Functional D(R,O) Grasp: A Language-Guided Cross-Embodiment Functional Dexterous Grasping

Anonymous Author(s)
Affiliation
Address
email

Abstract: Functional dexterous grasping is a challenging capability essential for robots to achieve intent-aligned interactions with objects. Existing methods primarily focus on grasp stability without addressing functional intent. In this work, we present Functional D(R,O) Grasp, a language-guided framework that enables intent-aligned grasp generation while ensuring cross-platform adaptability. We learn platform-agnostic intermediate representations that enable translation from functional grasp language input to execution across different robotic hands. This framework generates appropriate grasps for objects based on their intended use, covering multiple functional requirements (use, hold, handover, liftup). We demonstrate that our approach achieves a 75.1% success rate in simulation on unseen objects, significantly outperforming baselines. Real-world experiments with the LeapHand platform further validate our approach. Our work bridges the gap between functional intent and cross-platform dexterous execution, enabling robots to perform purposeful grasps with a single unified model.

Keywords: Dexterous Grasping, Functional Manipulation, Cross-Platform Capabilities

1 Introduction

Dexterous robotic grasping, particularly intent-aligned functional grasping, represents a critical milestone in advancing robotic systems toward practical applications. The ability for robots to grasp objects in ways that fulfill specific functional requirements—whether for object manipulation, tool use, or human-robot interaction—is essential for effective operation in real-world environments.

Significant progress has been made in stable dexterous grasping through various approaches. Traditional optimization-based methods first achieved stable grasps by modeling contact forces and friction cones, while more recent learning-based techniques have improved both efficiency and success rates. These approaches include direct joint angle generation through diffusion models or reinforcement learning, object-centric methods using contact points or heatmaps, and implicit hand-object representations. In parallel, functional grasping for two-finger grippers has advanced through vision-language models that can identify task-appropriate grasp points. However, cross-platform functional dexterous grasping—where a single model can generate functionally appropriate grasps across different robotic hand designs—remains substantially underdeveloped.

Two key challenges impede progress in this domain. First, most existing methods prioritize grasp stability without adequately addressing functional requirements. This limitation stems primarily from how dexterous hand datasets are typically collected in simulators or through optimization methods that prioritize stability metrics, making it difficult to incorporate diverse functional intents. Second, cross-platform functional dexterous grasping presents significant technical hurdles. Approaches using functional contact maps and contact points can generalize across platforms but require lengthy optimization processes. Diffusion models easily incorporate functional language as

Submitted to the 9th Conference on Robot Learning (CoRL 2025). Do not distribute.

conditional input but lack physical interaction during generation, often leading to suboptimal grasps. D(R,O) Grasp offers cross-platform generalization with controllable optimization time but assumes consistent wrist poses between input and output, limiting its application to functional grasping where wrist pose adjustments are often necessary.

To address these limitations, we present Functional D(R,O) Grasp, a language-guided framework for cross-platform functional grasping. Our approach first translates functional language instructions into wrist poses and contact anchor points, which refine the coarse interaction intent. These elements then feed into a platform-agnostic intermediate representation that unifies hand-object distance relationships, enabling precise joint configuration synthesis across different dexterous hand platforms. This coarse-to-fine pipeline bridges the semantic gap between high-level functional intent and low-level interactions while possessing cross-platform capabilities. Our contributions are summarized as follows:

- We enable functional grasping across multiple dexterous robotic hands in a single model through a coarse-to-fine pipeline with platform-agnostic intermediate representation.
- We develop semantic-conditioned grasping strategies that achieve 75.1% success rate on generating functionally appropriate grasps for unseen objects, significantly outperforming existing functional grasping baselines.
- We create a workflow for generating high-quality dexterous hand grasping data by mapping human functional demonstrations to collision-free robotic hand configurations through retargeting and optimization.

2 Related Works

Learning-Based Dexterous Grasping. Data-driven approaches for dexterous grasping have made significant advances and can be categorized into three main approaches. The first approach generates joint values directly through diffusion models [1, 2]. However, these methods typically show limited cross-platform generalization. Additionally, they lack physical interaction abilities during both training and generation processes, and often need test-time adaptation [3, 4] or denoising guidance [5] to work well. The second approach employs contact points [6] or affordance maps [7] to predict grasp interactions. While supporting cross-platform adaptation, these approaches face computational challenges due to the high-dimensional solution space. The third approach, represented by [8], uses neural networks to model hand-object distances, offering cross-platform capabilities and effective grasping. However, due to consistency requirements in robot encoding, this approach typically constrains output wrist poses to remain close to input poses, limiting application flexibility. In contrast, our approach flexibly accommodates conditional inputs without constraints while maintaining cross-platform generalization.

Functional Grasping. Functional grasping bridges human intent and robotic manipulation capabilities, representing a critical research direction in robotics. For parallel grippers, recent approaches have leveraged 3D vision and multimodal models. GraspSplats [9] constructs feature-enhanced 3D Gaussian models to segment functional regions, while feature distillation grasping [10] employs Distilled Feature Fields for semantic extraction. CoPA [11] implements a hierarchical perception approach using Set-of-Mask annotations processed through GPT-4V for grasp region localization. Robo-ABC [12] leverages a database of annotated functional contact points with CLIP [13] for retrieval-based transfer. These approaches primarily provide single-point coordinates requiring subsequent grasp sampling like [14], limiting their applicability to dexterous hands requiring complex optimization. Extending to dexterous functional grasping, contact code methodologies [15, 16, 17] segment both object and hand into different regions, creating paired contact codes to guide grasping through optimization. These methods require meticulous manual annotations for each object's functional regions, limiting their scalability to novel objects. Other approaches [4] utilize conditional diffusion models to accommodate diverse functional requirements but often cannot escape the aforementioned limitations of diffusion models. In contrast, our work enables language-guided functional dexterous grasping with cross-platform adaptability and efficient optimization.

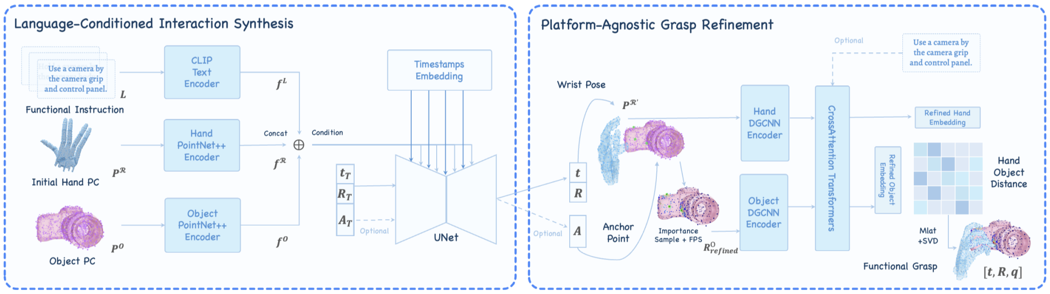

Figure 1: Overview of our Functional D(R,O) Grasp framework. Left: Language-Conditioned Interaction Synthesis translates functional instructions into wrist poses and contact anchor points via a diffusion model. Right: Platform-Agnostic Grasp Refinement converts these interaction elements into a unified hand-object distance representation to generate precise joint configurations across different robotic hands.

Dexterous Grasping Datasets. Dexterous grasping datasets have evolved significantly, with contributions including [18, 19]. However, these primarily rely on simulation and optimization, limiting their capacity to represent diverse functional intents. OakInk [20] provides MoCap-based functional grasping data using the MANO [21] hand model, covering four grasping intents across various object categories. Our methodology builds upon these human demonstrations, constructing corresponding robotic hand datasets through efficient retargeting [22] and grasp energy-based optimization [23, 24].

3 Methodology

3.1 Problem Formulation

Let an object point cloud with $N_{\mathcal{O}}$ points be $\mathbf{P}^{\mathcal{O}} \in \mathbb{R}^{N_{\mathcal{O}} \times 3}$, where each point contains 3D position coordinates.

The complete dexterous hand configuration can be represented as a tuple $[\mathbf{t}, \mathbf{R}, \mathbf{q}]$, where $\mathbf{t} \in \mathbb{R}^3$ is the 3D translation of the wrist, $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is the rotation matrix representing wrist orientation, and $\mathbf{q} \in \mathbb{R}^{D_q}$ represents the finger joint angles with D_q dependent on the specific robotic hand.

For each robot hand, we sample point clouds at fixed positions on the surface of each link, denoted as $\{\mathbf{P}_{\ell_i}\}_{i=1}^{N_\ell}$, where N_ℓ is the number of links. Given a hand configuration $[\mathbf{t}, \mathbf{R}, \mathbf{q}]$, we apply forward kinematics to obtain the corresponding robot point cloud $\mathbf{P}^{\mathcal{R}} \in \mathbb{R}^{N_{\mathcal{R}} \times 3}$.

For each grasp, we define a structured language instruction $\mathcal{L}$ in the format: *[Grasp Intent] a [Object Name] by [Part]*, where [Grasp Intent] can be one of {Use, Hold, Handover, Liftup}, [Object Name] identifies the target object, and [Part] specifies the primary contact part of the object. This instruction is encoded into a language embedding $\mathbf{f}^{\mathcal{L}} \in \mathbb{R}^{D_f}$ using a ViT-B/32 text encoder.

3.2 Coarse-to-Fine Functional Grasp Synthesis

We propose a coarse-to-fine approach that progressively refines language instructions into precise grasp configurations through platform-agnostic intermediate representations. Our approach consists of two key stages: (1) language-conditioned synthesis of coarse interaction elements (wrist pose and contact anchor points), and (2) refinement of these elements into precise hand configurations through a cross-platform intermediate representation.

3.2.1 Language-Conditioned Interaction Synthesis

Unlike previous approaches that generate complete joint configurations directly, we first translate functional language instructions into essential interaction elements that define how the hand should engage with the object. Specifically, we model wrist pose $[\mathbf{t}, \mathbf{R}]$ and functional contact anchor points

$\mathbf{A} \in \mathbb{R}^{K\times 3}$ on the object surface, where K is the number of anchor points. We set $K = 4$ in our implementation.

We implement a conditional denoising diffusion probabilistic model (DDPM) [25]. We first extract features from both the object and robot representations. The object point cloud $\mathbf{P}^{\mathcal{O}} \in \mathbb{R}^{N_{\mathcal{O}}\times 3}$ and robot hand point cloud $\mathbf{P}^{\mathcal{R}} \in \mathbb{R}^{N_{\mathcal{R}}\times 3}$ are processed through PointNet++ [26] encoders, getting feature representation $\mathbf{f}^{\mathcal{O}}$, $\mathbf{f}^{\mathcal{R}} \in \mathbb{R}^{N_f\times D_f}$ respectively. These features are concatenated with the language embedding $\mathbf{f}^{\mathcal{L}}$ to form the conditional input $\mathbf{f} = [\mathbf{f}^{\mathcal{R}}; \mathbf{f}^{\mathcal{O}}; \mathbf{f}^{\mathcal{L}}]$ for the diffusion model.

During the diffusion process, we follow a fixed noise schedule β_t to gradually corrupt the original interaction elements through a Markov process:

$$q([\mathbf{t}_t, \mathbf{R}_t, \mathbf{A}_t]|[\mathbf{t}_{t-1}, \mathbf{R}_{t-1}, \mathbf{A}_{t-1}]) = \mathcal{N}([\mathbf{t}_t, \mathbf{R}_t, \mathbf{A}_t]; \sqrt{1-\beta_t}[\mathbf{t}_{t-1}, \mathbf{R}_{t-1}, \mathbf{A}_{t-1}], \beta_t \mathbf{I}) \quad (1)$$

With $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}t = \prod i = 1^t \alpha_i$, this corruption process can be expressed directly in terms of the original elements:

$$q([\mathbf{t}_t, \mathbf{R}_t, \mathbf{A}_t]|[\mathbf{t}_0, \mathbf{R}_0, \mathbf{A}_0]) = \mathcal{N}([\mathbf{t}_t, \mathbf{R}_t, \mathbf{A}_t]; \sqrt{\bar{\alpha}_t}[\mathbf{t}_0, \mathbf{R}_0, \mathbf{A}_0], (1-\bar{\alpha}_t)\mathbf{I}) \quad (2)$$

The diffusion model is trained with a mean-squared error objective:

$$L_{simple} = \mathbb{E}_{t,[\mathbf{t}_0,\mathbf{R}_0,\mathbf{A}0],\epsilon}[||\epsilon - \epsilon_\phi([\mathbf{t}_t, \mathbf{R}_t, \mathbf{A}_t], \mathbf{f}, t)||^2] \quad (3)$$

where ϵ_ϕ is a transformer-based noise prediction network with cross-attention to the conditional embedding $\mathbf{f}$. During sampling, the model reverses the diffusion process to generate interaction elements conditioned on the feature representation:

$$p_\theta([\mathbf{t}, \mathbf{R}, \mathbf{A}]|\mathbf{f}) = p([\mathbf{t}_T, \mathbf{R}_T, \mathbf{A}_T]) \prod_{t=1}^{T} p_\theta([\mathbf{t}_{t-1}, \mathbf{R}_{t-1}, \mathbf{A}_{t-1}]|[\mathbf{t}_t, \mathbf{R}_t, \mathbf{A}_t], \mathbf{f}) \quad (4)$$

3.2.2 Platform-Agnostic Grasp Refinement

We design a platform-agnostic intermediate representation that seamlessly integrates the coarse interaction elements from the previous stage. This representation translates them into a unified hand-object distance matrix that satisfies the functional intent, enabling precise prediction of joint parameters across various robotic hand platforms.

First, we derive a contact importance map $\Omega \in \mathbb{R}^{N_{\mathcal{O}}}$ from the predicted anchor points $\mathbf{A}$. For each point p_i in the object point cloud, we compute its distance to the nearest anchor point:

$$d(p_i, \mathbf{A}) = \min_{a_j \in \mathbf{A}} ||p_i - a_j||_2 \quad (5)$$

We then normalize these distances using a sigmoid-based function to create the contact importance map:

$$\Omega_i = 1 - 2 \cdot (\text{Sigmoid}(2d(p_i, \mathbf{A})) - 0.5) \quad (6)$$

This importance map highlights regions of the object that should be contacted based on the functional intent.

We use importance sampling based on the values in Ω to select 256 contact-critical points $\mathbf{P}^{\mathcal{O}}_{\text{crit}}$ from the object point cloud. We then sample an additional 256 points $\mathbf{P}^{\mathcal{O}}_{\text{FPS}}$ using farthest point sampling (FPS) to ensure comprehensive coverage of the object geometry. The final object representation is the concatenation of these point sets with their corresponding importance values:

$$\mathbf{P}^{\mathcal{O}}_{\text{refined}} = \{[\mathbf{P}^{\mathcal{O}}_{\text{crit}}, \Omega_{\text{crit}}], [\mathbf{P}^{\mathcal{O}}_{\text{FPS}}, \Omega_{\text{FPS}}]\} \in \mathbb{R}^{512\times 4} \quad (7)$$

Next, we reposition the hand point cloud using the predicted wrist pose while maintaining an open finger configuration with small random variations to ensure diversity in the initialization:

$$\mathbf{P}^{\mathcal{R}'} = \text{FK}\left([\mathbf{t}, \mathbf{R}, \mathbf{q}_{\text{init}}], \{\mathbf{P}_{\ell_i}\}_{i=1}^{N_\ell}\right) \in \mathbb{R}^{N_{\mathcal{R}}\times 3} \quad (8)$$

We extract point-wise features from both the hand and refined object point clouds using DGCNN [27] encoders and incorporate language information into these features:

$$\tilde{\phi}^{\mathcal{R}} = \mathcal{F}_{\text{int}}(f_d^{\mathcal{R}}(\mathbf{P}^{\mathcal{R}'}), \mathbf{f}^{\mathcal{L}}) \in \mathbb{R}^{N_{\mathcal{R}} \times D_f} \tag{9}$$

$$\tilde{\phi}^{\mathcal{O}} = \mathcal{F}_{\text{int}}(f_d^{\mathcal{O}}(\mathbf{P}^{\mathcal{O}}_{\text{refined}}), \mathbf{f}^{\mathcal{L}}) \in \mathbb{R}^{512 \times D_f} \tag{10}$$

where $\mathcal{F}_{\text{int}}$ is a feature integration function that combines point features with language embeddings.

Following [8], we establish correspondences between robot and object features using cross-attention transformers, resulting in transformed feature representations $\psi^{\mathcal{R}}$ and $\psi^{\mathcal{O}}$. We then compute a distance representation between each pair of hand and object points:

$$\mathcal{D}(\mathcal{R}, \mathcal{O})_{ij} = \mathcal{K}(\psi_i^{\mathcal{R}}, \psi_j^{\mathcal{O}}) \tag{11}$$

where $\mathcal{D}(\mathcal{R}, \mathcal{O})_{ij}$ represents the predicted distance between the i-th hand point and the j-th object point, and $\mathcal{K}$ is implemented as a softplus function followed by a MLP.

Through spatial point cloud localization algorithms, we derive the final grasp joint values $\mathbf{q}$ from this distance matrix, resulting in a complete grasp configuration $[\mathbf{t}, \mathbf{R}, \mathbf{q}]$.

We train this refinement network using a combination of losses:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{dist}}\mathcal{L}_{\text{dist}} + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}} + \lambda_{\text{SE(3)}}\mathcal{L}_{\text{SE(3)}} \tag{12}$$

where $\mathcal{L}_{\text{dist}}$ measures L1 distance between predicted and true hand-object distances, $\mathcal{L}_{\text{depth}}$ prevents collisions using SDF, $\mathcal{L}_{\text{SE(3)}}$ calculates differences between predicted and true 6D poses.

It is worth noting that our platform-agnostic intermediate representation is highly flexible, capable of accepting anchor points as input and solving for grasps in a short time, making it straightforward to interface with higher-level vision-language models (VLMs) or vision-language action models (VLAs), thereby further broadening its range of applicable scenarios. Additionally, the representation can also accommodate other conditional inputs to guide the learning of hand-object distance relationships, demonstrating its versatility across various functional grasping contexts.

3.3 Dataset Construction

We leveraged human hand functional demonstrations from the OakInk dataset [20] and converted them to dexterous hand configurations for multiple robotic platforms. The dataset construction workflow consists of the following two components, which efficiently generate collision-free functional grasps:

Human-to-Robot Grasp Retargeting. Using the AnyTeleop [22] framework, we retargeted MANO [21] hand parameters to ShadowHand (5-finger), Allegro (4-finger) and LeapHand (4-finger). To address size differences, we applied appropriate scaling to both the MANO hand and objects to optimize the retargeting process.

Since retargeting alone does not guarantee force closure and may introduce penetration issues, we applied grasp energy-based optimization from BoDex [23] to refine the generated grasps. After optimization, we validated each grasp in MuJoCo simulation following the DexGraspBench protocol and retained only the successful cases. Detailed retargeting and optimization parameters can be found in Appendix B.

Functional Language Construction. For each grasp, we constructed a functional language instruction following the format "[Grasp Intent] a [Object Name] by [Part]". We used OakInk's original annotations for [Grasp Intent] and [Object Name], while determining [Part] through analysis of hand-object interactions. Let $F = \{f_1, f_2, ..., f_5\}$ represent the fingertip points on the hand and $\mathcal{P}_j$ denote points belonging to object part j. We first determine each fingertip's contact part:

$$C(f_i) = \begin{cases} \arg\min_j \min_{p \in \mathcal{P}_j} \|f_i - p\| & \text{if } \min_{p \in \mathcal{P}} \|f_i - p\| < 0.05m \\ \arg\min_j \|f_i - c_j\| & \text{otherwise} \end{cases}$$

where c_j is the centroid of part j. The primary contact part is then:

$$[\text{Part}] = \arg\max_j \sum_{i=1}^{5} \mathbb{I}[C(f_i) = j]$$

where $\mathbb{I}[\cdot]$ is the indicator function. This approach effectively identifies the primary interaction region even for suspended grasps with limited contact points.

The functional contact anchor points $\mathbf{A}$ for training our model are constructed from the object points that have minimal distances to each fingertip link.

4 Experiment

4.1 Dataset

Following the dataset construction workflow described in Section 3.3, we use three retargeting robotic hand datasets: Shadowhand, Allegro and LeapHand. We split the dataset by objects with an 8:1:1 ratio for training, validation, and testing.

4.2 Evaluation Metrics

To comprehensively evaluate our approach, we employ two complementary metrics that assess both physical grasp stability and functional intent alignment:

Success Rate: We evaluate grasp stability in mujoco simulation. Each grasp starts from a pre-grasp pose and closes to a squeeze pose. We apply gravity along six orthogonal directions. A grasp is considered successful if the object's displacement remains within 5 cm for over 3 seconds in all directions.

Functionality: We assess functionality via chamfer distance metrics and human evaluation. For human assessment, each object category, we sample 3 objects and 2 functional instructions. Annotators are shown grasp images with corresponding instructions. Our evaluation protocol includes two assessment types: (1) comparative evaluation, where participants rank pairs of grasps based on their functional appropriateness, and (2) absolute scoring, where participants rate each grasp on a 0-3 scale (3: fully satisfies the functional intent, 2: mostly satisfies, 1: partially satisfies, 0: does not satisfy).

4.3 Qualitative Results

We evaluate our approach against baseline methods and analyze cross-platform performance to validate our coarse-to-fine functional grasping framework's effectiveness.

Comparison with Diffusion-Based Methods. Since existing functional dexterous grasping models and their corresponding datasets are not publicly available, we compare with Scene-Diffuser [2], a representative diffusion-based hand pose generation method. We modified Scene-Diffuser to accept functional language embeddings as input to enable fair comparison. The results are shown in Table 1.

Our approach significantly outperforms the baseline in success rate, achieving 75% compared to Scene-Diffuser's 41%. We attribute this improvement to our coarse-to-fine design philosophy. By employing diffusion models as generators of initial representations, we effectively process conditional language inputs and refine them into appropriate wrist pose and anchor point representations. Subsequently, our platform-agnostic intermediate representation layer is particularly well-suited for processing and applying low-level conditional inputs, enabling accurate hand configuration across multiple dexterous platforms and refining finger-object contacts through the hand-object distance representation. Functionality [Waiting for writing].

Cross-Platform Performance. To evaluate our framework's cross-platform capabilities, we trained models on different combinations of robotic hand data. The results in Table 1 show that our multi-platform model (3 hand version) trained on Shadowhand, Allegro, and LeapHand data achieves a

Table 1: Comparison with baseline on unseen objects

Model	SSR↑ (Success Rate)	CD↓ (Distance)
Scene-Diffuser (Shadowhand only)	41%	3.06
Ours (Shadowhand only)	66%	2.64
Ours (3 hand version)	**75.1%**	**2.61**
Ours Allegro (3 hand version)	65%	5.8
Ours LeapHand (3 hand version)	40%	6.5

Table 2: Success counts in real-world experiments across different objects (out of 10 trials)

Object	Success
Lotion Pump	-
Cylinder Bottle	-
Mug	-
Teapot	-
Bowl	-
Cup	-
Knife	-
Pen	-
Bottle	-
Headphones	-
Average	-

higher success rate (75.1%) compared to the single-platform model (66%). This performance gain demonstrates the benefit of learning from diverse hand morphologies, which enhances the model's ability to generalize functional grasping principles.

4.4 Real-Robot Experiments

We conducted real-world experiments using LeapHand and an overhead RealSense D435 camera. We tested our method on 10 unseen household objects with varying geometries and functional requirements. Each object was tested with 10 execution trials, Table 2 shows the success rates across different objects.

5 Conclusion

We present Functional D(R,O) Grasp, a language-guided framework for functional dexterous grasping with cross-platform adaptability. Our coarse-to-fine approach first predicts appropriate wrist poses and anchor points through a conditional diffusion model, then optimizes finger configurations using hand-object distance representations. This platform-agnostic intermediate representation effectively bridges the gap between language-specified intent and physical execution across different robotic hands. Experimental results demonstrate our method achieves a 75.1% success rate on unseen objects in simulation and transfers successfully to real-world scenarios using the LeapHand platform.

Our current approach has two main limitations: performance degrades when handling objects from unseen categories beyond our training distribution, and the functional categories we explore (use, hold, handover, liftup) do not yet cover the full spectrum of manipulation intents.

Future directions include enriching the hand-object representation methods to provide more robust intermediate representations, and gradually improving support for out-of-distribution object grasping.

References

[1] J. Lu, H. Kang, H. Li, B. Liu, Y. Yang, Q. Huang, and G. Hua. Ugg: Unified generative grasping. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXVII*, page 414–433, Berlin, Heidelberg, 2024. Springer-Verlag.

[2] S. Huang, Z. Wang, P. Li, et al. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16750–16761, 2023.

[3] H. Jiang, S. Liu, J. Wang, and X. Wang. Hand-object contact consistency reasoning for human grasps generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11107–11116, 2021.

[4] Y.-L. Wei, J.-J. Jiang, C. Xing, X.-T. Tan, X.-M. Wu, H. Li, M. Cutkosky, and W.-S. Zheng. Grasp as you say: Language-guided dexterous grasp generation. *arXiv preprint arXiv:2405.19291*, 2024.

[5] Y. Zhong, Q. Jiang, J. Yu, and Y. Ma. Dexgrasp anything: Towards universal robotic dexterous grasping with physics awareness. *arXiv preprint arXiv:2503.08257*, 2025.

[6] L. Shao, F. Ferreira, M. Jorda, et al. Unigrasp: Learning a unified model to grasp with multi-fingered robotic hands. *IEEE Robotics and Automation Letters*, 5(2):2286–2293, 2020.

[7] P. Li, T. Liu, Y. Li, et al. Gendexgrasp: Generalizable dexterous grasping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8068–8074, 2023.

[8] Z. Wei, Z. Xu, J. Guo, et al. D (r, o) grasp: A unified representation of robot and object interaction for cross-embodiment dexterous grasping. *arXiv preprint arXiv:2410.01702*, 2024.

[9] M. Ji, R. Qiu, X. Zou, et al. Graspsplats: Efficient manipulation with 3d feature splatting. *arXiv preprint arXiv:2409.02084*, 2024.

[10] W. Shen, G. Yang, A. Yu, et al. Distilled feature fields enable few-shot language-guided manipulation. *arXiv preprint arXiv:2308.07931*, 2023.

[11] H. Huang, F. Lin, Y. Hu, et al. Copa: General robotic manipulation through spatial constraints of parts with foundation models. *arXiv preprint arXiv:2403.08248*, 2024.

[12] Y. Ju, K. Hu, G. Zhang, et al. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation. In *European Conference on Computer Vision*, pages 222–239, 2025.

[13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[14] H. Fang, C. Wang, H. Fang, et al. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023.

[15] T. Zhu, R. Wu, X. Lin, et al. Toward human-like grasp: Dexterous grasping via semantic representation of object-hand. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15741–15751, 2021.

[16] T. Zhu, R. Wu, J. Hang, et al. Toward human-like grasp: Functional grasp by dexterous robotic hand via object-hand semantic representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12521–12534, 2023.

[17] Y. Zhang, J. Hang, T. Zhu, et al. Functionalgrasp: Learning functional grasp for robots via semantic hand-object representation. *IEEE Robotics and Automation Letters*, 8(5):3094–3101, 2023.

[18] Y. Chao, W. Yang, Y. Xiang, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021.

[19] 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 *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11359–11366. IEEE, 2023.

[20] L. Yang, K. Li, X. Zhan, et al. Oakink: A large-scale knowledge repository for understanding hand-object interaction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20953–20962, 2022.

[21] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, Nov. 2017. URL http://doi.acm.org/10.1145/3130800.3130883.

[22] Y. Qin, W. Yang, B. Huang, et al. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. *arXiv preprint arXiv:2307.04577*, 2023.

[23] J. Chen, Y. Ke, and H. Wang. Bodex: Scalable and efficient robotic dexterous grasp synthesis using bilevel optimization. *arXiv preprint arXiv:2412.16490*, 2024.

[24] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. V. Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, N. Ratliff, and D. Fox. curobo: Parallelized collision-free minimum-jerk robot motion generation, 2023.

[25] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[26] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[27] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

A Implementation Details

A.1 Point Cloud Processing

For the object point cloud, when processing input for the DDPM, we perform random sampling to obtain 2048 points from the original 65536 points. For point clouds input to the Platform-Agnostic Grasp Refinement, when using Anchor Points as a condition during training, we perform importance sampling based on Contact Map values to select 256 points, then use these points as initialization to perform Farthest Point Sampling (FPS) for the remaining 256 points, resulting in a total of 512 points. When not using Anchor Points as a condition, we directly perform random sampling to obtain 512 points.

B Human-to-Robot Retargeting

B.1 Retargeting Configuration

We follow the AnyTeleop [22] framework for retargeting, with scaling factors determined based on the size relationships between the MANO [21] Hand and the robotic hands. We enlarged both the objects and hands to achieve better remapping effects: ShadowHand with no scaling, Allegro with a scaling factor of 1.9, and LeapHand with a scaling factor of 1.8.

Since we use static frames for retargeting while the AnyTeleop retargeting process is designed for sequences, we repeated each static frame 20 times to create a sequence, better aligning with AnyTeleop's design flow.

B.2 Grasp Optimization Parameters

Since BoDex was originally designed for grasp synthesis, while our grasps are already reasonably correct configurations, we modified its optimization parameters to avoid large deviations from the initial retargeted positions. We reduced the joint angle search amplitude to 0.01 for ShadowHand and 0.05 for LeapHand and Allegro, maintaining a balance between optimization and position preservation.

C Additional Results

C.1 Ablation Studies

During the training process of our three-hand data version, we conducted ablation studies to evaluate the influence of different components.

Effect of Contact Anchor Points We observed that introducing contact anchor points can slightly reduce the Chamfer Distance (CD) while supporting more flexible inputs and faster convergence. However, we noticed a small performance decrease in success rate testing, as shown in Table 3.

Table 3: Ablation studies on the impact of contact anchor points

Configuration	Success Rate (SSR)	Chamfer Distance (CD)
Without Anchor Points	75.1%	2.61
With Contact Anchor Points	71.4%	2.58

Wrist Pose Prediction Analysis For wrist pose prediction, we found that without the prediction of wrist pose, the Platform-Agnostic Grasp Refinement component cannot correctly predict the corresponding hand-object distance representation matrix due to limitations in the robot encoder's ability to encode different rotations. This renders the component unable to function properly. We plan to further explore the influence of encoder configurations on this component in future experimental designs.