Robust and Trustworthy Deep Learning

THEMIS AI

January 11, 2023

Themis AI

Design, Advance, and Deploy Safe and Trustworthy AI

NEURAL INFORMATION PROCESSING SYSTEMS

ICML
International Conference On Machine Learning

BBC NEWS

TC

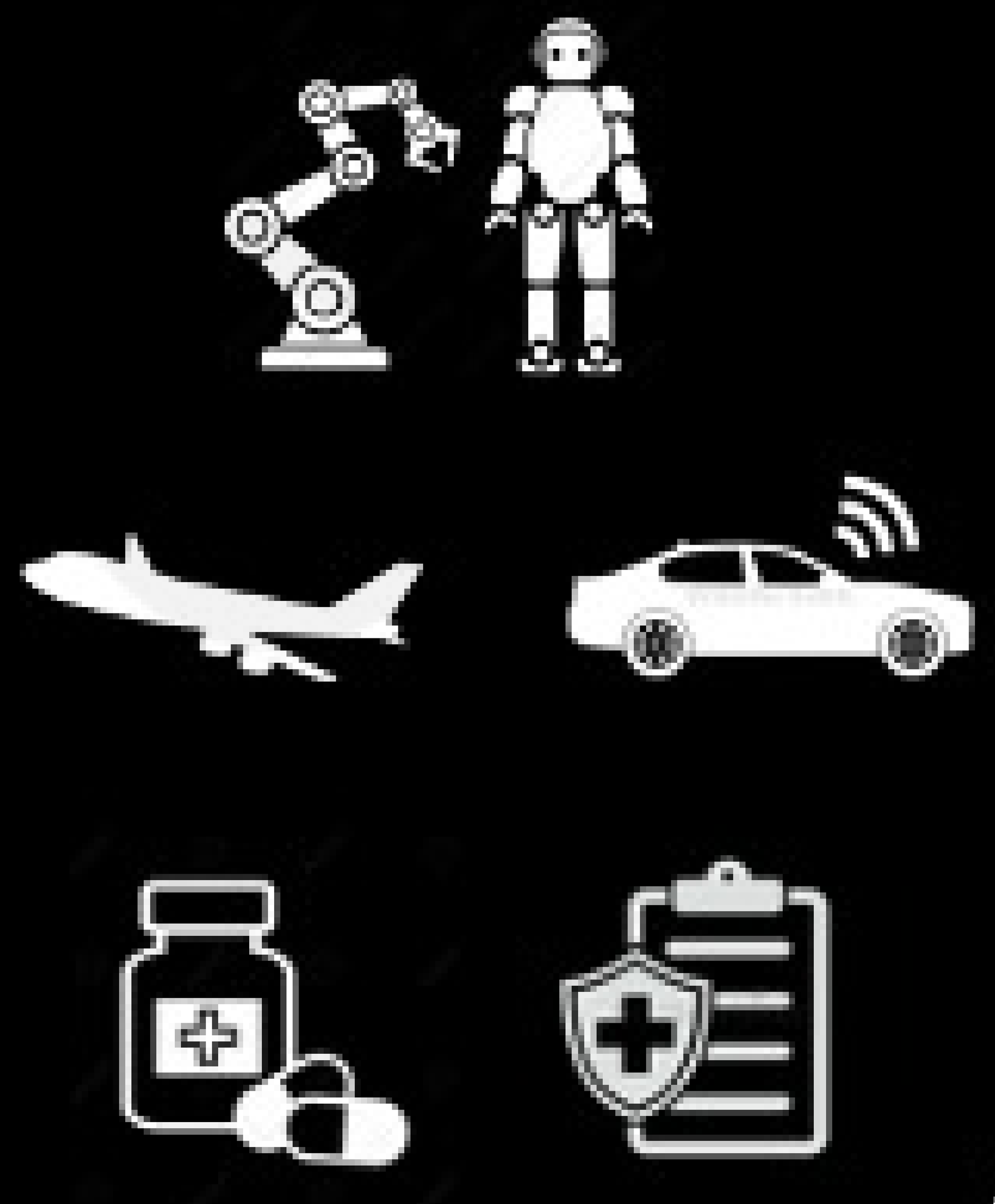
WIRED

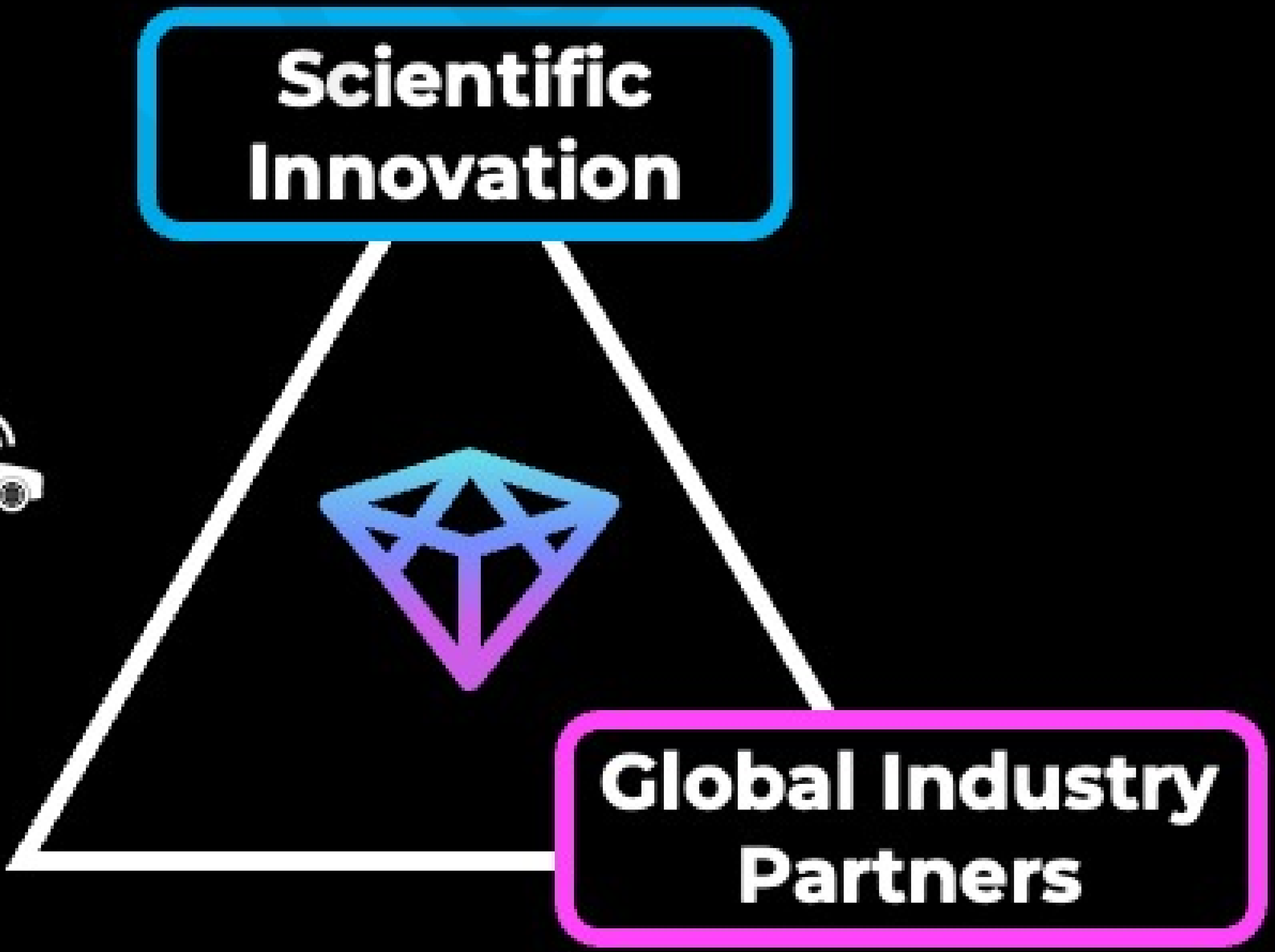
Daily Mail

Scientific Innovation

Themis AI

Design, Advance, and Deploy Safe and Trustworthy AI





THEMIS AI

Design, Advance, and Deploy Safe and Trustworthy AI

GitHub

CAPSA

Scientific Innovation

Large-scale Platforms

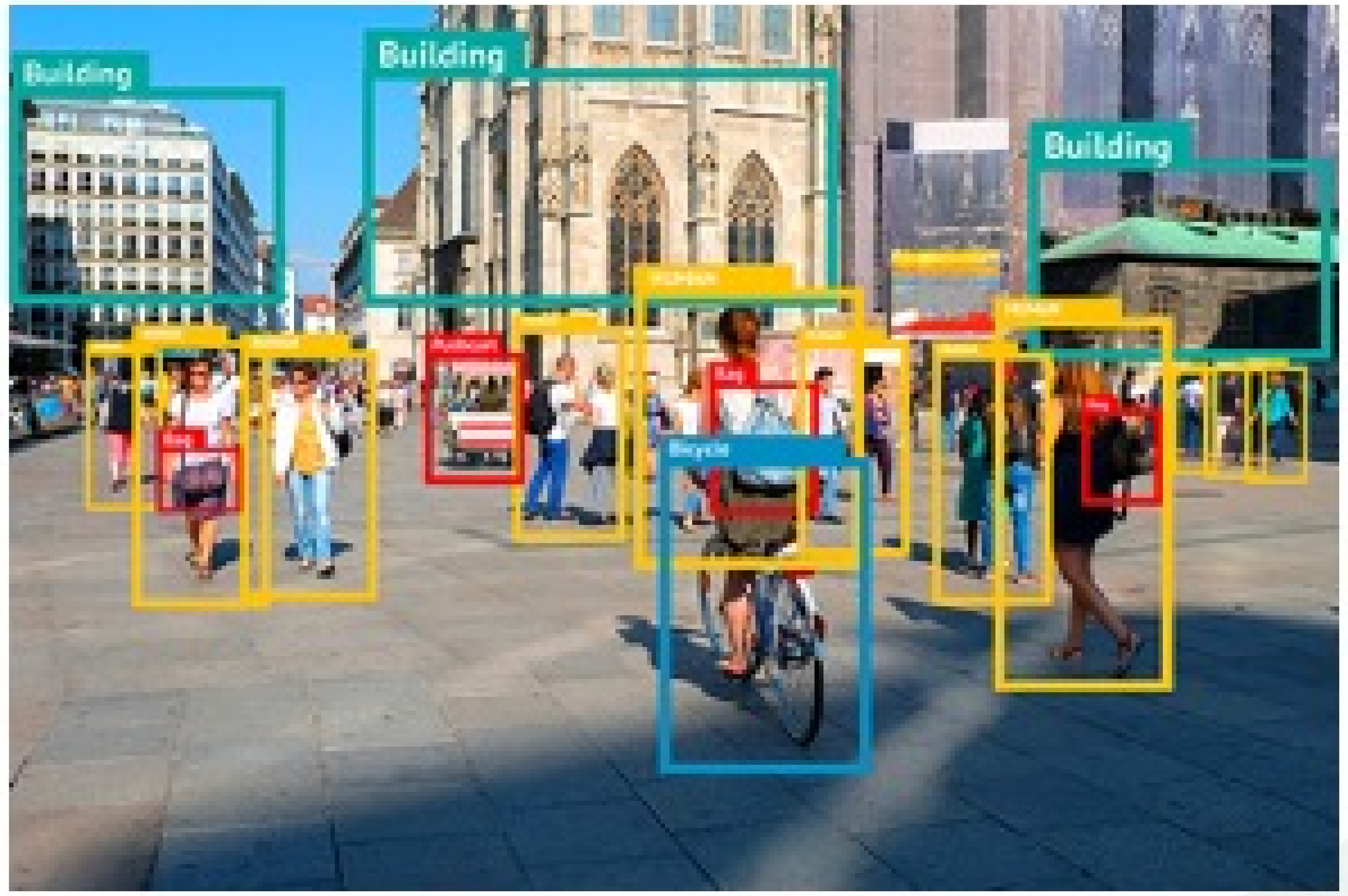
Global Industry Partners

Robust and Trustworthy Deep Learning

Sadhana Lolla
Machine Learning Scientist
Themis AI

AI in Safety-Critical Domains

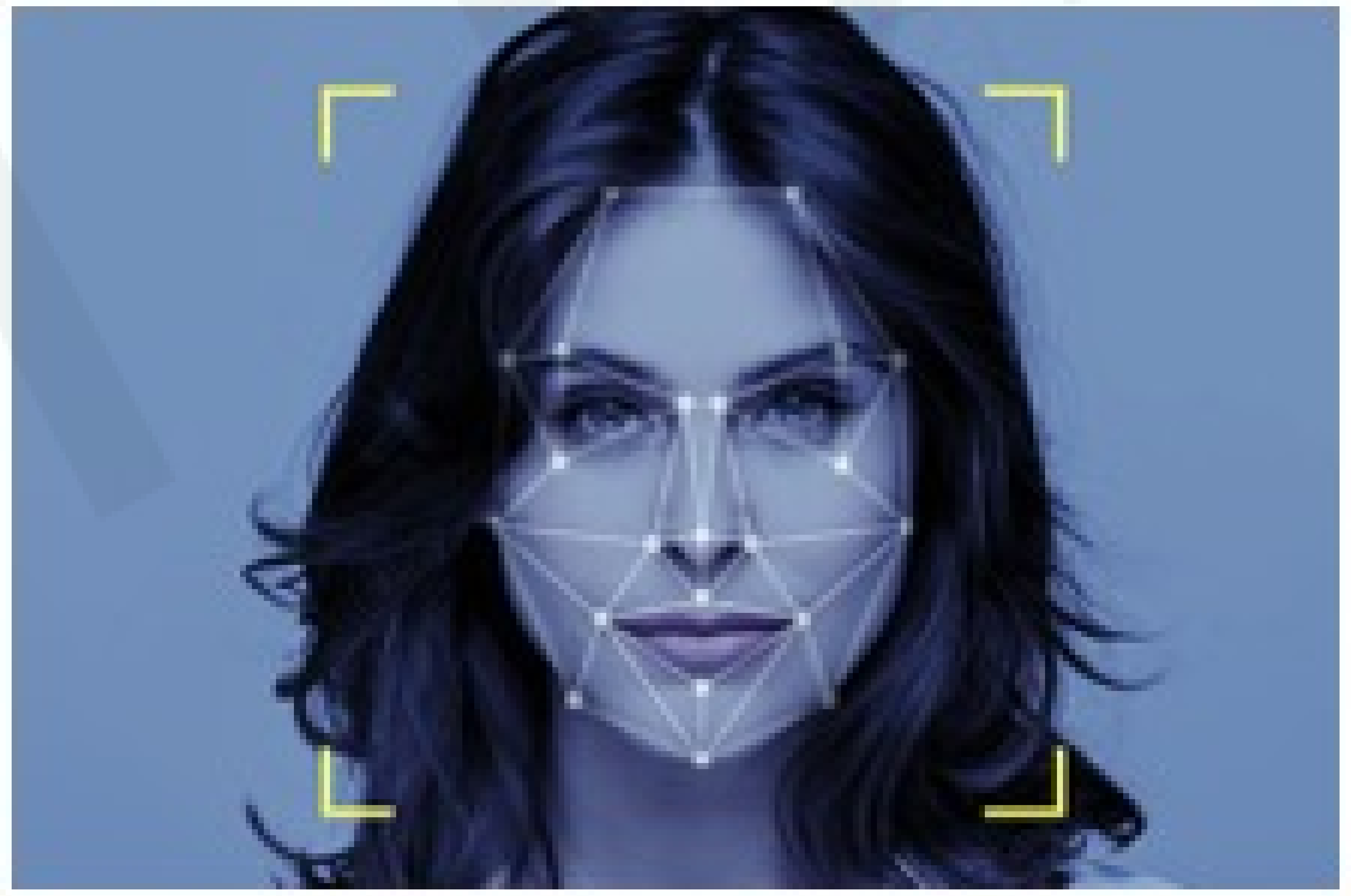
Scene planning



Robot-assisted surgery

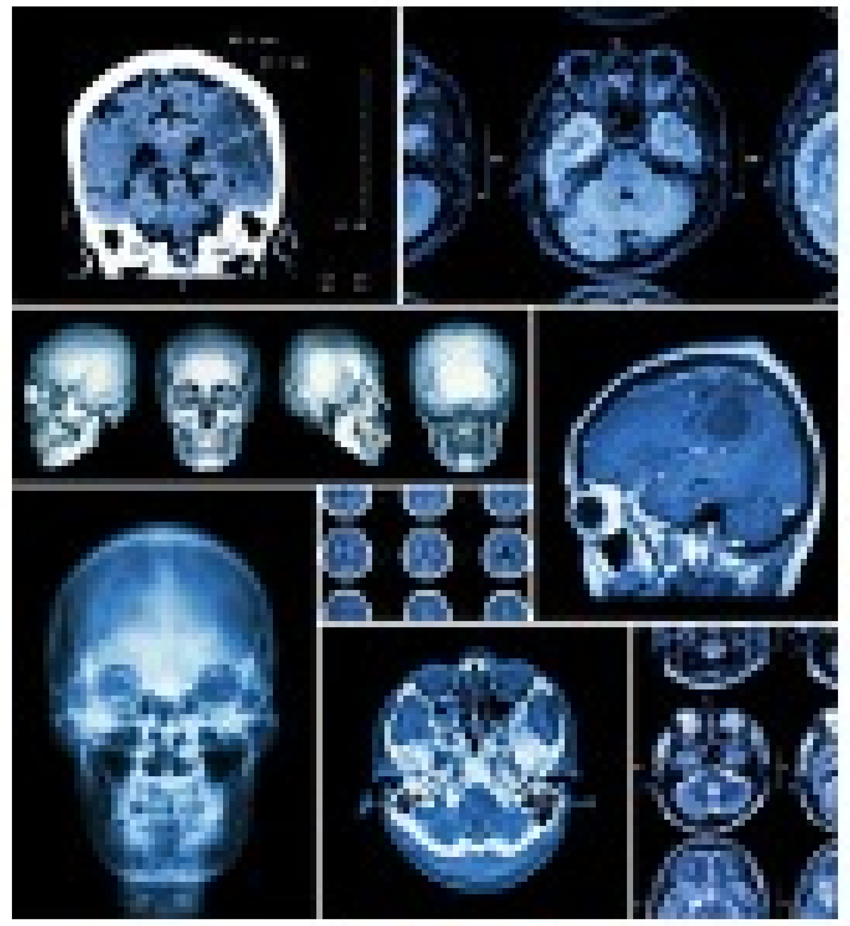
Drug discovery

Autonomous vehicles



Facial recognition

Robotics



Diagnostics

AI in the News

Millions of black people affected by racial bias in health-care algorithms

Artificial Intelligence has a gender bias problem – just ask Siri

GM's Cruise Recalls Self-Driving Software Involved in June Crash

Microsoft Plans to Eliminate Face Analysis Tools in Push for 'Responsible A.I.'

The New Chatbots Could Change the World. Can You Trust Them?

Tesla 'full self-driving' triggered an eight-car crash, a driver tells police

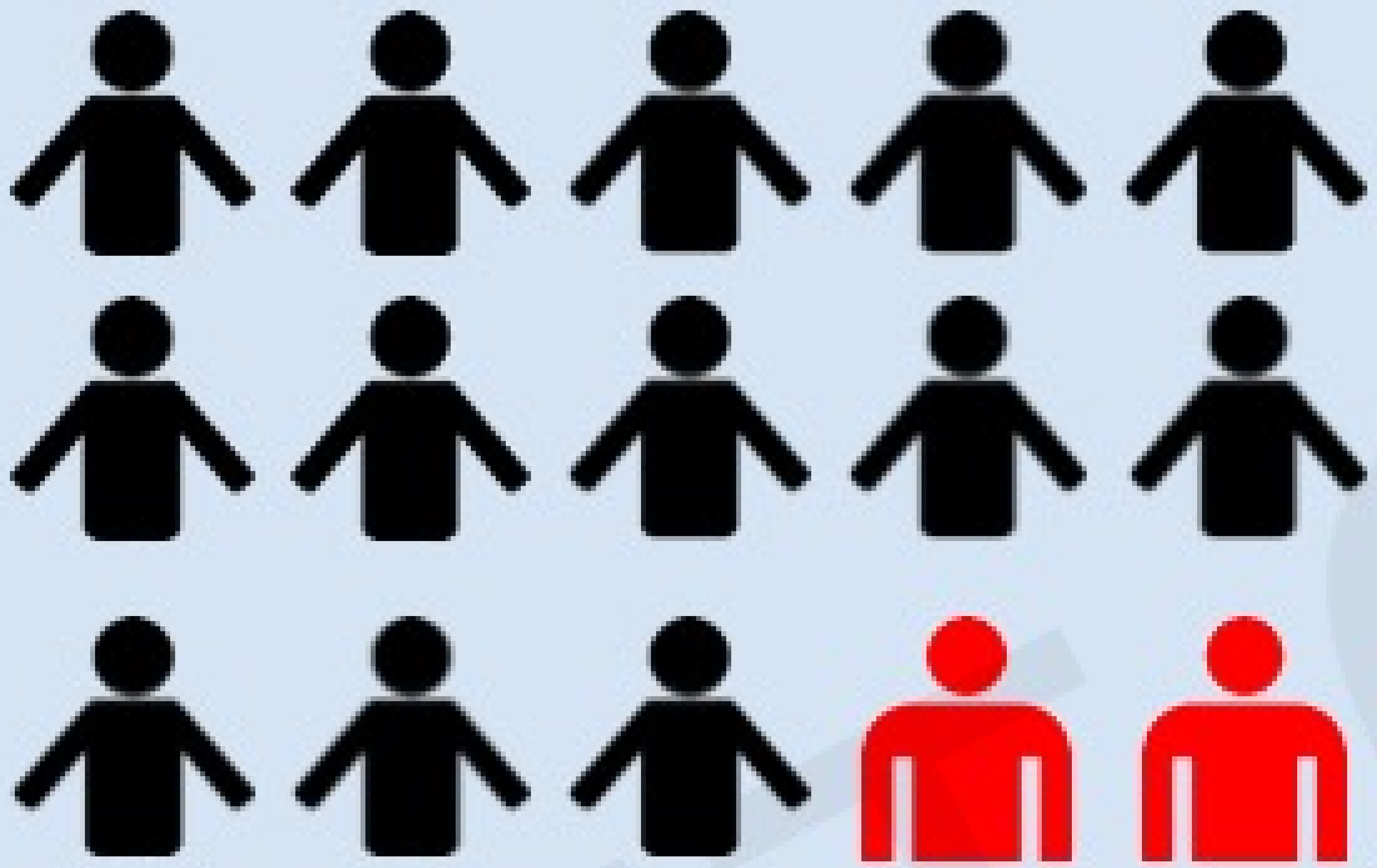
Minority Patients Often Left Behind By Health AI

Many Facial-Recognition Systems Are Biased, Says U.S. Study

Risks Rise As Robotic Surgery Goes Mainstream

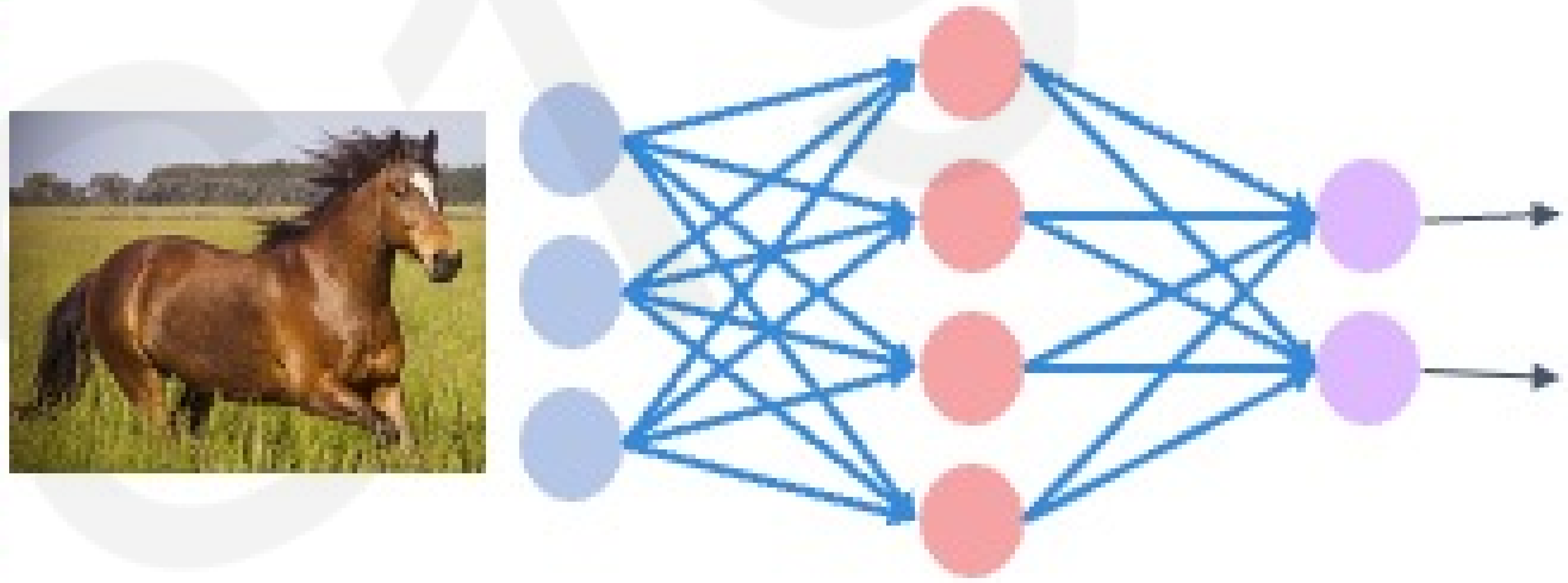
Challenges for Robust Deep Learning

Bias



What happens when models are skewed by sensitive feature inputs?

Uncertainty



Can we teach a model to recognize when it doesn't know the answer?

Bias in the AI Lifecycle

Industry Example: Facial Detection

Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	94.0%	79.2%	100%	98.3%	20.8%
FACE++	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	88.0%	65.3%	99.7%	92.9%	34.4%

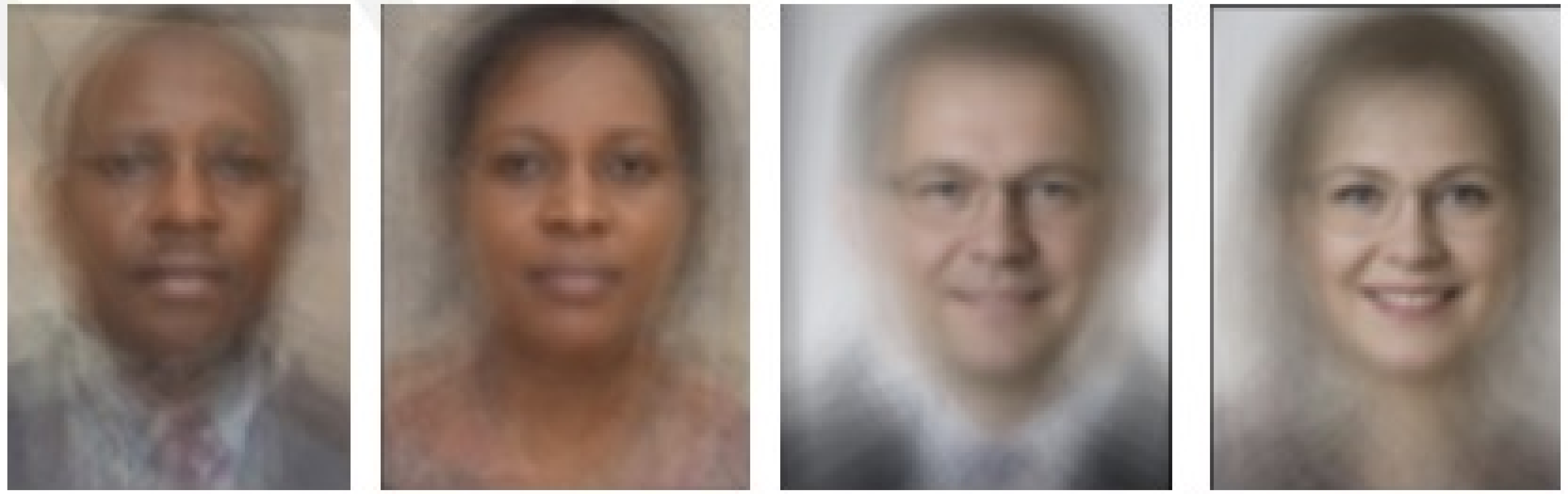
What types of bias were present in these models?

- Selection bias: proportion of data in dataset does not reflect the real world
- Evaluation bias: originally, these models were not evaluated on subgroups!

Industry Example: Facial Detection

Gender Classifier	Darker Male	Darker Female	Lighter Male	Lighter Female	Largest Gap
Microsoft	94.0%	79.2%	100%	98.3%	20.8%
FACE++	99.3%	65.5%	99.2%	94.0%	33.8%
IBM	88.0%	65.3%	99.7%	92.9%	34.4%

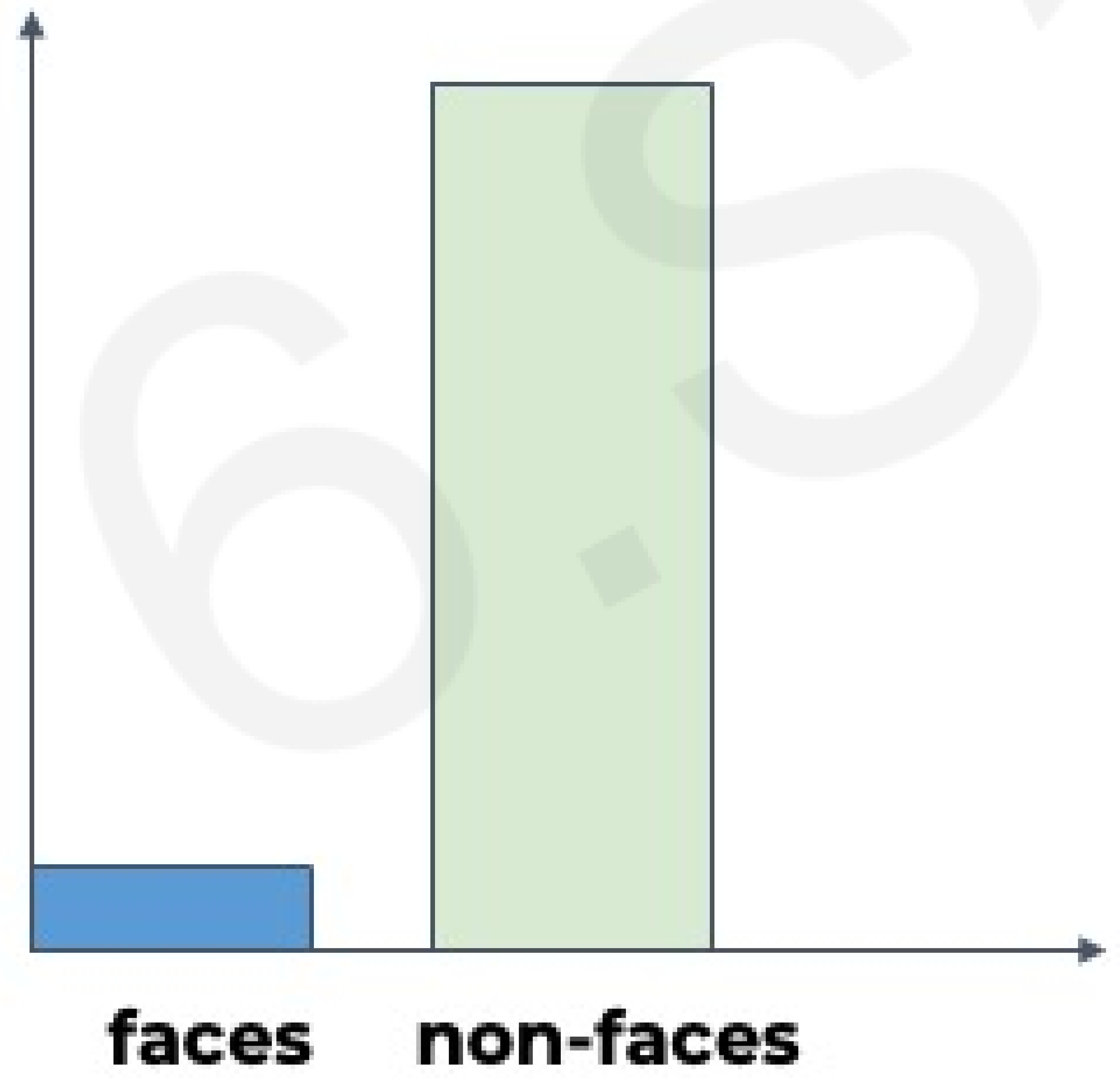
Pilot Parliaments Benchmark: a dataset designed to uncover biases by balancing race and gender



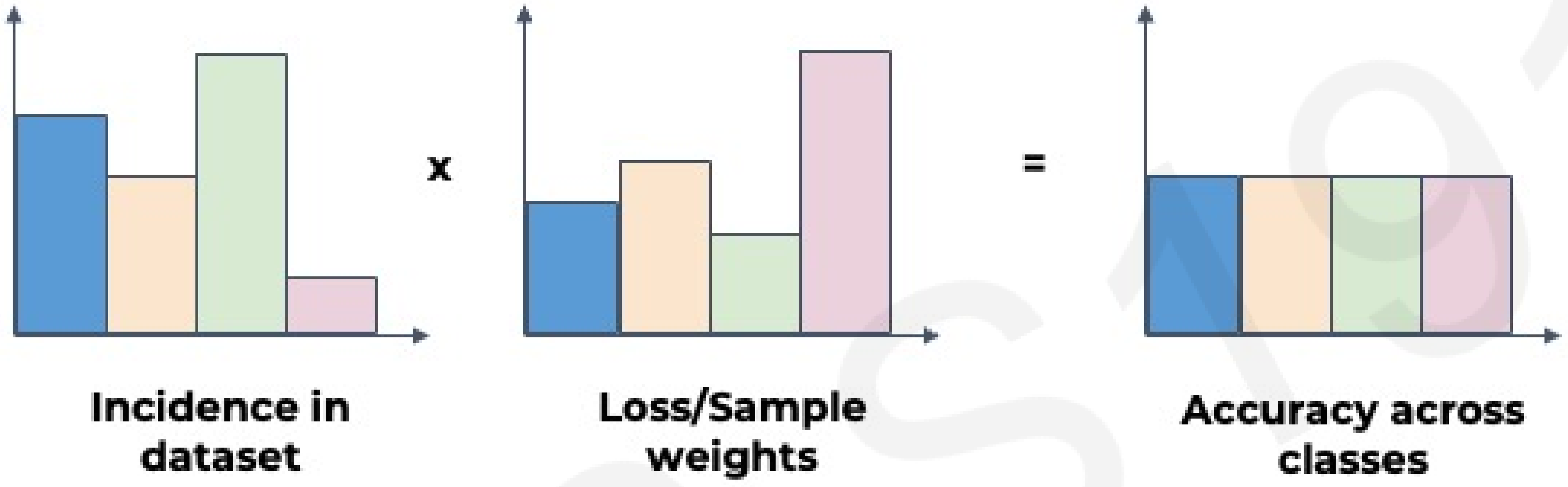
Class Imbalance

What happens when some classes are more represented than others?

Frequency of classes in dataset



Mitigating Class Imbalance



- **Sample Reweighting:** Sample more data points from underrepresented classes
- **Loss Reweighting:** Mistakes on underrepresented classes contribute more to loss
- **Batch Selection:** Choose randomly from classes so that every batch has an equal number of points per class

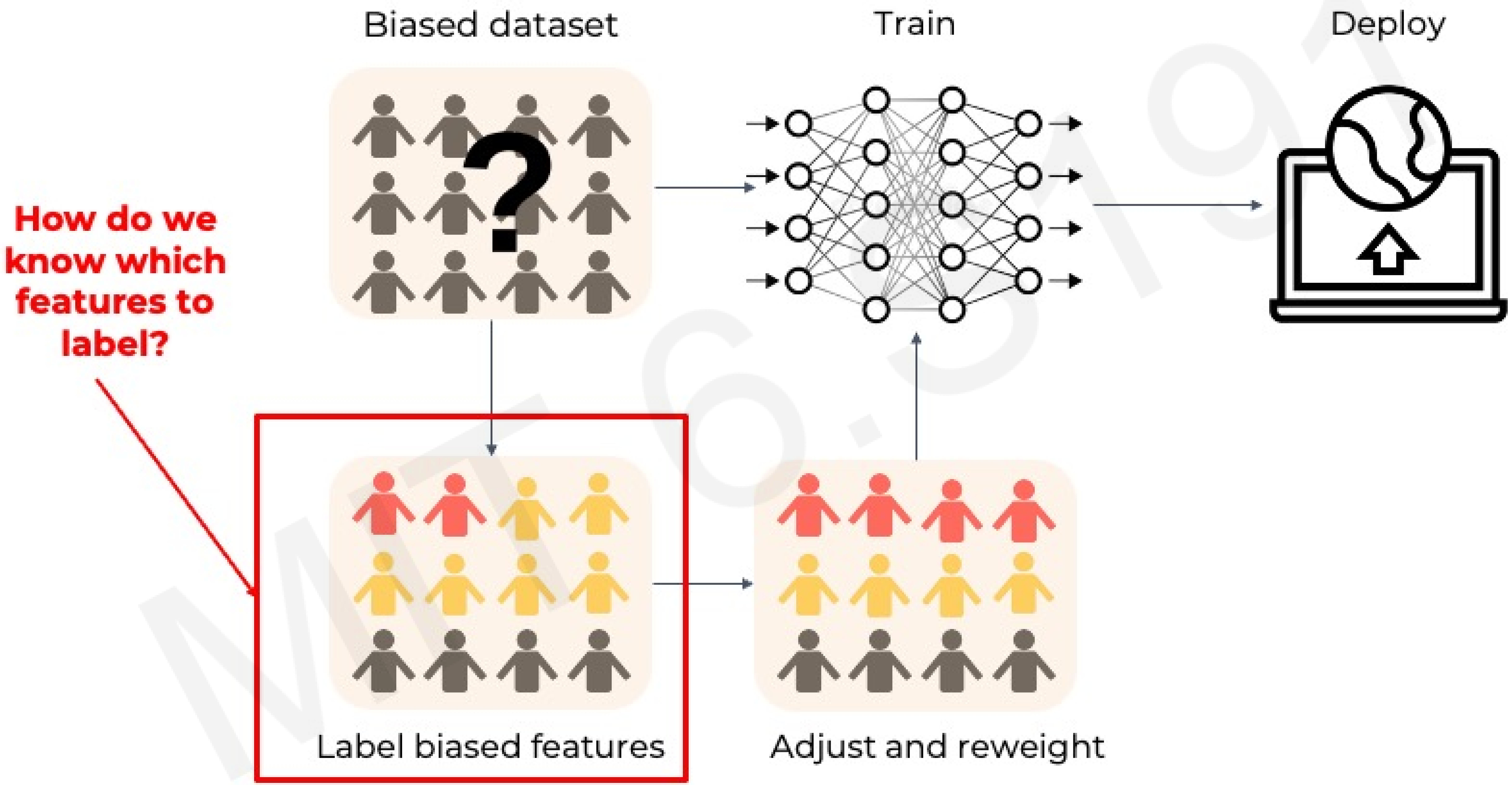
What about *latent features?*

Variations **within the same class** are important to capture while debiasing; otherwise we may overgeneralize!

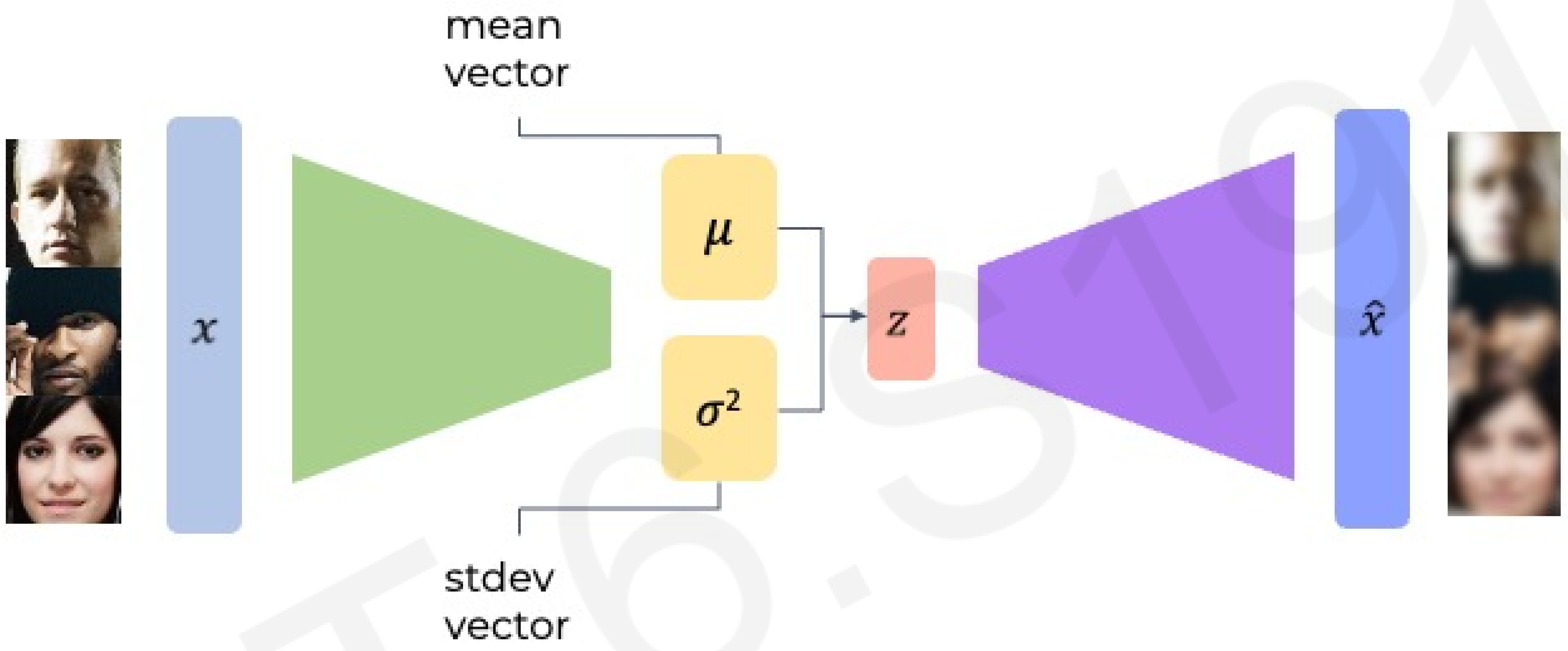
What are some latent features in the above dataset? Which ones may be underrepresented?

Recall from lab 2 and lecture 4!

Why is debiasing latent features difficult?



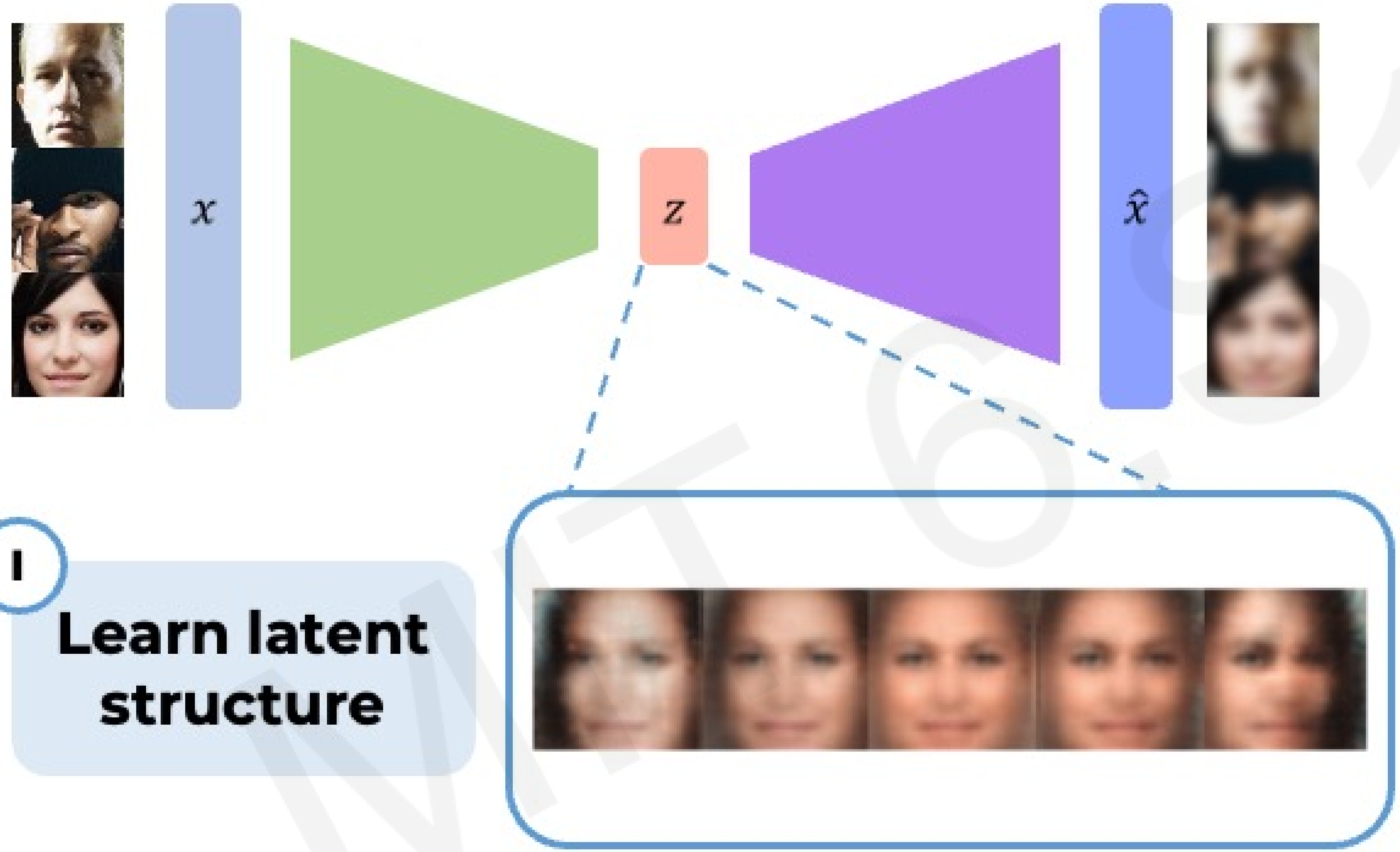
VAE Recap



Variational autoencoders (VAEs) are a probabilistic twist on autoencoders!

Recall from lab 2 and lecture 4!

Mitigating Bias Through Learned Latent Structure

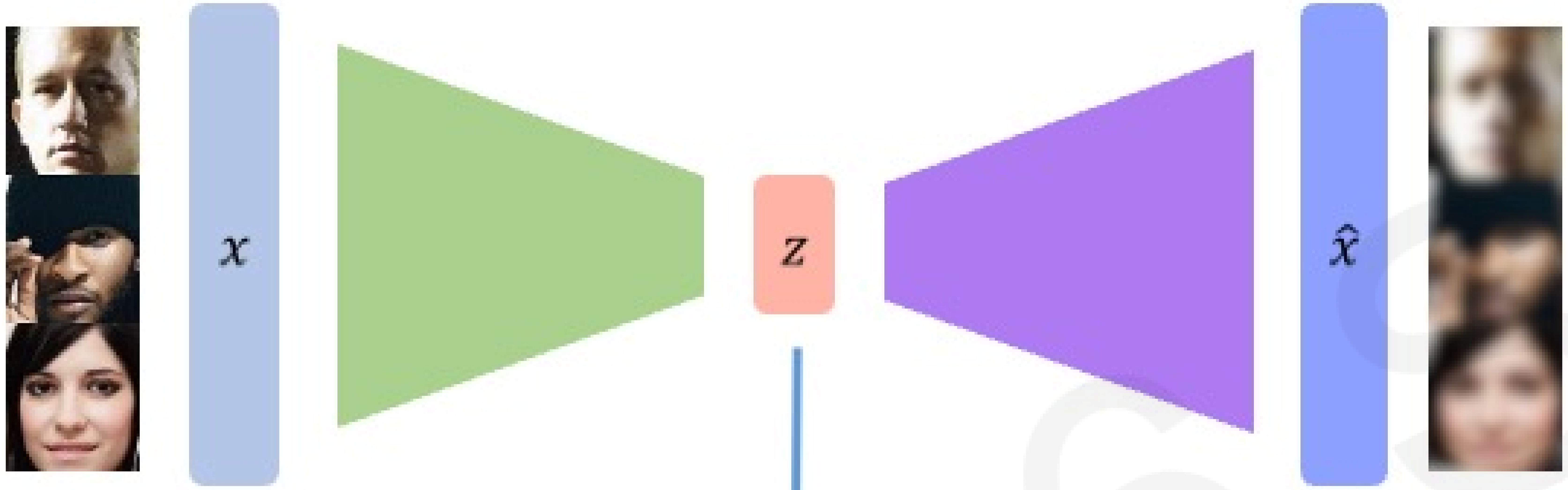


1

Learn latent structure

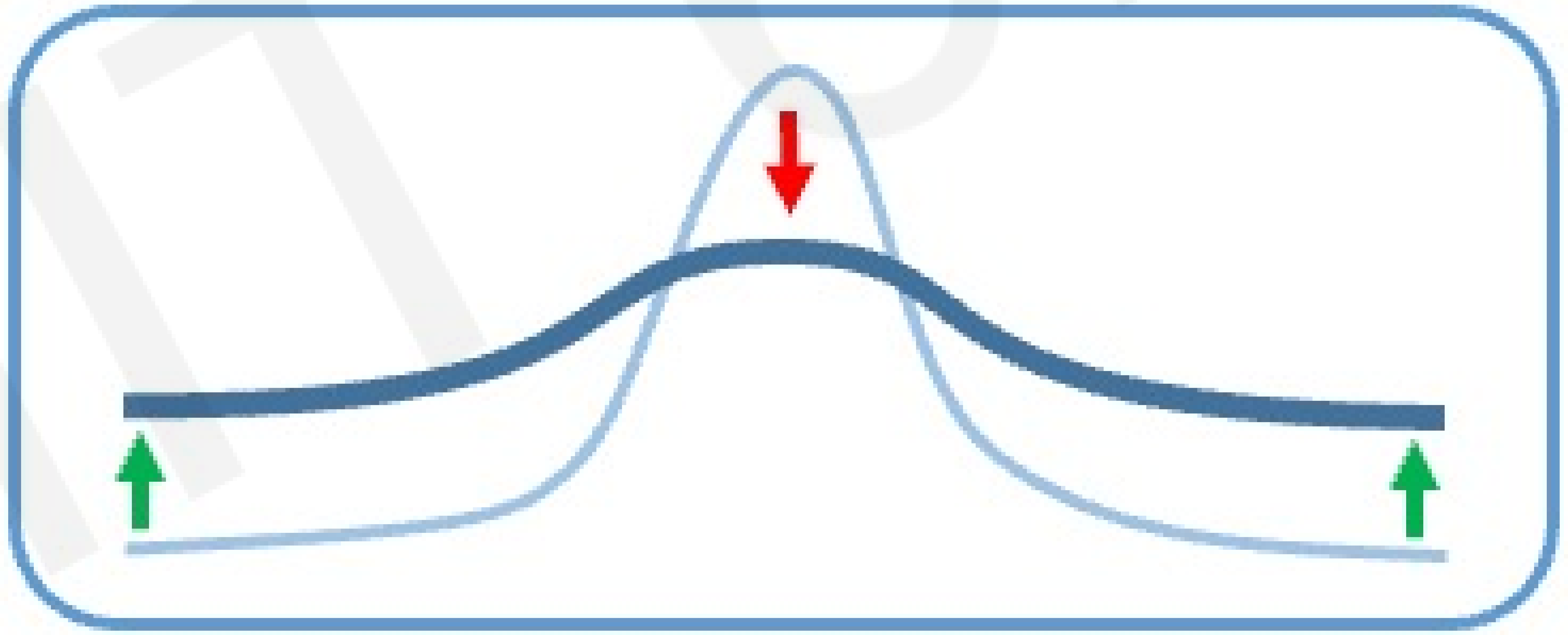
Mitigating Bias Through Learned Latent Structure

Mitigating Bias Through Learned Latent Structure

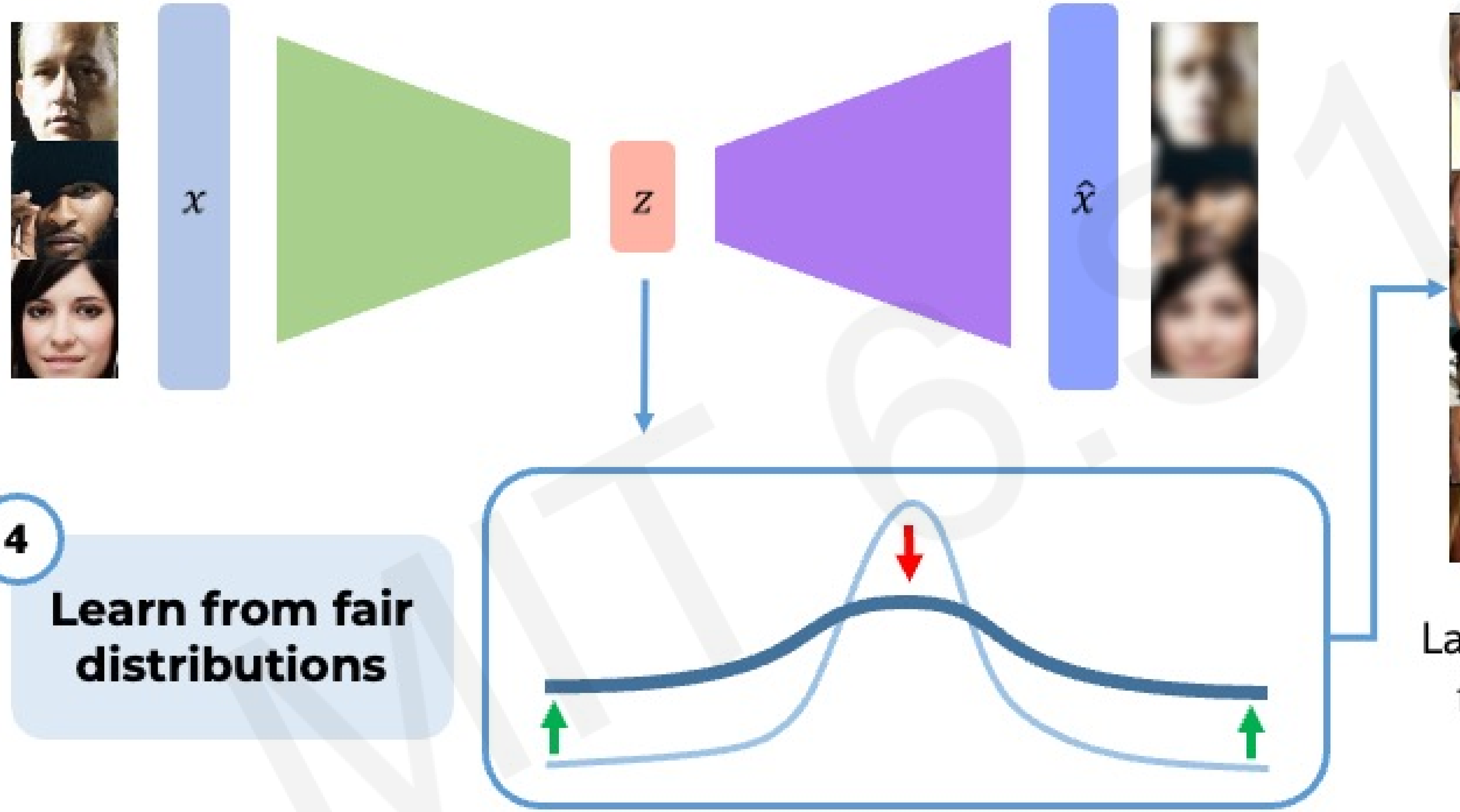


3

Adaptively guide learning



Mitigating Bias Through Learned Latent Structure



4

Learn from fair distributions

Latent distributions used to create fair and representative dataset

Using Latent Variables for Automated Debiasing

Approximate the distribution of the latent space with a joint histogram over the latent variables:

Important for Lab 3!

$$\underbrace{\hat{Q}(z|X)}_{\text{Estimated joint distribution}} \propto \prod_i \underbrace{\hat{Q}_i(z_i|X)}_{\text{Histogram for every latent variable } z_i}$$

Independence to approximate

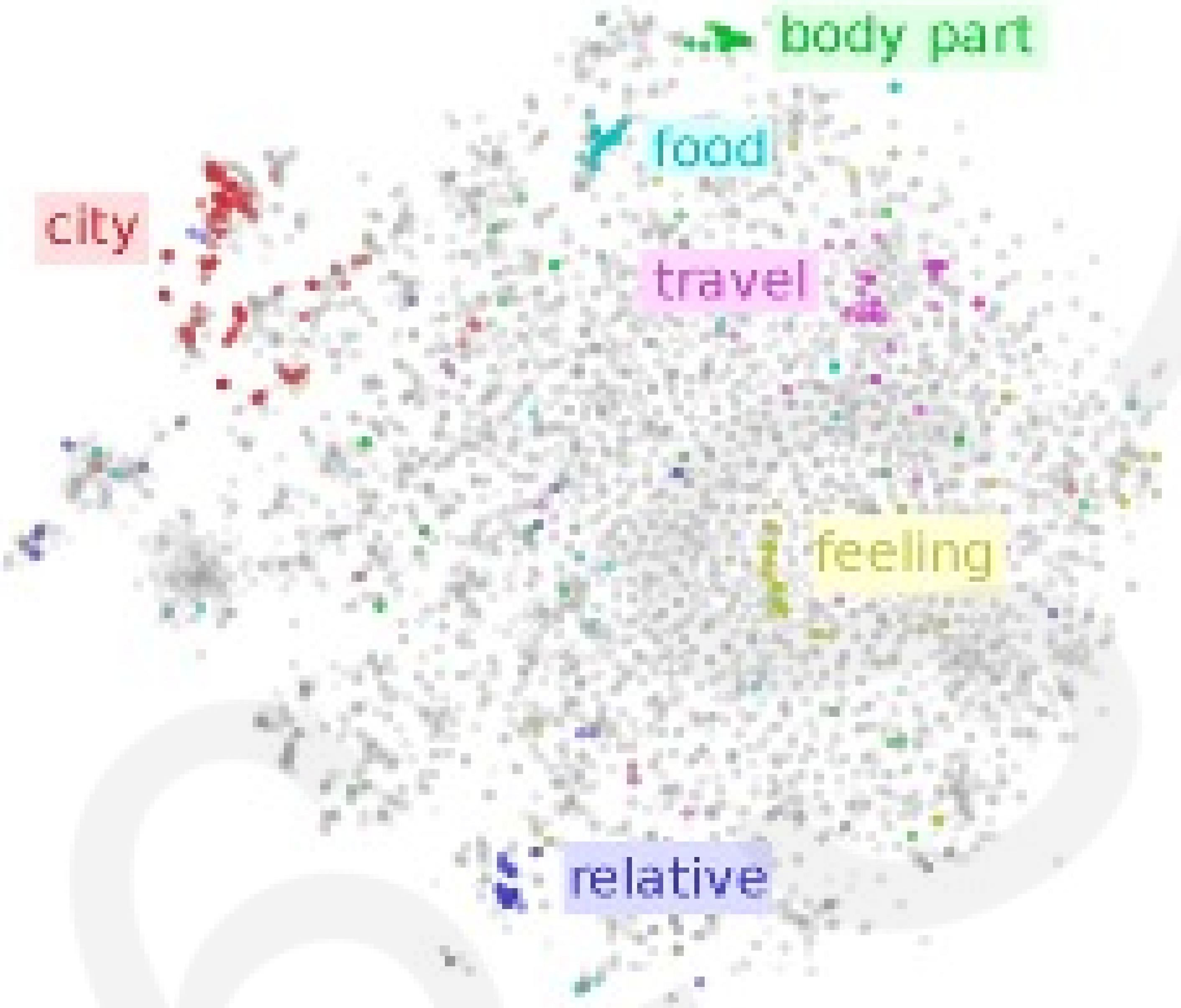
$$\underbrace{W(z(x)|X)}_{\text{Probability of selecting datapoint}} \propto \prod_i \frac{1}{\underbrace{\hat{Q}_i(z_i(x)|X)}_{\text{Histogram for every latent variable } z_i} + \underbrace{\alpha}_{\text{Debiasing parameter}}}$$

Evaluation: Decreased Categorical Bias

Other examples of real-world bias

Autonomous Driving
sunny, straight roads vs. adverse conditions

[Amini et al, *IROS* 2018]



Language Modeling
Encodes gender biases

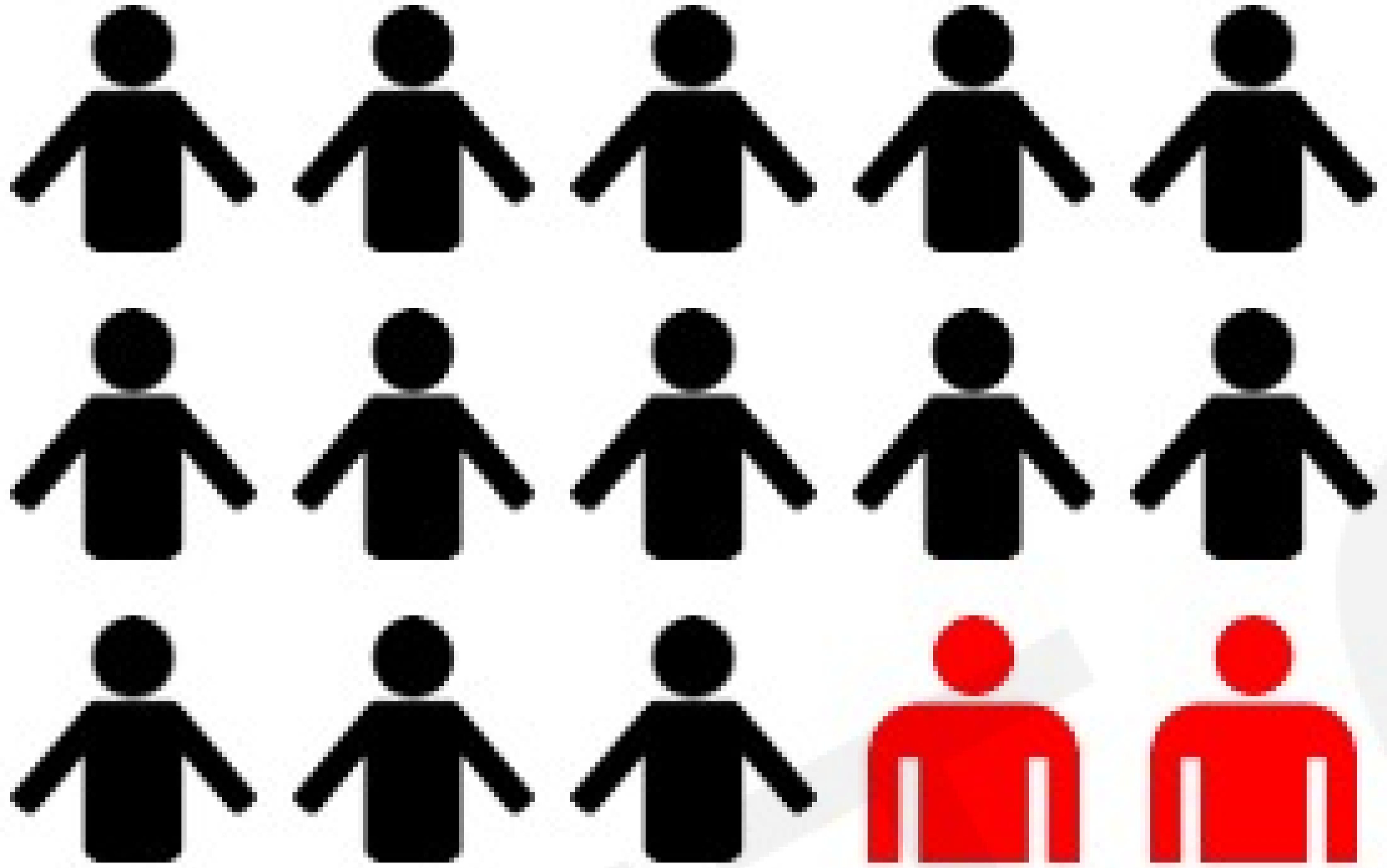
[Caliskan et al, *Science* 2017]

Healthcare Recommendation Algorithms
Encodes racial biases

[Obermeyer et al, *Science* 2019]

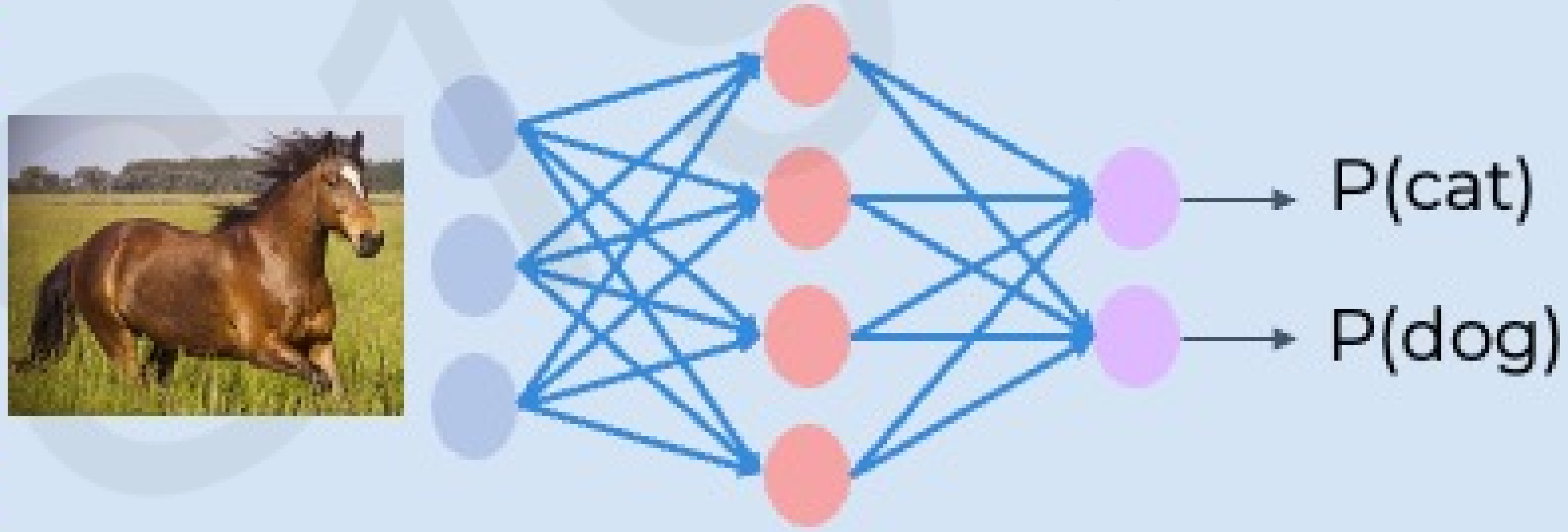
Challenges for Robust Deep Learning

Bias



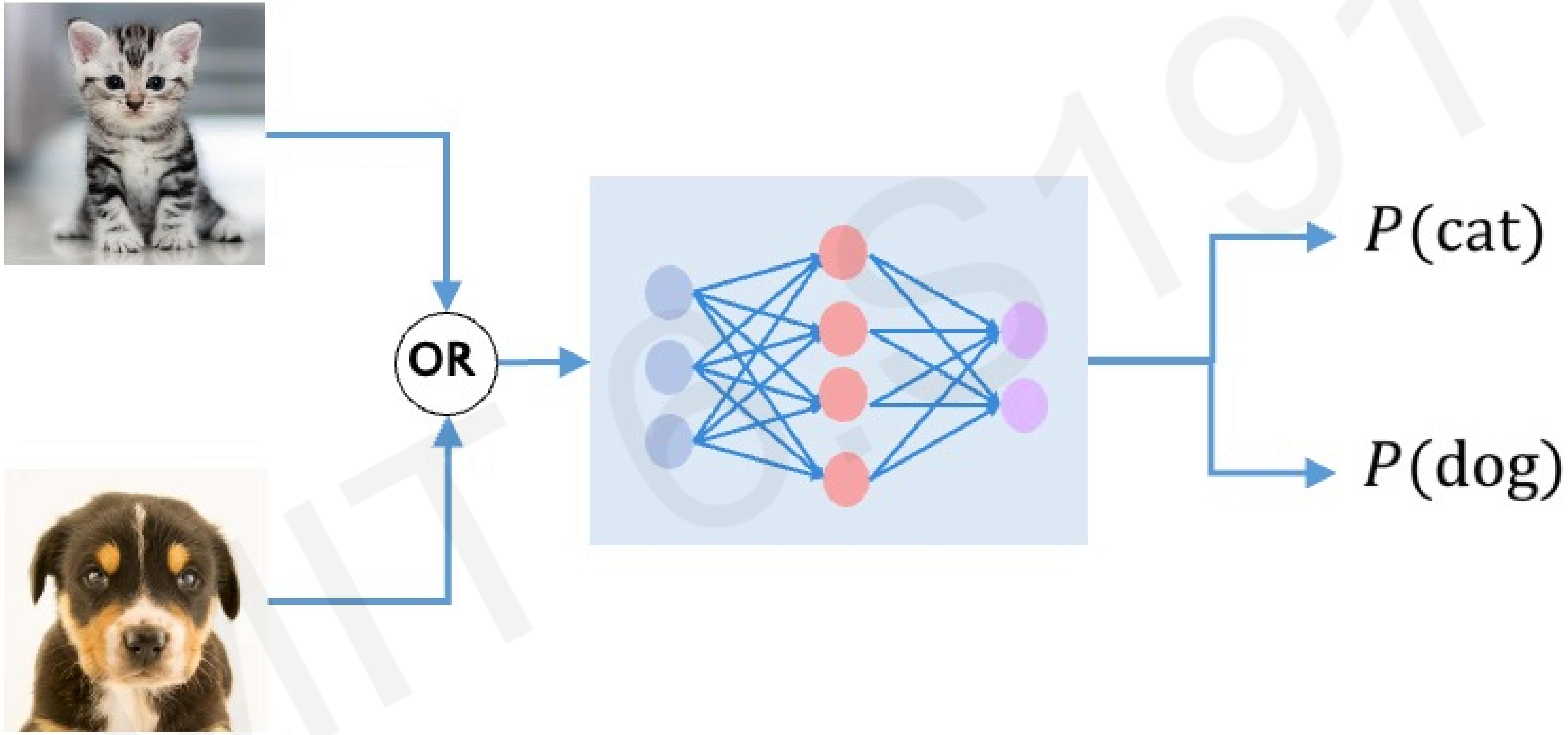
What happens when models are skewed by sensitive feature inputs?

Uncertainty



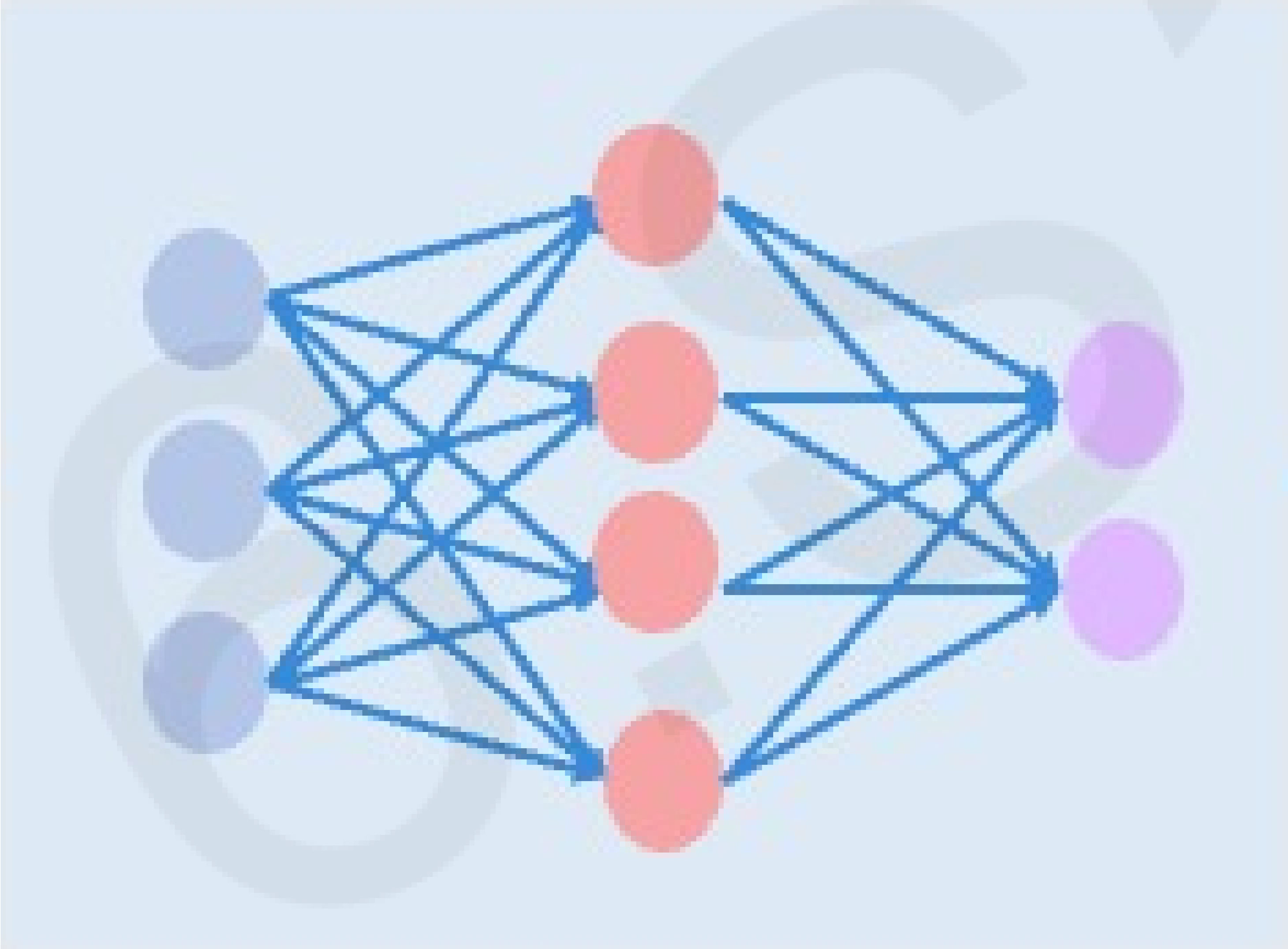
Can we teach a model to recognize when it doesn't know the answer?

What is uncertainty?



...not to be confused with likelihood

Models output a **probability distribution** regardless of input; however, this is not a confidence score!



$P(\text{cat}) = 0.2$

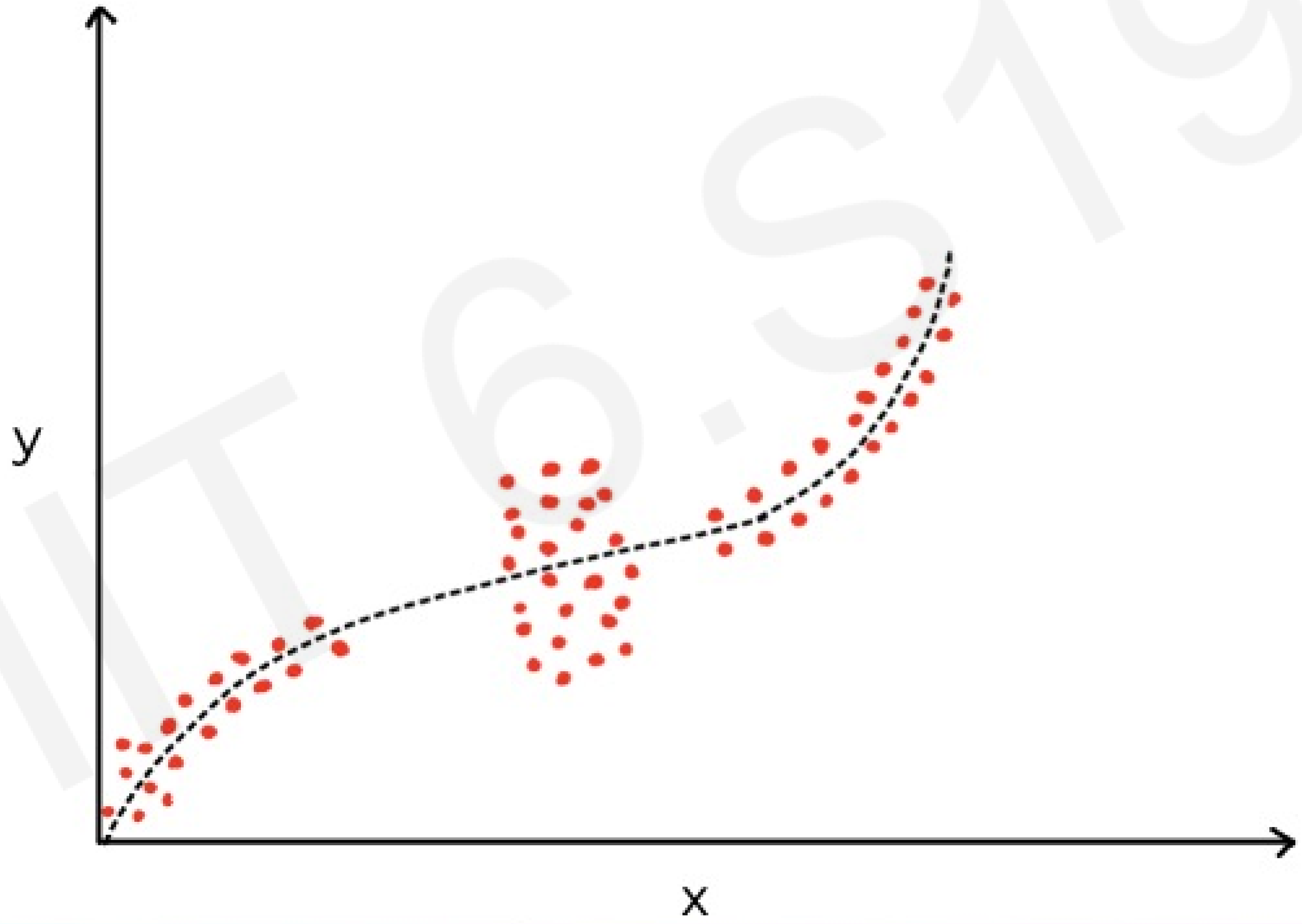
$P(\text{dog}) = 0.8$

Uncertainty estimation gives us a measure of **confidence** in the prediction

To mitigate scenarios like this:

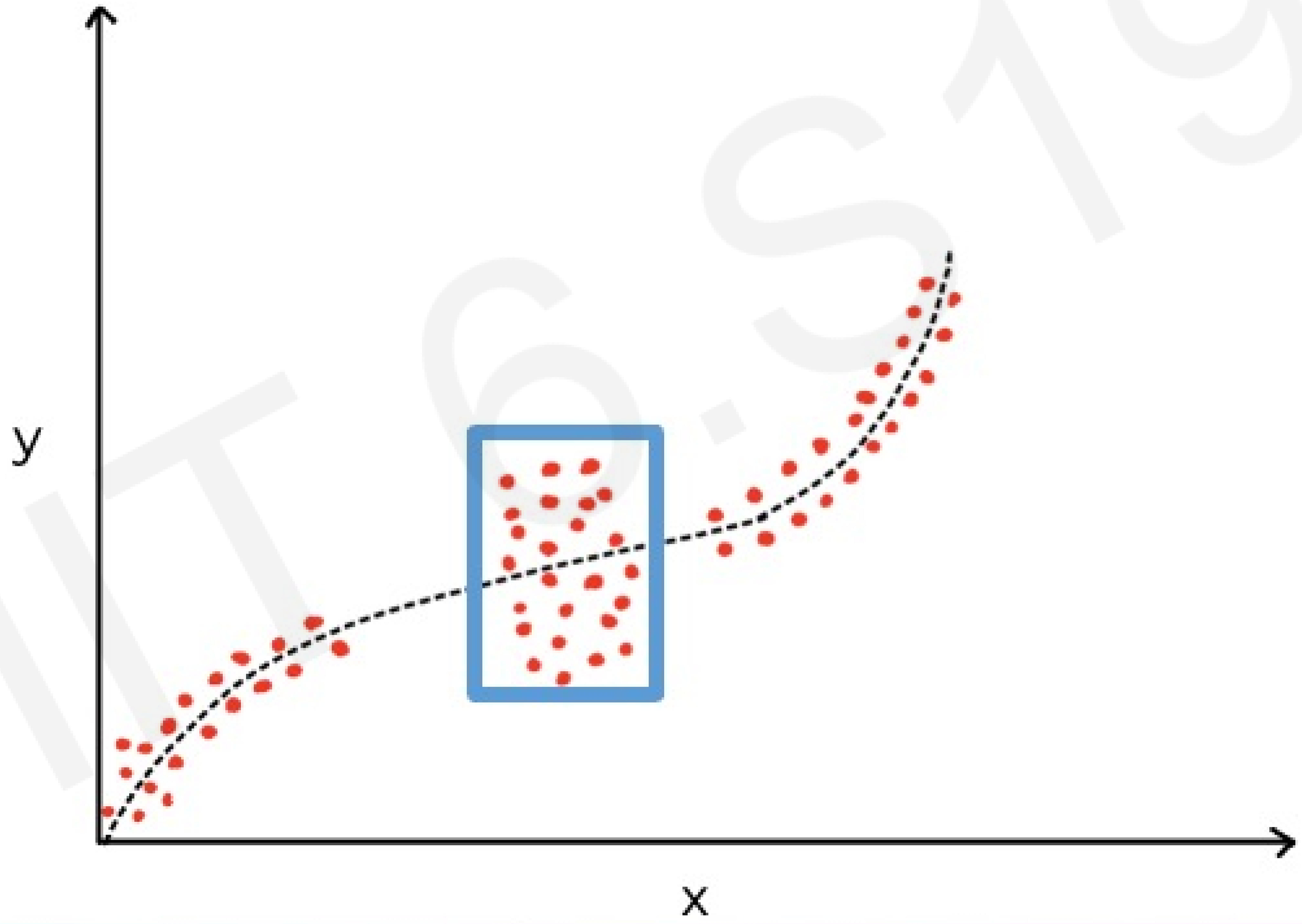
Types of Uncertainty in Neural Networks

Let's say we're trying to estimate the curve $y = x^3$, and our dataset looks like the red points below:



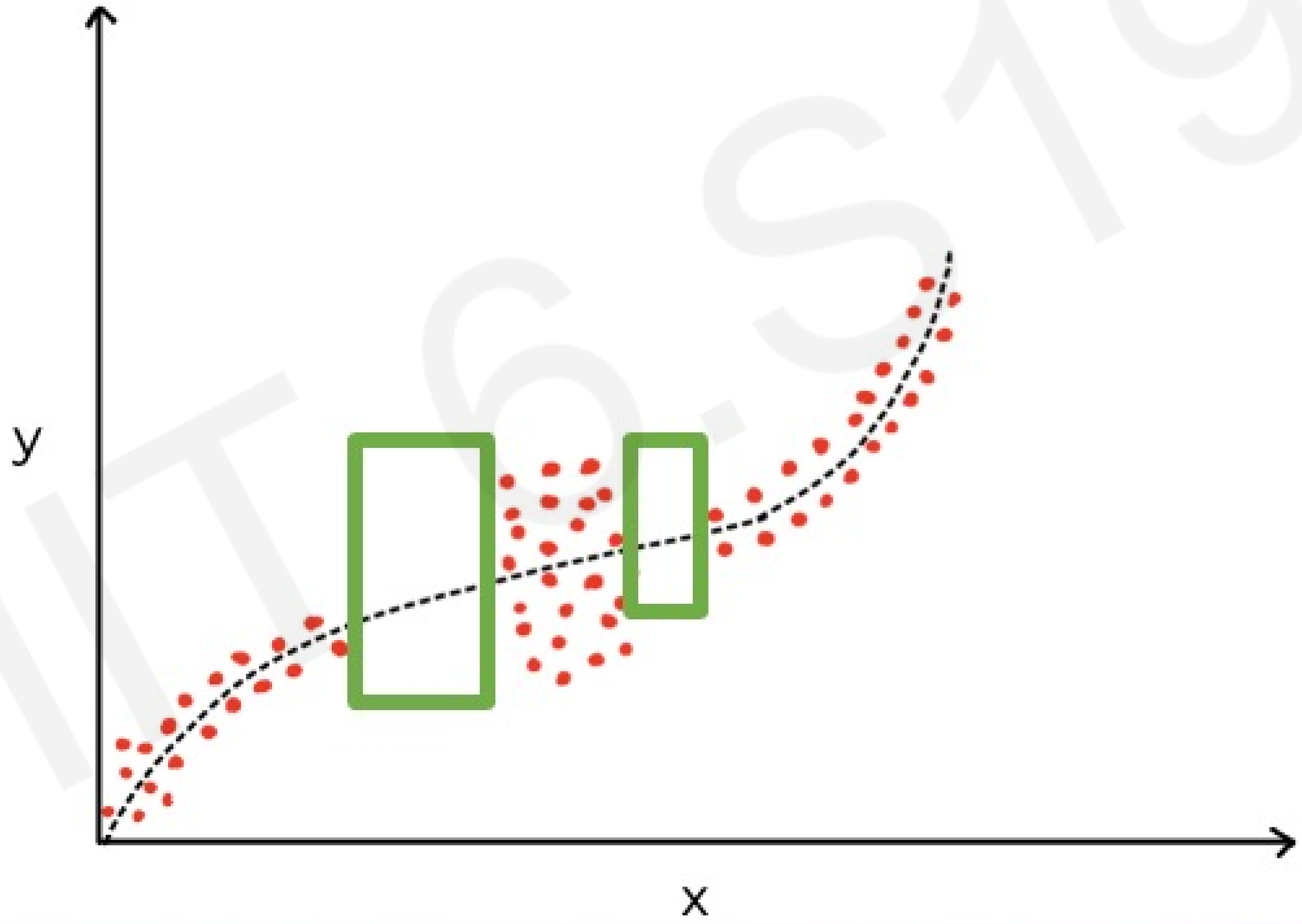
Types of Uncertainty in Neural Networks

The boxed area shows a region of high **data uncertainty**: very similar inputs have drastically different outputs

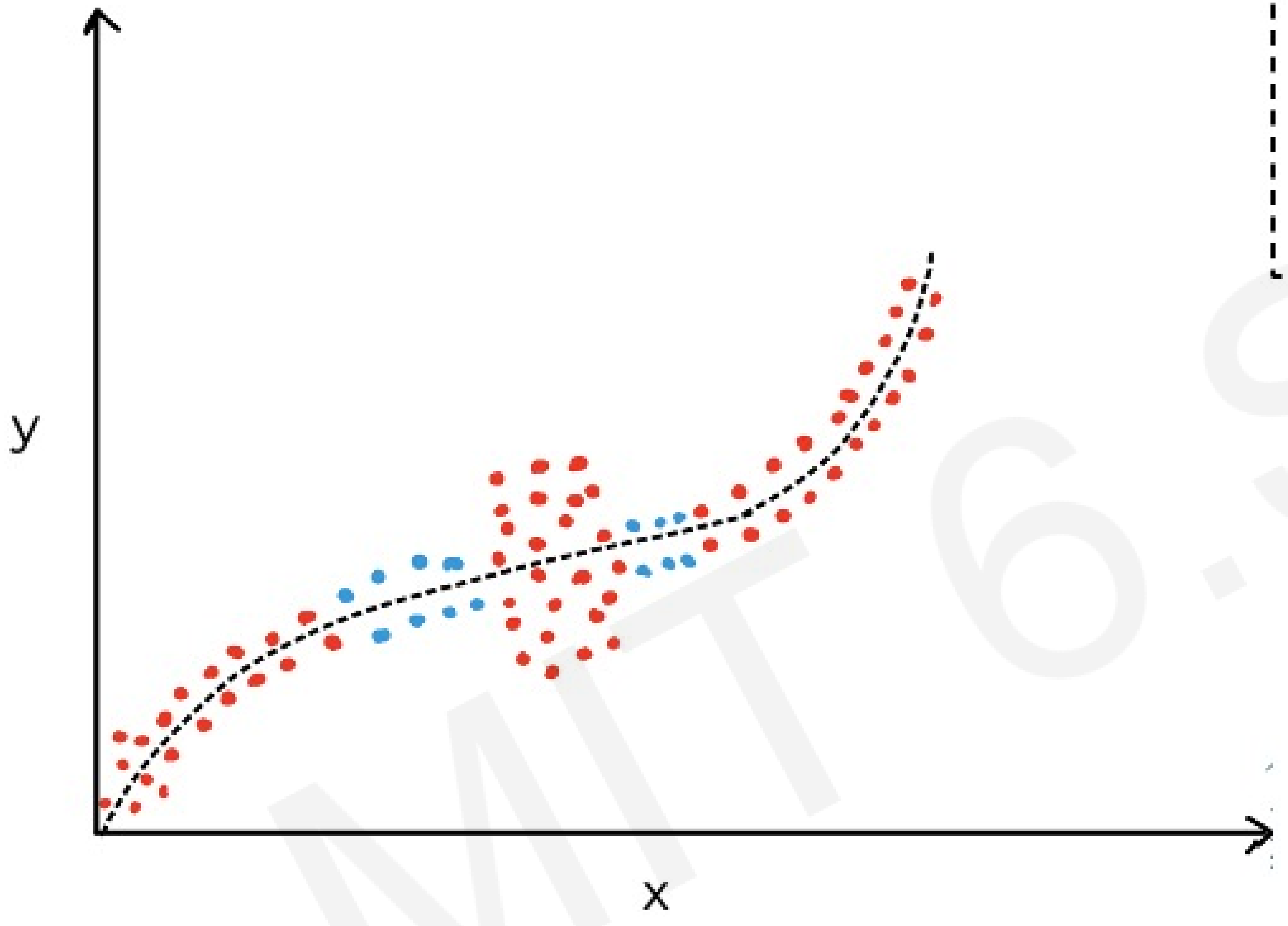


Types of Uncertainty in Neural Networks

The boxed area shows a region of high **model uncertainty**:
points here are out of distribution

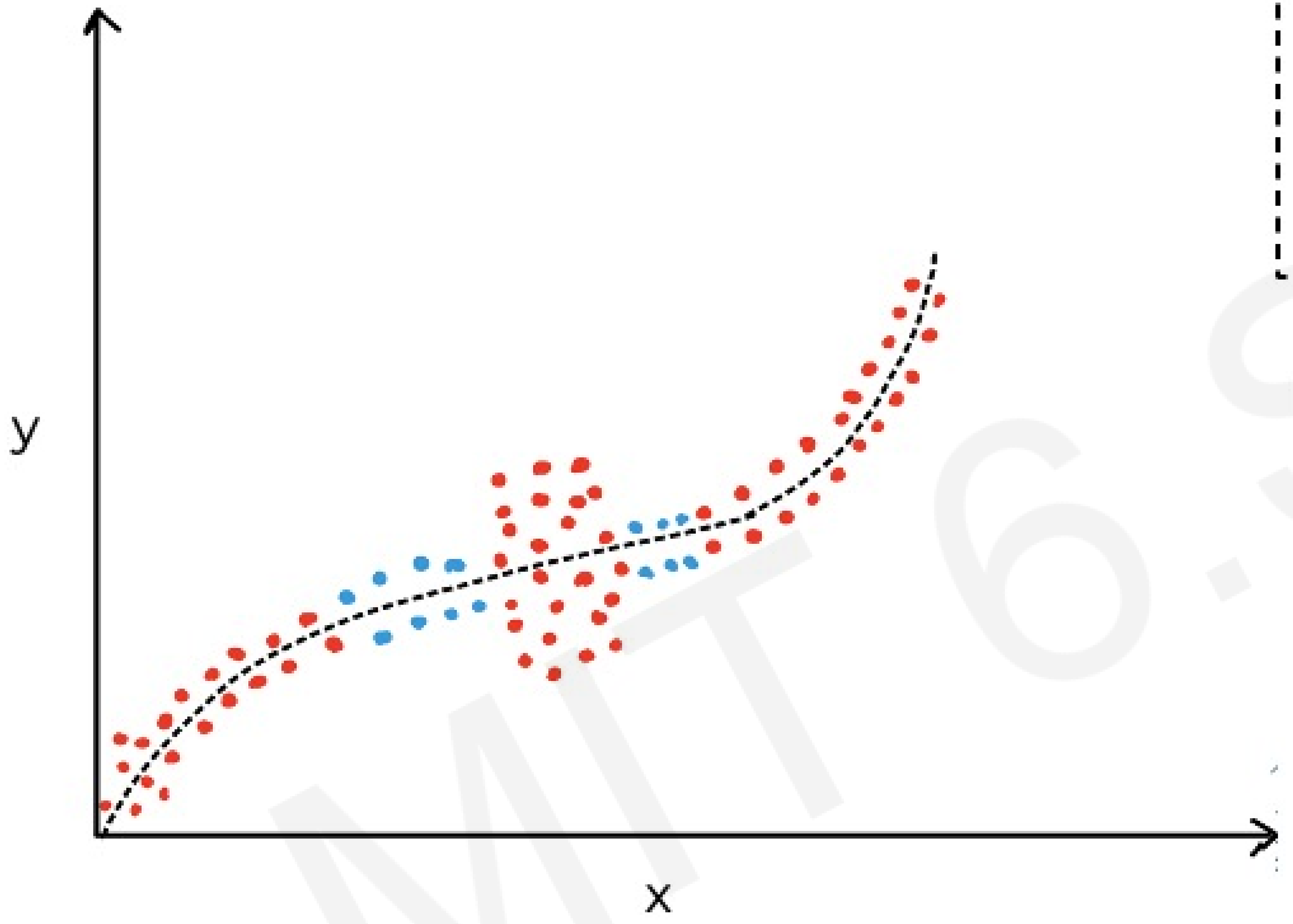


Augmenting datasets to mitigate uncertainty



Would adding the **blue** training points to our dataset reduce uncertainty? If so, which type of uncertainty?

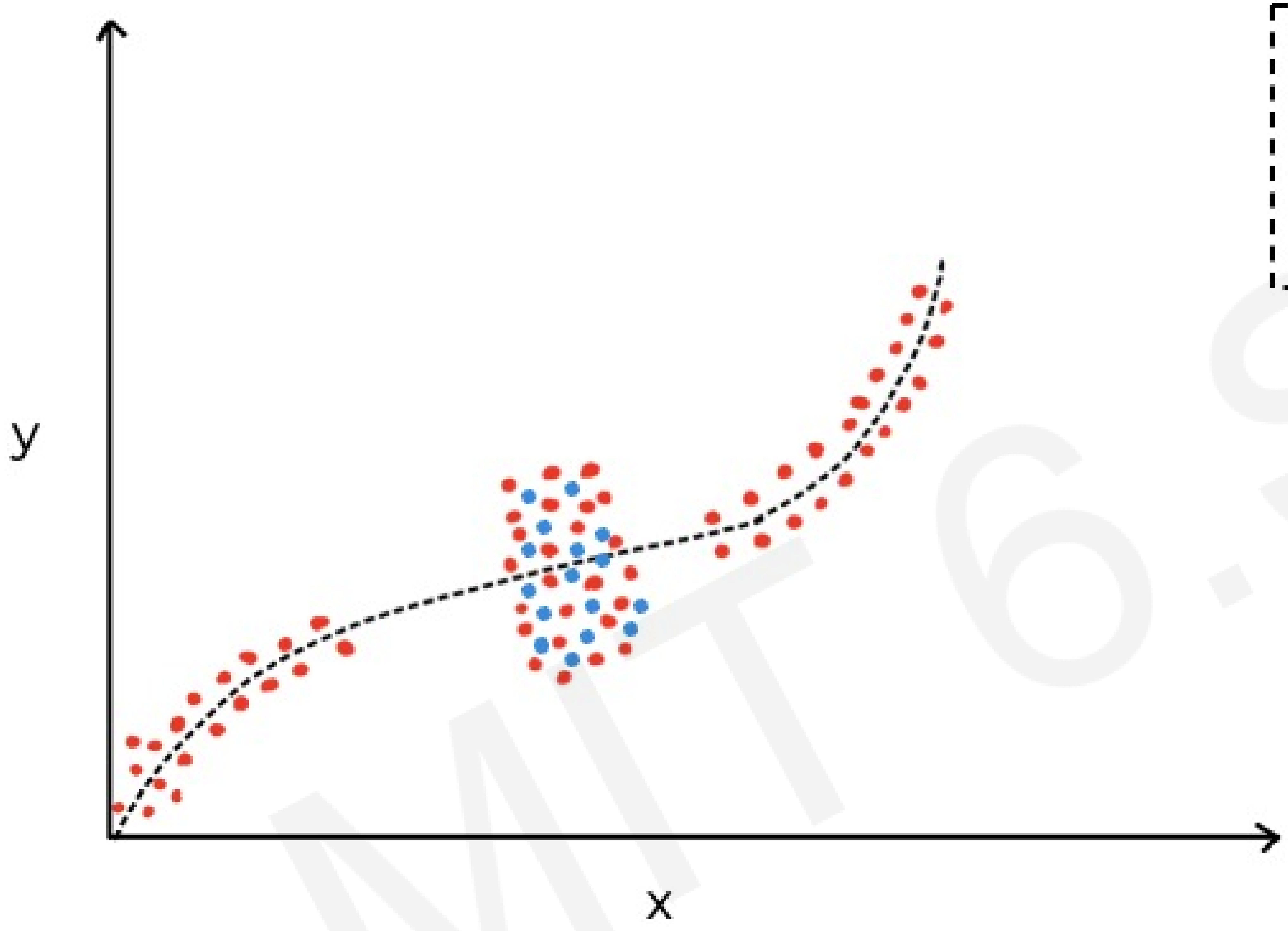
Augmenting datasets to mitigate uncertainty



Would adding the **blue** training points to our dataset reduce uncertainty? If so, which type of uncertainty?

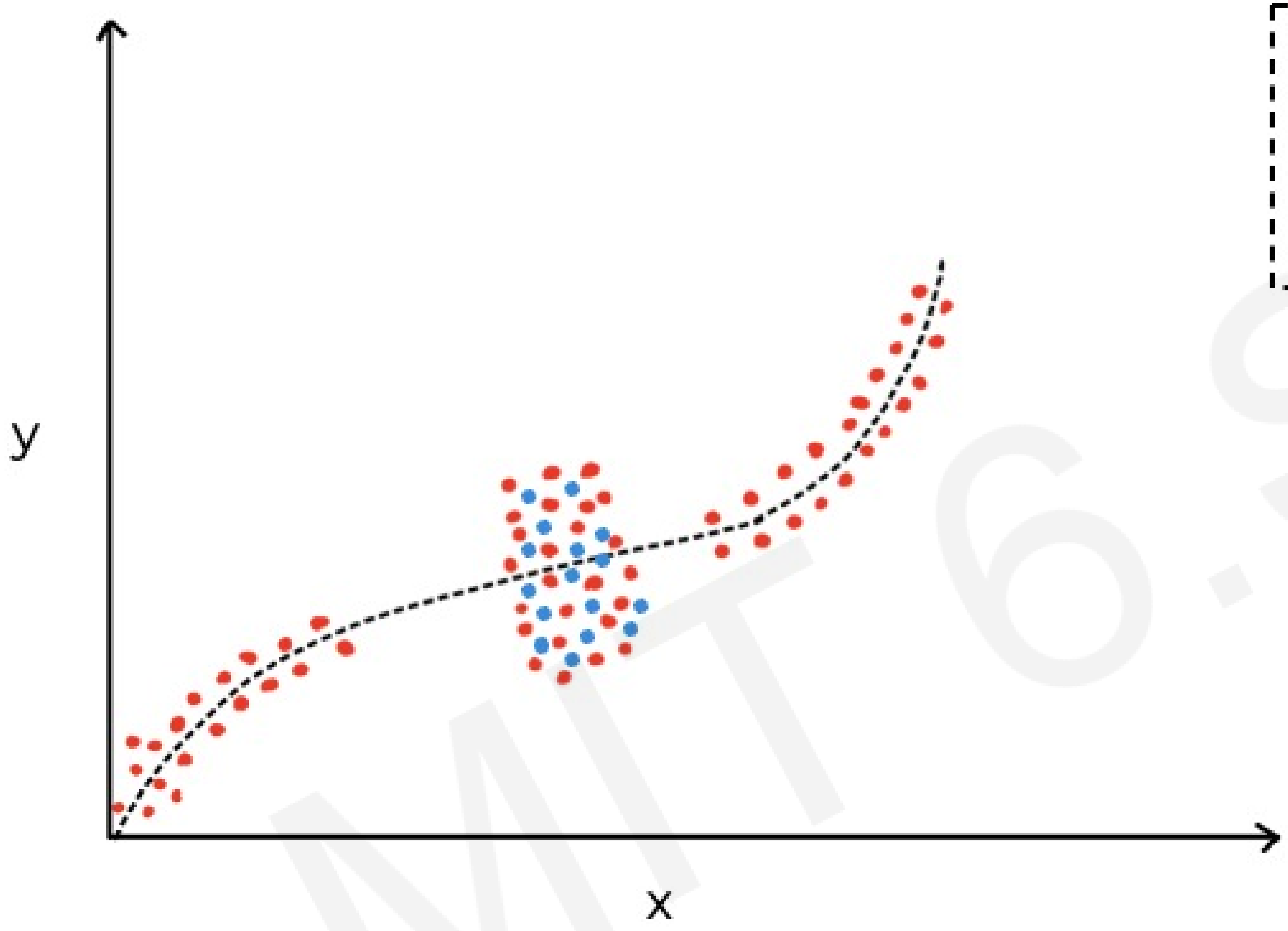
Model uncertainty is reduced by adding data!

Augmenting datasets to mitigate uncertainty



Would adding the **blue** training points to our dataset reduce uncertainty? If so, which type of uncertainty?

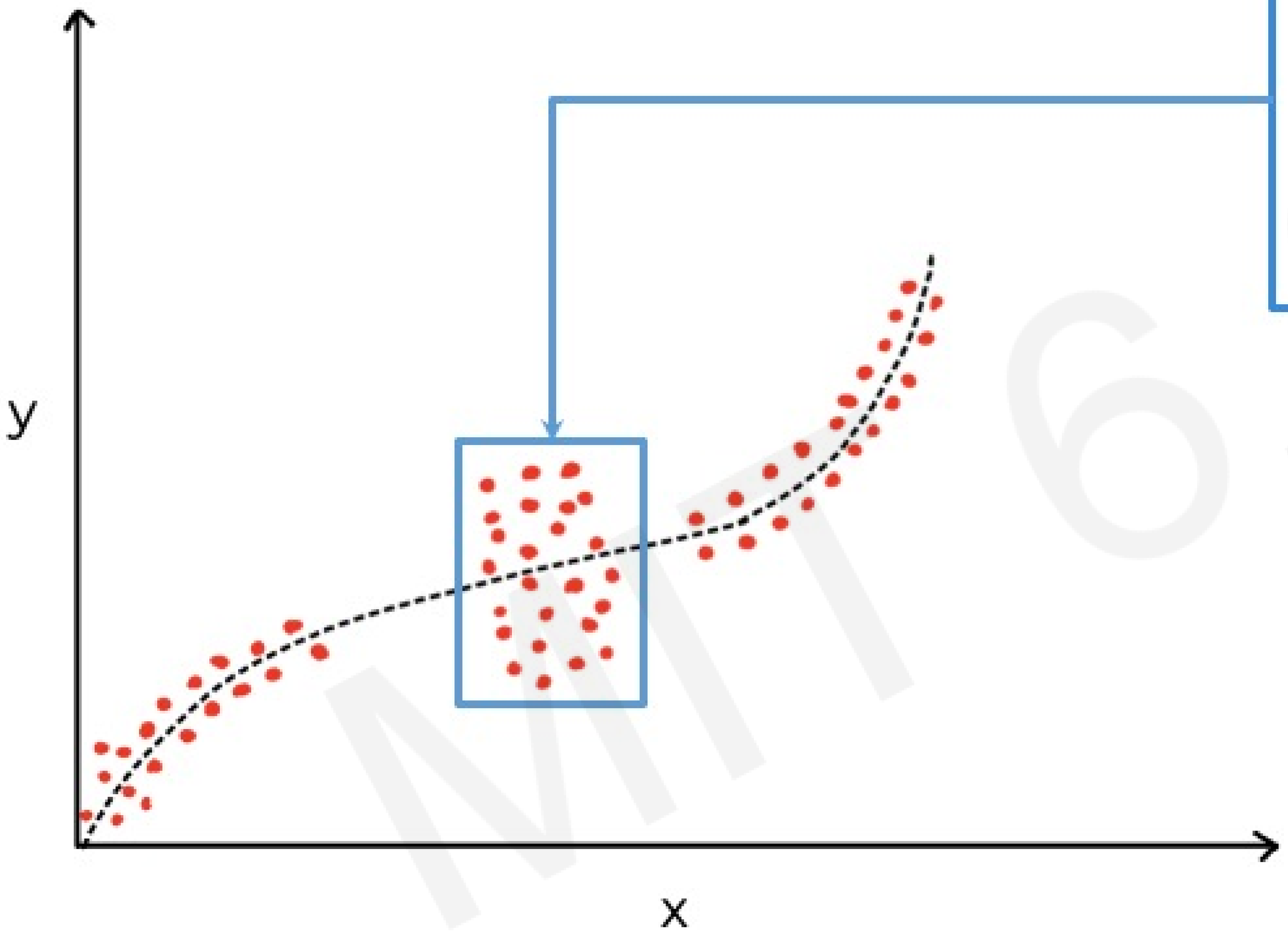
Augmenting datasets to mitigate uncertainty



Would adding the **blue** training points to our dataset reduce uncertainty? If so, which type of uncertainty?

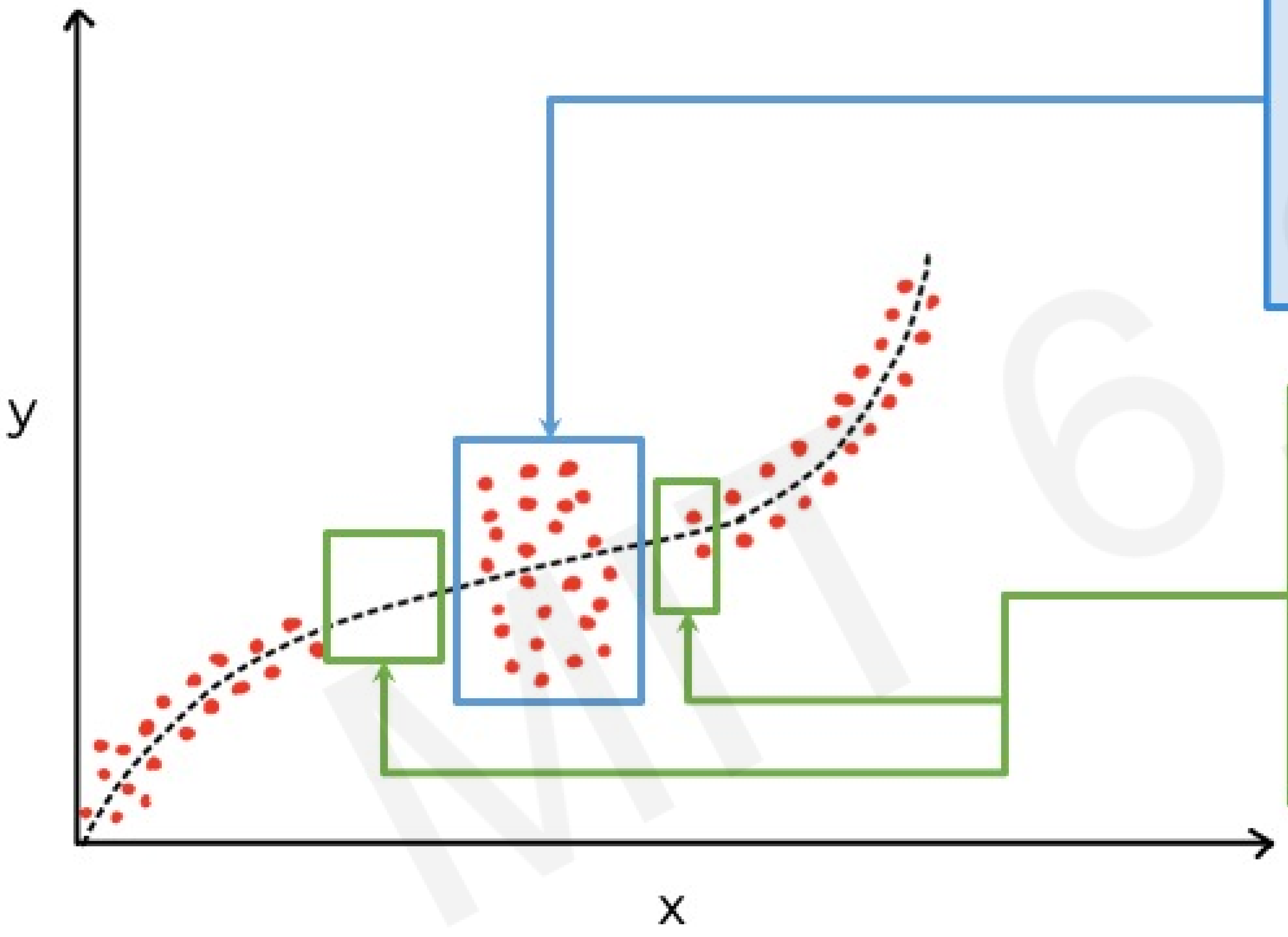
No-- data uncertainty is **irreducible!**

Aleatoric vs. Epistemic Uncertainty



- **Aleatoric** uncertainty = data uncertainty
- Irreducible!
- Can be **directly learned from data**

Aleatoric vs. Epistemic Uncertainty



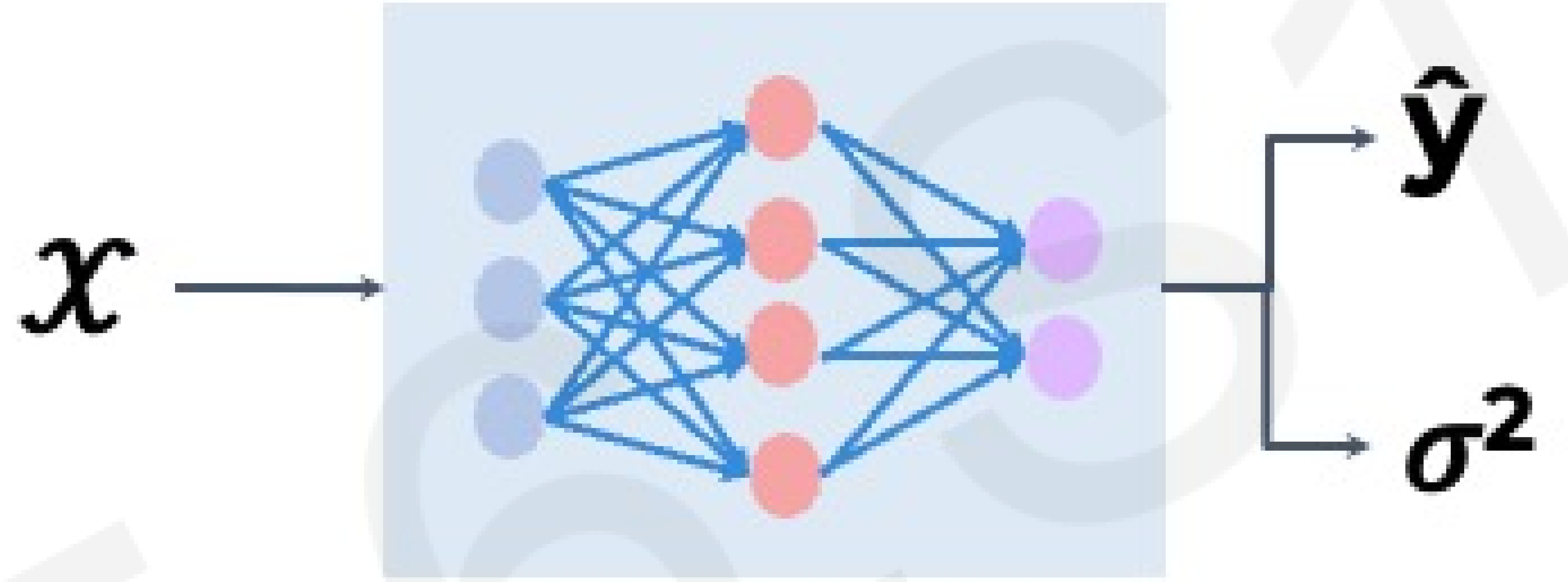
- **Aleatoric** uncertainty = data uncertainty
- Irreducible!
- Can be **directly learned from data**

- **Epistemic** uncertainty = model uncertainty
- Reducible by adding data!
- **Cannot** be directly learned from data

Estimating Aleatoric Uncertainty: Regression

Goal: learn a set of **variances** corresponding to the input

Higher variance → there is more uncertainty at this part of the dataset (more noise!)



$$f_\theta(x) \rightarrow \hat{y}, \sigma^2$$

This variance is **not constant** and depends on the value of x!

Negative Log Likelihood Loss to Learn Variance

Our current loss function does not take into account variance:

$$\mathcal{L} = \frac{1}{N} \times \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$$

By minimizing Mean Squared Error, we can learn the parameters of a multivariate **Gaussian** with mean y_i and **constant variance.**

Negative Log Likelihood Loss to Learn Variance

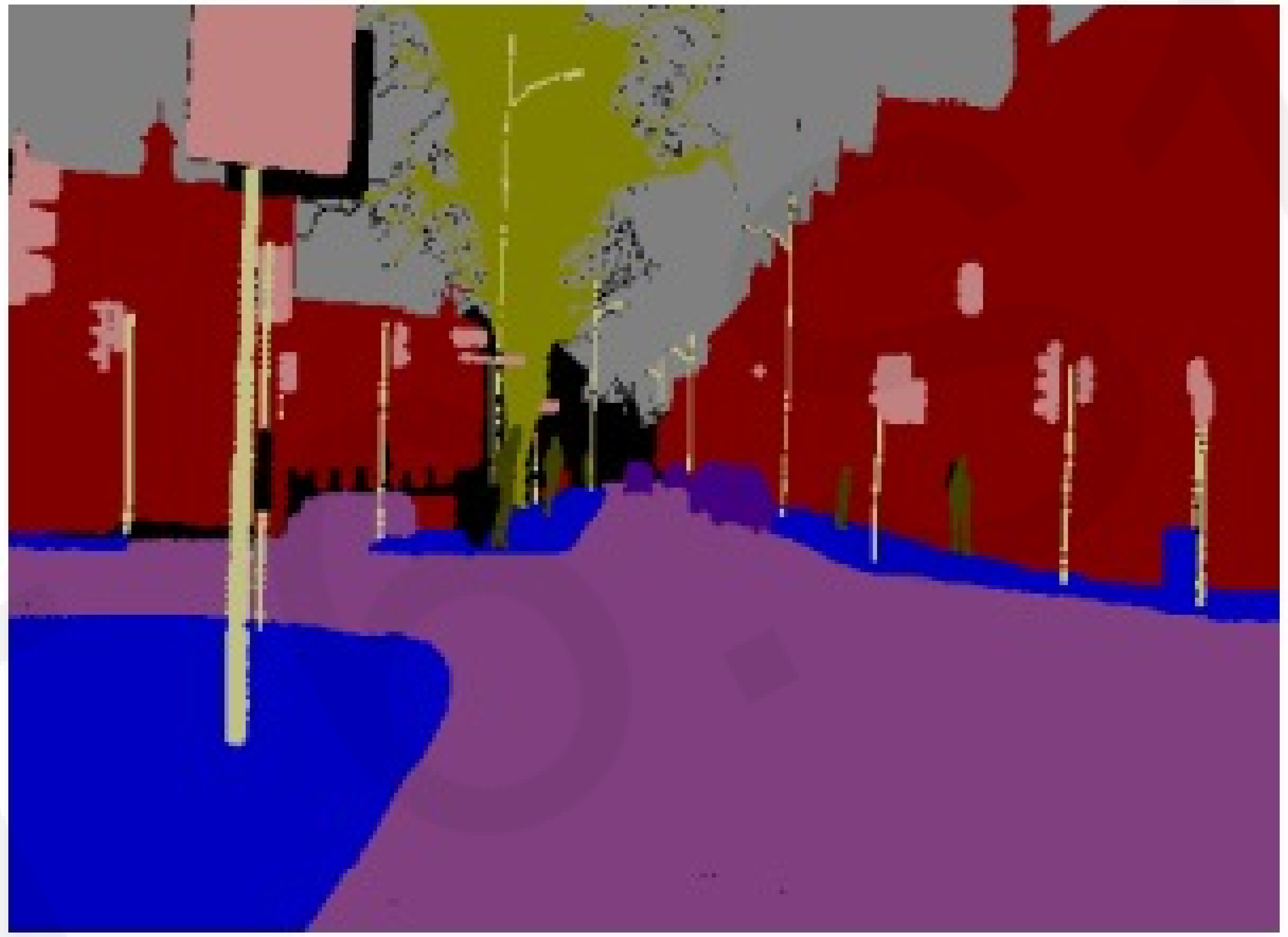
Negative Log Likelihood (NLL) is a **generalization** of MSE to **non-constant variances**:

$$\mathcal{L} = \frac{1}{N} \times \sum_{i=1}^{N} \frac{(\hat{y}_i - y_i)^2}{2\sigma_i^2} + \ln \sigma_i^2$$

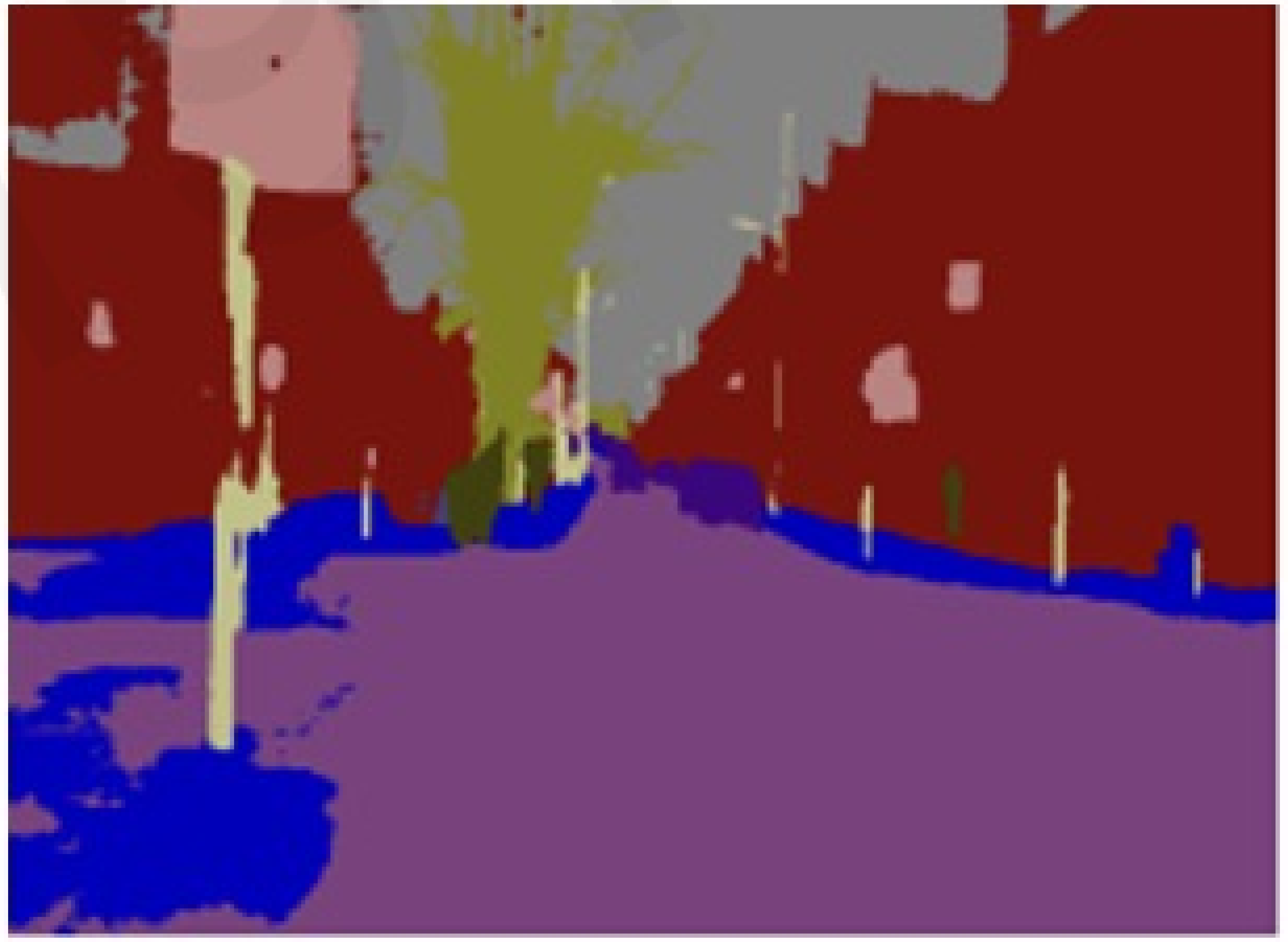
Aleatoric Uncertainty in the Real World: Semantic Segmentation

Semantic Segmentation: label every **pixel** of an image with its corresponding class

Inputs: RGB Images of scenes in cities



Labels: pixel-level masks of image



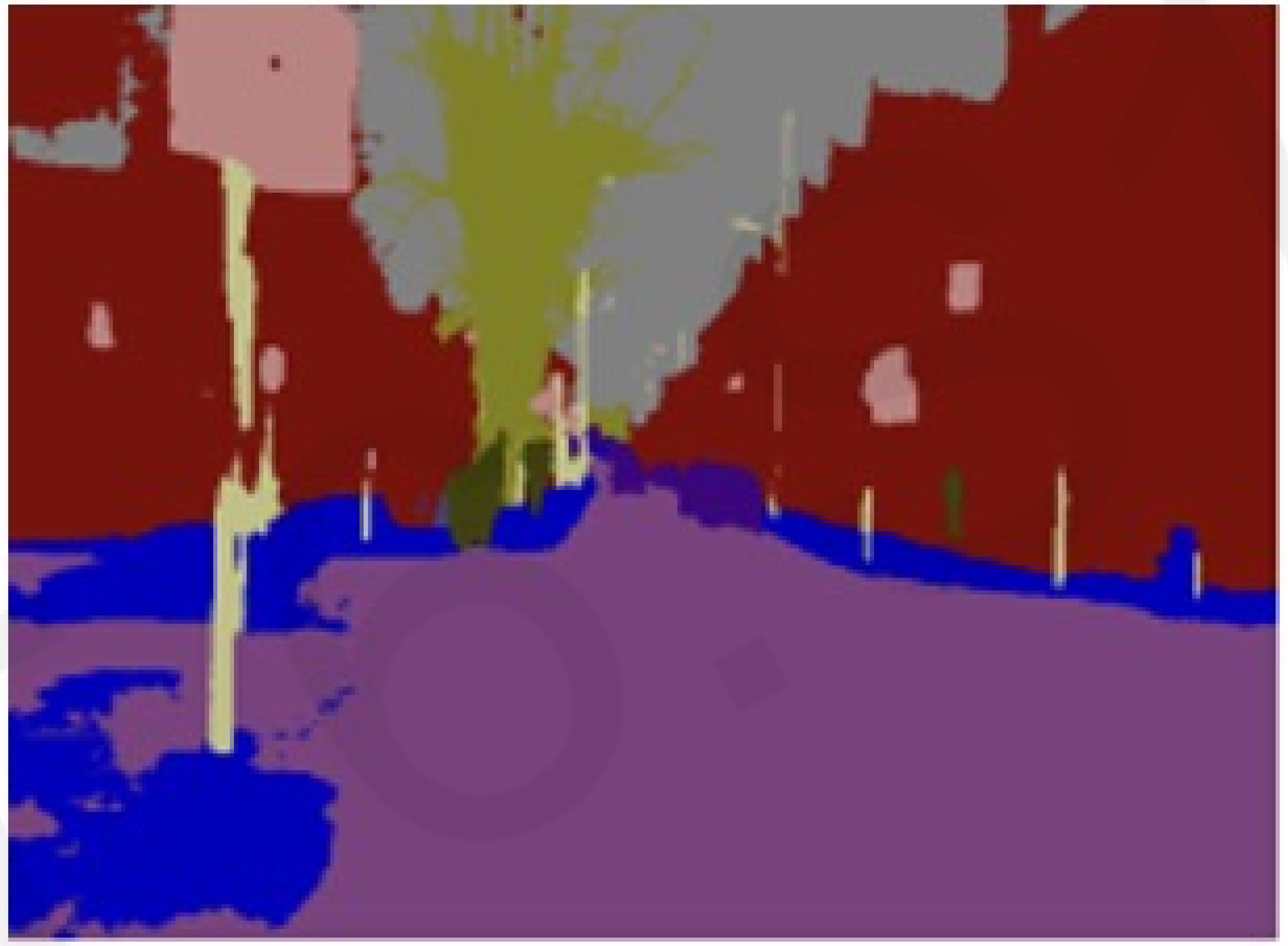
Outputs: predicted pixel-level masks of image

Which parts of this dataset have high **data** or aleatoric uncertainty?

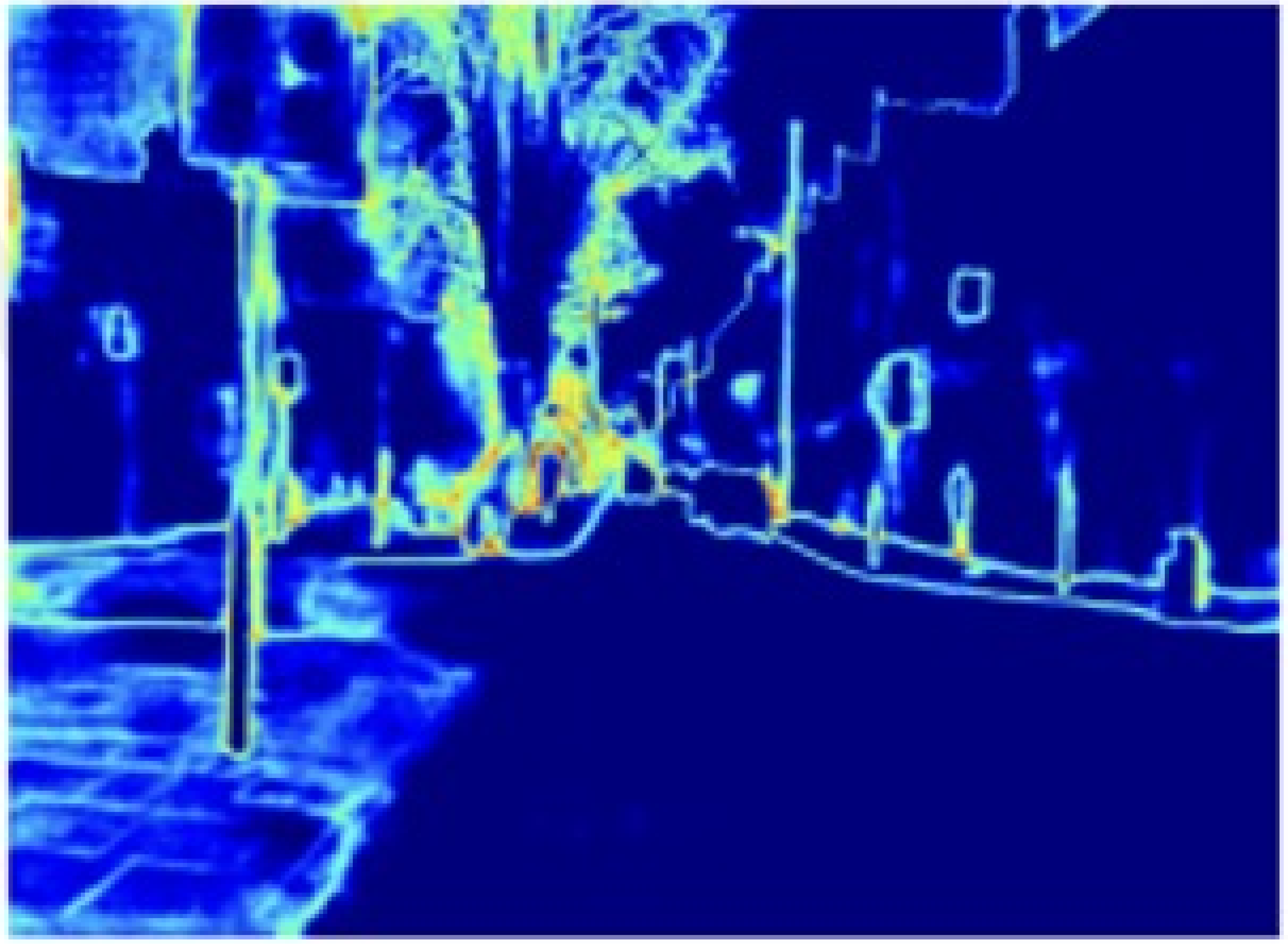
Aleatoric Uncertainty in the Real World: Semantic Segmentation

Semantic Segmentation: label every **pixel** of an image with its corresponding class

Inputs: RGB Images of scenes in cities

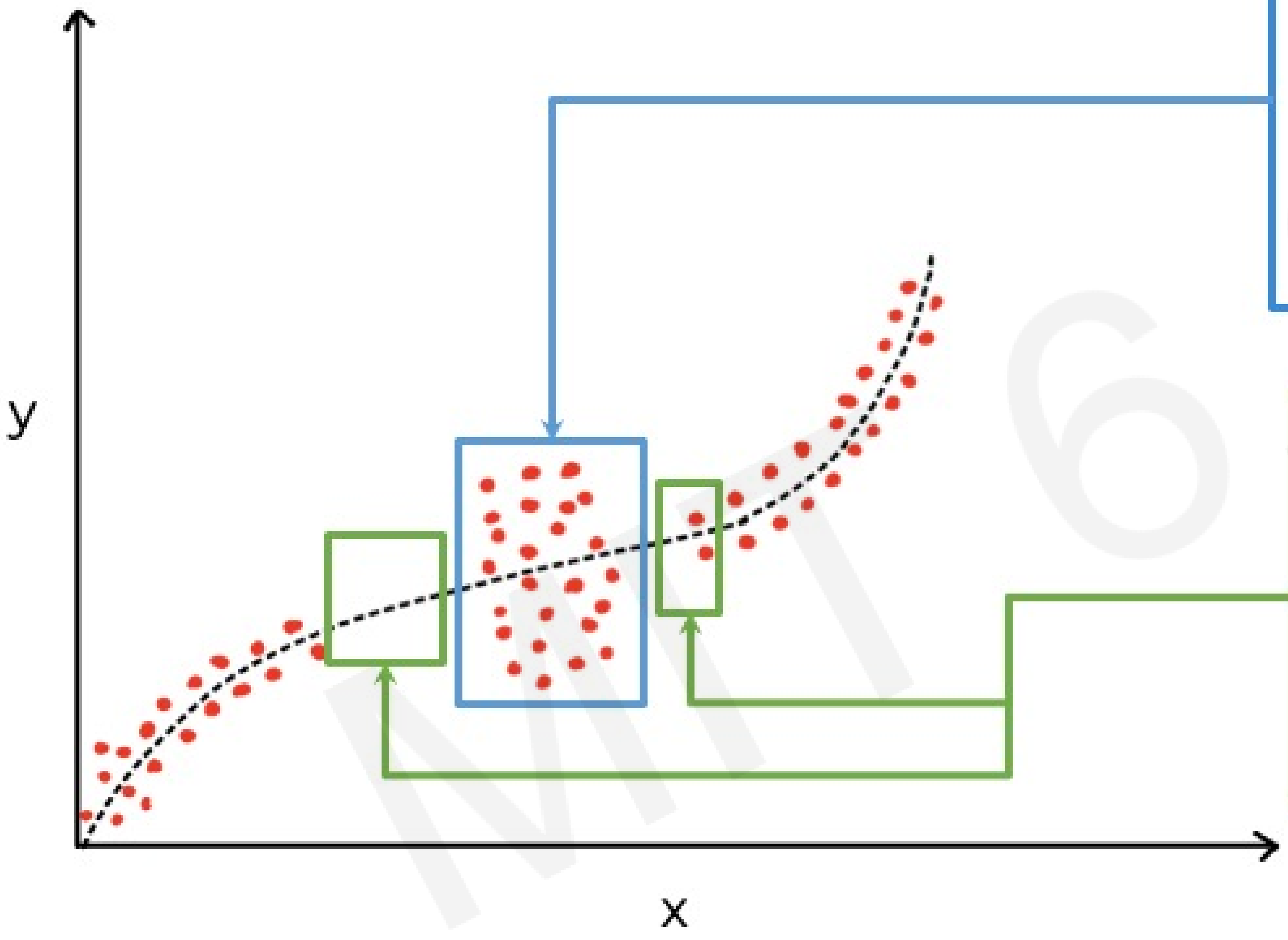


Outputs: pixel-level masks of labels



Corners and boundaries have high aleatoric uncertainty

Aleatoric vs. Epistemic Uncertainty

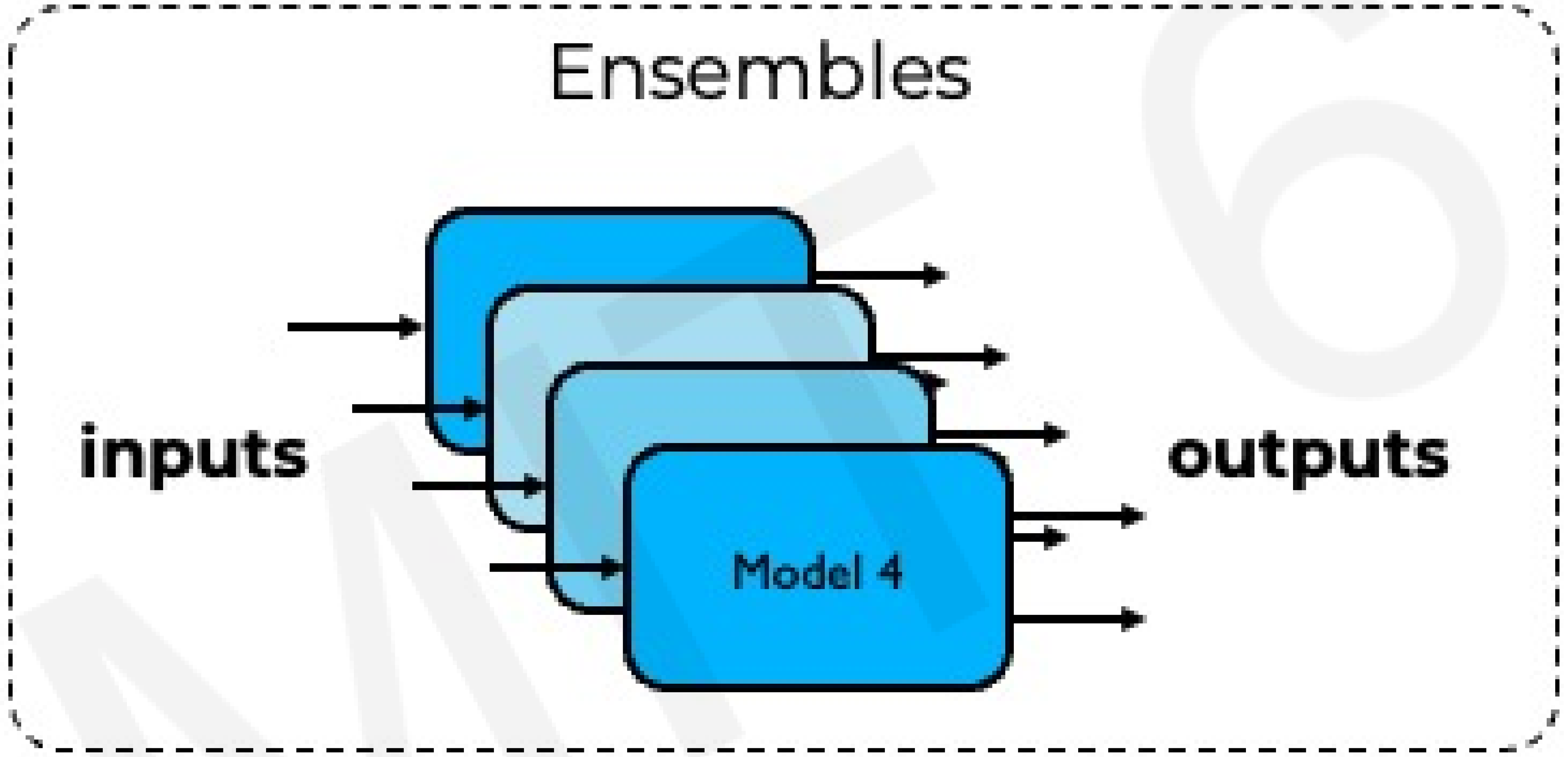


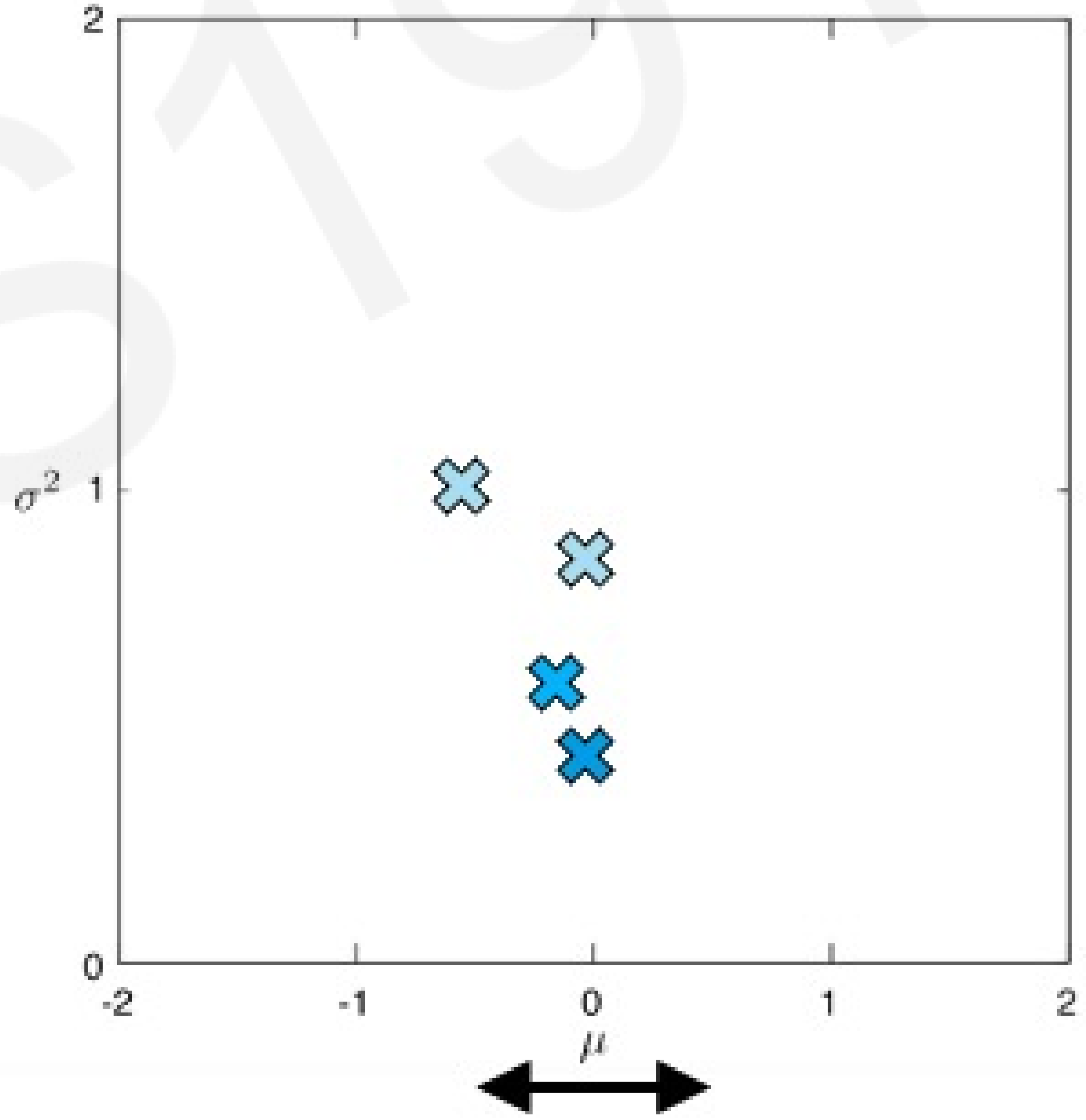
- **Aleatoric** uncertainty = data uncertainty
- Irreducible!
- Can be **directly learned from data**

- **Epistemic** uncertainty = model uncertainty
- Reducible by adding data!
- **Cannot** be directly learned from data

Introduction to Estimating Epistemic Uncertainty

What if we train the same network multiple times (an **ensemble** of networks) and compare outputs?





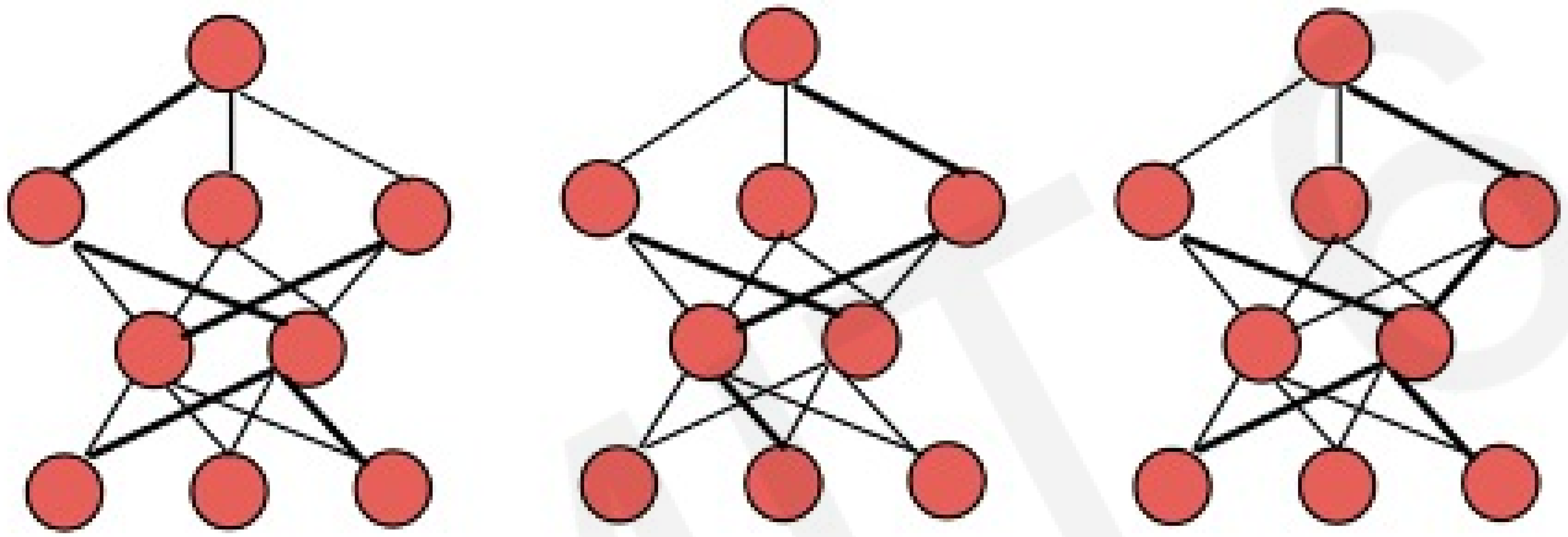
Estimating Epistemic Uncertainty through Sampling: Ensembling

- "Familiar" inputs → similar output for every network
- "Unfamiliar" inputs → different outputs for every network

```
num_ensembles = 5
for i in range(num_ensembles):
    model = create_model(...)
    model.fit(...)
```

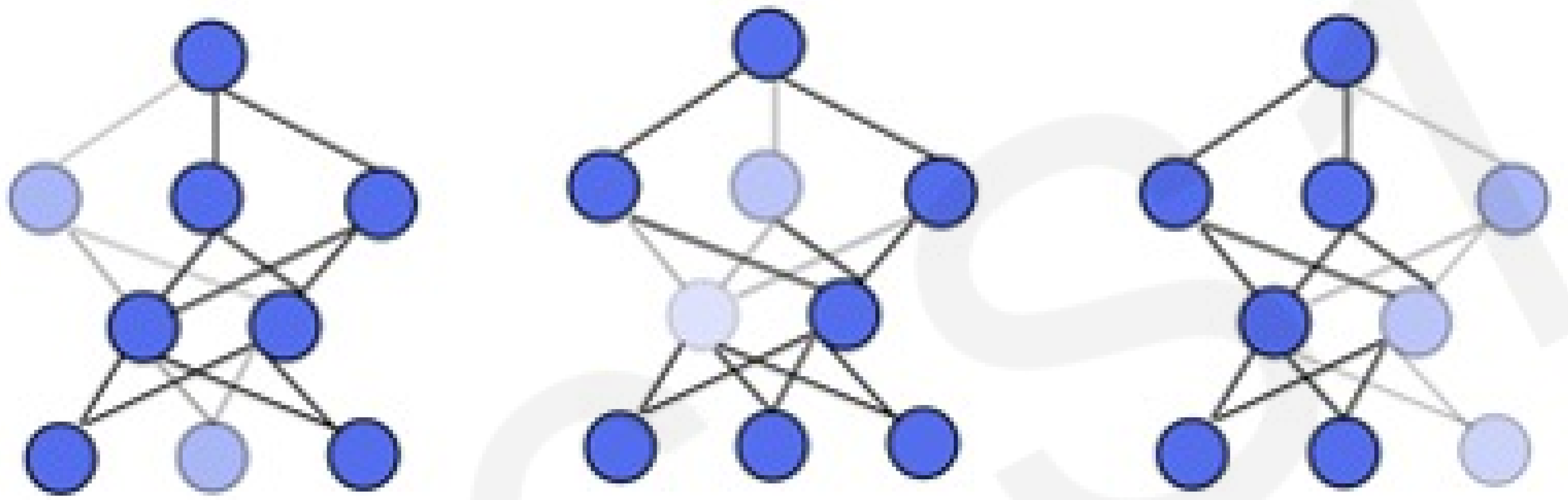
```
raw_predictions = [models[i].predict(x)
    for i in range(num_ensembles)]
```

```
mu = np.mean(raw_predictions)
uncertainty = np.var(raw_predictions)
```



Estimating Epistemic Uncertainty through Sampling: Dropout

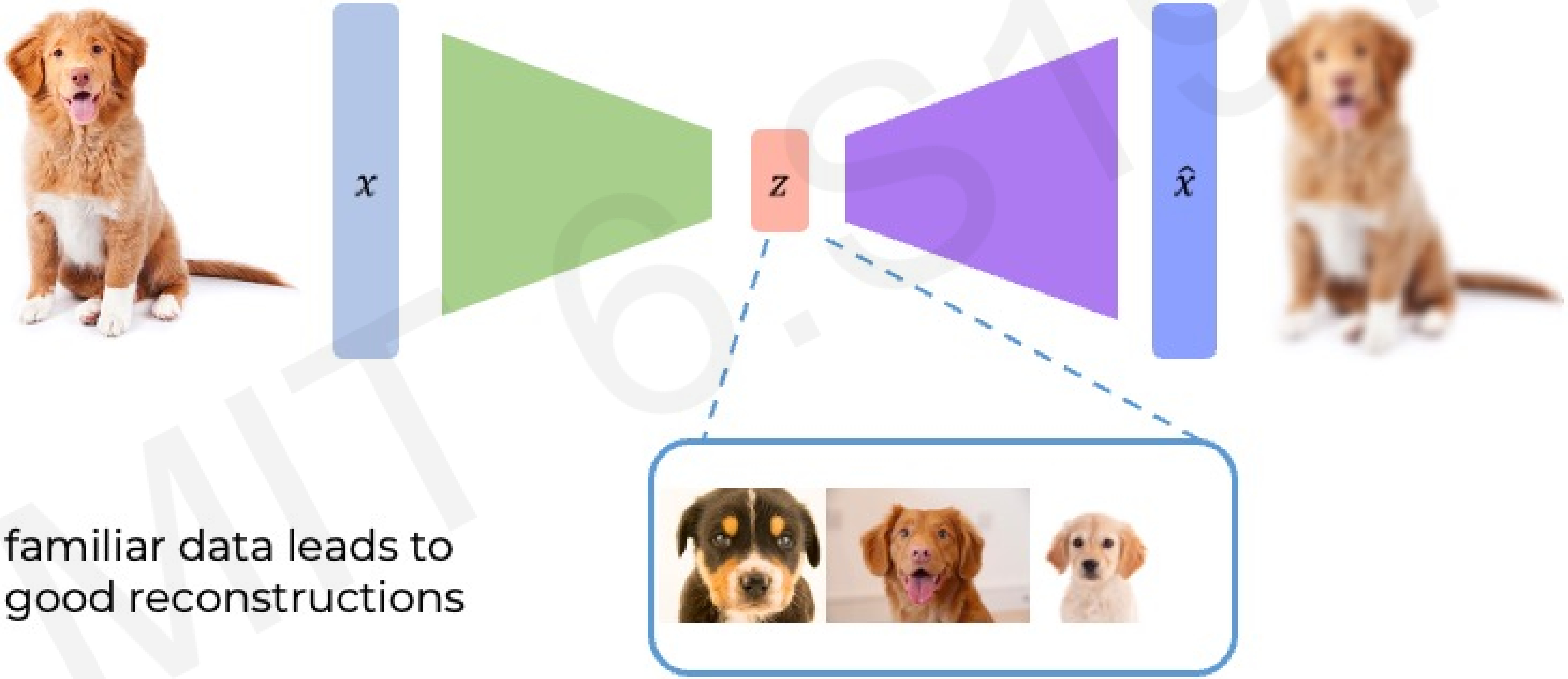
To introduce *stochasticity*, we can also add dropout layers and compute forward passes multiple times while saving memory and compute



```
for _ in range(T):
    forward_passes.append(model(x, dropout=True))
mu = np.mean(forward_passes)
uncertainty = np.var(forward_passes)
```

Estimating Epistemic Uncertainty: Reconstruction Error

In addition to sampling, we can use reconstruction error to measure how confident the model is in a prediction



familiar data leads to good reconstructions

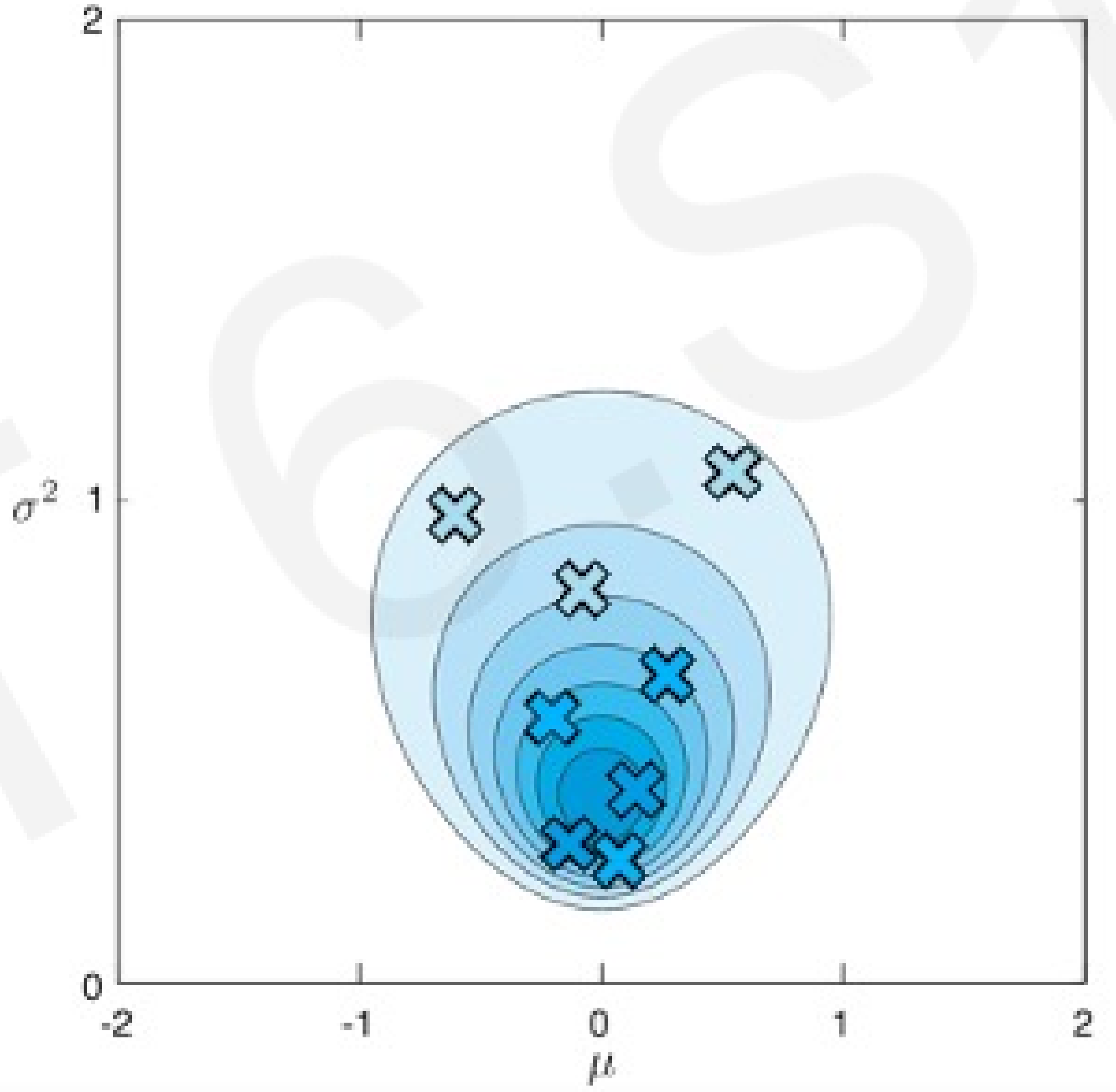
Estimating Epistemic Uncertainty: Reconstruction Error

In addition to sampling, we can use reconstruction error to measure how confident the model is in a prediction

OOD inputs lead to worse reconstruction error than familiar data

Estimating Epistemic Uncertainty: Evidential Deep Learning

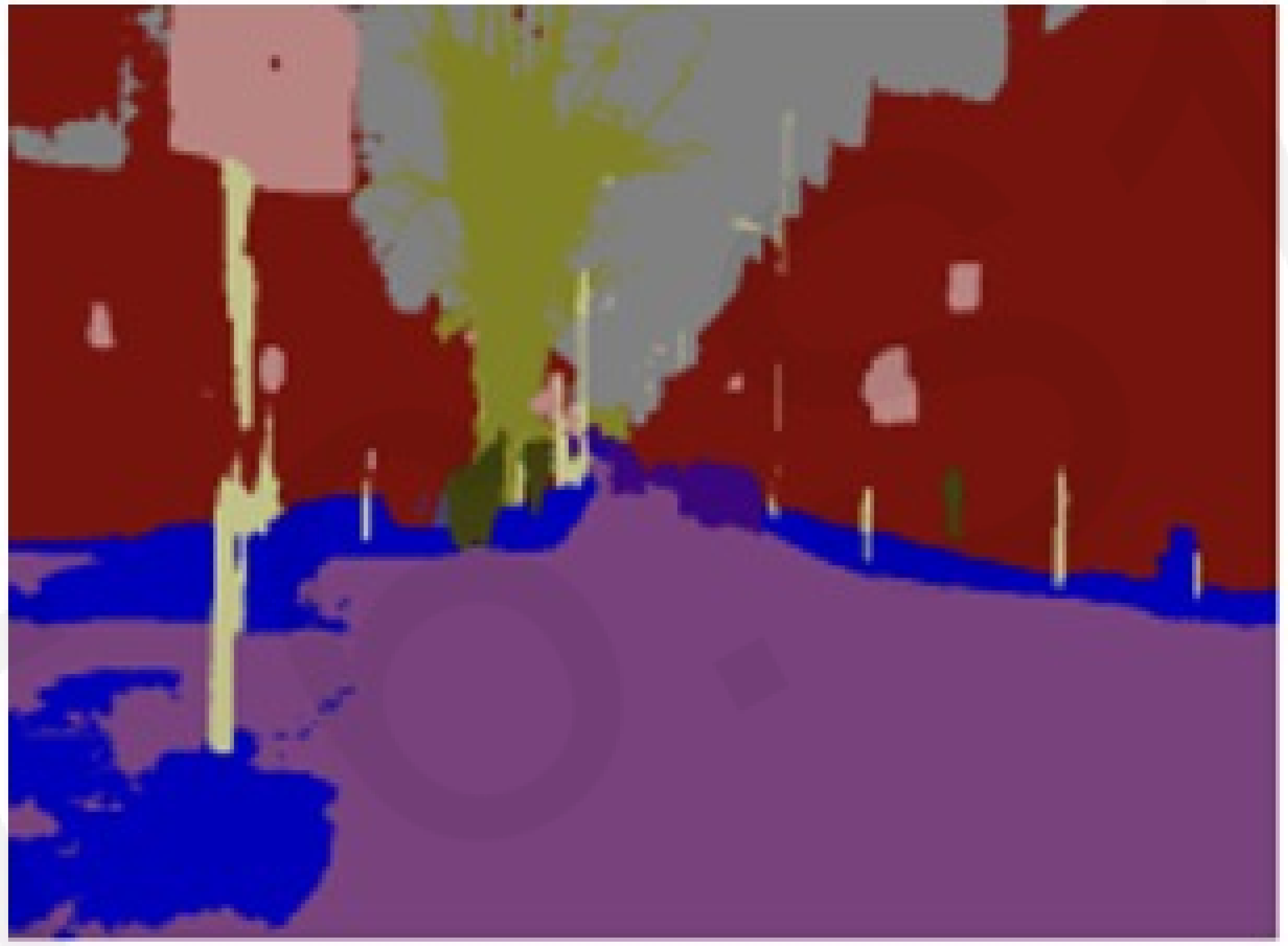
Learn the variance **directly**, without sampling by placing priors on the distribution that the evidence comes from.



Epistemic Uncertainty in the Real World: Semantic Segmentation

Semantic Segmentation: label every **pixel** of an image with its corresponding class

Inputs: RGB Images of scenes in cities



Outputs: pixel-level masks of labels

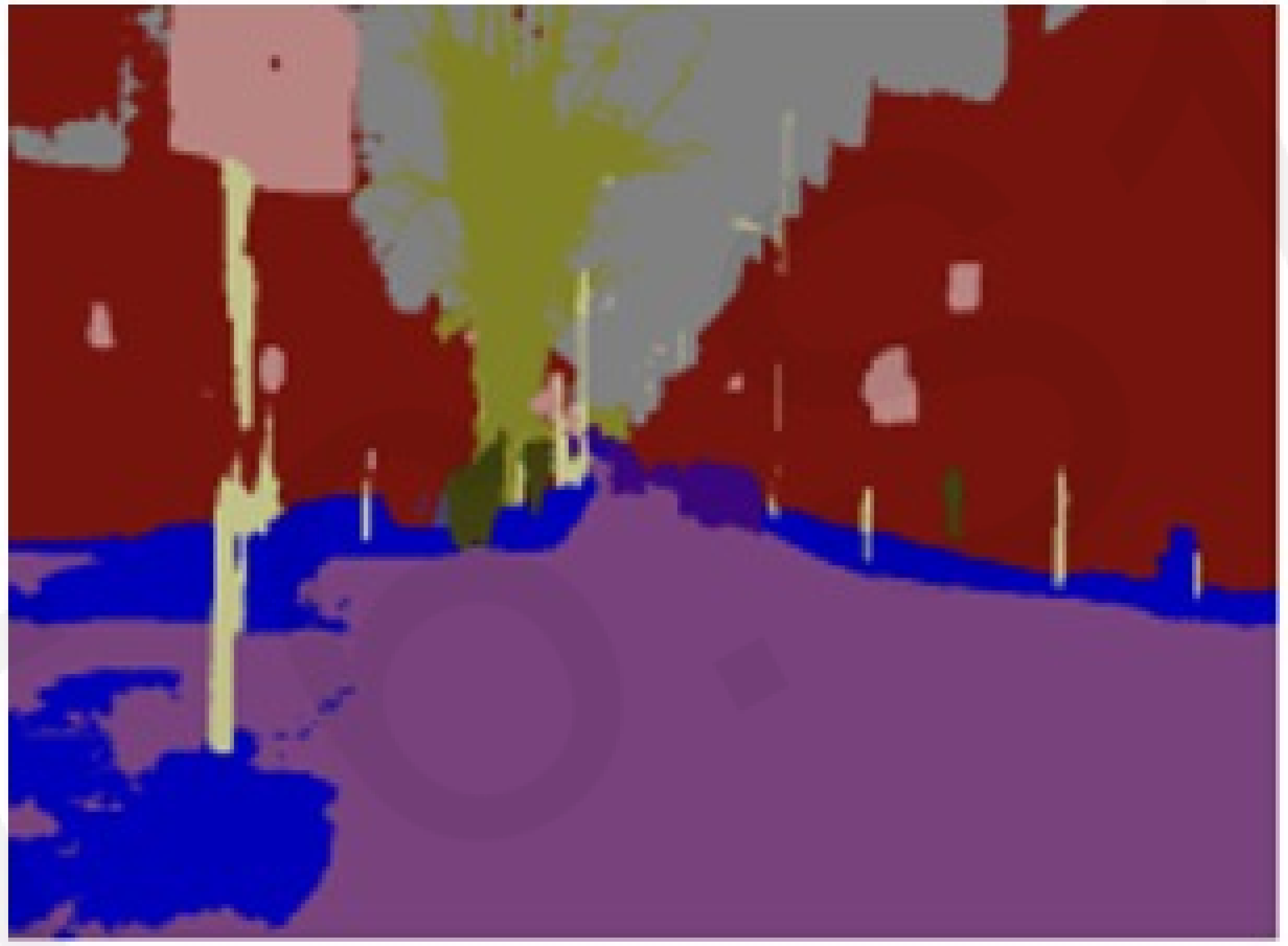
Which parts of this dataset have high **model** or epistemic uncertainty?

Kendall/Gal *NeurIPS* 2017

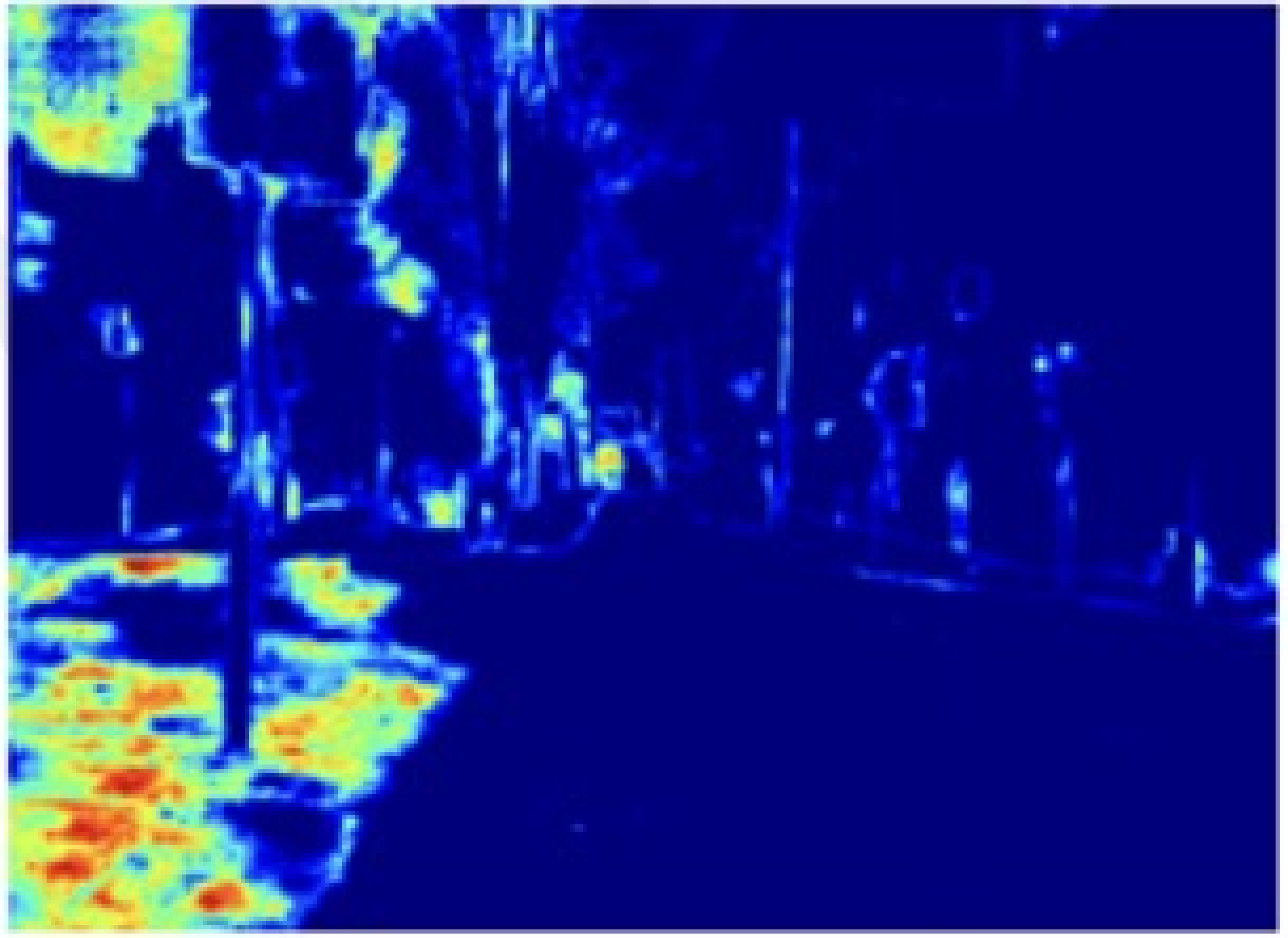
Epistemic Uncertainty in the Real World: Semantic Segmentation

Semantic Segmentation: label every **pixel** of an image with its corresponding class

Inputs: RGB Images of scenes in cities



Outputs: pixel-level masks of labels

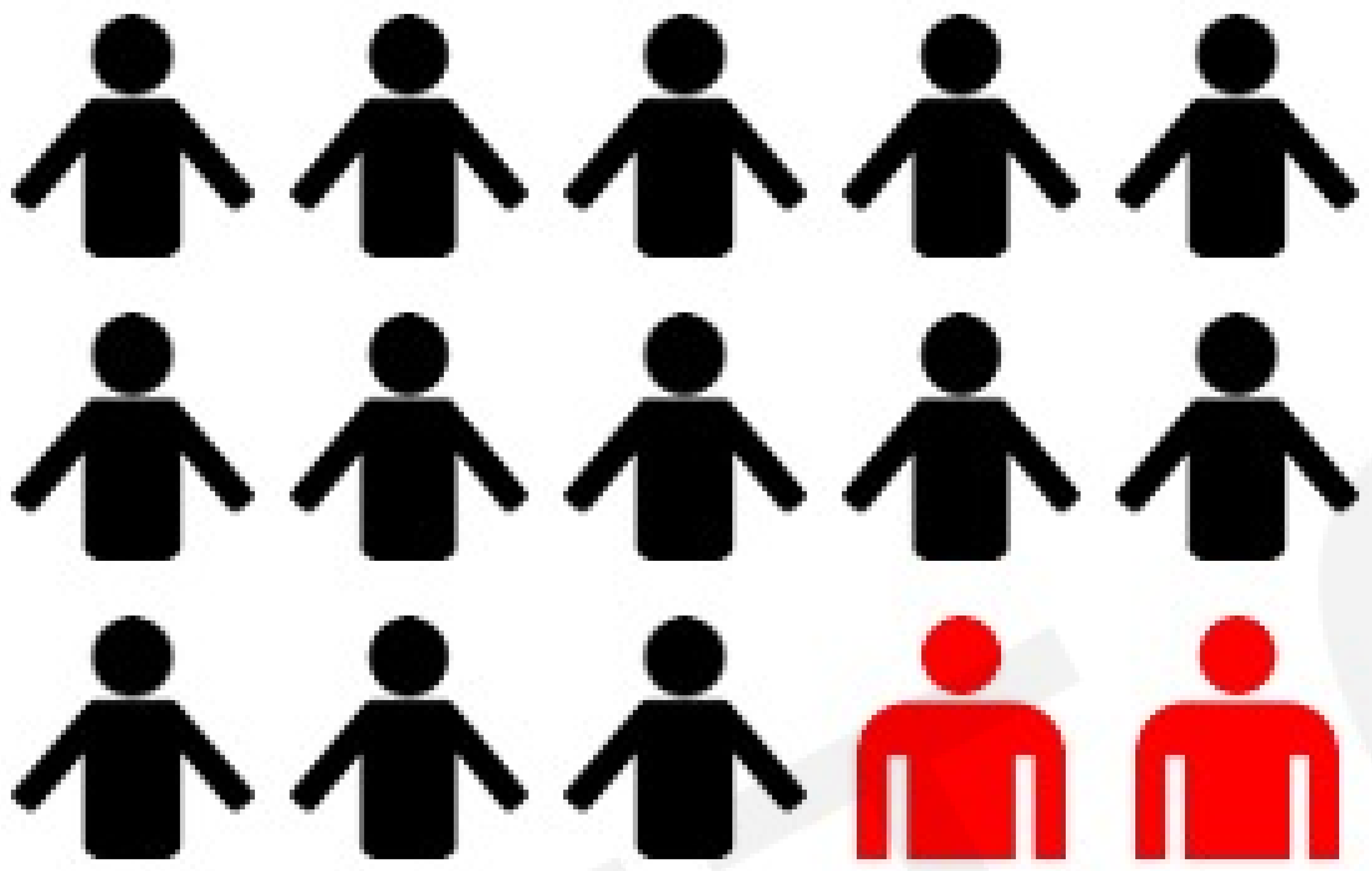


High epistemic uncertainty in areas of **discoloration**

Kendall/Gal *NeurIPS* 2017

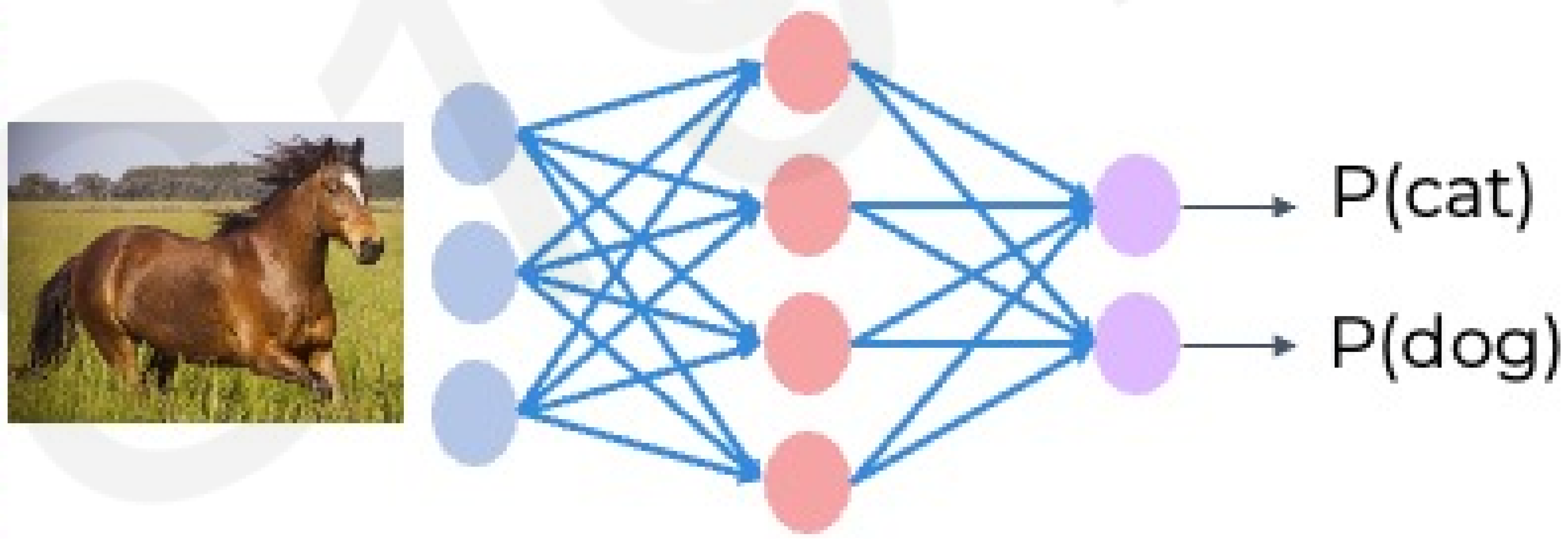
Challenges for Robust Deep Learning

Bias



What happens when models are skewed by sensitive feature inputs?

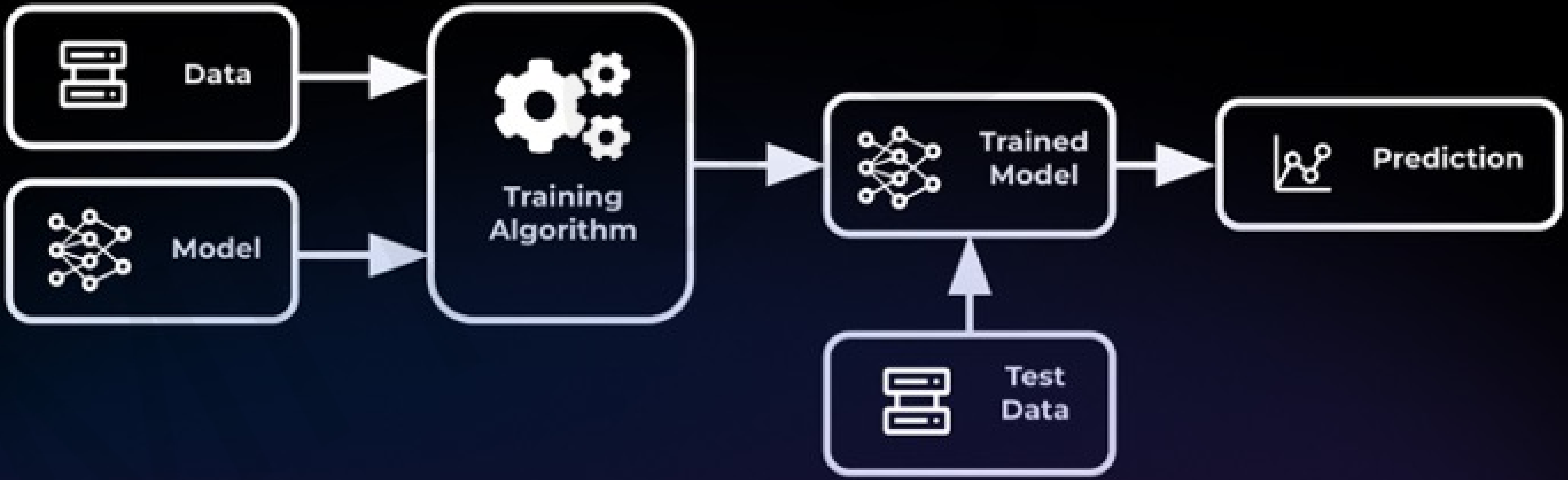
Uncertainty



Can we teach a model to recognize when it doesn't know the answer?

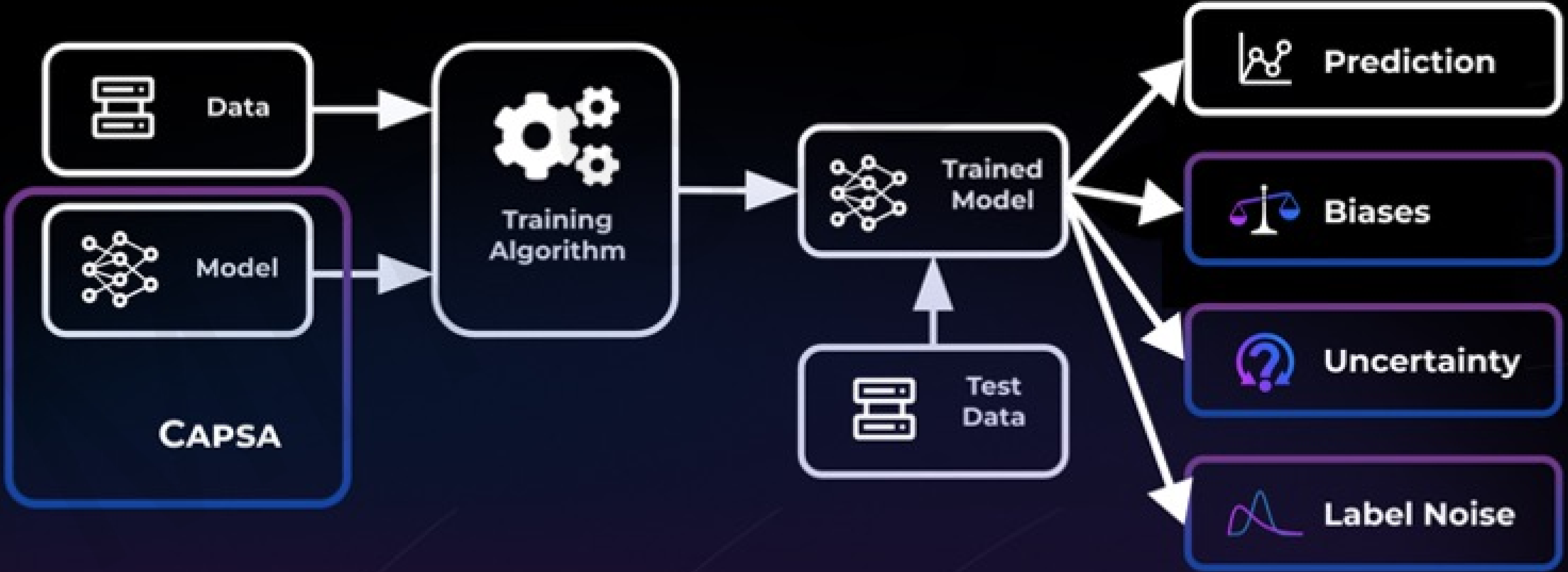
Using Risk-Awareness to Transform AI Workflows

CAPSA: A model-agnostic framework for risk estimation



CAPSA: A model-agnostic framework for risk estimation

A Data- And Model-Agnostic Neural Network Wrapper
For Risk-Aware Decision Making



*Capsa: Latin Root For A Capsule Or Container

Change the Future of Trustworthy AI With Us

```
train_data, test_data = load_dataset()
model = build_model(n_layers, n_neurons, ...)
model.train(train_data)
preds = model.predict(test_data)
```

```
train_data, test_data = load_dataset()
model = build_model(n_layers, n_neurons, ...)
model = capsa.HistogramWrapper(model, ...)
model.train(train_data)
preds, bias = model.predict(test_data)
```

CAPSA: A model-agnostic framework for risk estimation

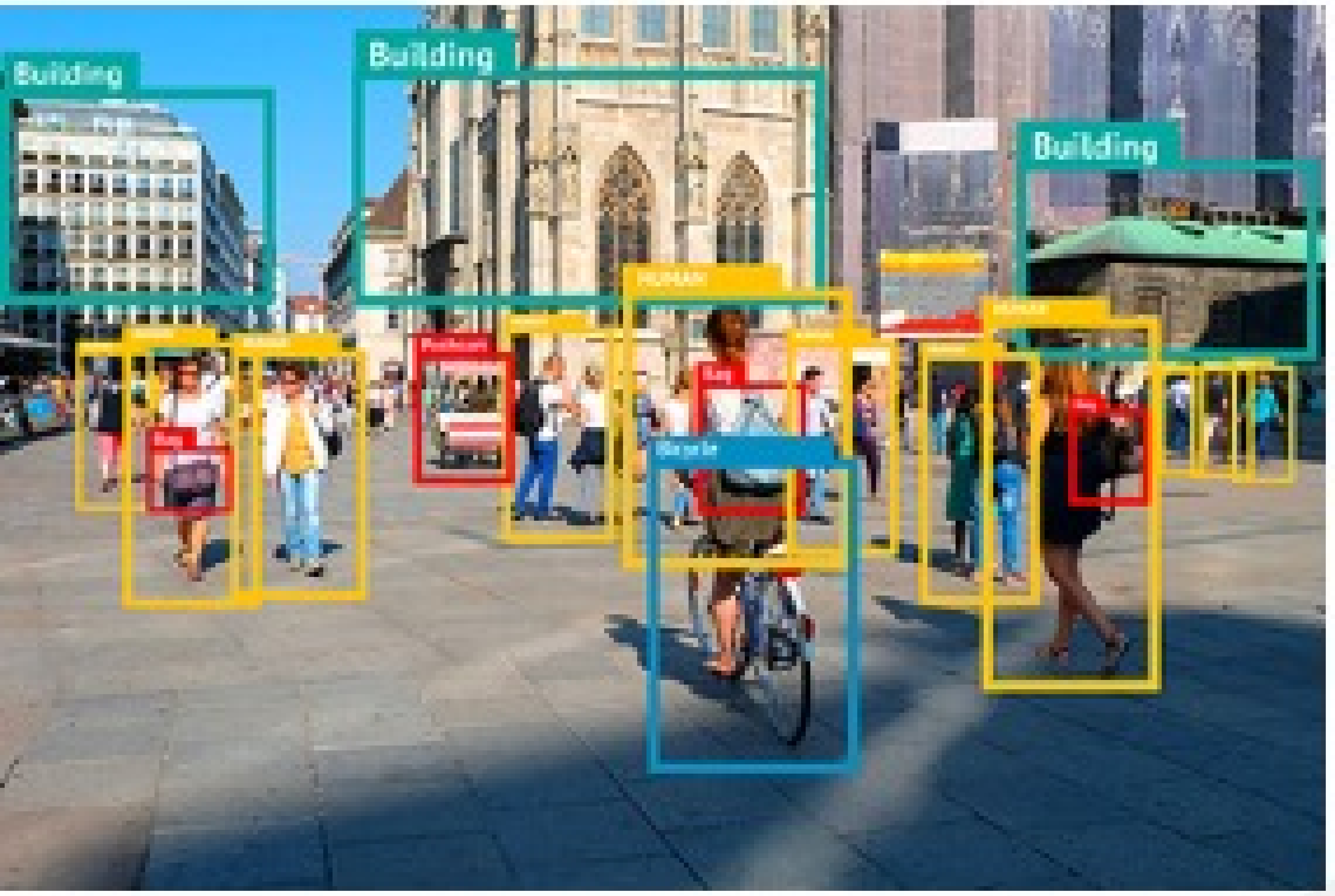
CAPSA "wraps" models so that they are **risk-aware** by changing and adding necessary components for each metric wrapper.

CAPSA: A model-agnostic framework for risk estimation

Directly plugs into existing training pipelines, providing insight into **bias** (density and imbalance) as well as **aleatoric** (data), and **epistemic** (model) uncertainty

Unlocking the Future of Trustworthy AI

Themis is unlocking the key to deploy deep learning safety across fields:

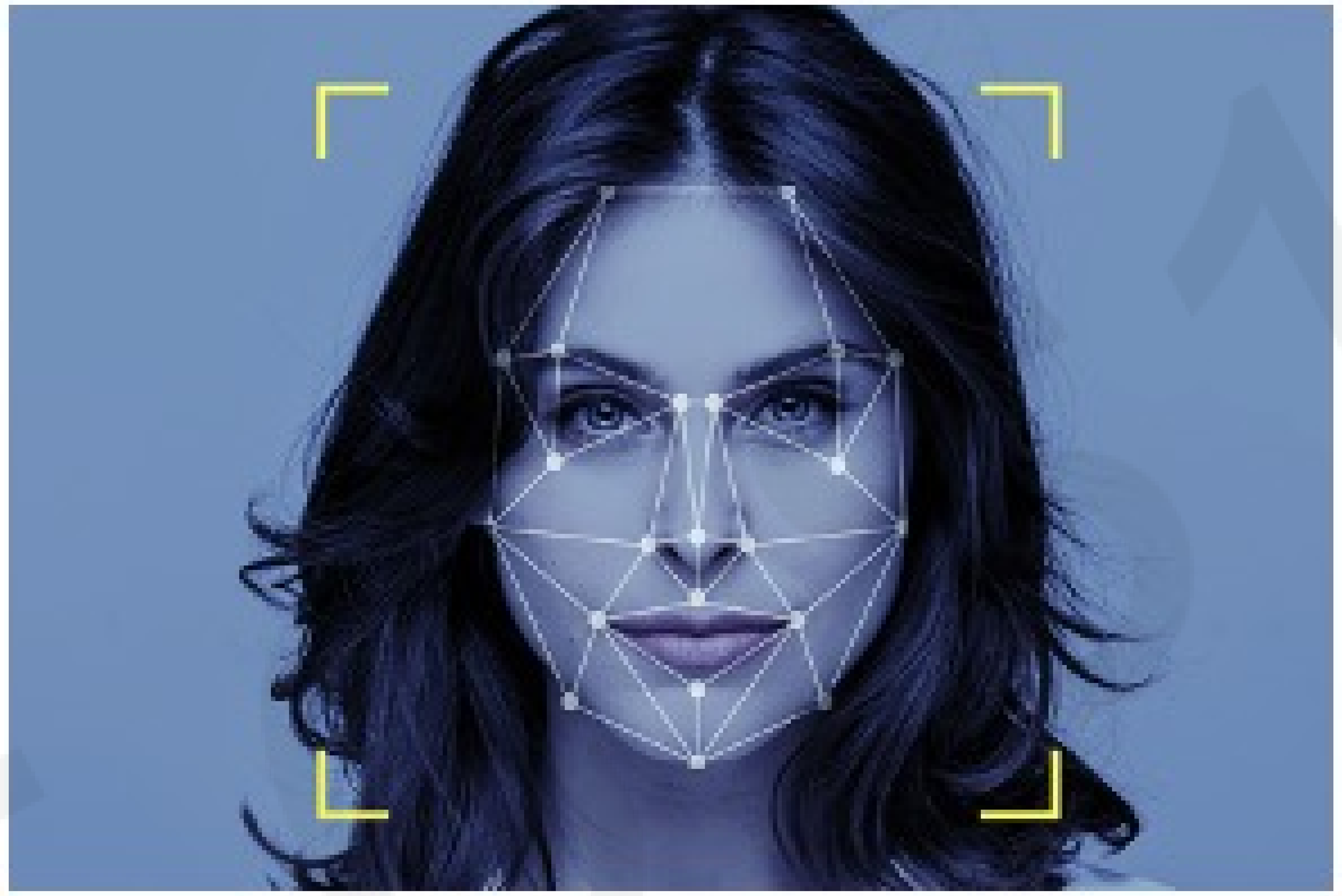


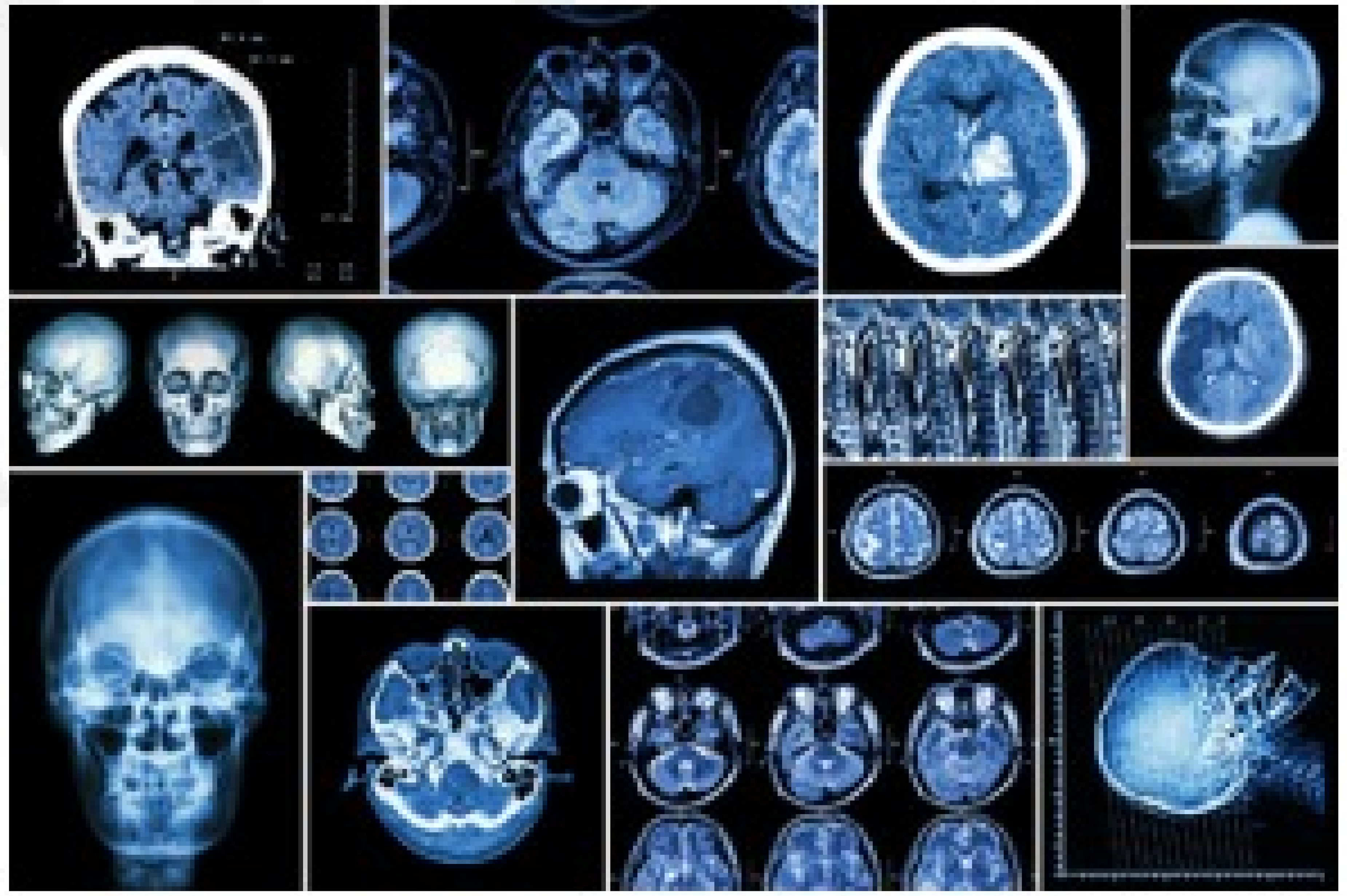
When should a human take control of an autonomous vehicle?

What types of data are underrepresented in commercial autonomous driving training pipelines?

Unlocking the Future of Trustworthy AI

Themis technology can answer safety-critical questions across fields:





When is a model uncertain about a life-threatening diagnosis?

What types of patients might drug discovery algorithms be biased against?

Today: How can we improve commercial facial detection systems?

Change the Future of Trustworthy AI Together With Us

Scientific Innovation

Transform AI Workflows

Open Source Tools

Global Industry Reach

We're Hiring!

careers@themisai.io

Apply!!!

introtodeeplearning.com/jobs.html