84.60	81.00	**67.22**	79.56	71.29
3DAPNet [14]	87.92	90.33	81.06	88.13	68.25	78.60	80.60	62.50	79.67	76.85
Sim-Grasp [20]	89.72	90.76	83.37	90.83	**76.19**	87.00	82.40	63.06	82.98	67.38
GraspControl	**91.94**	**92.82**	**88.75**	**92.26**	73.54	**87.80**	**83.60**	60.28	**83.87**	**80.08**

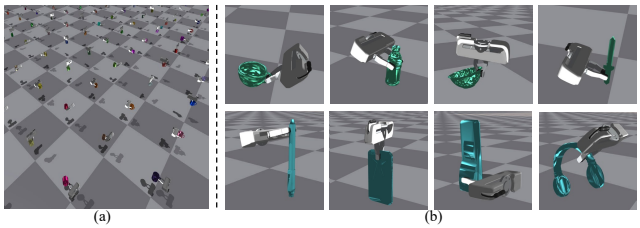


Fig. 5. Simulation platform (a) Part of test mug in simulation platform. (b) Qualitative results of grasp results.

TABLE IV
THE GRASP PERFORMANCE USING DIFFERENT INPUT DATA AND WHETHER OR NOT $\mathcal{L}_{multi}$ IS UTILIZED (%).

Method	Mug	Bottle	Bowl	Knife
RGB	43.19	49.67	37.50	46.22
RGB w/ $\mathcal{L}_{multi}$	46.11	51.41	38.75	51.26
Sketch	89.17	91.41	82.98	89.39
Sketch w/ $\mathcal{L}_{multi}$	**91.94**	**92.82**	**88.75**	**92.26**

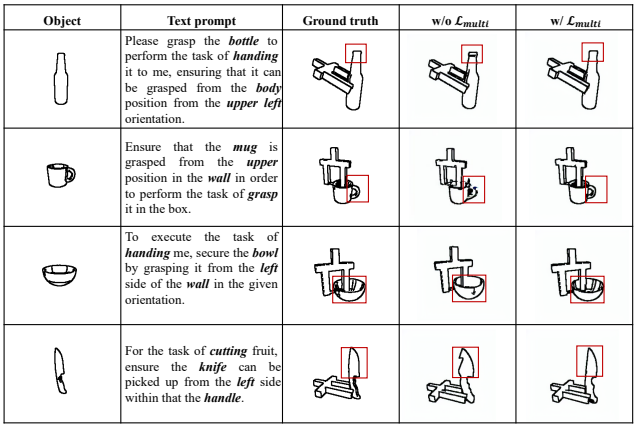


Fig. 6. Grasp sketches are generated for four categories of object sketches and grasp text prompts, both with and without $\mathcal{L}_{multi}$, and are compared to the ground truth.

and GraspLDM. Its Task-Match reaches 80.08%, exceeding 3DAPNet and Sim-Grasp by 3.23% and 12.7%, respectively. GraspControl maintains leading performance on all seen categories and demonstrates strong zero-shot generalization to the four unseen objects. It is worth emphasizing that GraspControl exhibits particularly strong performance in the bowl category, where even slight deviations in the grasp pose often lead to collisions between the gripper and the bowl wall, resulting in grasp failures. The success rate on the unseen category headphone is slightly lower, likely due to its more intricate geometric structure. The superior performance of GraspControl is mainly due to the use of generated sketches as grasping prompts, which effectively capture the object–gripper relationship and improve visual–language alignment. Moreover, the 3D reconstruction module provides detailed shape and spatial information, thereby enhancing the robustness and controllability of the generated grasps.

D. Ablation Study

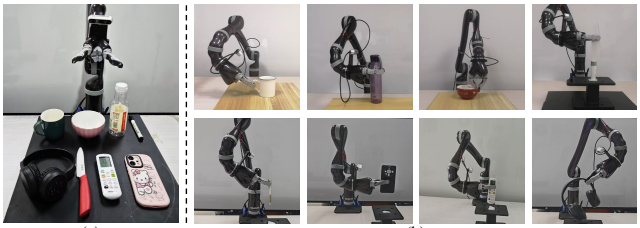
We conducted ablation study in a simulation environment to evaluate each component of GraspControl. To validate the effectiveness of sketches in GraspControl, we use two different forms of input, RGB images and sketches, to guide the generation of grasp poses and compare the grasp success rates. Additionally, we verify the effectiveness of the multi-modal attention loss $\mathcal{L}_{multi}$. Specifically, we divided the experiments into the following four groups: *RGB*, *RGB w/* $\mathcal{L}_{multi}$, *Sketch*, and *Sketch w/* $\mathcal{L}_{multi}$, and the quantitative results are shown in Tab. IV. When leveraging *RGB*, the success rate is generally low, usually below 50%. Although *RGB w/* $\mathcal{L}_{multi}$ has significantly improved the success rate, it still fails to meet the expected requirements. This is due to the redundant color and detail information in RGB images, which makes it challenging to align well with the language describing the task. In contrast, *Sketch* achieves a higher success rate. This is because sketches emphasize the contours and structural features of objects. Aligning these sketches with grasp text prompts allows for a more intuitive expression of the grasp position and approach orientation. When leveraging *Sketch w/* $\mathcal{L}_{multi}$, we find that the grasp success rate has improved. Our multi-modal attention loss not only preserves the structural features of objects from the initial sketches but also enhances the focus on spatial location information in grasp text prompts, thereby improving the consistency between visual and language features.

Fig. 6 shows the qualitative results of *Sketch* and *Sketch w/* $\mathcal{L}_{multi}$ on the test dataset for each category. It can be observed that after introducing the multi-modal attention loss,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

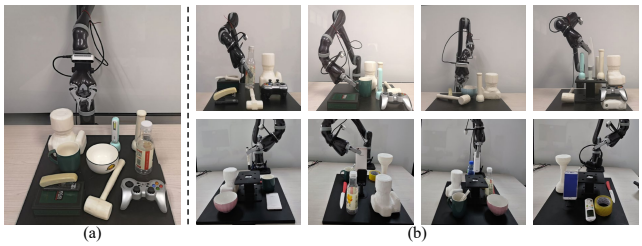
TABLE V
THE GRASP PERFORMANCE OF GRASPCONTROL AND OTHER ADVANCED METHODS IN SINGLE SCENARIOS.

Method	Seen Categories				Unseen Categories				Average	Task-Success
	Mug	Bottle	Bowl	Knife	Pen	Phone	Rem.	Head.		
Contact-GN [6]	17/25	19/25	16/25	19/25	15/25	18/25	19/25	13/25	68.00%	-
TransGrasp [7]	19/25	21/25	19/25	20/25	17/25	20/25	**20/25**	11/25	73.50%	-
GraspLDM [32]	18/25	19/25	15/25	20/25	17/25	17/25	18/25	14/25	69.00%	-
GraspGPT [13]	19/25	20/25	19/25	20/25	16/25	19/25	**20/25**	12/25	72.50%	62.07%
3DAPNet [14]	21/25	21/25	20/25	22/25	19/25	17/25	17/25	10/25	73.50%	66.67%
Sim-Grasp [20]	22/25	23/25	20/25	23/25	**21/25**	20/25	19/25	**15/25**	81.50%	60.12%
GraspControl	**24/25**	**24/25**	**22/25**	**24/25**	20/25	**21/25**	18/25	11/25	**82.00%**	**79.88%**



(a) (b)

Fig. 7. Qualitative results in single scenarios. (a) Experiment setting. (b) Grasps.



(a) (b)

Fig. 8. Qualitative results in complex scenarios. (a) Experiment setting. (b) Grasps.

the contours of the sketches generated by grasp text prompts are closer to the ground truth.

E. Real Robot Experiment

To verify the practical application of our method, we conduct real-world robot grasp experiments. The experiments are performed using a Kinova Jaco robotic arm with an Intel RealSense D435 camera. To ensure smooth and feasible motion execution, we adopt an optimization-based trajectory planner. We evaluate our method under two scenarios: (1) a single-object environment, where isolated objects are placed without external interference; (2) a complex environment, where obstacles are positioned around the target object to introduce physical constraints.

1) Single-object Environment: We evaluate five objects per category, placing each in five random poses and performing one grasp attempt to record success or failure. Similar to the simulation experiments, we compared GraspControl with the six advanced methods mentioned above. Fig. 7 shows the grasping demonstration and Tab. V shows the quantitative results (where Rem. and Head. denote Remote and Headphone, respectively). Among them, GraspControl delivers the best overall performance. Its average grasp success rate is 8.5% higher than that of the visual-based method TransGrasp and 0.5% higher than that of the visual–language method Sim-Grasp. Its Task-Success reaches 79.88%, representing a 13.18% improvement over 3DAPNet, demonstrating stronger task-execution capability. For the seen categories, GraspControl records only one failed grasp on objects such as Mug, Bottle, and Knife. GraspControl also exhibits strong generalization, with success rates remaining above 18/25 for Pen, Phone, and Remote. The total inference time of our method is 5.29s, including 1.15s for grasp sketch generation, 3.88s for grasp model generation, and an average of 0.26s for grasp generation. This speed is acceptable for many household or service robot manipulation tasks (such as tidying or delivering objects). These results confirm GraspControl's cross-category generalization ability in real-world environments.

TABLE VI
THE GRASP PERFORMANCE OF GRASPCONTROL AND OTHER ADVANCED METHODS IN COMPLEX SCENARIOS.

Method	Seen Categories				Unseen Categories			
	Mug	Bottle	Bowl	Knife	Pen	Phone	Rem.	Head.
GraspGPT [13]	17/25	18/25	16/25	17/25	11/25	15/25	12/25	10/25
3DAPNet [14]	16/25	19/25	15/25	19/25	**12/25**	13/25	11/25	10/25
Sim-Grasp [20]	20/25	22/25	19/25	20/25	**12/25**	15/25	**16/25**	**12/25**
GraspControl	**23/25**	**24/25**	**21/25**	**24/25**	10/25	**17/25**	14/25	9/25

2) Complex Environment: To demonstrate that our method can generate grasps at specific positions on objects in specific orientations based on grasp text prompts, we randomly placed obstacles around the target objects. As shown in Fig. 8, we conduct 25 experiments for each category.

As shown in Tab. VI, in the complex environments, the performance of other methods decreases compared to the single-object environments, whereas GraspControl maintains nearly unchanged performance on the seen categories. Specifically, GraspControl keeps an average success rate of over 21/25 across all seen categories and shows generalization to unseen objects. Overall, this is attributed to the rich positional and orientation information contained in the grasp text prompts and sketches within GraspControl. With these detailed prompts, GraspControl is able to generate precise grasping strategies, enabling it to solve grasping tasks in complex environments.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

3) Real-world Sketch Extraction: We obtain real-world sketches by localizing objects with semantic segmentation, applying Canny edge detection, and refining contours using Gaussian filtering and dilation (GD). To evaluate robustness with real camera inputs, we compare two settings: raw Canny edges and Canny+GD. With raw Canny edges, the method achieves 71.00% Grasp Success Rate and 63.42% Task-Success Rate, averaged across both seen and unseen categories. Applying Canny+GD improves these values to 82.00% and 79.88%, respectively. These results confirm the applicability of our method to real-world scenarios.

V. CONCLUSIONS

We present GraspControl, which utilizes texts and sketches of objects as grasp prompts to control grasp synthesis in complex scenarios. We devise a grasp-oriented vision-language model that takes grasp text prompts and object sketches as input to generate 2D grasp sketches. These sketches subsequently guide the reconstruction of 3D objects along with their corresponding grasps. To train this model, we collect a dataset with grasp language instructions and sketches of the gripper alongside target objects, and leverage a multi-modal attention mechanism to emphasize the grasping area of the object. Extensive simulated and real-world experiments demonstrate that our method surpasses state-of-the-art grasping algorithms in both grasp diversity and quality.

REFERENCES

[1] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic *et al.*, "Deep learning approaches to grasp synthesis: A review," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3994–4015, 2023.

[2] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[3] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3629–3635.

[4] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2901–2910.

[5] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.

[6] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 13 438–13 444.

[7] H. Wen, J. Yan, W. Peng, and Y. Sun, "Transgrasp: Grasp pose estimation of a category of objects by transferring grasps from only one labeled instance," in *European Conference on Computer Vision*. Springer, 2022, pp. 445–461.

[8] x. Fang, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3929–3945, 2023.

[9] P. Ardón, E. Pairet, R. P. Petrick, S. Ramamoorthy, and K. S. Lohan, "Learning grasp affordance reasoning through semantic relations," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4571–4578, 2019.

[10] L. Antanas, P. Moreno, M. Neumann, R. P. de Figueiredo, K. Kersting, J. Santos-Victor, and L. De Raedt, "Semantic and geometric reasoning for robotic grasping: a probabilistic logic approach," *Autonomous Robots*, vol. 43, pp. 1393–1418, 2019.

[11] A. Murali, W. Liu, K. Marino, S. Chernova, and A. Gupta, "Same object, different grasps: Data and semantic knowledge for task-oriented grasping," in *Conference on robot learning*. PMLR, 2021, pp. 1540–1557.

[12] C. Tang, D. Huang, L. Meng, W. Liu, and H. Zhang, "Task-oriented grasp prediction with visual-language inputs," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 4881–4888.

[13] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, "Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping," *IEEE Robotics and Automation Letters*, 2023.

[14] T. Nguyen, M. N. Vu, B. Huang, T. Van Vo, V. Truong, N. Le, T. Vo, B. Le, and A. Nguyen, "Language-conditioned affordance-pose detection in 3d point clouds," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 3071–3078.

[15] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Learning task constraints for robot grasping using graphical models," in *2010 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2010, pp. 1579–1585.

[16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[17] C. Ferrari, J. F. Canny *et al.*, "Planning optimal grasps." in *ICRA*, vol. 3, no. 4, 1992, p. 6.

[18] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[19] C. Tang, D. Huang, W. Dong, R. Xu, and H. Zhang, "Foundationgrasp: Generalizable task-oriented grasping with foundation models," *IEEE Transactions on Automation Science and Engineering*, pp. 1–1, 2025.

[20] J. Li and D. J. Cappelleri, "Sim-grasp: Learning 6-dof grasp policies for cluttered environments using a synthetic benchmark," *IEEE Robotics and Automation Letters*, 2024.

[21] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[22] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[23] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, "3d affordancenet: A benchmark for visual object affordance understanding," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1778–1787.

[24] M. Hassanin, S. Khan, and M. Tahtali, "Visual affordance and function understanding: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–35, 2021.

[25] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[26] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 392–18 402.

[27] H. Chefer, Y. Alaluf, Y. Vinker, L. Wolf, and D. Cohen-Or, "Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–10, 2023.

[28] X.-Y. Zheng, H. Pan, P.-S. Wang, X. Tong, Y. Liu, and H.-Y. Shum, "Locally attentional sdf diffusion for controllable 3d shape generation," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–13, 2023.

[29] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 3212–3217.

[30] M. A. Fischler and R. C. Bolles, "Random sample paradigm for model consensus: A apphcatlons to image fitting with analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[31] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[32] K. R. Barad, A. Orsula, A. Richard, J. Dentler, M. Olivares-Mendez, and C. Martinez, "Grasplдm: Generative 6-dof grasp synthesis using latent diffusion models," *IEEE Access*, 2024.