Autonomous Robots
https://doi.org/10.1007/s10514-020-09907-y

Generative Attention Learning: a “GenerAL” framework for high-performance multi-fingered grasping in clutter

Bohan Wu[1] · Iretiayo Akinola[1] · Abhi Gupta[1] · Feng Xu[1] · Jacob Varley[2] · David Watkins-Valls[1] · Peter K. Allen[1]

Received: 17 September 2019 / Accepted: 3 February 2020

Abstract

Generative Attention Learning (GenerAL) is a framework for high-DOF multi-fingered grasping that is not only robust to dense clutter and novel objects but also effective with a variety of different parallel-jaw and multi-fingered robot hands. This framework introduces a novel attention mechanism that substantially improves the grasp success rate in clutter. Its generative nature allows the learning of full-DOF grasps with flexible end-effector positions and orientations, as well as all finger joint angles of the hand. Trained purely in simulation, this framework skillfully closes the sim-to-real gap. To close the visual sim-to-real gap, this framework uses a single depth image as input. To close the dynamics sim-to-real gap, this framework circumvents continuous motor control with a direct mapping from pixel to Cartesian space inferred from the same depth image. Finally, this framework demonstrates inter-robot generality by achieving over 92% real-world grasp success rates in cluttered scenes with novel objects using two multi-fingered robotic hand-arm systems with different degrees of freedom.

Keywords Grasping · Multi-fingered hands · Deep reinforcement learning · Visual attention

1 Introduction

High-DOF grasping includes multi-fingered and non-top-down grasping. This area of robotics research remains active because of its wide application in different domains, ranging from using robot arms for warehouse handling to using humanoid robots for home assistant robotic applications. We can categorize state-of-the-art robotic grasping methodologies largely into *classical grasp planning* approaches that optimize closed grasp quality metrics and *learning-based* methods that learn from examples or experience. The data-driven methods have become more prevalent in recent years as they leverage many recent advancements in the deep learning community.

We observe that the majority of the learning-based methods employ low-DOF robotic hands (e.g., parallel-jaw grippers). These methods also often limit the range of grasp approach direction (e.g., top-down-only grasps), mainly due to real-world sample complexity concerns. In other words, learning high-DOF grasping is difficult because a large amount of real-world data needed to learn this high-dimensional task often makes learning intractable. While circumventing high-DOF grasping with low-DOF hardware or grasp poses reduces the dimensionality of the problem, doing so excludes many solutions that could be used for

This work was supported in part by a Google Research Grant and National Science Foundation Grants CMMI-1734557 and IIS-1527747.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10514-020-09907-y) contains supplementary material, which is available to authorized users.

✉ Bohan Wu
bohan.wu@columbia.edu

Iretiayo Akinola
iakinola@cs.columbia.edu

Abhi Gupta
asg2233@columbia.edu

Feng Xu
fx2155@columbia.edu

Jacob Varley
jakevarley@google.com

David Watkins-Valls
djw2146@columbia.edu

Peter K. Allen
allen@cs.columbia.edu

[1] Robotics Lab, Columbia University, New York, USA

[2] Robotics at Google, New York, USA

Published online: 20 February 2020

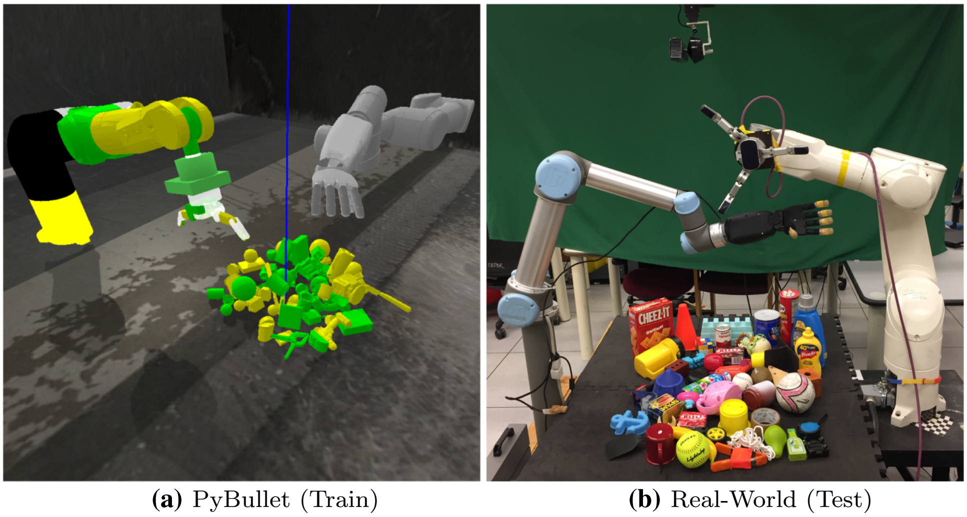

(a) PyBullet (Train) (b) Real-World (Test)

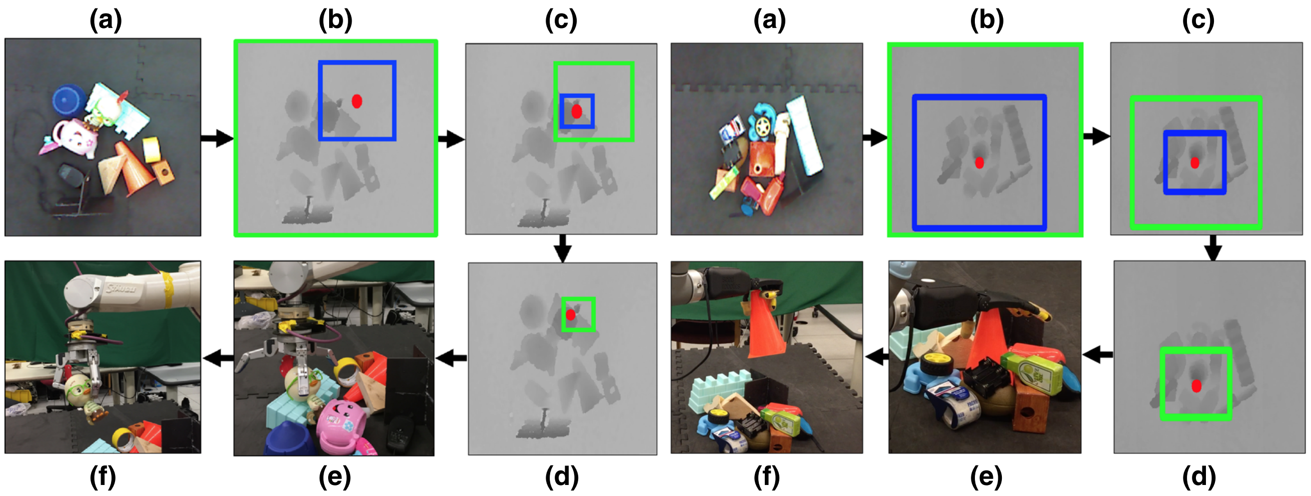
Fig. 1 Hardware and Simulation Setup. **a** Grasping in PyBullet simulation used to train our grasping policy on Staubli–Barrett (left half of image) and UR5-Seed (right half of image). **b** Real-world grasping for testing the trained algorithm on UR5-Seed (left half of image) and Staubli–Barrett (right half of image). The table shows all seen and novel objects used in Staubli–Barrett and UR5-Seed experiments

applications like semantic grasping or grasping for dexterous manipulation. For example, top-down grasping of a bottle could hamper a pouring manipulation task. Besides, since a full 6-DOF grasping system subsumes the more popular 4-DOF methods, the learned system can be left to decide if a 4-DOF system will be sufficient based on the given grasping situation. The choices of the learned algorithm can be analyzed to see which scenarios resulted in reduced-DOF grasps versus other grasp poses. This way, we leave the debate on whether a reduced-DOF grasping system is sufficient entirely to the learned algorithm. Finally, many low-DOF methodologies, though successful in some settings, cannot fully generalize to high-DOF robotic hands or non-top-down grasp poses. These considerations give rise to the need for a robust framework that not only generalizes across robotic hands with arbitrary degrees of freedom but also succeeds at grasping in complex scenarios such as dense clutter of novel objects.

Generative Attention Learning, or GenerAL, provides a general framework to address a fundamental paradox in learning high-DOF grasping. This paradox is between the attempt to increase the robot's DOFs to fully capture the robot's action capacity and the competing objective of keeping the sample complexity (i.e., the amount of training data needed to learn a good grasp) manageable. Trained entirely in simulation, GenerAL transfers directly to real-world grasping without the need for additional training, as shown in Fig. 1. While including more DOFs in grasping robots, such as allowing non-top-down grasping and using multi-fingered hands, can increase their potential to perform better grasps, it also increases the complexity of the problem. The increased complexity affects the stability of many learning algorithms during training, especially for continuous action spaces and their effectiveness and robustness when transferred to real-world settings. Currently, policy gradient methods solve this paradox well *usually in simulation*. By combining advanced policy optimization procedures with neural-network functional approximators, this family of algorithms can solve complex tasks in simulation with high-dimensional observation and action spaces (Schulman et al. 2017). While these methods can capture the potential of the higher action spaces, the on-policy nature of policy gradient methods requires a level of sample complexity that is almost insurmountable in physical environments without large-scale parallelized robotic systems (Levine et al. 2017; Kalashnikov et al. 2018). Also, the brittle nature and complex manifold of robotic grasping where a slight perturbation of a good solution can result in a bad grasp (Rosales et al. 2011) means that optimizing in higher dimensions is more complicated.

GenerAL learns the finger joint configurations for robotic hands with arbitrary degrees of freedom and 6-DOF grasp pose (3D position and 3D orientation) that will return successful grasps. The framework's end-to-end, generative architecture takes as input a single depth image of a cluttered or non-cluttered scene and reinforcement-learns a policy that generates a 6-DOF grasp pose and all finger joint angles (hence the term "generative"). Each component of this grasp is proposed per-pixel and then converted into the appropriate space. For example, the grasp position output by the grasping policy is in the depth image's pixel space and later converted to Cartesian space using the pixel-to-Cartesian mapping (i.e., point cloud) inferred from the depth image.

Learning a reinforcement learning (RL) policy operating directly in the image pixel space not only closes the visual sim-to-real gap but also gives rise to a novel attention mechanism. This mechanism learns to zoom-in and focuses on sub-regions of the depth image to grasp better in dense clutter (hence the term "attention"). Shown in Fig. 2, the proposed mechanism optionally crops the depth image sequentially to gradually zoom into a local region of the scene with a



(a) Staubli-Barrett

(b) UR5-Seed

Fig. 2 Generative Attention Learning (GenerAL) for multi-fingered grasping. GenerAL can be adapted to robots with different DOFs, such as the Stabuli–Barrett robot (Left) and the UR5-Seed robot (Right). Given a scene of cluttered objects (**a**), our method takes in a single depth image and gradually zooms into a local region of the image to generate a good grasp. **b**, **c** and **d** show the zooming process, in which the green bounding box represents the portion of the depth image the robot observes in the *current* timestep, and the blue bounding box represents the portion of the depth image the robot wants to observe in the *next* timestep. In the end, a full-DOF grasp is learned based on the final zoomed image (**d**) as shown in (**e**) and with the final pick-up shown in (**f**)

higher chance of generating good grasps. This mechanism also learns to stop cropping the image after enough zooming. It then generates full-DOF grasps with variable end-effector positions, non-top-down orientations, and all finger joint angles of the hand. In summary, our contributions are:

1. A general framework that can solve grasping in clutter effectively across different multi-fingered or parallel-jaw hand-arm robots;
2. A novel attention feature that enables the robot to reason about cluttered scenes and focus on favorable grasp candidates using a zoom-in mechanism;
3. A generative algorithm for full-DOF grasps that configures 6-DOF end-effector pose and all finger joint angles;
4. Simulation-based learning that uses depth and geometry alone (i.e., no texture) to allow accurate domain transfer to real scenes;
5. Multiple experiments across different robotic hands with different DOFs as well as different grasping scenarios—simulation versus real-world, single-object versus clutter, seen versus novel objects, and top-down versus non-top-down camera view—that produce consistently high levels of grasping success (90% +).

Compared to our prior work Wu et al. (2019a) (available on arXiv), this paper presents the following major improvements:

1. Extension of the pixel-attentive multi-fingered grasping algorithm in Wu et al. (2019a) (available on arXiv) to a fully general framework that applies to robotic hands with arbitrary degrees of freedom;
2. Introduction of a fully general mathematical formulation for the problem of high-DOF grasping under the new framework;
3. Thorough experimentation of the new framework on an entirely different robot (UR5-Seed in Sect. 4.2) to further demonstrate the framework's generality across multiple dimensions: robot hands with various DOFs, single-object vs. clutter, seen versus novel objects, simulation versus real environment, and top-down versus non-top-down camera view;
4. Four new sets of ablation experiments on UR5-Seed that provide a few new insights into the key factors of success in high-DOF grasping;
5. A new finger-closing algorithm for the anthropomorphic Seed hand to overcome its internal overloading and overheating errors;
6. Detailed discussions of the hardware modification applied to the UR5-Seed robot to overcome its mechanical limitations;
7. Elaboration on how our framework can be directly combined with an instance or segmentation algorithm to achieve object-specific grasping;
8. Extended discussions in each section of the paper to provide deeper insights into our framework.

2 Related work

2.1 Generalizable learning-based robotic grasping

Advancements in deep learning have given rise to the rapid development of learning-based techniques for robotic grasping. While many recent works have successfully applied learning-based approaches to robotic grasping in specific settings, few works have demonstrated the generalizability of their methods to different hardware and grasping scenarios. Many non-learning geometric approaches (Miller et al. 2003; Berenson et al. 2007; Akinola et al. 2018) have been developed for and tested on multiple robot hands. However, to the best of our knowledge, we are unaware of learning-based methods that demonstrate similar across-hardware generalizability and work for hands with different DOFs. Learning based grasping research works typically pick a specific robot arm and hand, focus their algorithms on either a parallel-jaw gripper (Mahler et al. 2017; Kalashnikov et al. 2018; Zhao et al. 2019) or a dexterous hand (Varley et al. 2015; Schmidt et al. 2018); deal with single-object, multi-object (Wang et al. 2019) or highly cluttered scenes (Fischinger et al. 2013; Zeng et al. 2018a); and generate 4-DOF such as top-down grasps (Morrison et al. 2018) or full 6-DOF grasps (Schmidt et al. 2018). Existing grasping methods can handle different combinations and subsets of these cases, but a generalized learning-based formulation that works across all these scenarios is needed. Our work fills this void by presenting such a generalized learning-based approach to robotic grasping along multiple dimensions. We show that our method:

1. Works across different robot hardware with varying degree-of-freedom hands;
2. Can handle full 6-DOF grasp poses;
3. Deals with seen and novel objects;
4. Obtains high grasp success rates in both single-object and cluttered grasping scenes.

2.2 Learning grasping under sample complexity challenges

Bohg et al. (2014) gives a review of learning-based grasping methods. Some of these techniques use a supervised learning approach where a model uses grasping examples for training (Lenz et al. 2015; Morrison et al. 2018). On the other hand, there are also RL-based techniques that train a grasping policy based on trial and error-learning to perform actions that would result in grasp success. A significant issue common to both supervised and RL methods is the challenge of sample complexity. A majority of the data-hungry techniques requires a large quantity of data-ranging from thousands to millions. However, real-world robotics data are costly to collect, with labeled data even more expensive. An increase in sample complexity can result from the curse of input or action dimensionality, dealing in continuous spaces instead of discrete spaces, increase in neural network capacity, and change in learning paradigm (RL vs. supervised). For example, learning a 6-DOF grasp pose for a multi-fingered hand will likely require much more data and is more susceptible to learning stability issues than learning a 4-DOF grasp pose for a parallel-jaw hand.

Recent works attempted a variety of algorithms and procedures to tackle the challenges associated with sample complexity in grasping. The first branch of attempts avoids on-policy RL methods. It uses alternative algorithms with lower sample complexity, such as supervised convolutional neural networks (CNNs) (Varley et al. 2015; Morrison et al. 2018), value-function based deep RL such as Deep Q-learning (Kalashnikov et al. 2018; Quillen et al. 2018; Zeng et al. 2018a), RL with a mixture of on-policy and off-policy data (Kalashnikov et al. 2018), and imitation learning (Hsiao and Lozano-Perez 2006). The second branch of attempts uses various procedures to limit the search space of the learning algorithm. For example, one can leverage the image-to-coordinate mapping based on the point cloud computed from the camera's depth image. This way, the algorithm can only learn to choose a point in the point cloud from the image as opposed to the desired 3D position in the robot's coordinate system (Varley et al. 2015; Morrison et al. 2018), a philosophy that inspired our approach. Alternatively, one can restrict the robot's DOFs to top-down grasps only (Morrison et al. 2018). The third branch learns to grasp in simulation and proposes sample-efficient sim-to-real frameworks such as domain adaptation (Bousmalis et al. 2018), domain randomization (Tobin et al. 2017), and randomized-to-canonical adaptation (James et al. 2018) to transfer to the real world.

2.3 Vision-based grasping

Real-world grasping requires visual information about both the environment and the graspable objects to generate stable grasps. This can be RGB (Levine et al. 2017; Kalashnikov et al. 2018), RGB-D (Varley et al. 2015; Lu et al. 2017; Zeng et al. 2018a, b; Chen et al. 2019), or depth-only data (Varley et al. 2017; Watkins-Valls et al. 2019; Lundell et al. 2019). In this work, we only use a single depth image as input and avoid RGB information due to two considerations. First, texture information could be less useful than local geometry in the problem of grasping. Second, RGB images are arguably harder to simulate with high fidelity than depth maps, and using them increases the visual sim-to-real gap. The debate on how best to bridge the domain gap between simulation and the real world remains active, and there is no consensus on which approach works best.

Attention mechanism This refers to a class of neural network techniques that aims to determine which regions of an input image are essential to the task at hand. By applying convolutional or recurrent layers to the input, these techniques generate a saliency map. This map has a per-pixel score proportional to the importance of each location in the image. While previous works have applied this mechanism to saliency prediction (Pan et al. 2016) and image classification (Mnih et al. 2014), we use attention to improve grasping in cluttered scenes. A previous work (Wang et al. 2018) used the attention mechanism to determine a cropping window for an input image that maximally preserves image content and aesthetic value. We apply this idea to predict which region of a cluttered scene to zoom-in on towards achieving improved robotic grasping success. We set up this attention-zooming mechanism in a fully reinforcement-learned manner. Similar to our work, Gualtieri and Platt (2018) used the idea of visual attention in generating 6-DOF grasp poses for parallel-jaw grippers. In contrast, GenerAL is not limited to parallel-jaw grippers and can work with hands with a different number of fingers. Besides, our attention mechanism is designed to handle scenes with dense clutter.

2.4 Multi-fingered grasping

Multi-fingered grasping, commonly known as *dexterous grasping* in the literature, has been tackled using classical grasping methods (Berenson and Srinivasa 2008). These methods use knowledge of the object geometry and contact force analysis to generate finger placement locations in a way that encloses the target object and is robust to disturbances, i.e., the object stays in hand under environmental perturbations (Okamura et al. 2000). Some of these methods sequentially solve for each finger location where the placement of a new finger depends on the placement of the previously placed ones (Hang et al. 2014). On the other hand, some methods reduce the dimensionality of the finger joints into a smaller set of grasp primitives to perform the grasp search and optimization in a smaller subspace (Saut et al. 2007; Ciocarlie et al. 2007; Berenson and Srinivasa 2008; Ciocarlie and Allen 2009).

On the other hand, deep learning for dexterous grasping is an open area of research. Varley et al. (2015) developed a supervised learning method that proposes heatmaps for finger placement locations in pixel space, which guide a subsequent grasp planning stage. It is, therefore, a hybrid based method. More recently, Schmidt et al. (2018) proposed a fully learned approach that predicts a 6D grasp pose from a depth image. This method uses supervised learning and requires a training dataset of good grasps. In contrast, GenerAL takes an RL approach that learns to grasp more successfully via trial and error and does not require any grasp datasets.

In summary, our method takes in a depth image and uses a policy gradient method to predict full 6-DOF grasp pose and all finger joint angles. Tobin et al. (2018) presented an autoregressive approach that can be extended to learn a full-DOF grasp pose. However, they only show top-down grasp results for parallel-jaw hands. Another closely related work (Viereck et al. 2017) based on simulated depth images proposed a supervised learning method that predicts grasps based on an input depth image obtained from a hand-mounted camera. Their method greedily moves the gripper towards the predicted grasp pose while capturing new depth images continuously.

In contrast to their method, GenerAL does not require moving the robot arm to take a closer shot (depth image) of the scene. Instead, GenerAL captures the depth image only once and uses a learned attention mechanism to shift focus and zoom into the image to a level that will maximize grasp success. Also, while Viereck et al. (2017) produces top-down grasps for parallel-jaw grippers, our method is not constrained in this manner. To the best of our knowledge, GenerAL is the first high-DOF grasping framework for 6-DOF grasp pose and all finger joint angles that demonstrated to successfully generalize across hands with various degrees of freedom, clutter scenes, and novel objects.

3 Formulation of high-DOF grasping

3.1 RL and POMDP formulation

In our formulation of high-DOF grasping, a grasping robotic *agent* interacts with a clutter *environment* to maximize the expected reward (Sutton and Barto 1998). The environment is a Partially Observable Markov Decision Process (POMDP), since the agent cannot observe (1) RGB information or (2) the complete 3D geometry of any object or the entire scene. To foster good generalization and transfer of our framework, we model this environment as an MDP defined by $\langle \mathcal{S}, \mathcal{A}, \rho_0, \mathcal{R}, \mathcal{T}, \gamma \rangle$ with an observation space $\mathcal{S}$, an action space $\mathcal{A}$, an initial state distribution $\rho_0 \in \Pi(\mathcal{S})$, a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, a dynamics model $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$, a discount factor $\gamma \in [0, 1)$, and an infinite horizon. $\Pi(\cdot)$ defines a probability distribution over a set. The agent acts according to stationary stochastic policies $\pi : \mathcal{S} \rightarrow \Pi(\mathcal{A})$, which specify action choice probabilities for each observation. Each policy π has a corresponding $Q_\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ function that defines the expected discounted cumulative reward for taking action a from observation s and following the policy π from that point onward.

3.2 Formulation of end-effector pose

A grasping robot, parallel-jaw or multi-fingered, generally has an end-effector (EE) whose pose $\mathrm{EE}_{\mathrm{pose}}$ has a position $\mathrm{EE}_{\mathrm{pos}}$ and orientation $\mathrm{EE}_{\mathrm{ori}}$:

$$\mathrm{EE}_{\mathrm{pos}} = \{\mathrm{EE}_x, \mathrm{EE}_y, \mathrm{EE}_z\} \in \mathbb{R}^3 \tag{1}$$

$$\mathrm{EE}_{\mathrm{ori}} = \{\mathrm{EE}_{\mathrm{roll}}, \mathrm{EE}_{\mathrm{pitch}}, \mathrm{EE}_{\mathrm{yaw}}\} \in \mathbb{R}^3 \tag{2}$$

$$\mathrm{EE}_{\mathrm{pose}} = \mathrm{EE}_{\mathrm{pos}} \cup \mathrm{EE}_{\mathrm{ori}} \in \mathbb{R}^6 \tag{3}$$

Here, the end-effector orientation $\mathrm{EE}_{\mathrm{ori}}$ is represented as the rotational transform $\{\mathrm{EE}_{\mathrm{roll}}, \mathrm{EE}_{\mathrm{pitch}}, \mathrm{EE}_{\mathrm{yaw}}\}$ from the unit-x vector $[1, 0, 0]$. Even though $\mathrm{EE}_{\mathrm{ori}}$ can be defined instead using the $\{x, y, z, w\}$ quaternion, we find our current formulation more convenient for readers to understand our framework later on.

3.3 Formulation of robot DOFs

Our framework circumvents the high dynamics sim-to-real gap of learning continuous motor control by first disentangling the robot's DOFs Ψ_{robot} into two subsets of DOFs: arm-DOFs Ψ_{arm} and finger-DOFs Ψ_{finger}:

$$\Psi_{\mathrm{robot}} = \Psi_{\mathrm{arm}} \cup \Psi_{\mathrm{finger}} \tag{4}$$

The framework then learns arm-DOFs Ψ_{arm} *implicitly* by learning the end-effector pose $\mathrm{EE}_{\mathrm{pose}}$ and learns finger-DOFs Ψ_{finger} *explicitly* in its action space $\mathcal{A}$.

The arm-DOFs Ψ_{arm} determine the position and orientation of the end-effector pose via forward kinematics h_{FK} and vice versa via inverse kinematics solver h_{IK}:

$$h_{\mathrm{FK}} : \{\phi(\psi) \mid \psi \in \Psi_{\mathrm{arm}}\} \rightarrow \mathrm{EE}_{\mathrm{pose}} \tag{5}$$

$$h_{\mathrm{IK}} : \mathrm{EE}_{\mathrm{pose}} \rightarrow \{\phi(\psi) \mid \psi \in \Psi_{\mathrm{arm}}\} \tag{6}$$

where ϕ refers to the current joint angle of a certain degree of freedom of the robot:

$$\phi : \Psi \rightarrow \mathbb{R} \tag{7}$$

The finger-DOFs Ψ_{finger} decide how the fingers close and have no control over the end-effector pose. Hereafter, we use N to refer to the number of finger-DOFs the robot has:

$$N = |\Psi_{\mathrm{finger}}| \tag{8}$$

Depending on the combination of arm and hand robots, the appropriate classification can be subtle. For example, the arm-DOFs of the UR5-Seed robot in Sect. 4.2 include not just the six DOFs of the UR5 arm, but also the three wrist-related DOFs from the Seed hand.

4 Hardware and simulation setup

To demonstrate our framework's robustness across different high-DOF robots, we experimented with two multi-fingered robotic hands with different number of DOFs. They include a BH-280 Barrett Hand mounted on a Staubli-TX60 Arm ("Staubli–Barrett" hereafter) and an anthropomorphic RH8D Seed Hand[1] mounted on a UR5 Arm ("UR5-Seed" hereafter). We use these two robots in both real-world and PyBullet (Coumans and Bai 2016) simulation, as shown in Fig. 1. In the real-world (Fig. 1b), a top-down Kinect Depth Camera is mounted statically on top of the grasping scene. Below we formally disentangle each robotic hand's DOFs, while Appendix A.1 elaborates on the mechanical structures of the BH-280 Barrett hand and the anthropomorphic RH8D Seed hand.

4.1 Staubli–Barrett

The Staubli–Barrett (SB) robot has four finger-DOFs and six arm-DOFs:

$$\begin{aligned}
\Psi^{\mathrm{SB}}_{\mathrm{arm}} &= \{\text{Staubli-joint-1}, \ldots, \text{Staubli-joint-6}\}\\
\Psi^{\mathrm{SB}}_{\mathrm{finger}} &= \{\text{lateral-spread}, \text{finger-1}, \ldots, \text{finger-3}\}\\
N^{\mathrm{SB}} &= |\Psi^{\mathrm{SB}}_{\mathrm{finger}}| = 4
\end{aligned}$$

4.2 UR5-Seed

The UR5-Seed (US) robot has five finger-DOFs and nine arm-DOFs:

$$\begin{aligned}
\Psi^{US}_{\mathrm{arm}} &= \{\text{UR5-joint-1}, \ldots, \text{UR5-joint-6},\\
&\quad \text{wrist-rotation}, \text{wrist-flexion}, \text{wrist-adduction}\}\\
\Psi^{US}_{\mathrm{finger}} &= \{\text{thumb-adduction}, \text{thumb-flexion},\\
&\quad \text{index-flexion}, \text{middle-flexion}, \text{ring-flexion}\}\\
N^{US} &= |\Psi^{US}_{\mathrm{finger}}| = 5
\end{aligned}$$

5 The Generative Attention Learning framework for high-DOF grasping

Our framework models the task of high-DOF grasping as an infinite-horizon MDP. During each episode, the robot makes a single grasp attempt on the scene. During each timestep t of the episode, the robot either (1) zooms into a local region of the scene via a reinforcement-learned attention mechanism or (2) terminates the episode and generates a grasp based on the current zoom level.

To begin, during the first timestep $t = 1$, a single depth image of the grasping scene is first captured by a depth camera and resized to 224 × 224: $s^{\mathrm{depth}}_t \in \mathbb{R}^{224\times224}$. This depth

[1] For structural details of the RH8D Seed hand, kindly see http://www.seedrobotics.com/rh8d-dexterous-hand.html.

image, along with a scalar ratio indicating the current image's zoom level $s_t^{\text{scale}} \in \mathbb{R}$, serves as the robot's observation: $s_t = \{s_t^{\text{depth}}, s_t^{\text{scale}}\} \in \mathbb{R}^{224\times224+1} \in \mathcal{S}$. The scalar ratio s_t^{scale} gives the robot an ability to gauge the actual size of the objects during zooming or grasping and is initially 1 since no zooming was previously performed. Next, both the depth image and the current zoom level are fed into a four-branch CNN f, which has a shared encoder-decoder backbone g_{backbone} and four branches $\{b^{\text{position}}, b^{\text{attention}}, b^{\text{rpy}}, b^{\text{fingers}}\}$:

$$\begin{aligned} f^x(s_t) &= b^x(g_{\text{backbone}}(s_t^{\text{depth}}), s_t^{\text{scale}}), \\ &\quad \forall x \in \{\text{position, attention, rpy, fingers}\} \end{aligned} \tag{9}$$

This four-branch CNN outputs a total of $(6+N)$ two-dimensional maps, which encode $(6+N)$ independent probability distributions from which each component of the action a_t is sampled:

$$\begin{aligned} a_t &= \{a_t^{\text{position}}, a_t^{\text{zoom}}, a_t^{\text{scale}}, a_t^{\text{roll}}, a_t^{\text{pitch}}, a_t^{\text{yaw}}\} \\ &\quad \cup \{a_t^{\psi} \mid \psi \in \Psi_{\text{finger}}\} \\ \text{where } & f^{\text{position}}(s_t) \rightarrow a_t^{\text{position}}, \\ & f^{\text{attention}}(s_t) \rightarrow \{a_t^{\text{zoom}}, a_t^{\text{scale}}\}, \\ & f^{\text{rpy}}(s_t) \rightarrow \{a_t^{\text{roll}}, a_t^{\text{pitch}}, a_t^{\text{yaw}}\}, \\ \text{and } & f^{\text{fingers}}(s_t) \rightarrow \{a_t^{\psi} \mid \psi \in \Psi_{\text{finger}}\} \end{aligned} \tag{10}$$

Given this action a_t, the robot either zooms into a local region of the depth map (blue bounding box in Fig. 3) or directly performs a fully-defined grasp. The "attention" actions $\{a_t^{\text{zoom}}, a_t^{\text{scale}}\}$ allow the robot to pay attention to a local region of the scene to make better grasps. Therefore, we term this local region the robot's "region of attention". Among the $(6+N)$ action scalars:

1. a_t^{position} represents the robot's end-effector position EE_{pos} during grasping and the center location of the robot's region of attention during zooming;
2. a_t^{zoom} and a_t^{scale} represent the zoom vs. grasp decision flag and the scale of the zooming respectively;
3. a_t^{roll}, a_t^{pitch} and a_t^{yaw} represent the roll, pitch, yaw of the end-effector EE_{ori} respectively during grasping;
4. $\{a_t^{\psi} \mid \psi \in \Psi_{\text{finger}}\}$ represents each of the finger-DOFs in Ψ_{finger} for the high-DOF hand prior to finger-closing.

Below we discuss each of them in detail.

5.1 The position map

The Position Map f^{position} encodes the robot's end-effector position EE_{pos} during grasping and the center location of the robot's region of attention (red dot in Fig. 3) during zooming. Instead of encoding this position in Cartesian coordinates $\{x, y, z\}$, which will result in a very large and continuous action space to learn from, we observe that effective grasp positions can be associated with a point in the scene's point cloud, which is a discrete and smaller action space. Therefore, we encode this position using a single-channel 2D map of logits for a spatial-softmax distribution (Levine et al. 2016): $f^{\text{position}} : S \rightarrow \mathbb{R}^{224\times224}$, from which a pixel location a_t^{position} can be sampled:

$$\begin{aligned} a_t^{\text{position}} &\sim \pi(\cdot \mid s_t) \\ &= \text{spatial-softmax}(\text{logits} = f^{\text{position}}(s_t)) \\ &\in [1, 224 \times 224] \end{aligned} \tag{11}$$

Here, GenerAL used the standard spatial-softmax operation with the default temperature scaling of 1. While different temperature scaling could have made the learned distribution more or less extreme, the default scaling of 1 worked well empirically, and the model was able to learn the right scale for the logits.

Given this pixel location a_t^{position}:

1. if the robot decides to *zoom*, a bounding box centered around a_t^{position} with a scale determined by a_t^{scale} is cropped from the original depth map and resized back to 224×224. The resulting image becomes s_{t+1}^{depth}: the input depth map for the next timestep $t+1$;
2. if the robot decides to *grasp*, a_t^{position} represents a unique point in the point cloud; the depth value at this pixel location a_t^{position} is converted to an $\{x, y, z\}$ Cartesian location that the end-effector will be located before closing its fingers and trying to grasp.

Because this pixel location enables the robot to zoom into the robot's region of attention and place the end-effector on a local point, we term this pixel location a_t^{position} the robot's "point of attention". Whether the robot decides to zoom in or grasp depends on the output of the Attention Maps, which we discuss in the next section.

5.2 The attention maps

The Attention Maps $f^{\text{attention}}$ make two decisions. First, they decide whether the robot should (1) zoom further into the depth map or (2) stop zooming further and start grasping. Second, they determine the level of zooming the robot should perform to acquire a better grasp down the road if the first decision is to zoom rather than grasp. These two decisions are important for grasping in dense clutter because while zooming into a cluttered scene can enable the robot to pay

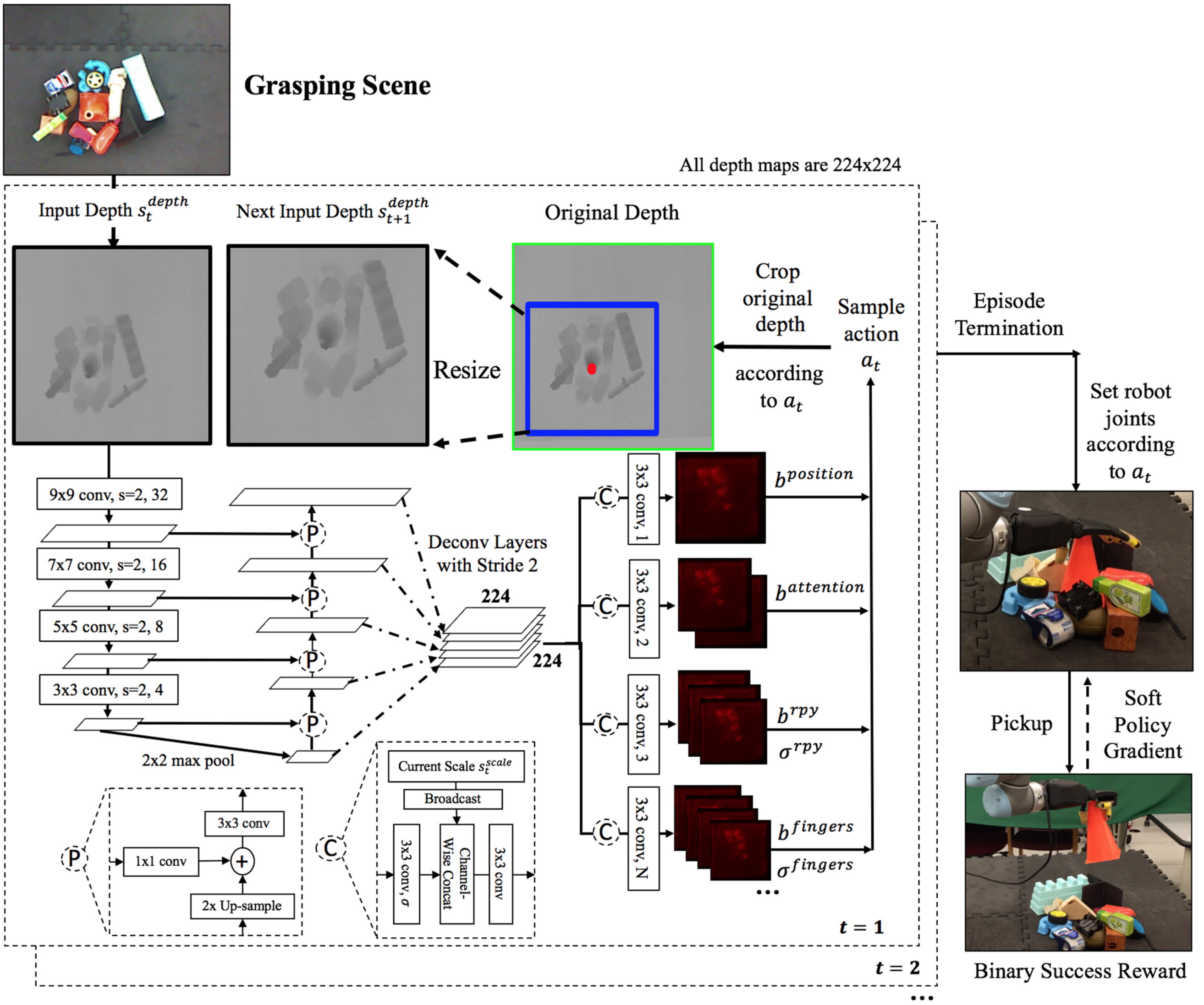


Fig. 3 Generative Attention Learning of multi-fingered grasping architecture. The 224 × 224 input depth map of a grasping scene s_t^{depth} is accepted as input (top left) into a feature-pyramid four-branch CNN that outputs $(6 + N)$ activation maps. The "P" blocks indicate feature pyramid blocks, giving scale invariance capability to the CNN. The "C" blocks indicate how GenerAL introduces the current zoom level s_t^{scale} into each branch. All convolutional layers have ReLU activations and strides of 1 unless otherwise specified. For example, "9 × 9 conv, $s = 2$, 32" refers to 9 × 9 kernel, stride of 2, 32 channels and ReLU activation. The number of deconvolutional layers ranges from 1 to 5 to upsample various intermediate feature maps back to 224 × 224. Each red-black map proposes pixel-wise grasp configurations and their probabilities. An action encoding the end-effector position, orientation, and all finger-DOFs is sampled from these maps, as well as a learned binary flag to decide whether to zoom further into the depth map at a certain scale or stop zooming and start grasping. If the decision is to zoom, the original depth map is cropped (blue bounding box on "Original Depth" image) according to the sampled action, resized and fed back into the CNN for the next timestep $t + 1$, forming "attention". As the episode terminates, the robot acquires a binary success reward from the environment, and the policy gradient gets back-propagated into the entire network. As mentioned previously, N refers to the number of finger-DOFs the robotic hand has. For the Staubli–Barrett, $N = 4$. For the UR5-Seed, $N = 5$

attention to a less visually-cluttered environment, too much zooming can cause the robot to lose sight of nearby objects.

Moreover, these two decisions should be different for different points of attention a_t^{position}. For example, if the current point of attention corresponds to a 3D point located on top of an object, then grasping could be a better decision than zooming. On the contrary, if the current point of attention corresponds to a 3D point located on the table where the objects reside, then zooming could be a better decision than grasping. Similar reasoning applies to the zoom level. Therefore, instead of encoding these two decisions as two one-size-fits-all scalars, we use a two-channel map where each pixel on the map represents how much the robot intends to zoom versus grasp and the zoom scale for every possible point of attention:

$f^{\text{attention}} : S \rightarrow \mathbb{R}^{224\times 224\times 2}$. The first value on each pixel is the p parameter for a Bernoulli distribution, and the robot makes the zoom versus grasp decision a_t^{zoom} by sampling a binary digit from this distribution:

$$\begin{aligned} a_t^{\text{zoom}} &\sim \pi(\cdot \mid s_t, a_t^{\text{position}}) \\ &= \text{Bern}(\text{sigmoid}(f^{\text{attention}}(s_t)_{(a_t^{\text{position}},1)})) \\ &\in \{0, 1\} \end{aligned} \tag{12}$$

If $a_t^{\text{zoom}} = 1$, the robot zooms further into the depth map. If $a_t^{\text{zoom}} = 0$, the robot stops zooming and makes the grasp. The second value on each pixel represents the sigmoid-activated mean of a Gaussian distribution from which the robot samples the zoom scale a_t^{scale}. This zoom scale is a scalar that represents the height and width of the desired region of attention as a fraction of the current image size (224 × 224), while the height to width aspect ratio remains the same:

$$\begin{aligned} a_t^{\text{scale}} &\sim \pi(\cdot \mid s_t, a_t^{\text{position}}) \\ &= \mathcal{N}(\text{sigmoid}(f^{\text{attention}}(s_t)_{(a_t^{\text{position}},2)}), \sigma_{\text{scale}}) \end{aligned} \tag{13}$$

5.3 The RPY (Roll-Pitch-Yaw) orientation maps

The RPY Orientation Maps (f^{rpy}) determine the end-effector orientation EE_{ori} of a grasp by specifying the roll, pitch, and yaw rotations from the unit-x vector $[1, 0, 0]$, as elaborated previously. Similar to the case of the Attention Maps, the RPY values ought to be different for different points of attention a_t^{position}. For example, if the current point of attention corresponds to a 3D point located on top of an object, then a good set of RPY values should correspond to a near-top-down grasp. On the contrary, if the current point of attention corresponds to a 3D point located on the front of the object, then a good set of RPY values should correspond to a more forward-facing grasp. Therefore, GenerAL does not represent the three RPY values using three one-size-fits-all scalars $\{\alpha, \beta, \gamma\}$ across all possible points of attention. Instead, GenerAL uses a three-channel map where each pixel on the map determines the three RPY values for every possible point of attention: $f^{\text{rpy}} : S \rightarrow \mathbb{R}^{224\times 224\times 3}$.

To determine each of the three RPY components $\text{rpy}_i \in \{\text{roll}, \text{pitch}, \text{yaw}\}$ for each a_t^{position}, the robot samples from a Gaussian distribution, whose mean μ_{rpy_i} is determined by the per-channel value of the pixel at the corresponding a_t^{position} and whose standard deviation σ_{rpy_i} is determined by a learned scalar parameter across all possible a_t^{position}:

$$\begin{aligned} a_t^{\text{rpy}_i} &\sim \pi(\cdot \mid s_t, a_t^{\text{position}}) \\ &= \mathcal{N}(\mu_{\text{rpy}_i}, \sigma_{\text{rpy}_i}) \times \pi \\ &= \mathcal{N}(\text{activation}_{\text{i}}(f^{\text{rpy}}(s_t)_{(a_t^{\text{position}},i)}), \sigma_{\text{rpy}_i}) \times \pi \end{aligned} \tag{14}$$

Here, modeling orientation using Gaussian distribution is a natural approach. Using an orientation-specific distribution such as Bingham distribution can also work for GenerAL.

For each rpy_i orientation component {roll (γ), pitch (β), yaw (α)}, the activation functions "activation$_{\text{i}}$" are *tanh*, *sigmoid*, and *tanh* respectively. This configuration results in an effective range of $[-\pi, \pi]$, $[0, \pi]$, and $[-\pi, \pi]$. Mathematically, the resulting orientation transformed from the unit-x vector $[1, 0, 0]$ using $\{\alpha, \beta, \gamma\}$ has a unit directional vector $\mathbf{p}$ of:

$$\mathbf{p} = [\cos\alpha \cos\beta, \sin\alpha \cos\beta, -\sin\alpha] \tag{15}$$

Note that the pitch angle range is $[0, \pi]$ as opposed to $[-\pi, \pi]$ because only pitch values within $[0, \pi]$ produce meaningful grasps with the end-effector facing *downwards* as opposed to upwards, i.e. the z component in Eq. 15 will be negative, but not necessarily 90$^{\text{o}}$ top-down.

5.4 The finger joint maps

The N finger joint maps f^{fingers} determine the pre-grasp finger joint angles of the hand *before* closing all fingers. Each of the N maps represents the joint angle for each of the finger-DOFs Ψ_{finger} of the hand: $f^{\text{fingers}} : S \rightarrow \mathbb{R}^{224\times 224\times N}$. Note that this formulation applies to robots with arbitrary DOFs.

To propose angle for each of the finger-DOFs:

$$\psi \in \Psi_{\text{finger}} \tag{16}$$

given a_t^{position}, each joint angle a_t^{ψ} is sampled from a Gaussian distribution and then scaled by the scaling factor scale_{ψ}. The mean of this Gaussian distribution μ_{ψ} is determined by the sigmoid-activated value at pixel location a_{position} of the corresponding finger joint map and the standard deviation σ_{ψ} is a learned scalar parameter across all possible a_{position}:

$$\begin{aligned} a_t^{\psi} &\sim \pi(\cdot \mid s_t, a_t^{\text{position}}) \\ &= \text{scale}_{\psi} \times \mathcal{N}(\mu_{\psi}, \sigma_{\psi}) \\ &= \text{scale}_{\psi} \times \mathcal{N}\left(\text{sigmoid}\left(f^{\text{fingers}}(s_t)_{(a_t^{\text{position}},i)}\right), \sigma_{\psi}\right) \end{aligned} \tag{17}$$

For the Barrett hand used, the scale_{ψ} is $\pi/2$ for the lateral-spread joint and 0.61 for each of the other three finger-DOFs. This configuration gives an effective range of $[0, \pi/2]$ for the lateral-spread and $[0, 0.61]$ for the three finger-joints. We restrict the finger-1, 2, and 3 joint ranges to be a quarter of the maximum range $[0, 2.44]$ because outside of this range, the hand is nearly closed. We restrict the lateral-spread to $[0, \frac{\pi}{2}]$ because outside of this range, no meaningful grasps can be generated (all fingers will be on the same side of the hand).

For the Seed hand, the thumb-adduction joint has scale$_\psi$ of $\pi/2$, while all other four finger-DOFs have scale$_\psi$ of 0.37, which is a quarter of the joint angle maximum of 1.48 for reasons similar to the Barrett hand.

5.5 Soft proximal policy optimization

GenerAL optimizes its generative network using a policy gradient formulation. Under this formulation, GenerAL trains through model-free deep RL, which means it does not explicitly model the environment dynamics. Indeed, discriminative RL such as deep Q-learning can require less training effort. The necessity of modeling the grasping problem as a generative model is to avoid generating potentially suboptimal actions. Concretely, even though discriminative models in deep RL can accurately evaluate the expected cumulative reward of an arbitrary state-action pair, it typically uses the Cross-Entropy Method (CEM) to come up with an action. This action is not necessarily globally optimal when the action space is continuous.

Let θ be the parameter weights of the entire network f and π_θ be the RL policy the robot is trying to learn: $\pi_\theta : \mathcal{S} \rightarrow \Pi(\mathcal{A})$. The robot's goal is to maximize the cumulative discounted sum of rewards:

$$\underset{\theta}{\text{maximize}}\ \mathbb{E}_{\pi_\theta}\left[\sum_t \gamma^{t-1} r_t\right] \tag{18}$$

The reward during the final timestep t_{final} is a binary indicator of whether the robot successfully picked any object up:

$$r_{t_{\text{final}}} = \mathbb{1}\{\text{pick-up is successful}\} \tag{19}$$

To begin, we follow the standard policy optimization objective:

$$\underset{\theta}{\text{maximize}}\ \mathcal{L} = \mathbb{E}_{\pi_\theta}[\pi_\theta(a_t \mid s_t) Q_{\pi_\theta}(s_t, a_t)] \tag{20}$$

Next, we opted out baseline subtraction for variance reduction since it empirically does not improve performance significantly.

Finally, we substitute the action probability $\pi_\theta(a_t \mid s_t)$ with the Clipped Surrogate Objective (Schulman et al. 2017) and apply a soft advantage target to balance between exploration and exploitation (Haarnoja et al. 2018):

$$\begin{aligned}\underset{\theta}{\text{maximize}}\ \mathcal{L}^{PG} &= \mathbb{E}_{\rho_0, \pi_\theta}[\min(\lambda_t(\theta), \\ &\text{clip}(\lambda_t(\theta), 1-\epsilon, 1+\epsilon))(Q_t - \alpha \log \pi_\theta(a_t \mid s_t))]\end{aligned} \tag{21}$$

where

$$\lambda_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \tag{22}$$

Table 1 Hyperparameters: soft proximal policy optimization

Hyperparameter	Value
Base learning rate	1×10^{-4}
Number of epoches per batch	10
Number of actors	14
Batch size	500
Minibatch size	96
Discount rate (γ)	0.99
GAE parameter (λ) (Staubli–Barrett)	0.95
GAE parameter (λ) (UR5-Seed)	0
PPO clipping coefficient (ϵ)	0.2
Value function coefficient (c_1)	0
Gradient clipping	20
Temperature parameter (α) (Staubli–Barrett)	0
Temperature parameter (α) (UR5-Seed)	5×10^{-4}
Optimizer	Adam

During zooming, no grasp is generated and a_t is defined only by $\{a_t^{\text{position}}, a_t^{\text{zoom}}, a_t^{\text{scale}}\}$. During grasping, a_t is defined by every component except a_t^{scale}. Therefore:

$$\begin{aligned}&\log \pi_\theta(a_t \mid s_t) \\ &\quad = \log \pi_\theta(a_t^{\text{position}} \mid s_t) + \log \pi_\theta(a_t^{\text{zoom}} \mid s_t, a_t^{\text{position}}) \\ &\qquad + a_t^{\text{zoom}} \times \log \pi_\theta(a_t^{\text{scale}} \mid s_t, a_t^{\text{position}}) \\ &\qquad + (1 - a_t^{\text{zoom}}) \times \sum_{\psi \in \Psi} \log \pi_\theta(a_t^{\psi} \mid s_t, a_t^{\text{position}})\end{aligned} \tag{23}$$

where $\Psi = \{\text{roll, pitch, yaw}\} \cup \Psi_{\text{finger}}$.

Table 1 details Soft PPO hyperparameters. Here, a Generalized Advantage Estimation (GAE) (Schulman et al. 2015) parameter of 0 indicates that GenerAL computes strictly Monte-Carlo returns, whose advantage estimates will have lower bias and higher variance. Conversely, a GAE parameter close to 1 will have advantage estimates with higher bias and lower variance. Since our reward structure is simply a one-hot grasp success scalar indicator, the variance of the policy gradients is manageable even without a high GAE parameter. Having a GAE parameter of 0 for the UR5-Seed demonstrates this. In GenerAL, both GAE hyperparameters of 0 and 0.95 are good parameters for both robots.

5.6 The CNN architecture

Shown in Fig. 3, the CNN architecture draws inspiration from Feature Pyramid Networks (Lin et al. 2016). During an episode, the input depth map is being "zoomed-in" every timestep until the very last. Therefore, the CNN needs to have strong scale invariance (i.e., robustness against change in the scale of the scene objects), hence the feature pyramid blocks in the network.

5.7 Learning standard deviation network parameters

Both the RPY values and the pre-grasp finger joint angles are sampled from independent Gaussian distributions, each of which contains standard deviation parameters: $\{\sigma_{\text{roll}}, \sigma_{\text{pitch}}, \sigma_{\text{yaw}}\} \cup \{\sigma_{\psi} \mid \psi \in \Psi_{\text{finger}}\}$. Concretely, these σ's are learned through standard gradient back-propagation along with all parameters in the rest of the neural network. The gradient is created by the reward signal, which is a one-hot grasp success indicator.

A single σ scalar parameter is created and learned for each Gaussian-sampled scalar in the action space: $\{a_t^{\text{roll}}, a_t^{\text{pitch}}, a_t^{\text{yaw}}\} \cup \{a_t^{\psi} \mid \psi \in \Psi_{\text{finger}}\}$. As a result, there are $(3 + N)$ number of σ's to learn in the entire network, where N is the number of finger-DOFs the robot has. These σ values have the practical effect of allowing the robot to sample different action components with a different standard deviation. For example, since a_t^{pitch} determines how top-down the grasp will be and it is an essential action component, the robot might restrict σ_{pitch} to be relatively small, so that the pitch value a_t^{pitch} does not fluctuate too much. In other words, each σ value is not a function of the point of attention. To extend each σ value to be point-of-attention-dependent, one can learn a 2D map of σ values for each action component. In this case, there will be a total of $[224 \times 224 \times (3 + N)]$ σ values to learn in the entire network. This extension can have both the benefit of greater network capacity and the caveat of more network parameters to learn.

5.8 Robot-specific grasping

Different robotic hands have different control mechanisms for grasping. Below we elaborate on the robot-specific implementation details for how we close fingers to form a grasp.

5.8.1 Staubli–Barrett

The BH-280 Barrett hand can perform both position and velocity control for all its fingers. During grasping as well as lifting, we use velocity control for all fingers at 0.2 rad per second, which allows the fingers to keep exerting force onto the object after contact.

5.8.2 UR5-Seed

The RH8D Seed hand can only perform position control on its joints. Furthermore, the hand frequently raises overloading and overheating errors when the finger motors attempt to keep closing onto the object after contact. Forming a stable grasp requires the fingers to continue applying force on the object even after contact. As a result, we developed Algorithm 1, which allows fingers to keep closing post-contact via position control while also occasionally pulling back to avoid overloading and overheating errors.

Algorithm 1 Error-free grasping via seed hand

```
Ψ ← {thumb-flexion, index-flexion, middle-flexion, ring-flexion}
for finger-DOF $\psi \in \Psi$ do
    Initialize joint angle maxima $\beta_{\max\psi} \leftarrow 2\pi$
    Initialize previous joint angle $\beta_{\text{old}\psi} \leftarrow 0$
end for
for target joint angle $\beta_{\text{target}} = 0, 0.25\pi, \ldots, 2\pi$ do
    for finger-DOF $\psi \in \Psi$ do
        Read current joint angle $\beta_{\psi}$ from motor
        if $0 < \beta_{\psi} - \beta_{\text{old}\psi} < 0.2$ and $\beta_{\max\psi} > \beta_{\psi}$ then
            The finger was blocked by objects
            $\beta_{\max\psi} \leftarrow \beta_{\psi} - 0.15$
        else
            $\beta_{\max\psi} \leftarrow \beta_{\psi} + 0.5$
        end if
        $\beta_{\text{old}\psi} \leftarrow \beta_{\psi}$
        Move finger joint $\psi$ to min[$\beta_{\text{target}}, \beta_{\max\psi}$] rad
    end for
end for
```

In summary, the full Generative Attention Learning for High-DOF Grasping procedure is outlined in Algorithm 2.

5.8.3 Frame of reference of the end-effector

In Algorithm 2, the parent frame of each robotic hand is the robot arm's world frame, whose origin locates at the base of the robot arm. GenerAL generates a grasp position that is independent of the frame of reference since the 3D position of a point in the point cloud can be in an arbitrary parent frame.

5.9 Rationale

5.9.1 Reinforcement learning versus supervised learning

Reinforcement learning (RL) methods often learn tasks with a relatively large number of sequential actions, while the task of grasping under the GenerAL framework has a fewer number of sequential actions. The primary consideration for using RL as opposed to supervised learning to train GenerAL is that RL does not require a good dataset of grasp action examples that contain an intelligent combination of zooming and grasping. Indeed, if collecting a large dataset of grasp action examples in simulation can be done at low costs of time and labor, supervised learning will also be an option.

Algorithm 2 Generative attention learning of high-DOF grasping

```
Initialize $zoom$ to $True$
Initialize $\theta$ to trained model
$\delta \leftarrow 4cm$ for Staubli–Barrett; $\delta \leftarrow 3cm$ for UR5-Seed
$s^{\text{depth}}, s^{\text{scale}} \leftarrow$ single depth map, 1
while zoom do
    Sample action $a$ given $s^{\text{depth}}, s^{\text{scale}}$
    $zoom \leftarrow a^{\text{zoom}}$
    if $zoom$ then
        Crop original depth image around $a^{\text{position}}$ and at scale ($a^{\text{scale}} \times s^{\text{scale}}$) to acquire new $s^{\text{depth}}$
        New $s^{\text{scale}} \leftarrow s^{\text{scale}} \times a^{\text{scale}}$
    end if
end while
for $\psi \in \Psi_{\text{finger}}$ do
    Move finger-DOF $\psi$ to angle $a^{\psi}$
end for
$\text{EE}_{\text{ori}} \leftarrow \{a^{\text{roll}}, a^{\text{pitch}}, a^{\text{yaw}}\}$
Transform $a^{\text{position}}$ to Cartesian coordinates $\{x, y, z\}$ using point cloud inferred from depth map, and a $\delta$ offset along target end-effector orientation $\text{EE}_{\text{ori}}$
$\text{EE}_{\text{pos}} \leftarrow \{x, y, z\}$
$\text{EE}_{\text{pose}} \leftarrow \text{EE}_{\text{pos}} \cup \text{EE}_{\text{ori}}$
Move end-effector to pose $\text{EE}_{\text{pose}}$ using the arm-DOFs $\Psi_{\text{arm}}$ and inverse kinematics solver $h_{\text{IK}}$
Close robot fingers until maximum effort and lift hand according to Sect. 5.8
```

Nevertheless, it is also less straightforward as to what the optimal amount of zooming the grasp action examples in the dataset should perform before proposing a grasp pose. Through RL, GenerAL receives automatic supervision from the simulation environment without the need to collect such a dataset and learns an appropriate level of zooming that would lead to the highest grasp success rate in relative terms.

5.9.2 Learning attention

The design of our framework enables the robot to focus on a sub-region of the entire cluttered scene to grasp better *locally*, or "attention". Without attention, the robot observes too much *global* visual information in the cluttered scene such that it is difficult to grasp a *local* object well. With attention, the robot can gradually zoom into the scene and focus on fewer and fewer objects as the episode continues. Since the task of generating proper attention that will lead to good grasps *in the future* and the task of producing a good grasp *now* require similar reasoning around the objects' local geometry, *one* single CNN branch f^{position} can learn to perform both tasks.

During training, the CNN receives upstream gradient signals encoding how successful the grasp was. Therefore, the CNN learns to update its weights such that:

1. It outputs a Position Map encoding a good grasp position if the episode terminates *at the current timestep*;
2. It outputs the Attention Maps that zoom appropriately into the depth image to yield a good grasp when the episode ends later.

The attention mechanism can gradually learn to focus on less cluttered environments by just randomly generating actions from the initial policy. This is because concentrating on more cluttered environments would likely lead to pick-up failure while focusing on less cluttered environments would likely lead to pick-up success. Since the reward structure is simply a one-hot indicator of grasp success, the attention mechanism receives unbiased, direct reward signals that encourage focusing on less cluttered environments and discourage focusing on more cluttered environments. Partly because of this direct supervision, there is no specific requirement for the initialization of the parameters in the training stage for it to successfully learn the attention mechanism, although better initialization can lead to faster training in simulation.

5.9.3 Solving the challenge of high real-world learning sample complexity

While one can use our framework to learn to grasp directly in real-world environments, this is inefficient without highly parallelized robot farms due to:

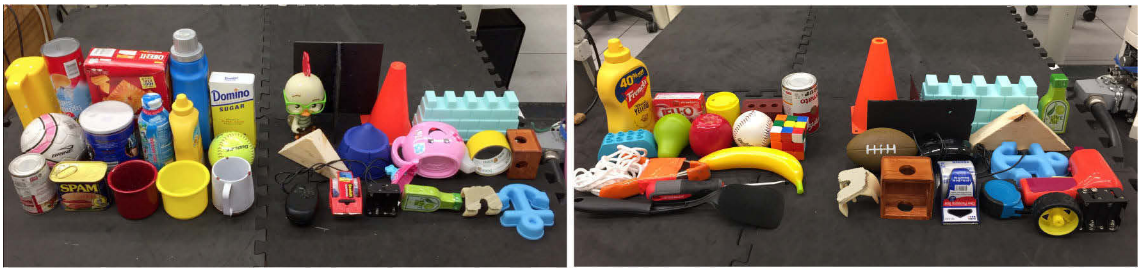
1. The high sample complexity requirement of policy gradient methods;
2. The slow execution of real robots;
3. The difficulty of generating near-i.i.d. cluttered grasping environments in the physical world.

Instead, we opted to learn directly in simulation and transfer to the real environment *without additional learning*. This high sim-to-real fidelity originates from the observation that the main visual sim-to-real gaps for vision-based learning come from texture (RGB) information, rather than depth information.

6 Experiments

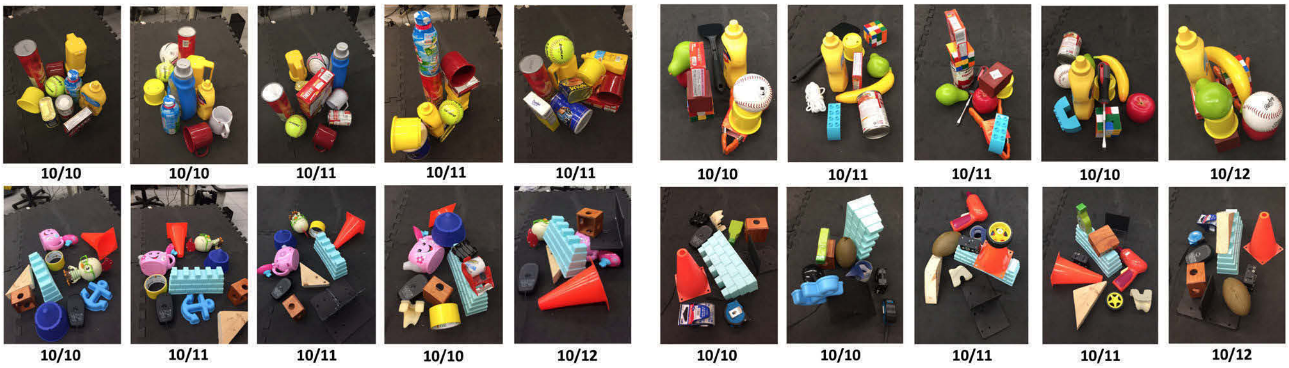
6.1 Training

We train GenerAL entirely in simulation. During training, a single seen object or a cluttered scene of multiple seen objects is loaded with equal probability. We place one object in a single-object scene and a random number of objects from 2 to 30 for a simulated cluttered scene. The number of training grasp attempts required to reach convergence range from 5000 to 15,000, depending on the robot and the hyperparameters used. Here, the amount of training grasp attempts



(a) Staubli-Barrett **(b)** UR5-Seed

Fig. 4 15 seen and 15 novel objects used in the **a** Staubli–Barrett and **b** UR5-Seed experiments. Left half of image: 15 seen objects. Right half of image: 15 Novel objects



(a) Staubli-Barrett **(b)** UR5-Seed

Fig. 5 Results for a few cluttered scenes used for real-world experiments for **a** Staubli–Barrett and **b** UR5-Seed. In each scene, there are 10 objects randomly and densely placed on the table. Sometimes objects in the scene overlap with each other, as shown in the two rightmost scenes in each row for each robot. We report # successful grasps/# grasp attempts. For example, for the top right scene in "Staubli–Barrett", the robot cleared the scene—picking up all 10 overlapping objects with a total of 11 trials. Top row: scenes with seen objects geometrically similar to those used during training. Bottom row: scenes with novel objects geometrically different from those in training

per scene is 1, and the number that ranges from 5000 to 15,000 is for all the scenes. In other words, 5000 to 15,000 random scenes are constructed to train in simulation. The training time is roughly 24 h on a single GPU, 13-virtual CPU machine.

One might expect GenerAL to require some smart neural network initialization to train such a high dimensional grasping policy. In reality, we randomly initialized the neural network and trained the system from scratch. The key reason why GenerAL's neural network can learn from random initialization is due to selecting the point cloud space as the grasp position space, rather than the cartesian space of the world as in Kalashnikov et al. (2018). Under GenerAL's design, the grasping configuration space becomes much lower-dimensional than that of Kalashnikov et al. (2018). It is, therefore, much easier for a random policy to "get lucky" and select a point in the point cloud that leads to a successful grasp. At random initialization (i.e., before training), the initial grasp success rate (i.e., reward) is typically around 1.5%, which is still sparse but learnable. From another viewpoint, GenerAL does not learn a policy to generate every waypoint to the target graspable object as in Kalashnikov et al. (2018). Instead, GenerAL used planning to create the arm trajectory so that the neural network can focus on learning the best final grasp pose, which also leads to less sparse rewards at the time of random initialization.

6.2 Testing

We test GenerAL in both simulation and real-world. Using the ShapeNet Repository (Chang et al. 2015) in simulation, we use 200 + seen objects from the YCB and KIT datasets and 100+ novel objects from the BigBIRD dataset. We evaluate 500 grasp attempts per experiment in simulation. In real-world grasping, we use 15 YCB-like seen objects and 15 novel objects shown in Fig. 4. Note that the real-world objects used in UR5-Seed experiments are, on average, lighter in weight than those in Staubli–Barrett, due to the lower weight load of the Seed hand. We evaluate real-world single-object performance across ten trials per seen or novel object, and real-world cluttered scene performance across 15 cluttered scenes, each with ten cluttered or overlapping objects, as shown in Fig. 5. Video and code of GenerAL can be found at Online Resource 1 (original resolution) or Online Resource

Table 2 Main experiments and ablation results (% grasp success ± SD)

Objects	Single object		Cluttered scene		Single object		Cluttered scene	
	Seen	Novel	Seen	Novel	Seen	Novel	Seen	Novel
Robot	Staubli–Barrett				UR5-Seed			
GenerAL (Sim)	93.8 ± 2.6	94.9 ± 1.4	92.5 ± 1.8	91.1 ± 3.7	94.8 ± 3.9	92.7 ± 2.8	92.5 ± 3.4	91.7 ± 3.0
GenerAL (Real)	96.7 ± 6.2	93.3 ± 8.1	92.9 ± 5.8	91.9 ± 6.7	96.0 ± 7.4	96.0 ± 7.4	94.2 ± 6.9	93.5 ± 6.0
Ablation (simulation)								
No attention	86.9 ± 4.7	85.2 ± 2.7	70.9 ± 6.3	72.2 ± 3.6	83.7 ± 5.3	81.7 ± 6.0	72.7 ± 6.9	69.2 ± 7.0
Top-down	88.6 ± 2.1	87.0 ± 2.7	74.8 ± 2.9	70.8 ± 5.9	82.3 ± 3.5	82.3 ± 6.0	78.7 ± 2.4	77.1 ± 5.3
Low-DOF	50.4 ± 6.7	44.5 ± 5.4	49.0 ± 2.4	45.1 ± 4.4	71.0 ± 4.5	71.9 ± 7.8	66.7 ± 4.4	72.8 ± 6.9
60^o Camera	92.3 ± 2.1	91.9 ± 3.4	91.8 ± 3.6	91.6 ± 2.5	94.6 ± 4.9	92.2 ± 3.4	90.8 ± 4.5	92.3 ± 3.6

2 (low resolution) and https://github.com/CRLab/GenerAL respectively.

In the video, almost all the grasps seem to be top-down for cluttered scenes. Indeed, top-down grasps in clutter can better avoid collisions between the end-effector and the objects, and this is one of the reasons why side grasps appear less frequently in cluttered scenes. However, the seemingly top-down grasps are, in fact, most often tilted by a small degree, a crucial orientation detail that has empirically led to higher grasp success rates. There are two critical ingredients for grasp success in clutter that are related to grasp orientations. The robotic agent needs to generate a grasp orientation that (1) avoids nearby objects and (2) is near-optimal for the target object. Strictly 90% top-down grasps usually cannot satisfy both criteria in cluttered scenes, while a slightly tilted grasp (e.g., 85% or 80% downward) can generally complete both tasks. Even though the cluttered scene grasps in the video seem 90% top-down, they are almost always slightly tilted to adapt to the locally cluttered environments effectively.

6.3 Results and discussion

Table 2 details the experimental results for both the Staubli–Barrett and the UR5-Seed robots. Applying our framework to these robots achieves over 90% grasp success rates consistently across multiple dimensions: seen versus novel objects, simulation versus real-world, single-object versus cluttered scenes, and top-down versus non-top-down camera view. Below we discuss each dimension in detail.

6.3.1 Preliminaries on statistical significance

In Table 2, we presented mean and standard deviation statistics for each experiment, which enabled us to examine whether the difference in performance between different experiments is statistically significant or not. We consider the performance difference statistically significant if the difference in **mean** performance of the two experiments is at least one standard deviation away. For example, the success rates between single-object simulation experiment (93.8 ± 2.6%) and single-novel simulation experiment (94.9±1.4%) for the Staubli–Barrett are statistically similar (as opposed to different) because the difference in mean (1.1%) is smaller than the standard deviation of either statistics (2.6% or 1.4%).

6.3.2 Generalization to novel objects

Shown in Table 2 Row "GenerAL (Sim)", the test success rates in simulation for seen versus novel objects are statistically similar for both single-object scenes (93.8 ± 2.6% vs. 94.9 ± 1.4% for Staubli–Barrett, 94.8 ± 3.9% vs. 92.7 ± 2.8% for UR5-Seed) and cluttered scenes (92.5 ± 1.8% vs. 91.1±3.7% for Staubli–Barrett, 92.5±3.4% vs. 91.7±3.0% for UR5-Seed), exhibiting good transfer to novel objects. In Row "GenerAL (Real)", we notice similarly stable transfer performance to novel objects for both robots in the real world.

The learned generalization to novel objects benefited from the partial observability of GenerAL's MDP formulation, discouraging the network from *overfitting* to seen objects. Since the depth map is the only input modality, visual features are partially observable compared to that of complete 3D geometry, making GenerAL select the safest grasp regardless of what ground-truth geometry might lay underneath the point cloud.

6.3.3 Generalization to real-world scenes

Comparing Row "GenerAL (Sim)" against Row "GenerAL (Real)", we observe high-fidelity real-world transfer given that there was no real-world training. This transferability is mainly due to using depth as the only input modality, which has a smaller visual sim-to-real gap compared to RGB information. We show the real-world performance of individual cluttered scenes of seen and novel objects in Fig. 5. Note that the cluttered scenes include severe overlap and occlusion (two rightmost images of each row for each robot).

6.3.4 Cluttered scene performance

Comparing Column "Single Object" to Column "Cluttered Scene", we observe good cluttered scene performance for both Staubli–Barrett and UR5-Seed. This performance comes mainly from the framework's attention mechanism and domain randomization, i.e., the varying number of objects randomly placed into the scene during training. Under the attention mechanism, the network learns to focus on fewer and fewer objects as the episode continues. This focus eliminates perceptual distraction from objects in the rest of the scene that are far away from the object of interest.

6.4 Ablation

6.4.1 Importance of attention mechanism to performance

We conducted experiments using a finite horizon of 1 instead of an infinite horizon, effectively preventing the robot from using attention to zoom into the scene. Comparing Row "No Attention" to Row "GenerAL (Sim)", we observe larger performance degradation for cluttered scenes (21.6% and 18.9% for seen and novel objects with Staubli–Barrett, and 19.8% and 22.5% with UR5-Seed respectively) than for single-object scenes (6.9% and 9.7% for seen and novel objects with Staubli–Barrett, and 11.1% and 11.0% with UR5-Seed respectively). Qualitatively, the lack of attention resulted in no learning occasionally. We attribute these findings mainly to the ablated network's inability to pay attention to local regions of the cluttered scenes during training.

6.4.2 Using top-down grasp only

By enforcing the value of a_t^{pitch} to $\frac{\pi}{2}$, we effectively restrict the robot to top-down only grasps. Comparing Row "Top-Down" against Row "GenerAL (Sim)" reveals larger performance degradation on cluttered scenes (17.7% and 20.3% for seen and novel objects for Staubli–Barrett, and 13.8% and 14.6% for UR5-Seed respectively) than for single-object scenes (5.2% and 7.9% for seen and novel objects for Staubli–Barrett, and 12.5% and 10.4% for UR5-Seed respectively). This difference is mainly because, in dense clutter, the robot needs non-top-down grasps to generate a better grasp pose for the target object that also avoids nearby objects.

6.4.3 Using low-finger-DOF grasps only

To examine the performance contribution of using multi-fingered as opposed to low-DOF hands, we enforce the lateral-spread $a_t^{\text{lateral\_spread}}$ of the Barrett hand to 0, effectively operating a two-fingered hand. Similarly, for the Seed hand, we enforce the thumb-adduction joint $a_t^{\text{thumb\_adduction}}$ to $\pi/4$ rad, at which point the thumb will close approximately in parallel to the other four fingers on the opposing side, effectively operating as a 4-finger-DOF (instead of 5-finger-DOF) hand. Comparing Row "Low-DOF" to Row "GenerAL (Sim)" reveals performance degradation of 43.4%, 50.4%, 43.5%, 46.0% on single-seen, single-novel, cluttered-seen, cluttered-novel scenes for Staubli–Barrett, and 23.8%, 20.8%, 25.8%, 18.9% for UR5-Seed respectively, indicating a relatively significant contribution from using high-DOF hands. Qualitatively, we observe frequent failures of both hands to grasp cylindrical or spherical objects in low-finger-DOF mode.

6.4.4 Generalization to non-top-down camera viewing angle

We also trained our algorithm with a non-top-down camera viewing angle (60^{o} downward). The simulation results in Row "60^{o} Camera" are statistically similar to Row "GenerAL (Sim)" (90^{o} top-down camera view) for both Staubli–Barrett and UR5-Seed, showing the framework's robustness to non-top-down camera view setup.

6.5 Generalization across different multi-fingered robots

Comparing all Staubli–Barrett experiments with UR5-Seed counterparts, we concluded three main findings.

First, the main, non-ablation results are mostly statistically similar, with both robots achieving over 90% success rates in both single-object and cluttered scenes, seen and novel objects, as well as simulation and real-world.

Secondly, the "Low-DOF" ablation experiments displayed much higher performance degradation for the Staubli–Barrett (43.4%, 50.4%, 43.5%, 46.0% for single-seen, single-novel, cluttered-seen, cluttered-novel respectively) than for UR5-Seed (23.8%, 20.8%, 25.8%, 18.9% respectively). Intuitively, for the three-fingered Staubli–Barrett, "Low-DOF" means disabling the lateral-spread between finger-1 and finger-2 and making the hand a two-fingered gripper in essence. In contrast, disabling the UR5-Seed thumb-adduction joint has a smaller negative effect because:

1. Four fingers are supporting the graspable object on the opposite side of the thumb, covering a wide space such that even if the thumb-adduction is fixed and suboptimal in joint angle, the grasp can still be stable;
2. Setting the thumb-adduction to a constant value still enables the robot to cover additional cartesian space for touching the object. In the Staubli–Barrett case, no space remains between the first two fingers, which increases the robot's likelihood of missing contact with the object.

Lastly, the performance degradation of the Staubli–Barrett robot in top-down only mode is better than the UR5-Seed in single-object scenes (5.2% and 7.9% for single-seen and single-novel with Staubli–Barrett vs. 12.5% and 10.4% respectively with UR5-Seed) but worse in cluttered scenes (17.7% and 20.3% for cluttered-seen and cluttered-novel with Staubli–Barrett vs. 13.8% and 14.6% respectively with UR5-Seed). We provide the following explanation for this phenomenon:

1. The Seed hand is mechanically smaller in reachable space that can be covered by the fingers compared to the Barrett hand, which means suboptimality in the grasp orientation has a larger effect on grasp quality for the UR5-Seed (smaller reachable space requires the hand to be more precise in position and orientation for a good grasp);
2. A smaller hand size has enabled to the Seed hand to avoid nearby objects since a smaller hand collides with nearby objects less often. Besides, a more interactive grasping control in Algorithm 1 can, in practice, squeeze-away nearby objects in the clutter that are colliding with the hand. For the larger Barrett hand's case, a suboptimal grasp orientation empirically creates undesirable collisions between the fingers and non-target objects nearby in the clutter such that performance degrades more than the UR5-Seed.

6.6 Zooming frequency

Qualitatively, we observed that for a well-trained policy, the number of zooming actions per grasp attempt in cluttered scenes can range from 0 to 6 or above and is, on average, around 4. In comparison, the policy zooms around once or twice on average for single-object scenes. This number will depend on the size of the objects and the clutter level of the grasping scene. The smaller the object appears in the depth image or, the more cluttered the grasping scene, the more often the policy zooms in to acquire a better understanding of the local scene and object geometry.

6.7 Training progress

At random initialization of the policy, the reward is around 1.5%. The reward will quickly approach to mid-to-high 80% success rates in the early stage of training. At the later stage of training, the reward will slowly converge to over 90% success rates shown in Table 2.

6.8 Comparison to planning methods

Classical grasp planning approaches can indeed excel at single-seen object scenes. However, they might not be robust to cluttered scenes or novel objects (whose meshes are unknown) due to a lack of generalizability. The performance of planning methods if the optimization over the closed grasp metrics is used will be not bad for single-seen objects but will be less robust than GenerAL for cluttered scenes or novel objects. GenerAL does not need explicit models of individual objects but directly optimizes for high pick-up success rate given a depth image of the scene.

6.9 Solution for framework limitations

6.9.1 Highly dense clutter

Most of the failure cases in clutter originate from objects being tightly cluttered with no spacing for the robot to insert its fingers. GenerAL ends up attempting to pick-up more than one object, which results in the objects sliding out during lift. Such cases are challenging to tackle unless the scene is perturbed. Since GenerAL runs iteratively, the failure during the attempted lift produces sufficient perturbation to the scene such that GenerAL can generate a successful grasp on the next try. Furthermore, our follow-on work (Wu et al. 2019b) (available on arXiv) can also overcome such challenges with the tactile-enabled BH-282 Barrett hand through its reinforcement-learned incremental finger closing procedure based on tactile sensory feedback.

6.9.2 Object-specific grasping

GenerAL learns grasp poses and pre-grasp finger joint angles that maximize the likelihood of pick-up success. This objective means that GenerAL has its learned preferences for which object it should grasp next in a cluttered scene. Nevertheless, when performing object-specific grasping, in which the next object to grasp is pre-determined (e.g., by humans), GenerAL can be directly combined with instance or semantic segmentation algorithms [such as Schnieders et al. (2019)] to achieve object-specific grasping in the following procedure:

1. A segmentation algorithm segments the cluttered scene using RGB image, depth image, or both.
2. The next target object to grasp is determined via some mechanism, such as by humans.
3. Instead of performing a *global* spatial-softmax sampling on the entire depth image for a_t^{position}, GenerAL performs a *local* spatial-softmax sampling only from the pixels that belong to the selected target object, according to the segmentation map.
4. Standard GenerAL procedures then follow.

6.9.3 Semantic grasping

Semantic grasping capability can be important in some manipulation tasks. GenerAL focuses on maximizing grasp success rate and therefore has its learned preferences for how to grasp the target object. GenerAL can be extended to semantic grasping in several ways:

1. The grasp position and orientation can be directly restricted to a certain range of values.
2. The learned distribution can be modified or masked out to reflect grasp preferences.
3. An orientation penalty can be added during training to discourage the policy from generating undesirable grasping orientations.
4. GenerAL can learn specific tasks like pick-and-place by changing the reward specification in its formulation to reflect the task.
5. GenerAL can grasp a specific part of an object similarly as the object-specific grasping procedure in Sect. 6.9.2. The primary difference is that in object-specific grasping, GenerAL performs local spatial-softmax sampling on the object-specific pixels, while in part-specific grasping, the pixel is selected directly by humans.

7 Conclusion

This work presents a novel and general framework to learn high-DOF grasping for hands with arbitrary degrees of freedom, without requiring any database of grasp examples. Our framework uses a policy gradient formulation and a learned attention mechanism to generate full 6-DOF grasp poses as well as all finger joint angles to pick-up objects in dense clutter given a single depth image. Entirely trained in simulation, our framework achieves 96.7% (single-seen), 93.3% (single-novel), 92.9% (cluttered-seen), 91.9% (cluttered-novel) pick-up success rates on the physical Staubli–Barrett robot and 96.0%, 96.0%, 94.2%, 93.5% respectively on the UR5-Seed robot, as well as statistically similar performance in simulation. These results, along with many ablation studies, exhibit the framework's robustness across robots with different degrees of freedom and good generalization to real-world grasping, cluttered scenes, novel objects, and non-top-down camera viewing angles.

Acknowledgements We are thankful to Wei Zhang and everyone at Columbia University Robotics Lab for useful comments and suggestions.

Compliance with ethical standards

Conflicts of interest Jacob Varley is a member of Robotics at Google. Peter K. Allen has received a research grant from Google Inc.

A Appendix

A.1 Mechanical structure of robotics hands in experiments

A.1.1 Staubli–Barrett

The finger joint angles of the Barrett Hand range from 0 (open) to 2.44 rad (close) for finger-1, finger-2 and finger-3 and from 0 to π rad for the lateral-spread. We used the original hand throughout all experiments.

A.1.2 UR5-Seed

For the anthropomorphic Seed hand, flexion joints curl the fingers toward the palm of the hand between 0 (open) to 1.48 rad (close), while adduction joints spread the fingers apart from 0 to $\pi/2$ rad. Therefore, only the thumb on the Seed hand may spread. The wrist-rotation, wrist-flexion and wrist-adduction DOFs, in addition to the finger flexion and adduction DOFs, contribute to a total of 8 DOFs for the Seed hand. However, these three wrist-DOFs belong to arm-DOFs Ψ_{arm} because they do not actuate any fingers. Each robotic finger consists of 3 joints, all of which are controlled by a single dyneema tendon (i.e. a Kevlar fiber string). The ring and pinky fingers are coupled such that a single actuator is responsible for the flexion of both fingers, while all other fingers are controlled by their own actuator, respectively.

A.2 Mechanical modification to the RH8D Seed hand

As shown in Fig. 6, fingers demonstrate under-actuated behavior during control. As the actuator moves through its full range of motion, the tendon pulls on the distal, intermediate, and proximal joints of a given finger in three stages:

1. Initially, the finger is fully open before closing (Fig. 6a). As the finger starts to close, the distal joint rotates almost completely before the intermediate joint even begins to move.
2. The distal joint, reaching its maximum displacement, ceases to rotate while the intermediate joint continues to move (Fig. 6b).
3. The proximal joint reaches its limit right after the intermediate joint stops moving (Fig. 6c).

While the fingers are anthropomorphic in design, human fingers would move differently to grasp an object. In particular, the proximal joint of the human finger will traditionally curl before any other joint at a greater rate than the distal joint. Fingers of the human hand will also rarely ever settle in a hook-like position as depicted in Fig. 6b—the fingers must sweep through a greater volume in order to sufficiently

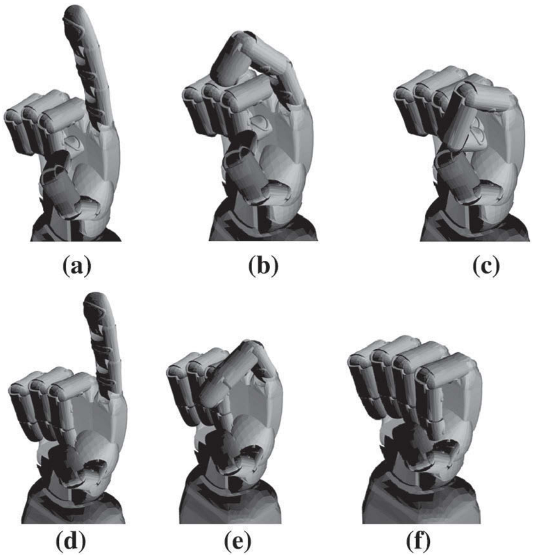


Fig. 6 The finger-closing behavior of the Seed hand before **a–c** and after **d–f** modification, using the index finger as an example

contact and grasp objects beyond thin cylindrical geometries. With these observations in mind, tape is applied around the distal joint (Fig. 1b) to inhibit it from rotating (Fig. 6d–f). This configuration effectively reduces the number of underactuated joints on each finger by one. Figure 6f depicts the resulting grasp form. Also, the fingers of the Seed hand are composed of very low-friction thermoplastic material. To increase friction on the fingers, we added small rubber caps to the fingertips (Fig. 1b).

References

Akinola, I., Varley, J., Chen, B., & Allen, P. K. (2018). Workspace aware online grasp planning. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE (pp. 2917–2924).

Berenson, D., & Srinivasa, S. S. (2008). Grasp synthesis in cluttered environments for dexterous hands. In *8th IEEE-RAS international conference on humanoid robots, 2008*, IEEE (pp. 189–196).

Berenson, D., Diankov, R., Nishiwaki, K., Kagami, S., & Kuffner, J. (2007). Grasp planning in complex scenes. In *7th IEEE-RAS international conference on humanoid robots, 2007*, IEEE (pp. 42–48).

Bohg, J., Morales, A., Asfour, T., & Kragic, D. (2014). Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, *30*(2), 289–309.

Bousmalis, K., Irpan, A., Wohlhart, P., Bai, Y., Kelcey, M., Kalakrishnan, M., Downs, L., Ibarz, J., Pastor, P., & Konolige, K., et al. (2018). Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE (pp. 4243–4250).

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., & Su, H., et al. (2015). Shapenet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012.

Chen, X., Chen, R., Sui, Z., Ye, Z., Liu, Y., Bahar, R., & Jenkins, O. C. (2019). Grip: Generative robust inference and perception for semantic robot manipulation in adversarial environments. arXiv preprint arXiv:1903.08352.

Ciocarlie, M., Goldfeder, C., & Allen, P. (2007). Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem. In *Robotics: Science and systems manipulation workshop-sensing and adapting to the real world*, Citeseer.

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

Coumans, E., & Bai, Y. (2016). *Pybullet, a python module for physics simulation for games, robotics and machine learning*. San Francisco: GitHub.

Fischinger, D., Vincze, M., & Jiang, Y. (2013). Learning grasps for unknown objects in cluttered scenes. In *2013 IEEE international conference on robotics and automation*, IEEE (pp. 609–616).

Gualtieri, M., & Platt, R. (2018). Learning 6-DOF grasping and pick-place using attention focus. arXiv preprint arXiv:1806.06134.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning (ICML)*.

Hang, K., Stork, J. A., & Kragic, D. (2014). Hierarchical fingertip space for multi-fingered precision grasping. In: *2014 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1641–1648).

Hsiao, K., & Lozano-Perez, T. (2006). Imitation learning of whole-body grasps. In: *2006 IEEE/RSJ international conference on intelligent robots and systems*, IEEE (pp. 5657–5662).

James, S., Wohlhart, P., Kalakrishnan, M., Kalashnikov, D., Irpan, A., Ibarz, J., Levine, S., Hadsell, R., & Bousmalis, K. (2018). Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. arXiv preprint arXiv:1812.07252.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., & Vanhoucke, V., et al. (2018). QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. arXiv preprint arXiv:1806.10293.

Lenz, I., Lee, H., & Saxena, A. (2015). Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, *34*(4–5), 705–724.

Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, *17*(1), 1334–1373.

Levine, S., Pastor, P., Krizhevsky, A., & Quillen, D. (2017). Learning hand-eye coordination for robotic grasping with large-scale data collection. In: *International symposium on experimental robotics (ISER)*.

Lin, T., Dollár, P., Girshick, R. B., He, K., Hariharan, B., & Belongie, S. J. (2016). Feature pyramid networks for object detection. *CoRR*,. arXiv:1612.03144.

Lu, Q., Chenna, K., Sundaralingam, B., & Hermans, T. (2017) Planning multi-fingered grasps as probabilistic inference in a learned deep network. In: *International symposium on robotics research*.

Lundell, J., Verdoja, F., & Kyrki, V. (2019). Robust grasp planning over uncertain shape completions. arXiv preprint arXiv:1903.00645.

Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J. A., & Goldberg, K. (2017). Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. arXiv preprint arXiv:1703.09312.

Miller, A. T., Knoop, S., Christensen, H. I., & Allen, P. K. (2003). Automatic grasp planning using shape primitives. In: IEEE international conference on robotics and automation, 2003. Proceedings. ICRA'03, IEEE (Vol. 2, pp. 1824–1829).

Mnih, V., Heess, N., & Graves, A., et al. (2014). Recurrent models of visual attention. In *Advances in neural information processing systems* (pp 2204–2212).

Morrison, D., Corke, P., & Leitner, J. (2018). Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. In *Robotics: science and systems (RSS)*.

Okamura, A. M., Smaby, N., & Cutkosky, M. R. (2000). An overview of dexterous manipulation. In *IEEE international conference on robotics and automation. Proceedings. ICRA* (Vol. 1, pp. 255–262).

Pan, J., Sayrol, E., Giro-i Nieto, X., McGuinness, K., & O'Connor, N. E. (2016). Shallow and deep convolutional networks for saliency prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 598–606).

Quillen, D., Jang, E., Nachum, O., Finn, C., Ibarz, J., & Levine, S. (2018). Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods. arXiv preprint arXiv:1802.10264.

Rosales, C., Porta, J. M., & Ros, L. (2011). Global optimization of robotic grasps. In *Proceedings of robotics: science and systems VII*.

Saut, J. P., Sahbani, A., El-Khoury, S., & Perdereau, V. (2007). Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces. In *2007 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE (pp. 2907–2912).

Schmidt, P., Vahrenkamp, N., Wächter, M., & Asfour, T. (2018). Grasping of unknown objects using deep convolutional neural networks based on depth images. In *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE (pp. 6831–6838).

Schnieders, B., Luo, S., Palmer, G., & Tuyls, K. (2019). Fully convolutional one-shot object segmentation for industrial robotics. In *Proceedings of the 18th international conference on autonomous agents and multiagent systems, international foundation for autonomous agents and multiagent systems* (pp. 1161–1169).

Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. In *International conference on learning representations*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge: MIT Press.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE (pp. 23–30).

Tobin, J., Biewald, L., Duan, R., Andrychowicz, M., Handa, A., Kumar, V., McGrew, B., Ray, A., Schneider, J., & Welinder, P., et al. (2018). Domain randomization and generative models for robotic grasping. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE (pp. 3482–3489).

Varley, J., Weisz, J., Weiss, J., & Allen, P. (2015). Generating multi-fingered robotic grasps via deep learning. In: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE (pp. 4415–4420).

Varley, J., DeChant, C., Richardson, A., Ruales, J., & Allen, P. (2017). Shape completion enabled robotic grasping. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE (pp. 2442–2447).

Viereck, U., Pas, A., Saenko, K., & Platt, R. (2017). Learning a visuomotor controller for real world robotic grasping using simulated depth images. In *Conference on robot learning (CORL)*.

Wang, S., Jiang, X., Zhao, J., Wang, X., Zhou, W., & Liu, Y. (2019). Efficient fully convolution neural network for generating pixel wise robotic grasps with high resolution images. arXiv preprint arXiv:1902.08950.

Wang, W., Shen, J., & Ling, H. (2018). A deep network solution for attention and aesthetics aware photo cropping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*, 1531–1544.

Watkins-Valls, D., Varley, J., & Allen, P. (2019). Multi-modal geometric learning for grasping and manipulation. In: *2019 International conference on robotics and automation (ICRA)*, IEEE (pp. 7339–7345).

Wu, B., Akinola, I., & Allen, P. K. (2019a). Pixel-attentive policy gradient for multi-fingered grasping in cluttered scenes. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*.

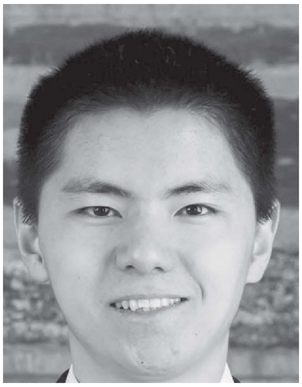
Wu, B., Akinola, I., Varley, J., & Allen, P. K. (2019b). MAT: Multi-fingered adaptive tactile grasping via deep reinforcement learning. In *Conference on robot learning (CORL)*.

Zeng, A., Song, S., Welker, S., Lee, J., Rodriguez, A., & Funkhouser, T. (2018a). Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE (pp. 4238–4245).

Zeng, Z., Zhou, Z., Sui, Z., & Jenkins, O. C. (2018b). Semantic robot programming for goal-directed manipulation in cluttered scenes. In *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE (pp. 7462–7469)

Zhao, J., Liang, J., & Kroemer, O. (2019). Towards precise robotic grasping by probabilistic post-grasp displacement estimation. Technical report, EasyChair.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Bohan Wu is an M.S. candidate in Computer Science at Columbia University. He received his Bachelor's Degree in Computer Science and Economics at Duke University, Durham, NC, USA. His research interests center around robotics and machine learning.

Iretiayo Akinola received a B.S. degree in Electronic and Electrical Engineering from Obafemi Awolowo University, Ile-Ife, Nigeria and an M.S. degree in Electrical Engineering from Stanford University, USA. He is currently pursuing a Ph.D. degree with the Computer Science Department, Columbia University, New York, USA. His research interests include vision-based robotic grasping and manipulation using planning and learning methods.

Abhi Gupta is a B.Sc. candidate at Columbia University. He is studying Computer Science and Operations Research at the School of Engineering and Applied Sciences. He enjoys developing electromyography-driven teleoperation systems. His research interests include reinforcement learning and robotic grasping.

Feng Xu received the B.S. degree in Vehicle Engineering from Jilin University in China. He is currently pursuing M.S. degree in Mechanical Engineering in Columbia University, New York, NY, USA. His research interests include robotic learning and perception.

Jacob Varley is a member of Robotics at Google. Prior to Google, Jake received his doctoral degree in Computer Science from Columbia University, where he was advised by Prof. Peter Allen. His research focuses on building robotic manipulation systems. He has a strong interest in the interplay between robotics, simulation, and machine learning. While at Columbia, Jake interned as a machine learning researcher at Clarifai, and spent the summer in the Hubo Lab at KAIST as an NSF EAPSI Fellow. Prior to graduate school, Varley earned a Bachelors in Computer Science (6-3) from MIT.

David Watkins-Valls is a Ph.D. Candidate in robotics at Columbia University. He received a B.S. degree in computer science from Columbia University in 2016 and an M.S. degree in computer science from Columbia University in 2017. He is an ARL fellow working with Nicholas Waytowich as part of navigation research. His work has been shown in several venues including ICRA and NEMS.

Peter K. Allen received the A.B. degree in mathematics-economics from Brown University, Providence, RI, USA, the M.S. degree in computer science from the University of Oregon, Eugene, OR, USA, and the Ph.D. degree in computer science from the University of Pennsylvania, Philadelphia, PA, USA. He is a Professor of Computer Science at Columbia University, New York, NY, USA, and the Director of the Columbia Robotics Lab. He is the recipient of the CBS Foundation Fellowship, Army Research Office Fellowship, and the Rubinoff Award for innovative uses of computers from the University of Pennsylvania. His current research interests include robotic grasping, 3-D vision and modeling, and medical robotics.