

Multi-FinGAN: Generative Coarse-To-Fine Sampling of Multi-Finger Grasps

Jens Lundell, Enric Corona, Tran Nguyen Le, Francesco Verdoja, Philippe Weinzaepfel,
Grégory Rogez, Francesc Moreno-Noguer, Ville Kyrki

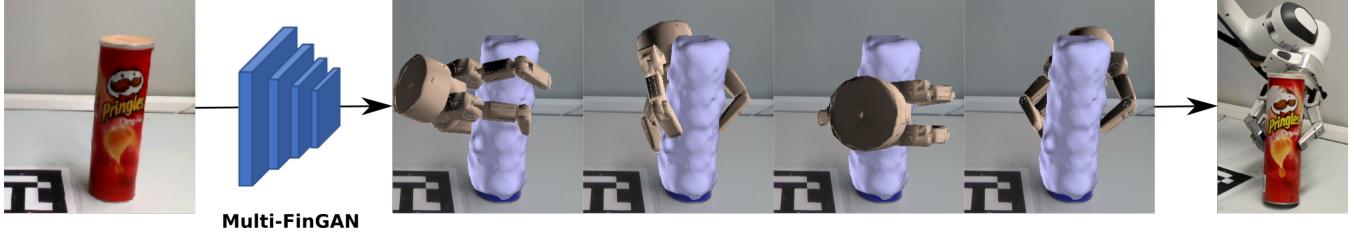


Fig. 1: From an input RGB-D image, Multi-FinGAN generates a diverse set of grasps from all around the object in about a second, and then executes the highest scoring grasp on the real robot.

Abstract— While there exists many methods for manipulating rigid objects with parallel-jaw grippers, grasping with multi-finger robotic hands remains a quite unexplored research topic. Reasoning and planning collision-free trajectories on the additional degrees of freedom of several fingers represents an important challenge that, so far, involves computationally costly and slow processes. In this work, we present *Multi-FinGAN*, a fast generative multi-finger grasp sampling method that synthesizes high quality grasps directly from RGB-D images in about a second. We achieve this by training in an end-to-end fashion a coarse-to-fine model composed of a classification network that distinguishes grasp types according to a specific taxonomy and a refinement network that produces refined grasp poses and joint angles. We experimentally validate and benchmark our method against a standard grasp-sampling method on 790 grasps in simulation and 20 grasps on a real *Franka Emika Panda*. All experimental results using our method show consistent improvements both in terms of grasp quality metrics and grasp success rate. Remarkably, our approach is up to 20-30 times faster than the baseline, a significant improvement that opens the door to feedback-based grasp re-planning and task informative grasping. Code is available at <https://irobotics.aalto.fi/multi-fingan/>.

I. INTRODUCTION

Generating multi-fingered grasps for unknown objects such as the one shown in Fig. 1 is still non-trivial and considerably more challenging than using parallel-jaw grippers for the same task. However, by actuating more joints, multi-fingered grippers allow a robot to perform more advanced manipulations, including precision grasping flat disks or power grasping spherical objects.

This work was supported in part by the Academy of Finland Strategic Research Council grant 314180, CHIST-ERA project IPALM, Finland 326304, Spain PCI2019-103386, and by the Spanish government with projects HuMoUR TIN2017-90086-R and María de Maeztu Seal of Excellence MDM-2016-0656.

J. Lundell (jens.lundell@aalto.fi), T. Nguyen Le, F. Verdoja, and V. Kyrki are with School of Electrical Engineering, Aalto University, Finland. E. Corona and F. Moreno-Noguer are with Institut de Robótica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain. G. Rogez and P. Weinzaepfel are with NAVER LABS Europe, Meylan, France.

Typical methods for multi-fingered grasp generation require known 3D object models and poses in order to sample a large space of candidate grasps and then evaluate them based on physical grasping metrics such as the ϵ -quality [1]. In case of unknown object poses, pose estimation is the *de facto* standard solution [2], while for unknown models estimating the shape through, *e.g.*, shape completion or mirroring has shown to work well in many scenarios [3]–[5]. Nevertheless, generating good candidate grasps with these methods is computationally expensive as it usually relies on a stochastic search process such as simulated annealing over a large search space. For instance, the search space for the robotic hand we consider in this work, the *Barrett hand* seen in Fig. 1, has 7 Degrees-Of-Freedom (DOF) which together with the 6D object poses (3 rotations and 3 translations) result in a 13 dimensional search space. Despite clever solutions to reduce the search space, such as limiting the search over eigengrasps [6], the process is still inherently slow (in the order of several tens of seconds) due to the stochastic search procedure.

In this work, we present a deep network inspired by recent work from the computer vision community on human hand grasp synthesis [7], [8] that can generate and evaluate multi-finger grasps on unknown objects in roughly a second. To achieve this, we devise a generative architecture for coarse-to-fine grasp sampling named *Multi-FinGAN* that is purely trained on synthetic data. Especially the integration of a novel parameter-free finger refinement layer based on a fully differentiable forward kinematics layer of the *Barrett hand* facilitates fast learning and robust grasp generation.

The proposed sampling method is quantitatively evaluated in both simulation, where we compare over 790 grasps against the baseline in terms of analytical grasp quality metrics, and on a real *Franka Emika Panda* equipped with a *Barrett hand*, where we evaluate grasp success rate on 10 grasps per method. In both cases, our approach demonstrates a significant reduction in running time compared to the baseline while still generating grasps with high quality metrics

and success rate.

In summary, the main contributions of this work are: (i) a novel generative method for multi-finger grasp selection that enables fast sampling with high coverage; (ii) a novel loss function for guiding grasps towards the object while minimizing interpenetration; and (iii) an empirical evaluation of the proposed method against state-of-the-art, presenting, both in simulation and on real hardware, improvements in terms of running time, grasp ranking, and grasp success rate.

II. RELATED WORK

When considering parallel-jaw grippers, a large corpus of data-driven generative [9]–[13] and classification based [14], [15] grasping methods exist. Many of these approaches [9], [14] reach a grasp success rate over 90% on a wide variety of objects by constraining to top-down-only grasps. However, as recently discussed by Wu *et al.* [16], the simplifications made in these works exclude many solutions that could be used for applications like semantic and affordance grasping [17] or multi-finger grasping for dexterous manipulation.

Despite the limitations of parallel-jaw grippers, alternative methods using multi-fingered hands have not seen as much development [16]–[22]. These approaches generally underperform both in terms of running time and grasp success rate compared to their parallel-jaw counterparts.

One of the earliest deep-learning-based multi-finger grasping work trained a network to detect the palm and fingertip positions of stable grasps directly from an RGB-D view [18]. That method achieved a 75% grasp success rate on 8 objects but relied on an external planner (GraspIt!) at run-time to generate grasp samples which made the method slow (16.6 seconds on average to generate a grasp). Our generative method also takes RGB-D images as input but does not require any external planner, making it a much faster solution.

To remove the need of a slow external planner, recent work also focused on generating grasps [19], [20], [22]. For instance, in [22], the authors train a network that regresses from a voxel grid representation of the object to the output pose and configuration of the gripper. Similar to ours, this work also employs the known forward kinematics equation of the gripper to compute a collision loss. At run-time the generated grasp is refined by searching over all ground-truth grasps, selecting the grasp closest to the generated one. The main drawback is that the grasps are viewpoint dependent so, to generate grasps in all possible locations around the object, the input representation needs to be rotated. Our method, on the other hand, can generate grasps from any orientation in just a single forward pass.

Aktas *et al.* [20] proposed a generative-evaluative model which both generates grasps and subsequently test them. Grasps are produced through stochastic hill-climbing on a product of experts, which is a sequential and time-consuming process. Our model, on the other hand, only requires one forward pass to generate a grasp that is then evaluated by computing analytical quality metrics [1].

The work most similar to ours is by Lu *et al.* [19]. It proposes a deep network that, given an initial grasp

configuration and a RGB-D or voxel representation of the object, optimizes a hand pose and finger joints to increase grasp success. The proposed method reached an average grasp success rate of 57.5% but requires roughly 5–10 seconds to generate a grasp. That work was later improved in terms of data-efficiency [21], but still the method required 3–10 seconds to generate a grasp. Our work, in comparison, does not require an explicit initial hand configuration as the network implicitly learns to predict such a configuration. Moreover, our method is much faster at generating grasps.

Tangential to training a multi-finger grasp sampler with supervised learning is to use Reinforcement Learning (RL). In [16], Wu *et al.* learned a deep 6-DOF multi-finger grasping policy directly from RGB-D inputs. That work introduced a novel attention mechanism that zooms in and focuses on sub-regions of the depth image to achieve better grasps in dense clutter. Although the policy was trained purely in simulation, it transferred seamlessly to the real world and attained a high grasp success rate on a diverse set of objects. Nevertheless, training such a method requires an elaborate simulation setup and fine-tuning of hyper-parameters for the RL method to work well.

III. PROBLEM FORMULATION

In this work, we consider the problem of grasping unknown objects with a multi-fingered robotic hand. This implies producing a grasp that does not interpenetrate the object but has several contact points with it. More formally, we train a model \mathcal{M} that takes as input an RGB-D image \mathbf{I} and produces a grasp type c , a 6D gripper pose \mathbf{p} , and a valid hand joint configuration \mathbf{q} :

$$\mathcal{M} : \mathbf{I} \implies \{c, \mathbf{p}, \mathbf{q}\} .$$

We represent the hand joint configuration \mathbf{q} by a 7-DOF *Barrett hand* shown in Fig. 1. c is a coarse grasp class within the 33-grasp taxonomy listed in [23]. Due to physical constraints, the *Barrett hand* can achieve only 7 out of these 33 grasp types, namely: small wrap, medium wrap, large wrap, power sphere, precision sphere, precision grasp, and pinch grasp. All grasps are in the object’s center of reference.

Furthermore, we assume that the hand joint configuration \mathbf{q} leaves a small gap between the fingers and the object as the gripper executes a close-hand primitive before attempting to actually lift the object. This assumption is reasonable to limit the impact of sensing uncertainties in the real world as we avoid the need to generate precise configurations that actually touch the surface of the object.

IV. METHOD

Our model for generating 6D multi-finger grasps inspired from [7] is visualized in Fig. 3. It consists of 6 different sub-modules: Shape Completion, Image Encoding, Multi-Label Grasp type Classification, Grasp Generation, Discriminator and Finger Refinement. All these modules are novel for robotic grasping except “shape completion”, which was used in [24] to shape complete voxelized input point-clouds using a deep network, and “image encoding”, which is a pre-trained

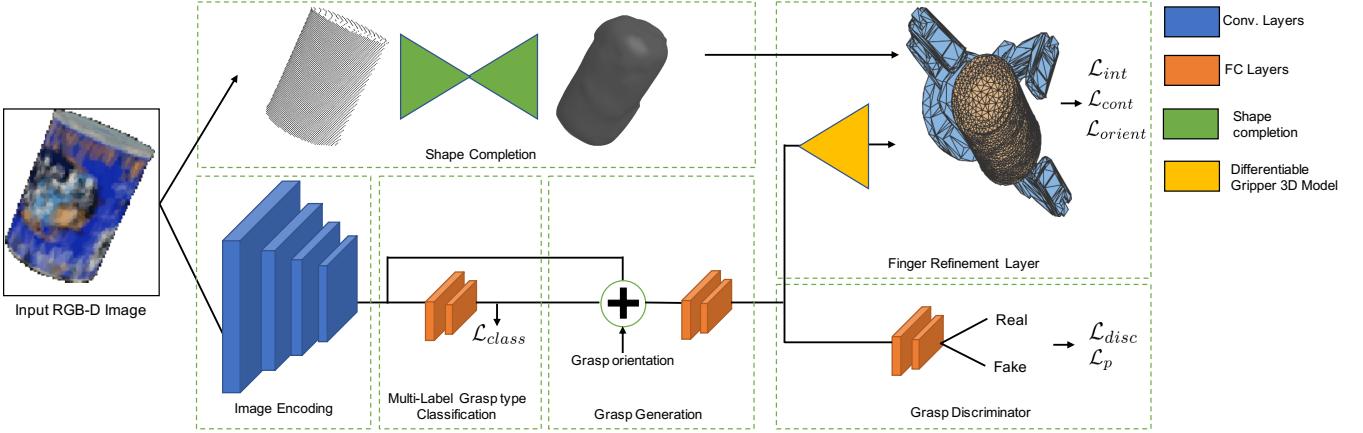


Fig. 3: The architecture of Multi-FinGAN.

ResNet-50 [25]. In the following subsections we will present each of the novel modules and their function. Finally, we will present the loss functions used to train the complete generative grasp planning architecture end-to-end.

A. Multi-Label Grasp Type Classification

The task of the Multi-Label Grasp Type Classification network is to classify which of the seven predefined grasp classes are feasible for a given object. For this purpose, the classification network is fed with the image representation of an object $E(\mathbf{I})$, where $E(\cdot)$ is a ResNet-50 encoding, and produces as output a grasp type c .

As objects can often be associated with multiple correct grasp categories, we frame the problem as a multi-label classification task. As such, we use a Sigmoid activation function at the output and the binary cross-entropy loss \mathcal{L}_{class} to train the network. To later choose one grasp among all the possible ones, we threshold the output to 0.5 and randomly choose one grasp type c that is classified as valid for that object.

B. Grasp Generation

The grasp type c estimated by the classification network is associated with a coarse hand configuration \mathbf{q}_c , *i.e.*, the average joint angles. The task of the Grasp Generation is to generate a first refinement of the hand configuration $\mathbf{q}_r = \mathbf{q}_c + \Delta\mathbf{q}$ along with a 6D hand pose $\mathbf{p} = \{\mathbf{t}, \mathbf{r}\}$ where \mathbf{t} is a translation and \mathbf{r} is a normalized axis-angle rotation.

Given the object mesh, we estimate the center of it \mathbf{t}_0 and have the network refine this translation as $\mathbf{t}^* = \mathbf{t}_0 + \Delta\mathbf{t}$. Similarly, we represent the refinement of the hand's rotation as $\mathbf{r}^* = \mathbf{r}_0 + \Delta\mathbf{r}$. At training time the input rotation \mathbf{r}_0 is set to a rotation of a ground-truth grasp with added zero-mean Gaussian noise while at test-time we sample uniform rotations and feed these to the network. The center of the object mesh \mathbf{t}_0 is known when training and during the simulation experiments, while during the physical experiments we use the center of the shape completed object.

All in all, the network is represented as a fully-connected residual network which takes as input the initial hand configuration, the object center, a noisy rotation, and the

image encoding $\{\mathbf{q}_c, \mathbf{t}_0, \mathbf{r}_0, E(\mathbf{I})\}$ and produces individual refinements $\{\Delta\mathbf{q}, \Delta\mathbf{t}, \Delta\mathbf{r}\}$.

C. Finger Refinement Layer

The Finger Refinement Layer is responsible to further refine the hand representation \mathbf{q}_r . To this end, we propose a novel fully-differentiable and parameter-free layer based on the forward kinematics equation of the *Barrett hand*. This layer takes as input the pose of the gripper \mathbf{p} and the coarse gripper representation \mathbf{q}_r and produces an optimized gripper configuration $\mathbf{q}^* = \mathbf{q}_r + \Delta\mathbf{q}^*$ that is close to the surface of the object but not in collision with it.

We optimize each finger independently with respect to the estimated object mesh. We denote $\Delta\mathbf{q}_j^*$ as the optimized position of joint j . This is calculated by rotating the articulated finger within its predefined physical limit θ_j until the distance δ_{θ_j} between the finger vertices $V_i^{\theta_j}$ and the object vertices O_k implies a contact between finger and object. Hand-Object contact is parameterized by a threshold hyperparameter t_d , following

$$\begin{aligned} \Delta\mathbf{q}_j^* &= \arg \min_{\theta_j} \{\delta_{\theta_j} + \epsilon - \mathbf{q}_{r,j}\} \quad \forall \theta_j \text{ s.t. } \delta_{\theta_j} < t_d , \\ \delta_{\theta_j} &= \min_i (\min_k (\|V_i^{\theta_j} - O_k\|)) . \end{aligned} \quad (1)$$

Note that we could have simply set $\mathbf{q}^* = \arg \min_{\theta} \delta_{\theta}$ for each joint as was proposed in [7] for a human hand model. However this would break backward differentiability, and instead, we explicitly calculate $\Delta\mathbf{q}^*$ and add it to \mathbf{q}_r .

The *Barrett hand* consists of three fingers made of two links (a proximal and a distal one). As such, (1) is solved for $j = 1, \dots, 6$, where we first rotate the proximal joints until contact and then proceed with the distal joints. To avoid interpenetration between the object and the gripper, we add an offset ϵ which we heuristically set to 0.5 cm in our experiments.

D. Discriminator network

Since the network does not have any supervision except for the classification task, we need to enforce that the generated grasps are realistic. To this end, we add a Wasserstein discriminator D [26] and train it with the gradient penalty [27].

More specifically, the objective to minimize using the grasp generation module G is

$$\begin{aligned}\mathcal{L}_{disc} &= \mathbb{E} [D(G(E(\mathbf{I}), \mathbf{q}_c, \mathbf{t}_0, \mathbf{r}_0)) - \mathbb{E} [D(\hat{\mathbf{q}}, \hat{\mathbf{t}}, \hat{\mathbf{r}})]], \\ \mathcal{L}_{gp} &= \mathbb{E} \left[\left(\|\nabla_{\tilde{\mathbf{q}}, \tilde{\mathbf{T}}, \tilde{\mathbf{r}}} D(\tilde{\mathbf{q}}, \tilde{\mathbf{t}}, \tilde{\mathbf{r}})\|_2 - 1 \right)^2 \right],\end{aligned}\quad (2)$$

where $\hat{\mathbf{q}}$, $\hat{\mathbf{t}}$, and $\hat{\mathbf{r}}$ are samples from the ground-truth data and $\tilde{\mathbf{q}}$, $\tilde{\mathbf{t}}$, and $\tilde{\mathbf{r}}$ are linear interpolations between predictions and those ground-truth samples.

E. Complementary loss functions

While the discriminator loss \mathcal{L}_{disc} helps in producing realistic looking grasps, it alone is not sufficient to guide the learning problem enough. Therefore, we propose a set of complementary losses.

To ensure that the generated grasps are close to the object, we add a contact loss

$$\mathcal{L}_{cont} = \frac{1}{|V_{cont}|} \sum_{v \in V_{cont}} \min_k \|v, O_k\|_2, \quad (3)$$

where O_k are the object vertices and V_{cont} are vertices on the hand that are often in contact with the object in our ground-truth grasps. We calculate V_{cont} as the vertices that are closer than 5 mm to the object in at least 8% of the ground-truth grasps. These are mainly located on the finger tips and the palm of the hand.

For a grasp to be successful, the gripper should be rotated towards the object of interest. To promote such behaviour, we add a loss function that penalizes the gripper if its approach direction $\hat{\mathbf{a}}$ is pointing away from the vector $\hat{\mathbf{o}}$ connecting the hand to the object's center:

$$\mathcal{L}_{orient} = 1 - \hat{\mathbf{a}}^\top \hat{\mathbf{o}}. \quad (4)$$

Finally, for a grasp to be successful it cannot interpenetrate the object. To encourage such behaviour, we add a loss that penalizes the distance between vertices V_i that are inside the object and the closest object vertex

$$\mathcal{L}_{int} = \frac{1}{|V_i|} \sum_{v \in V_i} A_v \min_k \|v, O_k\|_2, \quad (5)$$

where A_v is the average area of the incident faces of the vertex that is inside the object. Since uniform mesh tessellation cannot usually be assumed in robotics, *e.g.*, Fig. 4b, we add the term A_v to be robust to non-uniform tessellation.

Finally, the total loss is a linear combination of all the separate loss functions $\mathcal{L}_{tot} = w_{class}\mathcal{L}_{class} + w_{disc}\mathcal{L}_{disc} + w_{gp}\mathcal{L}_{gp} + w_{cont}\mathcal{L}_{cont} + w_{int}\mathcal{L}_{int} + w_{orient}\mathcal{L}_{orient}$, where each individual loss contribution is given a corresponding weight. The model is trained end-to-end.

F. Implementation details

The network was implemented in PyTorch 1.5.1. We use a pre-trained ResNet-50 [25] as the image encoder. The model was trained on 30 objects from the YCB object set and for each of them we synthetically rendered 100 novel



Fig. 4: Example of three synthetic RGB images used for training (a), and a grasp generated by our method (b).

viewpoints, some examples are shown in Fig. 4a. The images are resized to 256x256. We trained our networks with a learning rate of $1 \cdot 10^{-4}$ and a batch size of 100. The weights of the loss functions were experimentally set to $w_{class} = 1$, $w_{disc} = 1$, $w_{gp} = 10$, $w_{cont} = 100$, $w_{int} = 100$, $w_{orient} = 1$. The generator is trained once every 5 forward passes to improve the relative quality of the discriminator. We trained the networks for 800 epochs where we linearly reduced the learning rate for the last 400 epochs.

V. EXPERIMENTS AND RESULTS

The three main questions we wanted to answer in the experiments were:

- 1) Is Multi-FinGAN able to generate high quality grasps?
- 2) What are the contributions of the proposed loss functions?
- 3) Is our generative grasp sampler, which is purely trained on synthetic data, able to transfer to real objects?

In order to provide justified answers to these questions, we conducted three separate experiments. In the first experiment (Section V-B) we evaluate grasp quality and hand-object interpenetration in simulation. In the second experiment (Section V-C) we do an ablation study over the proposed loss functions and in (Section V-D) we finally evaluate grasp success rate on real hardware.

A. Dataset

To train our model, we manually generate a dataset of grasps on the YCB objects [28] using a *Barrett hand* in GraspIt! [29]. As previously mentioned, the hand can only attain 7 of the 33 grasp types listed in [23] and we therefore categorize each grasp according to these. As a final step, we generate additional grasps around the symmetry axes of the objects. In total, this amounts to over 4000 labeled grasps.

B. Grasping in Simulation

In the simulated grasping experiment, we evaluate how good our method is at producing high quality grasps that are not interpenetrating the object. We test our model on two different datasets: 33 objects from the YCB object set [28] and 49 objects from the recent EGAD! dataset [30]. The YCB object set contains both object models we trained on, and models that were held out during training; the EGAD! dataset contains completely novel objects.

We benchmark against the simulated annealing planner in GraspIt! [31] that ran for 75000 steps to generate 360 grasp candidates on average. To evaluate the quality of a grasp, we used the ϵ -quality metric which represents the radius of the

TABLE I: Simulation experiment results. \uparrow : higher the better; \downarrow : lower the better.

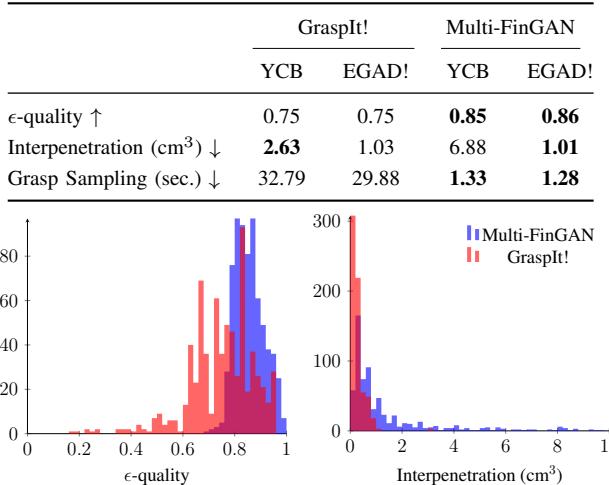


Fig. 5: Histograms showing all results obtained on both datasets by our approach and the baseline in terms of ϵ -quality and interpenetration (best viewed in color).

largest 6D ball centered at the origin that can be enclosed by the convex hull of the wrench space [32]. With our method, we render 5 different viewpoints for each object and 110 grasps per viewpoint. Out of this pool of grasps, we report the average performance on the 10 top-scoring grasps according to the ϵ -metric. We also average the performance of the 10 top-scoring grasps found with the baseline method. In total this amounts to 790 grasps per method.

Table I and Fig. 5 show the simulation results and Fig. 4b an example grasp using our method. To analyze the statistical differences between the methods we used a one sided Wilcoxon signed-rank test. Based on these results we can draw several interesting conclusions. Multi-FinGAN is able to achieve statistically better results ($\alpha = 0.001$) in terms of quality. The histogram in Fig. 5 is also confirming this result, showing that our data-driven grasp planning method is more consistent than the baseline at generating high quality grasps. However, in terms of interpenetration, GraspIt! shows a statistically significant improvement over our method ($\alpha = 0.001$). This result is most likely due to our method reaching a higher interpenetration on the YCB objects because of the presence of large objects which were not in the training-set. EGAD!, on the other hand, contains objects scaled to the size of the hand and our method achieves a low interpenetration on those. One thing to note, though, is that when looking at the interpenetration histogram in Fig. 5, the two methods do not perform radically differently. Finally, our method generates, on average, a grasp in around a second compared to the 30 seconds required by GraspIt!, which makes Multi-FinGAN 30 times faster than the baseline; a difference which is once again statistically significant ($\alpha = 0.001$).

C. Ablation study

To further evaluate the impact the proposed loss functions have on performance, we conducted an ablation study where

TABLE II: Ablation study on EGAD!

Loss removed	none	\mathcal{L}_{int}	\mathcal{L}_{cont}	\mathcal{L}_{orient}	\mathcal{L}_{disc}
ϵ -quality \uparrow	0.86	0.77	0.83	0.85	0.74
Interpenetr. (cm^3) \downarrow	1.01	56.24	3.63	2.99	27.50



Fig. 6: The objects used in the physical experiments.

we trained models with one of the following losses removed: the interpenetration loss \mathcal{L}_{int} , the contact loss \mathcal{L}_{cont} , the orientation loss \mathcal{L}_{orient} , and the discriminator loss \mathcal{L}_{disc} . We train each of these models as was described in Section V-A keeping the same weights and evaluated them on the EGAD! dataset by sampling 110 grasps from 5 different viewpoints and calculating the ϵ -quality and the intersection for the top 10 grasps.

The results are presented in Table II. As expected, a network trained with no interpenetration loss \mathcal{L}_{int} often intersects the object as this results in more contacts but the final ϵ -quality 0.77 is still not higher than 0.86 achieved with the model trained with all the losses.

Another interesting observation is that the model with no contact loss \mathcal{L}_{cont} still achieves a high ϵ -quality. Our hypothesis was that grasps generated with this model would not interpenetrate the object at all as the model would have learned to translate the gripper far from the object which would also have resulted in low quality. However, this was not the case and one possible explanation why is that the orientation and discriminator losses forces the grasps to be realistic and oriented towards the object.

A model trained without the orientation loss \mathcal{L}_{orient} barely impacts the quality of the grasps but does increase the interpenetration compared to the full model. However, this loss speeds up learning in the early stages of training as it acts as an inductive bias forcing the hand to be oriented towards the object.

The network with no discriminator loss \mathcal{L}_{disc} produces the lowest quality grasps. At the same time, it also produces gripper joint-configurations that are physically infeasible.

Overall, all of the models produce grasps with lower ϵ -quality and higher interpenetration compared to a model trained with all the losses. However, the grasp quality does not heavily deteriorate. This is probably due to the inherent power of the finger refinement layer which will always refine the gripper's fingers close to the object if that is possible. Nevertheless, the ablation study shows that all the losses have different purposes and leaving one out affects the final performance of the model. Therefore, we use the model with



Fig. 7: Example grasps proposed by Multi-FinGAN on real objects. The upper row shows the unsegmented input image and the bottom rows shows some grasps on the shape-completed object. The grasp shown in the red box failed as it was in collision with the object.

TABLE III: Real hardware experiment results

	GraspIt!	Multi-FinGAN
Grasp Success Rate (%) ↑	40	60
Shape Completion Time (sec.) ↓	7.7	7.4
Grasp Evaluation Time (sec.) ↓	34.4	1.7

all losses in our following Sim-to-Real experiment.

D. Sim-to-Real Grasp Transfer

To understand if grasps generated with our generative grasp sampler trained on synthetic data, transfer well to real objects, we conducted an experiment on a real *Franka Emika Panda* equipped with a *Barrett hand*. The goal was to grasp the objects shown in Fig. 6 which were chosen as they represent a high variability in both size and shape.

To capture the RGB-D image we used an Intel RealSense D435 camera looking at the scene from the side at a 45 degree viewpoint. For the extrinsic calibration of the camera we used an Aruco marker [33]. To provide our network with an RGB image of only the object, we segment it from the scene by subtracting the background and the table. To create a mesh of the segmented object, which is needed in both our method and the baseline, we used the shape-completion method detailed in [24]. For both methods we generated 20 grasps per object. We then calculated the intersection and quality metric of each grasp. The first physically reachable grasp with lowest intersection and highest quality metric was executed on the real robot. To evaluate if a grasp was successful, the robot had to grasp the object and, without dropping it, move to the start position and rotate the hand $\pm 90^\circ$ around the last joint. If the object was dropped during the manipulation, the grasp was considered unsuccessful.

The result of this experiment is shown in Table III. Based on these numbers we can see that our method reaches a grasp success rate of 60% compared to the baseline 40%, while being over 20 times faster.

An example of the input image fed to the network and a generated grasp is shown in Fig. 1. Although this image is not qualitatively as good as the training data visualized in Fig. 4a the method was still able to generate high quality grasps on such objects showing stable sim-to-real transfer. Additional grasps generated using our method are visualized in Fig. 7.

Based on the experiments, Multi-FinGAN never produced grasps that were too far from the object to be able to grasp

it. The main reason for grasp failure was that grasp ranking based on the ϵ -quality metric favored grasps that established many contact points with the object and, as shown in the leftmost image in Fig. 7, such grasps may not translate to good real-world grasps. Another reason for grasp failure was object scale: on big objects the generated grasps were always in collision with the object, as shown in the rightmost image in Fig. 7. Despite these limitations, the results still indicate stable sim-to-real grasp transfer.

VI. CONCLUSIONS AND FUTURE WORK

We presented Multi-FinGAN, a generative grasp sampling method that produces multi-fingered 6D grasps directly from an RGB-D image. The key insight was to reduce the search space by using a coarse-to-fine grasp generation method where we first generated coarse grasps based on a grasp taxonomy which subsequently were refined using a fully differentiable forward kinematics layer. We compared our model to the well known simulated annealing planner in GraspIt! both in simulation and on a real robot. The results showed that our model trained on synthetic data was significantly better than the baseline in generating higher quality grasps in simulation, and on real hardware it achieved a higher grasp success rate. At the same time it was also 20–30 times faster than the baseline.

Despite the good results, there is still room for improvements. Although the classification into grasp types reduced the search space and eased the learning of the model it requires a large dataset of labeled grasps which is time-consuming to gather. A more elegant solution is to not classify grasps according to a taxonomy but instead regress directly to joint angles allowing to train the model on other datasets such as the Columbia grasp database [34]. Another limitation is the computational time to evaluate the grasps which accounts for more than half the time needed to generate grasps. This time could be reduced by training a critic to evaluate multi-finger grasps but that is still an open problem.

In conclusion the work presented here shows that generating 6D coarse-to-fine multi-fingered grasps is both fast and leads to good grasps. This, in turn, opens the door to use dexterous hands for feedback-based grasping, task informative grasping and grasping in clutter.

REFERENCES

- [1] C. Ferrari and J. F. Canny, "Planning optimal grasps," in *ICRA*, vol. 3, 1992, pp. 2290–2295.
- [2] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, "A survey on learning-based robotic grasping," *Current Robotics Reports*, pp. 1–11, 2020.
- [3] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 2442–2447.
- [4] J. Lundell, F. Verdoja, and V. Kyriki, "Robust Grasp Planning Over Uncertain Shape Completions," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Macau, China: IEEE, Nov. 2019.
- [5] J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergström, D. Kragic, and A. Morales, "Mind the gap-robotic grasping under incomplete observation," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 686–693.
- [6] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dexterous grasping via eigengraps: A low-dimensional approach to a high-complexity problem," in *Robotics: Science and systems manipulation workshop-sensing and adapting to the real world*. Citeseer, 2007.
- [7] E. Corona, A. Pumarola, G. Alenya, F. Moreno-Noguer, and G. Rogez, "Ganhand: Predicting human grasp affordances in multi-object scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5031–5041.
- [8] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11807–11816.
- [9] D. Morrison, J. Leitner, and P. Corke, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.
- [10] A. Mousavian, C. Eppner, and D. Fox, "6-dof grapsnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2901–2910.
- [11] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6232–6238.
- [12] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.
- [13] U. Viereck, A. Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using simulated depth images," in *Conference on Robot Learning*. PMLR, 2017, pp. 291–300.
- [14] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Aparicio, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, July 2017.
- [15] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1357–1364, 2019.
- [20] U. R. Aktas, C. Zhao, M. Kopicki, A. Leonardis, and J. L. Wyatt, "Deep dexterous grasping of novel objects from a single view," *arXiv preprint arXiv:1908.04293*, 2019.
- [16] B. Wu, I. Akinola, A. Gupta, F. Xu, J. Varley, D. Watkins-Valls, and P. K. Allen, "Generative attention learning: a "general" framework for high-performance multi-fingered grasping in clutter," *Autonomous Robots*, pp. 1–20, 2020.
- [17] M. Kokic, J. A. Stork, J. A. Haustein, and D. Kragic, "Affordance detection for task-specific grasping using deep learning," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 91–98.
- [18] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015.
- [19] Q. Lu, M. Van der Merwe, B. Sundaralingam, and T. Hermans, "Multifingered grasp planning via inference in deep neural networks: Outperforming sampling by learning differentiable models," *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 55–65, 2020.
- [21] Q. Lu, M. Van der Merwe, and T. Hermans, "Multi-fingered active grasp learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 8415–8422.
- [22] M. Liu, Z. Pan, K. Xu, K. Ganguly, and D. Manocha, "Generating grasp poses for a high-dof gripper using neural networks," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1518–1525.
- [23] T. Feix, J. Romero, H.-B. Schmidmayer, A. M. Dollar, and D. Kragic, "The grasp taxonomy of human grasp types," *IEEE Transactions on human-machine systems*, vol. 46, no. 1, pp. 66–77, 2015.
- [24] J. Lundell, F. Verdoja, and V. Kyriki, "Beyond top-grasps through scene completion," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 545–551.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [26] S. Martin Arjovsky and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017.
- [27] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.
- [28] 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)*. IEEE, 2015, pp. 510–517.
- [29] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.
- [30] D. Morrison, P. Corke, and J. Leitner, "Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation," *IEEE Robotics and Automation Letters*, 2020.
- [31] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 3270–3275.
- [32] A. T. Miller and P. K. Allen, "Examples of 3d grasp quality computations," in *Proceedings 1999 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1999, pp. 1240–1246.
- [33] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, 2014.
- [34] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 1710–1716.