Imagine fleets of networked robots, ranging from delivery drones to autonomous cars and space rovers, that generate hundreds of gigabytes of rich video and LiDAR sensory streams as they traverse uncertain terrain for even a few hours. To make sense of their physical world, they might use increasingly compute-and-power-intensive models, such as deep neural networks (DNNs), for perception and sometimes even control. But what if a low-power, compute-limited drone can’t efficiently run, or even fit in memory, the most-accurate, but compute-intensive, machine vision models with millions of parameters? Or what if an autonomous car or Mars rover can’t fully store, upload, or process terabytes of field data it measures in just a few hours [10] for continual model learning? In such cases, my research area of cloud and networked robotics provides the exciting potential for robots to significantly transcend their on-board computing limits by selectively running, and even continually refining, compute-intensive models in central servers (or “the cloud”), with extensive annotation, storage, and compute capabilities.

Future robots need to make dynamic decisions to trade-off the benefits of the cloud for both distributed inference and learning, with severe network delay, storage, annotation, or cloud computing costs. When should a drone wait for slow, but highly-accurate predictions from a cloud DNN but otherwise rely on fast, local predictions? Is a new image seen by an undersea robot (with sparse network connectivity and limited storage) worth storing for eventual cloud upload and laborious annotation to improve a model? **To answer such questions, my research contributes decision-theoretic algorithms for robots to intelligently use the cloud, but never depend on it, for better accuracy but with minimal end-to-end systems costs.** For example, my algorithms in [2] allow drones to dynamically select between local and cloud vision DNNs based on their real-time uncertainty, but also allow larger robots to act as distributed, passive collectors of “interesting” training data to prioritize for upload, annotation, and model learning [1].

**Research Philosophy:** Though cloud robotics was envisioned as early as 2010 [11], I strongly believe that today is an extremely opportune time to invest in the field, due to rapid advances in cheap, fast, power-efficient AI chips, as well as highly-accurate, compute-intensive models. My algorithms allow future robots to dynamically benefit from, and interleave, both these *largely-separate* advances, based on their real-time network conditions. Since embedded AI and cloud DNN architectures are rapidly evolving, my algorithms place a core emphasis on using modular, fully-functional robot and cloud models. Modularity allows my general abstractions to persist even if future embedded chips can run improved DNN architectures, and also allows a robot to fall back to an intact local model during network outages.

My philosophy is shaped by being an early engineer at my adviser’s startup Uhana (acquired by VMWare), where I built deep reinforcement learning (RL) controllers for cellular network traffic scheduling...
problems [4], some of which underwent successful proof-of-concept trials with network operators. This practical experience inspired me to develop broad abstractions in my PhD [1-4] that anticipate and decouple themselves from specific industry advances or platforms, which enables applications on devices ranging from smart cameras to industrial sensors. Accordingly, my research style is driven by a strong interest for academia to stay ahead, but also complement, industry by: (1) experimentally quantifying cloud benefits and costs in new robotics applications, (2) abstracting these trade-offs in general learning problems, and (3) building systems with the latest embedded AI and cloud models to stay relevant.

Research Contributions: While there has been seminal prior work [10-14] in cloud robotics, my research differentiates itself in the following dimensions:

1. Introduction of decision-theoretic learning algorithms for robot-cloud coordination: The majority of today’s cloud robotics algorithms use hand-engineered, myopic heuristics to decide when a robot should offload computation to the cloud, which often miss out on key, context-dependent benefits of cloud accuracy. My key research insight is to cast robot-cloud offloading as a multi-step learning problem for both fundamental and intuitive reasons - robots must exploit their local computation as much as possible, but also predict and explore when the cloud will help them most, which is hard to analytically model, but can be learned through training. In [2], I formalized robot-cloud offloading as a problem of sequential model selection under uncertainty, where a robot flexibly trades-off low-cost, local DNN inference with slow, but accurate, cloud DNN inference to minimize multi-step network congestion and delay. My approach, instantiated on hardware as a deep reinforcement learning (RL) offloader, led my paper at RSS 2019 to be selected as a finalist for best student paper for significantly outperforming today’s solutions.

2. Bridging the gap between robotics and computer systems research: Current robotics literature often underestimates key systems costs of robot-cloud communication. My interdisciplinary background in robotics and systems has led me to motivate these costs using rigorous experiments [1, 2, 4] before tackling them algorithmically. For distributed inference, I have quantified network congestion issues when streaming LIDAR point clouds using today’s de-facto robot autonomy stacks [2]. For distributed learning, I collected a novel video dataset of four months of road construction and self-driving cars on public roads, allowing me to quantify the costs of cloud storage, human annotation, and cloud computing in robotic vision applications [1]. Such experiments directly influence my algorithms for minimal communication.

3. General mathematical abstractions that anticipate future technology trends: Embedded AI hardware and cloud-grade DNNs are rapidly evolving, motivating careful abstractions in my general algorithms. By writing papers with two of the latest embedded AI chips [1-2] (Fig 2b), spending two years at a startup, and even co-organizing a workshop with industrial cloud robotics leaders, I have kept up with rapid industry advances. To stay ahead of them, my papers [1-3] integrate general abstractions, such as advocating fully-functional, modular robot and cloud perception models, which are not even confined to be DNNs. Modularity also allows for robustness in cases of network outages. Further, even if future embedded AI chips can scalably run “cloud-grade” models, there will still be a fundamental trade-off between model capacity, accuracy, and inference speed (Fig 1, 2b). Thus, my RL offloader [2] for interleaving low-complexity and compute-intensive DNNs will still apply if both models are co-located on the same future embedded AI chips, due to carefully-chosen abstractions.

Current Research (Learning Algorithms for Cloud Robotics + Formal Methods)
My research on cloud robotics is motivated by two coupled challenges that future robots will increasingly face: (1) the volume of rich video and LIDAR sensory streams is growing, leading robots to (2) use increasingly compute-and-power intensive models, such as DNNs, for perception and sometimes control. The interplay between my related projects is depicted in Fig. 2a, where the notation Fig.2a-x in my subsequent project descriptions refers to the module marked by number x in Fig. 2a. My projects are:
1. Distributed Inference for Real-time Perception, Fig 2a-1 (RSS 2019): As introduced before, my key research contribution was to model cloud offloading as a general problem of sequential model selection under uncertainty [2]. The insight was to develop a deep RL-based offloader that exploits a fast, low-power, local robot model but also explores the benefit of a slower, more accurate, compute-intensive model. By designing a concise, informative state space and simple action space (which specifies the models a robot can use), I built a compact RL-offloader that runs in real-time in hardware and is 18 times smaller than even mobile-optimized DNNs. Hardware experiments over real wireless links establish that our system outperforms today’s robotics benchmarks by over 2 times.

![Image of distributed inference system]

Fig 2: (Left, a) My current research projects. (Right, b) My long-term research goals anticipate advances in both embedded AI and deep learning in the cloud, allowing robots to get the best of both worlds.

2. Distributed, Continual Learning from High-Volume Robot Sensory Streams, Fig 2a-2:
Can field robots act as distributed collectors of “interesting”, rare training data used to continually improve computer vision models? My key contribution [1] was to build an “intelligent sampler” that sits on-board a robot and stores only limited, task-relevant training examples to alleviate systems bottlenecks in network transfer, dataset annotation time, and cloud computing/storage costs. I collected four months of video footage from evolving construction sites and field testing of self-driving cars in the Silicon Valley to demonstrate the benefits of continual model learning. The key insight of our sampling algorithm was to use the cloud to send the robot a few target images of interest, a nominal model trained with a few examples of such images, and confidence thresholds that help a robot identify such targets from irrelevant, “noise” images. Crucially, by adjusting these thresholds with feedback from the cloud as the robot uploads more samples, we can quickly improve vision models with minimal acquired images. The re-trained models run scalably on the state-of-the-art Google Edge Tensor Processing Unit (TPU) USB stick (Fig 2b) for low-power robots. Our system insight is to delegate compute-intensive tasks of training models and adjusting thresholds to the cloud, while keeping on-robot sampling logic simple. Specialized DNNs and annotated videos are provided to researchers at: [https://sites.google.com/view/harvestnet/](https://sites.google.com/view/harvestnet/).

3. Data-driven Network Traffic Management, Fig 2a-3 (AAAI 2018, Industry Collaboration): Future robots, or more generally Internet-of-Things (IoT) devices, might use cellular networks to upload selected field data, send sensor updates, or receive software or map updates. Such sensor updates are often high-volume, but delay-tolerant, so should ideally be sent when they do not compound network congestion for real-time, latency-sensitive applications such as interactive mobile traffic. My key research contribution [4] was to develop deep-learning-based forecasting models of real network congestion levels in Melbourne, Australia, gathered in collaboration with a network operator. Using this data, I formulated the problem of interleaving delay-tolerant IoT traffic with real-time mobile traffic as an optimal control problem with stochastic congestion dynamics. In simulations, my deep-RL based traffic scheduler outperformed model predictive control and several benchmarks in its ability to safely schedule more IoT traffic over a day by proactively scheduling traffic during forecasted drops in network congestion.
4. Formal verification of Cyber-Physical Systems: A prime vision in robotics has been to provide formal guarantees on robot safety, using diverse methods ranging from controller synthesis from signal temporal logic (STL) specifications to hybrid systems verification. However, such techniques often fail to provide safety guarantees for complex robotic systems with uncertain interactions with humans. My initial work attempts to blend formal methods with data-driven learning, by extracting key templates of human behavior from data. These templates can then be symbolically represented with formal languages, model-checked, and, most importantly, used for proactive decision-making schemes that attempt to infer a human’s true intended interaction with a robot. Formal verification remains a secondary focus of my research, having led to papers [5-8], but I plan to apply it to verify the safety of cloud offloading logic.

Future Research

Fig 3. (Left, a) Task-driven autoencoder. (Right, b) Can network-limited robots train their own models?

1. Task-driven, minimal sensor representations for robotic perception and control (Figs 2a-4, 3a)
How should we encode and stream video or LIDAR if it will never be seen by a human, but rather just by a robot or DNN in the cloud? Today's video and LIDAR representations are designed for human perception, and thus often lead to high-bandwidth transmissions over congested wireless networks for downstream machine inference tasks. In my workshop paper [3], I described a deep RL-based formulation where robots send minimal, task-relevant features for machine perception, using feedback on salient parts of the sensory stream from a DNN running in the cloud. Further, in my workshop paper [9], I showed early promise for a training scheme that co-designs how to compress sensory data with the task goal of a “consumer” DNN in the cloud. Ideally, such methods would send much less data over a wireless network but still achieve the same vision or control performance as sending all pixels in a scene. This will be a multi-year research effort that spans novel learning algorithms that trade-off minimality with interpretability and will culminate in system prototypes for tele-operation and control.

2. The “Almost” Self-Learning Robot: (Semi) Federated-Learning for Robots (Figs 2a-5, 3b)
With advances in embedded AI, we can now train models directly on-device, instead of the cloud, allowing robots to directly train specialized models for their unique operating conditions with their own private data. This promise was first envisioned by federated learning, an algorithm for multiple mobile devices to train models with private data, but send gradients to the cloud for collective learning. A key assumption inherent to federated learning, that breaks down in robotics, is that data-labels are present on-device in order to train a local model. In practice, real-world robots might need to occasionally ask a human to annotate images and extrapolate annotations for large amounts of private data stored on-robot. I envision robots that crawl Mars, seas, or even subterranean mines that occasionally “surface” to send 50-100 images per week to human supervisors, and automatically specialize models themselves without leaking private data. Concretely, I will extend ideas on weak-supervision and formulate algorithms that take “safe” gradients for model learning by intelligently weighing samples with explicit and extrapolated human labels. I have already successfully experimented with on-device training on the Google Edge TPU.
3. Robotic Control Across Private Data Boundaries (Fig 4a)
My research limits communication to minimize network congestion. However, an exciting area is to abstract the notion of a bandwidth-limited network link to a “private data boundary” which limits the scope and volume of data two companies can share for cooperative control. In my startup experience, I witnessed firsthand how cellular operators have a wealth of city-scale connectivity data, including private user mobility patterns, that can be used to deliver fine-grained congestion forecasts to external companies, such as ride-sharing services, to proactively route future robotic taxi fleets or control bitrates for mobile video streaming. I envision algorithms that allow a network operator and an external, proprietary controller to trade succinct forecasts and control feedback that preserve user privacy with high-level data exchanges. Strikingly, this setup has some structural similarities to cloud robotics, since two machine-learning models (in this case a forecaster and controller) must learn how to trade succinct data and control feedback across a physical data boundary, albeit with an added notion of privacy.

4. The Neural Signal Processing Toolchain: Towards a theory for composing modular, cascaded neural networks from robotic sensors to the cloud (Fig 4b)
Today, we have a plethora of pre-trained DNNs ranging from fast, compute-efficient models to larger, compute-intensive counterparts, often from industry. While a large body of recent research attempts to carefully examine the internal layers of such DNNs to understand their inner workings, I propose we should also view them as “black-boxes” with standard interfaces, such as a map from fixed size images to a set of bounding boxes. Then, even if the architectures of models change, we can still design general pipelines with DNNs that encode video at a robot, decode it in the cloud, and learn how to best deliver data from a sensor to a cloud DNN “consumer” in a standard interface. I believe academia should develop a theory for how to compose modular neural networks with various latencies and size profiles that filter, offload, encode, decode, and finally consume sensor data, in order to keep pace with rapid changes and software updates to publicly available DNNs from industry.

5. Beyond self-driving cars in academia: Cloud Robotics for Manufacturing, Agriculture, etc.
I have talked to manufacturing robotics companies who are facing network connectivity issues with multiple low-cost delivery robots and industrial sensors sending large data-streams on crowded factory floors. Further, my algorithm in [2] can apply to shared autonomy on the factory floor, where say 20 independent robots assemble parts using local models, but learn when to query a human supervisor with abstractly “limited communication bandwidth” only when they are highly uncertain. Inspired by Microsoft’s FarmBeats project [15], which improves farm efficiency by data-driven insights from soil sensors and aerial drone maps processed in a central server on a farmer’s barn, I also envision algorithms for bandwidth-efficient communication between agricultural drone sensors and an ad-hoc barn “cloud”.

**My research vision for future PhD students:**
Today is an opportune time to research cloud robotics, given rapid advances in cheap, efficient embedded AI, next-generation cellular networks, and increasingly compute-intensive deep learning. I hope to inspire future graduate students to leverage such advances to build systems and general learning algorithms to address the “robot-cloud gap” (Fig. 2b), while collecting novel sensor datasets for domains that uniquely interest them. This will allow academia to stay ahead of industrial advances, while allowing agricultural, manufacturing, and space rovers to significantly transcend their local compute limits.
**My Publications (Selected):**


**General References:**


