Single-View Scene Point Cloud Human Grasp Generation

Supplementary Material

1. Background for Diffusion Model

A diffusion model, in the context of machine learning and particularly in generative modeling, is a type of model that generates data by reversing a diffusion process. This process gradually adds noise to the data, and then learns to reverse this process to generate new data.

The forward process is a Markov Chain that gradually adds Gaussian noise to the data over a series of time steps. If we start with data x_0, after T time steps, the data becomes a sample from a standard Gaussian distribution, x_T, shown as,

$$\begin{cases} x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}, \\ x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \end{cases} \tag{1}$$

where α is a noise coefficient based on time t, $\bar{\alpha}_t = \prod_{i=1}^{t} \bar{\alpha}_i$, and ϵ is noise sampled from $\mathcal{N}(0, I)$. The formula in the top line represents the single-step diffusion process, while the formula in the bottom line allows for approximating the result of any step, i.e., x_t, from x_0 in one step.

The reverse process trains a model to approximate the reverse conditional distribution $p(x_{t-1}|x_t)$, shown as,

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta^2(x_t, t)), \tag{2}$$

where μ and σ^2 is the mean and variance of the distribution x_{t-1}, θ is the model predicting ϵ from x_t.

DDPM [3] and DDIM [6] are two sampling strategies in reverse process. For DDPM, it learns to reverse the noise addition process step by step to reconstruct the original data from the noise. The model learns the distribution $p(x_{t-1}|x_t$ and predicts the noise ϵ, shown as,

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

$$= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)\right), \tag{4}$$

$$x_{t-1} = \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right) + \sigma_\theta(x_t, t)z, \tag{5}$$

where σ_θ is untrained time dependent constants and z is the random Guassian noise.

DDIM simplifies the sampling process of DDPM and allows for faster sampling and control over the generation process. Specifically, DDIM introduces a skip-step sampling strategy that significantly reduces the sampling time of diffusion, and it is proven that this strategy does not affect the accuracy of the diffusion results. For instance, in a process where DDPM requires 1000 sampling steps, DDIM can achieve similar results with just five sampling steps at intervals, such as 1000, 800, 600, 400, and 200. The process can be shown as,

$$x_{t-1} = \sqrt{\overline{\alpha}_{t-1}}x_0 + \sqrt{1-\overline{\alpha}_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^{(t)}(x_t) + \sigma_t\epsilon_t, \tag{6}$$

where $\sigma_t = \eta\sqrt{\frac{1-\overline{\alpha}_{t-1}}{1-\overline{\alpha}_t}\cdot\left(1-\frac{\overline{\alpha}_t}{\overline{\alpha}_{t-1}}\right)}$, η is the DDIM [6] sampling coefficient and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is standard Gaussian noise independent of x_t.

Conditional Diffusion Model The conditional diffusion model learns a conditional distribution $p_\theta(\boldsymbol{x}_0|\boldsymbol{c})$. The process can be shown as,

$$p_\theta\left(\boldsymbol{x}_0|\boldsymbol{c}\right) = p\left(\boldsymbol{x}_T\right)\prod_{t=1}^{T} p\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}\right), \tag{7}$$

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{c}) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t, \boldsymbol{c}), \sum_\theta(\boldsymbol{x}_t, t, \boldsymbol{c})) \tag{8}$$

2. Whole Process and Evaluation Metrics

2.1. Whole Process of Our S2HGrasp

Here we give a detailed explanation of the whole process of our model.

During training process, as Algorithm 1 shows, our model takes single-view scene point clouds $\mathcal{P}_s$ along with a corresponding human grasp label $\mathcal{H}_0$ in MANO format [5] as input. $\mathcal{P}_s$ and $\mathcal{H}_0$ are fed into the Global Perception module and the DiffuGrasp module, respectively. In the Global Perception, we get the features $\mathcal{F}_s$ of single-view scene point clouds through a scene encoder and $\mathcal{F}_s$ will act as the condition in DiffuGrasp. Then we pass $\mathcal{F}_s$ through the GSP and the GCP for point cloud completion and single-view point cloud classification tasks, respectively. This enables the model to have global perception of partial objects. After training and supervision in both the point cloud completion and classification tasks, features $\mathcal{F}_s$ now roughly captures the global characteristics of the object and we use them as a condition for the DiffuGrasp module. In the DiffuGrasp module, we add Gaussian noise to the normalised hand parameters and feed hand features $\mathcal{F}_h$ and scene features $\mathcal{F}_s$ into a transformer decoder to predict the original hand parameters $\mathcal{H}_0'$.

During testing process, as Algorithm 2 shows, the input is single-view scene point clouds $\mathcal{P}_s$. Similarly, the point clouds go through the scene encoder to obtain features $\mathcal{F}_s$. Our model randomly samples hand parameters $\mathcal{H}_T$ from a Gaussian distribution, and then, using hand features $\mathcal{F}_h$ and scene features $\mathcal{F}_s$ as condition, predicts the original hand parameters $\mathcal{H}_0'$. We employ a skip-step denoising strategy similar to DDIM [6] to progressively generate reasonable

Algorithm 1 Training

Input: noise schedule α, extracted object feature $\mathcal{F}_o$, hand parameters $\mathcal{H}_0$, denoising transformer decoder θ, normalising function Norm, de-normalising function $\mathrm{de\text{-}Norm}$.

repeat
  Sample $t \sim Uniform(1, ..., T)$
  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$
  $\mathrm{H}_{0n} = \mathrm{Norm}(\mathcal{H}_0)$
  $\mathcal{H}_{tn} = \sqrt{\bar{\alpha}_t}\mathcal{H}_{on} + \sqrt{1-\bar{\alpha}_t}\epsilon$
  $\mathcal{H}_t = \mathrm{de\text{-}Norm}(\mathrm{H}_{tn})$
  $\mathcal{F}_h = \mathrm{Encoder}(\mathrm{MANO}(\mathcal{H}_t))$
  $\mathcal{H}'_0 = \theta(\mathcal{F}_o, \mathcal{F}_h)$
  $\theta = \theta - \eta\nabla_\theta|\mathcal{H}'_0 - \mathcal{H}_0|$
until converged

Algorithm 2 Sampling

Input: noise schedule α, extracted object feature $\mathcal{F}_o$, denoising transformer decoder θ, normalising function Norm, de-normalising function $\mathrm{de\text{-}Norm}$.

Sample $\mathrm{H}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, ..., 1$ **do**
  $\mathcal{H}_t = \mathrm{de\text{-}Norm}(\mathrm{H}_t)$
  $\mathcal{F}_h^t = \mathrm{Encoder}(\mathrm{MANO}(\mathcal{H}_t))$
  $\mathcal{H}'_0 = \theta(\mathcal{F}_o, \mathcal{F}_h^t)$
  $\mathrm{H}'_0 = \mathrm{Norm}(\mathcal{H}'_0)$
  $\boldsymbol{\epsilon}_\theta^{(t)}(\mathrm{H}_t) = \frac{\sqrt{\frac{1}{\bar{\alpha}_t}}\cdot \mathrm{H}_t - \mathrm{H}_0}{\sqrt{\frac{1}{\bar{\alpha}_t}-1}}$
  $\sigma_t = \eta\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot(1-\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}})}$
  $c = \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}$
  Sample $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathrm{H}_{t-1} = \mathrm{H}'_0 \cdot \sqrt{\bar{\alpha}_{t-1}} + c\boldsymbol{\epsilon}_\theta^{(t)}(\mathrm{H}_t) + \sigma_t\boldsymbol{\epsilon}_t$
end for
$\mathcal{H}_0 = \mathrm{de\text{-}Norm}(\mathrm{H}_0)$
return $\mathcal{H}_0$

hand parameters. Note that the GSP and GCP in the Global Perception module will not be used in testing process.

2.2. Evaluation Metrics

Penetration. The penetration metric primarily consists of two parts: penetration depth and penetration volume. Penetration depth is determined by the maximum distance from the points on the hand within the object's mesh to the object's surface. For the penetration volume, we voxelize the hand and object meshes with a voxel size of 0.5 cm and calculate the intersection shared by the two 3D voxels.

Grasp displacement. The grasp displacement is used to indicate the stability of the hand grasping the object. To assess stability, we simulate the object and generated grasp in accordance with the procedure described in [2, 7]. We place the generated hand grasp and the object in a simulation environment, fixing the hand's position and posture. The simulator calculates the forces of the fingertips exerted on the object according to penetration volume and the object's self-gravity, and then these forces will be used to calculate the displacement of the object's center of mass. The forces on the fingertips have a positive correlation with the volume of penetration of the fingertips into the object. In simple terms, if the forces exerted by the fingertips on the object are strong enough to counteract the object's own gravity, then the grasp is considered stable, and the calculated displacement will be zero. However, if the forces exerted by the fingertips are not strong enough, the object will move under the influence of the combined effects of these forces. We use this displacement to measure the grasp stability, and compute the mean and variance of the displacement for all samples in the testing set, where smaller displacement indicates better grasp stability.

Perceptual score. Perceptual score is used to evaluate the naturalness of the generated grasps. We employ a manual scoring approach for all the test samples. Specifically, we present all the generated grasps along with their corresponding scene meshes and invite 50 participants to rate the naturalness of these grasps, using a scale from 1 to 5, where higher scores indicate higher naturalness.

Contact ratio. The contact ratio is used to analyze whether there is contact between the hand and the object. We primarily calculate the proportion of samples in the entire testing set where the hand and the object contact. This is done by measuring the distance from each point on the hand to the nearest point on the object. We then assess whether there are distances less than 0.5 centimeters in the sample, indicating contact between the hand and the object. Contact ratio provides an alternative perspective for evaluating the validity of generated grasps.

3. More Analysis Experiments

We ablate the Global Category Perception, combination of single-view point cloud and completion results and the plane loss. We also provide additional experiments for different time steps of DiffuGrasp, the selection of loss weights, applying the Global Perception to CVAE and comparison between different categories of objects.

3.1. The Global Category Perception

We also ablate the Global Category Perception (GCP) in the Global Perception module, as shown in Table 3. The results indicate that the GCP module, through the single-view scene point cloud classification task, helps the model to better perceive the overall shape of the object, thereby improving the stability of the grasp.

Time Step		4	5	8	10(ours)	12	15	20
Penetration	Depth(cm) ↓	0.21	0.20	0.21	0.21	0.20	0.20	0.20
	Volume(cm^3)↓	6.78	6.81	6.79	6.58	6.57	6.42	6.37
Grasp Displace	Mean(cm)↓	2.69	2.72	2.70	2.73	2.76	2.80	2.81
	Variance(cm)↓	3.12	3.23	3.14	3.16	3.16	3.19	3.18
Contact	Ratio(%)↑	99.47	99.51	99.48	99.41	99.42	99.31	99.26

Table 1. Ablation study of different time steps.

Category		Bottle	Bowl	Camera	Knife	Lotion pump	Mug	Trigger sprayer	Wineglass
Number		5625	862	1013	1060	551	10353	197	137
Penetration	Depth(cm) ↓	0.22	0.22	0.18	0.48	0.23	0.15	0.82	1.53
	Volume(cm^3)↓	6.47	7.94	11.01	3.42	5.66	6.49	6.61	3.54
Grasp Displace	Mean(cm)↓	2.68	3.99	3.65	1.64	2.05	2.74	2.24	1.48
	Variance(cm)↓	3.05	3.63	4.03	2.26	2.55	3.14	2.84	2.23
Contact	Ratio(%)↑	99.32	99.77	99.7	95.09	99.82	99.86	97.97	98.48

Table 2. Comparison between different categories of objects. "Number" presents the number of single-view scene point clouds for each category of objects.

Method	Penetration		Grasp Displace	Contact
	Depth↓	Volume↓	Mean±Variance↓	Ratio↑
w/o GCP	0.23	6.43	2.82±3.28	99.28
w/ GCP(ours)	0.21	6.58	**2.73±3.16**	**99.41**

Table 3. Ablation study of GCP.

Method	Penetration		Grasp Displace	Contact
	Depth↓	Volume↓	Mean±Variance↓	Ratio↑
w/o comp	0.25	9.93	3.37±3.64	99.34
w/ comp(ours)	**0.21**	**6.58**	**2.73±3.16**	**99.41**

Table 4. Ablation study of combination of single-view point cloud and completion results.

3.2. Combination of Single-view Point Cloud and Completion Results

As mentioned in the main part, during training, we combine the input single-view point clouds with the results of point cloud completion to form a new object for calculating the cmap loss and penetration loss. The results in Table 4 show that this method of combination effectively prevents the generated hand from penetrating into the invisible parts of the object and also enhances the stability of the grasp.

3.3. Plane Loss

As shown in Table 5, we ablate the role of our proposed plane loss. The results demonstrate that our proposed plane loss can slightly improve the quality of the grasp. However, this loss function is to prevent the generated grasp from contacting the tabletop. During the testing process, we observe that in all test samples, the generated hand does not have contact with the tabletop, which also confirms the effectiveness of this loss function.

Method	Penetration		Grasp Displace	Contact
	Depth↓	Volume↓	Mean±Variance↓	Ratio↑
w/o plane loss	0.21	6.71	2.87±3.31	99.15
w/ plane loss(ours)	0.21	**6.58**	**2.73±3.16**	**99.41**

Table 5. Ablation study of the plane loss.

Method	Penetration		Grasp Displace	Contact
	Depth↓	Volume↓	Mean±Variance↓	Ratio↑
w/ GP	0.24	**12.84**	**2.88±3.28**	**99.05**
w/o GP(GC)	0.23	13.54	3.10±3.49	98.07

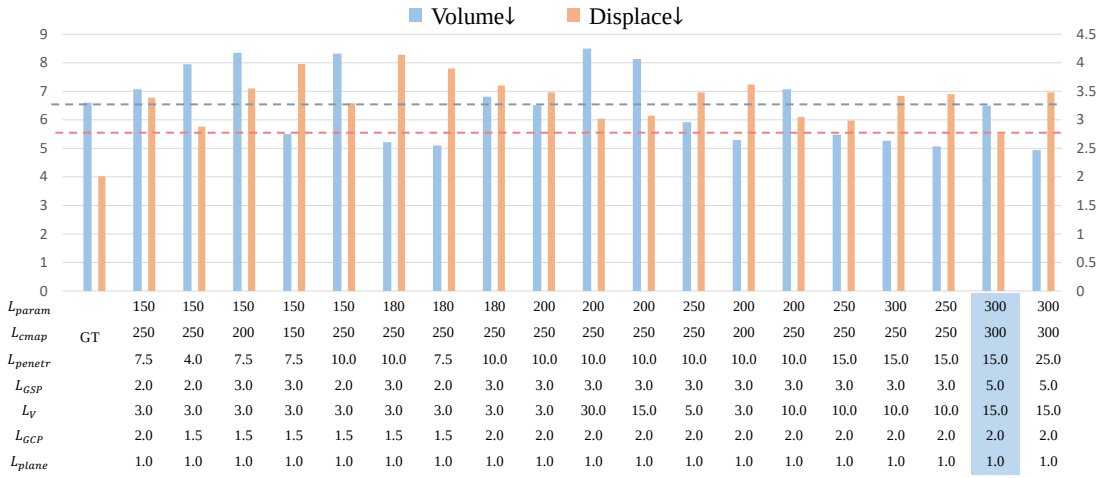
Table 6. Results of applying our Global Perception module (GP) to GraspCVAE.

3.4. Time Steps of DiffuGrasp

In Table 1, we study the different time steps between different time steps. The results show that using different time steps yields similar results and the number of time steps has a minimal impact on the results. Therefore, we select $T = 10$ for the best overall performance.

3.5. Selection of Loss Weights

Fig. 1 shows the selection of the loss weights. Since each loss function affects the others and evaluating grasp quality requires a comprehensive consideration of various metrics, analyzing a single loss function or evaluation metric is not meaningful. Therefore, after determining the approximate range of weights for each loss function, we considered all loss functions together and chose embedding volume and movement distance as the main indicators. In the table, we compared 20 sets of main loss weights, where the parameters highlighted with a blue background represent our selection for the best overall performance. Among these two indicators, we prioritized movement distance because ensuring grasp stability is paramount. As can be seen in



L_{param}	GT	150	150	150	150	150	180	180	180	200	200	200	250	200	200	250	300	250	300	300
L_{cmap}		250	250	200	150	250	250	250	250	250	250	250	250	200	250	250	250	250	300	300
L_{penetr}		7.5	4.0	7.5	7.5	10.0	10.0	7.5	10.0	10.0	10.0	10.0	10.0	10.0	10.0	15.0	15.0	15.0	15.0	25.0
L_{GSP}		2.0	2.0	3.0	3.0	2.0	3.0	2.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	5.0	5.0
L_V		3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	3.0	30.0	15.0	5.0	3.0	10.0	10.0	10.0	10.0	15.0	15.0
L_{GCP}		2.0	1.5	1.5	1.5	1.5	1.5	1.5	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
L_{plane}		1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Figure 1. Selection of loss weights. The weights highlighted with a blue background represent the best weights we select.

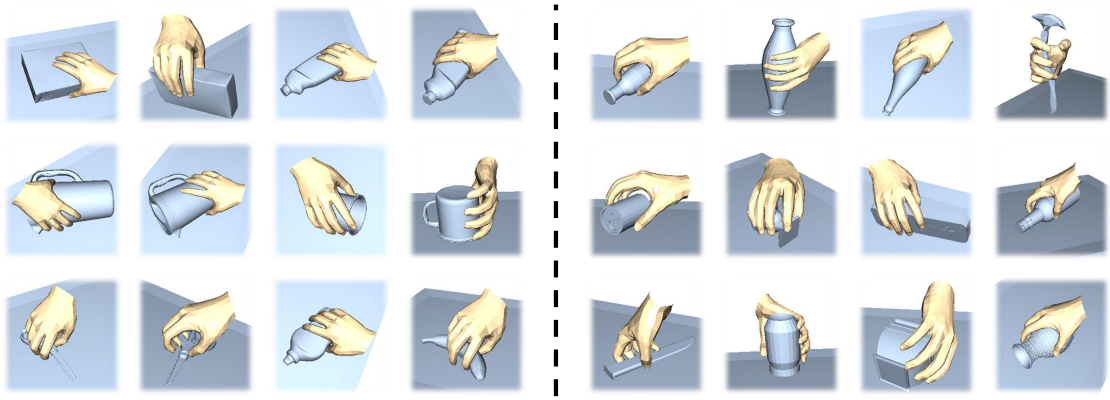


Figure 2. Visualization of generated grasps of testing sets constructed with HO3D [1] (left) and Obman[2] (right) objects.

the figure, the grasps obtained with our chosen parameters ensure stability while also minimizing hand-object embedding, indicating a high quality of the generated grasps.

3.6. Global Perception for CVAE

Table 6 shows the results of applying our Global Perception module to GraspCVAE [4]. The results demonstrate that applying our Global Perception module to GraspCVAE can also reduce the penetration between the hand and the object and enhance grasp stability. This confirms that our Global Perception module indeed improves the model's global perception ability. It is not only effective for our diffusion-based model but also beneficial for models based on CVAE.

3.7. Comparison between Different Categories

We select eight main object categories from our dataset and statistically analyze the grasp metrics, as shown in the Table 2. It is evident that different categories of objects have varying shapes and, consequently have different levels of grasping difficulty. For our model, the challenge posed by different object categories to the Global Perception module also varies. Overall, bottles and mugs present a lower difficulty, whereas cameras are more challenging. The primary reason is that the shapes of bottles and mugs are relatively simple, making it easier for the model to perceive their overall shape. In contrast, cameras have more complex shapes, making it harder to perceive their overall characteristics, leading to generated grasps that are more likely to penetrate into invisible parts of the object.

4. More visualization

4.1. Visualization of S2HGD Results

More visualization results of our S2HGrasp for S2HGD are shown in Fig. 3. The results on the left are for View-S2HGD and results on the right are for Object-S2HGD.

4.2. Visualization of HO3D and Obman Object Testing Sets

Additional visualization results of HO3D [1] (on the left) and Obman[2] (on the right) are shown in Fig. 2. We choose 10 objects from HO3D and 30 from Obman dataset to construct two testing sets. The results confirm that our S2HGrasp has a generalization capability for unseen objects.

Figure 3. Additional visualization of generated grasps of our S2HGrasp on our S2HGD datasets. The grasps on the left represent the results of View-S2HGD, while the grasps on the right represent the results of Object-S2HGD.

References

[1] Shreyas Hampali, Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020. 4

[2] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019. 2, 4

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 2020. 1

[4] Hanwen Jiang, Shaowei Liu, Jiashun Wang, and Xiaolong Wang. Hand-object contact consistency reasoning for human grasps generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 4

[5] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022. 1

[6] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1

[7] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 2016. 2