Supplementary Materials for "FFHFlow: Diverse and Uncertainty-Aware Dexterous Grasp Generation via Flow Variational Inference"

Contents

1 Derivation of Variational Lower Bound

To learn a probabilistic model $p_\theta(\mathbf{g}|\mathbf{x})$ parameterized by θ, we optimize it by maximizing its variational lower bound. Here $\mathbf{g}$ denotes grasp configuration, and $\mathbf{x}$ is a partially observed point cloud. Assume that the real posterior $p_\theta(z|x,g)$ is defined within the deep lantent variation model framework according to the Bayes formula: $p_\theta(z|x,g) = \frac{p_\theta(g|z,x)p_\theta(z|x)}{p_\theta(g|x)}$, where $p_\theta(g|x) = \int p_\theta(g|z,x)p_\theta(z|x)dz$ is the so-called model evidence. Based on Jensen Inequality, we can derive the variational lower bound with an approximate posterior of the latent variable $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{g})$ step by step with the following:

$$
\begin{aligned}
\log p_\theta(\mathbf{g}|\mathbf{x}) &= \log \int p_\theta(\mathbf{g},\mathbf{z}|\mathbf{x})d\mathbf{z} \\
&= \log \int p_\theta(\mathbf{g}|\mathbf{z},\mathbf{x})p(\mathbf{z}|\mathbf{x})d\mathbf{z} \\
&= \log \int p_\theta(\mathbf{g}|\mathbf{z},\mathbf{x})\frac{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{g})}{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{g})}p_\theta(\mathbf{z}|\mathbf{x})d\mathbf{z} \\
&\geq \int q_\phi(\mathbf{z}|\mathbf{x},\mathbf{g}) \log [\frac{p_\theta(\mathbf{g}|\mathbf{z},\mathbf{x})}{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{g})}p_\theta(\mathbf{z}|\mathbf{x})]d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x},\mathbf{g}) \log p_\theta(\mathbf{g}|\mathbf{z},\mathbf{x})d\mathbf{z} - \\
&\quad \int q_\phi(\mathbf{z}|\mathbf{x},\mathbf{g}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{g})}{p_\theta(\mathbf{z}|\mathbf{x})}d\mathbf{z} \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{g})}[\log p_\theta(\mathbf{g}|\mathbf{z},\mathbf{x})] - KL(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{g})||p_\theta(\mathbf{z}|\mathbf{x})).
\end{aligned}
\tag{1}
$$

Here $p_\theta(\mathbf{g}|\mathbf{z},\mathbf{x})$ represents our Grasp Flow, $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{g})$ for Variational Network, and $p_\theta(\mathbf{z}|\mathbf{x})$ for Prior Flow. In practice, we add a hyper-parameter β for the second KL-divergence term to control the trade-off between the information stored in the latent and the regularization induced by the structure of the prior. Concisely speaking, a high β will enable the model learn more shape-aware lantents but may be less informative for grasp prediction and vice versus. Therefore, a proper value is to be tuned for achieving a balance according to the specific task.

2 Data Generation Pipeline

2.1 Training, Evaluation and Testing Objects

For the data generation, we collect 89 graspable objects filtered from KIT datasets according to their graspability. We split 89 KIT [1] objects into a training set containing 77 objects, shown in Figure 1, and a test set of 12 objects as "similar" objects (Baking Soda, Bath Detergent, Broccoli Soup, Cough Drops, Curry, Fizzy Tablets, Instant Sauce, Nut Candy, Potato Dumpling, Spray Flask, Tomato Soup, Yellow SaltCube). We further include 9 YCB objects with distinct geometric shapes to the training set as "novel" objects (Bowl, Baseball, Power Drill, Plastic Pear, Plastic Banana, Mug, Clamp, Toy Airplane parts), illustrated in Figure 2.

2.2 Heuristic grasp planner

We derive the heuristic grasp planner from [2]. Here is a more detailed explanation. The heuristic grasp planner samples the grasp poses based on the normal of each object point cloud. We extend the target point cloud in the normal direction with a random value between 4.5 and 11.5 cm. Then, we add translation noise of $\pm 1cm$ in 3D space. To improve the data generation efficiency, the y-axis of the palm pose is aligned with the more extended object side and oriented upwards. Afterward, we add rotation noise of ± 0.7 rad around the x-direction and ± 0.35 rad around y- and z- directions.

To efficiently sample the 15-dof joint configuration, we apply eigengrasps from Ciocarlie [3] to sample a valuable subspace. Ciocarlie's work was inspired by the Neuroscience community, which showed that the joint DoFs of human hands during real-world grasping trials were primarily not operating independently but coordinated. More than 80 % of the variation in the data could be explained by the two first components of the principal component analysis (PCA). These components

Figure 1: **The training objects from KIT dataset for data generation**

Figure 2: **The testing objects of 12 KIT dataset as "similar" and 9 YCB objects as "novel".**

were termed eigengrasps, as almost any grasp joint configuration can be synthesized as a linear combination of a few eigengrasps. Thus, we design four eigengrasps $\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}, \mathbf{e_4} \in \mathbb{R}^{15}$.

$$\theta = \sum_{i=1}^{4} k_i \mathbf{e}_i \tag{2}$$

The full joint configuration θ is computed through sampling the coefficients $k_i \in [0, 1]$.

2.3 Grasp data generation pipeline

The data generation pipeline is similar with [2]. First, we randomly spawn an object in front of the robot. Then, a point cloud is recorded by the simulated camera. Afterward, we generate grasp samples based on heuristic grasp planner explained in Section 2.2, which are further filtered by Moveit in terms of reachability and collision. The robot then will execute the sampled grasp and attempt to lift it, where the grasp success is labeled automatically. This process is repeated for all objects with multiple random poses.

Since the grasp distribution is only object-dependent, the model should predict the same grasp distribution given different partial views of the same object. Therefore, we apply a data augmentation strategy by randomly spawning every object with 50 different initial poses to increase the dataset capacity by 50 times. In total, we generated a dataset of around 180k grasps, of which 30k resulted in success.

2.4 Experiment setup

Figure 3 shows our simulation setup. We use a Panda robot model with the DLR-HIT II hand as the end-effector. A simulated Realsense D415 camera is used to capture the point cloud. Afterward, the scene point cloud is captured by a Realsense D415 camera and then segmented with plane removal from RANSAC [4] to obtain the segmented object point cloud. The Basis Point Set (BPS)-encoded point cloud, after being segmented with plane removal, is fed as input to different models to grasp synthesis and ranking. Grasping success is defined as the ability for the DLR-HIT Hand II to lift the object 20 cm above its resting position without slippage.

The top grasp with the highest score is subsequently selected for execution. We conduct up to 20 trials per object in our simulation experiments. To facilitate a fair comparison for the grasp generator without a grasp evaluator, we evaluate the grasp samplers in simulation by executing the top 20 grasps instead of the single top-most one.

We choose the 12 test objects from the KIT dataset for the experiments in simulation. Each test object is spawned in simulation 20 times in random positions and random yaw-angle orientations. After recording each point cloud, we segment the object from the ground plane via RANSAC [4]. We combine the segmented object point cloud with random samples from the base distribution of FFHFlow model, namely a univariate Gaussian $\{\mathbf{z}\}^{100} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to generate 100 grasps per object. Afterward, Grasp Evaluator will rank all the generated grasps with predicted success probability. The grasp with the highest score will be executed. Therefore, we include grasp failures, which happen during the grasp execution phase, but exclude the failures where the robot collides with the object on the way to reach the grasp pose.

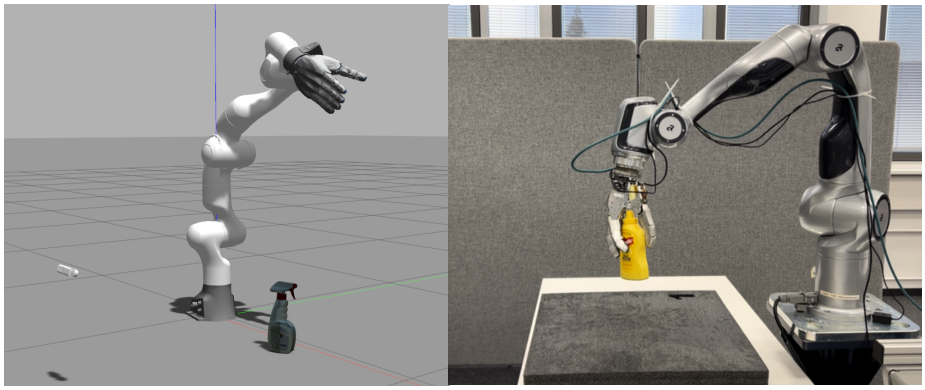


Figure 3: **The simulation setup in Gazebo and the real world setup**

The real-world setup is also demonstrated in Figure 3. For each method, we perform 80 grasps with 8 YCB objects [5] in the real world with a free workspace, and 20 grasps with 4 objects in a constrained workspace. Since our method is robot-independent, we choose Diana's robot arm, which is kinematically similar to the Franka robot, for real-world experiments.

2.5 Implementation Details

We first pre-process the input point cloud with BPS encoding [6], which is reported to work similarly well with PointNet++ [7] but with less compute. This reduces the overall computation and decrease the inference time. Afterwards the point cloud features are extracted with fully-connected residual block (FC Resblock) and are further conditioned on the flow model. We use a similar architecture for the point cloud feature extractor, variational inference network, and grasp evaluator, *i.e.*, a network with multiple fully-connected residual blocks. We use skip connections from each input to each fully-connected residual block (FC ResBlock) or fully-connected (FC) layer. The core building block of both models is the FC ResBlock, which consists of two parallel paths from input to output. One path consists of a single FC layer, the other path has two FC layers. Each is followed by a layer of batchnorm (BN).

Based on the *normflows* package [8], we implement the Grasp Flow (for both *FFHFlow-cnf* and *FFHFlow-lvm*) and Prior Flow (only in *FFHFlow-lvm*) with an 8-layered conditional Glow [9] where each layer has a 4-layered Multi-Layer Perceptron (MLP) for predicting the parameters of the affine operation. Both models are trained with a learning rate of $1\mathrm{e}{-4}$ and a mini-batch size of 64 for 16 epochs or $20k$ iterations. The objective in Equation (1) is then optimized with a linearly increased β from $1\mathrm{e}{-7}$ to $1\mathrm{e}{-1}$ in each iteration based on the *AdamW* optimizer [10] for *FFHFlow-lvm*. We also use Monte Carlo sampling to approximate the expectation operation in Equation (1). The number of samples is empirically set to 1. Moreover, during evaluation, we apply a positive offset of 0.2 rad on predicted joint configurations to ensure a more stable grasp.

2.6 Metric: Coverage

Coverage (Cov): It measures the fraction of grasps in the ground truth grasp set $\mathbf{G}_{gt}$ that is matched to at least one grasp in the generated set $\mathbf{G}_{gen}$:

$$Cov(\mathbf{G}_{gen}, \mathbf{G}_{gt}) = \frac{|\{\arg\min_{\mathbf{G}_{gt}} d(\mathbf{g}_{gen}, \mathbf{g}_{gt}) | \mathbf{g}_{gen} \in \mathbf{G}_{gen}\}|}{|\mathbf{G}_{gt}|} \tag{3}$$

For each grasp in the generated set $\mathbf{G}_{gen}$, its nearest neighbor based on L2 distance in the ground truth set $\mathbf{G}_{gt}$ is marked as a match. Coverage (Cov) can be used to quantify the diversity of the generated grasp set with the ground truth set as reference.

3 Additional Experimental Results

Table 1: Results Comparison on Cov and Run-time

Methods (w/o eval)	Cov ↑	Run-time ↓
FFHNet [2]	$22.5\% \pm 1.6\%$	30ms
FFHNet-prior	$24.4\% \pm 1.0\%$	31ms
FFHFlow-cnf	$30.0\% \pm 0.2\%$	70ms
FFHFlow-lvm	$30.3\% \pm 0.3\%$	130ms
FFHFlow-lvm-light	$29.9\% \pm 0.4\%$	60ms

3.1 Per-object Simulation and Real-world Results

We also include all the per-object simulation results in Table. 2 and Table. 3. We also include per-object results for real-world experiment with unconfined workspace in Table. 4 and with confined workspace in Table.5. Note that in Table.4, Chips Can have no results since they are too large for our hand to grasp.

Table 2: Per-object Success Rate Comparison for Similar Objects in Simulation

Methods	Objects												Average Succ Rate
	Baking Soda	Bath Detergent	Broccoli Soup	Cough Drops	Curry	Fizzy Tablets	Instant Sauce	Nut Candy	Potato Dumpling	Spray Flask	Tomato Soup	Yellow SaltCube	
Heuristic	6/20	2/20	4/20	4/20	8/20	8/20	2/20	4/20	3/20	2/20	3/20	4/20	20.9%
cVAE [2]	19/20	**19/20**	**19/20**	18/20	**19/20**	**20/20**	15/20	12/20	16/20	19/20	13/20	14/20	84.6%
GAN [11]	16/20	15/20	17/19	18/19	**19/20**	15/19	18/20	14/20	13/20	**19/19**	**20/20**	**19/20**	86.0%
Diffusion [12]	19/20	14/20	16/20	18/19	18/20	19/20	16/20	17/20	**20/20**	17/19	19/20	18/20	88.2%
FFHFlow-cnf	19/20	18/20	17/20	17/20	17/20	**20/20**	15/20	15/20	17/20	18/20	15/20	17/20	85.4%
FFHFlow-lvm	**20/20**	**19/20**	**19/20**	**19/20**	**19/20**	**20/20**	**20/20**	**18/20**	18/20	**20/20**	17/20	18/20	**94.6%**

Table 3: Per-object Success Rate Comparison for Novel Objects in Simulation

Methods	Objects									Average Succ Rate
	Power Drill	Baseball	Bowl	Mug	Pear	Banana	Extra Large Clamp	C Toy Airplane	B Toy Airplane	
Heuristic	3/20	2/20	4/20	9/20	3/20	3/20	0/20	8/20	0/20	17.8%
cVAE [2]	12/20	16/20	3/10	13/20	12/20	5/20	**5/20**	**19/20**	4/20	52.4%
GAN [11]	**15/18**	**17/20**	**7/12**	7/12	14/20	1/20	1/20	11/20	**7/20**	49.4%
Diffusion [12]	12/19	15/20	7/17	13/20	16/20	**10/20**	0/20	15/20	3/20	51.7%
FFHFlow-cnf	13/20	14/20	2/17	**15/20**	8/20	1/20	1/20	9/20	2/20	36.7%
FFHFlow-lvm	13/18	13/20	2/11	14/20	**18/20**	6/20	3/20	15/20	5/20	**52.7%**

3.2 Uncertainty-aware Grasp Evaluation

Uncertainty Quantification: For the experiment conducted for Figure 4, we collect an evaluation set and generate 100 grasp candidates for each partial view. For each grasp, we obtain the likelihoods of Grasp Flow and Prior Flow, as well as the evaluator scores. To assess the quality of the generated grasps, we utilize the Flexible Collision Library (FCL) to predict collisions for each grasp and Gazebo to evaluate the grasp stability of the remaining non-collided grasps. The x-axis represents the percentage of top-ranked values retained, ranging from 100% to 10%, while the y-axis shows the failure rate.

In Figure 4, we observe a clear *negative* correlation between the grasp evaluator score and the failure rate due to collision. In contrast, Prior Flow and Grasp Flow demonstrate the ability to reduce collision. among which Prior Flow exhibits the strongest correlation with the collision rate, highlighting its potential for capturing shape awareness. In the bottom plot, both the evaluator score and Grasp Flow likelihoods exhibit a strong correlation with grasp stability. The grasp evaluator outperforms Grasp Flow as it was specifically trained to distinguish positive grasps from negative ones. However, the Prior Flow, representing the object-level shape uncertainty, is less relevant to grasp stability.

Ablation Study: We conduct an ablation study presented in Table 6 to understand the trade-off between increasing grasp quality (grasp evaluator) and lowering view-level shape uncertainty (Grasp Flow), namely the optimal value of the additive coefficient (ϵ). By increasing the impact of lowering ϵ, we can see the performance first increases and drops. The optimal value is 0.01, indicating the major contribution to grasp success from the grasp evaluator.

Table 6: ϵ in Introspective Grasp Evaluation

Additive Coefficients (ϵ)	0.0	0.01	0.1	0.5	1.0
Similar	90.5%	**94.6%**	90.6%	78.6%	63.0%
Novel	50.9%	**52.7%**	50.9%	34.3%	25.3%

Table 4: Per-object Success Rate Comparison for Objects in Real-World Unconfined Workspace

Methods	Objects									Average Succ Rate
	Sugar Box	Apple	Tomate Soup Can	Pudding Box	Mug	Mustard Bottle	Chips Can	Baseball	Foam Brick	
cVAE [2]	**9/10**	2/10	6/10	**10/10**	3/10	6/10	-	4/10	**10/10**	62.5%
FFHFlow-cnf	4/10	4/10	**8/10**	9/10	**7/10**	4/10	-	5/10	**10/10**	63.75%
FFHFlow-lvm	8/10	**6/10**	**8/10**	**10/10**	6/10	**7/10**	-	**8/10**	9/10	**77.5%**

Table 5: Per-object Success Rate Comparison for Objects in Real-World Confined Workspace

Methods	Objects				Average Succ Rate
	Foam Brick	Pudding Box	Baseball	Tomato Soup Can	
cVAE [2]	0/5	0/5	0/5	2/5	10.0%
FFHFlow-lvm	4/5	3/5	3/5	3/5	**65.0%**

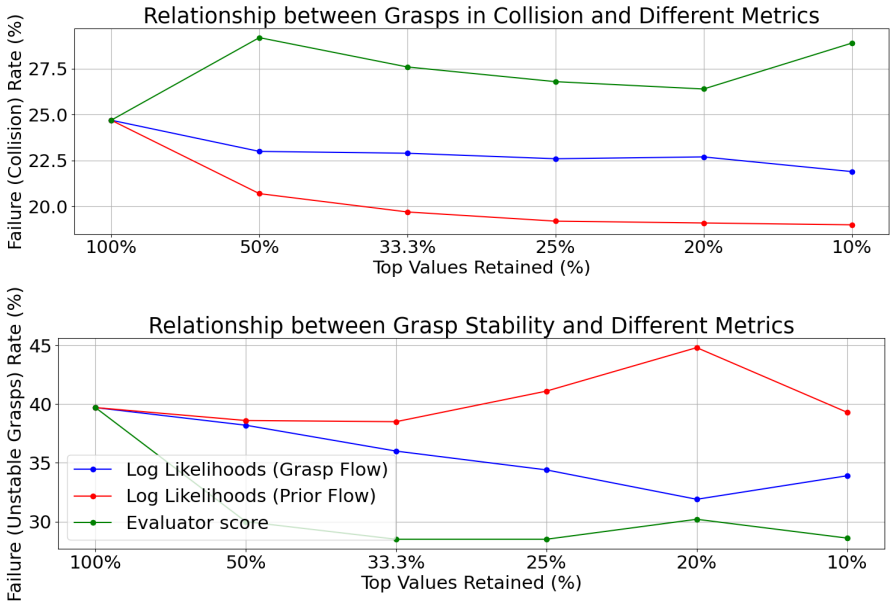


Figure 4: **Number of collided (Top) and unstable (bottom) grasps** filtered with an increasing threshold (higher the better). Likelihoods from Grasp Flow (Blue) achieves a more optimal balance between grasp stability and collision.

3.3 Point Cloud Latent Feature Visualization

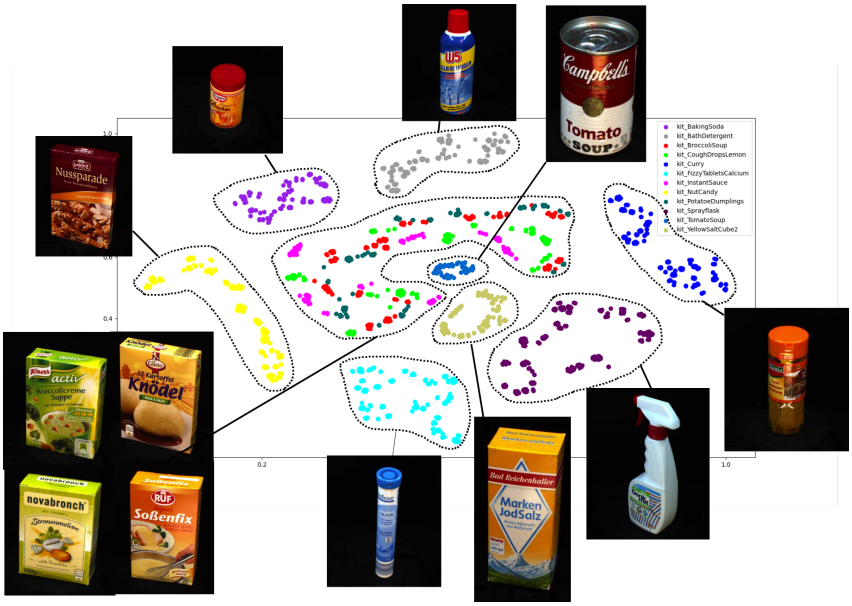
In this subsection, we compare the point cloud latent feature visualization from three models, namely *FFHFlow-cnf*, *FFHFlow-lvm* and FFHNet [2] in Figure 5.

Though *FFHFlow-cnf* has achieved encouraging improvements in terms of diversity and accuracy when compared to the Conditional Variational Autoencoder(cVAE)-based approach, we found *FFHFlow-cnf* less generalizable with limited performance gain. We attribute this problem to the inadequate expressivity of the latent feature, especially when the model needs to understand the complicated relationships between the grasps and the partially observed point clouds of different objects. For example, from our empirical observation, the latent features are assumed to be capable of extracting *two-level hierarchical grasp-relevant information* such as object shape or category from the partially seen object point clouds. (1) *object level* summarizes the grasp-related clues of different objects, such as a box and a bottle; (2) *instance level* subsumes the grasp-associated details of an instance of the same object but captured from different viewpoints.

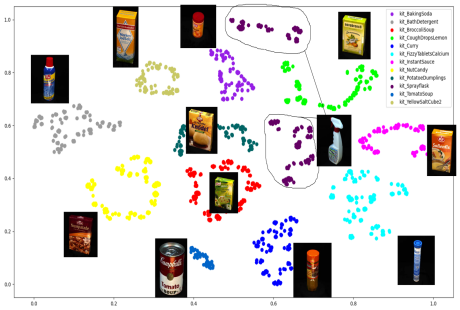
We note that for *FFHFlow-cnf*, we generate the feature visualization with different random seeds to the one in the paper. Nevertheless, both exhibit similar behaviors, further confirming the underperformance of *FFHFlow-cnf* in extracting geometrically meaningful features.

3.4 Experiments of Grasping in Cluttered Scenarios

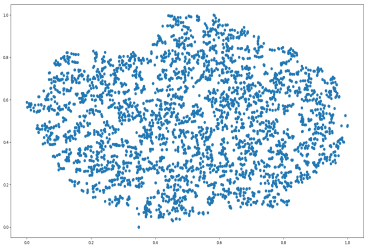
A diverse grasp generator further enables its application for grasping objects in clutter. We propose our grasping pipeline for cluttered unknown objects leveraging Large Language Models (LLMs) and Vision Language Models (VLMs). We first prompt ChatGPT 4o [13] to obtain object names shown on the table and further feed these namse to Grounded SAM [14] to segment out objects. The



(a) *FFHFlow-lvm*



(b) *FFHFlow-cnf*



(c) cVAE

Figure 5: **Point Cloud Feature Visualization** based on t-SNE from (a) *FFHFlow-lvm*, (b) *FFHFlow-cnf* and cVAE in [2]. We illustrate t-SNE features on all 12 KIT test objects used in the simulation. The objects in (a) with similar shapes are closer on the feature space, especially the four boxes on the left bottom part, demonstrating the geometric meaningfulness of the latent features. The cVAE-based approach depicts the least meaningful feature visualization, where the latent samples are drawn from an input-independent prior.

nearest objects will be chosen to be grasped first. We randomly add one obstacle to each different scene to increase the clutterness, to further mimic the household scenarios, show in Figure 6. We convert partial point cloud to meshes and further filter out collisions between environment and the robotic hand using Flexible Collision Library (FCL) [15].

Figure 6: The cluttered scenes contain four unknown objects with additionally unknown obstacles, namely the flower vase and the drawer.

We conducted the grasping experiment for 4 cluttered scenes, each with 4 different objects. We evaluate the cluttered grasping performance with success rate (SR) and clearance rate (CR) in manuscript. The success rate is measured by successful grasps out of all grasp attempts, and the clearance rate is measured by the number of times robots can clear the scene.

FFHFlow-lvm outperforms FFHNet [2] with 7.8% in terms of success rate with a better clearance rate. We observe several failures from FFHNet [2] where a less diverse grasp generator fails to generate valid grasps for occluded objects, especially the blue bowl under the flower vase and the foam brick close to the drill, where the top grasps will be filtered by collision. We further illustrate the influence of diverse grasp distribution in cluttered scenes in Figure 7.

3.5 Visualization of Predicted Grasp Palm Poses and Joints

To show the enhanced diversity, we first compare the grasp palm pose distribution of different approaches shown in Figure 14. By comparing horizontally, we can inspect that our flow-based variational approach, *FFHFlow-lvm* can model the target multi-modal distribution with higher fidelity. Meanwhile, *FFHFlow-cnf* achieves similar results as *FFHFlow-lvm*, especially for box-like objects, but still generates relatively flattened top grasps for cylinder-like objects, such as 2,3,5 rows in 14. In contrast, the cVAE-based approach can only predict less diverse grasps due to the *mode-collapse* problem.

On the other hand, we also visualized the grasps of the full hand, including both the palm poses and hand joint configurations for grasping in clutter in Figure 8 and in Figure 10, single objects in the real-world in Figure 9, from *FFHFlow-lvm*. By inspecting these figures, we can see the dexterity in the predicted hand joints. Moreover, when comparing the grasps from *FFHFlow-lvm* and FFHNet in Figure 8, we can see the diversity of the hand, including the palm and the finger joints, are greater for *FFHFlow-lvm*.

3.6 Failure analysis for Simulation and Real-world Experiments

In the simulation experiment, as shown in Figure 12, *FFHFlow-lvm* causes 2 failures (15.4%) from unstable grasp palm pose, 9 failures (69.2%) from wrong joint configurations, and 2 failures (15.4%) from collisions between the hand and the target object. Failures resulting from joint configurations depict grasps where fingers often are not close enough to apply sufficient force in simulation. This kind of failure normally doesn't exist in real-world experiments. Because the hand impedance controller tends to close the finger more if it's not in contact. However, since this controller cannot be simulated, we replace it with a positional controller. Meanwhile, *FFHFlow-cnf* causes 6 failures from grasp poses, 8 from joint configurations, and 4 from collision. We observe that *FFHFlow-cnf* tends to fail more often because of wrong-predicted grasp poses and collisions. This reason holds for the baseline FFHNet [2] as well (13 failures from grasp poses, 13 from joint configurations, and 7 from collision).

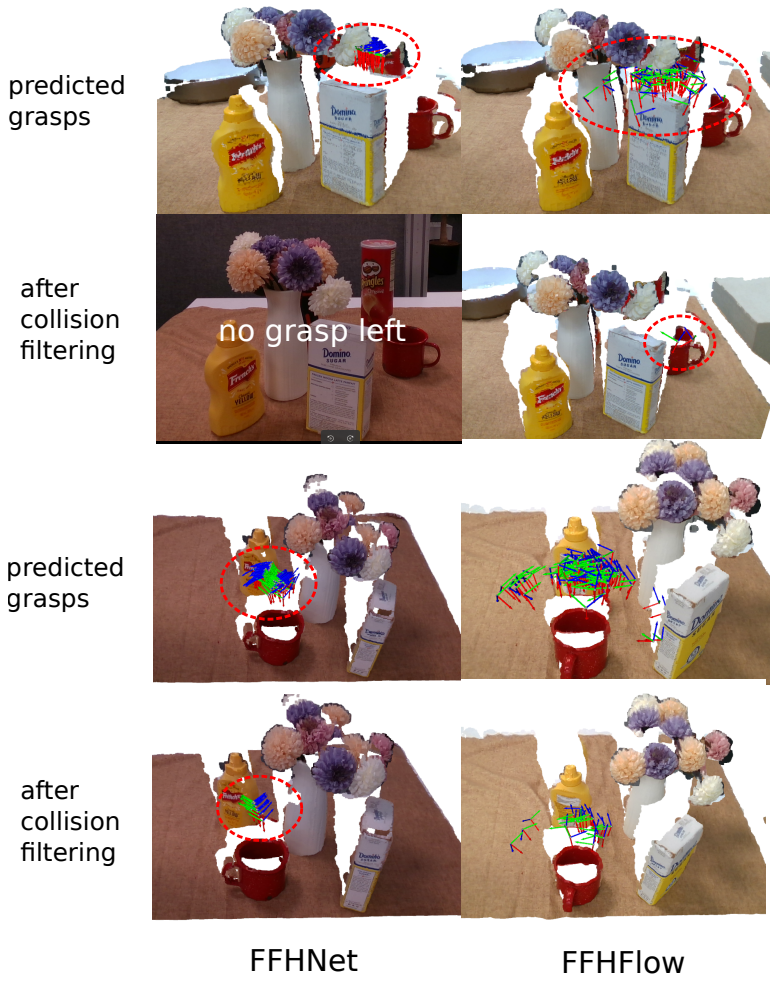


Figure 7: **The grasp distribution generated by FFHNet [2] and *FFHFlow-lvm* before and after collision filtering. A less diverse grasp distribution restricts its application to cluttered scenes.**

In the real-world experiments, by analyzing the errors in Figure 11, when compared to *FFHFlow-cnf* and FFHNet [2], *FFHFlow-lvm* has much fewer failures from unstable grasp pose and a similar number of those from collisions. This trend verifies the superior generalization ability of *FFHFlow-lvm*. On the other hand, for the objects with a low success rate, *FFHFlow-cnf* tends to grasp the corner from a tilted angle instead of the body for the sugar box (40%). FFHNet [2] failed a lot for metal mugs (30%) due to its bias toward top grasps that are harder than side grasps. Apple (30% on average) has the lowest success rate for all models because of its slippery surface, which is often the reason for unstable grasp pose.

3.7 Ablation Study for FFHFlow-cnf

To conduct a fair comparison between *FFHFlow-cnf* and *FFHFlow-lvm* , we increase the size of *FFHFlow-cnf*, namely doubling the layers of the flow. In Table 7, even with a two-times larger size, we can only observe slight improvement, which highlights the inherent limitation on the expressiveness in the latent space of *FFHFlow-cnf*.

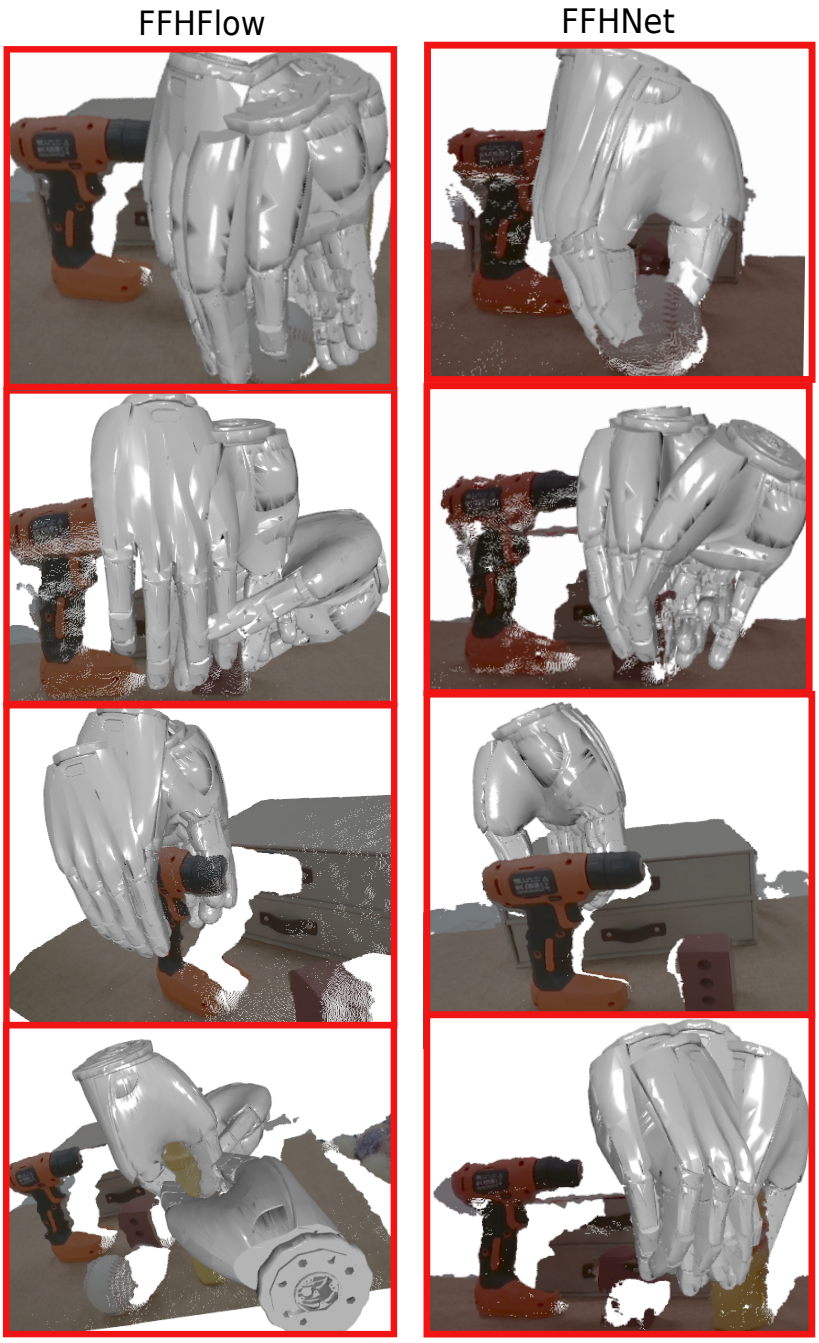


Figure 8: **Comparion on visualization of top 5 scored grasps in the cluttered scene in real-world experiments. FFHFlow demonstrates the ability to generate grasps with better diversity.**

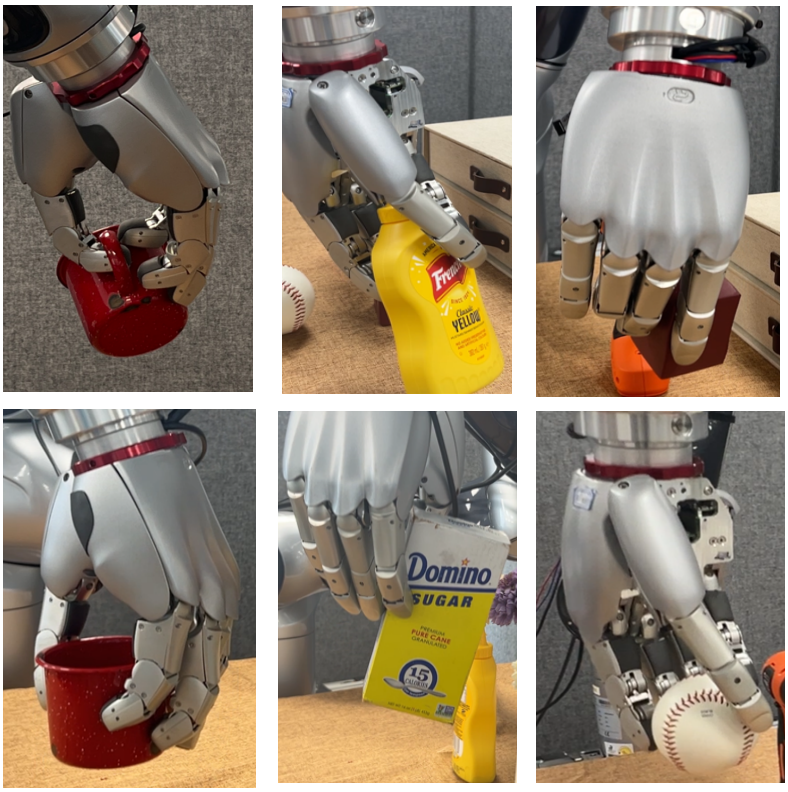


Figure 9: **Exemplar screenshots of grasps in real-world experiments.**

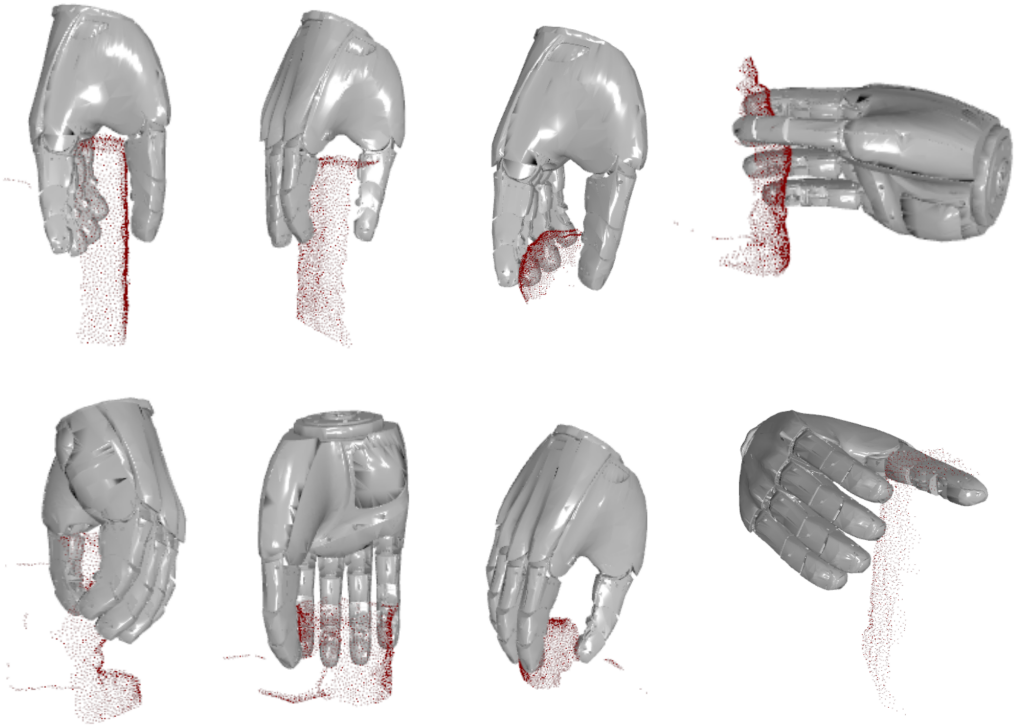


Figure 10: **Exemplar grasp visualization in real-world experiments.**

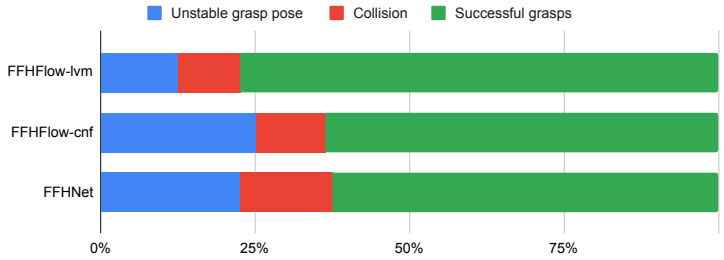


Figure 11: **Failure Analysis for the Real-world Experiment**

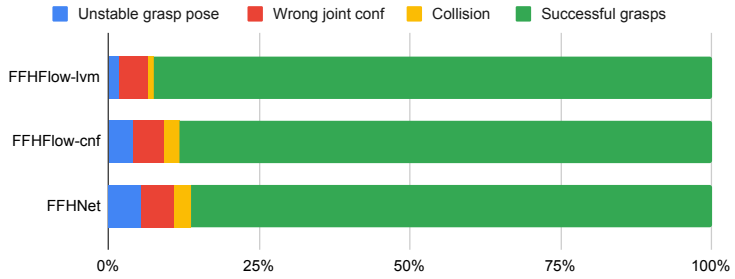


Figure 12: **Failure Analysis for the Simulation Experiment**

Table 7: Ablation Study for FFHFlow-cnf

Methods (size)	Objects												Average Succ Rate
	Baking Soda	Bath Detergent	Broccoli Soup	Cough Drops	Curry	Fizzy Tablets	Instant Sauce	Nut Candy	Potato Dumpling	Spray Flask	Tomato Soup	Yellow SaltCube	
FFHFlow-cnf (8 layers)	95.0%	100.0%	95.0%	90.0%	100.0%	80.0%	95.0%	85.0%	100.0%	100.0%	70.0%	70.0%	92.5%
FFHFlow-cnf (16 layers)	95.0%	90.0%	95.0%	95.0%	100.0%	95.0%	90.0%	90.0%	95.0%	95.0%	90.0%	85.0%	92.9%

3.8 Ablation Study of FFHFlow-lvm

The question of "What are the critical factors in the proposed models that influence the performance most?" is of particular interest for better understanding the proposed models, in particular *FFHFlow-lvm* for its complexity. From the results of an ablation study in Table 8, we can draw the messages that positional encoding pre-processing, adding conditional base distribution only to grasp flow generator can help alleviate over-fitting and improve the generalization performance. Here positional encoding is applied on Euler angles (3D) to obtain 60D, compared to the baseline of using 6D rotation representation [16], originally already used in FFHNet [2].

The potential reason for the benefit of positional encoding can be the better capability of expressing high-frequency information from low-dimensional data such as 3-d angel vectors in our case [17]. Moreover, the size ratio of two flows in the model seems to influence the training stability. When the model has two different number of layers assigned to the grasp and prior flow, the coverage is much lower and the simulation evaluation failed due to some feasible predicted values from the model.

Evaluating Predicted Finger Joints To investigate how much the predicted finger joints matter, we conduct an ablation study on comparing the success rate of grasping with and without the predicted joints in simulation. In case of grasping without predicted joints, we set the corresponding joints with $0.2\ rad$ to approximate a power grasp for each object. In Table 9, we can observe a clear drop when grasping without the predicted joints for both methods, confirming their positive effects for precise grasp synthesis. Moreover, the increase brought by *FFHFlow-lvm* (9.6%) is significantly higher than that of the cVAE approach [2] (1.6%). Such improvement can demonstrate the overall benefits of our proposed models for not only the predicted palm poses but also the predicted joints.

Table 8: Ablation study of *FFHFlow-lvm* on Cov and Success rate

Ablated Models (w/o eval)	Cov ↑	Succ Rate ↑
FFHFlow-lvm	30.2%	94.6%
6D (w/o-positional-encoding)	30.1%	92.3%
both-flows-cond-base	30.4%	89.8%
both-flows-w/o-cond-base	30.7%	88.2%
grasp-flow-4-layers	28.8%	-
both-flows-4-layers	30.2%	93.3%

Table 9: Predicted Joints Evaluation

Methods	Objects												Average Succ Rate
	Baking Soda	Bath Detergent	Broccoli Soup	Cough Drops	Curry	Fizzy Tablets	Instant Sauce	Nut Candy	Potato Dumpling	Spray Flask	Tomato Soup	Yellow SaltCube	
FFHNet [2] w/o Joints	95.0%	95.0%	95.0%	90.0%	95.0%	100.0%	75.0%	60.0%	80.0%	95.0%	65.0%	70.0%	84.6%
FFHNet [2]	90.0%	95.0%	100.0%	95.0%	95.0%	95.0%	100.0%	65.0%	85.0%	90.0%	65.0%	60.0%	86.2%
FFHFlow-lvm w/o Joints	90.0%	70.0%	100.0%	90%	100.0%	90.0%	85.0%	85.0%	85.0%	75.0%	65.0%	85.0%	85.0%
FFHFlow-lvm	95.0%	95.0%	95.0%	100.0%	95.0%	95.0%	100.0%	85.0%	100.0%	90.0%	100.0%	85.0%	94.6%

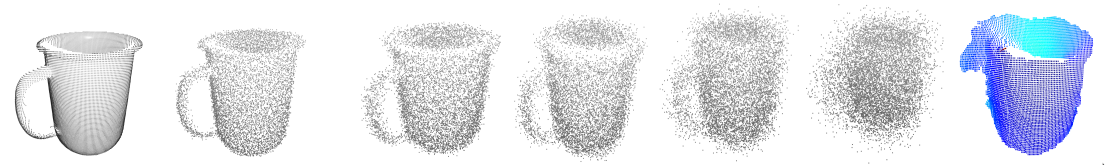
3.9 Influence of Point Cloud Noises to FFHFlow-lvm

We add random gaussian noise to the point cloud in simulation and feed it to *FFHFlow-lvm*. The results in Table 10 demonstrate its negative influence of noises on success rate. We observed a roughly linear performance drop between 0mm to 5mm and then the performance drops dramatically from 5mm with 75.3% to 10mm with 29.9%. Given a real world point cloud in Table 10, it's noise level is estimated to be between 0 and 1 mm. Therefore we can expect a performance drop of around $3-4\%$ given same level of noise from real world, ignoring all other sim2real gap. Other sim2real gap for point cloud could be missing pixels from physical camera and imperfect segmentation mask.

To better minimize the negative influence of noise or improve the robustness against noise, we could in principle train the model with the simulated noised point cloud.

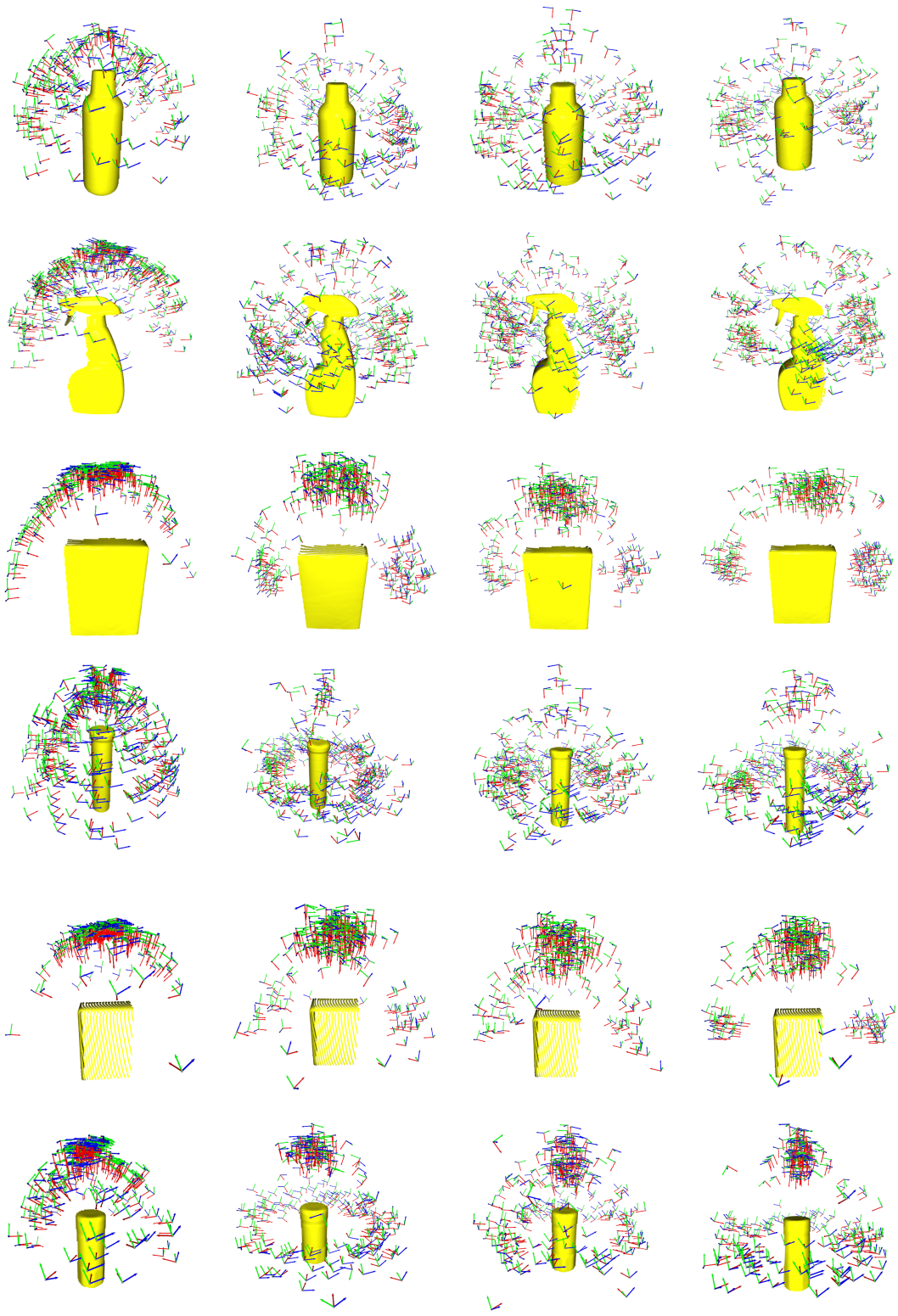
Table 10: Success Rate Drop vs Point Cloud Noise

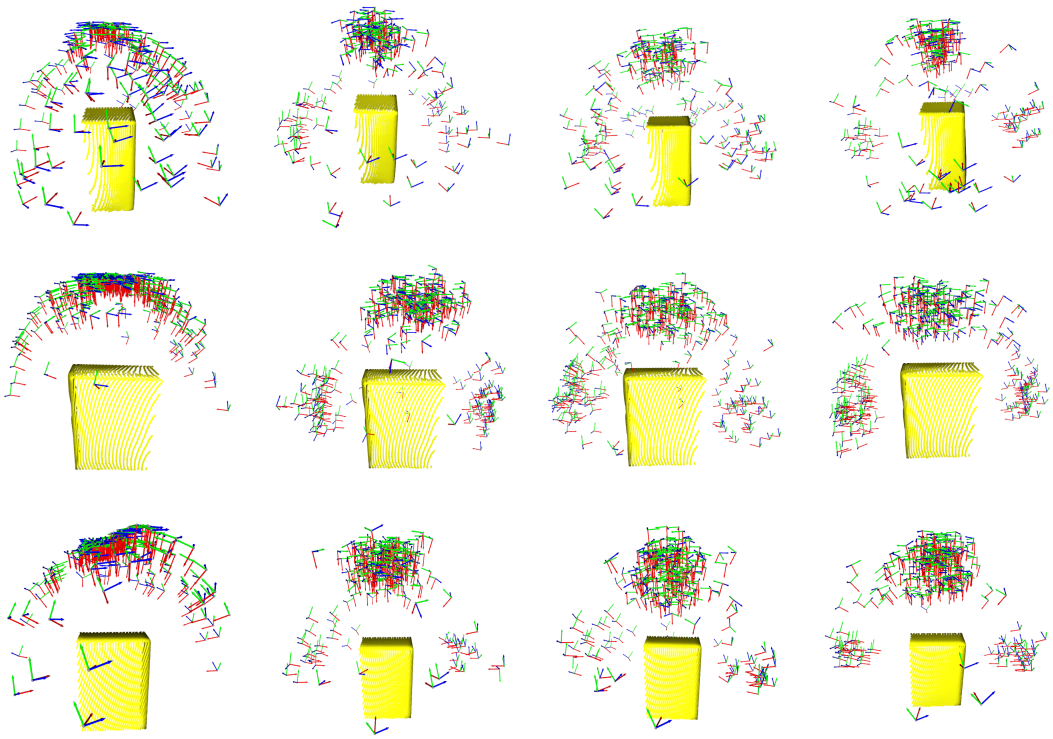
	Gaussian Noise					
Standard deviation	0 mm	1mm	2mm	3mm	5mm	10mm
FFHFlow-lvm	94.6%	91.2%	88.3%	83.3%	75.3%	29.9%



(a) Original point cloud (b) 1mm noise (c) 2mm noise (d) 3mm noise (e) 5mm noise (f) 10mm noise (g) Real world point cloud

Figure 13: **Visualization of Point Cloud applied with different magnitude of Gaussian Noise.** We apply noise generated from a zero mean and a parameterized standard deviation Gaussian distribution to original point cloud. The added standard deviation is in a range from 1mm till 10mm. We can see the point cloud gets more fuzzy and almost not recognizable after 5mm. Compared to (g) real world point cloud, we can estimate its noise level is between 0-1 mm standard deviation.





(a) cVAE (b) *FFHFlow-cnf* (c) *FFHFlow-lvm* (d) Ground truth

Figure 14: **Visualization of grasp pose distributions** from (a) cVAE in [2], (b) *FFHFlow-cnf*, and (c) *FFHFlow-lvm* and (d) ground truth.

References

[1] A. Kasper, Z. Xue, and R. Dillmann. The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, 31(8):927–934, 2012.

[2] V. Mayer, Q. Feng, J. Deng, Y. Shi, Z. Chen, and A. Knoll. Ffhnet: Generating multi-fingered robotic grasps for unknown objects in real-time. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 762–769, 2022. doi:10.1109/ICRA46639.2022.9811666.

[3] M. T. Ciocarlie and P. K. Allen. Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research*, 28(7):851–867, 2009.

[4] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

[5] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015.

[6] S. Prokudin, C. Lassner, and J. Romero. Efficient learning on point clouds with basis point sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4332–4341, 2019.

[7] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[8] V. Stimper, D. Liu, A. Campbell, V. Berenz, L. Ryll, B. Schölkopf, and J. M. Hernández-Lobato. normflows: A pytorch package for normalizing flows. *arXiv preprint arXiv:2302.12014*, 2023.

[9] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[10] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[11] Q. Feng, D. S. M. Lema, M. Malmir, H. Li, J. Feng, Z. Chen, and A. Knoll. Dexgangrasp: Dexterous generative adversarial grasping synthesis for task-oriented manipulation, 2024.

[12] Z. Weng, H. Lu, D. Kragic, and J. Lundell. Dexdiffuser: Generating dexterous grasps with diffusion models, 2024.

[13] OpenAI. Chatgpt (july 11 version), 2024. URL https://www.openai.com. Large language model.

[14] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[15] J. Pan, S. Chitta, and D. Manocha. Fcl: A general purpose library for collision and proximity queries. In *2012 IEEE International Conference on Robotics and Automation*, pages 3859–3866, 2012. doi:10.1109/ICRA.2012.6225337.

[16] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks, 2020. URL https://arxiv.org/abs/1812.07035.

[17] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.