SeededGrasp: Language-Guided Grasping in Complex Scenes with Multiple Embodiments

Yang Xu[1,2,*] **Gurpreet Singh Mukker**[1,*] **Raymond Wang**[3,*]
Jasper Gerigk[1,2], **Maria Attarian**[1,2,4], **Igor Gilitschenski**[1,2]

[1]University of Toronto [2]Vector Institute [3]University of British Columbia [4]Google Deepmind

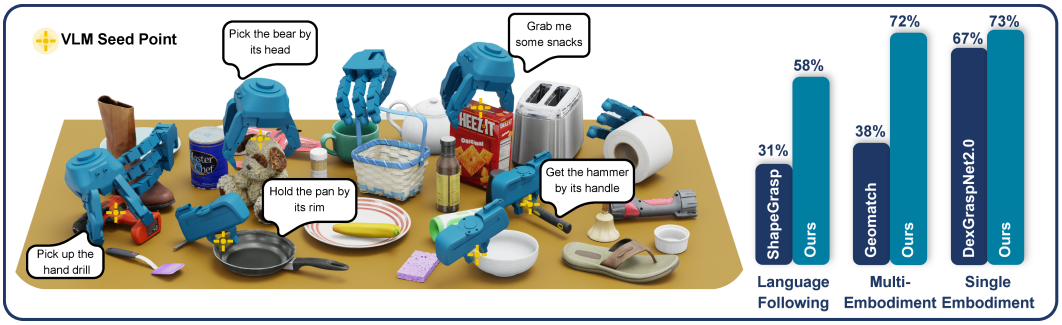


Figure 1: Example grasps and performance summary of our proposed method, SeededGrasp.

Abstract: Practical robotic grasping in complex scenes requires both 3D spatial reasoning and alignment with task-specific requirements. Vision-language models (VLMs) offer a natural way to specify these requirements using language, but existing approaches either use a VLM to predict the grasp directly with limited spatial awareness, or train the VLM together with the grasping model, which requires significantly more data and compute. These limitations impede performance and have prevented scaling to multiple embodiments in complex scenes. We address this by proposing SeededGrasp, a novel data-efficient framework that enables a VLM to predict a seed point to be used as conditioning for a subsequent lightweight grasp-generation model. Our architecture decouples high-level semantic reasoning from low-level geometric execution, enabling multi-embodiment support while bypassing the need for expensive end-to-end training. To enable training such models, we release the first multi-embodiment tabletop grasping dataset comprising over 2.5M grasps in cluttered scenes. Experimental results demonstrate that our approach outperforms existing baselines, achieving 72% success in simulation and 78% in real-world grasping experiments. See our project site for data and code: https://uoft-isl.github.io/seeded-grasp/.

Keywords: Dexterous Grasping, Natural Language, Multi-Embodiment

1 Introduction

Dexterous grasping is a fundamental requirement for general manipulation but remains a challenging task. To act effectively in the physical world, a robot must make contact with objects in a manner that is precise, stable, and task-aware. Additionally, in contact-rich, multi-object environments such as cluttered scenes, models must respect the objects' surroundings [1, 2]. Moreover, because downstream tasks impose different functional requirements, the ability to follow language instructions becomes highly advantageous [3, 4, 5]. This challenge is further compounded by the wide variety of

[*]Shared first.

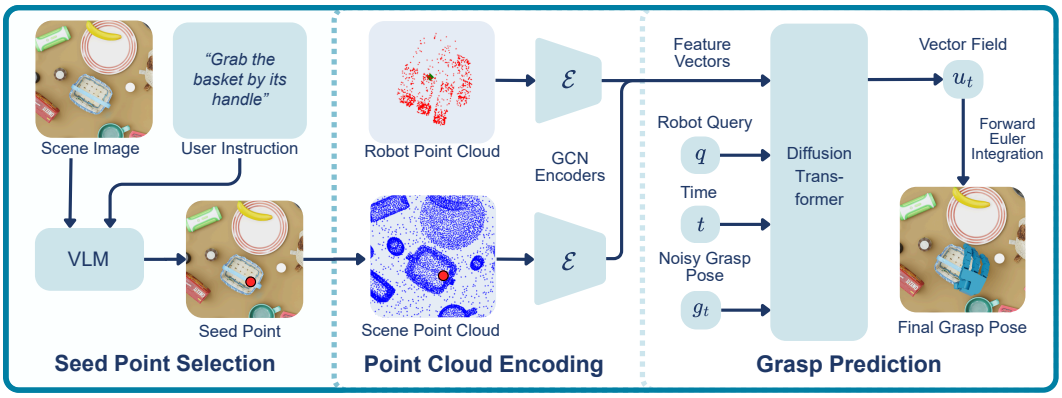


Figure 2: We use a VLM to select a seed point. The point cloud encoding and grasp prediction sections are trained in an end-to-end fashion and predict a grasp pose conditioned on the seed point. During training we use seed points derived from the dataset.

robotic end-effectors. Simple two-finger grippers, while widely used, struggle with more complex object geometries because they are limited to pinch grasps. In contrast, grippers with high dexterity, such as the Allegro Hand, can approach objects in multiple ways but are difficult to control [6, 7]. Supporting multiple embodiments offers the additional advantage of improving grasping performance, making a single multi-embodiment model more useful and data-efficient than specialized ones [8, 9, 10, 11]. Together, these considerations motivate the problem addressed in this paper: designing a system that connects high-level linguistic intent to low-level contact geometry, while remaining effective in cluttered scenes and across diverse embodiments.

Existing approaches address only parts of this problem, and important gaps remain. Some methods focus on isolated objects, which simplifies perception and contact reasoning but does not capture the complexity of real-world cluttered scenes [12, 13]. Works that study language-grounded grasping systems often require large-scale annotated datasets and expensive end-to-end training involving a VLM [5, 14], or leverage pre-trained VLMs to predict high-level representations with complex pose optimization pipelines [3, 15, 16]. These approaches do not scale well to cluttered multi-object settings with dense occlusions and/or multi-embodiment grasping where gripper geometries differ greatly and large-scale datasets cannot be collected for every gripper. In addition, existing grasping datasets do not simultaneously provide both cluttered multi-object scenes and multi-embodiment poses [2, 12, 17]. This mismatch motivates our approach and the creation of a new dataset.

In this work, we present SeededGrasp, a modular framework for language-grounded, multi-embodiment grasping in cluttered scenes. Our key idea is to use a VLM to select a task-specific point from the scene point cloud, that indicates where the grasp should be centered. This seed point allows the VLM to isolate the target object and its relevant part while giving the grasping model flexibility in how to grasp it. Conditioned on this seed point, we train a lightweight flow matching model for grasp generation [18]. To train our model, we develop a synthetic data generation pipeline and curate a new dataset containing 2.56M grasps across 610 scenes, 334 objects, and 3 grippers. We evaluate SeededGrasp both in simulation, where it outperforms all baselines, and real-world experiments, where we demonstrate its sim-to-real capabilities.

In summary, our contributions are twofold:

1) A novel approach using a VLM to zero-shot predict a seed point which conditions a lightweight flow-matching model for efficient, multi-embodiment grasp generation.
2) A new grasping dataset of 2.56M grasps across 610 cluttered scenes spanning three grippers.

2 Related Work

Analytic and simulation-based grasp synthesis. Early work on robotic grasping commonly formulates grasp generation as a search or optimization problem with analytic stability criteria [19].

Recent optimization-based approaches have substantially improved the speed and quality of dexterous grasp synthesis [20, 21, 22]. While these methods are still relatively unwieldy at inference time and generally do not work on cluttered scenes or incorporate semantic intent from language, they are very valuable for generating datasets of valid grasps.

Datasets. Progress in grasp learning has been closely tied to dataset quality. Large synthetic datasets have enabled major advances in both parallel-jaw and dexterous grasping [12, 23, 24], and more recent datasets have expanded to cluttered scenes [2, 25] and multiple embodiments [9, 17]. However, none of the existing datasets provide data for multi-embodiment grasping in cluttered scenes. In this work, we build our dataset upon MultiGripperGrasp (MGG), which itself filters grasps generated by GraspIt [19] in simulation for stability across various objects and grippers.

Grasping in cluttered scenes. A major challenge in manipulation is moving from isolated objects in mid-air to cluttered, contact-rich scenes where the model must reason about occlusion and interactions with surrounding objects. Recent learning-based methods address this by predicting grasp distributions directly from scene observations, often using point clouds or depth images as input [1]. For dexterous hands, this has led to generative models, contact-aware representations, and collision-aware formulations that better capture the complexity of multi-object scenes [2, 26, 27]. Even so, most of this work focuses on geometric feasibility in clutter and is typically developed for a fixed gripper embodiment whereas our work trains a shared model across multiple embodiments.

Multi-embodiment grasping. Another research direction studies how grasping models can generalize across different grippers and dexterous hands. Rather than training a separate policy for each end-effector, these methods condition prediction on robot geometry or kinematic structure [8, 11, 28, 29]. Some approaches use hand-agnostic intermediate representations, such as contact maps or keypoints, and then solve for specific gripper configurations [9, 10]. Others directly encode morphology and learn grasp generation end-to-end [11, 30, 31]. Collectively, these methods show that conditioning on gripper structure and multi-embodiment training can improve grasp quality. However, they are primarily focused on generating geometrically valid grasps and do not address multi-object scenes and language grounding like our work.

Language-grounded grasping. Natural language is an intuitive interface for specifying what object to grasp and how. Supporting such conditioning has motivated language-grounded methods. Several recent approaches [3, 16] use VLMs as a plug-and-play module to infer task-relevant regions, affordances, or object parts from text and subsequently rely on heuristics to propose and finetune grasps. These often lead to imprecise results. Other works build larger systems that combine VLM-based semantic grounding with generative modeling techniques like flow matching [18], training the entire pipeline end-to-end [5, 14, 32, 33]. However, these systems require large-scale, language annotated datasets and are expensive to train. Our method bridges both approaches by introducing a seed point as an efficient interface connecting a pretrained VLM with a learned flow matching grasp generation module, eliminating the need for a language-annotated dataset.

3 Dataset Generation

To train our seed point conditioned grasp prediction model, we require a dataset of valid grasps in cluttered scenes for each of the grippers. Rather than generating raw grasp poses from scratch, we build upon the MultiGripperGrasp (MGG) dataset [17], which provides millions of synthetic, mid-air grasp poses across a diverse set of grippers and objects. Adapting this dataset involves generating cluttered scenes and transforming these stable mid-air poses into valid in-scene grasps for training.

The MGG dataset was re-filtered using its original infrastructure, which includes mid-air gravity and 6-axis grasp tests [17], and several object and gripper models were refined to ensure high grasp quality. Following this filtering process, the three grippers yielding the highest volume of quality data while representing a good mix of both parallel-jaw and dexterous grasping kinematics were selected: the Franka Panda, Robotiq 3-Finger, and Allegro grippers.

Cluttered scenes were generated using a split of 284 objects for training and 50 unseen objects for evaluation from the Google Scanned Objects [34] and YCB [35] datasets. These objects encompass both simple shapes and irregular geometries. Each scene was constructed by randomly sampling one to ten objects, initializing them in mid-air with random orientations, and dropping them on a tabletop. This procedure yields cluttered scenes featuring both sparse and dense regions. In total, 504 training and 106 evaluation scenes were generated.

Grasps for each of these scenes were generated by transforming existing MGG grasps and checking if they were viable. For every object present, its associated mid-air grasps were transformed into the scene's global frame. Naturally, many of them are infeasible in the cluttered environment due to collisions with the table or other objects. Because conducting dynamic, in-scene lift tests for millions of grasps is computationally prohibitive, a set of heuristic collision checks efficiently determined feasibility. Specifically, the gripper had to approach the object from a viable angle (≥ 0.5 rad), all gripper links had to maintain clearance above the table (≥ 0.5 cm), and the gripper should not excessively intersect with objects (by enforcing a 10 mm interpenetration threshold).

For each scene, we generate a 2048-point point cloud by fusing four RGB-D images captured at cardinal directions and downsampling it using farthest point sampling with a bias away from the table plane. We capture a bird's-eye view (BEV) for seed point selection by the VLM during inference for this paper, but our approach generalizes to arbitrary viewpoints where target objects are visible.

Robot Model	Total Grasps	After Refiltering	In-Scene (Train)	In-Scene (Eval)
Franka Panda	2.76 M	1.57 M	1.61 M	469 K
Robotiq 3 Finger	2.72 M	990 K	246 K	27 K
Allegro	2.77 M	766 K	161 K	48 K

Table 1: Grasp Dataset Statistics by Robot Model. While all three grippers started with the same amount of grasps, Robotiq and Allegro grasps were viable in scene at a significantly lower frequency.

In total, our dataset consists of 2.56 million grasp poses split between training and evaluation sets for the three robot grippers (see Table 1). Notably, the dataset distribution is skewed toward the Franka Panda gripper. Its simpler kinematics allow for a higher percentage of grasp poses to pass filtering. While our pipeline allows us to balance the distribution with further targeted data generation, further generation was found to be unnecessary for the scope of this work (Section 5.2).

4 Model

As shown in Figure 2, our grasp generation pipeline consists of two stages: the first predicts the seed point, and the second predicts the grasp pose based on the seed point prediction. This approach is inspired by DexGraspNet2.0 (DGN2.0) [2], but does not require training a seed point prediction model; instead, we use an off-the-shelf VLM to predict a suitable grasping location based on a BEV scene image and user instructions (Section 4.3). This predicted seed point is then passed to a trained flow-matching model to synthesize a grasp pose for the selected gripper (Sections 4.1 and 4.2). This design enables precise natural-language control over grasp generation, incurs low training costs, and does not require generating and filtering numerous candidate poses. Furthermore, the approach is not specific to the selected VLM in this modular design, enabling natural performance scaling from future VLM improvements.

4.1 Model Architecture

Point Cloud Encoding. Both gripper and scene geometries are colourless point clouds, denoted as $p_r \in \mathbb{R}^{1024\times 3}$ and $p_c \in \mathbb{R}^{2048\times 3}$ respectively. The point coordinates are Fourier-encoded [36] and concatenated with both the local normal vector ($n \in \mathbb{R}^3$) and the curvature ($\sigma(p) \in \mathbb{R}$), which are estimated via a Principal Component Analysis of local neighborhoods to enrich surface representation. Furthermore, each point's representation in the scene point cloud also contains its distance to

the seed point. Following GeoMatch [10], Graph Convolutional Networks encode the point clouds using the sixteen nearest neighbors, yielding per-point feature vectors $f_p \in \mathbb{R}^{512}$. To extract global representations for each point cloud, per-point feature vectors are aggregated using both max and mean pooling. Additionally, to capture the local geometry surrounding the target grasp location, features from the sixteen nearest neighbors to the seed point in the scene point cloud are concatenated with the global scene features. A learnable query $q_r \in \mathbb{R}^{512}$ is introduced for each robot geometry to encode the specific gripper selection. Concatenating these supplementary vectors produces the final feature vectors $f_r \in \mathbb{R}^{3\times512}$ and $f_c \in \mathbb{R}^{18\times512}$ for the robot and scene point clouds respectively.

Grasp Pose Parameterization. Each pose is parameterized as a vector $g = (T, R, \theta)$, where $T \in \mathbb{R}^3$ is the translation, $R \in \mathrm{SO}(3)$ is the rotation, parameterized with Euler angles [37], and θ is the joint configuration of the gripper within the world frame. Because the number of actuated joints varies across the evaluated grippers, the pose representation is padded to accommodate the maximum degrees of freedom, which is sixteen joints for the Allegro hand.

Grasp Pose Generation. The distribution of stable grasp poses conditioned on robot and scene features, $p(g \mid f_r, f_c)$, is inherently complex and multi-modal. Consequently, we use flow matching to model and sample from $p(g \mid f_r, f_c)$ [18]. The flow matching model $u_\theta(t, g_t, f_r, f_c)$ is trained to predict a ground truth vector field $u_t(t, g_t, f_r, f_c)$. Given a time step $t \in [0, 1]$ and conditioning features f_r and f_c, the model transforms a noisy grasp pose sample g_t into a sample from the target data distribution. By numerically solving the corresponding Ordinary Differential Equation (ODE) (detailed in Section 4.3), an initial noise sample g_0 is iteratively transformed into a clean grasp pose g_1. The underlying network employs a standard Diffusion Transformer (DiT) architecture, which utilizes cross-attention mechanisms to inject the robot and scene conditioning signals [38].

4.2 Model Training

Data Preparation. SeededGrasp is trained on our entire synthesized grasp dataset. To train the grasp prediction model without querying a VLM each time, we sample one of the eight points on the target object closest to the gripper's center of palm position to act as a ground-truth seed point. We mitigate overfitting to simpler object and gripper geometries, which naturally pass the heuristic filtering at higher rates due to fewer collisions between them, by sampling grasp poses uniformly across grippers, scenes, and objects. To promote generalization, random Z-axis rotations, XY-plane translations, and Gaussian noise are applied to the point clouds and grasp poses where applicable. Furthermore, all grasp poses are min-max normalized to a range of $[-1, 1]$ with distinct bounds for the joint angles of each specific gripper to account for variations in joint limits, scales, and units, facilitating multi-embodiment learning.

Training Details. The network is optimized using a conditional flow-matching loss, $\mathcal{L}_{\text{cfm}}$, computed across the translation, rotation, and joint angle components of the predicted vector field u_θ. The total loss is formulated as a linear combination: $\mathcal{L} = \lambda_{\text{trans}}\mathcal{L}_{\text{trans}} + \lambda_{\text{rot}}\mathcal{L}_{\text{rot}} + \lambda_{\text{joints}}\mathcal{L}_{\text{joints}}$, where the $\lambda = (0.4, 0.04, 0.8)$ coefficients serve as scaling factors to balance the respective terms. Unused joint dimensions for specific grippers are masked during the computation of $\mathcal{L}_{\text{joints}}$, and a subsequent averaging operation ensures that joint losses remain balanced across the different embodiments. Instead of the standard L2 loss, an L1 loss is employed for $\mathcal{L}_{\text{cfm}}$:

$$\mathcal{L}_{\text{cfm}} = \mathbb{E}_{t\sim\beta(1.5,1.0),\ (g_t,f_r,f_c)\sim\mathcal{U}}\left[|u_\theta(t, g_t, f_r, f_c) - u_t(t, g_t, f_r, f_c)|\right]$$

This substitution improves the precision with which generated grasp poses adhere to scene geometries. Furthermore, the time step t is sampled from a Beta distribution during training to bias the loss toward later time steps ($t \to 1$), where highly accurate vector fields are critical for finalizing precise grasp poses [32]. Classifier-Free Guidance is also integrated to strengthen the model's adherence to the conditioning signals [39]. By randomly dropping out the conditioning features f_r and f_c during training, the network simultaneously learns both conditional and unconditional vector fields. During inference, these fields are extrapolated to push the generated samples toward stronger conditioning (using $w = 1.1$), thereby improving alignment with the surrounding scene geometry.

Method	Success (%) (Allegro Data Only)	
	Graspness Seeds	VLM seeds
DGN2.0	53.15	66.67
Ours	**66.22**	**72.97**

Method	Success (%)		
	Franka	Robotiq	Allegro
Geomatch	25.35	35.71	37.97
Ours	**71.38**	**72.16**	**72.07**

Table 2: Left: Our method and DGN2.0 [2] trained only on Allegro data across seed points from the Graspness module of DGN2.0 and a VLM. Right: On multi-embodiment data, SeededGrasp outperforms Geomatch [10] on complex objects in cluttered scenes.

4.3 Model Inference

VLM Seed Point Prediction. During inference, a VLM is prompted to zero-shot identify a stable grasping location that aligns with the user's directive (prompt detailed in Section A.6). The predicted point in the image is subsequently projected onto the 3D scene point cloud and passed to the generation module. VLMs accomplish this task effectively without requiring fine-tuning or few-shot examples. We ablate different models in Section 5.2 and find that Gemini models perform best.

Generating Grasp Poses. The final grasp pose is generated by denoising an initial pose state conditioned on the chosen seed point, the current prediction point clouds, and the target robot query vector. To simplify the inference trajectory and minimize numerical error accumulation, the initial state g_0 is initialized as a fixed canonical grasp pose adjacent to the seed point, rather than pure noise. Forward Euler Integration with a step size of 0.2 is used to numerically integrate g_0 through the predicted vector field $u_\theta(t, g_t, f_r, f_c)$ from $t = 0$ to $t = 1$.

5 Experimental Results

Our approach supports a broader set of features than prior work, so we cannot do a single one-to-one comparison. Therefore, we evaluate our method along three distinct axes to isolate specific features. First, we show that the model is able to generate successful grasps more effectively using seed points than prior work, and that VLMs can produce better seed points than a specially trained seed point generation module (Section 5.1.1). Secondly, we show that SeededGrasp can effectively generate grasps across all embodiments (Section 5.1.2). Thirdly, we demonstrate that seed points are a more effective VLM interface for language-conditioned grasping (Section 5.1.3). We then ablate gripper encoding, VLM model, and dataset size in Section 5.2. Finally, we perform real-world experiments and demonstrate sim-to-real capabilities in Section 5.3. The flow matching model was trained for 48 hours using two A100 GPUs, and Gemini 3.1 Flash is used for seed point prediction in all evaluations.

5.1 Simulation Results

5.1.1 Seed Point Selection and Conditional Grasp Prediction: We first compare our flow matching model to DGN2.0 as both can accept the same seed points. As DGN2.0 only supports a single gripper, we train both methods only using Allegro data. SeededGrasp performs better than the baseline in single embodiment grasping by 13% (Table 2) when conditioned on seed points generated by the Graspness Score Module from DGN2.0. Further, seed points predicted by Gemini 3.1 Flash show higher success rates for both methods than the specially trained seed generation module in DGN2.0 [2] (Table 2). Unlike DGN2.0, our model does not need explicit object segmentation masks or an additional feature vector produced by its graspness module and supports multiple end-effectors.

5.1.2 Multi-Embodiment Performance: As DGN2.0 only supports a single gripper, we compare our method to Geomatch [10] for multi-embodiment performance (Table 2). Our model is able to handle the more complicated objects and cluttered scenes better than Geomatch, which was designed for mid-air single object grasping, increasing the success rate by roughly 35%.

This difference is caused by Geomatch's architecture [10] which supports simultaneous multi-embodiment training by predicting contact points and then relying on an Inverse Kinematics (IK)

Method	Obj. & Part Recog. Succ. (%)		Overall Grasp Succ. (%)
	Easy Prompt	Difficult Prompt	
ShapeGrasp w/ Obj.Seg.	57.5	51.6	31.1
GraspMAS	42.5	29.0	24.5
Ours	**89.6**	**80.6**	**58.2**

Table 3: Language-conditioned grasp quality. Center: SeededGrasp selects the correct object and its part at a higher rate. Right: It achieves a higher success rate of grasps for prompts, where the language instruction was correctly followed.

optimization to determine the joint angles. We found this to be a fragile process requiring an accurate initial guess that Geomatch's heuristic frequently failed to provide on complex geometries present in our dataset. This is exacerbated by the fusion of multi-camera point clouds yielding incomplete, hollow objects without a bottom. More details can be found in Figure 6 in the Appendix.

5.1.3 Language Conditioned Grasp Prompting Methods: We compare SeededGrasp against ShapeGrasp [3] and GraspMAS [16], two alternative zero-shot techniques that allow VLMs to predict grasps, on their ability to follow prompts and grasp objects successfully. Unlike our method, both baselines only support parallel-jaw grippers and rely entirely on BEV images of the scene. ShapeGrasp segments the object in the image, then prompts the VLM to choose one that best aligns with the user instruction. GraspMAS utilizes an agentic closed-loop process composed of a Planner, Observer, and Coder to choose an object and predict grasps.

To assess the method's ability to follow language instructions and produce good grasps, we created a test dataset of 226 pairs of scene-prompt queries. These were split between 164 *Easy* and 62 *Difficult* prompts, which described the task respectively directly or indirectly. See prompt examples in Section A.3. Predictions were manually scored for adherence to user instruction.

As shown in Table 3, SeededGrasp outperforms both baselines in correctly recognizing the target object, identifying the correct part, and producing a valid grasp. Both baselines struggle as they rely exclusively on a BEV image of the scene sent to VLM and are unable to utilize point cloud information. Our seed point interface eliminates the need for complex prompting architectures and avoids the drawbacks of relying exclusively on a VLM for grasp prediction. For more details on their specific failures, see Section A.3.1.

5.2 Ablation Studies

Gripper Encoding. The model encodes gripper information using both robot point cloud and a learnable gripper-specific query vector. As shown in Table 4, using both is important, particularly for the more kinematically complex Robotiq 3-Finger and Allegro grippers. They allow the network to better capture the specific geometric intricacies of each embodiment and generalize across grippers.

Method	Success (%)		
	Franka Panda	Robotiq 3-Finger	Allegro
SeededGrasp w/o Query Vector	72.41	69.07	59.01
SeededGrasp w/o Robot PC	**72.76**	67.53	62.16
SeededGrasp	71.38	**72.16**	**72.07**

Table 4: Removing either gripper representation decreases the success rate

VLM. Because the proposed framework is agnostic to the specific VLM, the grasp success rates of several prominent VLM families are evaluated to assess the impact of the backbone architecture. Table 5 shows that Gemini 3.1 Flash outperforms competing architectures and highlights that choosing the right VLM is important given current VLM capabilities (see Table 6 in Appendix for per gripper results). Additionally, using a higher reasoning level did not help performance.

	GPT 5.4	Qwen 3.5 Flash [40]	Gemini 3.1 Flash
Success (%)	42.61	54.42	**71.87**

Table 5: Avg. Success Rates across all three grippers (Franka, Robotiq 3-Finger, and Allegro) using various VLMs for seed point prediction. Gemini 3.1 Flash outperforms others by a large margin.

Dataset Scale. To evaluate the impact of data volume on generation quality, the model was retrained using 25%, 50%, and 75% of the training scenes. As the dataset size increases, the performance improves as well but saturates at around 71.5%, indicating only marginal improvements if the dataset size was further increased (See the Figure 10 in Appendix for plot). Reducing the number of training scenes disproportionately degrades the performance of the Robotiq 3-Finger and Allegro grippers. This disparity is likely due to the Franka Panda having a higher density of viable grasp poses per scene, making it more robust to dataset downsampling.

5.3 Real-World Evaluation

Figure 3: All objects used in real setup

For the real-world experiment, we evaluate the quality of predicted grasps on a set of real-world objects using a Delto DG-3F-B three-finger gripper. We used two stationary ZED stereo cameras to generate a point cloud of the table scene. As the model was not trained on the Delto gripper, we map the Robotiq 3-Finger gripper predictions to the Delto using constant offsets on certain joints and fixed values for the additional joints. During execution, we approach the objects from above the grasp location and do force-closure after grasping by further closing the two joints at the end of each finger. We conducted ten trials for each of the fifteen unseen objects with four random distractor objects in the scene. Relying on the Robotiq 3-Finger predictions has the drawback that the Delto fingers are not as long or thick as the Robotiq ones. Nevertheless, the model achieved an overall success rate of 78%. See Table 7 in the Appendix for the individual object success rates.

6 Limitations

In real-world deployments, grippers are mounted on robotic arms that cannot reach every predicted pose, especially when avoiding obstacles. Since our model currently predicts a grasp pose without awareness of the arm's kinematics, it may propose infeasible actions. Future work should address grasp selection and arm motion planning jointly. Additionally, the model's flexibility could be enhanced by moving beyond single-image VLM conditioning, as incorporating multi-view perspectives could improve spatial reasoning and robustness in cluttered scenes where optimal grasp regions are occluded in some views. Generalization could also be improved by transitioning from embodiment-specific learned queries to a unified representation in joint space, which would facilitate better transfer to unseen hardware across diverse morphologies. Finally, broadening the diversity of the training data is necessary, as the current over-representation of power grasps in the MGG dataset limits the model's effectiveness when handling flat objects that sit flush against a surface.

7 Conclusion

We present SeededGrasp, a framework for language-grounded, multi-embodiment grasping in cluttered scenes. SeededGrasp uses a VLM-predicted seed point to encode semantic intent combined with a flow matching model for grasp generation. Trained on our synthetic dataset, it achieves strong grasping performance in both simulation and real-world tests. This demonstrates that seed point conditioning is an effective bridge between language grounding and grasp generation.

Acknowledgments

This research was supported in part by the Vector Institute and Digital Research Alliance of Canada.

References

[1] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. Contact-GraspNet: Efficient 6-DoF grasp generation in cluttered scenes. In *ICRA*, 2021.

[2] J. Zhang, H. Liu, D. Li, X. Yu, H. Geng, Y. Ding, J. Chen, and H. Wang. DexGraspNet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes. In *CoRL*, 2024.

[3] S. Li, S. Bhagat, J. Campbell, Y. Xie, W. Kim, K. P. Sycara, and S. Stepputtis. ShapeGrasp: Zero-shot task-oriented grasping with large language models through geometric decomposition. In *IROS*, 2024.

[4] 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*, 8(11):7551–7558, 2023.

[5] J. He, D. Li, X. Yu, Z. Qi, W. Zhang, J. Chen, Z. Zhang, Z. Zhang, L. Yi, and H. Wang. DexVLG: Dexterous vision-language-grasp model at scale. In *ICCV*, 2025.

[6] J. Chen, Y. Ke, L. Peng, and H. Wang. Dexonomy: Synthesizing all dexterous grasp types in a grasp taxonomy. In *RSS*, 2025.

[7] T. Feix, J. Romero, H.-B. Schmiedmayer, A. M. Dollar, and D. Kragic. The GRASP taxonomy of human grasp types. *IEEE Transactions on Human-Machine Systems*, 46(1):66–77, 2016.

[8] X. Fei, Z. Xu, H. Fang, T. Zhang, and L. Shao. T(r,o) grasp: Efficient graph diffusion of robot-object spatial transformation for cross-embodiment dexterous grasping. arXiv, 2025.

[9] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, and S. Huang. GenDexGrasp: Generalizable dexterous grasping. In *ICRA*, 2023.

[10] M. Attarian, M. A. Asif, J. Liu, R. Hari, A. Garg, I. Gilitschenski, and J. Tompson. Geometry matching for multi-embodiment grasping. In *CoRL*, 2023.

[11] Z. Wei, Z. Xu, J. Guo, Y. Hou, C. Gao, Z. Cai, J. Luo, and L. Shao. $\mathcal{D}(\mathcal{R}, \mathcal{O})$ Grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping. In *ICRA*, 2025.

[12] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang. DexGraspNet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *ICRA*, 2023.

[13] A. Mousavian, C. Eppner, and D. Fox. 6-DOF GraspNet: Variational grasp generation for object manipulation. In *ICCV*, 2019.

[14] S. Deng, M. Yan, S. Wei, H. Ma, Y. Yang, J. Chen, Z. Zhang, T. Yang, X. Zhang, H. Cui, Z. Zhang, and H. Wang. GraspVLA: A grasping foundation model pre-trained on billion-scale synthetic action data. In *CoRL*, 2025.

[15] J. Jian, Y.-L. Wei, C. Mou, Y. Lin, X. Zhu, Y. Shen, W.-S. Zheng, and R. Hu. Zerodexgrasp: Zero-shot task-oriented dexterous grasp synthesis with prompt-based multi-stage semantic reasoning, 2025. URL https://arxiv.org/abs/2511.13327.

[16] Q. Nguyen, T. Le, H. H. Nguyen, T. Vo, T. D. Ta, B. Huang, M. N. Vu, and A. Nguyen. GraspMAS: Zero-shot language-driven grasp detection with multi-agent system. In *IROS*, 2025.

[17] L. F. Casas Murillo, N. Khargonkar, B. Prabhakaran, and Y. Xiang. MultiGripperGrasp: A dataset for robotic grasping from parallel jaw grippers to dexterous hands. In *IROS*, 2024.

[18] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. In *ICLR*, 2023.

[19] A. T. Miller and P. K. Allen. GraspIt!: A versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4):110–122, 2004.

[20] T. Liu, Z. Liu, Z. Jiao, Y. Zhu, and S.-C. Zhu. Synthesizing diverse and physically stable grasps with arbitrary hand structures using differentiable force closure estimator. *IEEE Robotics and Automation Letters*, 7(1):470–477, 2022.

[21] D. Turpin, T. Zhong, S. Zhang, G. Zhu, J. Liu, R. Singh, E. Heiden, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg. Fast-Grasp'D: Dexterous multi-finger grasp generation through differentiable simulation. In *ICRA*, 2023.

[22] J. Chen, Y. Ke, and H. Wang. BODex: Scalable and efficient robotic dexterous grasp synthesis using bilevel optimization. In *ICRA*, 2025.

[23] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In *RSS*, 2017.

[24] C. Eppner, A. Mousavian, and D. Fox. ACRONYM: A large-scale grasp dataset based on simulation. In *ICRA*, 2021.

[25] H.-S. Fang, C. Wang, M. Gou, and C. Lu. GraspNet-1Billion: A large-scale benchmark for general object grasping. In *CVPR*, 2020.

[26] J. Lundell, F. Verdoja, and V. Kyrki. DDGC: Generative deep dexterous grasping in clutter. *IEEE Robotics and Automation Letters*, 6(4):6899–6906, 2021.

[27] Y. Li, W. Wei, D. Li, P. Wang, W. Li, and J. Zhong. HGC-Net: Deep anthropomorphic hand grasping in clutter. In *ICRA*, 2022.

[28] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg. UniGrasp: Learning a unified model to grasp with multifingered robotic hands. *IEEE Robotics and Automation Letters*, 5(2):2286–2293, 2020.

[29] Z. Xu, B. Qi, S. Agrawal, and S. Song. AdaGrasp: Learning an adaptive gripper-aware grasping policy. In *ICRA*, 2021.

[30] Y. Wei, M. Attarian, and I. Gilitschenski. GeoMatch++: Morphology conditioned geometry matching for multi-embodiment grasping. In *CoRL Workshop on Learning Robot Fine and Dexterous Manipulation*, 2024.

[31] R. Freiberg, A. Qualmann, N. A. Vien, and G. Neumann. Towards a multi-embodied grasping agent. *IEEE Robotics and Automation Letters*, 2026.

[32] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, S. Jakubczak, T. Jones, L. Ke, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, L. X. Shi, J. Tanner, Q. Vuong, A. Walling, H. Wang, and U. Zhilinsky. π_0: A vision-language-action flow model for general robot control. arXiv, 2024.

[33] J. Wen, Y. Zhu, J. Li, M. Zhu, K. Wu, Z. Xu, N. Liu, R. Cheng, C. Shen, Y. Peng, F. Feng, and J. Tang. TinyVLA: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025.

[34] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3D scanned household items. In *ICRA*, 2022.

[35] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The YCB object and model set: Towards common benchmarks for manipulation research. In *ICAR*, 2015.

[36] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2022.

[37] A. J. Bose, T. Akhound-Sadegh, G. Huguet, K. Fatras, J. Rector-Brooks, C.-H. Liu, A. C. Nica, M. Korablyov, M. Bronstein, and A. Tong. SE(3)-stochastic flow matching for protein backbone generation. In *ICLR*, 2024.

[38] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

[39] J. Ho and T. Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

[40] Q. Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

A Appendix

A.1 Evaluation Setup

Evaluations for all three axes were performed in Isaac Sim with a pass/fail metric for consistency. A grasp is considered a failure if there is significant penetration with objects during initialization, or if the gripper fails to lift the object 30 cm above the table without dropping. These criteria determine the success rates for all baseline and ablation studies.

A.2 Baseline Setup Details

DexGraspNet2.0: To ensure a fair comparison, our model and DGN2.0 were trained and evaluated using models trained on Allegro Hand data, which is morphologically similar to the Leap Hand which was originally used in the baseline by the authors. While the majority of its pipeline operates on the full scene point cloud, it strictly requires explicit object masks to select the initial seed point on the object of interest.

Geomatch: For Geomatch, which supports simultaneous multi-embodiment training, the method cannot be conditioned with user-generated seed points; instead, it determines its optimal input conditioning autonomously via a top-k parameter. Furthermore, because Geomatch does not support multi-object scenes out-of-the-box, it required object masks at the input and was evaluated exclusively on single-object scenes. And only single object scenes were initialized in the simulation for evaluation as well.

ShapeGrasp & GraspMAS: Both baselines operate with parallel-hand grippers and rely entirely on Bird's Eye View (BEV) images of the scene. ShapeGrasp works only with single object images and was provided with object segmentation masks. It has two modes of operation, 2D and 3D. For our evaluations, ShapeGrasp was evaluated in its 2D mode using `gemini-3-flash-preview` (identical to the model used in our method), as its native 3D mode failed too often during testing. GraspMAS, which utilizes an agentic closed-loop workflow consisting of a Planner, Observer, and Coder, was evaluated using `gpt-4o`, as this model yielded the best results for their specific pipeline.

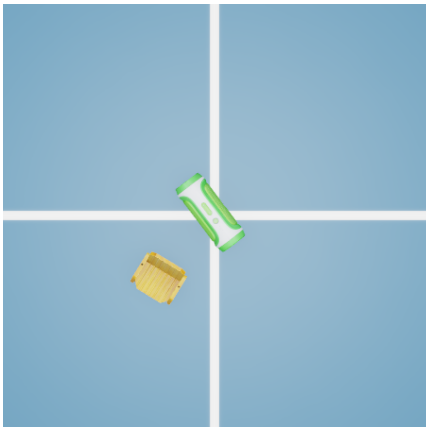
A.3 Language Conditioned Grasp Test Dataset

As described in Section 5.1, we created two sets of prompts to test the performance of the methods at identifying correct object and its part to accomplish the grasping task.

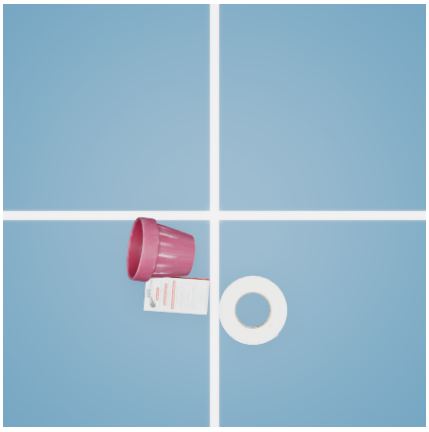
Easy Prompts: The 'easy' set of prompts were designed such that it required little to no thinking on the method's part in identifying the correct object or part of the object. The part to be grabbed was simply given in the prompt. Examples of such prompts are given below.

(a) "Grab the flip-flop by the pink and black straps"

(b) "Grasp the brown shoe by the front toe box"



(c) Grasp the green and white speaker around the center of its cylindrical body



(d) Grasp the red flower pot by the open top rim on the left

Figure 4: Examples of easy prompts for objects in scenes

Difficult Prompts: The 'difficult' set of prompts were designed such that it requires thinking on the method's part in identifying the correct object or part of the object. The prompt describes a task to be accomplished, the method finds the best part in the object to grab on to.

For example, in Figure 5a, the method should point at any point other than where the laces are. In example Figure 5b, the method should grab on to C-shaped body, in example Figure 5c, the red handles of the scissors must be grabbed.

(a) "Grab anywhere but where the shoe laces are tied"

(b) "Hold the clamp while I unscrew the handle."

(c) Grasp the scissors so you could easily cut something with them.

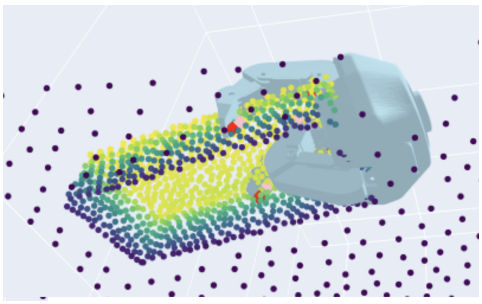
(d) Grasp the plane toy where the passenger sits

Figure 5: Examples of difficult prompts for objects in scenes

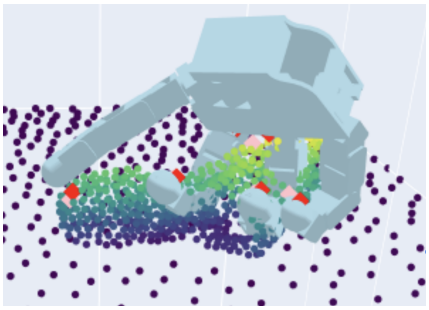
A.3.1 Failure Cases for Baselines

Geomatch: Most of the failures with this method result from object penetrations caused by poor interactions with the IK optimization algorithm in two ways. First, fusing multiple camera views results in partial point clouds with missing geometry underneath the object sitting on the table. Second, our dataset contains many complicated non-convex objects. Examples for both are provided in Figure 6

ShapeGrasp: Shapegrasp works by decomposing the objects in the input image and then selecting the part of the object that best matches with the input prompt that allows for the grasp to be successful. Figure 7 and Figure 8 highlight some of the good and bad examples of object part de-



(a) Failure due to partial scene point cloud



(b) Fail due to complicated object geometry

Figure 6: Examples of Geomatch failures

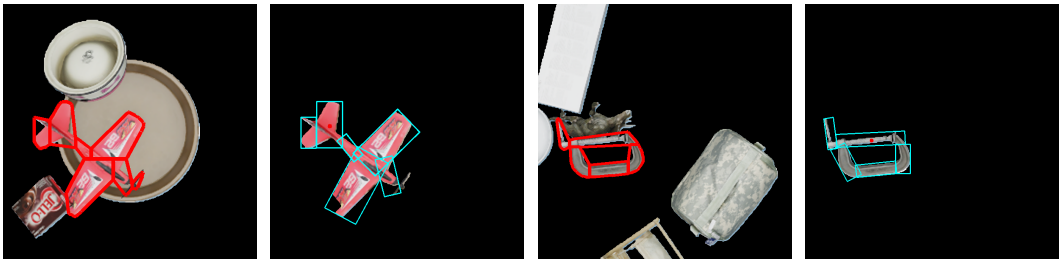


Figure 7: Good examples of part segmentation by Shapegrasp. Objects are split into reasonable number of parts which can be easily identified by their particular purpose

composition. The method performs well when the object are non-convex and can be decomposed into smaller convex parts (ex. the red toy plane and C-Clamp). For objects which are already mostly convex, the decomposition is either too aggressive like in the case of green speaker or too relaxed like in the brown shoe where the actual grasp prediction can be too imprecise.

GraspMAS: Failures for this baseline were a mix of recognizing wrong objects in the scenes or being too imprecise with the grasp location resulting in grabbing the wrong part of the object. Examples of such imprecise predictions are given in Figure 9.

Further, as mentioned in Section 5.1.3, both GraspMAS and ShapeGrasp use only a Bird's Eye View image of the scene. Identifying the correct part is only the first step; even within that localized region, the model must still evaluate numerous valid grasp locations and orientations across the full 3D geometry. This becomes even more difficult when the object geometries are highly complex in nature and are often occluded in the BEV image.

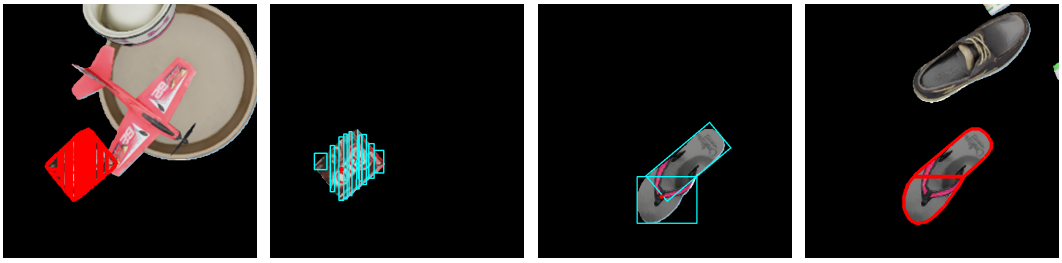
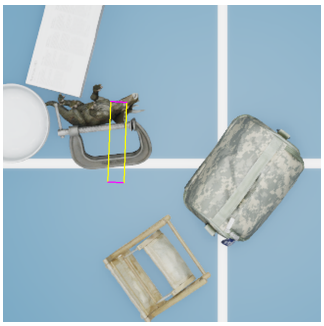
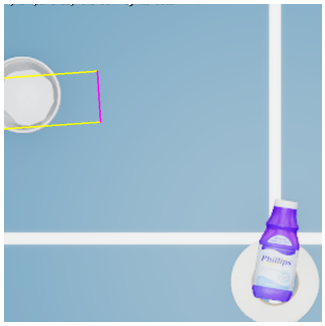


Figure 8: Bad examples of part segmentation by Shapegrasp. Objects are split too aggressively or it is too relaxed

”Pickup the tan flipflop by the heel portion of the sole”



”Hold the clamp by the tightening handle”



”Grab the white bowl by its outer rim”

Figure 9: Imprecise grasp locations from GraspMAS. First two examples show wrong part being grasped, and the third one is too imprecise with its location.

A.4 Dataset Size Ablation

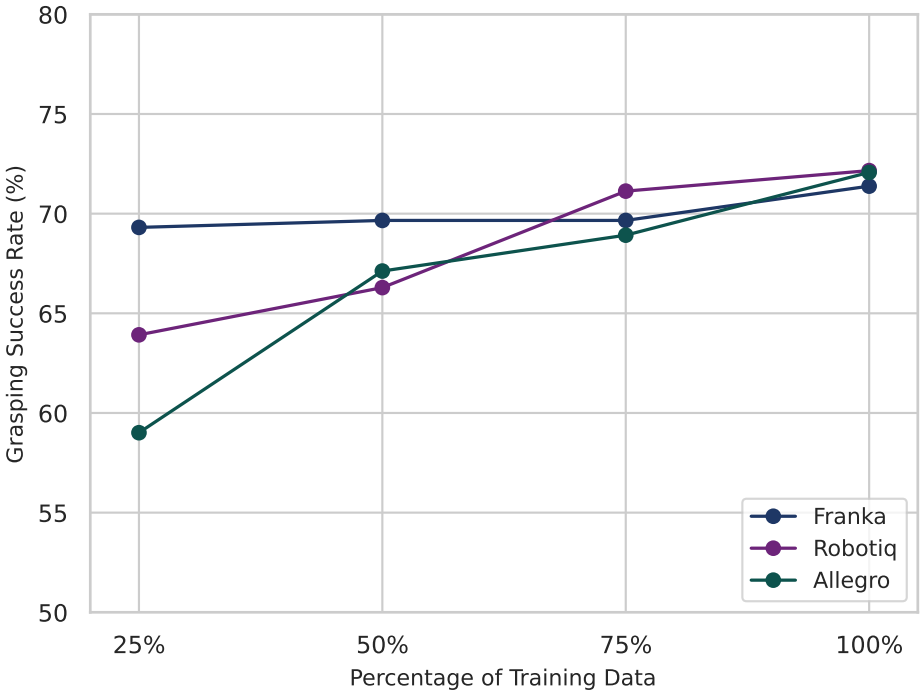


Figure 10: Success Rates with Varying Amount of Training Data

For the Franka gripper, there is little effect of decreasing the dataset size, while the Robotiq 3-Finger and the Allegro Hand do benefit from the entire dataset. However, the marginal performance improvement decreases when more of the dataset is used.

A.5 VLM Ablation

Table 6 is an extended version of Table 5 with success rates per gripper. The relative ranking of the three VLMs is the same for all grippers.

Method	Success (%)		
	Franka Panda	Robotiq 3-Finger	Allegro
GPT 5.4	37.7	41.47	48.65
Qwen 3.5 Flash [40]	46.79	51.85	64.63
Gemini 3.1 Flash	**71.38**	**72.16**	**72.07**

Table 6: Success Rates using various VLMs for seed point prediction for all three grippers. Gemini 3.1 Flash outperforms others by a large margin.

A.6 VLM Prompt

The prompt is sent as a chat-style request consisting of (i) a `role=system` message containing the system prompt (Box 1) and (ii) a single `role=user` message whose `content` is a sequence of alternating text and image parts (Box 2). For each scene i, the user message first appends a text segment of the form "Scene number i: Object description: <USER_INSTRUCTION>", and then appends the corresponding image as an `image_url` content part with a data URL of the form `<BASE64_IMAGE_i>`.

System Prompt (API message role = `system`)

You are a robotic manipulation specialist. Please help determine a stable grasp location for the specified object in the provided image of each scene. The robot gripper will attempt to grab the object at that exact location.
Express the desired grasp location as a pixel coordinate (y, x) normalized to the 0–1000 range.
Output the result as a JSON in the following format:
Example JSON result:

```
{
  "scene_number": {
    "object": "object_description",
    "coords": [y, x],
    "explanation": "brief_justification"
  }
}
```

Critical Requirements:

1. Ensure that the grasp location is free from collisions with surrounding objects.
2. Ensure that the location is on the surface of the object as it will be projected onto the 3D point cloud of the scene. The location must be on the target object, where the gripper will grasp; it cannot be in empty space for hollow objects.
3. Ensure that the gripper can encapsulate the object around the chosen grasp location. Grippers are standard sizes and cannot fully wrap around large objects. In such cases, edges or outcrops may be a better grasping location; do not attempt to grab the entire object body.
4. Ensure that your output follows the specified JSON format.

User Message Content (API message role = `user`), 3 scenes

Scene number 1:
Object description: <USER_INSTRUCTION>

```
{
    "type":"image_url",
    "image_url":
            {
```

```
                "url":"<BASE64_IMAGE_3>"
            }
```

Scene number 2:
Object description: <USER_INSTRUCTION>

```
{
    "type":"image_url",
    "image_url":
            {
                "url":"<BASE64_IMAGE_3>"
            }
```

A.7 Real World Individual Item Success Rates

The model was able to predict feasible grasps for all objects. The main cause of failures was that the gripper was too far from the object, causing it to close without actually grasping the object. This was amplified by the grasp sometimes being very angled.

Object	Success Rate
Pink shark plushy	7/10
Red plush dog	9/10
Black cardboard box	8/10
Toilet paper roll	9/10
Paper towel roll	9/10
Large rubber duck	6/10
Small rubber duck	9/10
Blue sneaker	7/10
Gray shoe	8/10
Blue towel	8/10
Bike helmet	8/10
Brown bucket	7/10
Blue mug	8/10
Green bottle	6/10
Multicolor broom	8/10

Table 7: Grasp success rate for individual real-world objects.