Adversarial Grasp Objects

David Wang*[1], David Tseng*[1], Pusong Li*[1], Yiding Jiang*[1],
Menglong Guo[1], Michael Danielczuk[1], Jeffrey Mahler[1], Jeffrey Ichnowski[1], Ken Goldberg[1,2]

Abstract— **Learning-based approaches to robust robot grasp planning can grasp a wide variety of objects, but may be prone to failure on some objects. Inspired by recent results in computer vision, we define a class of "adversarial grasp objects that are physically similar to a given object but significantly less "graspable" in terms of a specified robot grasping policy. We present three algorithms for synthesizing adversarial grasp objects under the grasp reliability measure of Dex-Net 1.0 for parallel-jaw grippers: 1) two analytic algorithms that perturb vertices on antipodal faces (one that uses random perturbations and one that uses systematic perturbations), and 2) a deep-learning-based approach using a variation of the Cross-Entropy Method (CEM) augmented with a generative adversarial network (GAN) to synthesize classes of adversarial grasp objects represented by discrete Signed Distance Functions. The random perturbation algorithm reduces graspability by 32%, 12%, and 32% for intersected cylinders, intersected prisms, and ShapeNet bottles, respectively, while maintaining shape similarity using geometric constraints. The systematic perturbation algorithm reduces graspability by 32%, 11%, and 21%; and the GAN reduces graspability by 22%, 36%, and 17%, on the same objects. We use the algorithms to generate and 3D print adversarial grasp objects. Simulation and physical experiments confirm that all algorithms are effective at reducing graspability.**

I. Introduction

Adversarial images [1], [2], [3], [4] are modified images that drastically alter the prediction made by a deep learning classifier while applying minimal perturbation to the original image. This paper defines "adversarial grasp objects," analogous to adversarial images for the domain of robust robot grasping.

Robust robot grasping of a large variety of objects can benefit a diverse range of applications, such as the automation of industrial warehousing and home decluttering. Recent research suggests that robot policies based on deep learning can grasp a wide variety of previously unseen objects [5], [6], [7], [8], but can be prone to failures on objects that may not be encountered during training [9]. In this work, we introduce algorithms to synthesize adversarial grasp objects for the application of examining grasp failure cases on a physical system.

Some adversarial image generation techniques involve performing constrained gradient-based optimization algorithms on the image classification loss [1]. However, it is challenging to apply these algorithms to grasping policies where the grasping performance is generally not a differentiable function of a network output. Instead, the grasp planned by a policy is the result of scoring, ranking, and pruning a set of grasp candidates for each object. To address this, we explore analytic methods and derivative-free optimization. We present three algorithms for synthesizing adversarial objects: two analytic algorithms that modify objects by perturbing vertices on antipodal faces subject to geometric constraints to maintain similarity to the input object, and an algorithm for synthesizing adversarial 3D object models using 3D Generative Adversarial Networks (GANs) [10] and the Cross Entropy Method (CEM) for derivative-free optimization. The third algorithm extends recent advances in GANs to synthesize a 3D Signed Distance Function (SDF) representation for objects that minimizes the quality of available grasps. We note that in this work, an SDF refers to a sampled grid of distances rather than a continuous function. This paper contributes:

1) A formal definition of adversarial grasp objects.

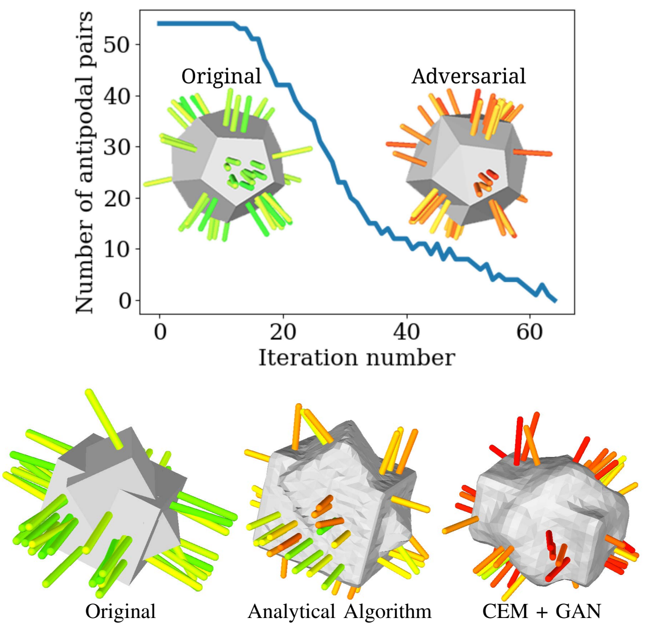

Fig. 1: Original objects vs. adversarial objects. The most robust 25 of 100 parallel-jaw grasps sampled on each object are displayed as grasp axes colored by relative reliability on a linear gradient from green to red. The angle of the friction cone was set to 10°. **Top**: Results from running an analytic algorithm on a dodecahedron mesh for 64 iterations, and a plot of the number of antipodal face pairs vs. the number of iterations of the algorithm. **Bottom row**: Results from applying an analytical algorithm and the CEM + GAN algorithm on a synthetically generated intersected prism.

[1] Dept. of Electrical Engineering and Computer Science;{dmwang, davidtseng, alanpusongli, yiding.jiang, m.guo, mdanielczuk, jmahler, goldberg}@berkeley.edu, jeffi@cs.berkeley.edu
[2] Dept. of Industrial Engineering and Operations Research;
[1,2] The AUTOLab at UC Berkeley (automation.berkeley.edu);
*Authors contributed equally to this work.

2) Two analytic algorithms to synthesize adversarial 3D objects for grasp planning from a given 3D object by performing constrained perturbations of vertices on antipodal faces.
3) A deep learning algorithm based on the Cross Entropy Method (CEM) for derivative-free optimization and deep Generative Adversarial Networks (GANs) that uses an SDF representation of 3D objects to generate a distribution of adversarial objects that look similar to objects from a prior distribution.
4) Simulated and physical experiments studying several categories of adversarial grasp objects generated by the algorithms for the Dexterity Network (Dex-Net) 1.0 grasp planner, which plans parallel-jaw grasps based on a robust quasi-static point contact model [11].

II. Related Work

Adversarial Images. Adversarial images [1], [2], [3], [4] are inputs with a small added perturbation that can change the output of an image classifier, and the problem of finding adversarial images is sometimes formulated as a constrained optimization problem that can be approximately solved using gradient-based approaches [1]. Yang et al. [12] developed a method to perturb the texture maps of 3D shapes such that their projections onto 2D image space can fool classifiers. We build on this line of research by studying adversarial 3D objects for robotic grasping.

Grasp Planning. Grasp planning considers the problem of finding a gripper configuration that maximizes the probability of grasp success. There are several classes of approaches.

Analytic approaches typically assume knowledge of the object and gripper state, and consider the ability to resist external wrenches [13] or constrain the object's motion [14] under perturbations and noises. Examples include GraspIt! [15], OpenGRASP [16], and the Dexterity Network (Dex-Net) 1.0 [11]. To satisfy the assumption of known state, analytic methods typically assume a registration-based perception system: matching sensor data to known 3D object models in the database [17], [18], [19], [20], [21], [22]. However, these systems may not scale well to novel objects and may be computationally expensive during execution.

Empirical approaches [23] use machine learning to develop models that map sensor readings directly to success labels from humans or physical trials. A limitation of empirical methods is that data collection may be time-consuming and error-prone.

Hybrid approaches make use of analytic models to automatically generate large training datasets for machine learning models [24], [25]. Recent results suggest that these methods can be used to rapidly train grasping policies to plan grasps on point clouds that generalize well to novel objects on a physical robot [8], [9], [26]. In this paper, we consider synthesizing adversarial 3D objects for the analytic supervisor used to train these hybrid grasp planning methods.

Generative Models. Deep generative models map a simple distribution (e.g., Gaussian) to a much more complex distribution, such as natural images. Common deep generative models fall into likelihood-based models [27], [28] and implicit likelihood-free models (e.g., Generative Adversarial Networks (GANs) [10]). During training of a GAN, a discriminator tries to distinguish the generated samples apart from the samples from the real data while a generator tries to generate samples to confuse the discriminator. Generative models have also been previously used in the domain of robot grasping, where Veres et al. [29] used conditional generative models to synthesize grasps from RGB-D images, and Bousmalis et al. [26] used GANs for simulation-to-reality transfer learning. There have also been some applications of deep generative models for 3D data. Some notable works in this area include the 3D-GAN work by Wu et al. [30], which uses a GAN to generate 3D reconstruction from an image, and the signed distance-based object generation by Jiang et al. [31], where different frequency components are generated by two separate networks. Mousavian et al. [32] use a variation autoencoder to map a partially observed point cloud to grasps. We expand on these directions by incorporating recent advances in GANs for images.

III. Problem Statement

A. Adversarial Grasp Objects

Let $\mathcal{X}$ be the set of possible states of all 3D objects, where the state consists of a 3D triangular mesh of the object and its pose. Let π be a robot grasping policy mapping a 3D object $\mathbf{x} \in \mathcal{X}$ to a grasp action $\mathbf{u}$. In this work, we only consider a parallel-jaw grasping policy. We assume that the policy can be represented as:

$$\pi(\mathbf{x}) \triangleq \underset{\mathbf{u} \in \mathcal{U}(\mathbf{x})}{\operatorname{argmax}}\ Q(\mathbf{x}, \mathbf{u})$$

where $\mathcal{U}(\mathbf{x})$ denotes the set of all reachable and collision-free (from any angle) grasps on $\mathbf{x}$, and Q is a quality function measuring the reliability or probability of success for a candidate grasp $\mathbf{u}$ on object $\mathbf{x}$.

We introduce the "graspability" $g_\gamma(\mathbf{x}, Q)$ of $\mathbf{x}$ as an approximation of π to measure of how well the policy can robustly grasp the object, taking into account grasps occluded by environment. To do this, we define graspability by the γ-percentile of grasp quality [33]:

$$g_\gamma(\mathbf{x}, Q) \triangleq \mathbb{P}_\gamma(\{Q(\mathbf{x}, \mathbf{u}) \mid \forall\, \mathbf{u} \in \mathcal{U}(\mathbf{x})\})$$

We then consider the problem of generating an adversarial grasp object: a 3D object that reduces graspability under a grasping policy with constrained changes to the input geometry. Let $\sigma(A, B)$ for subsets $A, B \subset \mathcal{X}$ be a binary-valued shape-similarity constraint between the two subsets of objects. We study the following optimization problem, which defines an adversarial grasp object $\mathbf{x}^*$:

$$\mathbf{x}^* \triangleq \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmin}}\ g_\gamma(\mathbf{x}, Q) \text{ subject to } \sigma(\{\mathbf{x}\}, S) = 1 \qquad \text{(III.1)}$$

where $S \subset \mathcal{X}$ is a subset of objects to which the generated object should be similar.

B. Robust Grasp Analysis

In this paper, we optimize adversarial examples with respect to the Dexterity Network (Dex-Net) 1.0 grasping policy [11]. Here, $\mathcal{U}(\mathbf{x})$ is a set of antipodal points on the object surface that correspond to a reachable grasp, where a pair of opposite contact points (v_1, v_2) are antipodal if the line between (v_1, v_2) lies entirely within the friction cones [11]. The quality function Q measures the robust wrench resistance which is the ability of a grasp to resist a target wrench under perturbations to the object pose, gripper pose, friction, and wrench under a soft-finger point contact model [9].

When calculating g, both the reward and policy are based on the Dex-Net 1.0 robust grasp quality metric and the associated maximal quality grasping policy. Within the Dex-Net 1.0 robust quality metric, $Q(\mathbf{x}, \mathbf{u})$ is defined as:

$$Q(\mathbf{x}, \mathbf{u}) \triangleq \mathbb{E}_{\mathbf{u'} \sim p(\cdot|\mathbf{u}), \mathbf{x'} \sim p(\cdot|\mathbf{x})}[R(\mathbf{x'}, \mathbf{u'})]$$

where $p(\mathbf{u'}|\mathbf{u})$ and $p(\mathbf{x'}|\mathbf{x})$ denote conditional distributions over $\mathbf{x}$ and grasp $\mathbf{u}$, and R measures grasp quality if the grasp is executed with zero uncertainty in object and gripper pose. We use the epsilon metric by Ferrari and Canny [34].

To calculate $g_\gamma(\mathbf{x}, Q)$ in practice, we uniformly sample a constant number of antipodal grasps on the object. We take N samples from the object and grasp pose distributions $p(\mathbf{u'}|\mathbf{u})$ and $p(\mathbf{x'}|\mathbf{x})$ and estimate the robust quality $Q(\mathbf{x}, \mathbf{u})$ by the sample mean $\hat{Q}(\mathbf{x}, \mathbf{u})$ for each grasp:

$$\hat{Q}(\mathbf{x}, \mathbf{u}) = \frac{1}{N} \sum_{i=1}^{N} R(\mathbf{x}_i, \mathbf{u}_i)$$

The empirical graspability $\hat{g}_\gamma(\mathbf{x}, Q)$ is estimated by taking the discrete γ-percentile of $\hat{Q}(\mathbf{x}, \mathbf{u})$ for all sampled grasps.

IV. Analytical Methods: Constrained Vertex Perturbation

In order to decrease the graspability of an object, we first consider an analytic approach to modify an existing 3D triangular mesh. Let the mesh $\mathbf{x}$ be specified by a set of vertices $\mathcal{V} = \{v_1, v_2, \dots v_n\} \subset \mathbb{R}^3$ and a set of faces $\mathcal{F} = \{f_1, f_2, \dots f_m\}$, where each face f_i is the triangle defined by three distinct elements of $\mathcal{V}$. Also, let $F_a = \{(f_i, f_j), \dots (f_p, f_q)\}$ be the set of pairs of antipodal faces, face pairs that contain a pair of antipodal points. Let the unit normal of face f_i be denoted by $\mathbf{n_i} \in \mathbb{S}^2$. Finally, let the antipodality angle φ between two faces be defined as $\varphi(f_i, f_j) = \arccos(-\mathbf{n_i}^T \mathbf{n_j})$.

A. Case Study: Simple Shapes

Dex-Net 1.0's graspability metric specifically considers the robustness of a parallel jaw grasp, which requires antipodal point pairs and can be susceptible to small pose variations. Thus, we consider the following iterative algorithm for analytically perturbing vertices to reduce the number of antipodal point pairs. Intuitively, we are attempting to decrease $|F_a|$, the number of antipodal faces by increasing φ between all pairs of antipodal faces until φ reaches the angle of the friction cone. In each iteration, we compute F_a. For each vertex v of each face in F_a (i.e., all vertices incident to a face in F_a), we consider perturbations in directions $\mathcal{W}$, a set of 6 randomly selected unit vectors. We then test perturbations $v' = v + \delta \mathbf{w}$ for each $\mathbf{w} \in \mathcal{W}$, where $\delta \in \mathbb{R}^+$ is a constant. We select the v' that maximizes $\sum_{i \in F_a} \varphi_i$, the sum of the antipodality angles between all antipodal pairs in F_a that contain a face that is incident to v. Results from applying this algorithm to a sample dodecahedron mesh to systematically decrease the number of antipodal faces can be seen in Fig. 1.

B. Sampling-Based Random Perturbation Algorithm

Since it is computationally expensive to run the previous algorithm on complex meshes with thousands of faces and vertices, we propose a sampling-based version of the above algorithm to avoid the overhead of computing the full set of antipodal faces. Consider the same mesh $\mathbf{x}$ as above. We want to perturb vertices while constraining the movement such that the surface normals of adjacent faces do not deviate by more than some angle α. This corresponds to the shape similarity constraint σ in Equation III.1, and in this case, $S = \{\mathbf{x}\}$, the original object itself.

In each iteration, we sample a pair of antipodal faces f_1 and f_2. We then randomly sample one of the vertices v_k of f_1 and f_2. Again, we consider a set of 6 directions $\mathcal{W}$, and for each direction $\mathbf{w} \in \mathcal{W}$, we compute the perturbation $\delta_w \in \mathbb{R}^+$ such that the antipodality angle φ between faces f_1 and f_2 is maximized subject to the shape similarity constraint σ when v_k is moved to $v_k + \delta_w \mathbf{w}$. Then, we take the minimum perturbation δ_w found along each of the 6 directions as the actual perturbation. By constraining the perturbations such that the shape similarity constraint is still satisfied after each step, the algorithm attempts to maintain local similarity of the region of perturbation while decreasing the graspability.

C. Antipodal Rotation Algorithm

An alternative to the method described in Sec. IV-B is to analytically rotate faces with the smallest angle necessary to make the antipodality angle between pairs of faces larger than the angle of the friction cone. To encourage the algorithm to make large structural changes instead of only local perturbations, if the mesh has more than d faces, we decimate the mesh until it has less than d faces. Then, each iteration of the algorithm uniformly samples a pair of antipodal faces f_1 and f_2. Let φ be the antipodality angle between f_1 and f_2, and let $\theta_{perturb} = \beta(\psi - \varphi)$, where ψ is the friction cone angle. Then we rotate f_1, f_2 by $\theta_{perturb}, -\theta_{perturb}$ respectively in a direction orthogonal to n_1 and n_2, where the sign of $\theta_{perturb}$ is chosen to maximize the new antipodality angle between f_1 and f_2. We note that rotating a face means applying the same rotation matrix to each of the vertices on the face. The tunable parameter $\beta \in (0, 1]$ determines the strength of the shape similarity constraint.

V. Deep Learning Algorithm: CEM + GAN

In this section, we describe a data-driven approach to generating adversarial grasp objects. As opposed to the ana-

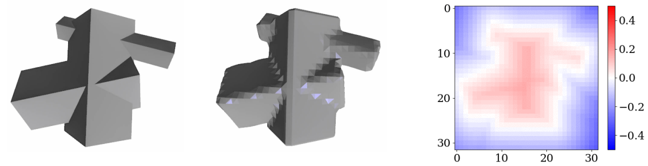

Fig. 2: Example result after converting a mesh to an SDF. The general shape of the mesh is preserved, and the conversion artifacts are due to the finite resolution of the SDF. Left: Original mesh. Middle: SDF after remeshing. Right: A sample cross section of the SDF.

lytical algorithm, which generates an adversarial version of an existing object, the cross-entropy method and generative adversarial network (CEM + GAN) algorithm takes as input a set $S \subset \mathcal{X}$ of objects and can output a set of generated objects similar to those in S. The following sections describe different aspects and components of the algorithm.

A. Signed Distance Generative Adversarial Network

Generative adversarial networks (GANs) [10] are a family of powerful implicit generative models that have demonstrated remarkable capabilities in generating high-quality samples with low inference complexity. In this algorithm, we train a GAN to obtain a generative model from which we can sample objects that are similar to the input data.

We use the Signed Distance Function (SDF) [35] as a representation for generating 3D geometry. SDFs are widely used in applications such as rendering, segmentation, or collision checking. In this work, an SDF is parameterized as a discrete, sampled grid of distances rather than a continuous function. An example is shown in Fig. 2. We note that the SDFs produce some artifacts due to the finite resolution. The SDF of a closed object $\mathbf{x}$ at a point v can be given as:

$$f(v) = \begin{cases} d(v, \partial\mathbf{x}), & \text{if } v \in \mathbf{x} \\ -d(v, \partial\mathbf{x}), & \text{if } v \notin \mathbf{x} \end{cases} \tag{V.1}$$

where $\partial\mathbf{x}$ denotes the boundary of $\mathbf{x}$, and d is the Euclidean distance from the closest boundary to a point. In this work, we use compact watertight meshes that have well-defined SDFs.

We draw on techniques used in Spectral-Normalization GAN (SNGAN) [36], which can generate high-fidelity images, and apply them to generate SDFs. We denote the standard Gaussian noise vector as $\mathbf{z} \in \mathbb{R}^{200}$ drawn from p_z, the empirical distribution defined by training data as p_{data}, the Generator as $G : \mathbb{R}^{200} \to \left[-1, 1\right]^{32\times32\times32}$, and the Discriminator as $D : \left[-1, 1\right]^{32\times32\times32} \to \mathbb{R}$. The output of the generator and input for the discriminator are SDFs. For the training objective, we use the hinge version of adversarial loss [37] as we found that it stabilizes training. The GAN objective is then

$$\mathcal{L}_D^{data} = -\mathbb{E}_{\mathbf{x}\sim p_{data}(\cdot)}[\min(0, -1 + D(\mathbf{x}))]$$

$$\mathcal{L}_D^{gen} = -\mathbb{E}_{\mathbf{z}\sim p_z(\cdot)}[\min(0, -1 - D(G(\mathbf{z})))]$$

$$\mathcal{L}_D = \mathcal{L}_D^{data} + \mathcal{L}_D^{gen}, \quad \mathcal{L}_G = -\mathbb{E}_{\mathbf{z}\sim p_z(\cdot)}[D(G(\mathbf{z}))]$$

Let Θ represent the space of parameter weights for the GAN, which maps a multivariate Gaussian distribution to $p_\theta(\mathbf{x})$, a probability distribution over objects $\mathcal{X}$ parameterized by some $\theta \in \Theta$. We can formulate a similar objective to Equation III.1, but instead optimizing for a distribution of objects that we want to be similar to some prior subset $S \subset \mathcal{X}$:

$$\theta^*(Q) = \arg\min_{\theta\in\Theta} \mathbb{E}_{\mathbf{x}\sim p_\theta(\cdot)}[g_\gamma(\mathbf{x}, Q)] \text{ subject to } \sigma(\mathcal{X}_\theta, S) = 1, \tag{V.2}$$

where $\mathcal{X}_\theta \subset \mathcal{X}$ is the support of the probability distribution p_θ for some parameter $\theta \in \Theta$.

B. Optimization via Resampling

One challenge in performing the optimization in Equation V.2 is that the graspability function $g_\gamma(\mathbf{x}, Q)$ is not differentiable; therefore, we need to perform derivative-free optimization by querying the function with different inputs and adjust the model parameters based on the responses of the function. The cross-entropy method [38] is an adaptive derivative-free optimization algorithm that has been widely applied. We are interested in finding the distribution of rare events that minimize a real-valued grasp quality function $g_\gamma(\mathbf{x}, Q)$ over $\mathcal{X}$.

As a starting point, the GAN is initialized with a prior distribution of objects $S \subset \mathcal{X}$ so that it generates objects similar in shape. We start by training the GAN on this prior set of objects. Then, in a resampling step, we use the GAN to generate objects and take a subset of the objects with the lowest graspability to use as training data to retrain the GAN. We continue alternating between training and resampling steps for a number of iterations.

In an independent work, Gupta et al. [39] apply similar techniques to optimize functions over genetic sequences with a GAN by feeding samples with desired properties back into the GAN to generate more sequences with the properties. This suggests that the techniques we use may be general and can potentially be extended to broader applications.

C. Shape Similarity

We now discuss the shape similarity constraint σ in Equation V.2 for the CEM + GAN algorithm. Let P_θ be the distribution over $\mathcal{X}$ induced by the model with parameter θ and P_S be the distribution empirically defined by S. We then define the shape similarity constraint $\sigma(\mathcal{X}_\theta, S)$ in Objective III.1 as $\mathrm{D}_{\mathrm{KL}}(P_S || P_\theta) < \epsilon$, where D_{KL} is the Kullback-Leibler divergence between two distributions, and $\epsilon > 0$ is a hyperparameter that can be controlled through the sampling percentile γ (smaller γ means more similar distributions). The GAN loss function implicitly enforces this shape similarity constraint as it has been shown that at the global optimum, the KL-divergence between the generated distribution and the original distribution is zero [37]. We note that the ϵ in the shape similarity constraint is necessary, since GANs do not usually reduce the loss to 0 in practice and we use multiple resampling iterations.

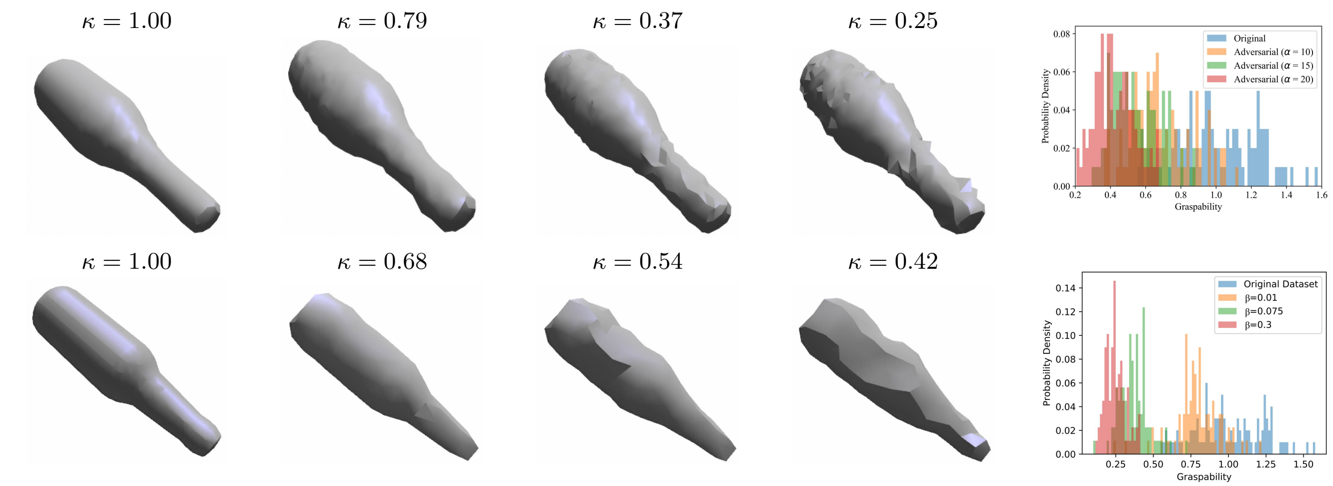

Fig. 3: Results from the analytical random perturbation algorithm and the analytical antipodal rotation algorithm. Both algorithms are able to decrease graspability on objects from all three datasets. We show the progression of an example from the bottles dataset as we increase the perturbation parameters from each algorithm. The metric κ is the mean normalized graspability of the generated dataset for the level of the perturbation parameter, where the graspability is the empirical 75^{th} percentile of samples from the grasp quality function. The histograms on the right show the graspability of all the objects. Top row: Analytical Algorithm. From left to right: original object, then perturbed versions using the surface normal constraint with $\alpha = 10$, $\alpha = 15$, and $\alpha = 20$, respectively. Bottom row: Antipodal Rotation Algorithm. From left to right: original object, then perturbed versions using the perturbation parameter $\beta = 0.01$, $\beta = 0.075$, and $\beta = 0.3$. The objects have been smoothed for visualization purposes with OpenGL smooth shading.

VI. Experiments

To test the three algorithms ability to reduce graspability on known objects, we run the algorithms on two synthetic datasets as well as on the ShapeNet [40] bottles category. To have all algorithms take in meshes or SDFs with similar resolution, we first converted all three datasets to SDFs. These SDFs are directly consumed by the GAN algorithm, while they are remeshed for the analytical algorithms. For the synthetic datasets, we used the process presented by Bousmalis et al. [26]. The intersected cylinders dataset consists of one large central cylinder with two smaller cylinders randomly grafted onto it. The intersected prisms dataset is similar but uses rectangular prisms instead. We vary the sizes of all three prisms. The bottle, cylinder, and prism datasets have averages of 1,391 vertices and 2,783 faces, 1,202 vertices and 2,400 faces, and 2,731 vertices and 4,739 faces, respectively, and have 479, 1000, and 1000 total objects, respectively. Examples from each of these datasets are shown in Fig. 4.

In the following experiments, we set friction coefficient to be 0.5. For the graspability metric $g(\mathbf{x}, Q)$, we chose $\gamma = 75\%$: often, one of the top 25% of grasps is accessible, so we choose to look at the worst case from this set. Consider a set of n generated objects $\{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n\} \subset \mathcal{X}$ from a prior dataset of objects. We define mean normalized graspability as $\kappa = c \cdot \frac{1}{n} \sum_{i=1}^{n} g_\gamma(\mathbf{x}_i, Q)$, where c is a normalizing constant. We note that the objects in the figures in the section have been smoothed for visual clarity to demonstrate the behavior of the algorithms, but the metrics represent the results of the objects without smoothing.

A. Random Perturbation Algorithm

We run the analytical algorithm for local perturbations of vertices on antipodal faces on 100 objects from each of the three datasets. We experimented with α values of 10, 15, and 20 degrees for the shape similarity constraint for maximum deviation in surface normals described in Section IV-B. We find that the analytical algorithm decreases the graspability metric for all datasets. With a value of $\alpha = 10$ degrees, the mean normalized graspability is decreased by 32% on the intersected cylinders dataset, 12% on the intersected prisms dataset, and 32% on the ShapeNet bottles datset. At each level of α, we observe that the objects from the prism dataset have the highest graspability; we conjecture that it is difficult to decrease the antipodality of large, prism surfaces with only local perturbations. Sample object examples along with their adversarial versions, the associated graspability metrics, and the distribution of graspability metrics before and after applying the analytical algorithm are shown in Fig. 3. Increasing α decreases the graspability at the cost of geometric similarity to the original object.

B. Antipodal Rotation Algorithm

We run the antipodal rotation algorithm on 100 objects from each of the three datasets. We use $d = 350$, and number of iterations equal to the number of faces of the mesh after decimation. $d = 350$ was chosen because for this number of faces, the decimated meshes appear similar to the original meshes by visual inspection. We experimented with $\beta = 0.01, 0.075, 0.3$ for the shape similarity constraint described in Section IV-C. We find that the antipodal rotation algorithm decreases the graspability metric for all datasets. With a value of $\beta = 0.01$, the mean normalized graspability is decreased by 32% on the intersected cylinders dataset, 11% on the intersected prisms dataset, and 21% on the ShapeNet bottles datset. At each level of β, we observe that the objects from the prism dataset have the highest graspability, similar to the other analytical algorithm above. Object examples and metrics are shown in Fig. 3. Increasing β decreases the graspability at the cost of geometric similarity to the

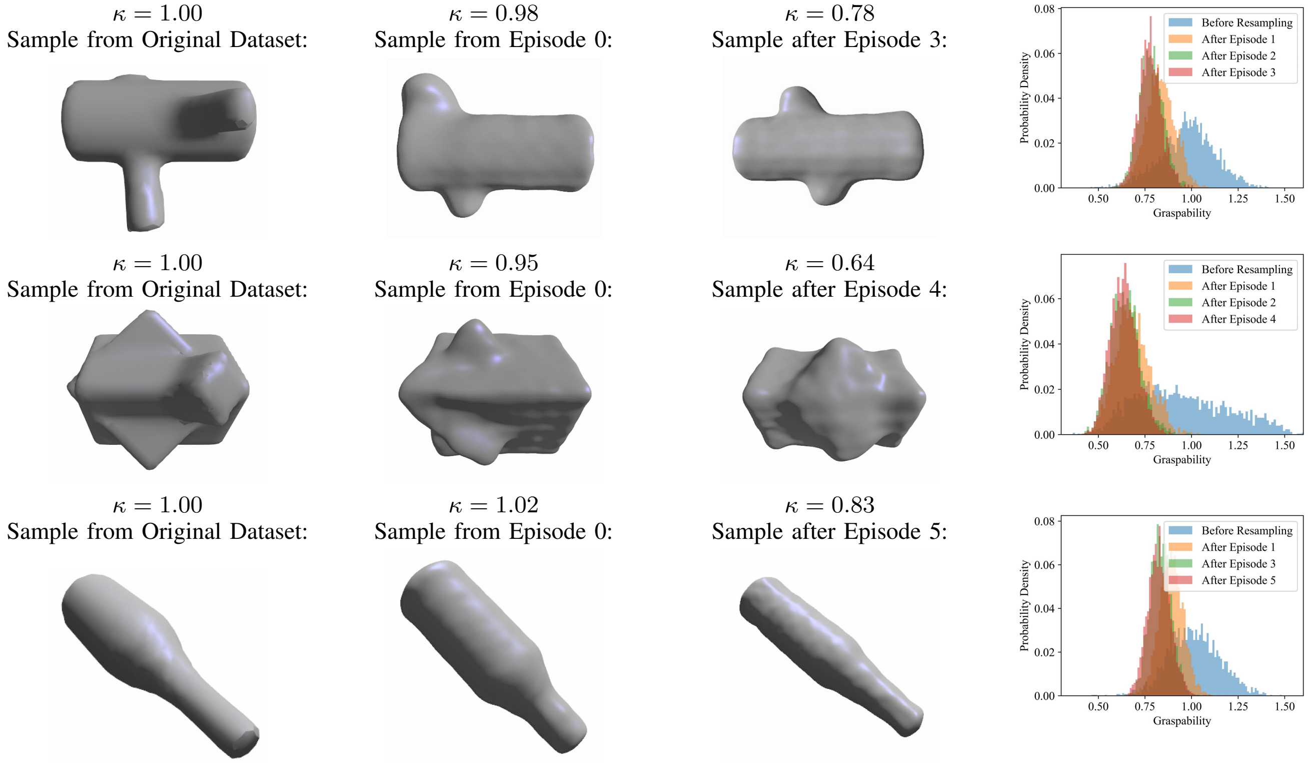

Fig. 4: CEM + GAN Algorithm. The images on the left are example objects from the GAN output distribution as the resampling progresses. "Original" means the original SDF dataset, "Episode 0" denotes the GAN trained on the prior dataset (the first GAN trained, or Episode 0), and "Episode n" denotes the n^{th} GAN trained excluding the first. The κ values are the mean normalized graspabilities over a set of 100 objects generated during the corresponding stage in the training, where the graspability is the empirical 75^{th} percentile of samples from the grasp quality function. The right image is the histogram showing the overall distribution of the graspability metric (normalized to the mean graspability Episode 0) on the GAN output distribution as resampling progresses. As the algorithm progresses through the episodes, the probability mass shifts towards lower graspability. The objects have been smoothed for visualization purposes with OpenGL smooth shading.

original object, corresponding to an increasingly relaxed shape similarity constraint.

C. CEM + GAN Algorithm

We train the resampling GAN on all three datasets. The datasets are preprocessed into signed distance field format with stride 0.03125 after being scaled such that the entire set has bounding boxes of approximately $1 \times 1 \times 1$.

For all three datasets, we sample 2500 new objects and keep 500 with the lowest graspability, and train the GAN for 16000 iterations between resampling steps. Resampling in all experiments rejects output grids that produce non-watertight meshes after remeshing, as producing meshes with non-orientable faces, gaps, self-intersection, or disjoint pieces is not desirable when generating a distribution of 3D objects. Such outputs are possible because the GAN does not explicitly enforce such constraints, but this rejection rate is very low: for bottles, no grids were rejected in any resampling iteration, and on the intersected sets, rejection rate remained below 10% in all episodes.

Examples of objects from the GAN output distributions and histograms showing the overall distribution of graspability over resampling episodes are shown in Fig. 4. After 3 resampling iterations on the intersected cylinders dataset, the mean normalized graspability is reduced by 22% relative to objects in the original dataset. Similarly, graspability is reduced by 36% on the intersected prisms dataset after 4 resampling iterations and by 17% on the ShapeNet bottles dataset after 5 resampling iterations.

D. CEM + GAN Failure Modes

In general, CEM + GAN can produce objects with richer geometric variations compared to analytical methods; however, since resampling decreases diversity of objects in the samples as objects with similar metric scores tend to have similar geometry, complete mode collapse, the phenomenon where a GAN outputs one distinct object regardless of the input [41], tends to occur after enough resampling episodes. We observed mode collapse by the 9^{th} iteration on all three datasets.

We experimented with several variations of the GAN architecture and observed that removing spectral normalization can lead to more diverse objects on the intersected cylinders dataset. In this experiment, mode collapse does not occur before the metric quality mean stops improving, reaching a decrease of 83% from the original dataset. However, these generated objects can deviate more significantly from the prior dataset. Some examples are shown in Section VII.

VII. Physical Experiments

A. Objects

We experiment with several types of objects. First, we consider a unit cube, a highly graspable object. We manually optimized an adversarial cube with minimal perturbation such that any pair of faces of the adversarial object has an antipodality angle of at least φ degrees by moving the midpoint on three adjacent faces of the cube by $\frac{\tan\varphi}{2}$ in the direction of the surface normal of the original face.

We also consider a cuboctahedron, a polyhedron with 6 square faces and 8 triangular faces. As it is difficult to manually design an adversarial version satisfying the property that all pairs of faces have an antipodality angle of at least φ, we used a modification of the analytical algorithm described in Section IV-A. We apply random perturbations until the property above is satisfied, rejecting perturbations that introduce concavities to the objects.

We also 3D printed two of the adversarial intersected cylinders objects generated using the alternative GAN architecture described in Section VI-D. We chose this class of objects to explore in physical trials, since it was the output generated by the CEM + GAN algorithm with the lowest graspability in simulation.

B. Experimental Setup

We used a custom parallel jaw gripper in which each jaw simulates a point contact using a small metal bearing (Fig. 5). The friction angle with this gripper and the printed objects is approximately 17°.

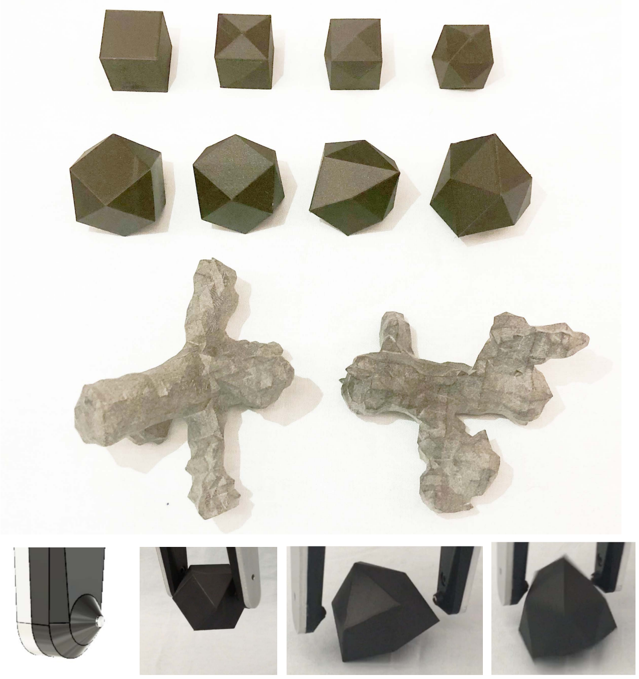

Fig. 5: Top: 3D printed objects with nylon plastic material. We use cubes and cuboctahedrons with $\varphi = 0, 10, 15, 26$ and adversarial intersected cylinder objects generated by the CEM + GAN algorithm. Bottom (left to right): Gripper design with metal ball bearings to simulate point contacts, successful grasp of the original cuboctahedron, attempted grasp on adversarial cuboctahedron ($\varphi = 26$), failure on adversarial cuboctahedron ($\varphi = 26$) in which the object rolls forward after slipping out of the gripper.

In physical trials, we first compute the stables poses of each object and potential grasps with their associated confidences using the Dex-Net 1.0 [11] analytic robustness model. On the physical system, we sample 5 grasps and execute each 3 times. To sample the grasps, we first sample a stable pose, using probabilities for each stable pose proportional to the computed feasibility probability of the stable pose. We then take the grasp with the highest confidence of success that has not yet been executed. When executing the grasp, we place the object in a bin, use a depth sensor to obtain a point cloud of the object, and then align it to the known 3D model of the object using the Super4PCS algorithm [42]. A grasp is successful if the robot arm is able to lift the object out of the bin and transport it to an adjacent bin.

C. Results

We conducted a total of 15 trials for each of the ten objects (Table I). The original cube is successfully grasped in all trials, and the success rate decreases as we increase the adversarial friction angle; the adversarial cube of 26 degrees is never successfully grasped. We observe a similar trend for the cuboctahedrons. The adversarial intersected cylinder objects generated by the CEM + GAN method are very difficult to grasp, as only we observed success in only 3 of 30 total trials for the two objects.

Object	Pred.	Actual
Original Cube	100.0	100.0
Adversarial Cube (10°)	100.0	80.0
Adversarial Cube (15°)	100.0	13.3
Adversarial Cube (26°)	0.0	0.0
Original Cuboctahedron	100.0	100.0
Adversarial Cuboctahedron (10°)	100.0	60.0
Adversarial Cuboctahedron (15°)	100.0	13.3
Adversarial Cuboctahedron (26°)	0.0	0.0
Adversarial Intersected Cylinder 1	0.0	13.3
Adversarial Intersected Cylinder 2	0.0	6.7

TABLE I: Physical Experiment Results. 5 grasps with 3 trials for each of 10 objects showing predicted and actual grasp success rates. Lower success rates for cases where the adversarial friction angle is close to the real friction angle may be due to small calibration errors.

VIII. Discussion and Future Work

We introduce adversarial grasp objects: objects that are geometrically similar to a given object, but decrease the predicted graspability, and present three algorithms that generate adversarial grasp objects.

The Dex-Net 1.0 graspability metric [11] models point contacts instead of area contacts, which can be disproportionately affected by surface roughness. In future work, we will explore extensions to different gripper types and to suction grasps. In computer vision, adversarial images can be used to train more robust neural network image classifiers. Similarly, we conjecture that adversarial grasp objects can be used to train more robust learning-based robot grasping policies.

Acknowledgement

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, Berkeley Deep Drive (BDD), the Real-Time Intelligent Secure Execution (RISE) Lab, and the CITRIS "People and Robots" (CPAR) Initiative. The authors were supported in part by donations from HP Inc., Siemens, Google, Toyota Research Institute, Autodesk, Honda, Intel, Comcast, and by equipment grants from PhotoNeo, NVidia, and Intuitive Surgical. We thank Keith Moore and Tony Lewis of HP Inc. for printing the CEM + GAN objects and our colleagues who provided helpful feedback, code, and suggestions, in particular Ajay Tanwani, Michael Laskey, Sanjay Krishnan, Matt Matl, Richard Liaw, and Daniel Seita.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013. [Online]. Available: http://arxiv.org/abs/1312.6199

[2] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against deep learning systems using adversarial examples," *CoRR*, vol. abs/1602.02697, 2016. [Online]. Available: http://arxiv.org/abs/1602.02697

[3] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *CoRR*, vol. abs/1607.02533, 2016. [Online]. Available: http://arxiv.org/abs/1607.02533

[4] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," *CoRR*, vol. abs/1707.07397, 2017. [Online]. Available: http://arxiv.org/abs/1707.07397

[5] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. Journal of Robotics Research (IJRR)*, vol. 34, no. 4-5, pp. 705–724, 2015.

[6] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. Journal of Robotics Research (IJRR)*, vol. 37, no. 4-5, pp. 421–436, 2018.

[7] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2016.

[8] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proc. Robotics: Science and Systems (RSS)*, 2017.

[9] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," in *Proc. Advances in Neural Information Processing Systems*, 2014.

[11] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg, "Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2016.

[12] D. Yang, C. Xiao, B. Li, J. Deng, and M. Liu, "Realistic adversarial examples in 3d meshes," *arXiv preprint arXiv:1810.05206*, 2018.

[13] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer handbook of robotics*. Springer, 2008, pp. 671–700.

[14] A. Rodriguez, M. T. Mason, and S. Ferry, "From caging to grasping," *Int. Journal of Robotics Research (IJRR)*, p. 0278364912442972, 2012.

[15] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2009, pp. 1710–1716.

[16] B. León, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales, T. Asfour, S. Moisio, J. Bohg, J. Kuffner, *et al.*, "Opengrasp: a toolkit for robot grasping simulation," in *Proc. IEEE Int. Conf. on Simulation, Modeling, and Programming of Autonomous Robots (SIMPAR)*. Springer, 2010, pp. 109–120.

[17] P. Brook, M. Ciocarlie, and K. Hsiao, "Collaborative grasp planning with multiple object representations," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2011, pp. 2851–2858.

[18] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Şucan, "Towards reliable grasping and manipulation in household environments," in *Experimental Robotics*. Springer, 2014, pp. 241–252.

[19] C. Goldfeder and P. K. Allen, "Data-driven grasping," *Autonomous Robots*, vol. 31, no. 1, pp. 1–20, 2011.

[20] C. Hernandez, M. Bharatheesha, W. Ko, H. Gaiser, J. Tan, K. van Deurzen, M. de Vries, B. Van Mil, J. van Egmond, R. Burger, *et al.*, "Team delft's robot winner of the amazon picking challenge 2016," *arXiv preprint arXiv:1610.05514*, 2016.

[21] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*. IEEE, 2011, pp. 858–865.

[22] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2013, pp. 4263–4270.

[23] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesisa survey," *IEEE Trans. Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[24] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4461–4468.

[25] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *Int. Journal of Robotics Research (IJRR)*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[26] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4243–4250.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[28] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *CoRR*, vol. abs/1601.06759, 2016. [Online]. Available: http://arxiv.org/abs/1601.06759

[29] M. Veres, M. Moussa, and G. W. Taylor, "Modeling grasp motor imagery through deep conditional generative models," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 757–764, 2017.

[30] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1912–1920.

[31] C. Jiang, P. Marcus, *et al.*, "Hierarchical detail enhancing mesh-based shape generation with 3d generative adversarial network," *arXiv preprint arXiv:1709.07581*, 2017.

[32] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," *arXiv preprint arXiv:1905.10520*, 2019.

[33] J. Mahler, B. Hou, S. Niyaz, F. T. Pokorny, R. Chandra, and K. Goldberg, "Privacy-preserving grasp planning in the cloud," in *Proc. IEEE Conf. on Automation Science and Engineering (CASE)*. IEEE, 2016, pp. 468–475.

[34] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 1992, pp. 2290–2295.

[35] S. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces*. Springer Science & Business Media, 2006, vol. 153.

[36] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *CoRR*, vol. abs/1802.05957, 2018. [Online]. Available: http://arxiv.org/abs/1802.05957

[37] J. H. Lim and J. C. Ye, "Geometric gan," *arXiv preprint arXiv:1705.02894*, 2017.

[38] R. Y. Rubinstein, "Optimization of computer simulation models with rare events," *European Journal of Operational Research*, vol. 99, no. 1, pp. 89 – 112, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221796003852

[39] A. Gupta and J. Zou, "Feedback gan (fbgan) for dna: a novel feedback-loop architecture for optimizing protein functions," *arXiv preprint arXiv:1804.01694*, 2018.

[40] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[41] H. Thanh-Tung, T. Tran, and S. Venkatesh, "On catastrophic forgetting and mode collapse in generative adversarial networks," *arXiv preprint arXiv:1807.04015*, 2018.

[42] N. Mellado, D. Aiger, and N. J. Mitra, "Super 4pcs fast global pointcloud registration via smart indexing," in *Computer Graphics Forum*, vol. 33, no. 5. Wiley Online Library, 2014, pp. 205–215.