RTML: Small: Collaborative: Robust Real=time Robot Perception through Combined Discriminative-Generative Techniques

Overview: In order for robots to autonomously perform manipulation tasks in human environments, they must know what objects are in their environment. Such understanding of objects, as well as their pose and geometry, is needed for robots to reason properly. For common human environments, such perceptual understanding remains an open problem that is especially challenging for complex cluttered environments. Object detection techniques used for robot perception have greatly improved over the past few years, due in part to the renewed interest and recent advancements of convolutional neural networks. However, these improvements in detection have often incurred two considerable downsides: 1) the expense of significant energy consumption and computational inefficiency and 2) the trend of overfitting to improve object detection performance. We propose the **Convolution-oriented Generative Inference (CoGI)** project to address both of these shortcomings through a two-stage approach to detection. Our work towards CoGI will reexamine the structure of convolutional neural networks (CNNs) for resource efficiency and the use of CNNs in the context of robot perception. Critical to CoGI is recasting CNNs as an efficient weighted convolutional operator. Such a recasting avoids the need for hard thresholding, and allows for a second stage of generative inference to robustly and simultaneously recognize objects and their poses.

Key Words: scalability for co-robots, perception for manipulation; hardware-software co-design.

Intellectual merit: We propose the CoGI project to develop robot perception that offers a seamless way to get the best of both CNNs and generative and probabilistic methods. Our ultimate goal is to reach the point where mobile manipulation robots can compute all information needed for perception on-board and in real time. Towards this goal, we need to rethink how computational resources are designed, allocated, and used in these systems. With respect to resources, CNNs perform with higher accuracy by using high-precision arithmetic, and using architectures with many layers, both of which have a high computational and energy cost. Such machine precision may be unnecessary, and perhaps misdirected. The common case for such precision is motivated by the scenario of hard thresholding of image-space bounding boxes for object detection. Instead, we aim to treat the output of the CNN as purely an efficient convolution operation. Our approach to perception would enable more efficient traditional use of bounding box-style object detection, as well as enable new forms of perception that have the promise of greater robustness. The CoGI project will pursue the following goals:

- new generative-discriminative algorithms better suited for scene perception and object detection in unstructured, changing environments
- our target application is robot manipulation; however, thes algorithms could be used for a range of applications that require object detection.
- current proposal often are not suitable for real-time execution, especially under constrained resources (both computational and memory). What we offer is a pathway for real-time robot manipulation
- Our software algorithms will be developed with hardware implementation in mind. We will propose hardware compatible implementations that take advantage of pipelining, parallel execution, memory compression, and approximate computing in order to achieve real-time execution goals (even with limited hardware resources)
- For this project, our hardware exploration will be limited to FPGAs (or SoCs with FPGAs). However, our acceleration techniques (in HW and SW) will lend themselves to future implementations in custom designed integrated circuit (including those composed of emerging technologies.

1. **Simultaneous object and pose recognition**: Develop robust generative approaches to robot perception that use the output of CNNs purely as a weighted convolution such that hard decisions are avoided by the CNN itself. In CoGI, output of CNN convolution is used by generative methods to infer objects and their poses while maintaining a distribution of possible detections; and,
2. **Efficient architectures for convolution by CNN**: Develop efficient real-time CNN architectures directly in hardware suited to standalone convolution computation on-board robots; where optimizations focused towards convolution could drastically reduce computational cost and improve energy consumption.

Broader Impact: This projects aims to develop a new computational flow to aid in the design of robot perception techniques. Successful implementation of this project will pave the way in the near future for autonomous robots to perceive, grasp, and operate in real time objects that are found in a range of common human environments, which have proven difficult to this point. By autonomous, these robots will need to do all computation on-board, without relying on in-the-cloud computation. Graduate and undergraduate students will be involved. Emphasizing their prior experience with outreach efforts, women and underrepresented minorities will be especially encouraged to take part in this research project and engaged through new course offerings.

Project Description

1 Intellectual Merit

Technological advancements have led to a proliferation of machine learning systems used to assist humans in a wide range of tasks. However, we are still far from accurate, reliable, and resource-efficient operations of these systems. Convolutional neural networks (CNNs) for object recognition have now gained widespread use. For instance, robotic systems use CNNs for scene understanding [120], dexterous object manipulation [37], autonomous driving [8], and a plethora of compelling applications. While demonstrating high accuracy for certain applications, such CNNs incur the cost of vastly complex parametric models with high energy consumption profiles. In addition, CNNs recognition systems may also be vulnerable to errors (both benign and malicious) due to the effects of overfitting during the training process. Distorted objects and/or objects captured under poor lighting conditions could be enough to defeat the recognition abilities of a CNN [24]. Such perception errors can lead to (potentially disastrous) outcomes for systems acting in the real world. These challenges for robust artificial intelligence and machine learning become that much more challenging when an adversary can modify the environment to exploit the vulnerabilities of a CNN. For instance, in the context of object recognition for a robotic system, a possible malicious attack (through simple modifications of an environment) has the potential to drastically alter and even manipulate a robots final behavior.

The **primary objective** of this project is to advance research in machine learning algorithms amenable to **real-time computing by synthesizing deep learning and probabilistic generative inference**. In particular, by ***S***ynthesizing ***A***daptive ***G***enerative ***E***stimation (***SAGE***), we will explore generative-discriminative algorithms [66], [105], [14] that combine inference by deep learning (or other discriminative techniques) with sampling and probabilistic inference models to achieve robust and adaptive perception in adversarial environments. In comparison to existing research, the value proposition for SAGE is to get the best out of existing approaches to computational perception and robotic manipulation while avoiding their shortcomings. We want the robustness of belief space planning [52], [51] [53] without its computational intractability. The recall power of neural networks without excessive overfitting [61]. The efficiency of deterministic inference without its fragility to uncertainty [29], [73]. We posit such a balance can be achieved through synthesizing these ideas into a generative-discriminative approach to robot perception and, more generally, robust machine learning. Generative-discriminative algorithms will be especially advantageous when exposed to adversarial attack, in comparison to foundational ideas in this space [76], [77], [67], [49], [17]. Furthermore, we expect our approach will be more generally applicable to guard against broad categories of attack within constraints for accuracy, real-time computation, and energy-efficiency, and will inform the building of more efficient and robust systems dependent on machine learning.

To get the best of both worlds, we consider the state of the art as the relative strengths and weaknesses of deep learning and probabilistic inference for robust AI. We are particularly interested in complementary properties of these methods for making perceptual decisions, where the weaknesses of one can be addressed by the strengths of the other. Despite the strengths of CNNs, they have several shortcomings that leave them vulnerable to adversarial action, such as their *opacity* in understanding how its decisions are made, *fragility* for generalizing beyond overfit training examples, and *inflexibility* for recovering when false decisions are produced. For these methods, Goodfellow *et al.* [33] demonstrated that adversarial examples are misclassified both in the case of different architectures or different subsets of the training data. These weaknesses for CNNs play to the strengths of robustness for generative probabilistic inference techniques, which are inherently *explainable, general, and resilient* through the process of generating, evaluating, and maintaining a distribution of many hypotheses representing possible decisions. However, this robustness comes at the cost of computational efficiency. Probabilistic inference, in contrast to CNNs, is often computationally intractable with complexity that grows exponentially with the number of variables.

Generative-discriminative algorithms [106, 66] offer a promising avenue for robust perception. Such methods combine inference by deep learning (or other discriminative techniques) with sampling and probabilistic inference models to achieve robust and adaptive perception in adversarial environments. The value proposition for generative-discriminiative inference is to get the best out of existing approaches to computational perception and robotic manipulation while avoiding their shortcomings.

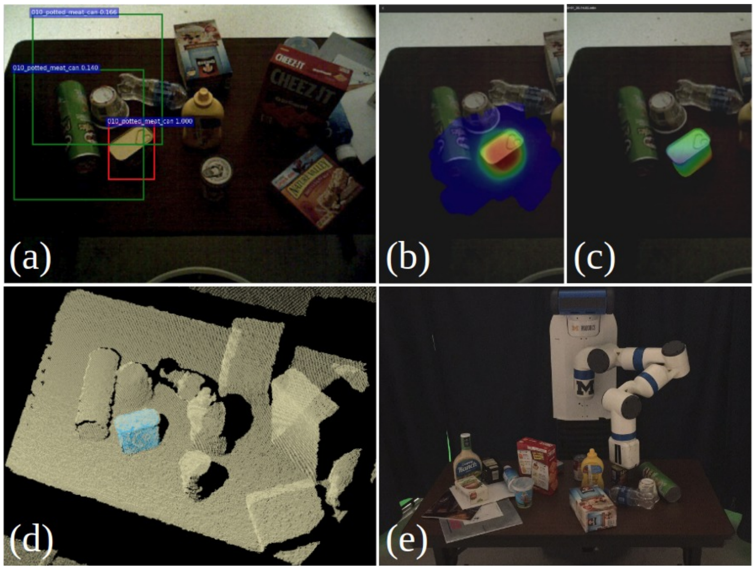

010_potted_meat_can 0.166
010_potted_meat_can 0.140
010_potted_meat_can 1.000
(a)
(b)
(c)
(d)
(e)

Figure 1: Our system perceiving and grasping an object in adversarially darkened lighting. Our proposed approach uses two stages of (a) CNN-based object detection bounding boxes with confidence score greater than 0.1 (green boxes), shown along with the ground truth (red box), and (b) sample-based generative inference. The (c) resulting estimate and (d) its localized pose (highlighted in cyan) enables (e) the Michigan Progress Fetch robot to accurately grasp the can of Spam.

Our project proposes SAGE as a two-stage method to explore generative-discriminative inference for robust perception in adversarial environments that combine the strengths of deep learning and probabilistic inference. Our efforts for SAGE are centered on formulating a joint model [103], [104], [106] that maintains a belief over possible object detections and poses, from which a scene estimate can be taken. This model leads us to a two-stage recognition method that considers factors from the process of detection (e.g., CNN-based convolution [32]) and process of object pose inference (e.g., Monte Carlo Localization [20]). Estimates taken from this joint model provide a final decision about the scene to be acted upon for manipulation, while maintaining probability mass about other viable explanations of the scene. A central point of exploration for SAGE is to avoid making hard decisions and thresholds until a final detection estimate is required for recognition and the consequent robot action. For example, in the first stage, a CNN would return the evaluation of all possible bounding box regions for all object labels in a color image. In the second stage, this convolution can then be used as a factor in a generative Monte Carlo sampling process for pose estimation on depth images. Fig. 1 shows such a robot manipulation task under dark scene.

While our approach will be evaluated within the framework of robot manipulation, we expect SAGE to be widely applicable to other systems that require object detection and scene perception. In addition, we posit that our approach will prove more robust for handling objects and scenes that have been maliciously manipulated or altered. Importantly, we offer an approach amenable to efficient acceleration, both in terms of software and hardware, which will allow for real-time machine learning, even within computationally and energy constrained platforms.

Given the PIs' expertise in robotics and design of energy-efficient systems, this project will initially focus on **real-time machine learning techniques for scene perception, object recognition, and pose estimation** under challenging (or even adversarial) environments, especially within the context of goal-directed robot manipulation within a confined embedded system. We will eventually broaden our scope to apply our techniques to provide users with **advanced information for scene understanding and detailed object recognition**. We will also propose adaptation to our approach to handle **computationally-efficient dynamic retraining** to handle new objects and scenes. Finally, we will propose **new hardware implementations of these algorithms** to provide real-time execution under limited hardware and power budgets.

2 Related Work

2.1 Perception for Manipulation

Perception is a critical step for robotic manipulation in unstructured environments. Ciocarlie *et al.* [16] proposed an architecture for reliable grasping and manipulation, where non-touching, isolated objects are estimated by clustering the surface normal of RGBD sensor data. The MOPED framework [17] has been proposed for object detection and pose estimation using iterative clustering estimation from multi-view features. A bottom-up approach is taken in [91] using RANSAC and Iterative Closest Point registration (ICP), relying solely on geometric information. Narayanan *et al.* [76] integrated global search with discriminatively trained algorithms to balance robustness and efficiency, which works on multi-object identification, assuming known objects.

For manipulation in dense cluttered environments, ten Pas and Platt [111] showed success in detecting grasp affordances from 3D point clouds. In [110], they sample grasp pose candidates based on their geometric plausibility, from which feasible grasp poses are selected by a CNN. Regarding manipulation with known object geometry models, [103, 21, 122] proposed generative sampling approaches to scene estimation for object poses and physical support relations. However, these methods used object detection bounding boxes with hard thresholding as the prior for generative sampling, which might cause false negatives.

2.2 Object Detection and Pose Estimation

Learning-based approaches have been used as modules in object pose estimation systems, or directly built end-to-end approaches. Sui *et al.* [106] proposed a sample-based two-stage framework to sequential manipulation tasks, where object detection results are used as prior of sample initialization. Mitash *et al.* [72] developed a two-stage approach, which ran stochastic sampling of congruent sets [71] to get object poses based on the semantic map from a segmentation network. Regarding end-to-end systems, PoseCNN [120] was proposed by constructing a neural-network that learned segmentation, object 3D translation, and 3D rotation separately. This work also contributed a large object dataset, called YCB-Video-Dataset, for benchmarking robotics pose estimation and manipulation approaches. DOPE [113] outperformed PoseCNN in estimation accuracy and robustness in dark and occluded scenes by training the network on a generated synthetic dataset from domain-randomization and photo-realistic simulation. Another recent work, DenseFusion [118], utilized two networks to extract RGB and depth features separately. Six-dimensional poses are learned from the combined feature and refined the pose by another residual network module.

Long *et al*. [68] propose fully convolutional networks (FCN) for semantic segmentation by replacing fully connected layers in traditional CNN with 1×1 convolutional layers. FCNs take images of arbitrary size and provide per-pixel classification labels. However, FCNs are not able to separate neighboring objects within the same category to obtain instance-level labels; hence we cannot directly re-task FCN for object detection purposes. Nonetheless, most unified approaches are based on FCN to localize and classify objects using the same networks. Recently, there has been a trend to utilize FCN to perform both object localization and classification [99], [97] [96], [44]. Sermanet *et al.* propose an integrated CNN framework for classification, localization and detection in a multiscale and sliding window fashion [99]. Morris *et al.* [75] propose a fully-convolutional Pyramid Network in operate at successive resolutions as information flows up the pyramid to the lowest resolution. In our preliminary work [66], the input to our first stage is a pyramid of images with different scales in order to generate a heatmap for the second stage. To perform object detection we replace fully connected layers with convolutional layers.

2.3 Adversarial attacks

There has been much exploration and investigation of CNNs for computational visual perception, resulting in a vast body of literature (including [32], [31], [97], [96], [121], [68], [99], [44]). For these methods, Szegedy *et al.* [107] demonstrated that adversarial examples are misclassified by different classifiers both in the case of different architectures or different subsets of the training data[45] [12][114]. These results are confirmed also in cases where the differences between these examples were indistinguishable to the human eye. Kurakin *et al.* [58] confirmed the results in a simple physical scenario. Papernot *et al.* [92] showed a case of a black-box attack against a neural network, where adversaries have no knowledge about the model. Others works proposed a possible solution during the training phase (e.g., Zheng *et al.* [124] presented a stability training method to avoid misprediction due small input distortion).

Similar deep learning methods have also been explored in the context of robot manipulation (e.g., [36], [116], [111], [120], [113]. In such cases, an adversary may not be able to directly alter images observed by a robot, but can alter the environment from which the robot is capturing its image observations. Similar to adversarial image manipulation, natural clutter could also be slightly and maliciously altered to deceive a CNN used for robot perception. In order to deal with adversarial scenarios, it is advantageous to rely not on adversarial training, which inherently relies on guessing the type of attack beforehand. Rather, we propose a technique that is inherently more robust to unknown attacks during the inference stage since it has some means of recovering from misleading information.

2.4 CNNs as a Convolution Engine

While Convolutional Neural Networks push the state-of-the-art in many machine learning applications, they often require millions of expensive floating-point operations for each input classification in order to solve hard problems with high accuracy. In particular, the dominant portion of the computation is performed in the convolutional layers and fully connected layers, which require neurons to perform a weighted sum of its inputs. This computation overhead limits the applicability of CNNs for platforms that are energy-constrained and/or require real-time computation and therefore motivates recent interest in designing low-power, low-latency CNNs based on low-precision computation (using fixed-point computation, or reduced-precision weights) [46, 50, 62, 94, 95, 125, 74]. In addition, alternatives to large, complex CNNs have been developed that offer fewer layers, less parameters, and compression [56, 47, 93, 43, 96].

In some of our recent work, we have analyzed the effect of precision scaling on both network accuracy and hardware cost (including memory footprint, power dissipation, and design area) [39, 109, 108]. These works have also investigated various training time methodologies to better manage the low-precision computation. While use of limited precision in neural networks has been proposed before (e.g., [64, 46, 95, 94, 38]), our work provided comprehensive exploration that precisely quantified the effect of numerical precision on energy consumption and computation time (as a function of network accuracy). In particular, we considered a broad range of numerical precisions and quantizations, including 32-bit floating-point arithmetic, fixed-point arithmetic, power-of-2 quantization of weights, and binary representation of weights. We also proposed increasing the number of operations by increasing network size, as needed to maintain accuracy while spending significantly less for each operation.

3 Preliminary Work

In our preliminary work [66] [14], we have found improved detection accuracy using CNNs with a second stage of generative inference based on importance sampling within a particle filtering algorithm. These findings underscore a basic value proposition: the more particle hypotheses we can computationally evaluate, the better our results will be for object and pose recognition. Our initial findings also suggest that such a two-stage approach can provide robustness to CNN errors, explainability over probable interpretations of a scene, and greater ability to tune for computational efficiency. Our preliminary work also suggests this same approach can provide resistance to errors due to malicious attack. Below, we provide more details on our problem formulation as well as some preliminary results.

3.1 Problem Formulation and Method

Given an RGB-D observation (Z_r, Z_d) from the robot sensor and 3D geometry models of a known object set, our aim is to estimate the conditional joint distribution $P(q, b|o, Z_r, Z_d)$ for each object class o, where q is the six DoF object pose and b is the object bounding box in the RGB image. The problem can be formulated as:

$$
\begin{aligned}
& P(q, b|o, Z_r, Z_d) & (1)\\
& = P(q|b, o, Z_r, Z_d)P(b|o, Z_r, Z_d) & (2)\\
& = \underbrace{P(q|b, o, Z_d)}_{\text{pose estimation}}\underbrace{P(b|o, Z_r)}_{\text{detection}} & (3)
\end{aligned}
$$

Equations (1) and (2) are derived using chain rule statistics and Equation (3) represents the factoring of object detection and pose estimation. Here, we assume that pose estimation is conditionally independent of RGB observation, while object detection is conditionally independent of depth observation.

Ideally, we could use Markov Chain Monte Carlo (MCMC) [42] to estimate the distribution of Equation (1). However, the state space of the entire states is so large that it is intractable to directly compute. End-to-end neural network methods can also be used to calculate the distribution [120, 113, 118]. These results place a heavy reliance on proper coverage of the input space in the training set. This data reliance makes such methods vulnerable to unforeseen environment changes. SUM [106] implements a combination of Equation (1) to filter over hard detections provided by a CNN, thereby enabling it to filter out false positive CNN detections. The limitation of SUM is its inability to recover from false negatives that are eliminated from consideration in object proposal and detection stages. On the other hand, our *GRIP* paradigm is able to compensate for data deficiency by employing a generative sampling method in the second stage.

We propose a two-stage paradigm to combine object detection and pose estimation. In the first stage of inference, PyramidCNN performs object detection and generates a prior distribution $P(b|o, Z_r)$ of 2D bounding boxes for each object label o. In the second stage, we perform generative multi-hypothesis optimization to estimate the joint distribution $P(q, b|o, Z_{(r,d)})$ for each object label o using the first stage output as prior. The second stage is implemented as an iterated likelihood weighting filter [70]:

$$
\underbrace{P(q_0, b_0|o, Z_{(r,d)})}_{\text{Sample Initialization}} = P(q_0|b_0)P(b_0|o, Z_r) \tag{4}
$$

$$
P(q_t, b_t|o, Z_{(r,d)}) = \eta \underbrace{P(Z_{(r,d)}|q_t, b_t, o)}_{\text{Likelihood}} \underbrace{\overline{P}(q_t, b_t|o, Z_{(r,d)})}_{\text{Proposal}} \tag{5}
$$

$$
\begin{aligned}
\overline{P}(q_t, b_t|o, Z_{(r,d)}) = \int\int \underbrace{P(q_t, b_t|q_{t-1}, b_{t-1})}_{\text{Diffusion}} \cdot \\
P(q_{t-1}, b_{t-1}|o, Z_{(r,d)})dq_{t-1}db_{t-1}
\end{aligned} \tag{6}
$$

where η is the normalizing factor. In Equation (4), initial pose q_0 is generated from bounding boxes b_0, which are sampled from the prior distribution generated by first stage. After the second stage, we get a probability distribution of pose estimation as shown in Equation (1). We consider the best estimate as the one with highest probability. Equivalently, best pose q^* satisfies,

$$
q^*, \cdot = \underset{q,b}{\operatorname{argmax}}\, P(q, b|o, Z_r, Z_d) \tag{7}
$$

The goal of the first stage is to provide a probability distribution map for an object class o in a given input image. To achieve this, we exploit the discriminative power of CNNs. Inspired by region proposal networks

(RPN) in [97], our PyramidCNN serves as a proposal method for the second stage. We choose VGG-16 networks [100] to extract features, which are directed to two fully convolutional networks (FCN) [68]: a classifier learning the object labels and a shape network learning the bounding box aspect ratios. The structure of PyramidCNN is detailed in Fig. 2.

The input to our networks is a pyramid of images at different scales. This enables the networks to detect objects with different sizes and appearing at various distances. Thus, the output contains a pyramid of heatmaps representing bounding boxes associated with confidence scores, positions, aspect ratios, and sizes for each object class. Different from end-to-end learning systems, we do not apply any threshold to the confidence scores in order to avoid any false negatives generated by the first stage.

The purpose of the second stage is to estimate the object pose by performing iterated likelihood weighting, which offers us robustness and versatility over the search space. This is critical in our context since the manipulation task heavily depends on the accuracy of pose estimations. We expect the second stage to perform robustly even with inaccurate detection from the first stage.

We use a set of weighted samples $\{q^{(i)}, w^{(i)}\}_{i=1}^{M}$ to represent the belief of object pose, where each 6D sample pose $q^{(i)}$ corresponds to a weight $w^{(i)}$. Given an object class o, its pose q, and the corresponding geometry model, we can render a 3D point cloud observation $\boldsymbol{r}$ using the z-buffer of a 3D graphics engine. Essentially, these rendered point clouds are what would be observed if the object had the hypothesized poses, which we refer to as *rendered samples* hereafter. The samples are initialized according to the first stage output. Our CNN produces a density pyramid that is essentially a list of bounding boxes with confidence scores. We perform importance sampling over the confidence scores and initialize our samples uniformly within the 3D workspaces indicated by sampled bounding boxes as shown in Eqn. (4). More samples are spawned within bounding boxes with higher confidence scores.

The weight of each sample is calculated by the likelihood function, which evaluates the compatibility of a sample with observations as shown in Equation (5). The likelihood function consists of several parts, including bounding boxes weight, raw pixel-wise inlier ratio, and feature-based inlier ratio. We first define the raw pixel-wise inlier function as:

$$\text{Inlier}(p, p^{'}) = \begin{cases} 1, & \text{if } ||p - p^{'}||_2 < \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $p, p^{'} \in \mathbb{R}^3$ refer to a point in observation point cloud $\boldsymbol{z}$ and a point in rendered point cloud from sample pose respectively. A rendered point is considered as an inlier if it is within a certain sensor resolution range ϵ from an observed point. The point-wise inlier ratio of a rendered sample is then defined as:

$$I = \frac{1}{|\boldsymbol{r}|} \sum_{(u,v) \in z} \text{Inlier}(\boldsymbol{r}_{(u,v)}, \boldsymbol{z}_{(u,v)}) \tag{9}$$

where (u, v) refers to 2D image indices in the rendered sample point cloud $\boldsymbol{r}$ and observation point cloud $\boldsymbol{z}$. $|\cdot|$ refers to point cloud size.

Besides raw point-wise inliers, we extract geometry feature point clouds from both rendered samples and observation point clouds and compute feature inlier ratios. Hereby, we enhance the robustness of the likelihood function by considering contextual geometric information from 3D point clouds. This term prunes wrong poses that agree with the observation only in individual points but neglect higher-level geometric information such as depth discontinuity and sharp object surfaces. We apply feature point extraction introduced by Zhang *et al.* [123] based on local surface smoothness,

$$c_{(u,v)} = \frac{1}{|\mathrm{N}(u,v)| \cdot ||\boldsymbol{p}_{(u,v)}||} || \sum_{(u',v') \in \mathrm{N}(u,v)} \Big(\boldsymbol{p}_{(u',v')} - \boldsymbol{p}_{(u,v)} \Big) ||, \tag{10}$$

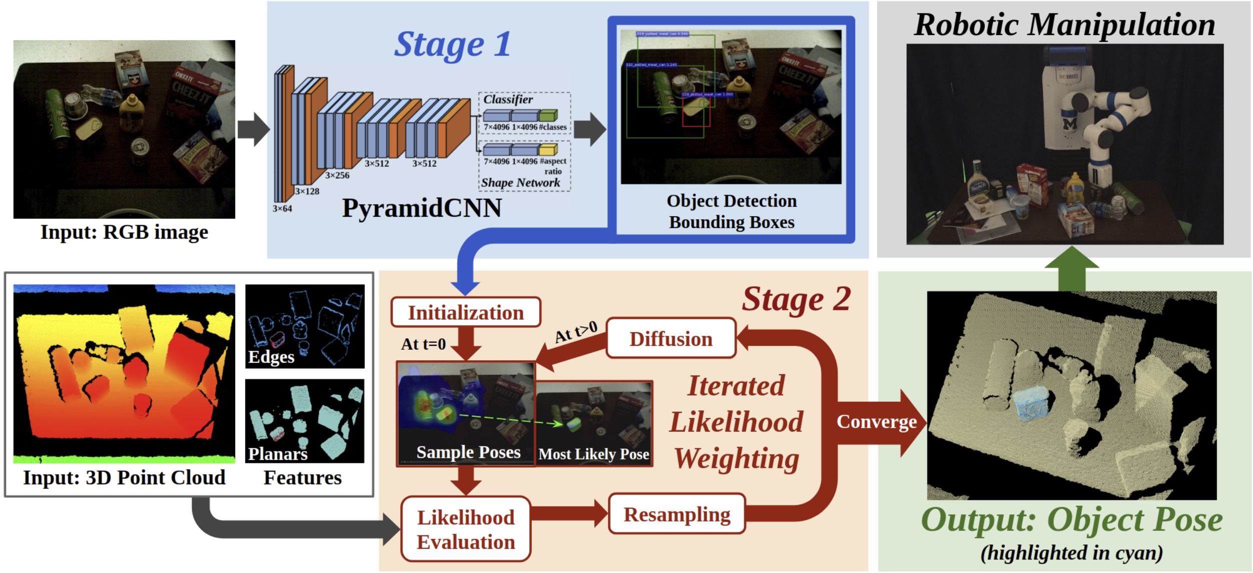


Figure 2: Overview of *GRIP*. The robot operating in a dark and cluttered environment is to grasp the meat can from its RGBD observation. Stage 1 takes the RGB image and generates object bounding boxes with confidence scores. Stage 2 takes the depth image and performs sample-based generative inference to estimate the pose for each object in the scene. The samples in Stage 2 are initialized according to bounding boxes from Stage 1. From this estimate, the robot performs manipulation on the meat can object.

where the smoothness value $c_{(u,v)}$ is calculated by adding all displacement vectors from $\boldsymbol{p}_{(u,v)}$ to each of its neighbor points $\mathrm{N}(u,v)$. The point cloud $\boldsymbol{p}$ here can be either rendered sample $\boldsymbol{r}$ or observation $\boldsymbol{z}$. The value is then normalized by the size of $\mathrm{N}(u,v)$ and the length of vector $\boldsymbol{p}_{(u,v)}$. Intuitively, c describes the depth changing rate within a certain local range, which has larger values in areas with acute depth changes and smaller values where object surfaces are consistent. We extract two features, edge points and planar points, by selecting point sets with largest and smallest c values respectively. To balance feature point density in areas with different observation quality, we set a maximum number of edge points and planar points to be extracted from a certain local area. Essentially, a point at (u,v) can be selected as an edge or a planar point only if its c value is larger or smaller than a threshold and if the number of selected points has not exceeded the limit. We find that the algorithm is insensitive to our feature extraction parameters. Finally, we apply feature extraction on both rendered sample and scene observation point cloud to get sample features and observation features. We use the same inlier calculation in Equations (8) and (9) to calculate feature inlier ratios.

The weight w of each hypothesis q is defined as

$$W(q) = \underbrace{\alpha_{box} w_{box} + \alpha_b I_b}_{\text{network terms}} + \underbrace{\alpha_r I_r + \alpha_e I_e + \alpha_p I_p}_{\text{geometric terms}} \tag{11}$$

where w_{box} is the confidence score of the bounding box. I_r is the ratio of pixel-wise inliers in the whole rendered sample point cloud. I_b is the inlier ratio in the portion of rendered sample that is within the bounding box (I_b is 0 if no rendered sample point falls into the bounding box). I_e and I_p are inlier ratios in sample edges and sample planars with respect to observation features. The coefficients α_* represent the importance of each likelihood term and sum up to 1. Notably, the first two terms, w_{box} and I_b, are heavily determined by the bounding boxes from the first stage PyramidCNN and describe the consistency between pose sample and first stage detection. We refer to them as *network terms*. The last three terms weigh how much the current hypothesis explains itself in the scene geometry. Therefore, we refer to them as *geometric* *terms*.

To produce object pose estimations, we follow the procedure of iterated likelihood weighting by first assigning a new weight to each sample. Resampling is done with replacement according to sample weights. During the diffusion process shown in Equation (6), each pose $q_t^{(i)}$ is diffused in the space subject to zero-mean Gaussian noises $\mathcal{N}_{T,t}(0, \sigma_{T,t}^2)$ and $\mathcal{N}_{R,t}(0, \sigma_{R,t}^2)$ with time-varying variances for translation and rotation respectively. The standard deviations $\sigma_{T,t}$ and $\sigma_{R,t}$ at iteration t are decayed according to $W(q_t^*)$, the weight of best pose estimation q_t^* at that iteration. Bounding boxes $b_t^{(i)}$ are diffused uniformly within the image. The algorithm terminates when $W(q_t^*)$ reaches a threshold $\bar{w}$, or the iteration limit is reached. Finally, we assume the pose weights of objects in the scene will be much higher than those for non-existing objects.

3.2 Preliminary Results

For our preliminary experiments, we use the YCB video dataset [120] as the training data for our first stage PyramidCNN. The YCB video dataset consists of 133,827 frames of 21 objects under normal conditions with balanced and adequate lighting but no occlusion. To test the performance of our two-stage method with baseline methods, PoseCNN [120] and DOPE [113], we collect a testing dataset (i.e., adversarial YCB dataset) from 40 scenes with 15 out of 21 objects from YCB video dataset under adversarial scenarios. In each scene, we place 5-7 different objects on a table and collect seven frames: one in normal lighting, one in darkness, two with different single light sources, and three with different cluttered object placements (see Fig. 3). The dark setting and two single-lighting settings cause bias in image pixels values from the training set and thus undermine network prediction. We refer to these settings as *varied lighting* for simplicity. In addition, object clutter causes occlusions as well as natural information loss and challenges the robustness of pose estimation algorithms. All the scene images and 3D point clouds are gathered by the RGB-D sensor on our Fetch robot. Ground truth bounding boxes and 6D poses are manually labeled.

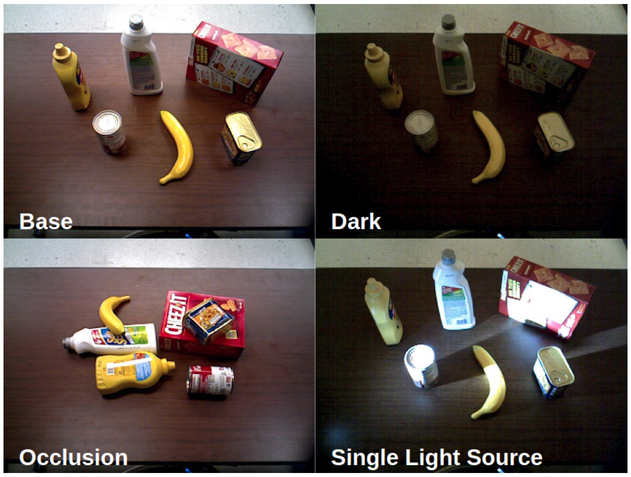


Figure 3: The base-setting data is collected with regular lighting without occlusions. The dark-setting data is collected with all lights off in the room. The single lighting data is collected with a flashlight.

The mean average precision (mAP) of our method for first stage PyramidCNN detection and final pose estimation results are listed in Table 1. We use sparse data to train the PyramidCNN for testing the robustness of the second stage inference. Further, the varied lighting and cluttered occlusion in testing yielded a low mAP score for the PyramidCNN output in comparison to unaltered environments. After second stage of generative sampling, the mAPs are improved beyond 0.5. Thus, our second stage has successfully improved pose estimation performance under adversarial scenarios.

mAP	Base	Varied Lighting	Occlusion
PyramidCNN	0.2824	0.1401	0.1711
GRIP	0.6739	0.5475	0.5069

Table 1: Mean Average Precision (mAP) of first stage PyramidCNN and *GRIP*.

We performed an extensive comparison of our method with PoseCNN (with ICP) on 15 of the 21 YCB

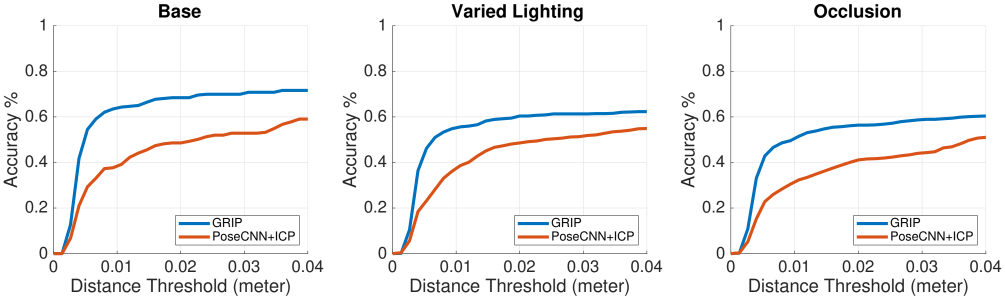


Figure 4: Overall pose estimation accuracy of 15 YCB objects using PoseCNN and our *GRIP* method.

objects. Fig. 4 shows our overall results for each object. *GRIP* outperforms PoseCNN+ICP for most objects under all three settings. All methods have worse performances under varied lighting and occlusions as opposed to basic setting. We can infer the strengths and weaknesses of each method from its performance variance among different objects. For example, PoseCNN with ICP performs better on symmetric objects such as 003_cracker_box and 061_foam_brick as opposed to others such as 021_bleach_cleanser. Symmetric objects contain repetitive features which are more likely to be captured by learning-based systems. *GRIP* performs better on objects that are well recognizable under depth camera. Large and compact objects such as 006_mustard_bottle and 024_bowl naturally generate dense and continuous 3D point cloud observations that effectively capture their geometry. Objects with thin or articulated parts, such as 037_scissors, 052_extra_large_clamp, and 025_mug, produce sparse point clouds around their handle-like parts that do not effectively reveal the scene geometry, especially object orientations. Hence, our *GRIP* algorithm best suits scenarios where rich depth sensory data are available due to detectable object dimensions and surface materials or high-definition depth sensors. Finally, distinguishing near-identical objects remains challenging. For instance, 051_large_clamp and 052_extra_large_clamp have identical colors and shapes and differ only insignificantly in sizes. This results in poor estimation accuracy by all methods. More extensive coverage of our experiments can be found in [14].

4 Research Questions

The key insight of our approach is to avoid hard thresholding, which introduces false positives and false negatives, until a final pose estimate is required. Avoiding hard thresholds increases the possibility of finding the real pose in adversarial environments. In addition, a generative second stage inherently provides an avenue for explainable perception, without requiring deciphering network weights. Also, this generative process readily extends to tracking over multiple instances of time through the inclusion of a proper process model.

Runtime and energy consumption are tightly related to computational complexity and memory accesses. We will consider modifications to the algorithm that inherently allow for better parallelization, pipelining, and streamlining of memory accesses. We will also consider techniques for approximate computing to limit the amount of computation (e.g., so computation is focused where it will most likely lead to useful information). Finally, through adjustments in numerical precision and quantizations, some amount of precision may be traded off for significant improvements in runtime and energy efficiency. Our ultimate goal is to develop a computational flow that will eventually enable robots to perceive and operate on objects in real time using on-board computation (and limited battery resources). We now describe research questions that have arisen during our initial exploration for our approach. These research questions form the basis for our plan of work for the proposed project.

4.1 Can we improve performance with better likelihood functions of object pose estimates?

In our preliminary work [14], particles of possible object poses are re-weighted through likelihood functions, which calculate the geometry consistency to the observed point cloud. More specifically, we use raw pixel-wise inlier functions, which count the number of point pairs that are close enough between the rendered point cloud of object pose and observation point cloud, to calculate probability of a particle estimate. This rendering-based approach gives promising accuracy, but is not computationally efficient. The efficiency can be improved through better representations or evaluations of object poses. Rather than raw rendered point clouds, previously handcrafted features, like point-pair-features [23], curve set features [63], 4-point congruent sets [1], etc., can serve as more compact representations. On the other hand, instead of pixel-wise inlier functions where all computations are done online, fewer feature points can be extracted from pose estimates and observations and matched. Insights from previous template matching methods can also be used to reduce computation by moving large proportions of computation beforehand and saving results to fast check data structures. For instance, various object appearance changes by occlusions or lighting can be stored as entries in hash tables to be compared with pose estimates. A recent work combining deep neural networks and 4-point congruent sets [71] was able to achieve near real-time object pose estimation with high accuracy [72]. We will investigate incorporation of feature-based methods to particle weight calculation to improve efficiency and accuracy.

4.2 Can we achieve better pose estimation by considering constraints between objects?

In our preliminary work, object pose estimation is conducted individually for each of the objects in the dataset, and the final result is drawn from those with highest confidence. The intuition comes when sometimes pose estimations can have infeasible physical relations, like floating over supporters or penetrating into each other. We aim to incorporate heuristics, such as object positional relations, to have better overall estimation. The heuristics can appear to affect the iteration of particles of different objects separately, or form an optimization of the pose estimates over all objects in the scene.

4.3 Can we maintain multi-modal distribution of object poses with multiple instances?

The two-stage method now has an assumption that only one instance appears in the scene, and the final object pose is taken wherever particle filter converges. The case becomes different when multiple instances appear simultaneously in the scene, and we will investigate more advanced particle selection techniques that maintain multiple modes in the probability distribution. This insight has been used as diverse particle selection in [83], [82] to preserve multiple modes during belief propagation for multiple human pose estimation, where optimization is used to select those particles that naturally maintain multiple modes instead of the common re-sampling process that is easy to lose modes with lower probability.

4.4 What kinds of CNNs are best suited for efficient real-time convolution?

CNNs have achieved their high accuracy by using architectures with many layers and high-precision arithmetic, which leads to high computational and energy costs. There have been many recent efforts focused on reducing some of this cost through precision, quantization, and layer adjustments, including our own work in [39], [109], [108]; however, this only addresses part of the problem for robot perception. In particular, these CNNs are still tuned for making hard decisions, particularly in terms of image classification and object detection. The result often leads to overfitting that translates to problematic false positive and negative detections, especially when variations in orientations, lighting, and occlusions come into play.

Figure 5 reports the accuracy achieved by the object detector implemented with various CNN models from the first stage of Faster-RCNN and our Pyramid-CNN approach. We can see that for the object detection stage of Faster-RCNN, more complex networks do not necessarily generate better results than smaller networks, even though more complex networks can produce less prediction error on ImageNet. For Pyramid-

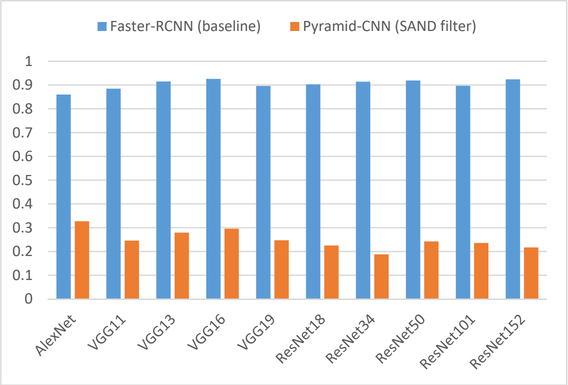


Figure 5: Object detection (first stage) accuracy (mAP) among different networks with progress dataset.

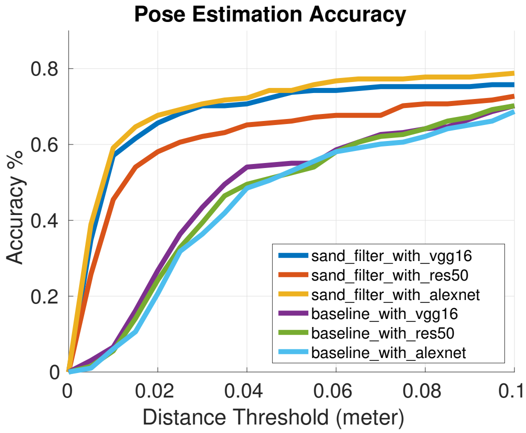


Figure 6: Pose accuracy of our *SAND filter* approach compared with *baseline* approach [66].

CNN, the performance for object detection is much lower than Faster-RCNN since Pyramid-CNN does not perform hard thresholding for the object detection stage; thereby generating many more detection outputs that lead to more false positives. However, our approach makes the generative sampling in the second stage explore and search more thoroughly in order to correct false detections. This also makes our approach less susceptible to mistakes from conditions that mimic adversarial attacks.

Figure 6 compares the pose estimation accuracy of the baseline and our discriminative-generative approach (called SAND) filter using AlexNet, VGG16, and ResNet50 for the object detection stage. We see that our *SAND filter* approach is consistently and significantly better than the *baseline* approach that uses Faster-RCNN with iterative closest point (ICP) within a tighter distance threshold, e.g. 0.02 meter. For the 0.1 meter threshold, the *SAND filter* approach is still better than the *baseline* approach [66].

We propose to evaluate a broad range of CNN architectures, not just to understand their computational complexity vs. accuracy tradeoff within the context of object detection, but more broadly as a convolution engine. Accuracy of the CNN may need to be redefined to better capture the engine's capability to reduce false negatives as its beliefs to the generative-inference engine to make final pose estimations.

4.5 What kinds of inference methods are best paired with the CNN outputs?

In our preliminary work, we have found improved detection accuracy using CNNs with second stage of generative inference based on importance sampling and particle filtering. On the other hand, we have also observed a large computational overhead in order to execute this sampling procedure, making the generative inference step the bottleneck in execution time. In particular, the current particle-based local search is not optimal due to its GPU memory allocation. Two questions arise from these observations: 1) Is such a large number of samples and iterations necessary to achieve high accuracy? and 2) Is there another approach that will reduce total memory usage and memory accesses, while still providing high accuracy? To answer these questions, we will investigate more closely the effects of varying the number of particles on accuracy and runtime. We will also analyze memory utilization more closely, and consider new approaches that may reduce memory accesses and/or high memory latency through better parallelization of our algorithms. Finally, we will consider algorithms outside the mainstream of probabilistic robotics, such as particle swarm optimization [55, 25], which may prove to be more energy-efficient.

Along these same lines, particle-based inference may not be the best approach to real time generative inference within our context of pose estimation. As part of a related project, we are exploring alternative methods of inference using loopy belief propagation on factor graphs. In particular, the PIs have recently proposed a factored approach to estimate the poses of articulated objects using an efficient implementation of nonparametric belief propagation [4]. Inspired by the work of [48] and [102], a "pull" message passing algorithm for nonparametric belief propagation (PMPNBP) is described, where only the most necessary pieces of information between an object's components are passed along. which reduces the

4.6 Does approximate computing have a role in our discriminative-generative approach?

Approximate computing has ben introduced as a means of meeting runtime or energy-efficiency requirements for various applications, especially ones that are tolerant to errors. In our prior work, we have explored techniques to automatically generate approximate circuits [79], [78], or opportunistically allowed approximation under aggressive dynamic voltage over scaling [90]. We will continue this exploration in this project as well. Of course, one of the challenges here is determining how to balance accuracy and runtime, especially understanding that dealing with adversarial clutter will require increasing accuracy of more conventional machine learning techniques.

4.7 What kind of hardware fabrics are suitable for facilitating real-time execution?

As a first evaluation, we intend to map our algorithms onto FPGAs. This will give us a good sense of speedup potential in hardware and will also allow us to experiment with various parallelization and pipelining schemes. Eventually, we plan to explore other fabrics that may bst exploit this parallelization, such as processing in memory. We believe Processing in Memory may be a good direction since we have found memory accesses to be a major bottleneck in our processing time. In addition, we expect iteration and difference operators of the generative algorithms will be a good fit for embedded computation of the PIM. We will leverage of our past experience in near-data-processing [15] to aid in this evaluation.

5 Evaluation and Project Plan

5.1 Relevance to RTML

The SAGE project is well-positioned to explicitly address the aims of real-time machine learning, the goal of achieving ubiquitous collaborative robots, a decades-long aspiration of robotics and AI. In the form of semantic mapping [57, 98], efficient and robust robot perception is essential to realize taskable goal-directed robots [59, 104]. SAGE builds upon our existing efforts in robot perception for problems such as perception for manipulation [103, 106, 21, 66, 14], efficient computation of image features [115], and design of low power CNN architectures [109][39][108]. SAGE also builds on the PIs' past efforts in *customizability* (through robot learning from demonstration [30, 34, 117, 122]), *lowering barriers to entry* (through the *rosbridge* protocol and Robot Web Tools open-source organization [80, 2, 112], and *societal impact* (through his numerous formal and informal efforts to include underrepresented minorities in robotics and computing, see Section 6).

5.2 Project Management

The SAGE project brings together an intellectually diverse team focused on real-time robot manipulation in adversarial environments. This project features researchers with a mix of expertise in design of reliable and energy-efficient computing systems, hardware acceleration, and approximate computing (Bahar) and robotics, artificial intelligence, and computational perception (Jenkins). Our collaboration enables our pursuit of fundamental research questions that would be difficult to pursue individually.

PI Bahar will lead in analyzing computational complexity and its implications on runtime and energy consumptions. More specifically, she will be responsible for leading the hardware design space exploration component of the project to realize a system that optimally combines deep learning with generative inference techniques. PI Jenkins will lead in developing the 2-stage SAGE algorithm and its evaluation, both for static and dynamically changing scenes. Both PIs will be involved in the overall analysis of the proposed approach, including its evaluation on robots in each of our labs at our respective institutions.

It is expected that this project will be executed as a tightly coordinated collaboration via weekly video conference meetings with the entire research group, as well as smaller face-to-face meetings with researchers from each institution. Such weekly meetings have already been occurring between our groups since September 2017. Figure 7 provides a timeline for the project and incorporates the design issues and goals discussed in Section 4.

We expect to have two PhD students working on the project over the course of three years, one from Brown and one from Michigan. Additionally, we also plan to involve multiple undergraduates and masters students from each institution on this project during its lifespan. These undergraduate and masters students may work on several aspects of this project for course credit as part of independent study classes, or as capstone projects, or alternatively they may conduct research over the summer months to help with the various aspects of the project.

This project proposes fundamental research that develops (i) theoretical foundations for robust methods in object classification, recognition, and pose estimation in adversarial environments, (ii) new and efficient hybrids of combined generative-discriminative perception, and (iii) computationally- and memory-efficient implementations of these algorithms using commercially available hardware platforms. Our preliminary results have shown great promise for our SAGE approach. Moving forward, there are several directions we plan to pursue that support the goals described in the GARD BAA. Below we list our expected milestones and directions for future exploration. Roman numerals indicate the phase when they will take place:

1. Evaluating vulnerabilities from adversarial attack: We will perform extensive experiments to understand the performance of neural networks in adversarial and inhospitable environments using various datasets and various levels of adversarial modifications to environments.

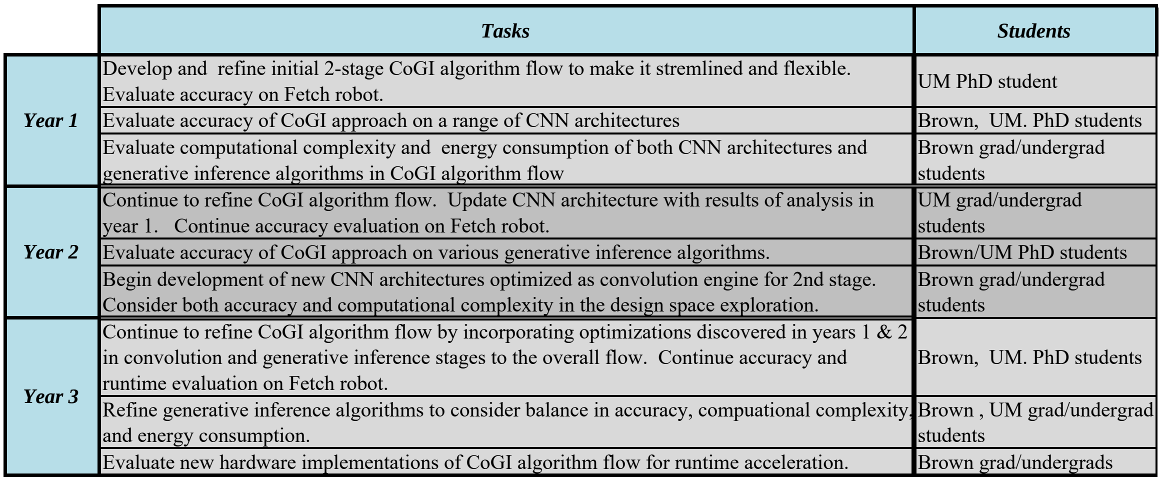
2. Hardware implementation of our current approach: We will implement our current 2-stage approach on commercial FPGAs. This will require more in depth profiling of our particle filtering algorithm, and using hardware optimization techniques that maximize hardware parallelization. We will leverage off our past efforts in hardware acceleration [115], [108], [109], [39].

3. Computational resource evaluation: Considering our current particle filter approach, our goal is to understand if i) such a large number of samples and iterations necessary to achieve high accuracy?, and ii) there are other approaches that will reduce run time, total memory usage, and memory accesses, while still providing high accuracy?

4. Expanding our approach to video and tracking: Phase II will focus on the natural extension of our generative-inference approach to handle robust, identification, localization, and tracking in video streams in the face of adversarial attacks.

5. Exploration of learned multi-modal likelihood functions: Generative filtering provides a straightforward means of sensor fusion across modalities. We will explore methods that fuse multiple deep learning models, one learned for each sensor modality, with a generative state estimator.

6. Hardware acceleration of video streams and multi-modalities: With further computational analysis and more extensive algorithm development, we will implement energy and computationally-efficient versions of our video stream algorithms in reconfigurable hardware.



	Tasks	*Students*
Year 1	Develop and refine initial 2-stage CoGI algorithm flow to make it stremlined and flexible. Evaluate accuracy on Fetch robot.	UM PhD student
	Evaluate accuracy of CoGI approach on a range of CNN architectures	Brown, UM. PhD students
	Evaluate computational complexity and energy consumption of both CNN architectures and generative inference algorithms in CoGI algorithm flow	Brown grad/undergrad students
Year 2	Continue to refine CoGI algorithm flow. Update CNN architecture with results of analysis in year 1. Continue accuracy evaluation on Fetch robot.	UM grad/undergrad students
	Evaluate accuracy of CoGI approach on various generative inference algorithms.	Brown/UM PhD students
	Begin development of new CNN architectures optimized as convolution engine for 2nd stage. Consider both accuracy and computational complexity in the design space exploration.	Brown grad/undergrad students
Year 3	Continue to refine CoGI algorithm flow by incorporating optimizations discovered in years 1 & 2 in convolution and generative inference stages to the overall flow. Continue accuracy and runtime evaluation on Fetch robot.	Brown, UM. PhD students
	Refine generative inference algorithms to consider balance in accuracy, compuational complexity, and energy consumption.	Brown , UM grad/undergrad students
	Evaluate new hardware implementations of CoGI algorithm flow for runtime acceleration.	Brown grad/undergrads

Figure 7: CoGI Project timeline for executing tasks.

5.3 Evaluation Plan

The recognition performance of CoGI will be evaluated with respect to mean Average Precision (mAP) on object detections on objects, 6 DoF accuracy on object poses, computation time and energy use during inference, and, most importantly, pick-and-place accuracy for manipulation in cluttered environments. Manipulation results using SAGE will be evaluated for mobile manipulation tasks (including object sorting) using the Michigan Progress Fetch robot in the laboratory of PI Jenkins as well as newly-acquired robot arms in the laboratory of PI Bahar.

The guiding motivating scenario for evaluations involving mobile manipulation will be sorting and delivery of objects in common human environments, such as in the dormant "Chez Betty" food co-op at Michigan). Beyond the scope of CoGI, this system is envisioned to become a snack delivery system amenable for use in academic/office buildings, and eventually assisted living facilties. In this scenario, a mobile manipulator will need to sort items on cluttered surfaces into appropriate categories, and then manipulate these object and environment features (e.g., doors and elevators) to fulfull orders. The proposed recognition methods will be used for perception in this scenarios, but designed to be generic such that it can be adapted to a range of CNN-based object detectors (or more specifically, convolutional operators). Likewise, the generative inference stage will be designed with flexibility and scalability in mind such that a range of inference techniques can be investigated within the same framework.

6 Broader Impacts

Broader Impact to Society: This projects aims to develop a new computational flow to aid in the design of robot perception techniques. Our approach to perception would enable more efficient traditional object detection, as well as enable new forms of simultaneous object and pose recognition that have the promise of greater robustness. Successful implementation of this project will pave the way in the near future for autonomous robots to perceive, grasp, and operate in real time objects that are found in a range of common human environments, which have proven difficult to this point. By autonomous, these robots will need to do all computation on-board, without relying on in-the-cloud computation. The proposed work has the potential to dramatically enhance the quality, efficiency, and capabilities of future robotic systems.

Undergraduate and Graduate Student Research Opportunities: Previous funding to PIs Bahar and Jenkins enabled them to expose both undergraduate and graduate students to cross-disciplinary research and activities, adding depth and perspective to their research projects. Students will have a chance to be involved in this research project at several levels—reading scientific papers about prior and related work;

understanding, using and designing CNN architectures and machine learning algorithms; testing hardware and software designs on real robots; and presenting their work at relevant conferences and workshops. Moreover, both PIs have actively involved several undergraduate students in her NSF research; the students' contributions were an important part of the work, which often led to publications where they were co-authors [54, 13, 60, 26, 27, 28, 3, 22, 89, 39, 119]. The PIs have found that these projects play a profound role in encouraging students to pursue advanced degrees in computer science or engineering after graduation. The PIs have worked with students from a broad range of backgrounds, including women and students from historically underrepresented groups, and both PIs will continue to encourage diverse student participation in their research.

Outreach to Under-represented Groups: Both PIs have been actively involved in numerous outreach activities over the years and plan to incorporate research from this project into ongoing outreach efforts. PI Bahar has been involved in organizing and speaking at several computer architecture workshops (sponsored by NSF, IBM, CRA-W and CDC) geared toward broadening participation of women and underrepresented minorities in computing (recent examples include [6, 7]). In addition, she has been a mentor for the CRA-W Distributed Research Experiences for Undergraduates (DREU) program for several years. PI Bahar has also initiated a semester-long introductory engineering course for high-school girls [65], and lectured regularly at Brown University's Spira and Artemis summer programs aimed at encouraging high school girls in the Providence area to pursue degrees in engineering and computer science [101, 5].

PI Jenkins has been a mentor for numerous undergraduate student interns, including 14 REUs. These interns have been mostly from Historically Black Colleges and Universities (through the CDC DREU and ARTSI BPC Alliance), include an NSF Graduate Research Fellowship winner. PI Jenkins will remain active in the two major venues for African-American participation in computing: the Conference for African-American Researchers in the Mathematical Sciences (CAARMS) and Tapia Celebration of Diversity in Computing. He has recently been a plenary speaker at both of conferences. Previously, he co-founded the Tapia Robotics Competition in 2007 and served as a lead organizer for CAARMS in 2017. For the past several years, PI Jenkins has lead a delegation of students from Michigan to participate at Tapia, using a patchwork of his discretionary and other sources of funding. He additionally participates in Michigan's activities at the National Society for Black Engineers Conference, including serving on the GEM Grad Lab special session, and for the Atlanta University Center Consortium Dual Degree Engineering Program (AUCC-DDEP). PI Jenkins currently serves on the CRA URMD Grad Cohort Steering Committee.

Curriculum Development Activities: The PIs are teaching several advanced undergraduate/graduate courses tightly related to this research project. Courses include *Computer Architecture*, *Low-power Design*, and *Design of Robotic Systems* (Bahar); *Introduction to Autonomous Robotics / Robot Modeling and Control*, *Autonomous Robotics Design Experience*, and *Robotics Systems Laboratory* and *Human-Robot Interaction Seminar* (Jenkins). As part of the objectives of this research, the PIs intend to extend the integration of research findings into these courses. As an additional means of outreach, the PIs will seek out undergraduate women and underrepresented minorities to help as teaching assistants for these courses. This will not only give these students valuable teaching experience, but will also allow students in these courses to see these TAs as role models.

Open Source Dissemination: Prof. Jenkins' group has a strong record of open source contributions, and furthering community spirit of code sharing and reuse. Recent open source contributions have focused on contributions supporting the Robot Operating System community. The brown-ros-pkg repository disseminated open-source releases of our research [19, 81, 9] and widely used ROS packages that enabled the use of platforms such as the iRobot Create, AR.Drone, Aldebaran Nao, and the original rosbridge/rosjs client/server system [18]. Prof. Jenkins' consistent focus on improving the interoperability and accessiblity of robotics to broader populations of scientists and citizens has led to the founding of the Robot Web Tools (RWT) organization [112, 2, 80]. Since 2012, RWT has been a stalwart multi-organization academia-industry collaboration providing usable and stable software for seamless interoperability between ROS, web technologies, and non-ROS run-time environments (e.g., embedded systems). These prior experiences in disseminating new software tools will be especially valuable for this new research project.

7 Results from Prior NSF Support

In the past several years, the PIs have been supported by several NSF research awards, some of them jointly. Only the most relevant are shown here.

Jenkins IIS-0844486, *CAREER: Robot Learning from Multi-valued Demonstration*, $558K, 6/09–9/14. **Intellectual merit:** The project aimed to provide tractable methods for learning robot controllers from ill-posed multi-valued demonstrations. **Broader impact:** Publications and broad open-source dissemination of source code that introduced extensions of infinite mixture regressors [35] and the Hierarchical Dirichlet Process Hidden Markov Model [9] suited to robot learning from demonstration time-series data. This project also ignited a resurgence of robot web technologies [80, 2, 112] aimed to crowdsource robot learning [19], and led to the emergence of the widely used Robot Web Tools open-source organization (`http://robotwebtools.org/`).

Jenkins IIS-1638047 *NRI: Collaborative Research: Sketching Geometry and Physics Informed Inference for Mobile Robot Manipulation in Cluttered Scenes*, $400K, 9/16–9/19. **Intellectual merit:** This project aims to enable robots to perceive and manipulation objects in cluttered environments through physically plausible scene estimation. This work uses rigid body simulation to ensure physical plausibility of scene estimates [21, 104] and sketching from human users to extract 3D object geometries in support of scene estimation [69]. **Broader Impacts:** This project includes research and mentoring opportunities for undergraduate and underrepresented students in computing and robotics.

Bahar: CCF-1420864 *SHF: Small: Automatic High-Level Synthesis of Approximate Computing Circuits*, 7/2014–7/2017, $450K. **Intellectual merit**: This project investigates new methods for the synthesis of approximate circuits that will be generated directly from their high-level behavioral descriptions. As evidence of research, the project has led to six conference publications [79, 40, 41, 108, 39, 109] and one journal publication [78], plus the tools produced from the project are released to the public domain on github at github.com/scale-lab. **Broader impact**: The project involves three graduate students and one undergraduate (who was a co-author on [39]).

Bahar: CNS-1319095, CSN-1519576: *Transparent and Energy-Efficient Speculation on NUMA Architectures for Embedded Multiprocessor Systems*, 9/2013–9/2016, $424,700. **Intellectual merit**: This grant extends our prior work on transactional memory from multi-core to many-core by considering extensions to cluster-based NUMA architectures. The principal objective of this project is to investigate how speculative hardware can improve energy consumption for multi-threaded applications in high-end embedded NUMA systems. In addition to implementing a lightweight version of hardware transactional memory on a many-core cluster, we also investigated transactional-friendly techniques for memory allocation and using the HTM framework for error recovery from timing errors. As evidence of research, so far this grant has led to eight publication [87, 85, 84, 10, 88, 11, 86, 119]. **Broader impact**: This grant supports an international collaboration that includes student exchanges between countries. It has supported two graduate students and one postdoc from Brown plus undergraduate students under REU supplements. Two undergraduates were first and second author on [119].

E References Cited

[1] Aiger, D., Mitra, N. J., and Cohen-Or, D. 4-points congruent sets for robust pairwise surface registration. *ACM transactions on graphics (TOG) 27*, 3 (2008).

[2] Alexander, B., Hsiao, K., Jenkins, O. C., Lee, J., Suay, B., and Toris, R. Robot Web Tools [ROS topics]. *IEEE Robotics & Automation Magazine 19*, 4 (2012), 20–23.

[3] Alves, N., Buben, A., Nepal, K., Dworak, J., and Bahar, R. A cost effective approach for online error detection using invariant relationships. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 29*, 5 (May 2010), 788–801. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5452115&tag=1.

[4] andShiyang Lu, K. D., Opipari, A., and Jenkins, O. C. Efficient nonparametric belief propagation for pose estimation and manipulation of articulated objects. *Science Robotics 4*, 30 (May 2019).

[5] Artemis. The artemis project. http://cs.brown.edu/people/orgs/artemis/2017, July 2017.

[6] Bahar, R. I. Panel discussion: Preparing to conquer the real world. 3rd Career Workshop for Women and Minorities in Computer Architecture, October 2017.

[7] Bahar, R. I. Panel discussion: Women in academia and industry. Joint CGO/HPCA/PPoPP 2018 Session, February 2018.

[8] Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K. End to end learning for self-driving cars. http://arxiv.org/abs/1604.07316, 2016.

[9] Butterfield, J., Osentoski, S., Jay, G., and Jenkins, O. Learning from demonstration using a multivalued function regressor for time-series data. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids 2010)* (2010), pp. 328–333.

[10] Carle, T., Papagiannopoulou, D., Moreshet, T., Bahar, R. I., and Herlihy, M. A transaction-friendly dynamic memory manager for embedded multicore systems. In *7th Workshop on Theory of Transactional Memory (WTTM)* (July 2015).

[11] Carle, T., Papagiannopoulou, D., Moreshet, T., Marongiu, A., Herlihy, M., and Bahar, R. I. Thrifty-malloc: A hw/sw codesign for the dynamic management of hardware transactional memory in embedded multicore systems. In *Proceedings of the International Conference on Compilers, Architectures and Synthesis for Embedded Systems* (New York, NY, USA, 2016), CASES '16, ACM, pp. 20:1–20:10.

[12] Carlini, N., and Wagner, D. A. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)* (2017), 39–57.

[13] Chang, E., and Jenkins, O. C. Sketching articulation and pose for facial animation. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, Switzerland, Switzerland, 2006), SCA '06, Eurographics Association, pp. 271–280.

[14] Chen, X., Chen, R., Sui, Z., Ye, Z., Liu, Y., Bahar, R. I., and Jenkins, O. C. Grip: Generative robust inference and perception for semantic robot manipulation in adversarial environments. *arXiv preprint arXiv:1903.08352v1* (2017).

[15] Choe, J., Huang, A., Moreshet, T., Herlihy, M., and Bahar, R. I. Concurrent data structures with near-data-processing: an architecture-aware implementation. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)* (June 2019).

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

[17] Collet, A., Martinez, M., and Srinivasa, S. S. The moped framework: Object recognition and pose estimation for manipulation. *The International Journal of Robotics Research* (2011), 0278364911401765.

[18] Crick, C., Jay, G. T., Osentoski, S., Pitzer, B., and Jenkins, O. C. rosbridge: ROS for non-ROS users. In *International Symposium on Robotics Research (ISRR 2011)* (Flagstaff, AZ, USA, August 2011).

[19] Crick, C., Osentoski, S., Jay, G., and Jenkins, O. Human and robot perception in large-scale learning from demonstration. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI 2011)* (2011).

[20] Dellaert, F., Fox, D., Burgard, W., and Thrun, S. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on* (1999), vol. 2, IEEE, pp. 1322–1328.

[21] Desingh, K., Jenkins, O. C., Reveret, L., and Sui, Z. Physically plausible scene estimation for manipulation in clutter. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on* (2016), IEEE, pp. 1073–1080.

[22] Donato, M., Cremona, F., Jin, W., Bahar, R. I., Patterson, W., and Zaslavsky, A. A noise-immune sub-threshold circuit design based on selective use of schmitt-trigger logic. In *ACM Great Lakes Symposium on VLSI* (May 2012), pp. 39–44.

[23] Drost, B., Ulrich, M., Navab, N., and Ilic, S. Model globally, match locally: Efficient and robust 3d object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010), pp. 998–1005.

[24] Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physical-world attacks on deep learning models. In *CVPR* (2018).

[25] Fan, X., Li, X., Wang, X., Xiao, Y., and Zhi, J. An approach based on particle swarm optimization for fast object detection. In *The Fourth International Workshop on Advanced Computational Intelligence* (Oct 2011), pp. 120–124.

[26] Ferri, C., Marongiu, A., Lipton, B., Moreshet, T., Bahar, R. I., Herlihy, M., and Benini, L. SoC-TM: Integrated HW/SW support for transactional memory programming on embedded MPSoCs. In *Proceedings of the 7th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis* (New York, NY, USA, 2011), CODES+ISSS '11, ACM, pp. 39–48. http://doi.acm.org/10.1145/2039370.2039380.

[27] Ferri, C., Wood, S., Moreshet, T., Bahar, I., and Herlihy, M. Embedded-TM: Energy and complexity-effective hardware transactional memory for embedded multi-core systems. *Journal of Parallel and Distributed Computing 70*, 10 (2010), 1042–1052. http://www.sciencedirect.com/science/article/pii/S0743731510000201.

[28] Ferri, C., Wood, S., Moreshet, T., Bahar, R. I., and Herlihy, M. Energy and throughput efficient transactional memory for embedded multicore systems. In *HiPEAC '10: Proceedings of the International Conference on High-Performance Embedded Architectures and Compilers* (2010), vol. 5952 of *Lecture Notes in Computer Science*, Springer, pp. 50–65. http://dx.doi.org/10.1007/978-3-642-11515-8_6.

[29] Fikes, R. E., and Nilsson, N. J. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence 2*, 3-4 (1971), 189–208.

[30] Fod, A., Matarić, M., and Jenkins, O. Automated derivation of primitives for movement classification. *Autonomous Robots 12*, 1 (Jan 2002), 39–54.

[31] Girshick, R. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1440–1448.

[32] Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014), IEEE, pp. 580–587.

[33] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[34] Grollman, D., and Jenkins, O. Dogged learning for robots. In *International Conference on Robotics and Automation (ICRA 2007)* (Rome, Italy, Apr 2007), pp. 2483–2488.

[35] GROLLMAN, D. H., AND JENKINS, O. C. Incremental learning of subtasks from unsegmented demonstration. In *International Conference on Intelligent Robots and Systems (IROS 2010)* (Taipei, Taiwan, Oct 2010), pp. 261–266.

[36] GUALTIERI, M., TEN PAS, A., AND JR., R. P. Category level pick and place using deep reinforcement learning. *CoRR abs/1707.05615* (2017).

[37] GUALTIERI, M., TEN PAS, A., SAENKO, K., AND PLATT, R. High precision grasp pose detection in dense clutter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016), p. 598605. https://doi.org/10.1109/IROS.2016.7759114.

[38] GYSEL, P., MOTAMEDI, M., AND GHIASI, S. Hardware-oriented approximation of convolutional neural networks. In *Workshop contribution at ICLR* (2016). arXiv preprint available at arXiv:1604.03168.

[39] HASHEMI, S., ANTHONY, N., TANN, H., BAHAR, R. I., AND REDA, S. Understanding the impact of precision quantization on the accuracy and energy of neural networks. In *ACM/IEEE Design Automation and Test Conference (DATE)* (March 2017).

[40] HASHEMI, S., BAHAR, R. I., AND REDA, S. DRUM: A dynamic range unbiased multiplier for approximate applications. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design* (Piscataway, NJ, USA, 2015), ICCAD '15, IEEE Press, pp. 418–425.

[41] HASHEMI, S., BAHAR, R. I., AND REDA, S. A low-power dynamic divider for approximate applications. In *Proceedings of the IEEE/ACM Design Automation Conference* (2016), DAC '16.

[42] HASTINGS, W. K. Monte carlo sampling methods using markov chains and their applications.

[43] HOWARD, A. G., ZHU, M., CHEN, B., KALENICHENKO, D., WANG, W., WEYAND, T., ANDREETTO, M., AND ADAM, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* (2017). Preprint available at arXiv:1704.04861.

[44] HUANG, L., YANG, Y., DENG, Y., AND YU, Y. Densebox: Unifying landmark localization with end to end object detection. *arXiv preprint arXiv:1509.04874* (2015).

[45] HUANG, S., PAPERNOT, N., GOODFELLOW, I., DUAN, Y., AND ABBEEL, P. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284* (2017).

[46] HUBARA, I., COURBARIAUX, M., SOUDRY, D., EL-YANIV, R., AND BENGIO, Y. Binarized neural networks. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 4107–4115.

[47] IANDOLA, F., AND KEUTZER, K. Small neural nets are beautiful: Enabling embedded systems with small deep-neural-network architectures. In *Proceedings of the Twelfth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis Companion* (New York, NY, USA, 2017), CODES '17, ACM, pp. 1:1–1:10.

[48] ISARD, M. Pampas: Real-valued graphical models for computer vision. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (2003), vol. 1, IEEE, pp. I–I.

[49] JOHO, D., TIPALDI, G. D., ENGELHARD, N., STACHNISS, C., AND BURGARD, W. Nonparametric bayesian models for unsupervised scene analysis and reconstruction. In *Proceedings of Robotics: Science and Systems* (Sydney, Australia, July 2012).

[50] JUEFEI-XU, F., BODDETI, V. N., AND SAVVIDES, M. Local binary convolutional neural networks. *CVPR as Spotlight* (2016). arXiv preprint available at arXiv:1608.06049.

[51] KAELBLING, L. P., LITTMAN, M. L., AND CASSANDRA, A. R. Planning and acting in partially observable stochastic domains. *Artificial intelligence 101*, 1 (1998), 99–134.

[52] KAELBLING, L. P., LITTMAN, M. L., AND MOORE, A. W. Reinforcement learning: A survey. *Journal of artificial intelligence research 4* (1996), 237–285.

[53] KAELBLING, L. P., AND LOZANO-PÉREZ, T. Integrated task and motion planning in belief space. *The International Journal of Robotics Research* (2013), 0278364913484072.

[54] Katzourin, M., Ignatoff, D., Quirk, L., Laviola, J. J., and Jenkins, O. C. Swordplay: Innovating game development through vr. *IEEE Computer Graphics and Applications 26*, 6 (Nov 2006), 15–19.

[55] Kennedy, J., and Eberhart, R. C. Particle swarm optimization. vol. 4, pp. 1942–1948.

[56] Krizhevsky, A. Learning multiple layers of features from tiny images. Tech. rep., University of Toronto, April 2009. http://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf.

[57] Kuipers, B. The spatial semantic hierarchy. *Artificial Intelligence 119*, 1 (2000), 191–233.

[58] Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial examples in the physical world. *CoRR abs/1607.02533* (2017).

[59] Laird, J. E., Gluck, K., Anderson, J., Forbus, K. D., Jenkins, O. C., Lebiere, C., Salvucci, D., Scheutz, M., Thomaz, A., Trafton, G., et al. Interactive task learning. *IEEE Intelligent Systems 32*, 4 (2017), 6–21.

[60] Lapping-Carr, M., Jenkins, O. C., Grollman, D. H., Schwertfeger, J. N., and Hinkle, T. R. Wiimote interfaces for lifelong robot learning. In *Using AI to Motivate Greater Participation in Computer Science - Papers from the AAAI Spring Symposium* (2008), vol. SS-08-08, pp. 61–66.

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

[62] Li, F., and Liu, B. Ternary weight networks. *arXiv* (2016). Preprint available at arXiv:1605.04711.

[63] Li, M., and Hashimoto, K. Curve set feature-based robust and fast pose estimation algorithm. *Sensors 17*, 8 (August 2017), 1782.

[64] Lin, Z., Courbariaux, M., Memisevic, R., and Bengio, Y. Neural networks with few multiplications. In *International Conference on Learning Representations (ICLR)* (2016). arXiv preprint available at arXiv:1510.03009.

[65] LincolnSchool. World class: Students wrap intro to engineering class at brown university. https://www.lincolnschool.org/news-events/news/ post/world-class-students-wrap-intro-to-engineering-class-at-brown-university-20170429, April 2017.

[66] Liu, Y., Sui, Z., Costantini, A., Ye, Z., Lu, S., Jenkins, O. C., and Bahar, R. I. Robust object estimation using generative-discriminative inference for secure robotics applications. In *International Conference on Computer-Aided Design (ICCAD)* (November 2018).

[67] Liu, Z., Chen, D., Wurm, K. M., and von Wichert, G. Table-top scene analysis using knowledge-supervised mcmc. *Robotics and Computer-Integrated Manufacturing 33* (2015), 110–123.

[68] Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3431–3440.

[69] Maghoumi, M., LaViola Jr., J. J., Desingh, K., and Jenkins, O. C. Gemsketch: Interactive image-guided geometry extraction from point clouds. In *International Conference on Robotics and Automation* (2018). *in press*.

[70] Mckenna, S. J., and Nait-Charif, H. Tracking human motion using auxiliary particle filters and iterated likelihood weighting. *Image Vision Comput. 25* (2007), 852–862.

[71] Mellado, N., Aiger, D., and Mitra, N. J. Super 4pcs fast global pointcloud registration via smart indexing. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 205–215.

[72] Mitash, C., Boularias, A., and Bekris, K. Robust 6d object pose estimation with stochastic congruent sets. *arXiv preprint arXiv:1805.06324* (2018).

[73] Mohan, S., Mininger, A. H., Kirk, J. R., and Laird, J. E. Acquiring grounded representations of words with situated interactive instruction. In *Advances in Cognitive Systems* (2012), Citeseer.

[74] Moons, B., Brabandere, B. D., Gool, L. V., and Verhelst, M. Energy-efficient ConvNets through approximate computing. In *IEEE Winter Conference on Applications of Computer Vision (WACV)* (March 2016), pp. 1–8.

[75] Morris, D. D. A pyramid CNN for dense-leaves segmentation. *15th Conference on Computer and Robot Vision (CRV)* (2018), 238–245.

[76] Narayanan, V., and Likhachev, M. Discriminatively-guided deliberative perception for pose estimation of multiple 3D object instances. In *Robotics: Science and Systems* (2016).

[77] Narayanan, V., and Likhachev, M. PERCH: perception via search for multi-object recognition and localization. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on* (2016), IEEE, pp. 5052–5059.

[78] Nepal, K., Hashemi, S., Tann, H., Bahar, R. I., and Reda, S. Automated high-level generation of low-power approximate computing circuits. *IEEE Transactions on Emerging Topics in Computing PP*, 99 (Aug. 2016).

[79] Nepal, K., Li, Y., Bahar, R. I., and Reda, S. ABACUS: A technique for automated behavioral synthesis of approximate computing circuits. In *Proceedings of the Conference on Design, Automation & Test in Europe* (2014), DATE '14, European Design and Automation Association, pp. 361:1–361:6.

[80] Osentoski, S., Jay, G., Crick, C., Pitzer, B., DuHadway, C., and Jenkins, O. Robots as web services: Reproducible experimentation and application development using rosjs. In *International Conference on Robotics and Automation (ICRA 2011)* (2011).

[81] Osentoski, S., Pitzer, B., Crick, C., Jay, G., Dong, S., Grollman, D., Suay, H. B., and Jenkins, O. C. Remote robotic laboratories for learning from demonstration. *International Journal of Social Robotics 4* (June 2012), 1–13.

[82] Pacheco, J. Diverse particle selection for inference in continuous graphical models. posted web slides, 2018. http://people.csail.mit.edu/pachecoj/pubs/dpmpSeminar.pdf.

[83] Pacheco, J., Zuffi, S., Black, M., and Sudderth, E. Preserving modes and messages via diverse particle selection. In *International Conference on Machine Learning* (2014), pp. 1152–1160.

[84] Papagiannopoulou, D., Capodanno, G., Moreshet, T., Herlihy, M., and Bahar, R. I. Energy-efficient and high-performance lock speculation hardware for embedded multicore systems. *ACM Trans. Embed. Comput. Syst. 14*, 3 (May 2015), 51:1–51:27.

[85] Papagiannopoulou, D., Marongiu, A., Moreshet, T., Benini, L., Herlihy, M., and Bahar, I. Playing with fire: Transactional memory revisited for error-resilient and energy-efficient mpsoc execution. In *Proceedings of the 25th Edition on Great Lakes Symposium on VLSI* (New York, NY, USA, 2015), GLSVLSI '15, ACM, pp. 9–14.

[86] Papagiannopoulou, D., Marongiu, A., Moreshet, T., Herlihy, M., and Bahar, R. I. Edge-TM: Exploiting transactional memory for error tolerance and energy efficiency. *ACM Transactions on Embedded Computing Systems (TECS)* (2017).

[87] Papagiannopoulou, D., Moreshet, T., Marongiu, A., Benini, L., Herlihy, M., and Bahar, R. Speculative synchronization for coherence-free embedded NUMA architectures. In *International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS)* (July 2014), pp. 99–106.

[88] Papagiannopoulou, D., Moreshet, T., Marongiu, A., Benini, L., Herlihy, M., and Bahar, R. I. Hardware transactional memory exploration in coherence-free many-core architectures. *International Journal of Parallel Programming (Springer), 46* (April 2018), 1304–1328.

[89] Papagiannopoulou, D., Prasertsom, P., and Bahar, R. I. Flexible data allocation for scratch-pad memories to reduce NBTI effects. In *IEEE International Symposium on Quality Electronic Design* (March 2013).

[90] Papagiannopoulou, D., Whang, S., Moreshet, T., and Bahar, R. I. IgnoreTM: Opportunistically ignoring timing violations for energy savings using HTM. In *IEEE Design Automation and Test in Europe (DATE)* (March 2019).

[91] Papazov, C., Haddadin, S., Parusel, S., Krieger, K., and Burschka, D. Rigid 3D geometry matching for grasping of known objects in cluttered scenes. *The International Journal of Robotics Research* (2012), 0278364911436019.

[92] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security* (2017), pp. 506–519.

[93] Paszke, A., Chaurasia, A., Kim, S., and Culurciello, E. ENet: A deep neural network architecture for real-time semantic segmentation. *arXiv* (2016). Preprint available at arXiv:1606.02147.

[94] Qiu, J., Wang, J., Yao, S., Guo, K., Li, B., Zhou, E., Yu, J., Tang, T., Xu, N., Song, S., Wang, Y., and Yang, H. Going deeper with embedded fpga platform for convolutional neural network. In *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (New York, NY, USA, 2016), FPGA '16, ACM, pp. 26–35.

[95] Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. XNOR-Net: ImageNet classification using binary convolutional neural networks. *arXiv* (2016). Preprint available at arXiv:1603.05279.

[96] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. *arXiv preprint arXiv:1506.02640* (2015).

[97] Ren, S., He, K., Girshick, R., and Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems* (2015), pp. 91–99.

[98] Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., and Beetz, M. Towards 3d point cloud based object maps for household environments. *Robot. Auton. Syst. 56*, 11 (Nov. 2008), 927–941.

[99] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229* (2013).

[100] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[101] Spira. Spira engineering camp. http://spiraengineeringcamp.wixsite.com, July 2017.

[102] Sudderth, E. B., Ihler, A. T., Freeman, W. T., and Willsky, A. S. Nonparametric belief propagation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2003), IEEE.

[103] Sui, Z., Jenkins, O. C., and Desingh, K. Axiomatic particle filtering for goal-directed robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015), IEEE, pp. 4429–4436.

[104] Sui, Z., Xiang, L., Jenkins, O. C., and Desingh, K. Goal-directed robot manipulation through axiomatic scene estimation. *The International Journal of Robotics Research 36*, 1 (2017), 86–104.

[105] Sui, Z., Ye, Z., and Jenkins, O. C. Never mind the bounding boxs, heres the SAND filters. *arXiv preprint arXiv:1808.04969* (2017).

[106] Sui, Z., Zhou, Z., Zeng, Z., and Jenkins, O. C. SUM: Sequential scene understanding and manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2017), IEEE.

[107] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[108] Tann, H., Hashemi, S., Bahar, R. I., and Reda, S. Runtime configurable deep neural networks for energy-accuracy trade-off. In *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis* (New York, NY, USA, 2016), CODES '16, ACM, pp. 34:1–34:10.

[109] TANN, H., HASHEMI, S., BAHAR, R. I., AND REDA, S. Hardware-software co-design of highly accurate multiplier-free deep neural networks. In *IEEE/ACM Design Automation Conference* (June 2017), pp. 28:1–28:6.

[110] TEN PAS, A., GUALTIERI, M., SAENKO, K., AND PLATT, R. Grasp pose detection in point clouds. *The International Journal of Robotics Research 36*, 13-14 (2017), 1455–1473.

[111] TEN PAS, A., AND PLATT, R. Localizing handle-like grasp affordances in 3D point clouds. In *Experimental Robotics* (2016), Springer, pp. 623–638.

[112] TORIS, R., KAMMERL, J., LU, D., JENKINS, O. C., OSENTOSKI, S., WILLS, M., AND CHERNOVA, S. Robot Web Tools: Efficient messaging for cloud robotics. In *IEEE Intelligent Robots and Systems* (2015), pp. 4530–4537.

[113] TREMBLAY, J., TO, T., SUNDARALINGAM, B., XIANG, Y., FOX, D., AND BIRCHFIELD, S. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790* (2018).

[114] ULIČNÝ, M., LUNDSTRÖM, J., AND BYTTNER, S. Robustness of deep convolutional neural networks for image recognition. In *International Symposium on Intelligent Computing Systems* (2016), Springer, pp. 16–30.

[115] ULUSEL, O., PICARDO, C., HARRIS, C. B., REDA, S., AND BAHAR, R. I. Hardware acceleration of feature detection and description algorithms on low-power embedded platforms. In *International Conference on Field Programmable Logic and Applications (FPL)* (Aug 2016), pp. 1–9.

[116] VARLEY, J., WEISZ, J., WEISS, J., AND ALLEN, P. Generating multi-fingered robotic grasps via deep learning. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Sept 2015), pp. 4415–4420.

[117] VONDRAK, M., SIGAL, L., HODGINS, J., AND JENKINS, O. Video-based 3D motion capture through biped control. *ACM Transaction of Graphics (TOG) (Proceedings of ACM SIGGRAPH) 31*, 4 (2012).

[118] WANG, C., XU, D., ZHU, Y., MARTÍN-MARTÍN, R., LU, C., FEI-FEI, L., AND SAVARESE, S. Densefusion: 6d object pose estimation by iterative dense fusion. *arXiv preprint arXiv:1901.04780* (2019).

[119] WHANG, S., RACHFORD, T., PAPGIANNOPOULOU, D., MORESHET, T., AND BAHAR, R. I. Evaluating critical bits in arithmetic operations due to timing violations. In *IEEE High Performance Extreme Computing Conference* (Sept. 2017).

[120] XIANG, Y., SCHMIDT, T., NARAYANAN, V., AND FOX, D. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)* (2018).

[121] XIE, S., GIRSHICK, R., DOLLÁR, P., TU, Z., AND HE, K. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on* (2017), IEEE, pp. 5987–5995.

[122] ZENG, Z., ZHOU, Z., SUI, Z., AND JENKINS, O. C. Semantic robot programming for goal-directed manipulation in cluttered scenes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (2018), IEEE, pp. 7462–7469.

[123] ZHANG, J., AND SINGH, S. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems* (2014), vol. 2, p. 9.

[124] ZHENG, S., SONG, Y., LEUNG, T., AND GOODFELLOW, I. Improving the robustness of deep neural networks via stability training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 4480–4488.

[125] ZHOU, S., WU, Y., NI, Z., ZHOU, X., WEN, H., AND ZOU, Y. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv* (2016). Preprint available at arXiv:1606.06160.

COLLABORATION PLAN

The interdisciplinary team working on this project brings together researchers with a mix of expertise in design of reliable and energy-efficient computing systems (Bahar) and robotics, artificial intelligence, and computational perception (Jenkins). This collaboration extends our combined efforts to address research questions that would be difficult to pursue individually. It is expected that this project will be executed as a tightly coordinated collaboration via weekly video conference meetings with the entire research group, as well as smaller face-to-face meetings with researchers from each institution. Such weekly meetings have already been occurring between our groups since September 2017.

Facilitating Collaboration

We expect to have two full-time PhD students working on the project over the course of three years, one from Brown and one from Michigan. Additionally, we also plan to involve multiple undergraduates and masters students from each institution on this project during its lifespan. These undergraduate and masters students may work on several aspects of this project for course credit as part of independent study classes, or as capstone projects, or alternatively they may conduct research over the summer months to help with the various aspects of the project. The Michigan and Brown students will be advised separately by the PIs from their respective institutions. However, the two PIs plan to oversee all students jointly, whether they are from their own home institution or not, and will treat their efforts as a unified project. It is expected that the PIs will serve on the dissertation committees of each others PhD students.

Coordinating Meetings

As has already begun, PIs Bahar and Jenkins will continue holding weekly teleconference meetings that include all undergraduate and graduate students involved in the project. In addition, Michigan and Brown students will continue to be in close contact with each other and will plan for additional communication as needed for the project. Likewise, the PIs will supplement the weekly teleconference meeting with one-on-one meetings with students at their respective home institutions. Finally, the PIs also plan to have face-to-face meetings with the entire research group at least once a year. These forms of coordinated meetings have worked well in the past, and the PIs plan to continue along the same lines.

DATA MANAGEMENT PLAN

Types of data

The following types of data will be generated during the project:

- *Source code*: We will release source code for the various algorithms, frameworks, and interactive tools developed as part of this research.
- *Curriculum materials*: We will post data from this project that is used in courses taught, including lecture slides and demo programs.

Data and Metadata Standards

The data in this project will conform to existing standards where possible, and will conform to clearly documented formats in cases where standards do not exist. In particular:

- *Source code*: Program source code will be written in standard programming languages (e.g., C/C++, Python, JavaScript).
- *Curriculum materials*: Curriculum materials will mostly consist of presentation slides (distributed as standard PDF files) and HTML web pages for project assignments and demo programs.

Policies for Access & Sharing

The data acquired in this project will be hosted on a project repository at Brown University and will be publicly available to everyone on the Internet. Intellectual Property rights in general will be managed according to the universitys intellectual property policy or by an open source software license (e.g. BSD).

Archiving and Preservation of Data

The primary mechanism for archiving and disseminating results will be publications in internationally recognized conferences and journals; increasingly, these venues also archive supplemental material containing additional results and technical details. Such papers, and when appropriate associated software and modeling results, will also be distributed in other ways via the internet. This includes the PIs research websites hosted at the University of Michigan (www.eecs.umich.edu) and Brown University (www.brown.edu), as well as externally maintained sites such as arXiv.org. We will also maintain replicas of both code and data repositories on disk backup at the project institutions.

Policies and Provisions for Re-use, Re-distribution

The PIs intend to make research results and the developed software available for research use. For tangible intellectual property, restrictions for commercial use will be determined by policies of the PIs' institutions. The PIs will honor any non-disclosure agreements that may be processed for external collaborations with industry or other universities during the course of the project.

FACILITIES

Brown University Computer Engineering Facilities and Resources

- *Equipment for Power Measurements:* We have a number of tools to aid in acquiring of the electrical measurements, including digital multimeters, programmable power supplies, and oscilloscopes.
- *IC Design Automation Tools*: The PIs have software licenses for a number of key design automation software tools including: (i) Cadence layout and schematic capture tools; (ii) Synopsys educational software bundle; (ii) Xilinx FPGA synthesis tools; (iv) various design tools from MentorGraphics and Aldec.
- *GPGPU Accelerators*: The PIs have 2 high-end Titan Xp accelerators from Nvidia as well as one Jetson TX2 and one Jetson TK1 board.
- *Computational Infrastructure*: PI Bahar's computer engineering laboratory has the following computational equipment infrastructure: (i) two Intel Dual Core based workstations; and (ii) six quad-core Intel corei7 based workstations, two of which are equipped with Nvidia Titan X GPU boards.
- *FPGA-based Accelerators*: The PIs have a Xilinx Virtex UltraScale FPGA VCU108 Evaluation Kit, a Xilinx Virtex UltraScale+ MPSoCZCU102 Evaluation Kit, and two microZed evaluation boards available in the Bahar lab as well licences for associated Vivado software.

Brown University Computer Science Facilities and Resources

The Department of Computer Science provides leading-edge computing technology to all its faculty and students. We have over 500 desktop systems running Linux, OSX, or Windows. Most of these are custom-built machines configured and assembled by the department's technical staff. A standard Linux desktop includes quad-core processors with up to 16GB of memory and dual 19" or single 24" flat-panel monitors. These systems are connected to the department's 1Gb/s switched Ethernet network with access to both Internet1 and Internet2 via the University's fiber-optic backbone. An 802.11ac (1.3Gb/s) wireless network is accessible most everywhere on campus.

The department operates three undergraduate computer clusters. The first, a banked auditorium, holds 73 systems running Debian Linux. This room serves as the primary computer facility for undergraduate computer science students. The second contains 25 seats, each with a Macintosh OSX and Debian Linux system. The layout of this space makes it an ideal room for sections, seminars, and interactive learning. The third contains 24 Linux systems and 29 docking stations for students to use with their laptops.

Five of our classroom meeting spaces are outfitted for video teleconferencing and a sixth has an advanced audiovisual system that supports recording and streaming of lectures and talks. Five research labs further enrich the environment with specialized hardware and advanced workstations from a variety of vendors.

Desktop and research systems are supported by a data center with fully redundant servers that offer a wide range of services. Central file storage is built upon IBM's General Parallel File System (GPFS). This approach provides a scalable, high performance, cost effective solution based on IBM hardware and currently hosts more than 400TB of usable RAID-6 storage. A Grid Engine cluster of 137 computational servers running Linux provides 2222 cores. The most powerful of these have 64 cores and 256GB of memory each. Five grid systems are GPU servers offering a total of 20 GPUs.

Brown University Super Computing Facilities

In addition to the equipment and computational resources available through computer engineering and computer science, the PIs have access to Browns Super computing facility, which offers heterogeneous computing platforms with different capabilities. The current total throughput of the facility is about 45

TFLOPS. The facility computational throughput is achieved through three computing clusters which can be used to conduct larg-scale simulations for the project.

1. One 200+ node cluster of the supercomputing facility is based on the IBM iDataPlex system with 270 server nodes, where each server has two quad-core Intel Xeon 5540 processors running at 2.53 GHz and fitted with 24 GB of DDR-3 DRAM memory.
2. A second cluster that has 34 server nodes, where each server has two six-core Intel 5650 processors and fitted with 48 GB of DDR-3 DRAM memory.
3. A third cluster that has 78 server nodes, where each server is fitted with two nVIDIA Tesla M2050 GPUs (each has 448 CUDA cores) and 3 GB GDDR5 memory.

The system is interconnected with a 40 Gb/s Quad-Data-Rate Infiniband switch with 324 ports. All computing nodes are diskless with I/O provided by IBM GPFS parallel filesystem with a total capacity of about 400 terabytes. The facility currently supports 430 researchers from Brown University, University of Rhode Island and the Marine Biological Laboratory in Woods Hole, Massachusetts.