Uncertainty from Motion for DNN Monocular Depth Estimation

Soumya Sudhakar, Vivienne Sze, Sertac Karaman

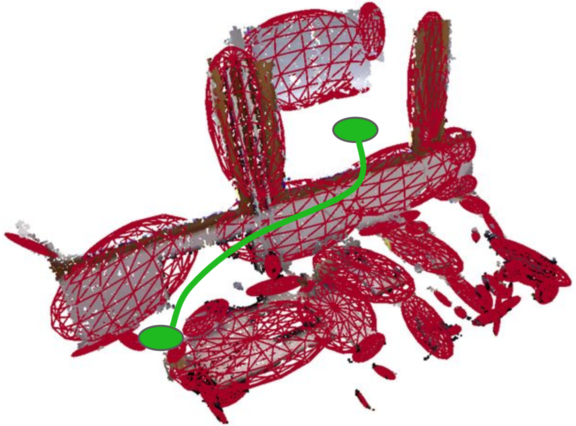
Low Energy Autonomy and Navigation (LEAN) Group

CICS - November 1, 2023

Task: Depth Estimation for Navigation

Robot

GMMap [**Li et al.**, 2023]

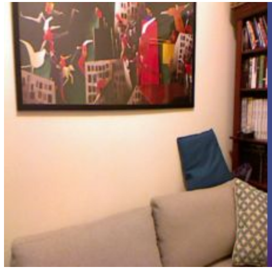


Task: Depth Estimation for Navigation

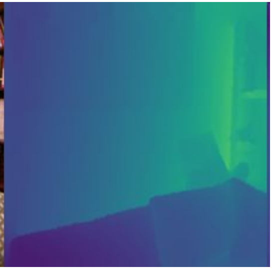
GMMap [**Li et al.**, 2023]

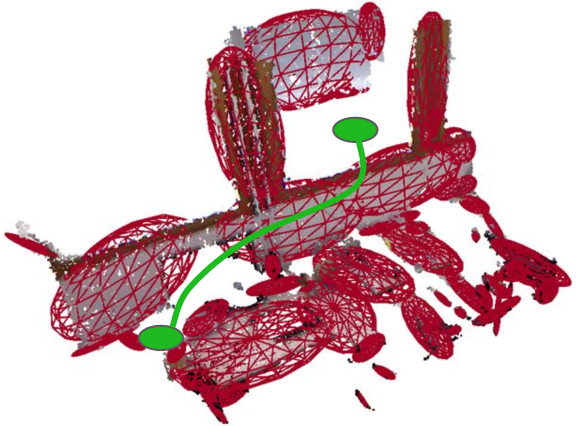
Robot

RGB



Depth





Autonomous navigation requires depth estimation to sense where obstacles and free space are in the world

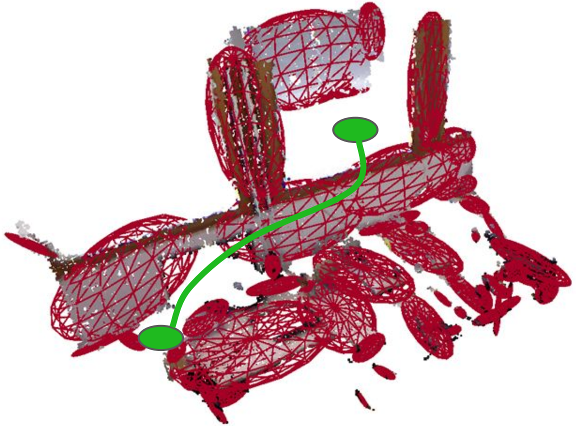
Task: Depth Estimation for Navigation

Robot

RGB

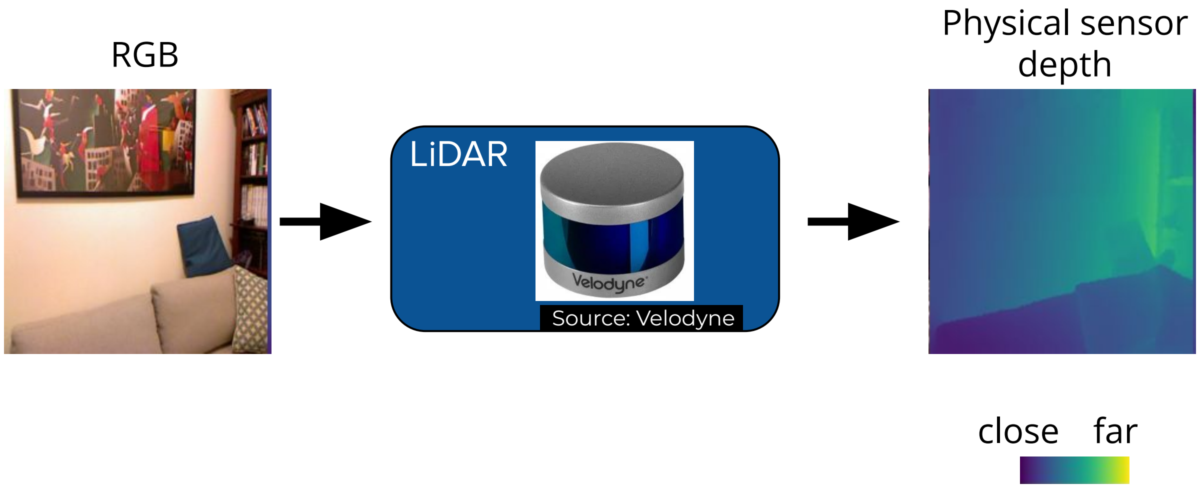
Depth

GMMap [**Li et al.**, 2023]

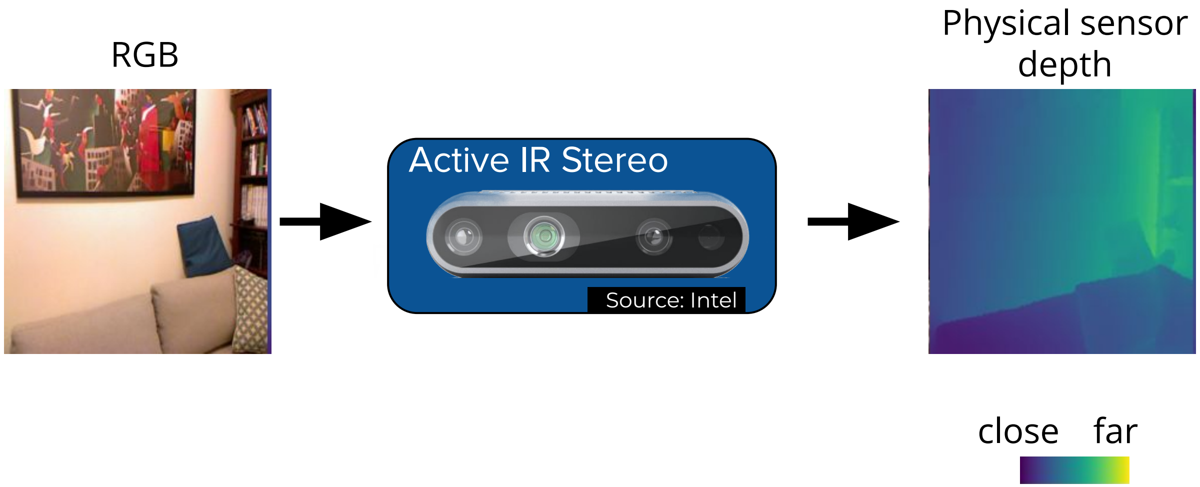


Autonomous navigation requires depth estimation to sense where obstacles and free space are in the world

Motivation: Traditional Depth Sensors are Power-Hungry, Bulky



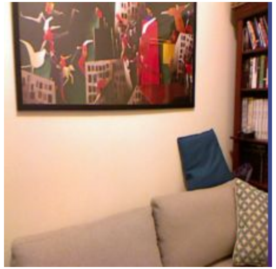
Motivation: Traditional Depth Sensors are Power-Hungry, Bulky



Motivation: Traditional Depth Sensors are Power-Hungry, Bulky

Challenges with traditional depth sensors:
power-hungry and bulky

RGB



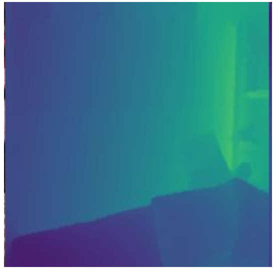
LiDAR

Source: Velodyne

Active IR Stereo

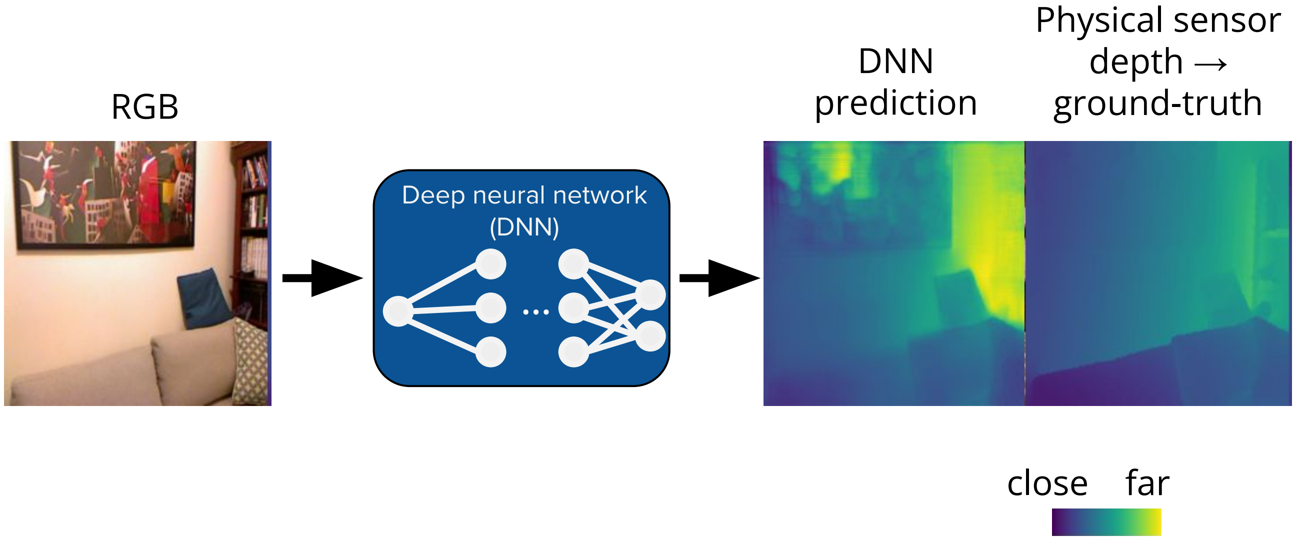
Source: Intel

Physical sensor depth



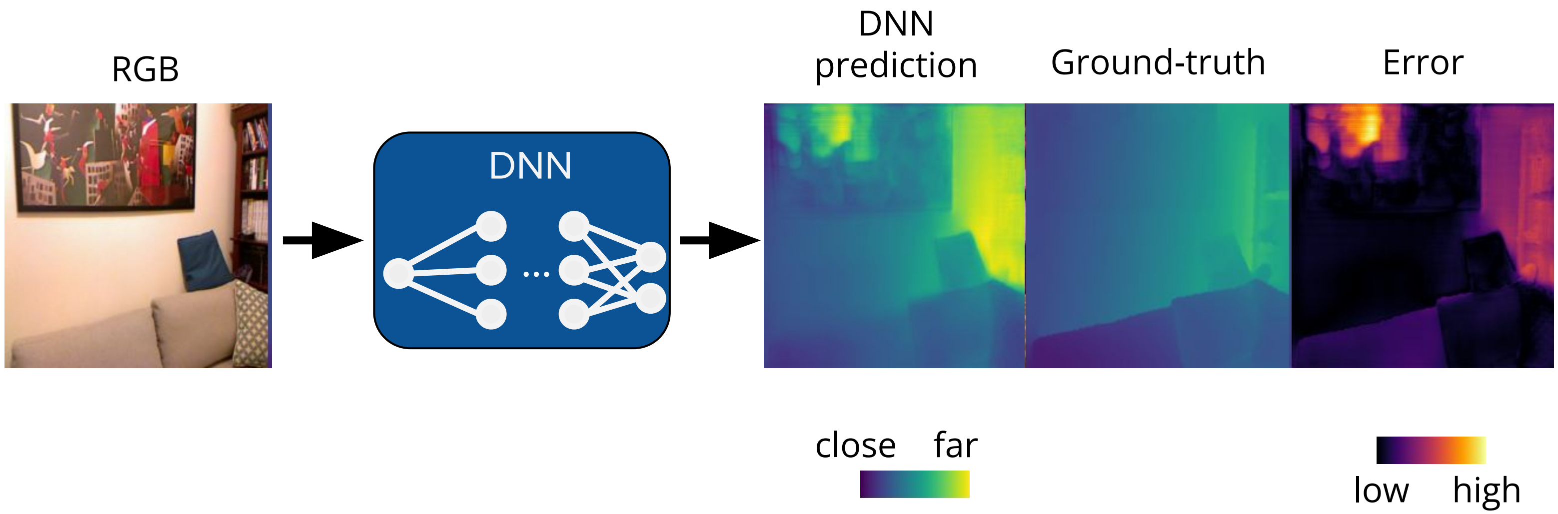
close far

Task: DNN Monocular Depth Estimation



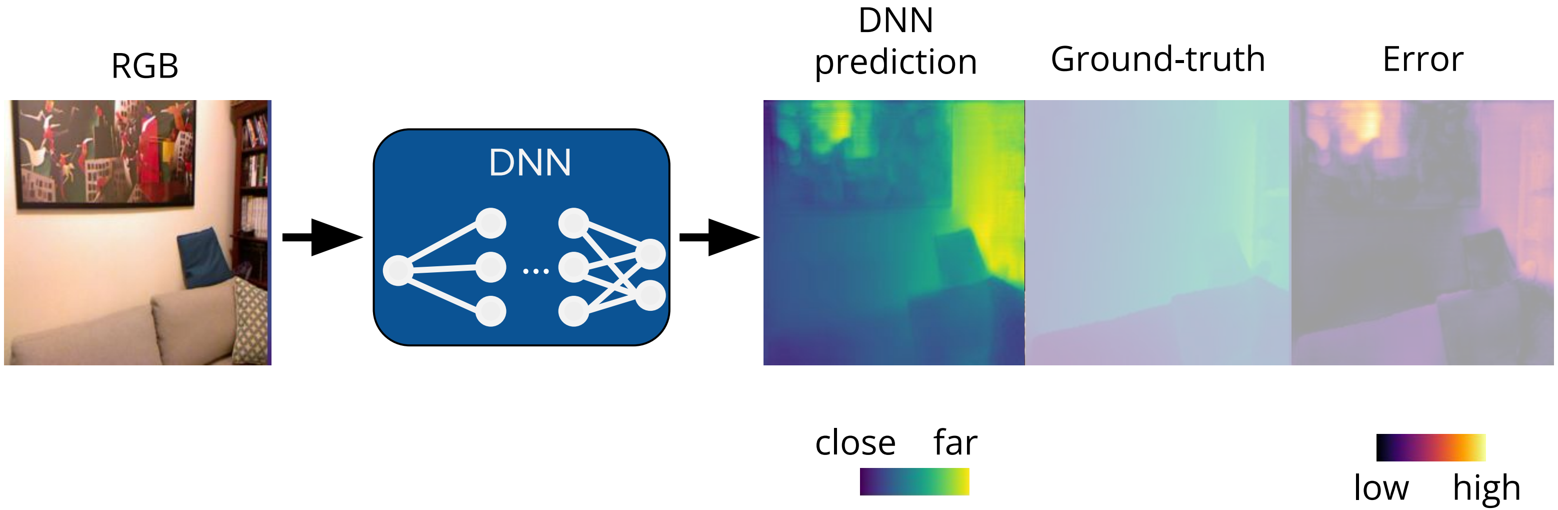
We use a DNN trained to predict per-pixel depth from a single RGB image to reduce energy and form factor of depth sensor

Task: DNN Monocular Depth Estimation



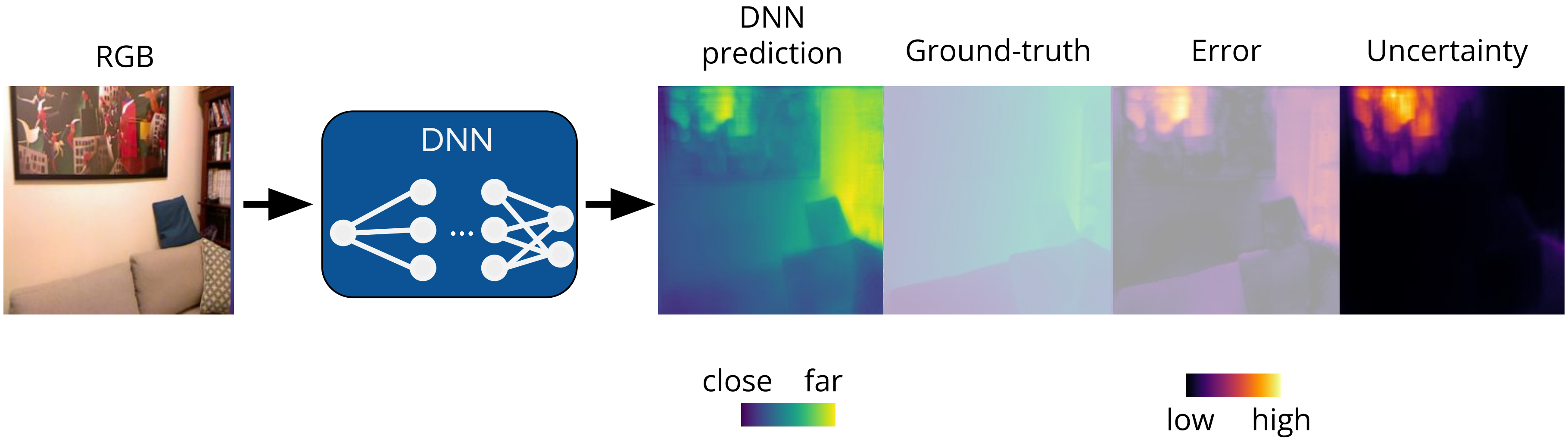
During training, access to ground-truth to see where DNN prediction has high error

Task: DNN Monocular Depth Estimation



During deployment, no access to ground-truth or error of DNN predictions

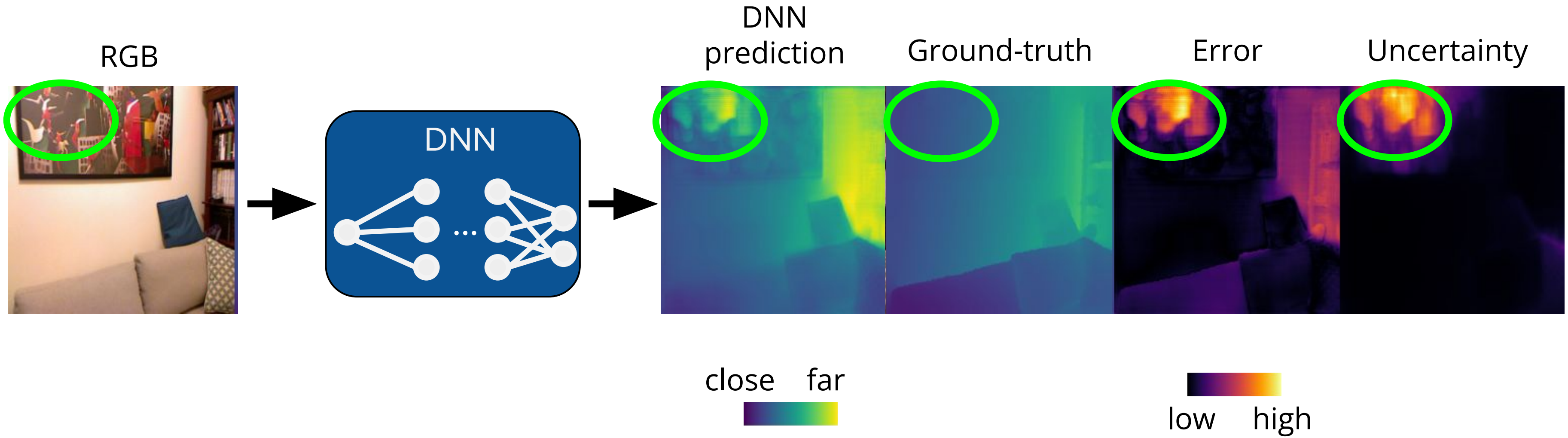
Task: Uncertainty Estimation for DNN Monocular Depth



Goal: DNNs that fail gracefully and estimate high-quality uncertainty on predictions

Task: High-Quality Uncertainty Estimation for DNN Monocular Depth

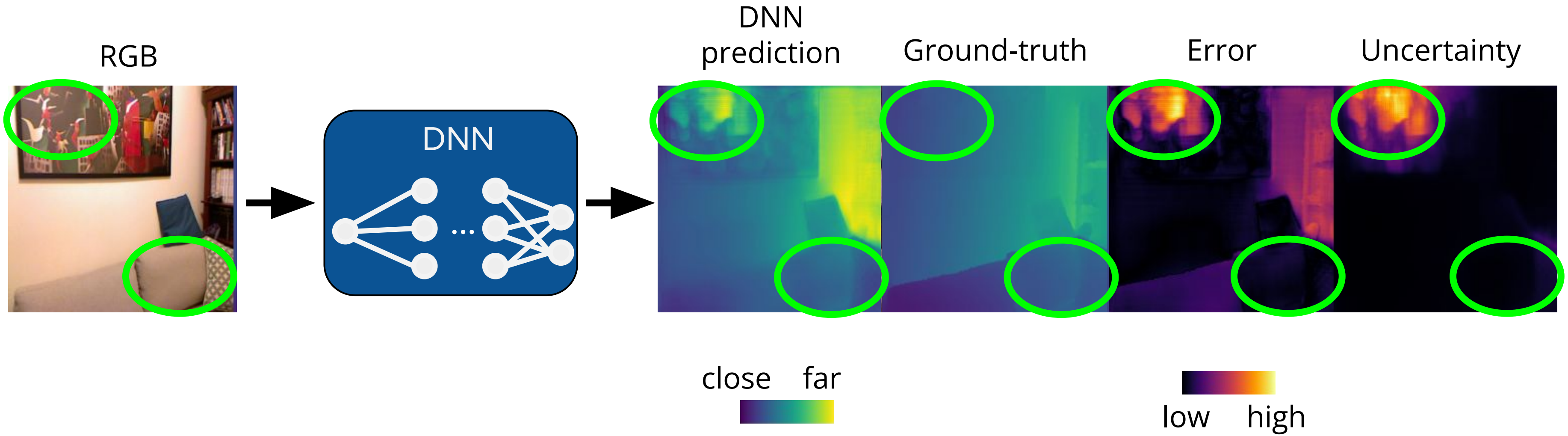
✔ High error ➜ high uncertainty



Task: High-Quality Uncertainty Estimation for DNN Monocular Depth

✔ High error ➔ high uncertainty

✔ Low error ➔ low uncertainty

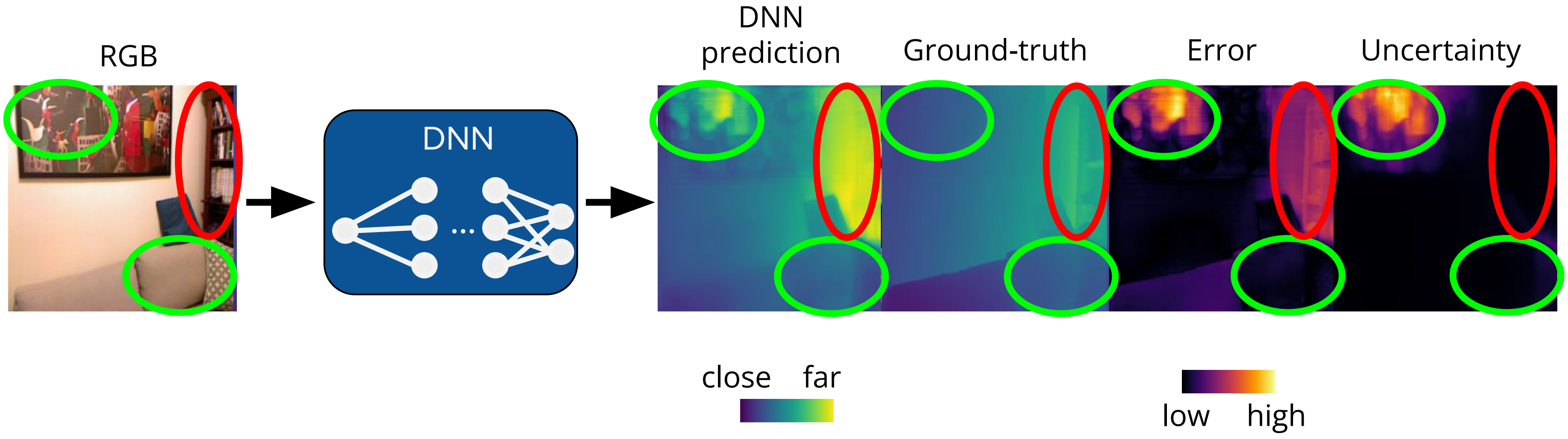


Task: High-Quality Uncertainty Estimation for DNN Monocular Depth

✔ High error ➔ high uncertainty

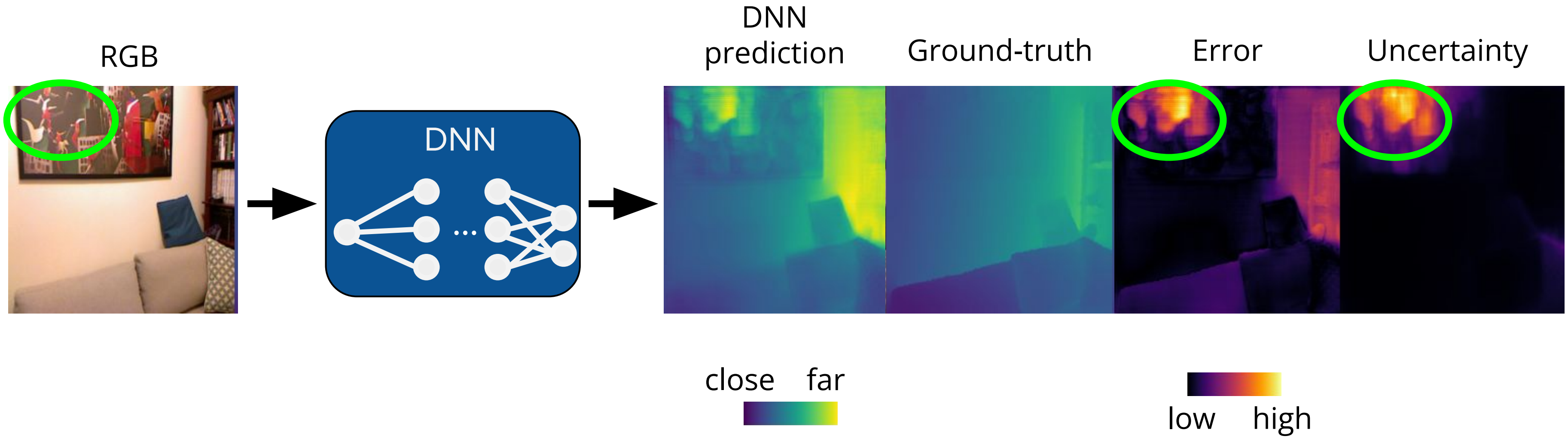
✔ Low error ➔ low uncertainty

❌ High error ➔ low uncertainty



Task: High-Quality Uncertainty Estimation for DNN Monocular Depth

✔ High error ➜ high uncertainty
✔ Low error ➜ low uncertainty
✘ High error ➜ low uncertainty
✘ Low error ➜ high uncertainty



High quality uncertainty estimation correlates uncertainty to error

How Do We Estimate DNN Uncertainty?

- We can ask the DNN to predict its own (aleatoric) uncertainty via a learned loss function

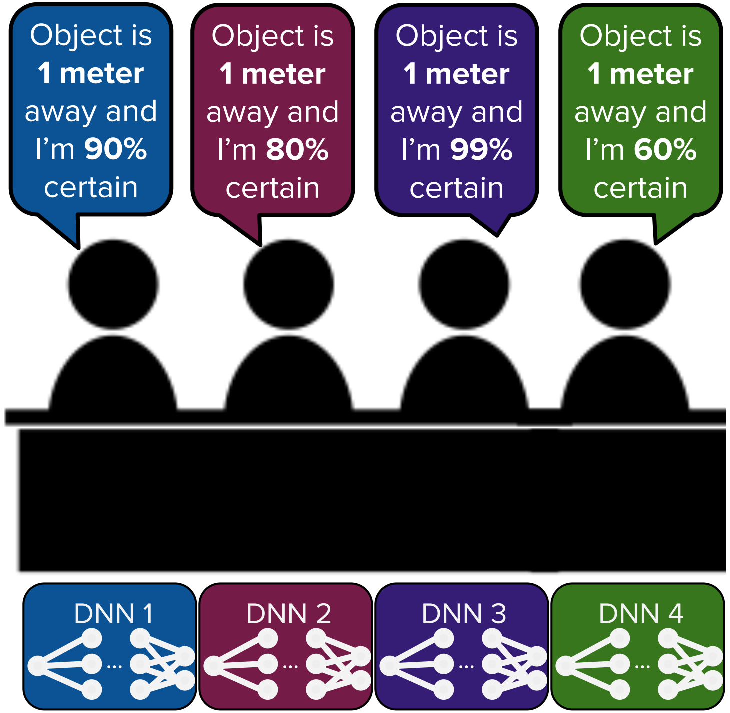
How Do We Estimate DNN Uncertainty?

- We can ask the DNN to predict its own (aleatoric) uncertainty via a learned loss function
- **Advantages:**
  - Captures uncertainty in the data that was captured during training (e.g., DNN is prone to error when lighting is poor)
  - Computationally efficient

How Do We Estimate DNN Uncertainty?

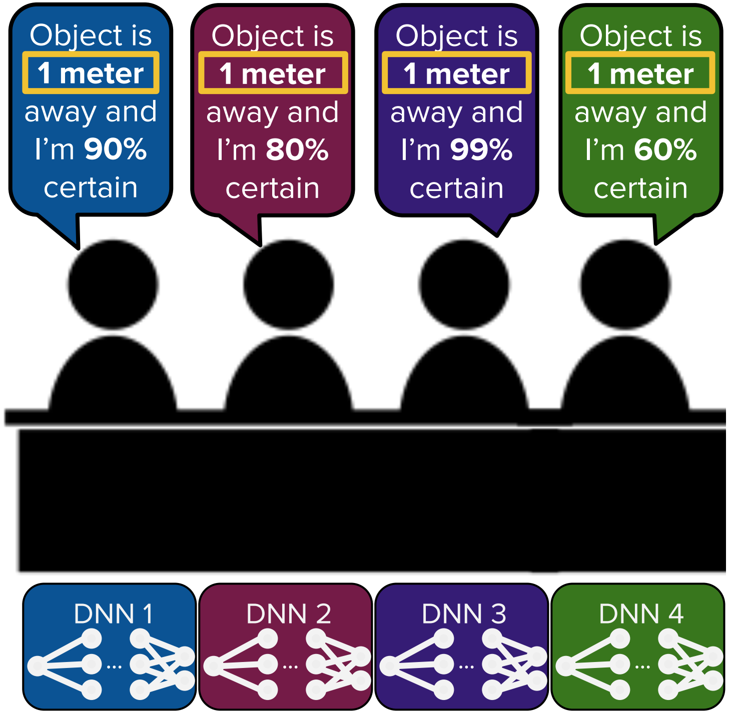
- We can ask the DNN to predict its own (aleatoric) uncertainty via a learned loss function
- **Advantages:**
  - Captures uncertainty in the data that was captured during training (e.g., DNN is prone to error when lighting is poor)
  - Computationally efficient
- **Disadvantages:**
  - Does not capture (epistemic) uncertainty inherent to the DNN model weights itself where the DNN does not know what it hasn't trained on before

Let’s Convene a Panel of Diverse Experts (DNN Ensemble)



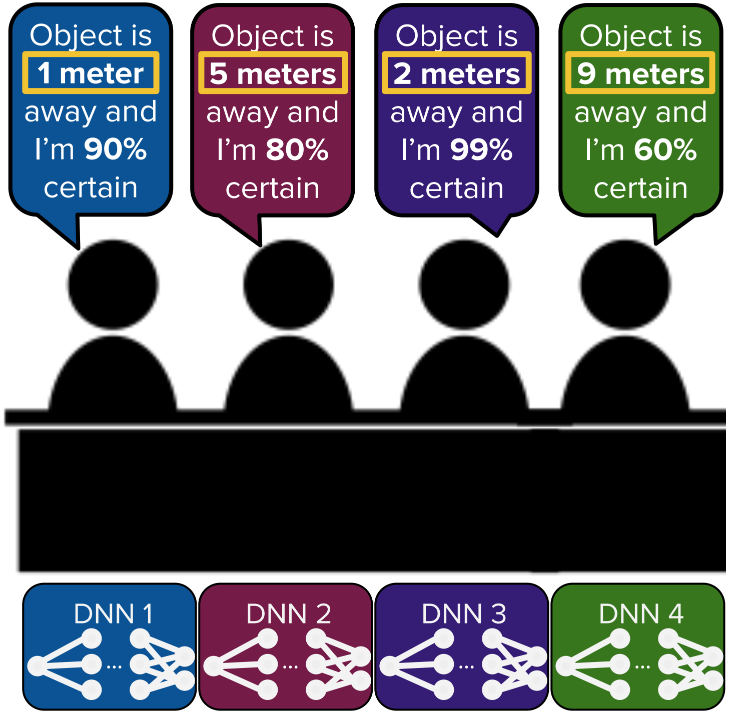
Model uncertainty is low when a diverse set of experts (DNNs) **agree** in their predictions

Let’s Convene a Panel of Diverse Experts (DNN Ensemble)



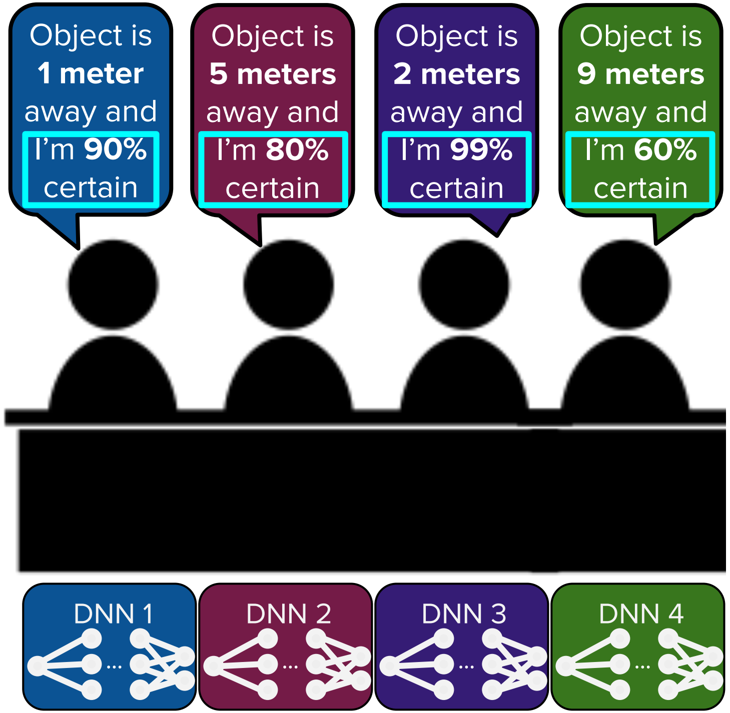
Model uncertainty is low when a diverse set of experts (DNNs) **agree** in their predictions

Let's Convene a Panel of Diverse Experts (DNN Ensemble)



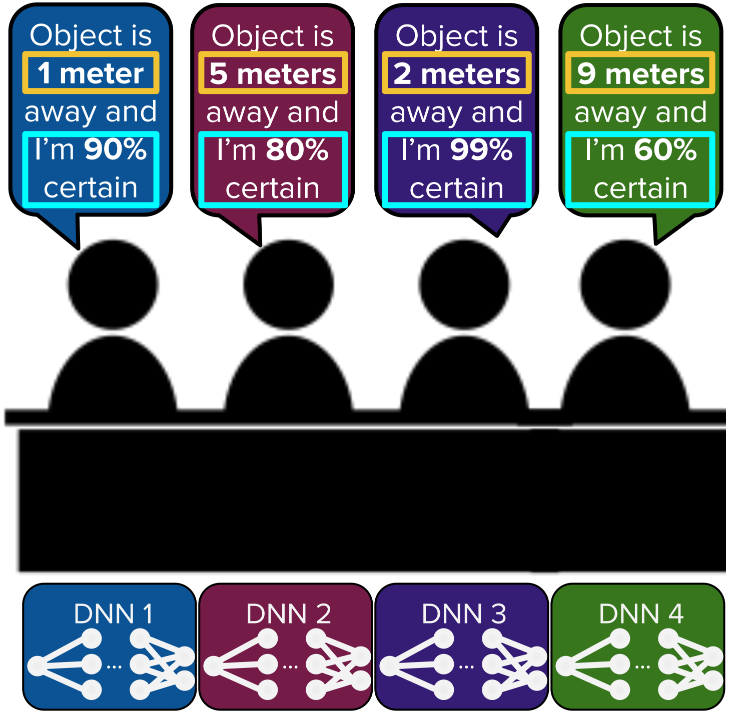
Model (epistemic) uncertainty is high when a diverse set of experts (DNNs) **disagree** in their predictions

Let’s Convene a Panel of Diverse Experts (DNN Ensemble)



Data (aleatoric) uncertainty is the average of the experts’ (DNNs) individual predicted uncertainty

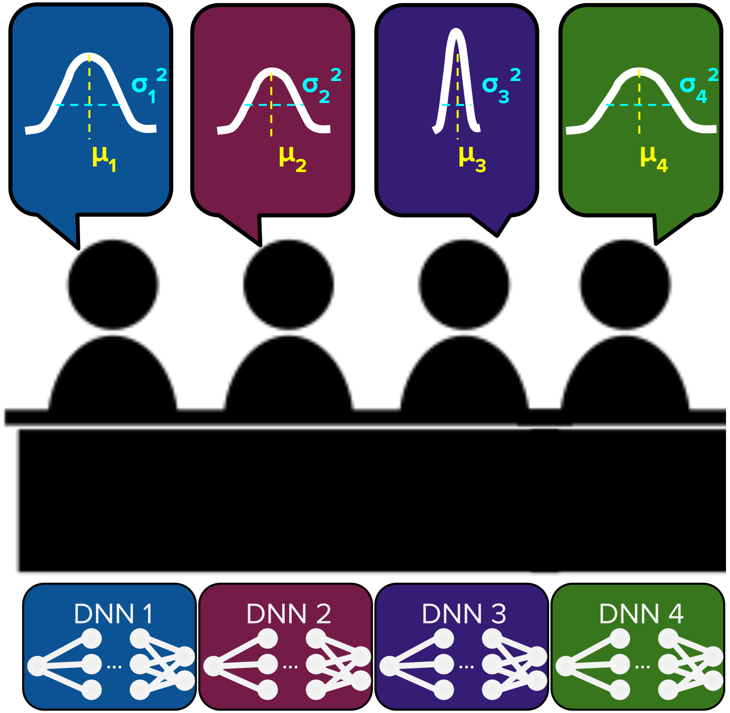
Let’s Convene a Panel of Diverse Experts (DNN Ensemble)



Total uncertainty =

$\underbrace{\text{model uncertainty}}_{\text{variance(experts' predicted depths)}} + \underbrace{\text{data uncertainty}}_{\text{average(experts' predicted variance)}}$

Let’s Convene a Panel of Diverse Experts (DNN Ensemble)



Each DNN predicts mean and variance of depth (Gaussian)

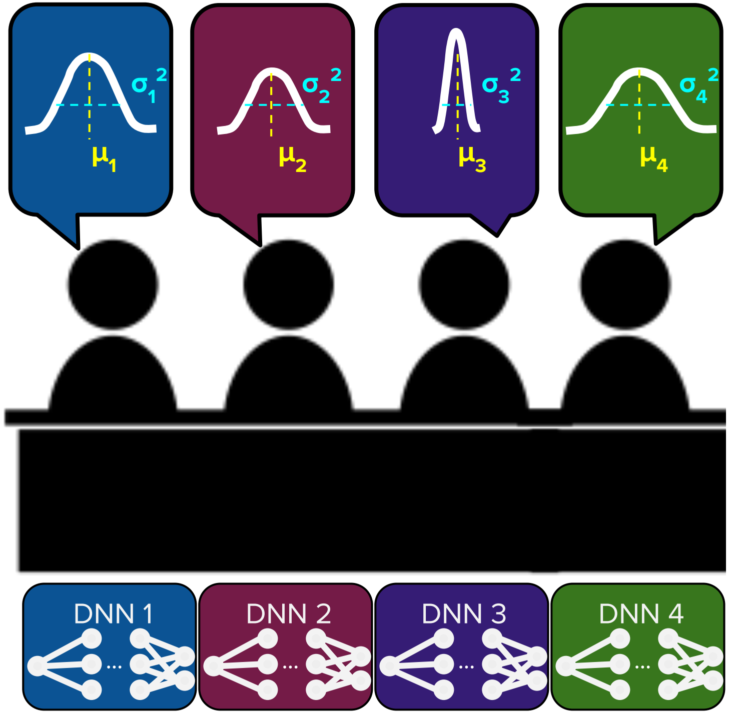
Total uncertainty =

model uncertainty + data uncertainty

variance(DNNs’ predicted μ depth)

average(DNNs’ predicted σ^2)

Motivation: Traditional Uncertainty Estimation is Expensive

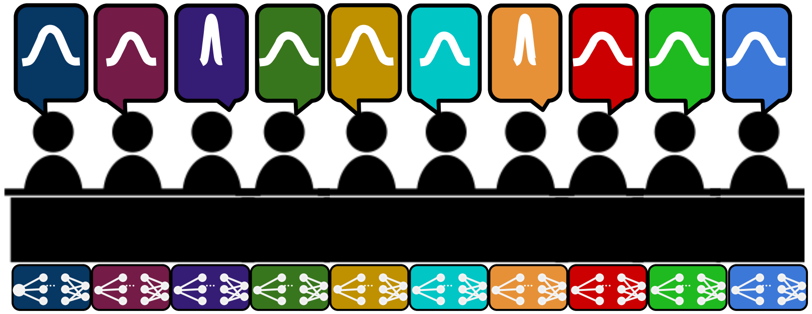


Now requires M inferences per input, making uncertainty estimation extremely computationally expensive

Total uncertainty = model uncertainty + data uncertainty

$$\text{Total uncertainty} = \underbrace{\text{model uncertainty}}_{\text{variance(DNNs' predicted } \mu \text{ depth)}} + \underbrace{\text{data uncertainty}}_{\text{average(DNNs' predicted } \sigma^2\text{)}}$$

Motivation: Traditional Uncertainty Estimation is Expensive



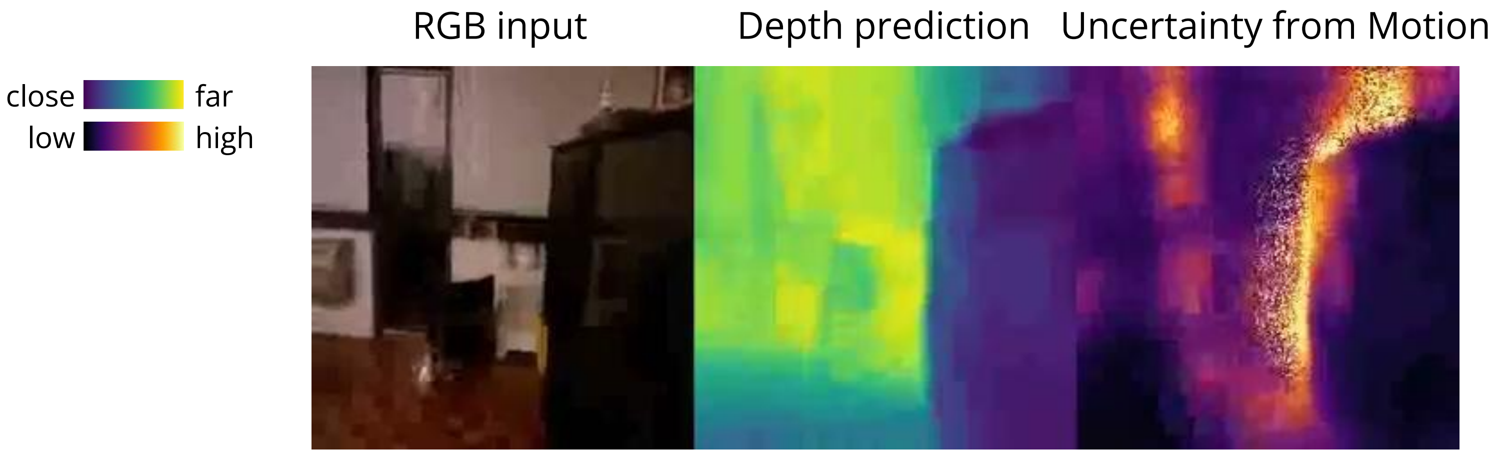
When M = 10, 10X inferences per input, making uncertainty estimation extremely computationally expensive

Total uncertainty =

$$\underbrace{\text{model uncertainty}}_{\text{variance(DNNs' predicted } \mu \text{ depth)}} + \underbrace{\text{data uncertainty}}_{\text{average(DNNs' predicted } \sigma^2\text{)}}$$

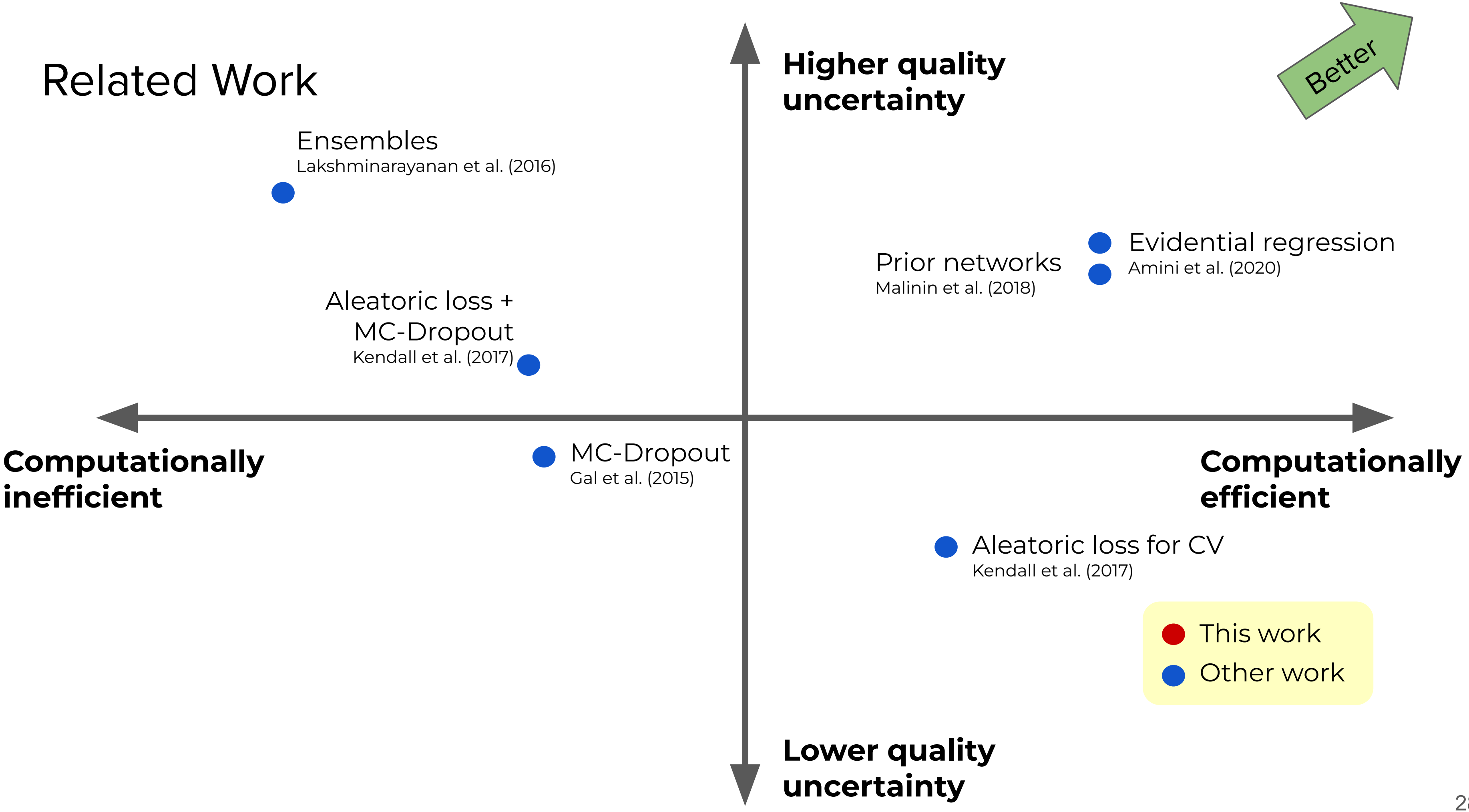
Contribution: Uncertainty from Motion (UfM)

We introduce a new algorithm called **Uncertainty from Motion (UfM)**

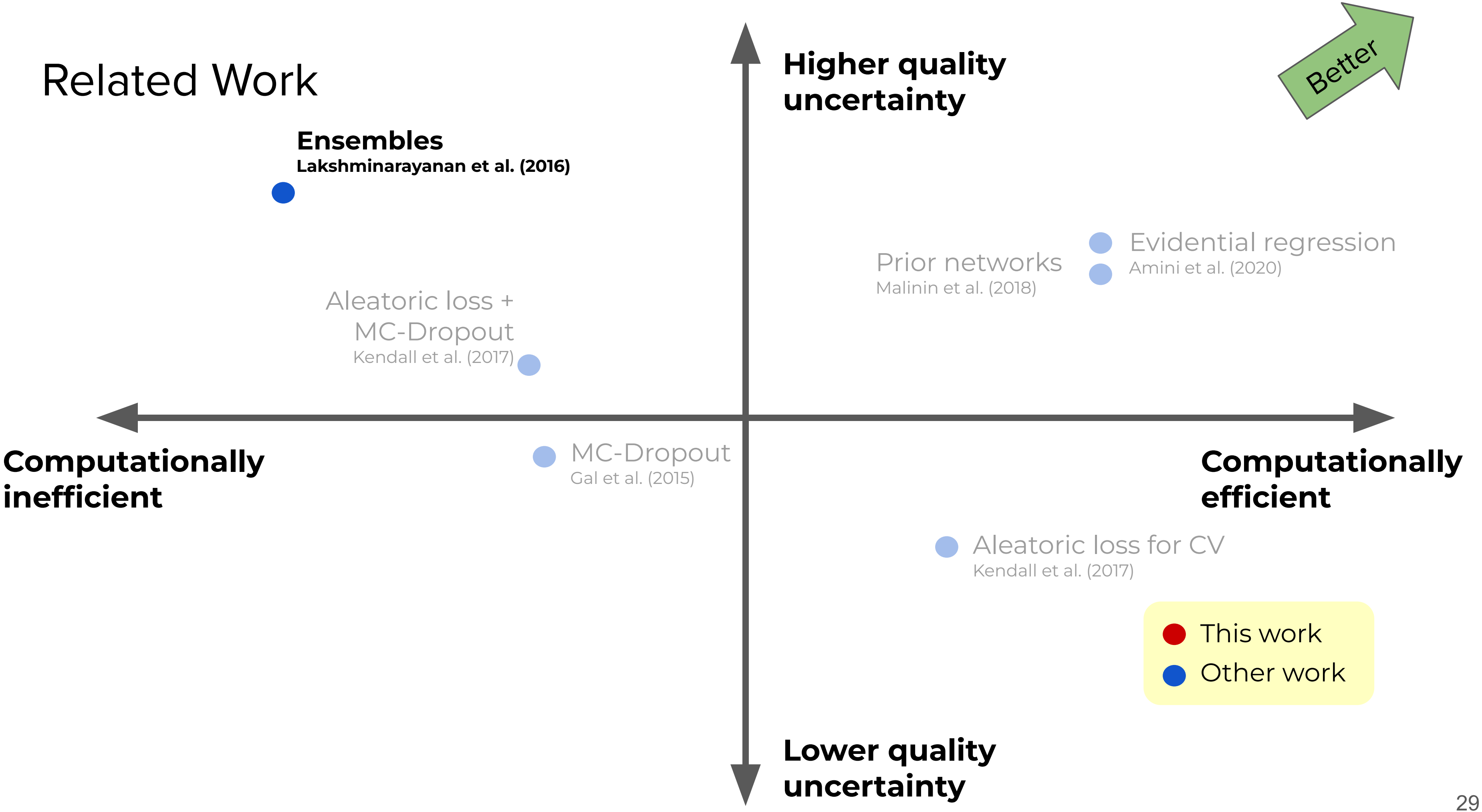


which enables close to state-of-the-art ensemble uncertainty quality while **only requiring one DNN inference per input.**

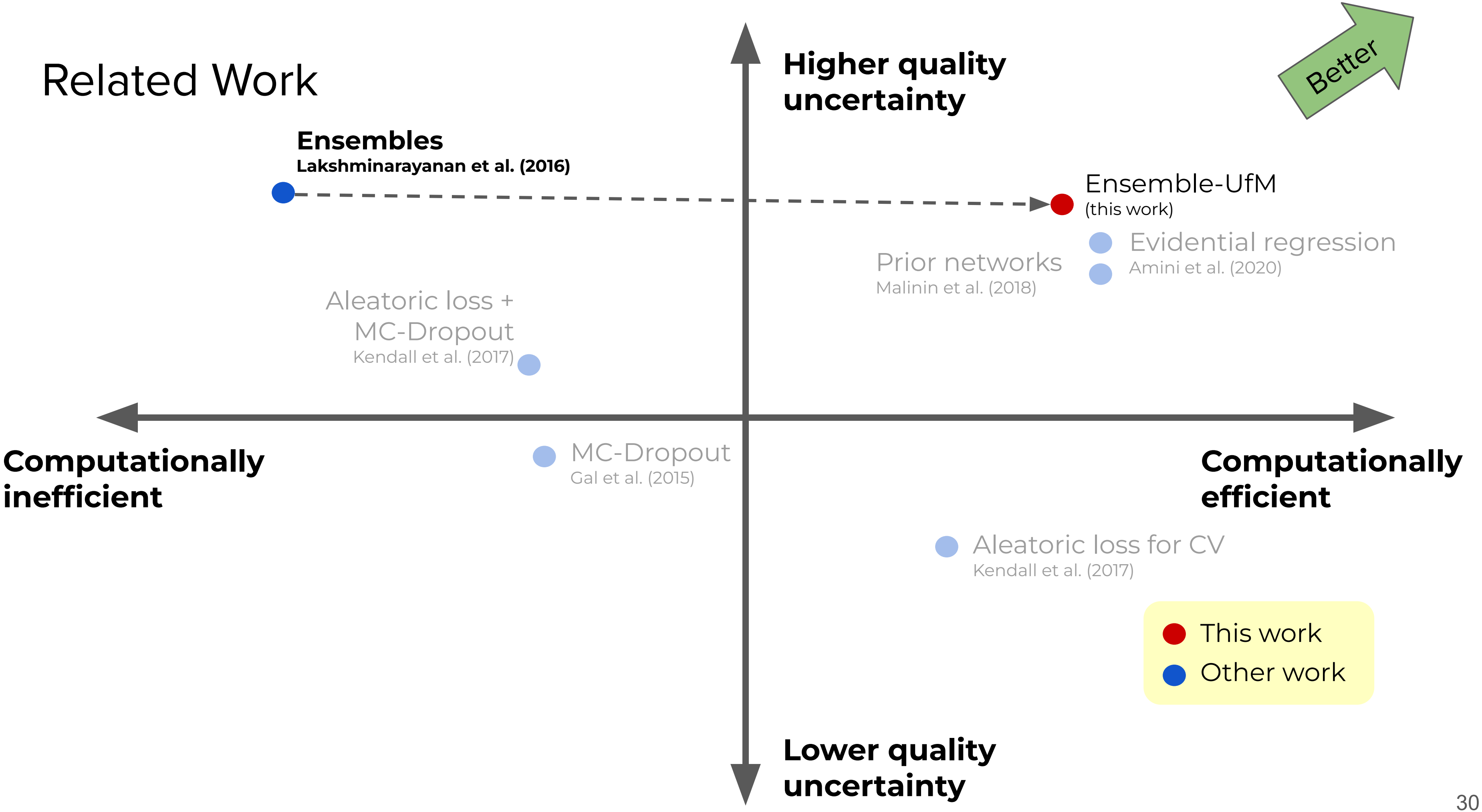
Related Work



Related Work

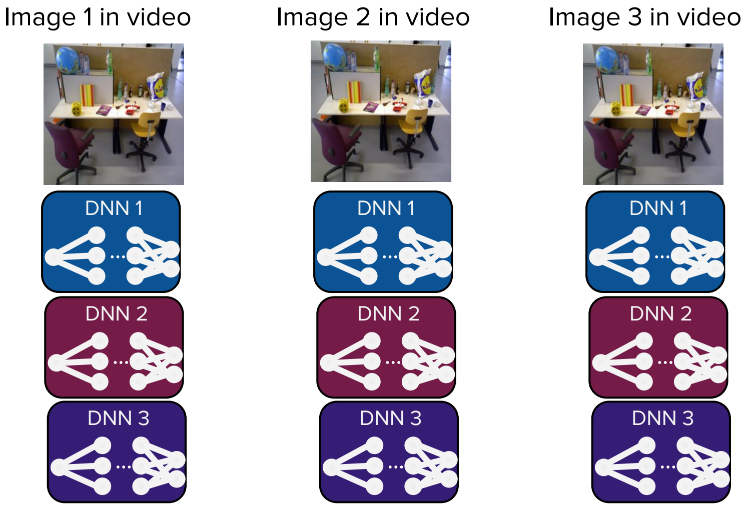


Related Work



Algorithm Overview

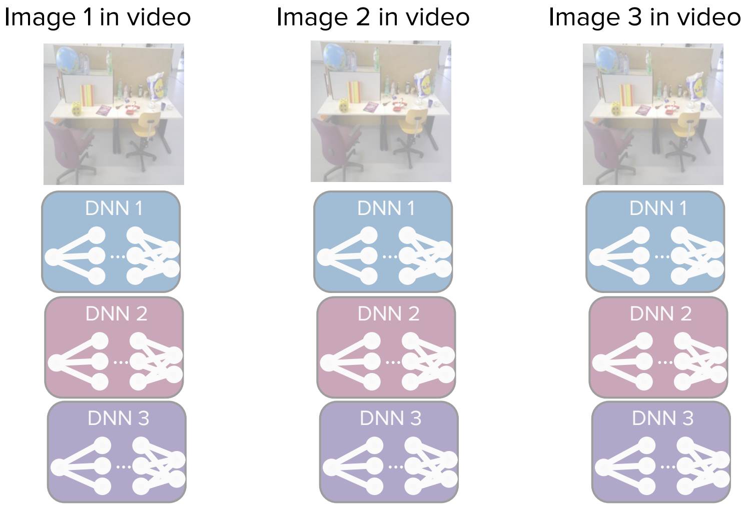
UfM enables us to obtain near ensemble uncertainty quality at a fraction of the latency and energy cost



where instead of running M inferences per input,

Algorithm Overview

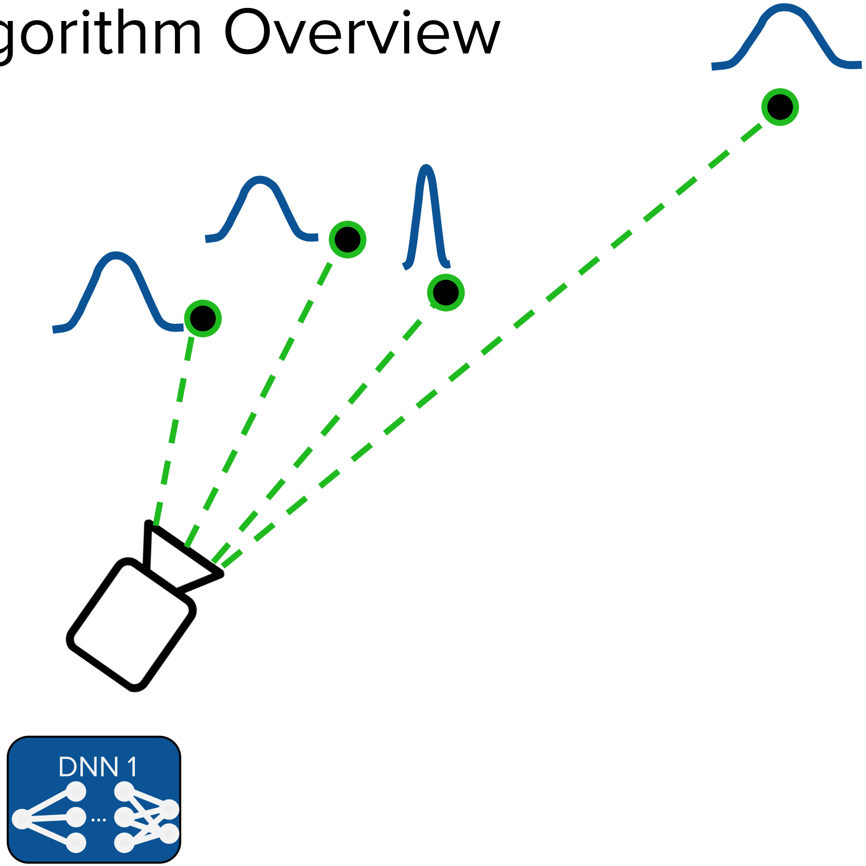
UfM enables us to obtain near ensemble uncertainty quality at a fraction of the latency and energy cost



where instead of running M inferences per input,

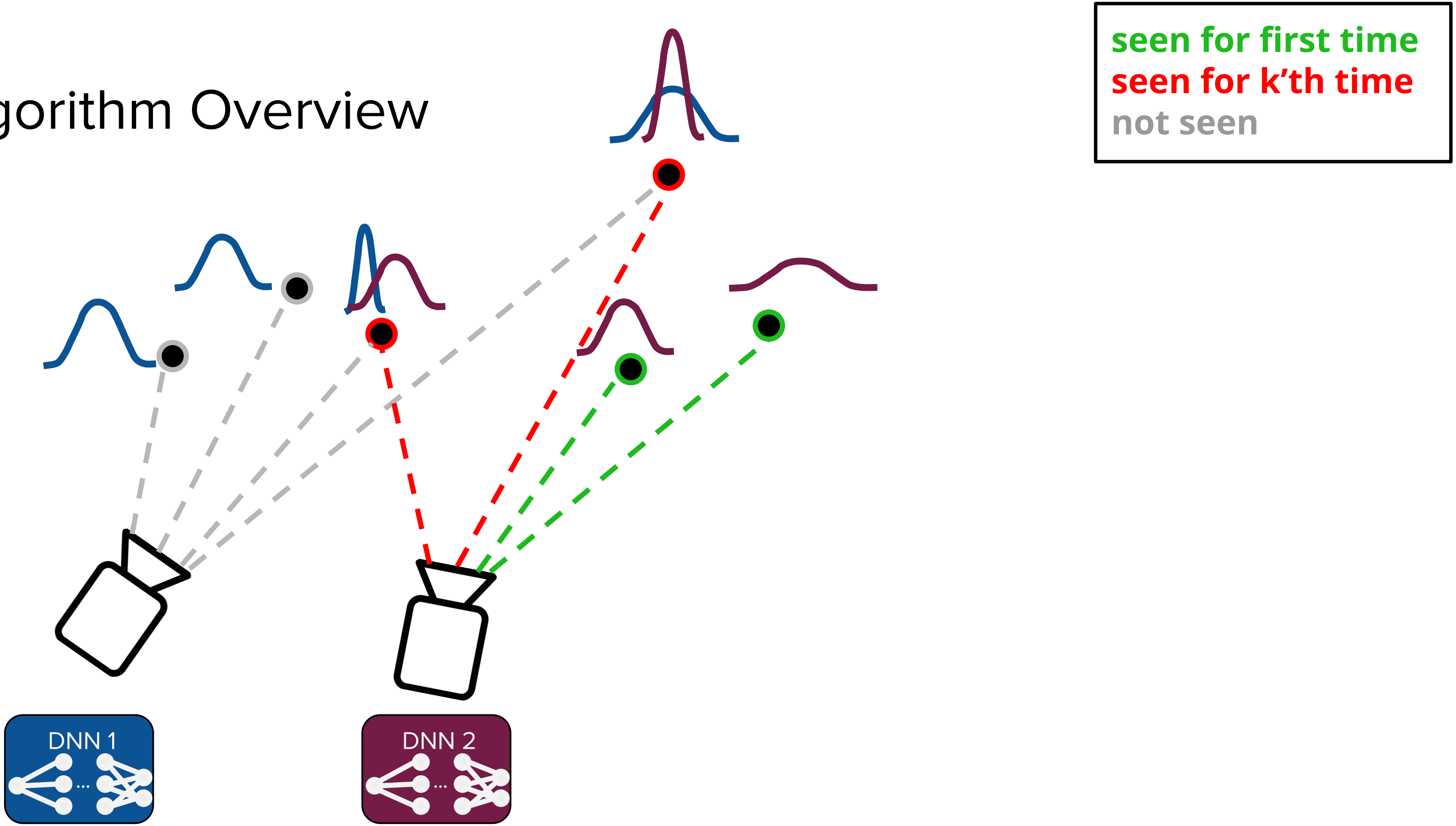
we can run one inference per input, and merge temporally

Algorithm Overview



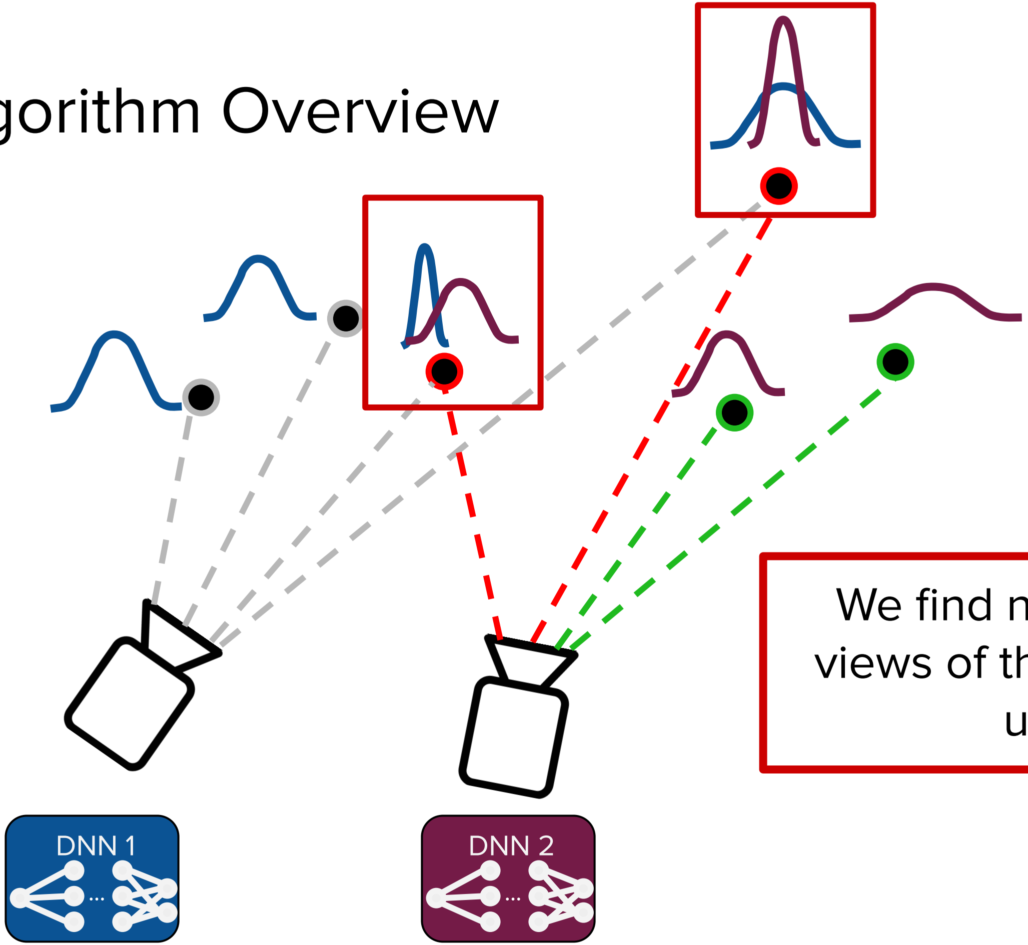
seen for first time
seen for k'th time
not seen

Algorithm Overview



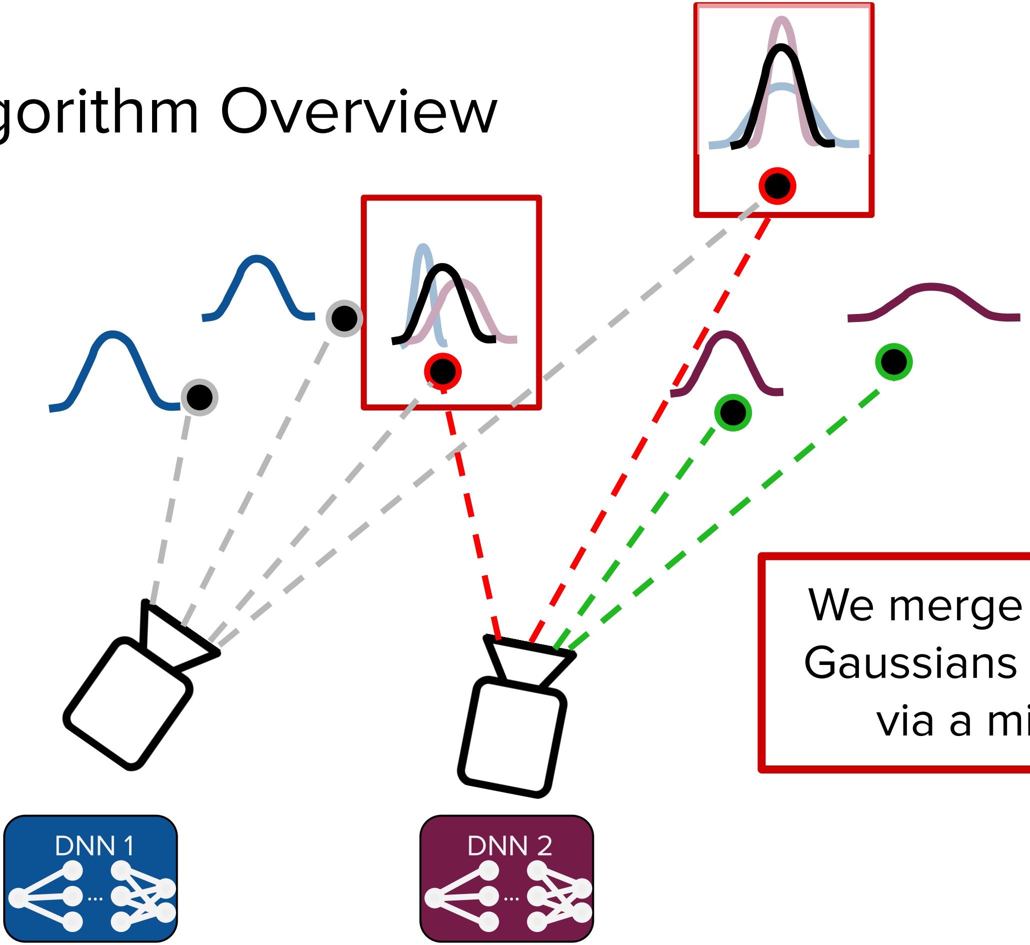
Algorithm Overview

seen for first time
seen for k'th time
not seen



We find noisy correspondences of views of the same point in 3D space using reprojection

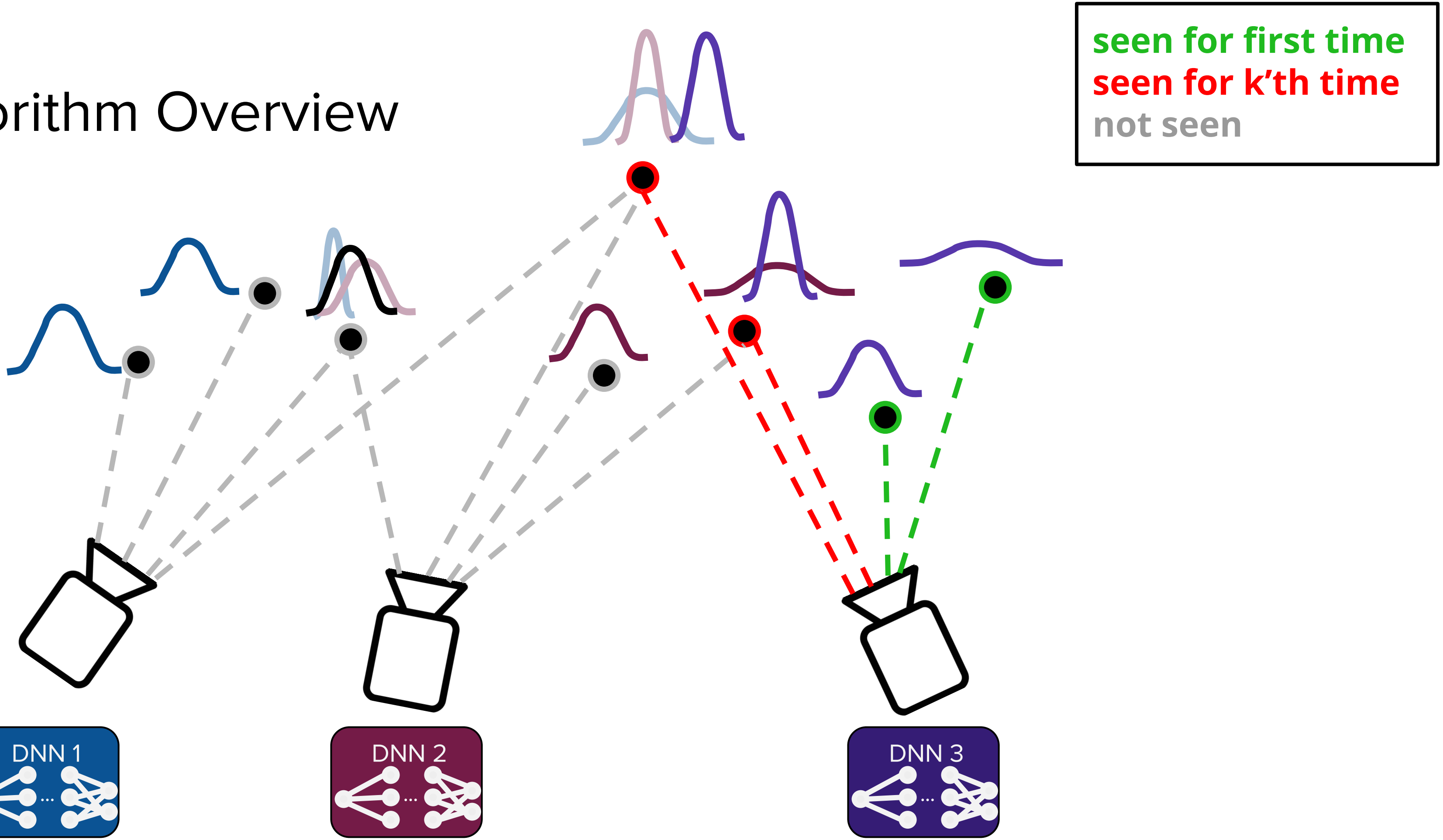
Algorithm Overview



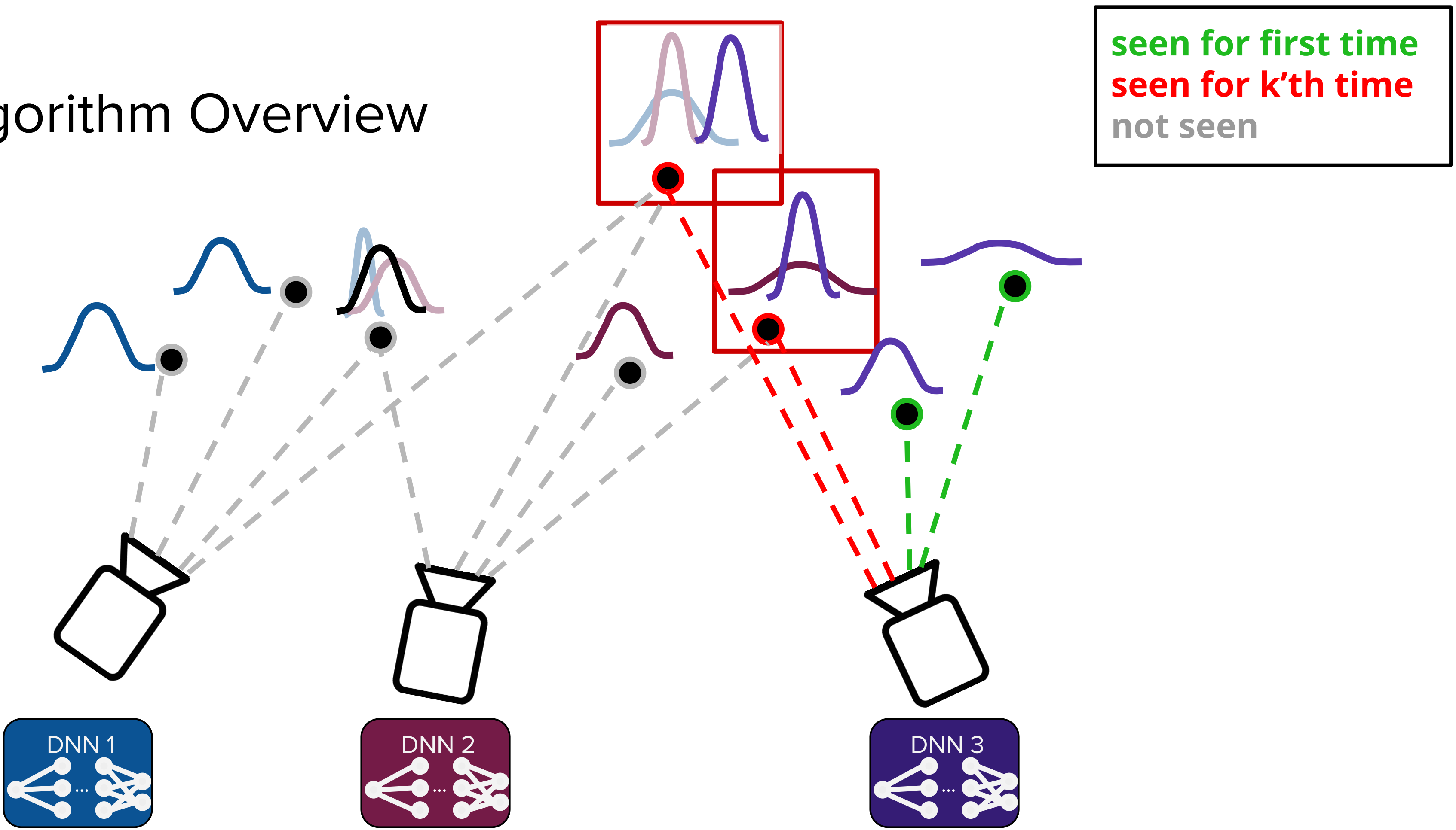
seen for first time
seen for k'th time
not seen

We merge the different DNNs' predicted Gaussians of the same point in 3D space via a mixture of Gaussians update

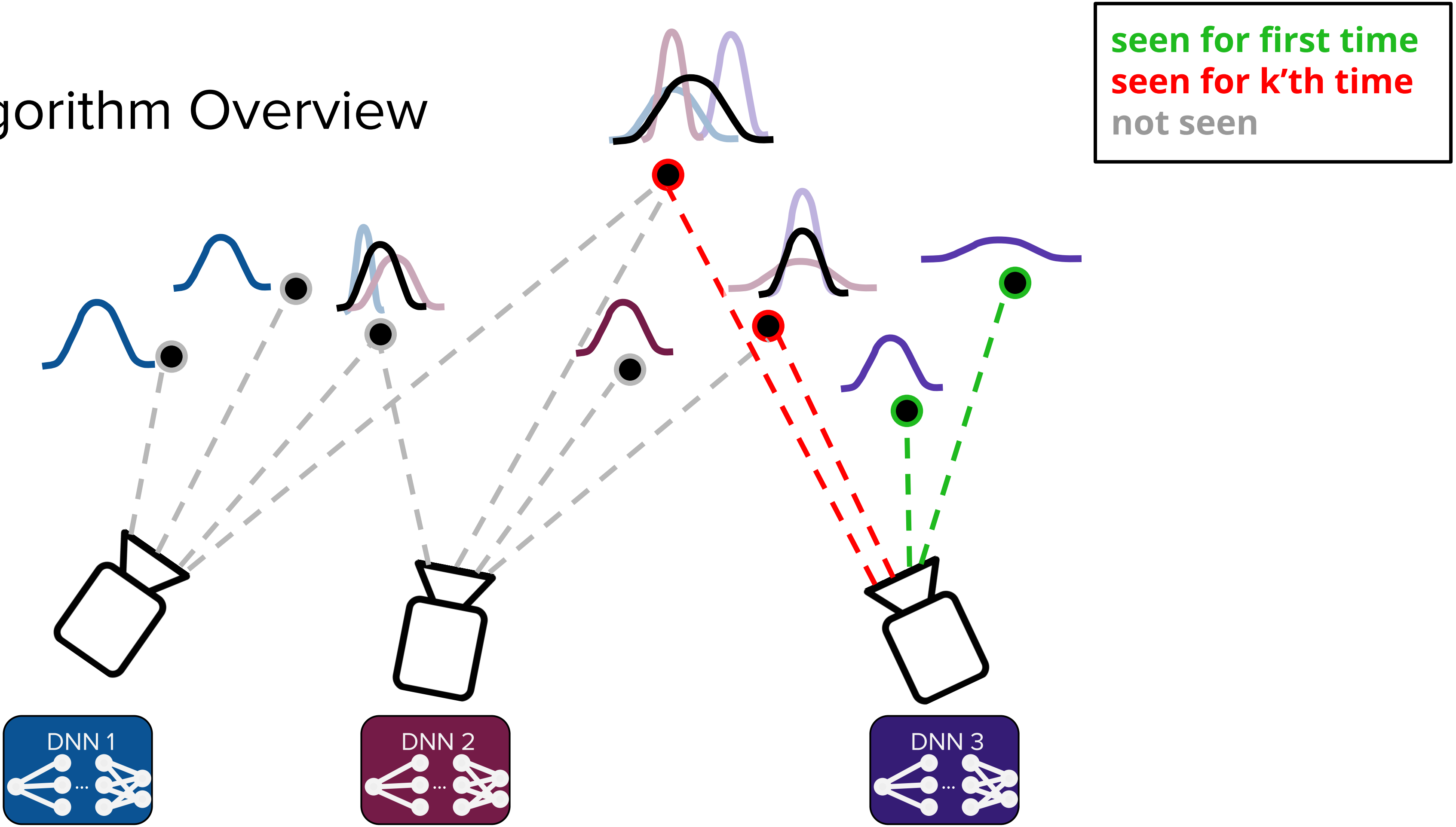
Algorithm Overview



Algorithm Overview

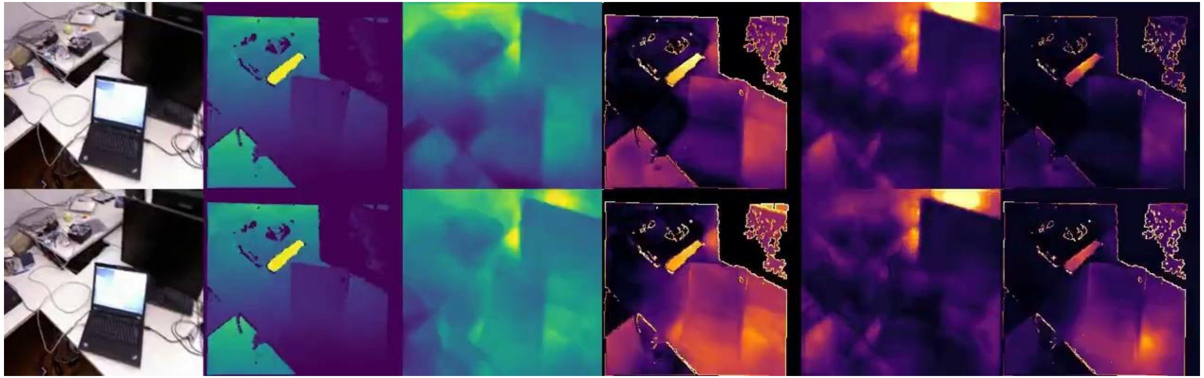


Algorithm Overview



Comparable Uncertainty Quality to Ensembles

RGB | Ground-truth | Depth Pred. | Error | Uncertainty | NLL (lower is better)



Ensemble

ECE ↓ = 0.30

(ensemble video sped up x6 to compare equivalent frames)

UfM

ECE ↓ = 0.26

FCDenseNet architecture on Nvidia RTX 2080 Ti

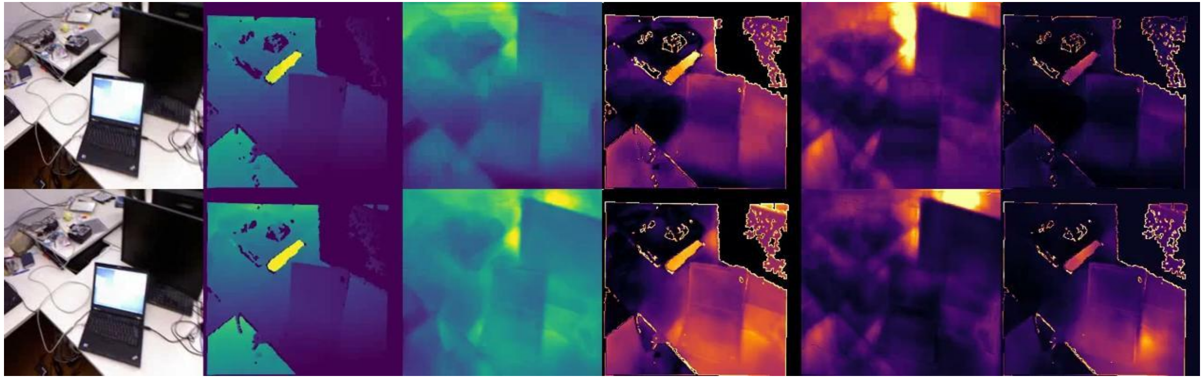
close — far

low — high

We show that the uncertainty quality is comparable to SOTA ensemble method

Lower Latency and Energy with UfM

RGB | Ground-truth | Depth Pred. | Error | Uncertainty | NLL (lower is better)



Ensemble
5 fps
45 Joules/frame

UfM
32 fps
5 Joules/frame

FCDenseNet architecture on Nvidia RTX 2080 Ti

close — far

low — high

Near ensemble uncertainty quality at a fraction of the latency and energy cost

Key Takeaways

1) DNN uncertainty conventionally requires M inferences per input since it requires a “panel of experts” (*e.g.*, an ensemble) to measure disagreement.
2) We can obtain near ensemble uncertainty quality with one inference per input, lowering the latency and energy cost of uncertainty estimation.
3) UfM uses the temporal redundancy in video inputs to merge per-pixel predictions across a sequence that are multiple views of the same point.

Sudhakar, Soumya, Sertac Karaman, and Vivienne Sze. "Uncertainty from Motion for DNN Monocular Depth Estimation." IEEE International Conference on Robotics and Automation (ICRA). May 2022.

Code to be released on Github

Work funded by NSF Cyber-Physical Systems, NSF Real-Time Machine Learning, and the MIT-Accenture Fellowship.