Distributed Perception and Learning Between Robots and the Cloud

Sandeep Chinchali
Stanford University
Robot sensory data + compute models are becoming increasingly complex

Fleets of *networked* robots

Growing volumes of rich video and LIDAR sensory streams

Increasingly compute-intensive models
How Can **Network Connectivity** Help Robots?

**Robot Model**

1. Distributed Inference
2. Distributed Learning

**Cloud model**

1. Better Inference Results (Real-time)
2. Better, *specialized* models (Offline)

**Challenge: Conserve System Resources**
Key Challenges of Cloud Robotics

1. Distributed Inference
2. Distributed learning

Challenge: Limit Communication

- Network Latency
- Annotation, Storage, Cloud Computing

~4 TB per car/ day [Intel]
1. **Distributed Inference: The Robot-Cloud Offloading Problem**

Can robots *learn* when to query the cloud for **higher accuracy** but **minimal communication cost**?
2. **Distributed Learning**: The Robot Sensory Sampling Problem

Can robots *intelligently sample* minimal training examples for continual learning in the cloud?
Outline

1. Motivation:
   A. The Robot-Cloud Accuracy Gap (of Deep Neural Nets)
   B. Network Costs of Streaming Sensory Data

2. Distributed Inference: The Robot-Cloud Offloading Problem

3. Distributed Learning: The Robot Sensory Sampling Problem

4. Future Directions in Networked Control
The Robot – Cloud Accuracy Gap

By what margin does the cloud improve perception accuracy?
Advances in Embedded AI

Edge Tensor Processing Unit (TPU) (March 2019, $75)

Quad-Core ARM CPU
Accuracy of Robot and Cloud DNNs

<table>
<thead>
<tr>
<th>Platform</th>
<th>Cost</th>
<th>Accuracy (mAP)</th>
<th>Inf. Time (ms)</th>
<th>DNN</th>
<th>Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge TPU</td>
<td>$75</td>
<td>0.169</td>
<td>6.89</td>
<td>MobileNet 2</td>
<td>5.5</td>
</tr>
<tr>
<td>ARM CPU</td>
<td>$57</td>
<td>0.170</td>
<td>285.56</td>
<td>MobileNet 2</td>
<td>4.7</td>
</tr>
<tr>
<td>Cloud GPU</td>
<td>$879</td>
<td>0.265</td>
<td>56.84</td>
<td>MobileNet 2</td>
<td>20</td>
</tr>
<tr>
<td>Cloud GPU</td>
<td>$879</td>
<td>0.426</td>
<td>447.5</td>
<td>Faster R-CNN</td>
<td>183</td>
</tr>
</tbody>
</table>
Cloud DNNs localize far-off objects

MobileNet 2 (Fast, power-efficient)  Faster R-CNN (slow, accurate)

MN2 takes 9 more seconds than Faster R-CNN to localize the far-off tractor. Ample time to query the cloud.
Outline

1. Motivation:
   A. The Robot-Cloud Accuracy Gap (of Deep Neural Nets)
   B. Network Costs of Streaming Sensory Data

2. Distributed Inference: The Robot-Cloud Offloading Problem

3. Distributed Learning: The Robot Sensory Sampling Problem

4. Future Directions in Networked Control
Hidden Costs of Network Congestion

Can we experimentally quantify the systems cost of streaming video and LIDAR?

Succinct Feedback

Minimal Data

Neural Net

Database
Network Costs of Cloud Communication

1. Congested Wireless Links

John Emmons *et. al.* “Cracking Open the DNN blackbox”

Chinchali *et. al.* “Neural Networks Meet Physical Networks” (ACM HotNets ’18)
Our Network Congestion Experiments

“ROS Ate My Network Bandwidth!”
(Robot Operating System [ROS] User Forums)

~70 Mbps
Outline

1. Motivation:
   A. The Robot-Cloud Accuracy Gap (of Deep Neural Nets)
   B. Network Costs of Streaming Sensory Data

2. **Distributed Inference**: The Robot-Cloud Offloading Problem

3. **Distributed Learning**: The Robot Sensory Sampling Problem

4. Future Directions in Networked Control
1. Distributed Inference: The Robot-Cloud Offloading Problem

Can robots learn to best interleave on-board and cloud computation?

Diagram showing succinct feedback and minimal data flows.
Cloud Offloading: A Dynamic Decision-Making Problem

Query the cloud for better accuracy? Latency vs. Accuracy vs. Power ...

Offload Logic

Offload

Congested, Stochastic

Cloud Model

Local Compute
Robot-Cloud Offloading: Sequential Model Selection

\textbf{Video Frames Over Time}

- Use Robot Model
- Use Past Prediction
- Use Cloud

\textbf{State}: sensory input, remaining query budget, age of past prediction, ...

\textbf{Action}: which model to choose?

- use past prediction: cheap, potentially stale prediction
- use robot model: more expensive, potentially low accuracy
- query cloud model: most expensive, highest accuracy, limited budget

\textbf{Key Constraint}: Limited Queries to the Cloud

\textbf{Reward}: balance \textit{prediction accuracy} with \textit{cost of querying model}
Robot-Cloud Offloading: A Learning Problem

Video Frames Over Time

State: Hard to model dynamics of Network and Video Evolution \(\rightarrow\) Model-Free RL

Action: which model to choose?
- use past prediction: cheap, potentially stale prediction
- use robot model: more expensive, potentially low accuracy
- query cloud model: most expensive, highest accuracy, limited budget

Constraint: Limited Queries to the Cloud

Reward: balance prediction accuracy with cost of querying model
A Learning-Based Approach to Cloud Offloading

Succinct Feedback

Minimal Data

Neural Net
Reinforcement Learning (RL)

**Goal:** Maximize the total reward \( \sum_t r_t \)

**Agent** \[ \to \] **Environment**

- **Action** \( a^t \)
- **Reward** \( r^t \)
- **Measure state** \( s^t \)

**Exploration vs. Exploitation Tradeoff**

- **Exploit:** On-board Robot Model
- **Explore:** Utility of Cloud by *learning*

**Hard to Model Dynamics**

- What will a robot see next?
- Wireless network evolution
Robot

Limited Network

Cloud

Past Predictions

State $s^t$

$\pi_{\text{offload}}$

$a^t = 2$

$x^t$

$a^t = \{0, 1\}$

$a^t = 3$

Cloud Model

$y^t_{\text{robot}}$

$y^t_{\text{Cloud}}$

Offload

Reward
The Robot Offloading MDP

\[ \mathcal{M}_{\text{offload}} = (S_{\text{offload}}, A_{\text{offload}}, R_{\text{offload}}, P_{\text{offload}}, T) \]

\[ \hat{y}_{\text{robot}}^{t}, \text{conf}_{\text{robot}}^{t} = f_{\text{robot}}(x^{t}) \]

\[ \hat{y}_{\text{cloud}}^{t}, \text{conf}_{\text{cloud}}^{t} = f_{\text{cloud}}(x^{t}) \]

Modular robot and cloud models

Widely Applicable Beyond Neural Nets

Only assume predictions + confidences
The Robot Offloading MDP: Action Space

\[
a^t_{\text{offload}} = \begin{cases} 
0, & \text{use past robot prediction } \hat{y}^t = f_{\text{robot}}(x^{\tau_{\text{robot}}}) \\
1, & \text{use past cloud prediction } \hat{y}^t = f_{\text{cloud}}(x^{\tau_{\text{cloud}}}) \\
2, & \text{use current robot prediction } \hat{y}^t = f_{\text{robot}}(x^t) \\
3, & \text{use current cloud prediction } \hat{y}^t = f_{\text{cloud}}(x^t) 
\end{cases}
\]

Past predictions, local model, or cloud

4 succinct actions: easy to run on hardware

No network? Use local model (safety)
The Robot Offloading MDP: State Space

\[ s^t_{\text{offload}} = \left[ \phi(x^t), f_{\text{robot}}(x^{T_{\text{robot}}}), f_{\text{cloud}}(x^{T_{\text{cloud}}}), \Delta t_{\text{robot}}, \Delta t_{\text{cloud}}, \Delta N_{\text{budget}}, T - t \right]. \]

1. Has the visual input \textbf{changed} much?
2. What do \textbf{stored}, past predictions say?
3. How \textbf{old} are past predictions?
4. What \textbf{cloud budget + time} remain?
The Robot Offloading MDP: Reward

\[ R_{\text{offload}}^t(s^t, a^t) = -\alpha_{\text{accuracy}} \]

\[ \mathcal{L}(y^t, \hat{y}^t) - \beta_{\text{cost}} \text{cost}(a^t) \]

\[ \text{Balance accuracy and systems cost} \]

\[ \text{Rigorously define costs from experiments} \]

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy (mAP)</th>
<th>Size</th>
<th>Inf. Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>MobileNet v1</td>
<td>18</td>
<td>18 MB</td>
<td>26 ms</td>
</tr>
<tr>
<td>MobileNet v2</td>
<td>22</td>
<td>67 MB</td>
<td>31 ms</td>
</tr>
<tr>
<td>Mask R-CNN</td>
<td>45.2</td>
<td>1.6GB</td>
<td>325 ms</td>
</tr>
</tbody>
</table>
1. **Distributed Inference**: The Robot-Cloud Offloading Problem

**Hardware Experiments**

- **Succinct Feedback**
- **Minimal Data**

**Neural Net**
Coherence Time

$t = 0$

Robot Model

FaceNet

SVM Classifier

$x^t$

Embed

Face A

90% Conf

$t = T$

Query Cloud
Deep RL beats benchmark offloading policies

Reward: Balance **Accuracy** and **Model Cost**
- RL: 2.3x Benchmark Reward

*Interleave* On-Robot and Server Compute

---

### Graph

- **Oracle**
- **RL**
- **Heuristic**
- **All-Robot**
- **All-Cloud**

![Graph showing comparison between Oracle, RL, Heuristic, All-Robot, and All-Cloud on mean accuracy and mean offloading cost.](image)

- Cloud, ID-0: 70.52%
- Cloud, ID-4: 34.5%
- Robot, ID-0: 85.70%
Hardware Experiments on Live Video + Embedded Compute Platform

Compact RL Offloader!

Chinchali et. al. “Network Offloading Policies for Cloud Robotics: A Learning—Based Approach” (RSS 2019, Finalist for Best Student Paper)
2. Distributed Learning: The Robot Sensory Sampling Problem

Can robots sample task-relevant training examples from large sensory streams?
Can we make actionable insights from growing robotic sensory data?

Learning Rare Events Can Improve Decision-Making Models

Robotic Fleets: *Distributed, passive* collectors of training data

Intelligent Sampling reduces systems bottlenecks
Outline

1. Motivation
2. Distributed Inference: The Robot-Cloud Offloading Problem
3. Distributed Learning: The Robot Sensory Sampling Problem
   A. Benefits of Cloud Re-training
   B. Systems Costs of Acquiring Training Images
   C. Sampling Approach
   D. Results
4. Future Directions in Networked Control
2. **Distributed Learning: The Robot Sensory Sampling Problem**

A. Why specialize vision models in the cloud?

- **Succinct Feedback**
- **Minimal Data**
Rationale 1: Specialization corrects errors

Default Vision Model
Pre-trained on COCO dataset

Error

Our Re-trained Model
Using minimal images

Corrected
Model specialization can correct key errors

Default Vision Model
Pre-trained on COCO dataset

Our Re-trained Model
Using minimal images
Rationale 2: The real world is *constantly changing*

3 months of *my dashcam* footage

Can robots update HD maps with roadblocks, potholes?
2. **Distributed Learning: The Robot Sensory Sampling Problem**

**B. What is the cost of distributed learning?**
Why sample?: Reduce **systems costs**

- ~13 GB of video clips (small)
- 340k images to select from

**Google Cloud Labeling Service:**
- $49 for 1000 bounding boxes
- $850 for 1000 masks

**Our experiments:** We labeled 1000 boxes in 1.4 hours (~200 images)

Worse for even 1% of autonomous car data

**Our Sampler:** $400 for 8k detailed annotations
2. Distributed Learning: The Robot Sensory Sampling Problem

C. But how do I automatically sample?: Technical Approach
Insight 1: **Minimal Images are Needed**

**Static Targets**: Law of Diminishing Returns

**Dynamic Targets**: Continual Learning Needed

Robots update HD maps to reflect real-world changes
Insight 2: **Efficiently** filter images of interest during inference

Transfer Learning: Adapt pre-trained CNN to new classes

**Insight:** Use CNN confidence scores/embeddings to filter targets (**red**) from noise (**blue**)

Embeddings/scores come “for free” during inference
Vision Model $i$

Target Images $i$

Sampler

Waymo > $thresh^i$?

Sample Cache $i$

Upload

Learning Round $i$

Send Model, Target Images, Thresholds $i$

Annotated Dataset $i$
Insight 3: Delegate compute-intensive tasks to the cloud

Use cloud feedback to adjust confidence thresholds from growing dataset
Annotate
Cache
Stored
Dataset
Robot
Vision Model
Limited Network
Cloud

Vision Model
Sampler
“Interesting”
Image?
Sample
Cache

Update
Model
Stored
Dataset
Re-train
Model
Annotate
Cache
Upload

“Interesting” Image?
2. The Robot Sensory Sampling Problem

D. Distributed Learning Results

Succinct Feedback

Minimal Data

Neural Net
Continual Learning on Road-Scene Domains

Sample only 18-34% of all possible images, reduces systems costs
Adapting Vision Models with Minimal Images

Default Vision Model
Pre-trained on COCO dataset

Our Re-trained Model
Using minimal images

Domain-Specific
LIDAR 94.9
waymo 92.8

truck 99.9
car 99.0
umbrella 75
truck 66.2
Future Directions

Task-Driven Representations for Control
Future Technology Trends

Edge TPU USB (March 2019, $75)
Nvidia Jetson TX2 (March 2017, $450)

Cloud AI:
- Constantly updated maps / databases
- Human annotation
- Model re-training
- Multi-robot view

Edge AI: Faster, Cheaper, and Smaller

Hardware gap may be transient, but the utility of the cloud and distributed learning will persist
Future 1. **Task-Driven** Representations for Perception

- More concise representations than raw images
- Co-design sensory representation with inference/control task

Efficient Embedded AI

Minimal Representation

Larger, Accurate DNNs
Early Results: **Task-Driven** Representations

**Co-design** representation with final task

- Encoder: $q(z^t | x^t; \theta_e)$
- Decoder: $p(x^t | z^t; \theta_d)$
- Pre-trained Model: $y^t = f(x^t; \theta_p)$

Reduce information by $8x$ for same classification accuracy (MNIST)
Future 2. (Semi) Federated Learning for Robots

Why not specialize models (re-train) on the robot (“edge”)?
Most training data resides on robot, extrapolate from few human labels

Robot-specific model + data

Ideally, no private data
Future 3. Control and Learning Across Data Boundaries

Network operators have a wealth of data, some private. Can agents trade concise representations for cooperative control?

Limit the scope and volume of data traded across boundary.

Re-think the separation principle for “data-driven” control.
Robot Network

Sensory Input

(1) Offloader
(2) Sampler
(3) Task-Encoder
(5) Private Training

Network

New Models + Feedback

(4) Traffic Scheduling

Cloud

Inference + Learning

Multi-Device Coordination

Task-Driven Representation

Thanks! Questions: csandeep@stanford.edu
IoT Traffic Scheduling (AAAI 2018)

Melbourne Central Business District, Rolling Average = 1 min

Diverse city-wide cell patterns
Extra Slides
<table>
<thead>
<tr>
<th>Device</th>
<th>Model</th>
<th>Acc. (mAP)</th>
<th>Infer. Time (ms)</th>
<th>Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPU</td>
<td>MN 2</td>
<td>0.169</td>
<td>6.9</td>
<td>5.5</td>
</tr>
<tr>
<td>CPU</td>
<td>MN 2</td>
<td>0.170</td>
<td>285.6</td>
<td>4.7</td>
</tr>
<tr>
<td>Cloud GPU</td>
<td>MN 2</td>
<td>0.265</td>
<td>56.8</td>
<td>20</td>
</tr>
<tr>
<td>Cloud GPU</td>
<td>Faster R-CNN</td>
<td>0.326</td>
<td>447.5</td>
<td>183</td>
</tr>
</tbody>
</table>
Motivating Technology Trends

1. **Rich Sensory Data**: High-bitrate + volume

2. **Compute-intensive Models**: Perception + Control, some deep nets

3. **Better wireless networks**: Low-latency access to compute (even cell towers!) *But* more mobile traffic!

4. **Robot – Cloud Computing Gap**: Edge AI: Fast, low-power, less-accurate* 
   Cloud AI: Slow, more-accurate* *but “global” view*
RL beats benchmark offloading policies

- > 2.6x reward of benchmarks
- RL: 70% of oracle reward
- All-Robot: today’s de-facto
RL intelligently, but **sparingly** queries cloud
Cloud Offloading as a Decision Problem

Contending goals
- Maximize Accuracy
- Minimize latency
- Limited Network Share

Optimal Control
Continual Learning on FaceNet Domains
Labeling Team: How do we proceed when there is no good visibility (highlighted in the image)

- Few cases the edges of the car are not visible
- Few cases the lidar is not clearly visible.
<table>
<thead>
<tr>
<th>Device</th>
<th>Model</th>
<th>Acc. (mAP)</th>
<th>Infer. Time (ms)</th>
<th>Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPU</td>
<td>MN 2</td>
<td>0.169</td>
<td>6.9</td>
<td>5.5</td>
</tr>
<tr>
<td>CPU</td>
<td>MN 2</td>
<td>0.170</td>
<td>285.6</td>
<td>4.7</td>
</tr>
<tr>
<td>Cloud GPU</td>
<td>MN 2</td>
<td>0.265</td>
<td>56.8</td>
<td>20</td>
</tr>
<tr>
<td>Cloud GPU</td>
<td>Faster R-CNN</td>
<td>0.326</td>
<td>447.5</td>
<td>183</td>
</tr>
</tbody>
</table>
(1) New Image

(2) Vision Model $i$

(3) Target Images, Thresholds $i$

(4) Sampler

(5) Sample Cache $i$

Store? Update Model $i+1$

(6) Annotate Cache

(7) Re-train Model

(8) Adapt Thresholds, Target Images

New Targets, Thresholds $i+1$
Growing Volumes of Robotic Sensory Data

• Learning Rare Events Can Improve Decision-Making Models

• Robotic Fleets: Distributed, passive collectors of training data

• Intelligent Sampling reduces systems bottlenecks
Limited Network Robot

Vision Model

Sampler

“Interesting” Image?

Sample Cache

Update Model

Stored Dataset

Cloud

Annotate Cache

Re-train Model

Upload

“Interesting” Image?