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

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

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

Summary

This Supplementary Material provides additional material for our paper titled *"Language-Driven 6-DoF Grasp Detection Using Negative Prompt Guidance"*. The material is organized as follows:

- Section 1 provides mathematical proof for our theory in the main paper and recalls the connection between diffusion and energy-based models.
- Section 2 provides a detailed discussion of related literature, including diffusion models in robotics and language-driven grasp detection.
- Section 3 provides additional statistics of our dataset and comparisons to others.
- Section 4 presents the implementation details of our method and other baselines.
- Section 5 provides ablation studies.
- Section 7 shows additional qualitative results of our method.
- Section 6 provides additional information about our robotic experiments.

1 Theoretical Findings

1.1 Proof of Proposition 1

Proof. We have the following derivation:

$$p\left(\mathbf{g}|\mathbf{S},\mathbf{t},\neg\tilde{\mathbf{t}}\right) = \frac{p\left(\mathbf{g},\mathbf{S},\mathbf{t},\neg\tilde{\mathbf{t}}\right)}{p\left(\mathbf{S},\mathbf{t},\neg\tilde{\mathbf{t}}\right)}$$

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

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

1.2 Connection between Diffusion and Energy-Based Models

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

Diffusion Models. Denoising diffusion probabilistic models (DDPMs) construct a forward diffusion process by gradually adding Gaussian noise to the

ground truth sample $\mathbf{x}_0$ through T timesteps. A neural network then learns to revert this noise perturbation process. Both the forward and the reverse processes are modeled as Markov chains:

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

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

In the reverse process, each step is parameterized by a Gaussian distribution with mean $\boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right)$ and covariance matrix $\tilde{\beta}_t\mathbf{I}$, where $\tilde{\beta}_t = \beta_t \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}$. Following the simplification in [11], we can keep the covariance fixed and formulate the reverse distribution as:

$$p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right) = \mathcal{N}\left(\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right), \beta_t\mathbf{I}\right). \tag{2}$$

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

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right)\right) + \sqrt{\beta_t}\mathbf{z}, \tag{3}$$

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

Energy-Based Models. Energy-Based Models (EBMs) [5, 6, 10, 21] are a family of generative models in which the data distribution is modeled by an unnormalized probability density. Given a sample $\mathbf{x} \in \mathbb{R}^D$, its probability density is defined as:

$$p_\theta\left(\mathbf{x}\right) \propto e^{-E_\theta(\mathbf{x})}, \tag{4}$$

where the energy function $E_\theta\left(\mathbf{x}\right) : \mathbb{R}^D \rightarrow \mathbb{R}$ is a learnable neural network. Langevin dynamics [6] is then used to sample from the unnormalized probability distribution to iteratively refine the generated sample $\mathbf{x}$:

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}} E_\theta\left(\mathbf{x}_{t-1}\right) + \sqrt{\lambda}\mathbf{z}, \tag{5}$$

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

The sampling procedure used by diffusion models in Equation 3 is functionally similar to the sampling procedure used by EBMs in Equation 5. In both settings, samples are iteratively refined starting from Gaussian noise, with a small amount of noise removed at each iterative step. At a timestep t, in DDPMs, samples are updated using a learned denoising network $\boldsymbol{\epsilon}\left(\mathbf{x}_t, t\right)$, while in EBMs, samples are updated via the gradient of the energy function $\nabla_{\mathbf{x}} E_\theta\left(\mathbf{x}_t\right) \propto \nabla_{\mathbf{x}} \log p_\theta\left(\mathbf{x}_t\right)$. Thus, we can view a DDPM as an implicitly parameterized EBM and apply similar composition techniques for EBMs as in [4] for DDPMs. More details about compositional DDPMs can be referred to in [15].

2 Remark on Related Works

Diffusion Models in Robotics. Recent years have witnessed diffusion models being applied to several robotic tasks. For instance, in policy learning, diffusion models have been employ for multi-task robotic manipulation [32], long-horizon skill planning [16], or cross-embodiment skill discovery [34]. Besides, the ability of diffusion models to generate realistic videos over a long horizon has enabled new applications in the context of robotics [2, 7, 12]. For example, Du *et al.* [7] proposed to learn universal planning strategy via text-to-video generation. In robot development, diffusion models have been leveraged for manipulator construction [35] or soft robot co-design [31]. Although diffusion models have also been explored for the task of grasp detection [19, 28], none of them address the task of detecting language-driven 6-DoF grasp poses in 3D cluttered scenes.

Language-Driven Grasp Detection. Language-driven grasp detection has emerged as an active research domain in recent years. Previous works have primarily focused on addressing this task using 2D images [25, 27, 29, 30, 33]. For instance, the authors in [26] presented a method that combines object grounding and task grounding to tackle the task of task-oriented grasp detection, while Xu *et al.* [33] proposed to jointly model vision, language, and action for grasping in clutter. Despite achieving promising results, these approaches are limited in their ability to handle complex 3D environments. To overcome this limitation, recent research has explored language-driven grasp detection in 3D data. In particular, Nguyen *et al.* [19] addressed the task of affordance-guided grasp detection for 3D point cloud objects, while Tang *et al.* [25] leveraged knowledge from large language models for task-oriented grasping. However, these methods are designed for single-object scenarios, limiting their applicability in cluttered settings. In contrast, our method is capable of detecting language-driven 6-DoF grasp poses in cluttered point cloud scenes.

3 Dataset Statistics

Table 1 shows our dataset statistics and comparisons to other 6-DoF grasp datasets.

Dataset	Text?	#objects	#grasps	#scenes	Cluttered?	Data type	Annotation
GraspNet-1B [9]	✗	88	∼1.2B	97K	✓	Real	Analysis
6-DoF GraspNet [18]	✗	206	∼7M	206	✗	Sim.	Sim.
ACRONYM [8]	✗	8872	∼17.7M	-	✗	Sim.	Sim.
Ours	✓	∼3M	∼200M	1M	✓	Synth.	Analysis

Table 1: Dataset statistics.

4 Implementation Details

4.1 Grasp Detection Methods for 3D Point Clouds

- Our LGrasp6D: The text embedding $\mathbf{t}$ produced by the pretrained CLIP ViT-B/32 and the negative prompt embedding $\tilde{\mathbf{t}}$ are 512-dimensional (512-D). We employ a PointNet++ [20] architecture for our scene encoder. The number of points per scene is 8192. The scene encoder extracts $n_S = 128$ scene tokens of 256-D. We employ 4 heads for the multi-head cross-attention block, with the output of 512-D. The timestep t is encoded by a sinusoidal positional encoder to obtain a 64-D vector. To speed up the training process, we freeze the scene encoder after the first 100 epochs.
- 6-DoF GraspNet: We modified the model to integrate the text embedding derived from the CLIP text encoder [23] into both the encoder and decoder of the variational autoencoder. Since our dataset does not include negative grasp poses, we refrained from employing additional refinement steps. This is also to ensure a fair comparison with other methods. The remaining architecture, hyperparameters, and training loss are inherited from the original work.
- SE(3)-DF [28]: We append the text embedding extracted by the CLIP text encoder [23] to the input of the feature encoder. As the signed distance function is not available for our 3D point clouds, we exclude the signed distance function learning objective from the framework. The remaining architecture, hyperparameters, and training loss are retained from the original work.
- 3DAPNet [19]: 3DAPNet jointly addresses the tasks of language-guided affordance detection and pose detection. To adapt this method to our problem, we remove the affordance learning objective from the original framework. The remaining architecture, hyperparameters, and training loss are inherited from the original work.

4.2 Grasp Detection Methods for Images

Methods in this section are used in our robotic experiment in Section 5.2 of our main paper. They are trained on the RGB-D images to predict rectangle grasp poses inherited from Grasp-Anything [30]. Specifically, each grasp pose is represented by $(g_x, g_y, g_w, g_h, g_\theta)$, where (g_x, g_y) is the center of the rectangle, (g_w, g_h) are the width and height of the rectangle and g_θ is the grasp angle.

- Language-supported versions of GG-CNN [17], Det-Seg-Refine [1], and GR-ConvNet [13]: We slightly modify these baselines by adding a component to fuse the input image and text prompt. Specifically, we utilize the CLIP text encoder [23] to extract the text embedding. Additionally, we employ the ALBEF architecture presented in [14] to fuse the text embedding and the visual features. The remaining training loss and architecture are inherited from the original works.

- CLIPORT [24]: The original CLIPORT framework learns a policy π, which is not directly applicable to our setting. Therefore, we modify its architecture's final layers by adding an MLP to output the rectangle grasp pose.
- CLIP-Fusion [33]: We follow the cross-attention module in CLIP-Fusion. The final MLP in the architecture is modified to output five parameters of the rectangle grasp pose.
- LGD [29]: We report results from the original paper.

5 Ablation Studies

Negative Guidance Scale. Recall that the negative guidance scale w plays an important role in controlling the strength of the negative guidance in the sampling process. We conduct an ablation study of the effect of the change in w on the grasp detection performance. Table 2 demonstrates that values of $w = 0.2$ (used in experiments in the main paper) and $w = 0.5$ yield the best results, whereas excessively small or large values of w detrimentally affect performance.

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

Table 2: Grasp detection performance with varying negative guidance scale.

Loss Function. Table 3 shows the performances when using varying ratios of $\mathcal{L}_{\text{negative}}$ (called ζ) and $\mathcal{L}_{\text{noise}}$ (which is $1-\zeta$). The results indicate that setting ζ to 0.1 or 0.2 yields strong accuracy, while either too high (0.4) or low (0.05) values significantly hurt the performance.

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

Table 3: Loss function analysis.

Backbone Variation. We conduct an ablation study on two different scene encoder backbone, i.e., PointNet++ [22] and Point Transformer [36], and two

different pretrained text encoders, i.e., CLIP ViT-B/32 [23] and BERT [3]. The number of parameters and results of all variants are shown in Table 4. We observe that in general, PointNet++ performs better than Point Transformer, and CLIP performs better than BERT. Variants using Point Transformer run significantly slower than those using PointNet++ due to the larger and more complicated architecture. Particularly, the combination of Point Transformer and CLIP obtains a competitive grasp detection performance compared to that of PointNet++ and CLIP; however, its inference time is considerably higher. This pattern is also observed when comparing CLIP and BERT text encoders. The gap in grasp detection performance between variants utilizing the CLIP ViT-B/32 text encoder and those employing BERT is substantial, highlighting CLIP's superiority in semantic language-vision understanding.

Scene Encoder	Text Encoder	CR↑	EMD↓	CFR↑	IT↓
Point Transformer [36] (23M)	BERT [3] (110M)	0.6428	0.4597	0.7583	2.0137
Point Transformer [36] (23M)	CLIP [23] (63M)	<u>0.6591</u>	<u>0.4167</u>	**0.7725**	1.9755
PointNet++ [22] (2M)	BERT [3] (110M)	0.6430	0.4225	0.7622	<u>1.5449</u>
PointNet++ [22] (2M)	CLIP [23] (63M)	**0.6649**	**0.4013**	<u>0.7706</u>	**1.4832**

Table 4: Scene encoder and text encoder backbone variation.

6 Robotic Experiments

We show 20 real-world daily objects used in robotic experiments in Figure 1. The sequences of actions when the KUKA robot grasps objects are presented in Figure 2. Figure 3 further shows the detection result of our LGrasp6D on point clouds captured by a RealSense camera mounted on the robot. The robotic experiments demonstrate that although our method is trained on a synthetic Grasp-Anything-6D dataset, it can generalize to detect grasp poses in real-world scenarios. More illustrations can be found in our Demonstration Video.

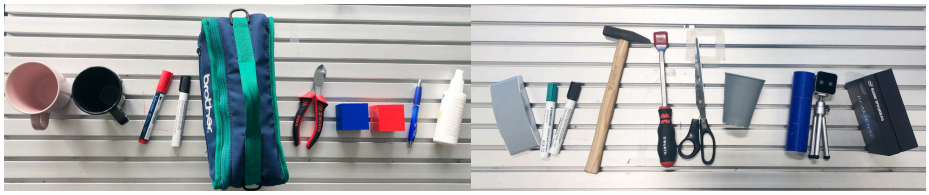

Fig. 1: Set of 20 objects used in the robotic experiments.

7 Additional Qualitative Results

Figure 4 shows more qualitative results to demonstrate the effectiveness of our method in detecting grasp poses for different objects in several point cloud scenes.

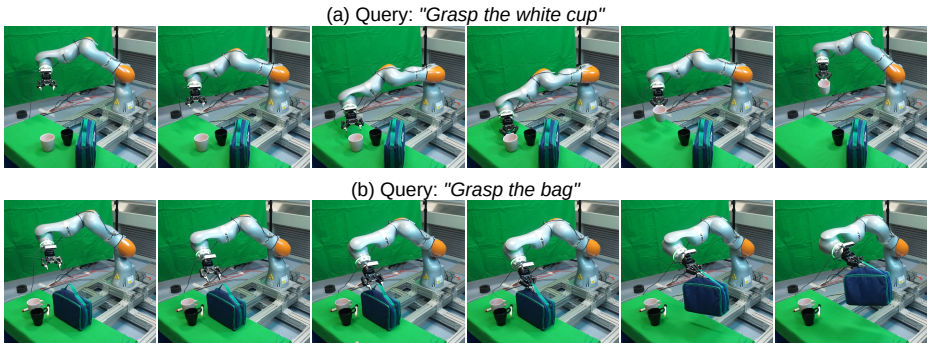

Fig. 2: Snapshots of two example robotic experiments.

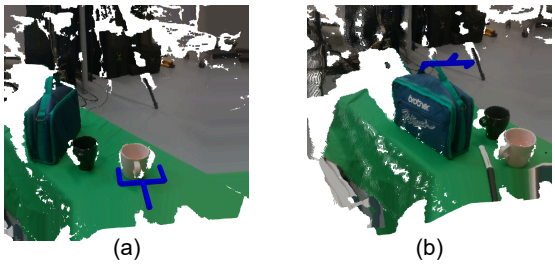

Fig. 3: Detection results in robotic experiments. Point clouds are captured from a RealSense camera with experiments in Figure 2.

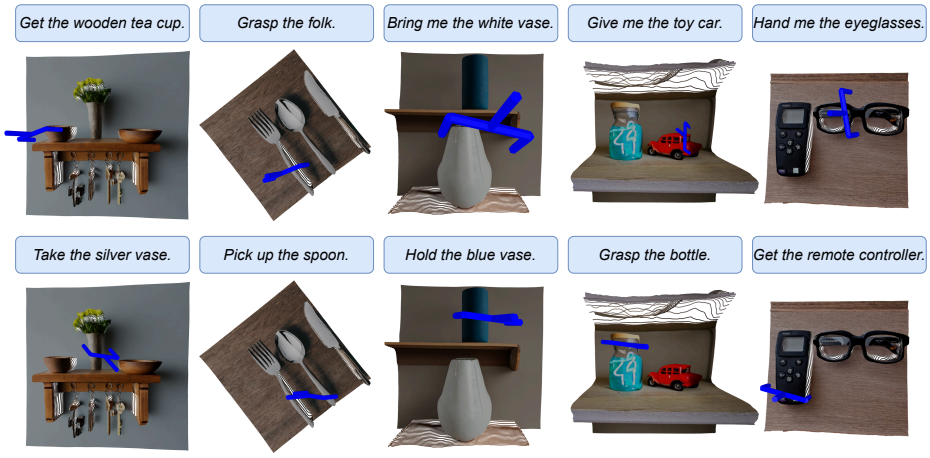

Fig. 4: Additional qualitative results.

References

1. Ainetter, S., Fraundorfer, F.: End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb. In: ICRA (2021)
2. Ajay, A., Han, S., Du, Y., Li, S., Gupta, A., Jaakkola, T., Tenenbaum, J., Kaelbling, L., Srivastava, A., Agrawal, P.: Compositional foundation models for hierarchical planning. NeurIPS (2024)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL (2019)
4. Du, Y., Li, S., Mordatch, I.: Compositional visual generation with energy based models. NeurIPS (2020)
5. Du, Y., Li, S., Tenenbaum, J., Mordatch, I.: Improved contrastive divergence training of energy-based models. In: ICML (2021)
6. Du, Y., Mordatch, I.: Implicit generation and modeling with energy based models. NeurIPS (2019)
7. Du, Y., Yang, S., Dai, B., Dai, H., Nachum, O., Tenenbaum, J., Schuurmans, D., Abbeel, P.: Learning universal policies via text-guided video generation. NeurIPS (2024)
8. Eppner, C., Mousavian, A., Fox, D.: Acronym: A large-scale grasp dataset based on simulation. In: ICRA (2021)
9. Fang, H.S., Wang, C., Gou, M., Lu, C.: Graspnet-1billion: A large-scale benchmark for general object grasping. In: CVPR (2020)
10. Grathwohl, W., Wang, K.C., Jacobsen, J.H., Duvenaud, D., Zemel, R.: Learning the stein discrepancy for training and evaluating energy-based models without sampling. In: ICML (2020)
11. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS (2020)
12. Ko, P.C., Mao, J., Du, Y., Sun, S.H., Tenenbaum, J.B.: Learning to act from actionless videos through dense correspondences. In: ICLR (2024)
13. Kumra, S., Joshi, S., Sahin, F.: Antipodal robotic grasping using generative residual convolutional neural network. In: IROS (2020)
14. Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., Hoi, S.C.H.: Align before fuse: Vision and language representation learning with momentum distillation. NeurIPS (2021)
15. Liu, N., Li, S., Du, Y., Torralba, A., Tenenbaum, J.B.: Compositional visual generation with composable diffusion models. In: ECCV (2022)
16. Mishra, U.A., Xue, S., Chen, Y., Xu, D.: Generative skill chaining: Long-horizon skill planning with diffusion models. In: CoRL (2023)
17. Morrison, D., Corke, P., Leitner, J.: Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. In: RSS (2018)
18. Mousavian, A., Eppner, C., Fox, D.: 6-dof graspnet: Variational grasp generation for object manipulation. In: ICCV (2019)
19. Nguyen, T., Vu, M.N., Huang, B., Van Vo, T., Truong, V., Le, N., Vo, T., Le, B., Nguyen, A.: Language-conditioned affordance-pose detection in 3d point clouds. ICRA (2024)
20. Ni, P., Zhang, W., Zhu, X., Cao, Q.: Pointnet++ grasping: Learning an end-to-end spatial grasp generation algorithm from sparse point clouds. In: ICRA (2020)
21. Nijkamp, E., Hill, M., Han, T., Zhu, S.C., Wu, Y.N.: On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In: AAAI (2020)

22. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. NeurIPS (2017)
23. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
24. Shridhar, M., Manuelli, L., Fox, D.: Cliport: What and where pathways for robotic manipulation. In: CoRL (2022)
25. Tang, C., Huang, D., Ge, W., Liu, W., Zhang, H.: Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping. RA-L (2023)
26. Tang, C., Huang, D., Meng, L., Liu, W., Zhang, H.: Task-oriented grasp prediction with visual-language inputs. IROS (2023)
27. Tziafas, G., Yucheng, X., Goel, A., Kasaei, M., Li, Z., Kasaei, H.: Language-guided robot grasping: Clip-based referring grasp synthesis in clutter. In: CoRL (2023)
28. Urain, J., Funk, N., Peters, J., Chalvatzaki, G.: Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In: ICRA (2023)
29. Vuong, A.D., Vu, M.N., Huang, B., Nguyen, N., Le, H., Vo, T., Nguyen, A.: Language-driven grasp detection. In: CVPR (2024)
30. Vuong, A.D., Vu, M.N., Le, H., Huang, B., Huynh, B., Vo, T., Kugi, A., Nguyen, A.: Grasp-anything: Large-scale grasp dataset from foundation models. In: ICRA (2024)
31. Wang, T.H.J., Zheng, J., Ma, P., Du, Y., Kim, B., Spielberg, A., Tenenbaum, J., Gan, C., Rus, D.: Diffusebot: Breeding soft robots with physics-augmented generative diffusion models. NeurIPS (2024)
32. Xian, Z., Gkanatsios, N., Gervet, T., Fragkiadaki, K.: Unifying diffusion models with action detection transformers for multi-task robotic manipulation. In: CoRL (2023)
33. Xu, K., Zhao, S., Zhou, Z., Li, Z., Pi, H., Zhu, Y., Wang, Y., Xiong, R.: A joint modeling of vision-language-action for target-oriented grasping in clutter. In: ICRA (2023)
34. Xu, M., Xu, Z., Chi, C., Veloso, M., Song, S.: Xskill: Cross embodiment skill discovery. In: CoRL (2023)
35. Xu, X., Ha, H., Song, S.: Dynamics-guided diffusion model for robot manipulator design. arXiv preprint arXiv:2402.15038 (2024)
36. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: ICCV (2021)