GeoDexGrasp: Geometry-aware Generation for Data-efficient and Physics-plausible Dexterous Grasping

Supplementary Material

Appendix Contents

A. PointSO for Semantic Orientation Extraction 1
1. Architecture Details 1
2. Integration in GeoDexGrasp 2
B. Implement Details 2
1. Loss Function Designs 2
2. Training Details 2
C. Model Introductions 4
1. Introduction of IPDF 4
2. Diffusion Models for Dexterous Grasp Generation 5
D. Evaluation Results 6
1. Grasp Diversity Experiments 6
2. Rotation Generalization Experiments 7
E. More Visualizations 7
F. Discussions about Rotations 7

A. PointSO for Semantic Orientation Extraction

PointSO[30] is a cross-modal 3D Transformer architecture as 7 designed for semantic orientation prediction from 3D point clouds and natural language descriptions. Given an object's point cloud X and a language description ℓ, PointSO predicts a unit direction vector $\mathbf{s}_\ell^X \in S(2)$ that corresponds to the semantic orientation described by ℓ.

A.1. Architecture Details

The model takes two modalities as input:**3D Point Cloud**: $X = \{\mathbf{x}_i \in \mathbb{R}^3\}_{i=1}^N$ representing the object geometry and **Language Description**: ℓ as an open-vocabulary phrase referring to directions, object parts, or interactions.

A.1.1. Feature Embedding and Cross-Modal Fusion

The feature embedding process handles both geometric and linguistic inputs in a unified framework. For the 3D point cloud X, we first sample N_s seed points using Farthest Point Sampling (FPS) and construct local neighborhoods through K-Nearest Neighbors (KNN) grouping. Local geometric features are then extracted via a lightweight PointNet module and processed through a 3D Transformer backbone to obtain comprehensive geometric embeddings. Concurrently, the language description ℓ is encoded using the CLIP text encoder, with the global text token serving as the linguistic representation for cross-modal fusion.

We employ a simple yet effective token-wise addition strategy for cross-modal fusion:

$$\mathbf{h}_i^{\text{fused}} = \mathbf{h}_i^{\text{point}} + \mathbf{h}^{\text{text}} \tag{10}$$

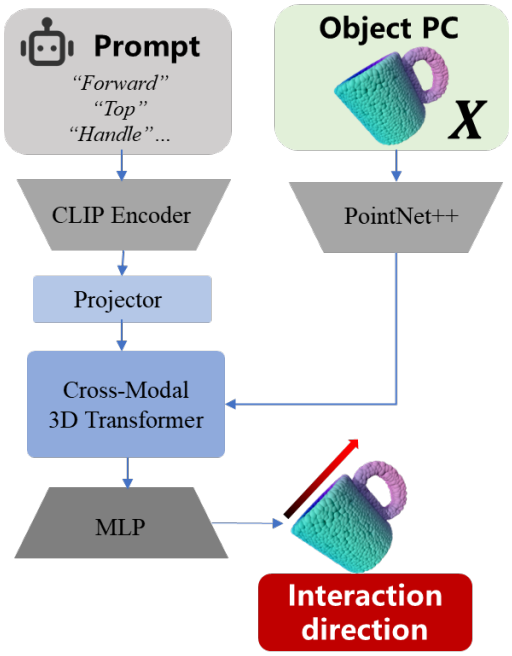

Figure 7. Architecture of PointSO.As a cross-modal 3D transformer,it finally outputs a unit direction vector.

where $\mathbf{h}_i^{\text{point}}$ represents point token features and $\mathbf{h}^{\text{text}}$ denotes the global text feature. This approach empirically outperforms alternative fusion methods such as cross-attention and concatenation for semantic orientation prediction, particularly benefiting from its ability to reinforce linguistic influence across network layers when processing short language inputs.

A.1.2. Orientation Prediction and Training Objective

The orientation prediction module processes the fused cross-modal features to generate the final semantic orientation vector. Specifically, we extract the special `[CLS]` token from the fused feature sequence and map it through an MLP head to a 3D vector space. The output is then normalized to obtain the unit direction vector: $\mathbf{s}_\ell^X = \frac{\mathbf{v}}{\|\mathbf{v}\|}$.

The model is optimized by minimizing the negative cosine similarity between predicted and ground-truth semantic orientations. The training objective is formally defined as:

$$\mathcal{L}_{\cos}(\mathbf{v}, \mathbf{k}) = 1 - \frac{|\mathbf{v} \cdot \mathbf{k}|}{\|\mathbf{v}\| \cdot \|\mathbf{k}\|} \tag{11}$$

$$\min_{\theta_{\text{so}}} \sum_{X_i \in \mathcal{D}} \sum_{\ell_j^i \in L_i} \mathcal{L}_{\cos} \left(\mathcal{F}_{\text{SO}}(X_i, \ell_j^i), \mathbf{s}_j^i \right) \tag{12}$$

where $\mathcal{D}$ represents the training dataset and L_i denotes the set of language descriptions associated with object X_i.

A.2. Integration in GeoDexGrasp

In our framework, PointSO serves as the **Interaction Direction Extractor**. Given an input point cloud X and fixed language prompts $l \in \{\text{"forward"}, \text{"top"}, \text{"handle"}\}$, we obtain semantic orientation vectors:

$$d_l^X = \text{PointSO}(X, l) \in \mathbb{R}^3 \tag{13}$$

The complete interaction direction representation is:

$$\Theta_d = [d_1^X, d_2^X, d_3^X] \in \mathbb{R}^{3\times 3} \tag{14}$$

These direction vectors capture the object's functional semantics and guide the root rotation generation in our pose-guided grasping pipeline.

B. Implement Details

B.1. Loss Function Designs

In the GeoDexGrasp framework, we introduce two geometric constraint loss functions to ensure the generated dexterous grasp poses are physically plausible and stable: the **Contact Encouragement Loss** (L_c) and the **Penetration Penalty Loss** (L_p). These losses[42] are integrated into the diffusion model training to explicitly model hand-object interactions, guiding the generation of grasps with sufficient contact and minimal penetration.

The Contact Encouragement Loss L_c encourages the inner surface of the hand (phalanges and palm) to approach the object surface for stable contact formation. We sample a point cloud $P_{\text{hand}} \in \mathbb{R}^{M\times 3}$ from the hand mesh and compute the minimum Euclidean distance for each hand point p_i to the object point cloud $P_{\text{obj}} \in \mathbb{R}^{N\times 3}$ as $d_i = \min_j \|p_i - p_j\|_2$, where $p_j \in P_{\text{obj}}$. The loss is then defined as:

$$L_c = \frac{1}{|S| + \eta} \sum_{i \in S} \sqrt{d_i} \tag{15}$$

where $S = \{i \mid d_i < d_{\text{threshold}}\}$ denotes the set of hand points within a distance threshold $d_{\text{threshold}}$, $|S|$ is the cardinality of S, and η is a small constant for numerical stability.

The Penetration Penalty Loss L_p prevents geometric interpenetration between the hand and object. Using the object's surface normals $N_{\text{obj}} \in \mathbb{R}^{N\times 3}$, we compute the signed distance for each hand point as:

$$s_i = \text{sign}\left((p_j - p_i) \cdot n_j\right) \cdot d_i \tag{16}$$

where n_j is the normal vector at the nearest object point p_j. The loss aggregates negative signed distances to penalize penetration:

$$L_p = \frac{1}{B} \sum_{i=1}^{B} \max(0, -s_i) \tag{17}$$

where B is the batch size.

These losses are combined with the diffusion model's noise prediction loss L_{MSE} through the final training objective:

$$L = L_{\text{MSE}} + \eta L_c + \tau L_p \tag{18}$$

where η and τ are hyperparameters balancing the geometric constraints. This formulation enhances physical rationality by ensuring contact sufficiency and geometric feasibility, leading to robust and generalizable grasp generation.

B.2. Training Details

B.2.1. SIM(3) Encoder Extraction Module Hyperparameters

The feature extraction module comprises two complementary components: the SIM(3)-equivariant backbone and the PointSO orientation foundation model. Both components employ distinct architectures and training strategies tailored to their respective functionalities.

Table 4. Hyperparameters for SIM(3)-equivariant backbone

Parameter	Value
Architecture Type	VN-Transformer
Latent Dimension (c_{dim})	32
Output Dimension (d_{out})	256
Pooling Strategy	Mean
Pre-training Method	Self-supervised
Batch Size	64

The geometric feature extraction employs a Vector Neuron PointNet (VN-PointNet) architecture that ensures SIM(3)-equivariance, providing consistent geometric representations under scaling, rotation, and translation transformations. The network configuration utilizes a latent dimension of 32 and produces 256-dimensional output features. Mean pooling aggregates point-wise features into global descriptors, ensuring permutation invariance while preserving geometric structure.

The self-supervised disentangled pre-training follows a dual-branch approach: a pose estimation branch regresses rotation parameters from equivariant features $\Theta_e \in \mathbb{R}^{C\times 3}$, while a shape reconstruction branch reconstructs complete point clouds from invariant features $\Theta_i \in \mathbb{R}^C$. This formulation encourages the separation of pose and shape semantics within the latent space.

For interaction direction inference, we leverage PointSO, a large-scale pre-trained orientation foundation model. The

Table 5. Hyperparameters for PointSO orientation model

Parameter	Value
Architecture Type	Transformer-based
Embedding Dimension	384
Network Depth	12 layers
Number of Attention Heads	6
Drop Path Rate	0.2
Group Size	32
Number of Groups	512
Text Encoder	ViT-B/32
Pre-training Epochs	300
Batch Size	256
Number of Points	10,000

transformer-based architecture with 12 layers and 6 attention heads processes 10,000 points per object. The model employs a drop path rate of 0.2 for regularization and uses ViT-B/32 as the text encoder for language-conditioned orientation prediction.

All feature extraction components operate in a frozen state during the main grasping training phase, ensuring stable geometric priors while efficiently leveraging pre-trained knowledge for improved data efficiency.

B.2.2. Implict PDF Module Hyperparameters

Table 6. Hyperparameters for IPDF rotation generation model

Parameter	Value
Point Cloud Encoder	PN-RotationNet
Fourier Components	1
MLP Layer Sizes	[256, 256, 256]
Training Queries per Step	4096
Learning Rate	0.001
Learning Rate Policy	Step
LR Decay Factor (γ)	0.5
LR Step Size (epochs)	10
Minimum Learning Rate	1×10^{-5}
Weight Decay	1×10^{-5}

The rotation generation module employs an Implicit Probability Density Function (IPDF) model defined on the SO(3) manifold to learn the conditional distribution $p(R|\xi)$ of root rotations. The architecture utilizes a PointNet-based rotation encoder (PN-RotationNet) to process pose-related geometric representations $\xi = \{\Theta_e, \Theta_d\}$, comprising equivariant features and interaction directions.

The implicit model configuration incorporates 1 Fourier component for frequency embedding and a 3-layer MLP with 256 hidden units per layer to approximate the probability density over rotation space. During training, 4096 rotation queries are sampled per optimization step to efficiently explore the SO(3) manifold. The model is optimized using negative log-likelihood minimization of ground-truth rotations $\mathcal{L} = -\log(p(R_{gt}|\xi))$.

The training schedule employs step-based learning rate decay, reducing the rate by factor 0.5 every 10 epochs with a minimum learning rate threshold of 1×10^{-5}. A conservative weight decay of 1×10^{-5} provides mild regularization while maintaining model capacity for capturing complex rotational distributions. This configuration enables effective learning of semantically meaningful rotation distributions conditioned on object geometry and interaction directions.

B.2.3. Diffusion Module Hyperparameters

Table 7. Hyperparameters for diffusion-based grasp generation

Parameter	Value
Diffusion Steps	100
Noise Schedule	Linear
Beta Range	[0.0001, 0.01]
Loss Type	L1
Learning Rate	1×10^{-4}
Batch Size	128
Training Epochs	3000
Gradient Clipping	[-0.1, 0.1]
Guidance Scale	1.0
Gradient Scale	0.1
Point Cloud Size	2048
Sampling Method	DPM-Solver++
Sampling Steps	10
Sampling Order	1

The shape-guided grasp generation employs a diffusion model to learn the conditional distribution $p(h|\tilde{X}, \zeta)$ of hand motions and root translations in Euclidean space, where $h = (\theta, R^{-1}t) \in \mathbb{R}^{K+3}$. The diffusion process is configured with 100 steps using a linear noise schedule with β ranging from 0.0001 to 0.01.

The training utilizes an L1 loss for noise prediction and operates with a substantial 3000-epoch training schedule to ensure convergence. A batch size of 128 balances memory constraints with gradient stability, while gradient clipping within $[-0.1, 0.1]$ prevents explosion during backpropagation. The model processes 2048-point clouds with rotation-invariant geometric representations $\zeta = \{\Theta_i, \Theta_s\}$ as conditioning variables.We further optimize this procedure with a linear schedule, that is, the weight of physical constrains starts from 0 and linearly increases to 1 as the number of epochs grows. This change helps our model can get faster rate of convergence and more stable loss curve.

C. Model Introductions

C.1. Introduction of IPDF

This appendix provides a comprehensive description of the Implicit Probability Density Field (IPDF) method [29] for representing non-parametric distributions over the rotation manifold $\mathbf{SO}(3)$. The core innovation of IPDF lies in its implicit representation of probability distributions via a neural network, enabling the modeling of complex, multi-modal densities arising from object symmetries, self-occlusion, and viewpoint ambiguity.

C.1.1. Theoretical Formulation

The goal is to model the conditional probability distribution $p(R|x)$ of a rotation $R \in \mathbf{SO}(3)$ given an input $x \in \mathcal{X}$. IPDF achieves this through a neural network $f : \mathcal{X} \times \mathbf{SO}(3) \mapsto \mathbb{R}$ that estimates unnormalized joint log-probabilities:

$$\log p(x, R) = f(x, R) + \log \alpha, \tag{19}$$

where α is an intractable normalization constant. Applying Bayes' rule and marginalizing over $\mathbf{SO}(3)$ gives:

$$p(R|x) = \frac{p(x, R)}{p(x)} = \frac{\exp(f(x, R))}{\int_{R'} \exp(f(x, R'))dR'}. \tag{20}$$

The normalization integral over $\mathbf{SO}(3)$ is approximated via discrete summation over an equivolumetric partition $\{R_i\}_{i=1}^N$:

$$\int \exp(f(x, R'))dR' \approx V \sum_{i=1}^{N} \exp(f(x, R_i)), \tag{21}$$

where $V = \pi^2/N$ is the volume per partition element. The conditional probability becomes:

$$p(R|x) \approx \frac{\exp(f(x, R))}{V \sum_{i=1}^{N} \exp(f(x, R_i))}. \tag{22}$$

This formulation enables probability evaluation at any continuous rotation while requiring only discrete sampling for normalization.

C.1.2. Network Architecture and Implementation

The implicit function $f(x, R)$ is parameterized by a multilayer perceptron (MLP) that accepts two inputs: a visual descriptor extracted from x using a pre-trained ResNet, and a query rotation R. We represent rotations as 3×3 matrices to avoid discontinuities present in other parameterizations. Following recent advances in implicit neural representations, we apply positional encoding to the elements of the rotation matrix to enhance the network's capacity to represent high-frequency details:

$$\gamma(r) = [\sin(2^0\pi r), \cos(2^0\pi r), \ldots, \cos(2^{m-1}\pi r)], \tag{23}$$

where r is an element of the flattened rotation matrix and m denotes the number of encoding frequencies. The encoded rotation is concatenated with the image descriptor and processed by the MLP to produce a scalar log-probability value.

The model is trained to maximize the likelihood of ground truth rotations (x, R_0) from the training set. The loss function is the negative log-likelihood:

$$\mathcal{L}(x, R_0) = -\log p(R_0|x), \tag{24}$$

where $p(R_0|x)$ is computed using the normalization approximation described above. During training, we employ a sampling strategy where the set $\{R_i\}$ includes the ground truth R_0 to ensure stable optimization. While an equivolumetric grid provides exact normalization, we found that random uniform sampling of rotations during training yields comparable performance while being computationally efficient.

C.1.3. Inference Procedures

IPDF supports two inference modes: single pose estimation and full distribution extraction. For single pose prediction, we solve:

$$R^* = \underset{R \in \mathbf{SO}(3)}{\arg\max}\, f(x, R) \tag{25}$$

via gradient ascent in the continuous rotation space. The optimization is initialized from the most promising candidate in a coarse equivolumetric grid, with projections back to $\mathbf{SO}(3)$ after each gradient step. For full distribution estimation, we evaluate $p(R_i|x)$ over a fine equivolumetric grid, enabling comprehensive uncertainty quantification and symmetry analysis.

The IPDF framework naturally extends to full 6-DoF pose estimation (rotation and translation) by augmenting the query with translation coordinates and sampling over the product space $\mathbf{SO}(3) \times \mathbb{R}^3$. Normalization proceeds similarly using a Cartesian grid for translation and an equivolumetric grid for rotation.Implicit-PDF provides a flexible, non-parametric approach to representing probability distributions on $\mathbf{SO}(3)$ that offers several advantages: (1) it requires no assumptions about distribution form or object symmetries during training; (2) it learns from single ground truth poses per example without uncertainty supervision; (3) it can represent arbitrary multi-modal distributions; and (4) it supports both point estimation and full distribution extraction. These properties make IPDF particularly suitable for pose estimation under symmetry, occlusion, and other sources of ambiguity, while maintaining competitive performance on standard pose estimation benchmarks.

In our framework, we first pre-define a rotation query grid covering the omnidirectional space, which consists of 4,096 discrete anchors uniformly sampled from the SO(3) space. Subsequently, we employ the network to predict the response intensity at each grid point. By applying Softmax normalization and volume element weighting, these responses are converted into a valid rotation probability density distribution. Finally, we calculate the cumulative probabilities based on this distribution and perform inverse transform sampling. This strategy ensures that rotation regions with high probability density are more likely to be selected, while simultaneously allowing the model to explore suboptimal solutions in the presence of ambiguity, thereby robustly representing rotation uncertainty. As illustrated in Figure 8, we visualize the rotation sampling results.

C.2. Diffusion Models for Dexterous Grasp Generation

C.2.1. Denoising Diffusion Probabilistic Models

The forward process is a fixed Markov chain that gradually adds Gaussian noise to the data. Given a data sample $\mathbf{h}_0 \sim q(\mathbf{h}_0)$ from the true data distribution, we define the forward process for $t = 1, \ldots, T$ as:

$$q(\mathbf{h}_t|\mathbf{h}_{t-1}) = \mathcal{N}(\mathbf{h}_t; \sqrt{1-\beta_t}\mathbf{h}_{t-1}, \beta_t\mathbf{I}) \tag{26}$$

where $\beta_t \in (0, 1)$ is a variance schedule that controls the amount of noise added at each step. The key insight is that we can directly sample $\mathbf{h}_t$ at any arbitrary timestep t without going through the entire Markov chain:

$$q(\mathbf{h}_t|\mathbf{h}_0) = \mathcal{N}(\mathbf{h}_t; \sqrt{\bar{\alpha}_t}\mathbf{h}_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{27}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. This allows us to express $\mathbf{h}_t$ directly as:

$$\mathbf{h}_t = \sqrt{\bar{\alpha}_t}\mathbf{h}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}) \tag{28}$$

As $t \to T$, $\bar{\alpha}_t \to 0$ and $q(\mathbf{h}_T|\mathbf{h}_0)$ approaches a standard Gaussian distribution $\mathcal{N}(0, \mathbf{I})$.

The reverse process aims to recover the original data distribution by learning to reverse the diffusion process. Starting from pure noise $\mathbf{h}_T \sim \mathcal{N}(0, \mathbf{I})$, we define the reverse process as a parameterized Markov chain:

$$p_\theta(\mathbf{h}_{0:T}) = p(\mathbf{h}_T) \prod_{t=1}^{T} p_\theta(\mathbf{h}_{t-1}|\mathbf{h}_t) \tag{29}$$

$$p_\theta(\mathbf{h}_{t-1}|\mathbf{h}_t) = \mathcal{N}(\mathbf{h}_{t-1}; \mu_\theta(\mathbf{h}_t, t), \Sigma_\theta(\mathbf{h}_t, t)) \tag{30}$$

where μ_θ and Σ_θ are neural networks that learn the mean and covariance of the reverse transition.

C.2.2. Conditional Process

The conditional forward process remains the same as the unconditional case:

$$q(\mathbf{h}_t|\mathbf{h}_{t-1}, O) = \mathcal{N}(\mathbf{h}_t; \sqrt{1-\beta_t}\mathbf{h}_{t-1}, \beta_t\mathbf{I}) \tag{31}$$

The training objective extends naturally to the conditional case:

$$\mathcal{L}_{\text{simple}}^{\text{cond}} = \mathbb{E}_{t,\mathbf{h}_0,O,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{h}_t, O, t)\|^2\right] \tag{32}$$

where the noise prediction network ϵ_θ now takes the conditioning information O as an additional input.

The denoising process is conditioned on the object representation through a cross-attention mechanism. Specifically, we first modulate the shape representation Θ_{i} with the scale representation Θ_{s} using point-wise Hadamard product:

$$\Theta_{\text{mod}} = \Theta_{\text{i}} \odot \Theta_{\text{s}} \tag{33}$$

The modulated representation is then projected through a linear layer to predict the FiLM parameters:

$$[\beta, \gamma] = \mathbf{W}_{\text{film}} \cdot \Theta_{\text{mod}} + \mathbf{b}_{\text{film}} \tag{34}$$

where $\mathbf{W}_{\text{film}}$ and $\mathbf{b}_{\text{film}}$ are learnable parameters, β and γ are the affine transformation coefficients.

These coefficients are used to modulate the 3D features V extracted from the PointTransformer:

$$V_{\text{condition}} = \gamma \odot V + \beta \tag{35}$$

The conditioned features $V_{\text{condition}}$ serve as the key and value in the cross-attention mechanism:

$$Q = \mathbf{W}_Q \cdot \phi(h_t, t) \tag{36}$$

$$K = \mathbf{W}_K \cdot V_{\text{condition}} \tag{37}$$

$$V = \mathbf{W}_V \cdot V_{\text{condition}} \tag{38}$$

where $\phi(h_t, t)$ represents the temporal embedding of the noisy hand pose at diffusion step t, and $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$ are learnable projection matrices.

The cross-attention is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{39}$$

This design enables the diffusion model to effectively incorporate both shape and scale information from the object representation throughout the denoising process, ensuring that the generated hand poses are geometrically consistent with the target object.

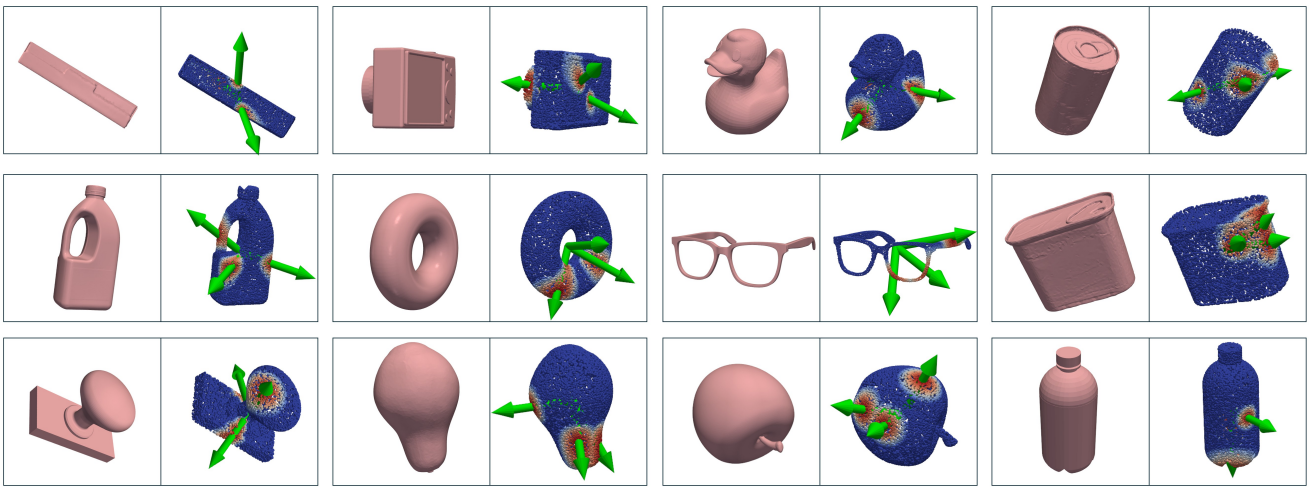

Figure 8. For each object, we draw ten random samples from the probability distribution and cluster them into three distinct directions. This allows us to represent potential multi-modal end-effector orientations.

D. Evaluation Results

D.1. Grasp Diversity Experiments

Diversity in grasping tasks plays a crucial role in enabling robust and adaptable robotic manipulation systems. As demonstrated in our experiments, a diverse set of grasping poses allows robots to handle various object geometries, environmental constraints, and task requirements effectively. High diversity ensures that the system can generate multiple viable grasping strategies for the same object, providing flexibility when certain grasps are infeasible due to obstacles, workspace limitations, or specific manipulation goals. Moreover, diverse grasping distributions enhance the system's generalization capability across unseen objects, as they capture a broader spectrum of successful interaction patterns. The mean standard deviation can be formulated as below:

$$\text{Div} = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left\|\mathbf{p}_{i,k}-\bar{\mathbf{p}}_k\right\|^2} \tag{40}$$

Our method achieved the highest or second diversity scores on five datasets: DexGRAB, RealDex, MultiDex, UniDexGrasp, and DexGraspNet, with particularly excellent performance on the MultiDex dataset reaching 0.268 as shown in Table 9. Although slightly lower than the DGA method on the RealDex dataset, overall, our method demonstrates significant advantages in grasping diversity, proving its effectiveness in generating diverse grasping poses.

Meanwhile, we also tested rotational diversity as Table 9 following the approach of [42]. Since the DexGraspAnything model does not include predictions for rotations, its rotational diversity is 0. That's to say, regardless of the initial pose of the object, DexGraspAnything can only output root-rotation-invariant actions, meaning it is limited to horizontal grasps and cannot adapt to grasping from other angles. This substantially undermines its practicality in real-world scenarios, and a noticeable decline in performance was observed under random rotation tests. In contrast, the UGG method incorporated pose deviations caused by object rotation during training, effectively implementing data augmentation, which resulted in a relatively minor performance decrease. Our proposed method introduces geometric representations that provide effective shape and scale information for the grasp generation model, thereby enhancing its ability to perceive object boundaries. Furthermore, through the first-stage semantic direction features and the second-stage IPDF model, our method implicitly establishes a correspondence between object orientation and hand pose. This design enables an equivariant response to inputs with different rotations, allowing the model to adapt its output predictions based on rotational changes, thereby reducing reliance on large-scale datasets. Consequently, our method achieves superior Out-of-Domain performance for these objects.

Table 8. Joint Angles Diversity Comparison across Different Methods and Datasets(cm)

Method	GRAB	RealD	MultiD	UDG	DGN
UGG	0.12	0.10	0.12	0.14	0.14
DGA	0.13	**0.14**	0.22	**0.23**	**0.23**
Ours	**0.15**	0.11	**0.27**	0.22	0.21

Table 9. Rotation Diversity Comparison across Different Methods and Datasets(degree)

Method	GRAB	RealD	MultiD	UDG	DGN
UGG	86.4	72.6	93.4	105.6	95.8
DGA	0	0	0	0	0
Ours	**124.2**	**95.1**	**121.9**	**110.3**	**112.2**

D.2. Rotation Generalization Experiments

In this section, we mainly compare two methods' adaptation ability towards rotation generalization using diffusion models, including DexGraspAnything[52] and UGG[27]. These approaches have adopted different definitions and processing methods for the motion space of dexterous hands when addressing grasping problems. DexGraspAnything constrains two degrees of freedom (DOFs) of the wrist and assumes that the object's pose is known during grasping; thus, the input to the diffusion model is 3+22 dimensions. UGG also constrains two DOFs of the wrist but comprehensively considers rotation, translation, and the positions of end joints.First, the 22 joints of the dexterous hand are encoded into the latent space via a VAE encoder, and then the latent variables, along with translation and rotation parameters, are fed into the diffusion model for noise prediction. Finally, all DOFs are output through the decoder.Our work decouples wrist rotation from joint motions and refines the process in a two-stage manner during motion execution.

During the experimental process, we kept the original training models unchanged and only applied unknown random rotations to the objects during testing to evaluate the models' generalization capability to unseen object poses. The experimental results are presented in the Table 10 and Fig 9 below, the last column represents the overall average success rates of these methods on the multidex dataset (including both the original dataset and after applying random rotations). Analysis of the results indicates that after applying random rotations, all models exhibited some degree of performance degradation. Specifically, the DGA method, which did not account for object rotation during training, relies entirely on the static distribution of the training dataset for diversity, leading to significant limitations in the freedom and diversity of root rotation as Fig 10.

We observed that the models' performance under random rotations varies depending on the inherent properties of different objects. For the highly symmetrical baseball, whether it undergoes rotation or not is nearly identical from the model's perspective, resulting in almost identical success rates. However, the other four test cases exhibit noticeable changes in grasping patterns after rotation, leading to varying degrees of performance degradation. Leveraging geometry cues and implicit modeling, our model demonstrates the most stable performance among all compared methods.

E. More Visualizations

More visualizations of the generated grasping poses by our methods on various objects are presented in Figure 8 and Figure12, our models produce more sound grasping poses for complex objects.

F. Discussions about Rotations

For scenarios where rotation and hand joint parameters are not decoupled, we conducted additional experiments to investigate the impact of end-effector rotation space size in SO(3) on grasp generation. As shown in Fig 11, we edited DexgraspAnything[52] to generate hand motions incorporating root rotations, and performed three comparative experiments by restricting the root rotation ranges to 30°, 60°, and 180° respectively. The experimental results demonstrate that as the root rotation space expands, the model struggles to converge to low noise error values.

From the theoretical perspective of diffusion models, the training objective is to minimize the error between predicted and actual noise. When the root rotation range expands from 30° to 180°, the volume of the SO(3) space increases significantly. As a three-dimensional rotation group, SO(3) possesses a non-Euclidean geometric structure whose complexity intensifies with expanding rotation ranges. Within smaller rotation ranges (e.g., 30°), the output space remains relatively compact with concentrated data distribution, enabling the model to readily learn a straightforward denoising mapping. However, when the rotation range expands to 180°, output space covers a wider range of possible rotational poses, resulting in a more dispersed and multimodal data distribution that increases the difficulty for the model to capture the complete distribution. Consequently, denoising network must handle broader variations, making it more challenging to reduce noise error value during training.

We argue that with larger root rotation spaces, the model needs to generalize to more rotational poses from limited data. Diffusion models rely on training data covering all modes of the output space, but if the rotation range becomes excessively large while the training data fails to adequately cover all possible orientations, the model tends to underfit. In our experiments, when the rotation range expanded to 180°, the model potentially failed to learn effective rotation invariance from the data, and the complexity of distribution modeling caused the optimization process to converge to local minima, maintaining high noise error values. We anticipate that future research could better address the inherent inconsistency between learning objectives on the SO(3) manifold and variable patterns in Euclidean space through novel architectural designs, thereby promoting more unified dexterous hand grasp motion generation frameworks.

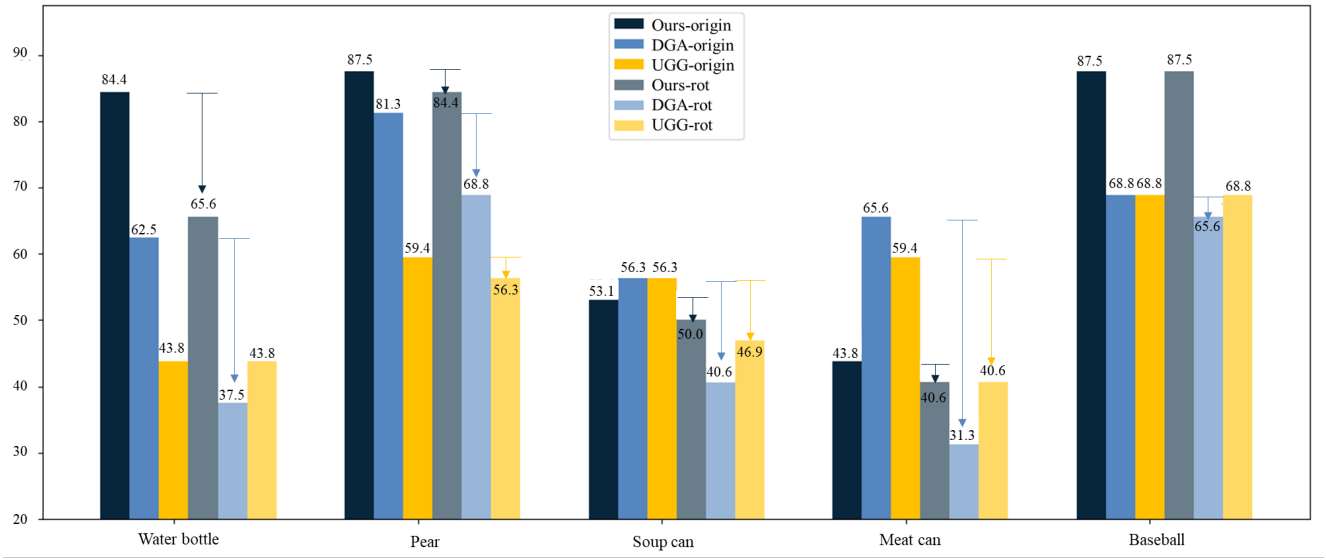

Figure 9. We tested the performance of these methods on the multidex dataset, calculating the grasping success rates for five types of examples both before and after applying rotation. It can be observed that our model demonstrates relatively stable performance when facing rotations.

Table 10. Performance comparison of dexterous grasp methods (Rot. columns with light gray background).

Method \ Object	Water Bottle		Pear		Soup Can		Meat Can		Baseball		Suc.	
	origin	Rot.	origin	Rot.	origin	Rot.	origin	Rot.	origin	Rot.	origin	Rot.
UGG[27]	43.8	43.8	59.4	56.3	**56.3**	46.9	59.4	40.6	68.8	68.8	55.3	42.7
DexGraspAnything[52]	62.5	37.5	81.3	68.8	**56.3**	40.6	**65.6**	31.3	68.8	65.6	79.1	51.9
GeoDexGrasp (Ours)	**84.4**	**65.6**	**87.5**	**84.4**	53.1	**50.0**	43.8	**40.6**	**87.5**	**87.5**	**81.2**	**65.6**

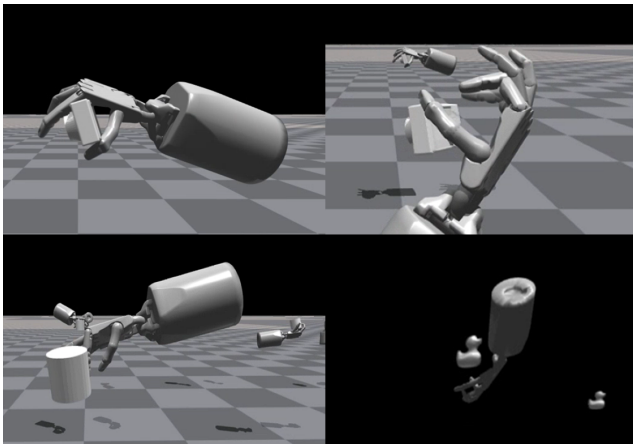

Figure 10. When objects are subjected to rotation, methods like DexGraspAnything exhibit significant performance degradation due to the difficulty in adapting to the new poses of objects.

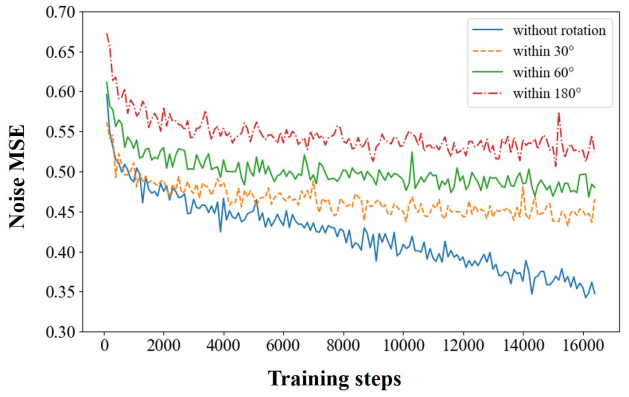

Figure 11. Apply root rotations to target objects, and then constrain the rotation range to 30°, 60°, and 180° in three comparative experiments.Results indicate that as the rotation space expands, the model becomes increasingly difficult to converge to a low noise error.

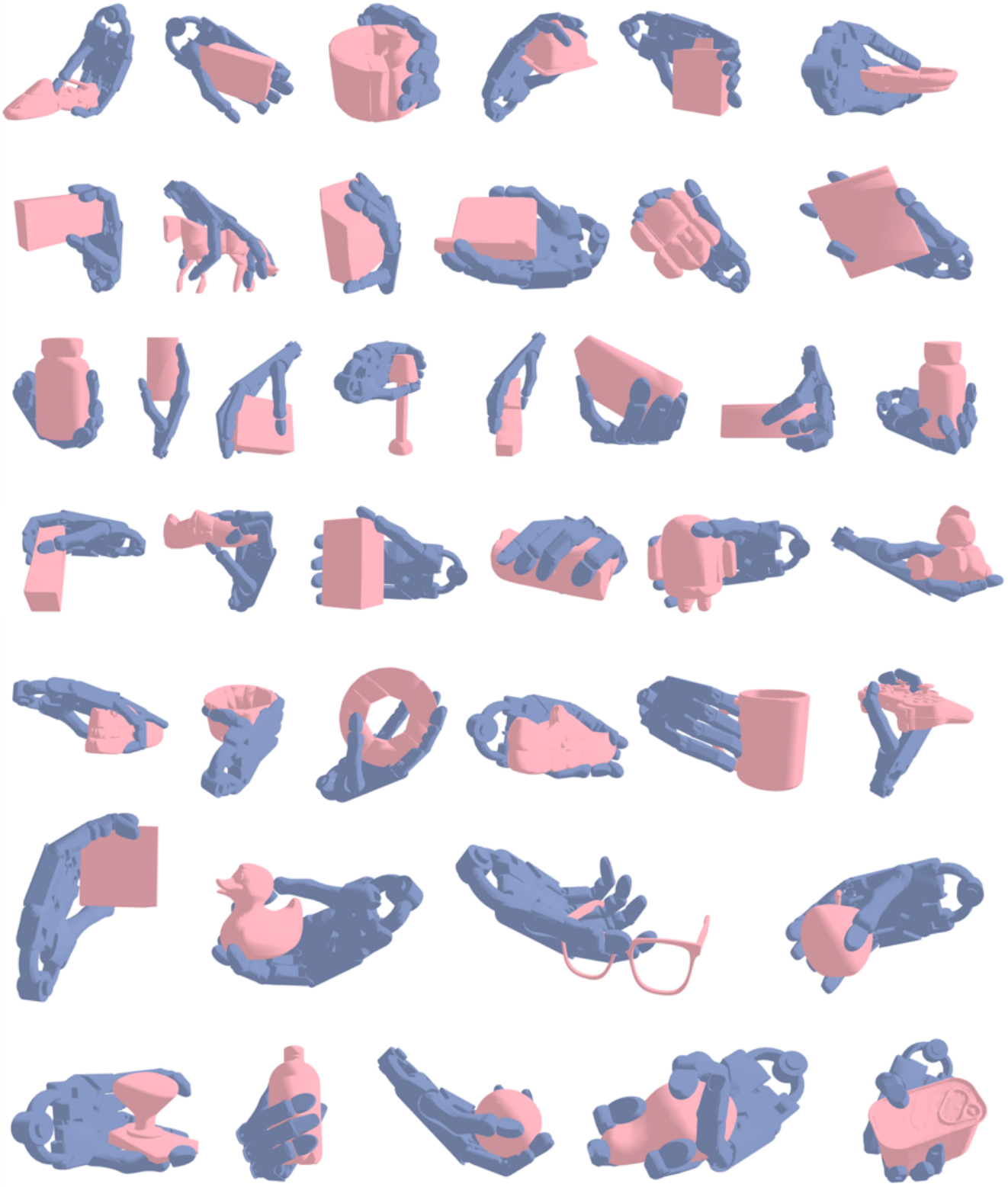

Figure 12. Our model's grasp results.