Predictive Uncertainty for Robot Perception

Ali Harakeh

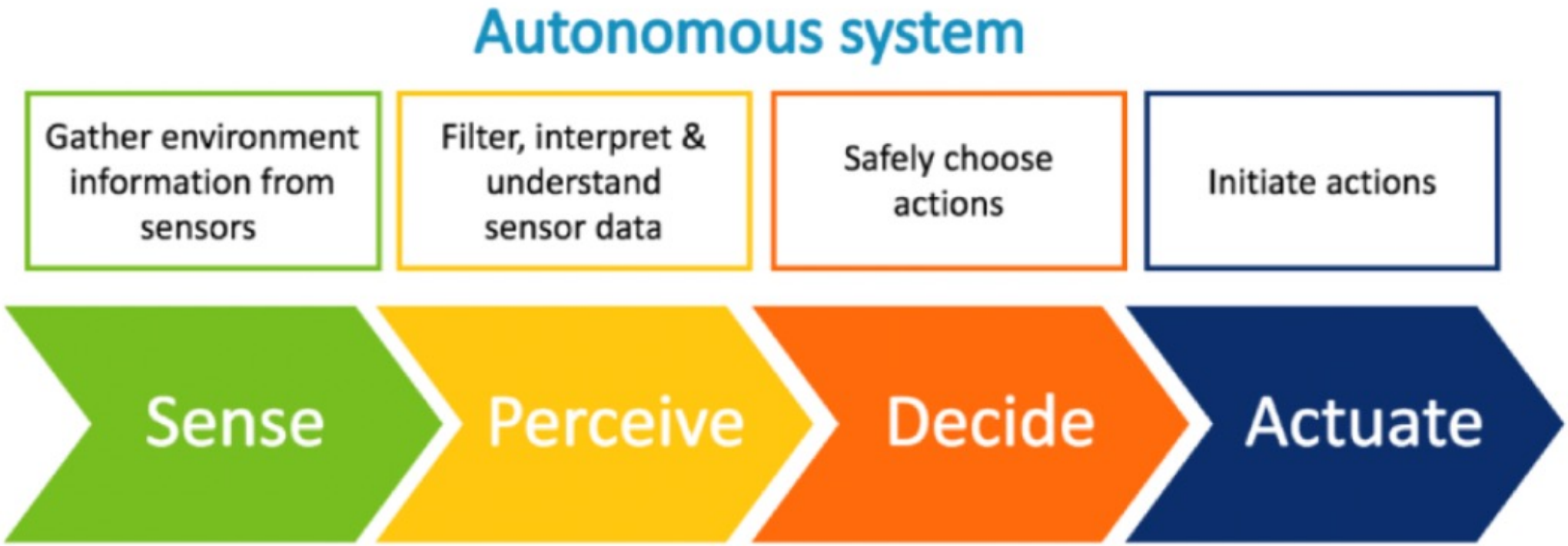
November 17th, 2021

Table of Contents

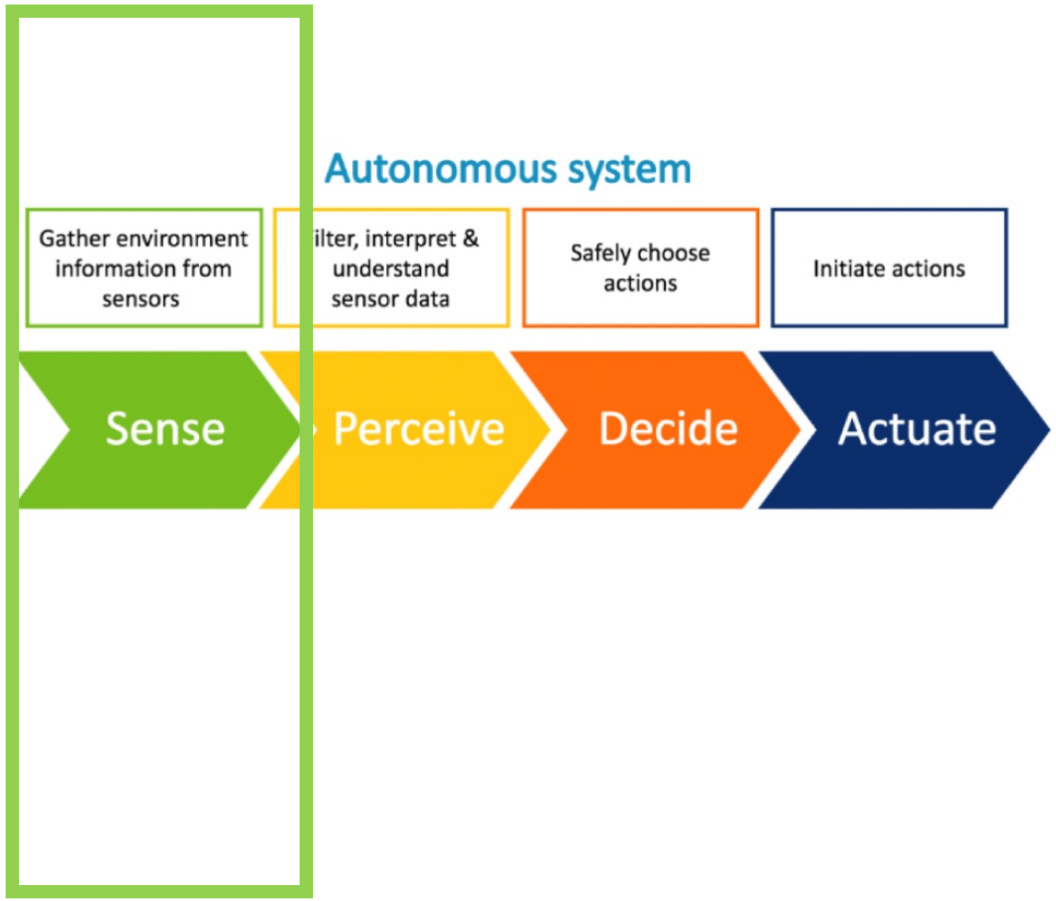
- Motivation
- Background
- Predictive Uncertainty: Estimation
- Predictive Uncertainty: Evaluation
- Conclusion

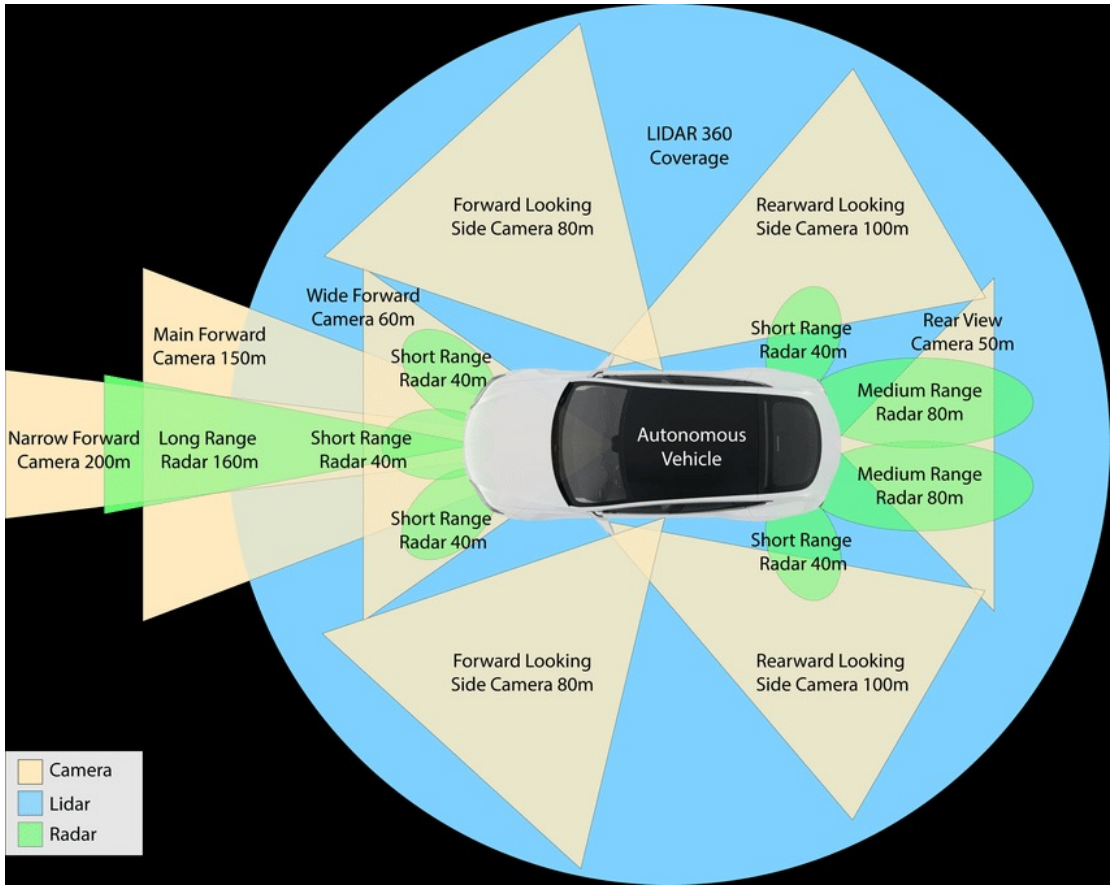
Motivation

Motivation | Importance of Predictive Uncertainty



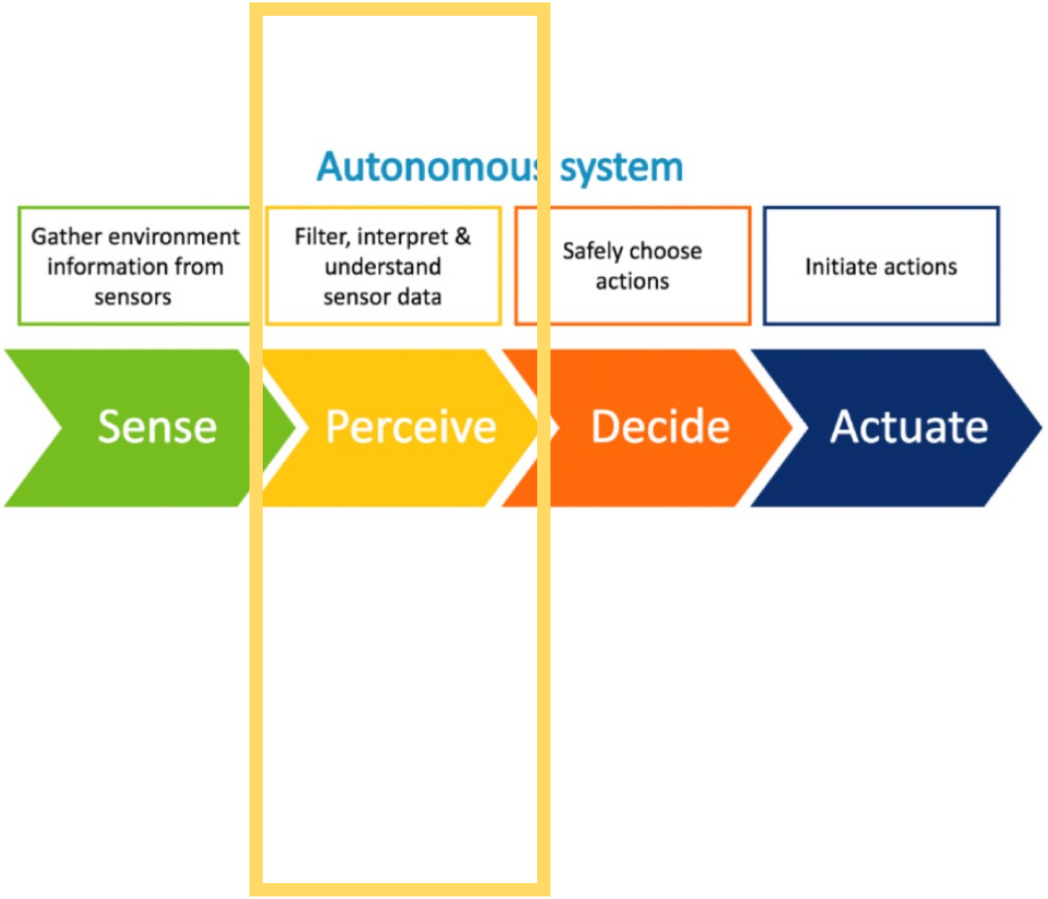
Motivation | Importance of Predictive Uncertainty





Motivation | Common Robot Perception Tasks

- Camera Image → Depth Estimation



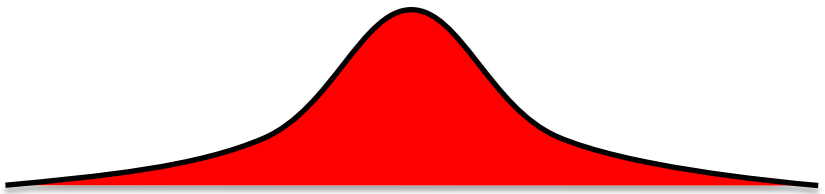
Motivation | What is Predictive Uncertainty

- **Predictive uncertainty** is a **mathematical tool** to help us quantify the trustworthiness of predictions generated by a robot perception model.

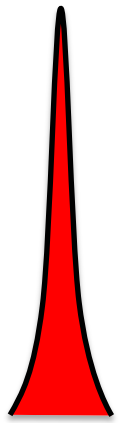
- Uncertainty → Naturally modeled using Probability Theory*.

- **Example:** $\mathcal{N}(\mu, \sigma^2)$

High Uncertainty

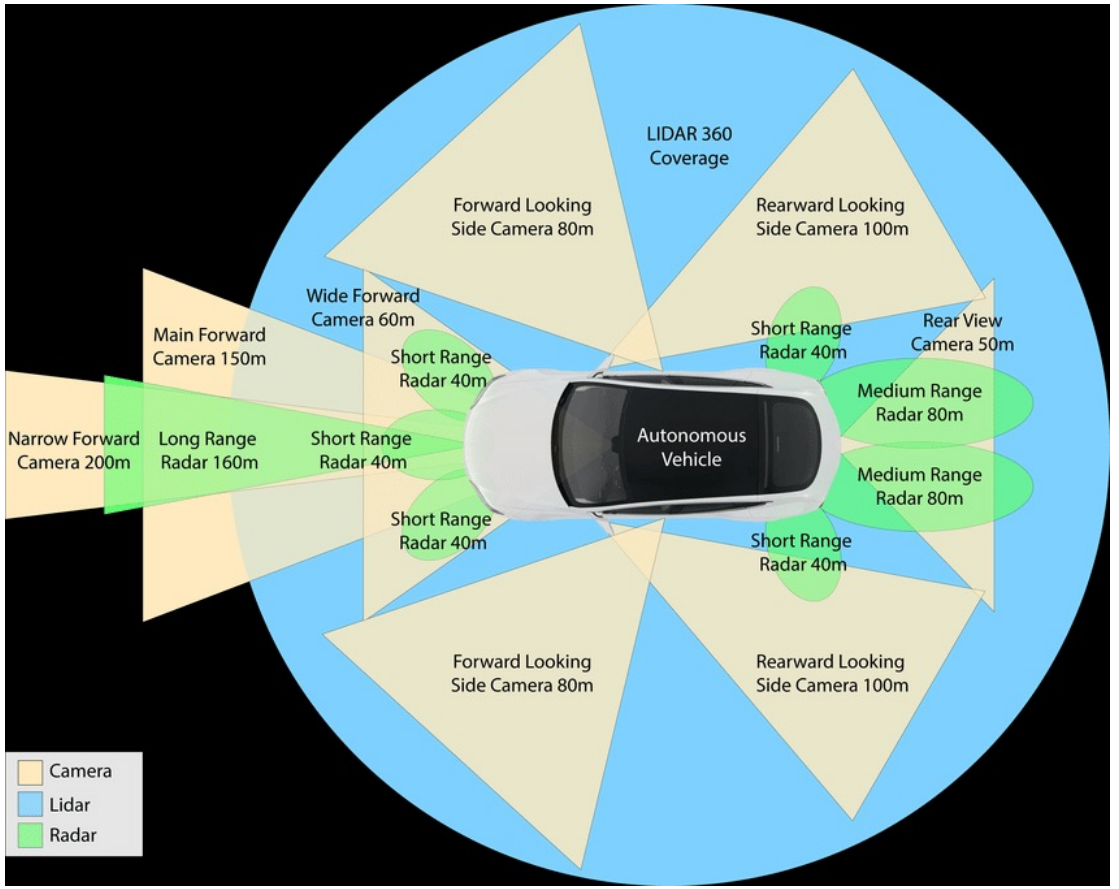


Low Uncertainty



Motivation | Importance of Predictive Uncertainty

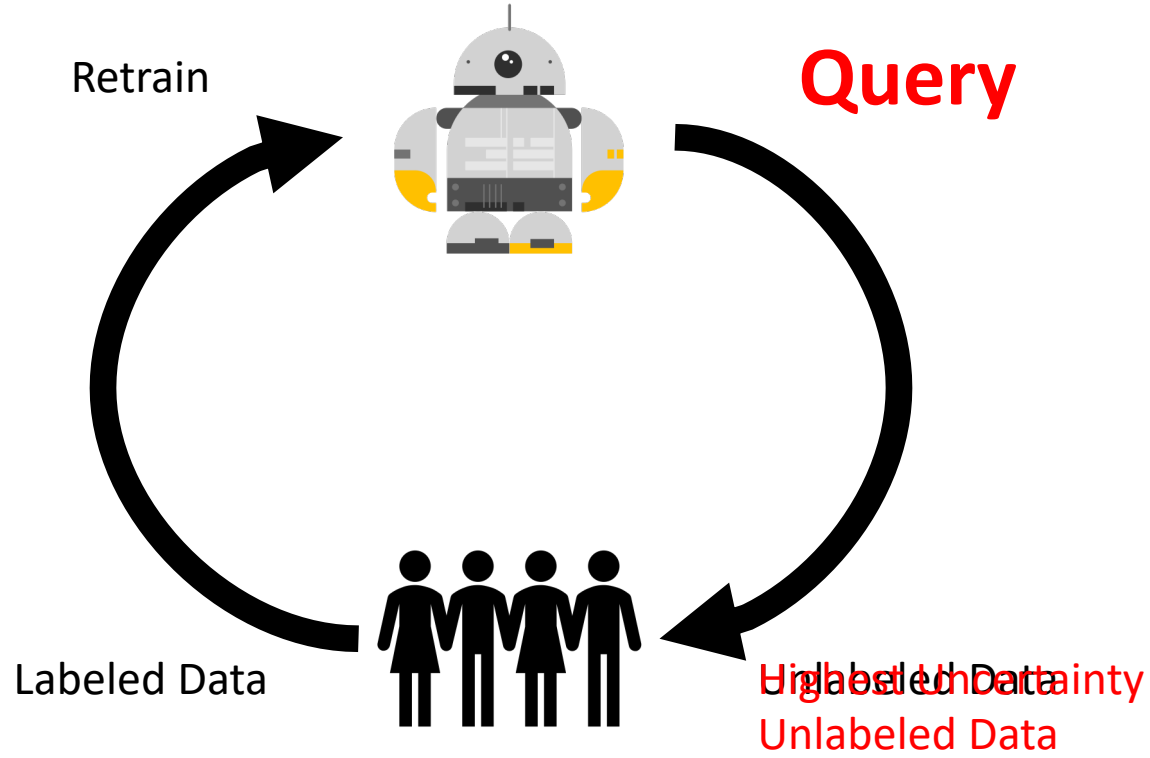
- **Example:** Sensor Fusion
  - Robots use many sensors to perform similar perception functions.
  - Multiple sensors can provide inconsistent information.
  - Uncertainty allows us to fuse multiple sources of information or to determine which one is more **trustworthy**!



Motivation | Importance of Predictive Uncertainty

- **Example:** Active Learning

  - Robot continuously collects data + predictions.
  - What data should we label to improve robot performance?
  - Annotation budget is not infinite!
  - **Uncertainty** allows us to query the most “challenging” frames.

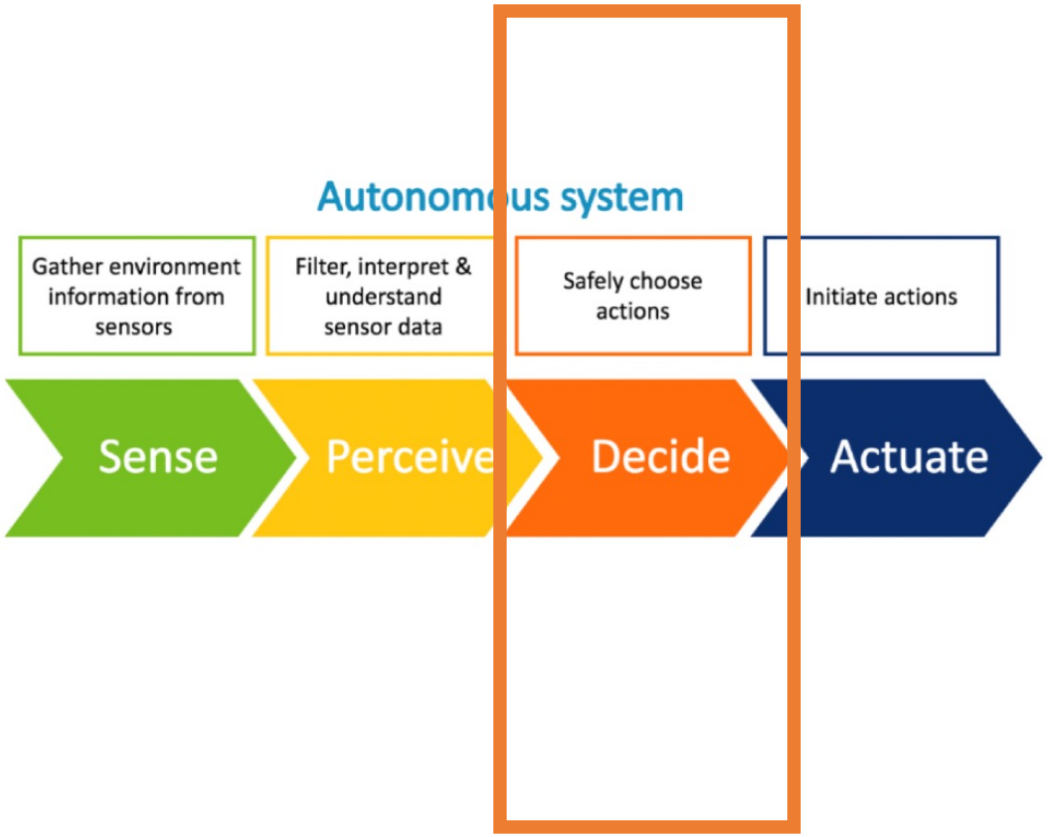


Motivation | Importance of Predictive Uncertainty

- **Example:** Robot Decision Making

  - **Theory:** Every admissible decision rule is a (generalized) Bayes rule.

  - **Practice:** When uncertainty increases, a system can change how it makes decisions to guarantee safe operation.



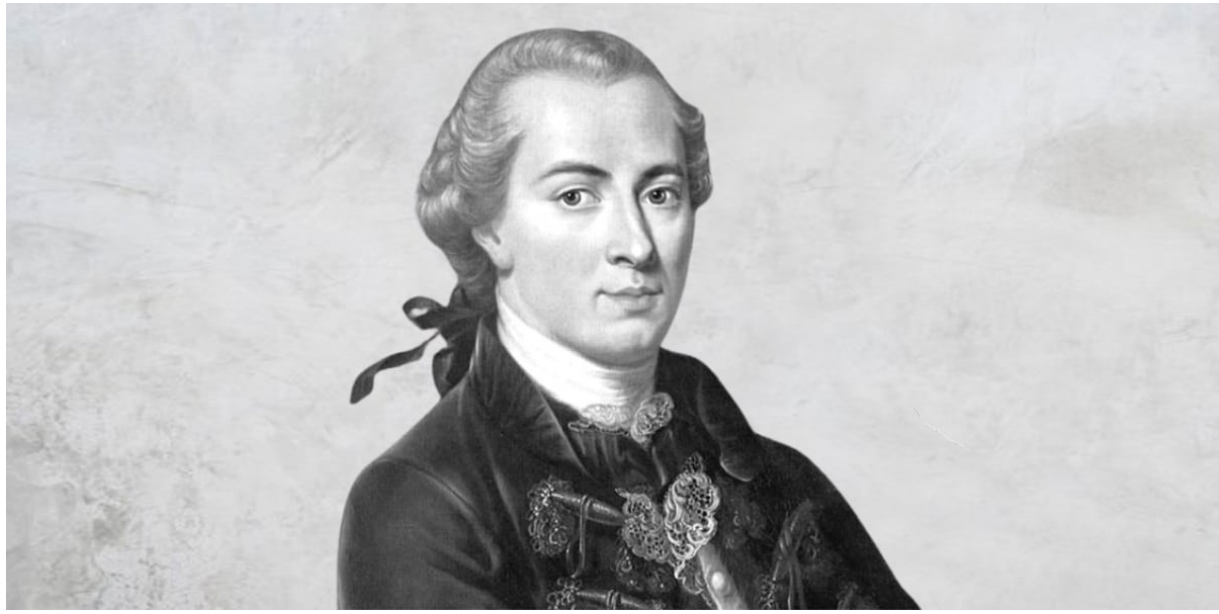
Motivation | Uncertainty to Quantify Knowledge


Critik
der
reinen Vernunft

von
Immanuel Kant
Professor in Königsberg.

Riga,
verlegts Johann Friedrich Hartknoch
1781.

We must avoid **false confidence bred from an ignorance of the probabilistic nature of the world**, from a desire to see **black and white** where we should rightly see **gray**.



Motivation | Uncertainty and Decision Rules

- Autopilot sends Tesla Model 3 to truck.

https://newsabc.net/autopilot-sends-tesla-model-3-to-truck/

- A Tesla on cruise control smashed into a tractor trailer in New Jersey, ripping off half its roof.

https://www.businessinsider.com/tesla-using-cruise-control-crashed-tractor-trailer-new-jersey-2021-3

Motivation | Summary

- Uncertainty is essential for many operations commonly performed by robot software such as fusion, decision making, etc.

- **Questions?**

- **Next: Some background.**

Background

Background | Machine Learning

- Given a **Dataset**

$$\mathcal{D} = \{x_n, y_n | n \in 1, \ldots, N\}$$

- and an **unknown** data generating function

$$y = f(x)$$

- we train a ML model with parameters θ to **estimate**

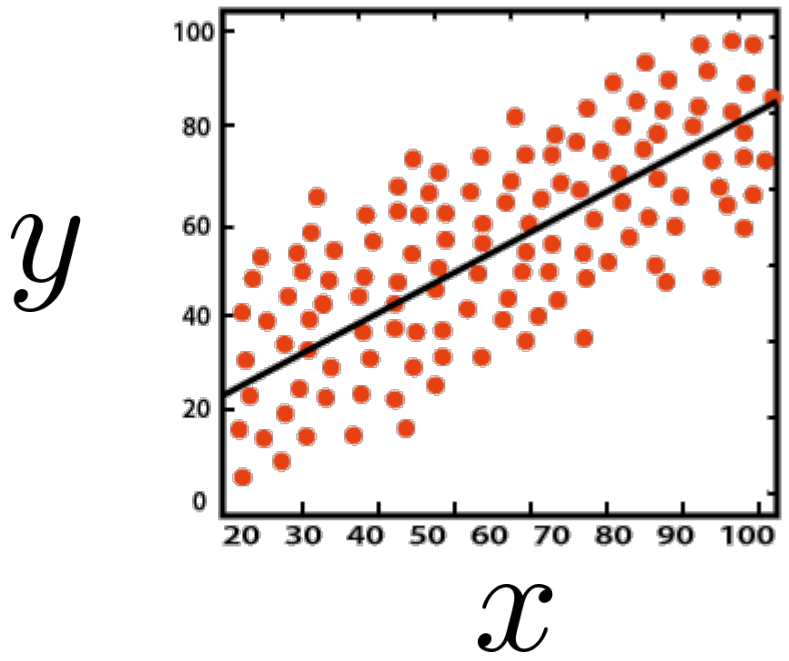
$$\hat{f}_\theta(x) \approx f(x)$$

- by minimizing a loss function

$$\mathcal{L}(y_n, \hat{f}_\theta(x_n))$$

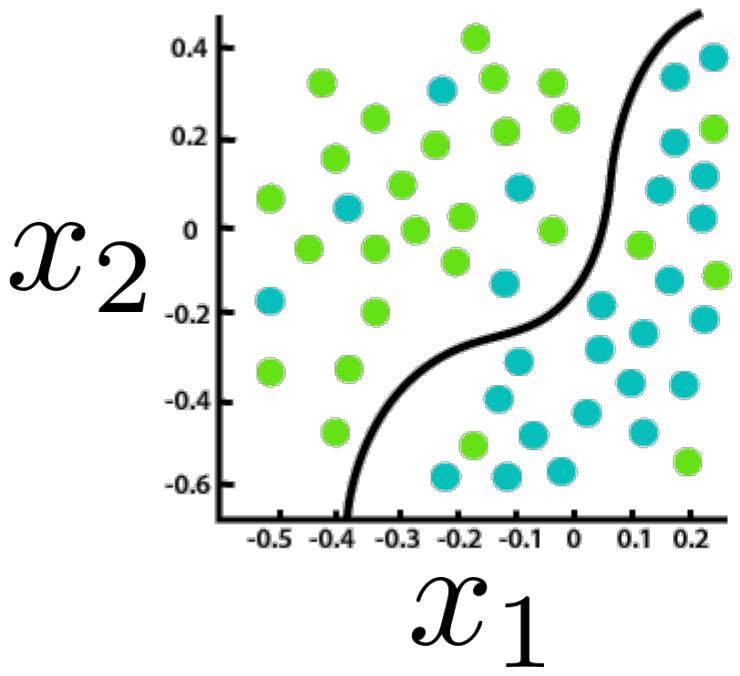
- Regression:

$$y \in \mathbb{R}^d$$



- Classification:

$$y \in \{1, \ldots, K\}$$



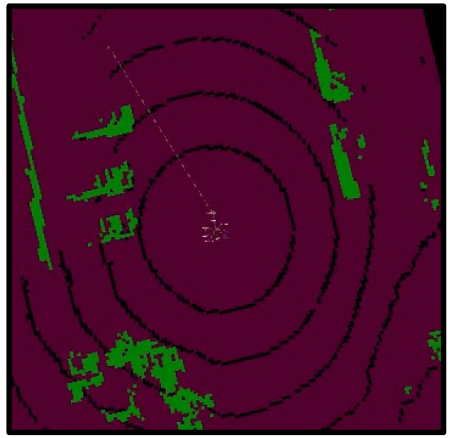
Background | Common Robot Perception Tasks

- Regression:
  - Depth Estimation

  - Object Localization

- Classification:
  - Semantic Segmentation

  - Occupancy Grid Mapping



Background | Sources of Uncertainty In Machine Learning

- Given two bins each containing 20 tokens:

20 Tokens:
Half worth $10,
Half worth $0

20 Tokens:
All worth $6

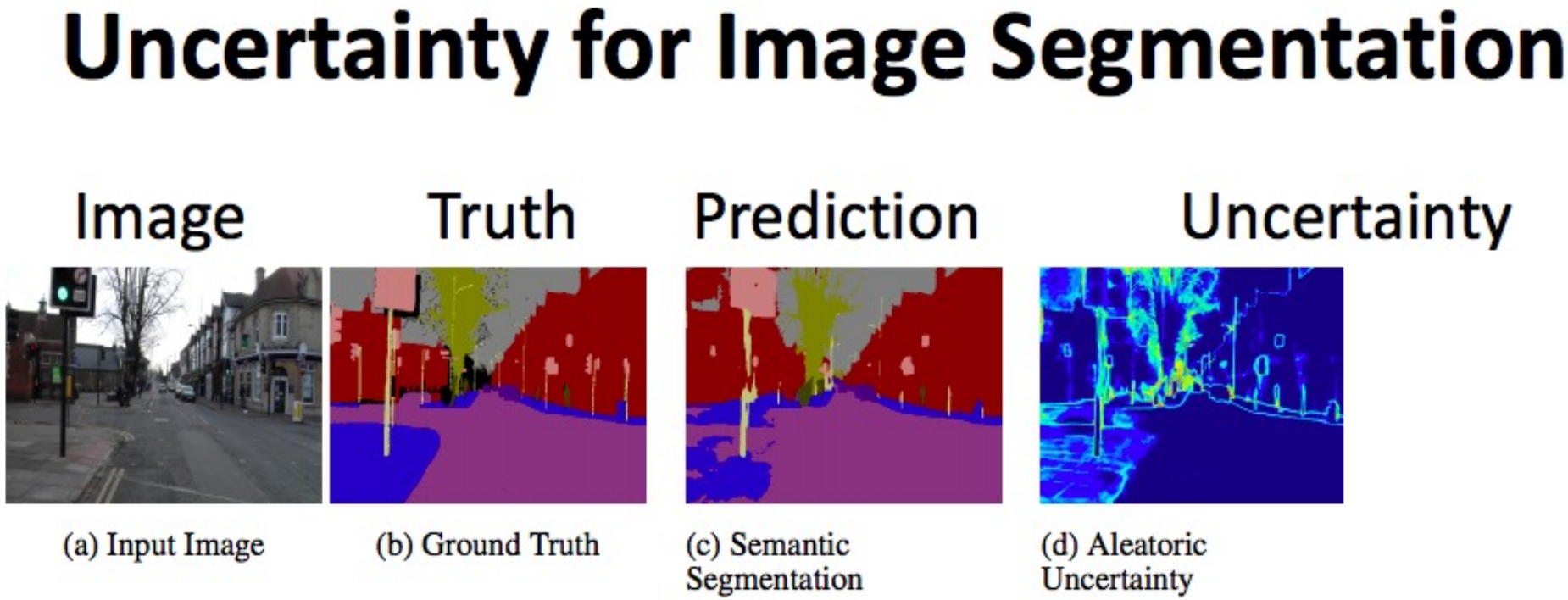
- You want to choose 5 tokens from the two bins.
- You want to maximize your profit.
- How will you choose these tokens? What is the source of your uncertainty?

- If you are allowed to observe **one token** from **bin 1,** would that change how you make your decision?

Background | Sources of Uncertainty In Machine Learning

- **Aleatoric Uncertainty:**
  - Results from the Stochasticity of the data generating process.
  - Cannot be reduced by collecting more data.

  - **Example:**



Background | Sources of Uncertainty In Machine Learning

- Given two bins each containing 20 tokens:

20 Tokens: All worth $10 OR All worth $0	**20 Tokens:** All worth $6

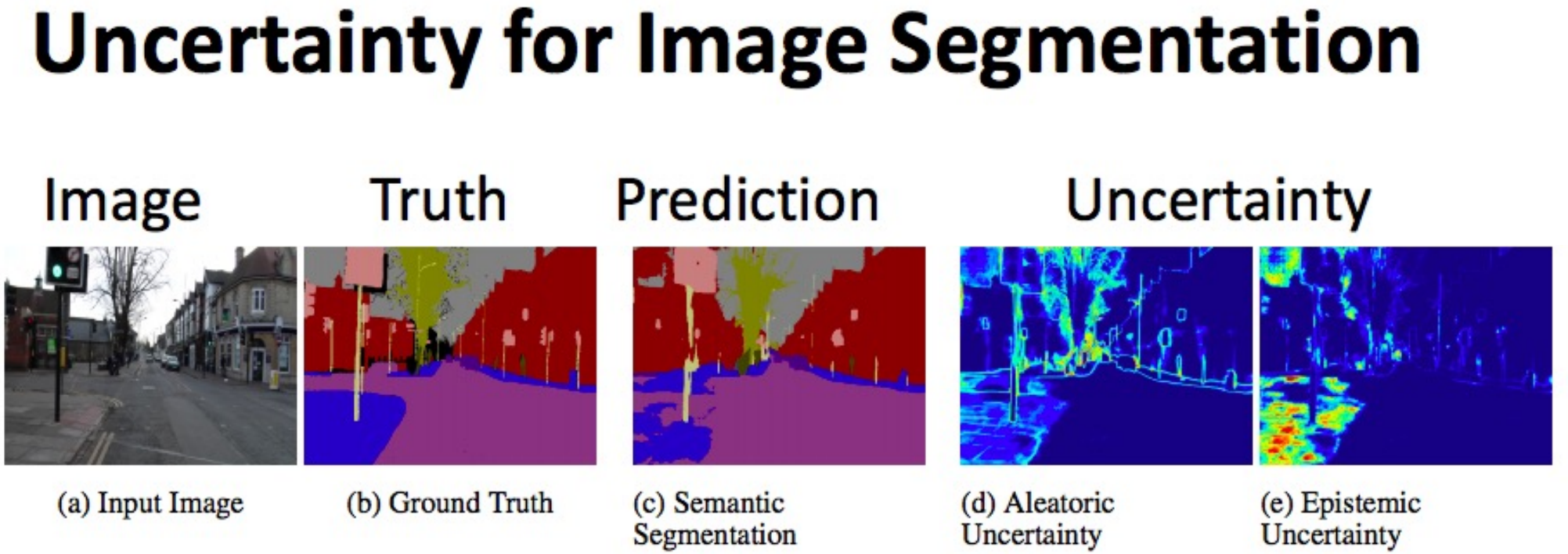
- You want to choose 5 tokens from the two bins.
- You want to maximize your profit.
- How will you choose these tokens? What is the source of your uncertainty?

- If you are allowed to observe **one token** from **bin 1,** would that change how you make your decision?

Background | Sources of Uncertainty In Machine Learning

- **Epistemic Uncertainty:**
  - Results from the ignorance of the best model parameters.
  - Can be reduced by collecting more data.

  - **Example:**



Predictive Uncertainty: Estimation

Background | Estimating Aleatoric Uncertainty

- Given a **Dataset**

$$\mathcal{D} = \{x_n, y_n | n \in 1, \ldots, N\}$$

- and an **unknown** data generating function / conditional distribution

$$y \simeq f(x) \quad y \simeq p(y|x)$$

- we train a machine learning model with parameters θ to **estimate**

$$\hat{f}_\theta(x) \approx f(x) \quad \hat{p}_\theta(y|x) \approx p(y|x)$$

- by minimizing a loss function

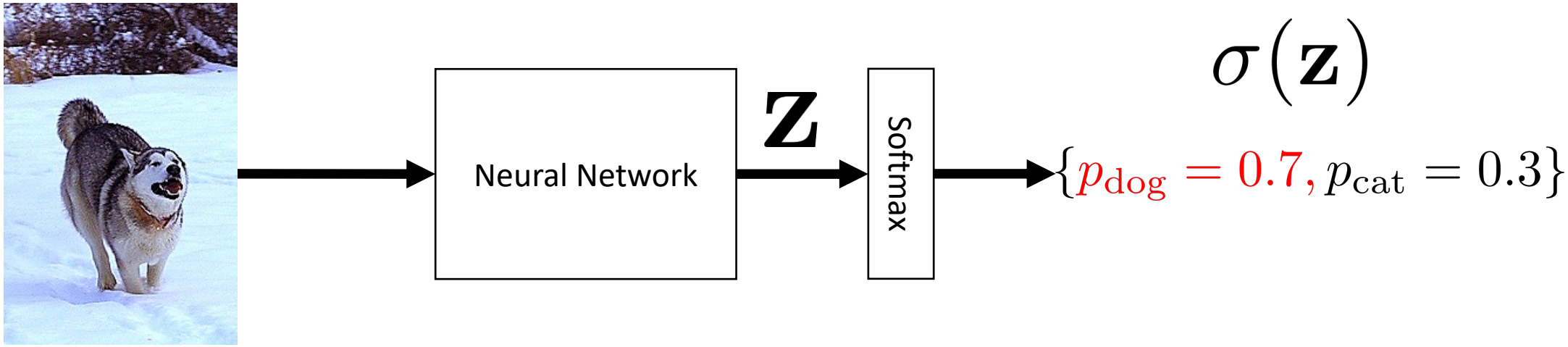
$$\mathcal{L}(y_n, \hat{f}_\theta(x_n)) = -\log \hat{p}_\theta(y_n|x_n)$$

$$\hat{p}_\theta(y|x) = Cat(p_1(x,\theta), \ldots, p_K(x,\theta))$$

$$\{\text{Dog}, \text{Cat}\}$$

$$\{p_{\text{dog}} = 0.7, p_{\text{cat}} = 0.3\}$$

Estimation | Classification Predictive Distributions

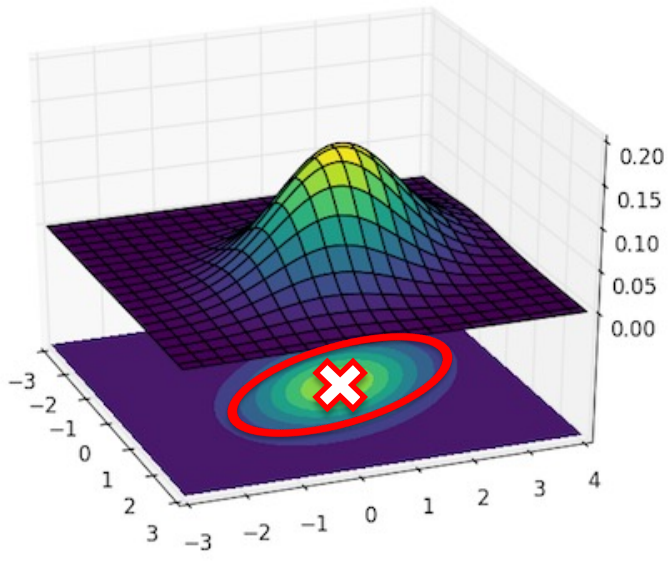


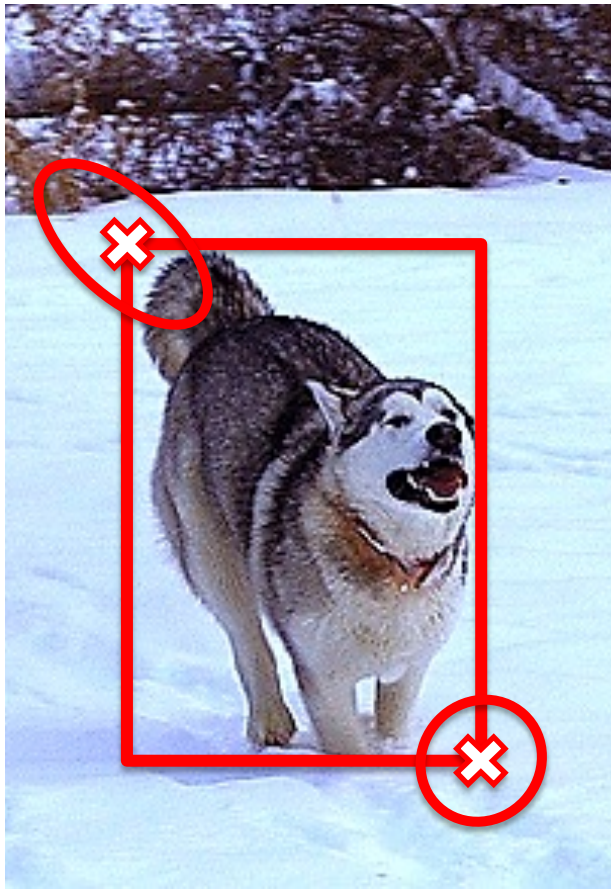
- **Softmax**:

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$$

Background | Regression Predictive Distributions

$$\hat{p}_\theta(y|x) = \mathcal{N}(\mu(x,\theta), \Sigma(x,\theta))$$





Estimation | Regression Predictive Distributions

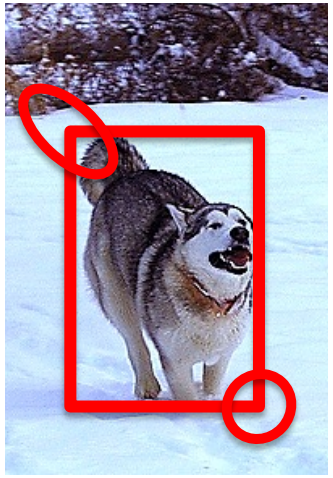
Neural Network

Mean

$\mu(x, \theta)$

variance

$\Sigma(x, \theta)$



Estimation | Regression Predictive Distributions

- $\Sigma(x, \theta)$ needs to be positive semidefinite.

- Two main solutions in literature:
  1. Assume uncorrelated components.

$$\Sigma(x, \theta) = diag(\exp(\mathbf{z})) = \begin{bmatrix} \exp(z_1) & 0 & \dots & 0 \\ 0 & \exp(z_2) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \exp(z_n) \end{bmatrix}$$

  2. Estimate Cholesky decomposition $L(x, \theta)$ such that:

$$\Sigma(x, \theta) = L(x, \theta)L^T(x, \theta)$$

Estimation | Summary (Aleatoric Uncertainty)

- Aleatoric Uncertainty:

$$\mathcal{L}(y_n, \hat{p}_\theta(y|x_n)) = -\log \hat{p}_\theta(y_n|x_n)$$

  - Classification → Softmax probabilities.

$$\hat{p}_\theta(y|x) = Cat(p_1(x,\theta), \ldots, p_K(x,\theta))$$

  - Regression → Variance estimation.

$$\hat{p}_\theta(y|x) = \mathcal{N}(\mu(x,\theta), \Sigma(x,\theta))$$

Estimation | Adding Epistemic Uncertainty

- Epistemic uncertainty → Lack of knowledge on which model generated that data.

$$\hat{p}_{\theta}(y|x) = \hat{p}(y|x,\theta)$$

- True Predictive distribution:

$$\hat{p}(y|x) = \int_{\theta} \hat{p}(y|x,\theta)p(\theta)d\theta$$

Prior

Estimation | Adding Epistemic Uncertainty

- Marginalizing over the parameters:

$$\hat{p}(y|x) = \int_{\theta} \hat{p}(y|x,\theta)p(\theta)d\theta$$

- **Solution 1**: **Variational Inference → Ensemble based solutions**

$$p(\theta) \simeq q_{\gamma}(\theta)$$

- Solution 2: MCMC sampling → Generative based solutions.
  - Too computationally expensive for reasonable robot performance bounds.

- **Approximate Variational Inference**:

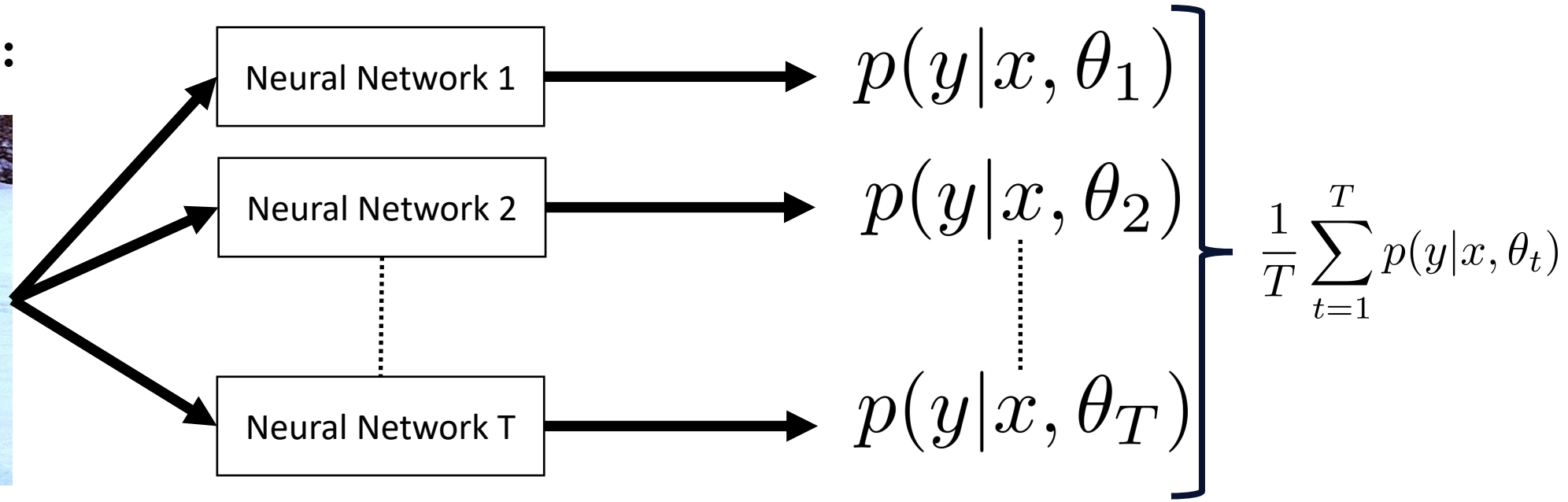
$$\hat{p}(y|x) = \int_{\theta} \hat{p}(y|x,\theta)p(\theta)d\theta$$

$$\approx \frac{1}{T}\sum_{t=1}^{T} p(y|x,\theta_t) \qquad \theta_t \sim q_\gamma(\theta)$$

Estimation | Ensembles

- **Train** T independent ensemble models by:
  1. Randomly shuffling the order of training data observed by each model.
  2. Randomly initializing model parameters **(we already do that!).**

- **Inference:**



- **Classification:** Easy

$$\hat{p}(y|x) = Cat(p_1(x), \ldots, p_K(x)) \mid p_i = \frac{1}{T}\sum_{t=1}^{T} p_{i,t} \quad \forall i = \{1, \ldots, K\}$$

- **Regression:** Assume each one of the T probability distribution a member of a uniformly weighted mixture of gaussians model.

$$\hat{p}(y|x) = \mathcal{N}(\mu(x), \Sigma(x))$$

$$\mu(x) = \frac{1}{T}\sum_{t=1}^{T} \mu(x, \theta_t)$$

$$\Sigma(x) = \frac{1}{T}\left(\sum_{t=1}^{T} \mu(x, \theta_t) f(x, \theta_t)^\intercal\right) - \mu(x)\mu(x)^\intercal + \frac{1}{T}\sum_{t=1}^{T} \Sigma(x, \theta_t)$$

Estimation | Ensembles

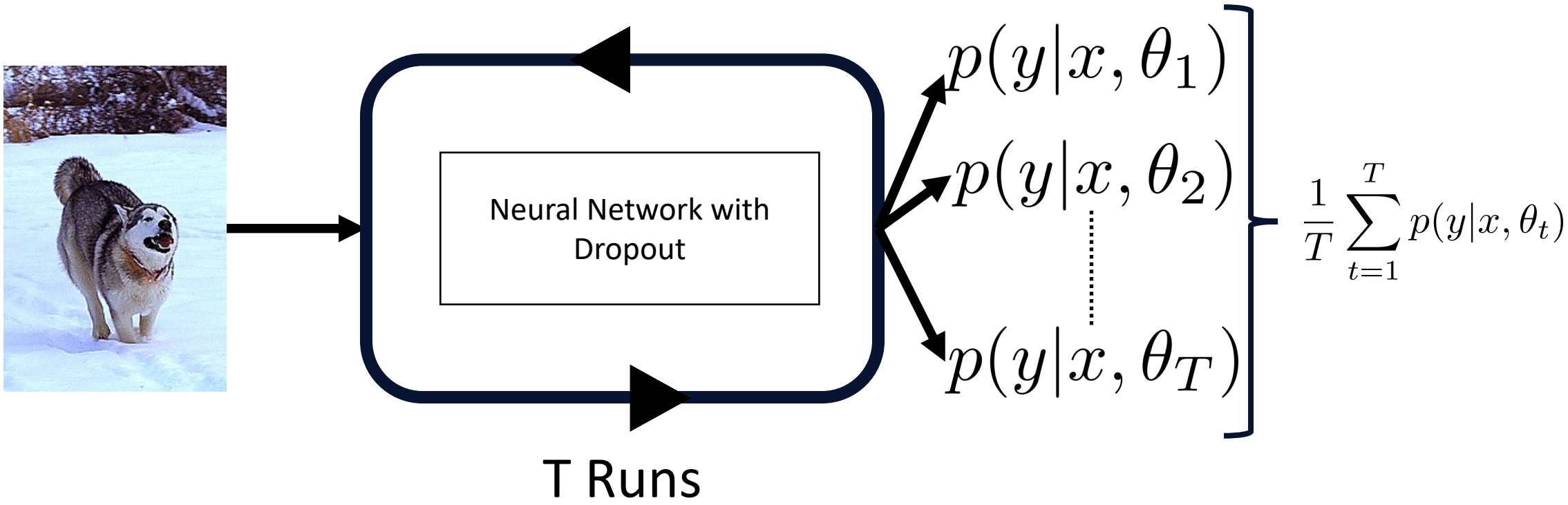
- **Pros:**
  - Extremely trivial to implement.
  - “Good enough” estimates that can be used for many application.

- **Cons:**
  - O(n) **runtime and memory**.
  - Some applications are shown to not benefit from ensembles (Object detection for example).

Estimation | Monte-Carlo Dropout

- **Train** a **single** model with dropout enabled.

- Run the **network T times with the same input** during inference while **keeping dropout enabled**.



Estimation | Monte-Carlo Dropout

- **Pros:**
  - Extremely trivial to implement.
  - Approximation of exponentially many ensembles with O(1) memory!

- **Cons:**
  - O(n) **runtime**.
  - Worse than ensembles on uncertainty estimation task.
  - Most of the time, results in worse prediction accuracy when compared to deterministic networks.

Summary | Epistemic Uncertainty

- Usually too computationally expensive for deployment on robots.

- If needed, use ensembles.

Predictive Uncertainty: Evaluation

Evaluation | Quality of Predictive Distributions

- **Meaningful predictive probability distributions:**
  - Low uncertainty (high confidence), when the predictor makes small/no mistakes.
  - High uncertainty (low confidence), when the predictor makes large mistakes.

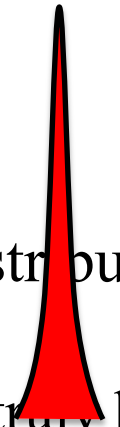
- Overconfident incorrect predictions can lead to non-optimal decision making in planning tasks.

- Underconfident correct predictions can lead to under-utilizing information in fusion operations.

Evaluation | Quality of Predictive Distributions

- The goal of probabilistic forecasting is to maximize the **sharpness** of the predictive distributions **subject to calibration**. [1]

- **Sharpness:** the concentration of the predictive distribution.
  - **Example:** Univariate Gaussians



- **Calibration:** the st ical consistency between the predictive distr utions and the ground truth
  - **Example:** 100 objects to be cars with 0.7 probability → 70 should truly be cars.

[1] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. "Probabilistic forecasts, calibration and sharpness." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69, no. 2 (2007): 243-268.

Evaluation | Quality of Predictive Distributions

- Given:

  $\hat{p}_\theta(y|x_n)$

  $y_i \sim p(y|x_n)$

- Scoring Rule $S(\hat{p}_\theta(y|x_n), y_n) \in \mathbb{R}$.

- A scoring rule is **proper if** its minimum value is achieved **only if** $\hat{p}_\theta(y|x_n) = p(y|x_n)$

- Proper scoring rules:
  - Measure both calibration and sharpness of a predictive distribution.
  - Useful as a **minimization objective for training**, as well as an **evaluation metric**!

Evaluation | Common Proper Scoring Rules

- **Negative log likelihood:**
  - Classification:

$$\mathcal{L}_{\text{NLL}}(\hat{p}_\theta(y|x_n), y_n) = \sum_{k=1}^{K} -y_{nk} \log p_k(x_n, \theta).$$

  - Regression:

$$\mathcal{L}_{\text{NLL}}(\hat{p}_\theta(y|x_n), y_n)) = \frac{1}{2N} \sum_{n=1}^{N} \underbrace{(y_n - \mu(x_n, \theta))^\intercal \Sigma(x_n, \theta)^{-1} (y_n - \mu(x_n, \theta))}_{\text{Squared mahalanobis distance}} + \underbrace{\log \det \Sigma(x_n, \theta)}_{\text{regularizer}}.$$

- NLL can suffer from many issues when used for training and evaluation.
- **Brier score**, continuous ranked probability score (**CPRS**), or the **Energy score** as alternatives.

Evaluation | Summary

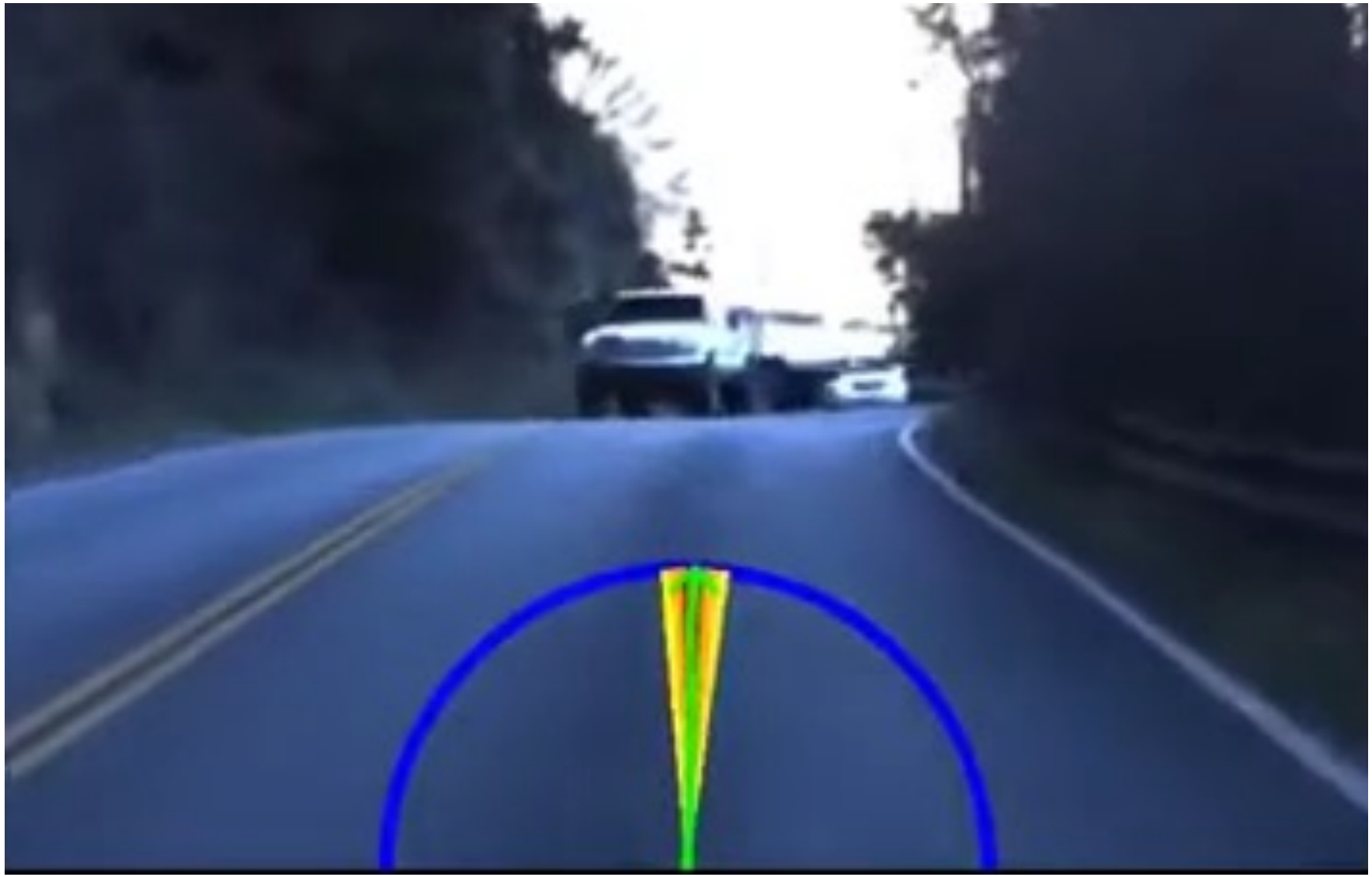
- Evaluation → Proper scoring rules.

- NLL is the most common proper score used to train predictive distribution estimators.

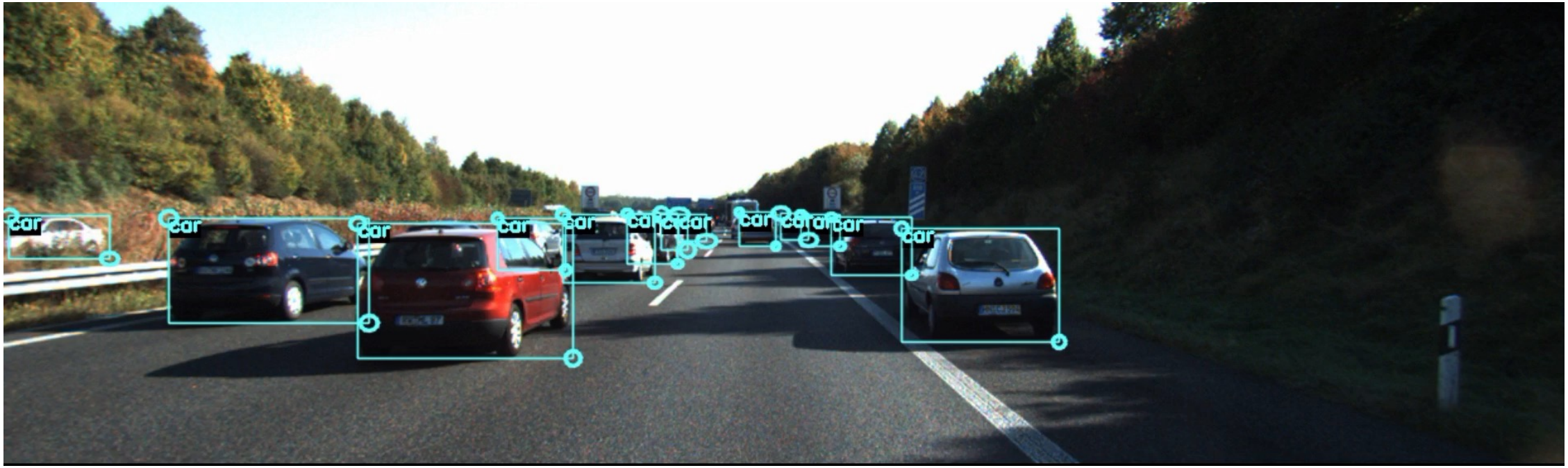
- Other metrics exists such as **Calibration errors** and **Uncertainty Errors**. Those are not proper so be careful if you use them.

Conclusion

Conclusion | Steering Angle Prediction



Conclusion | Object Detection



Conclusion | Challenges and Outlook

- Questionable decomposition of epistemic/aleatoric uncertainty.
  - Variance networks have been shown to be able to estimate epistemic uncertainty when used on their own.
  - Dropout has been shown to be able to “kind of” capture aleatoric uncertainty.

- Scalability of epistemic uncertainty algorithms is currently a substantial bottle neck for usage in robotics.

- Can we get ground truth uncertainty from simulators / multiple human labels?

Conclusion | Resources

- **Three** papers to read → 70% of information required to begin using uncertainty estimation.

1. **Eyke Hüllermeier and Willem Waegeman**. "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods." *Machine Learning* 110.3 (2021): 457-506.
2. **Alex Kendall and Yarin Gal**. "What uncertainties do we need in bayesian deep learning for computer vision?." *(NeurIPS 2017)***.**
3. **Tilmann Gneiting and Adrian E. Raftery.** "Strictly proper scoring rules, prediction, and estimation." *Journal of the American statistical Association* 102.477 (2007): 359-378.