Uncertainty Quantification (UQ) in AI

Bellairs Workshop, December 15-18, 2025

Nicolas THOME – Prof at Sorbonne University

ISIR Lab, MLIA Team

On leave at ILLS Lab, ETS/Mila/McGill

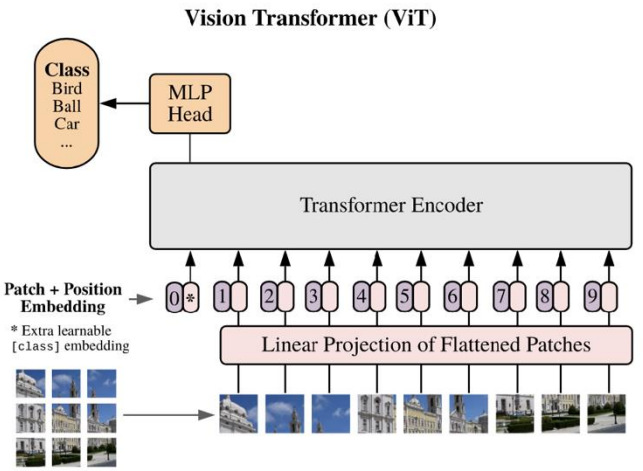
SORBONNE UNIVERSITÉ

ISIR INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE

MLIA Machine Learning & Deep Learning for Information Access

Context: AI/ML summer

- AI in the last decade: huge performance boost
  - Vision, NLP, multi-modal prediction robotics

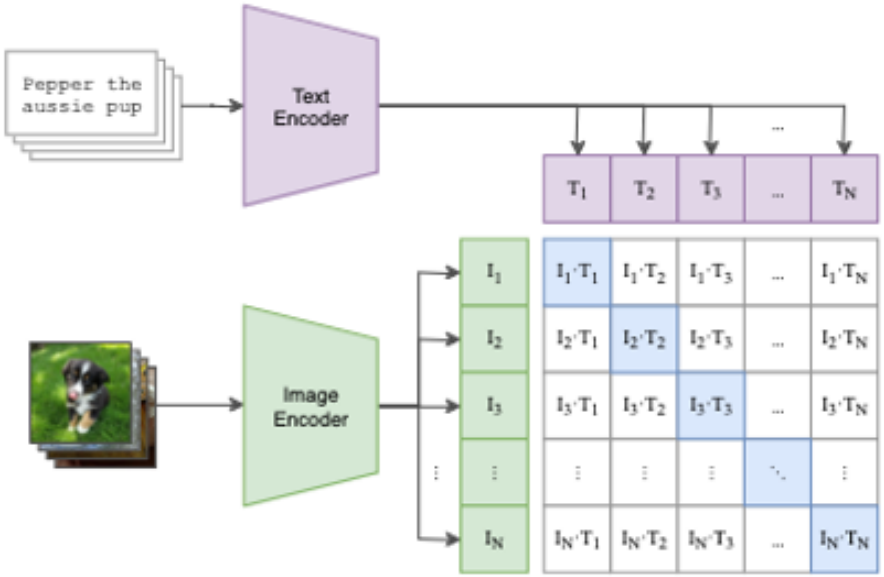


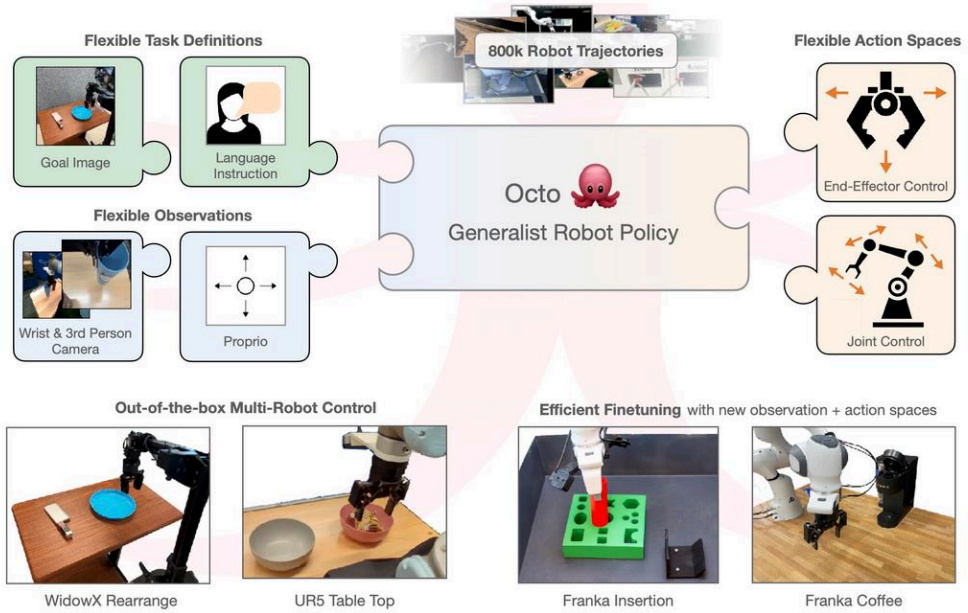
explain quantum theory to a child in the style of snoop dogg

Yo, little homie, let me break it down for you
Quantum theory, it's a crazy thing, it's true
See, the world is made up of tiny, tiny particles
That are always moving, and changing, and actin' wild

These particles, they don't always behave like we think
Sometimes they're waves, and sometimes they're points, it's a trip
They can be in two places at once, and they can be in two states
It's a mind-blowing concept, but that's the quantum fate

So don't be scared, little homie, it's all good
Quantum theory, it's just the way the world works, it's understood
It's a wild ride, but it's a cool one too
Quantum theory, it's the way the world does what it do.



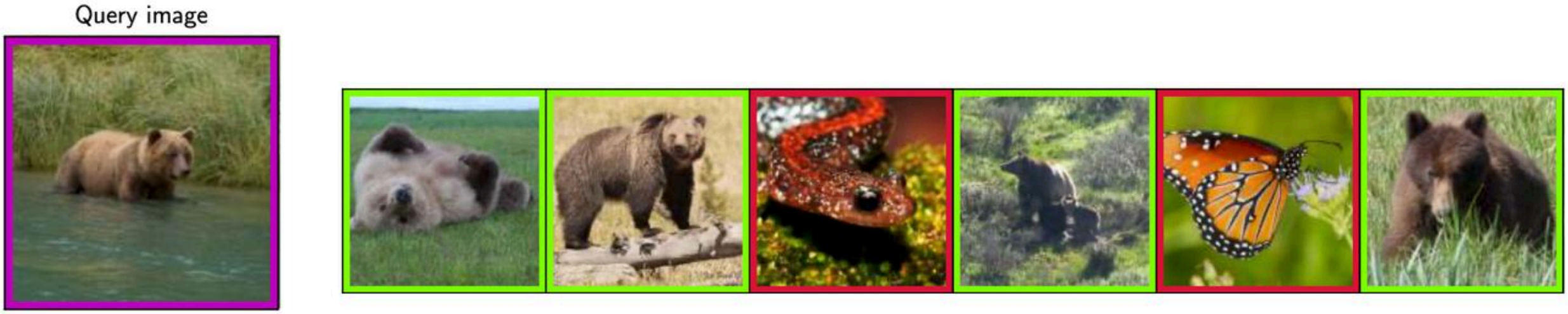


Context: robustness in deep learning

Several brittleness aspects in deep learning models

- Explainability, biases & shortcuts, fairness, *etc*

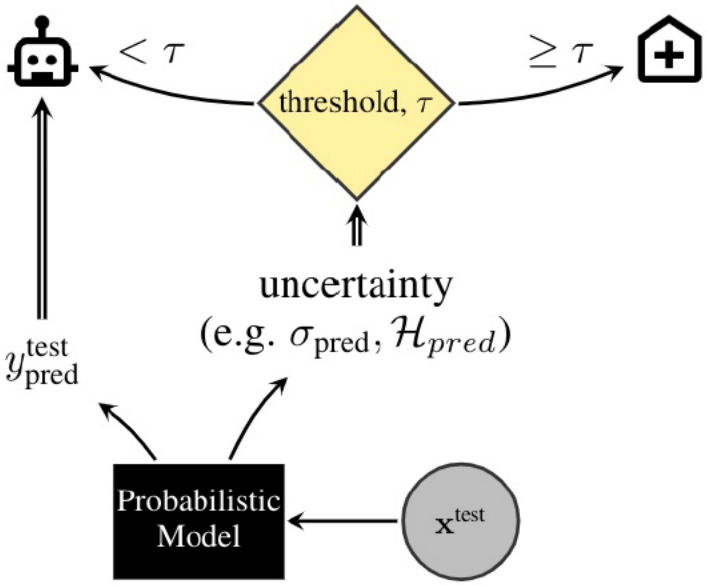
Stability: adversarial examples, mistake severity



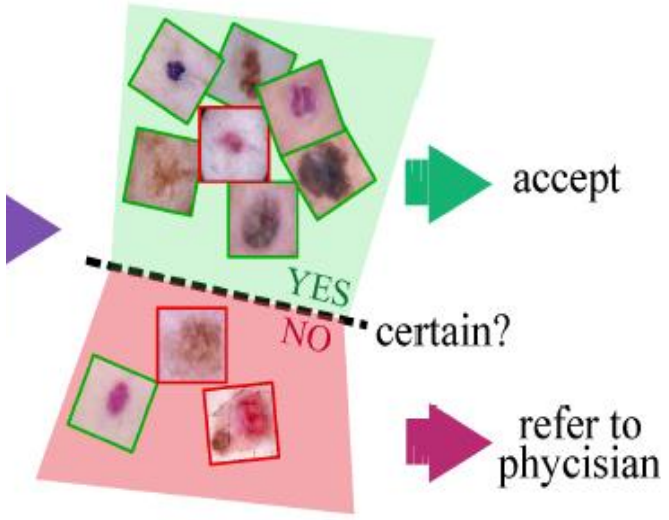
Context: robustness in deep learning

Uncertainty Quantification (UQ)

“Know when you do not know”



Abstain to make a prediction

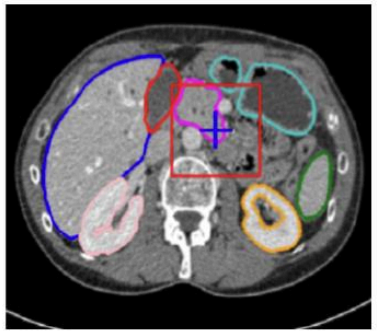
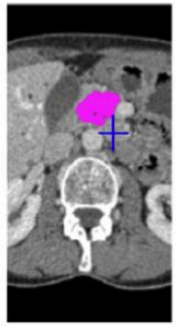


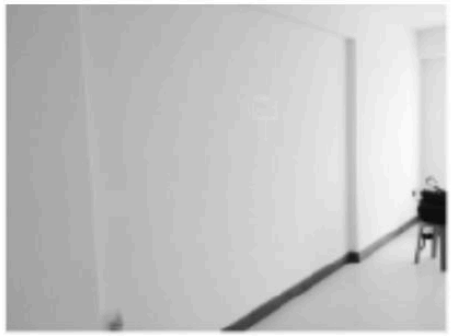
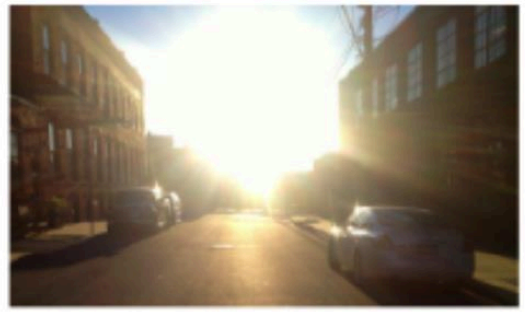
UQ: a challenge in DL

- **Which uncertainty score?**
- **What type of uncertainty: aleatoric, epistemic?**
- **Which tasks: calibration, failure prediction, anomaly detection?**

Sources of uncertainty

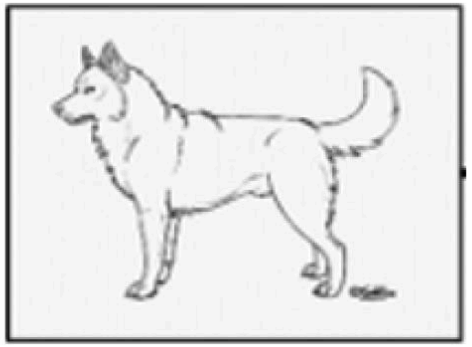
- Aleatoric uncertainty: data
  - Class confusion, ambiguous data, sensor noise

- Epistemic uncertainty: model
  - Distribution shift in p(x,y), *e.g.* x (snow, image->cartoon), or y (open set, new classes)



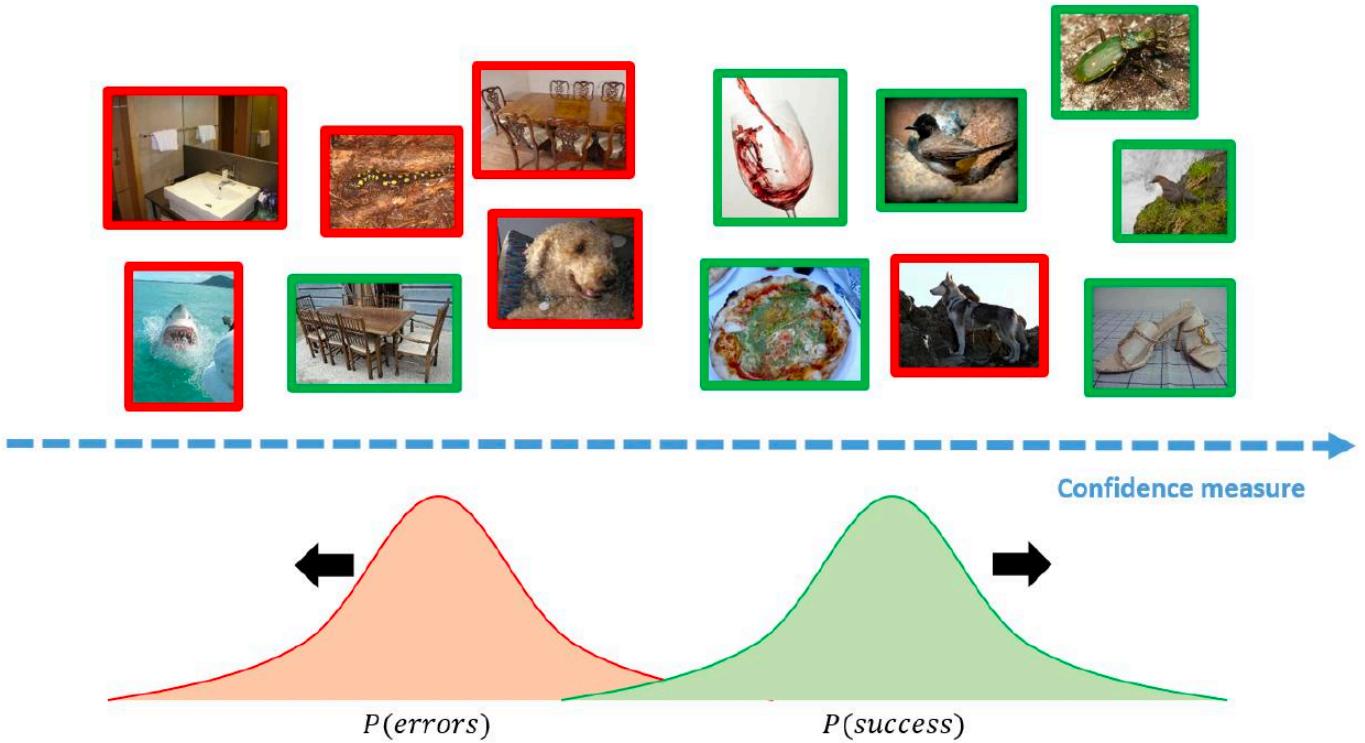



Uncertainty Quantification in deep learning

1. Calibration
2. Out-of-distribution / anomaly detection
3. Failure Prediction

Uncertainty <-> confidence quantification

- Confidence estimate $\hat{C}(\boldsymbol{x}_i)$ goal: **distinguish correct from erroneous predictions**



- Sort examples wrt $\hat{C}(\boldsymbol{x}_i)$
  - Evaluate capacity of $\hat{C}$ to assign larger prediction values for correct predictions than for errors

Calibration

Calibrated probabilities: estimated confidence C(x) ⇔ accuracy

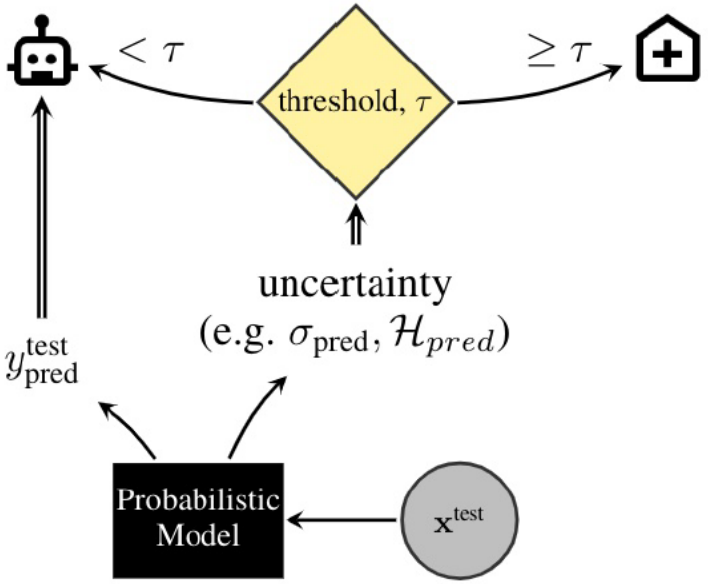
- Test **Data**: $\mathcal{D} = (X, Y) = \{(x_1, y_1), \ldots, (x_N, y_N)\}$
- $(\hat{y}_i, \hat{C}(\boldsymbol{x}_i))$ class prediction and confidence level
- Perfect calibration:

$$p\left(\hat{Y} = Y | \hat{C} = p\right) = p, \quad \forall p \in [0, 1]$$

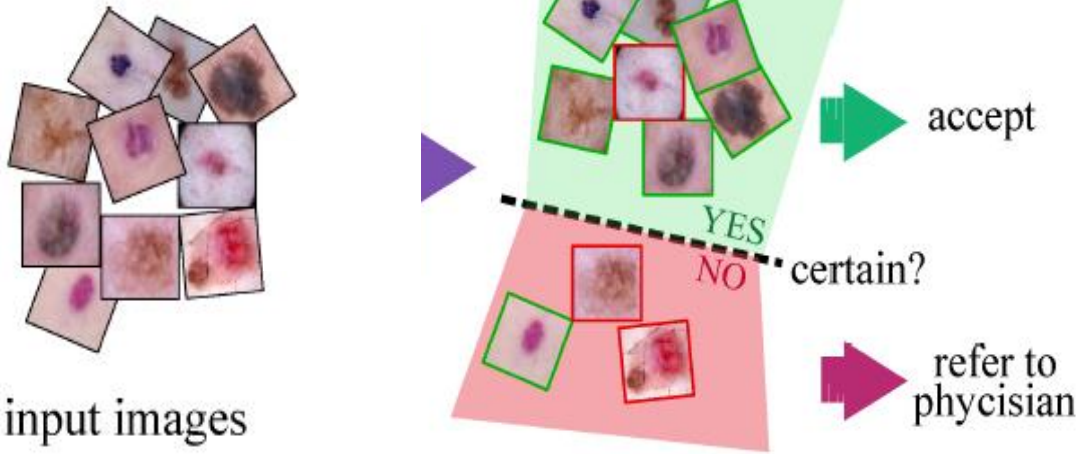
- Predicted confidence match actual accuracy
  - *e.g. given 100 predictions, each with confidence of 0.8, we expect that 80 should be correctly classified.*

Why does calibration matters?

“Know when you do not know”



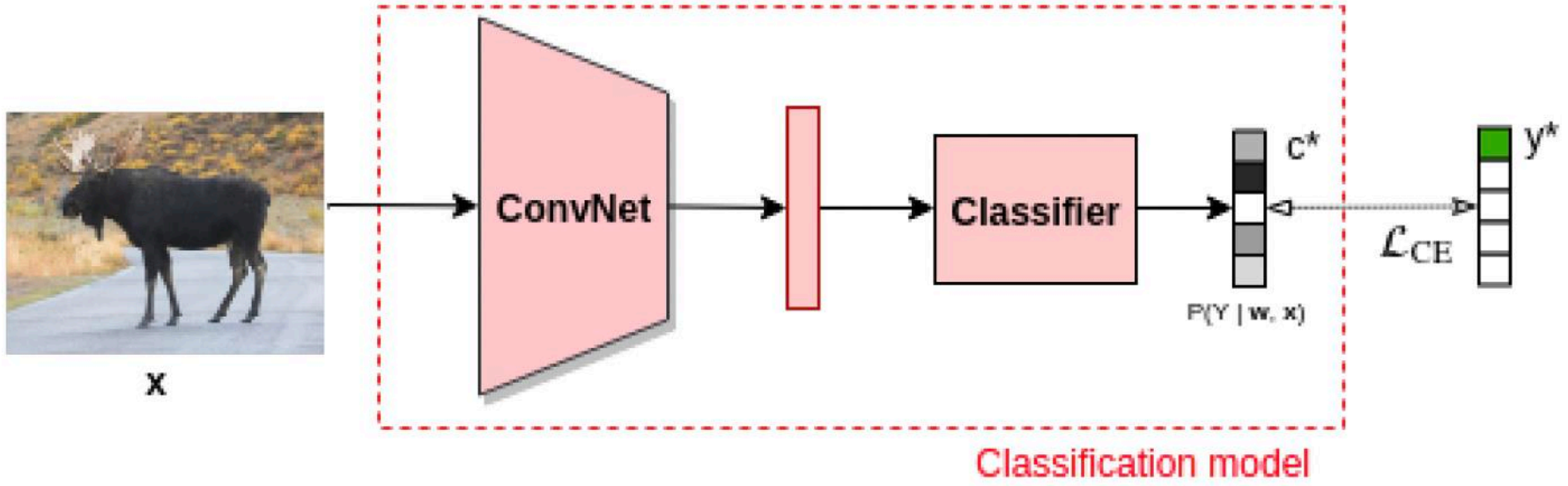
Abstain to make a prediction



- Predicted confidence c(x) = accuracy => c(x) > τ: statistical guarantees

Baseline UQ score: classification example

- Classification model trained on $\mathcal{D} = \{(\boldsymbol{x}_i, y_i^*)\}_{i=1}^N$



- Model prediction: $\hat{\boldsymbol{y}} = \arg\max_{k \in \mathcal{Y}} p(Y = k | \boldsymbol{w}, \boldsymbol{x})$
- Model confidence $\hat{C}(\boldsymbol{x})$:
  - Simple baseline for deep neural networks: $\boldsymbol{MCP}(\boldsymbol{x}) = \max\limits_{k \in \mathcal{Y}} p(Y = k | \boldsymbol{w}, \boldsymbol{x})$

Estimating calibration of confidence scores

Reliability Diagram

M interval bins.

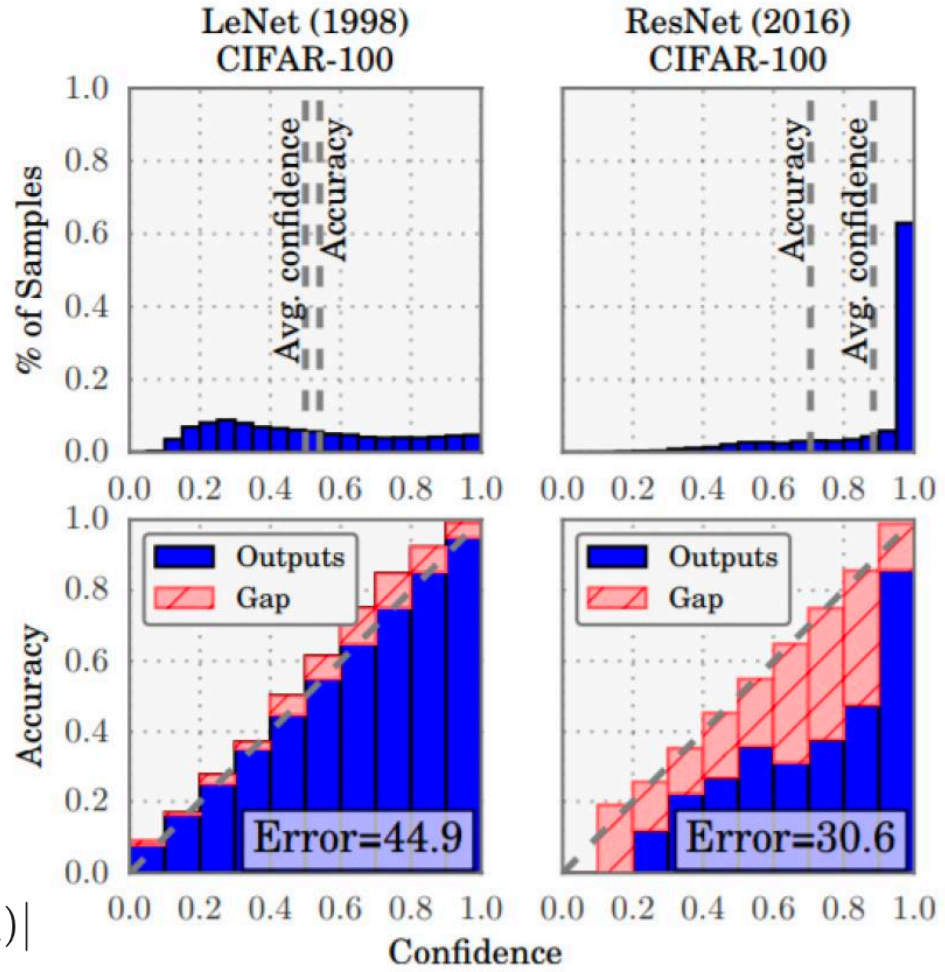
B_m set of samples whose predictions are in $I_m = (\frac{m-1}{M}, \frac{m}{M}]$

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} 1(\hat{y}_i = y_i)$$

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$$

perfect calibration

$$\iff acc(B_m) = conf(B_m) \quad \forall m$$



Expected calibration error: $\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \cdot |\mathrm{Acc}(B_m) - \mathrm{Conf}(B_m)|$

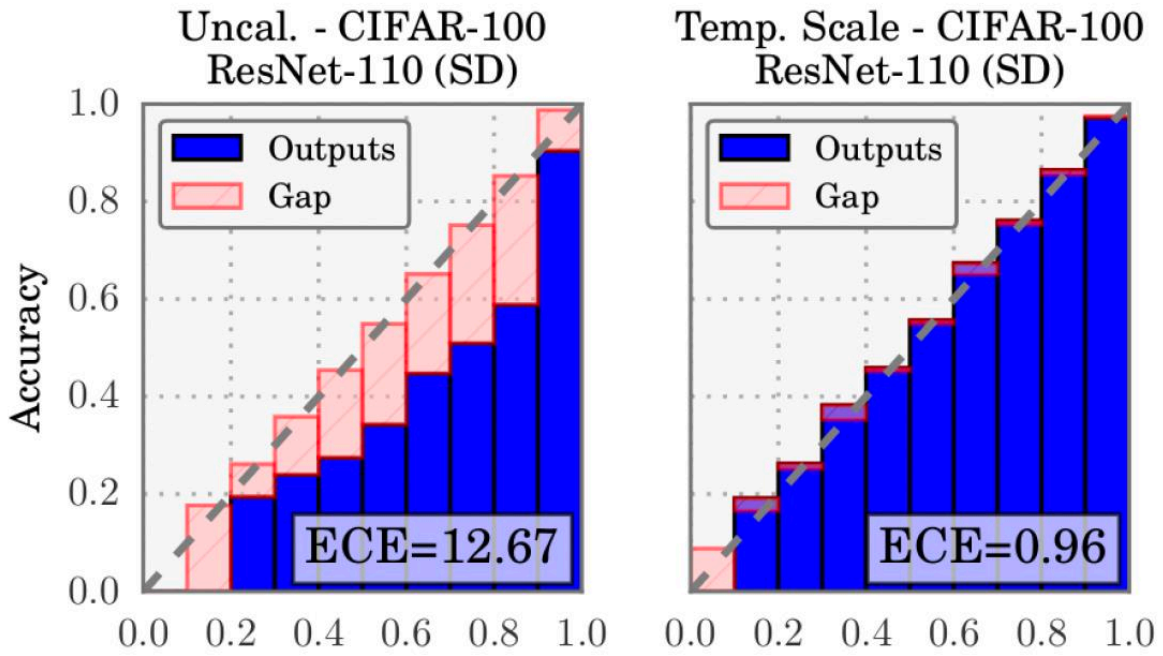
[Guo et al., 2017] **showed that modern neural networks are no longer well-calibrated!**

Post-hoc calibration

- Simple solution to over-confident prediction: temperature scaling [Guo et al., 2017]

$$P(\hat{y}_k) = \frac{e^{s_k/T}}{\sum_{k'=1}^{K} e^{s_{k'}/T}}$$

- temperature T optimized on val set s.t. $acc(B_m) = conf(B_m)$



How to evaluate calibration?

- Expected Calibration Error (ECE) for calibration only
- Proper scoring rules S(p,q):
  - S(p,q) is minimized of p=q. Ex: NLL or MSE for classification or regression
  - Using them for evaluating prediction models: include both discrimination (accuracy, AUC) and calibration
  - Brier Score (BS) & Continuous Ranked Probability Score (CRPS) for continuous outputs (regression)

$$\mathrm{BS} = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2$$

  - Negative Log-Likelihood (NLL) - Negative Log Predictive Density (NLPD)

Learning calibrated models

- **Fighting overconfidence:** controlling logit's scale, e.g., [Murugesan et.al., 2024]

$$\min_{\theta} \quad \mathcal{L}_{CE} + \lambda \sum_{k} |\tau_k - l_k|$$

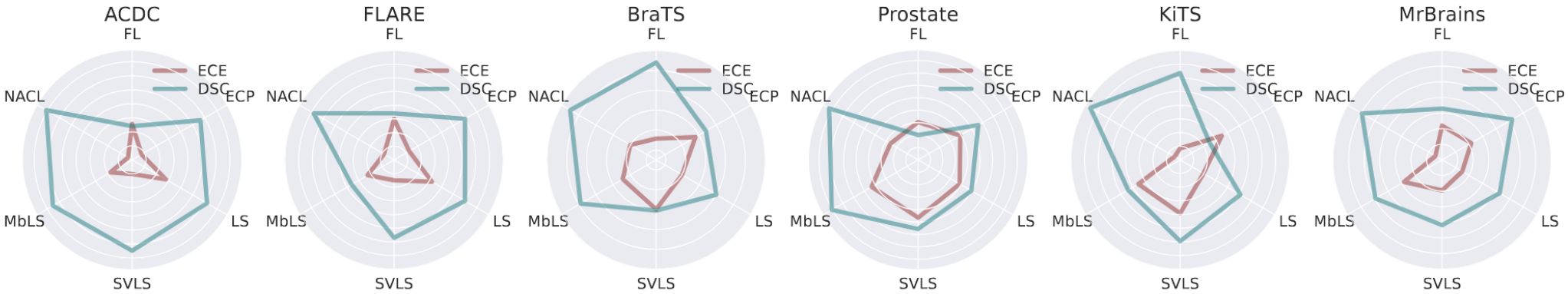


Figure 1: **Compromise between calibration and discriminative performance.** For each dataset, we show the discriminative (DSC) and calibration (ECE) results obtained by each method. We expect a *well-calibrated* model to achieve simultaneously large DSC (*in blue*) and small ECE (*in brown*) values.

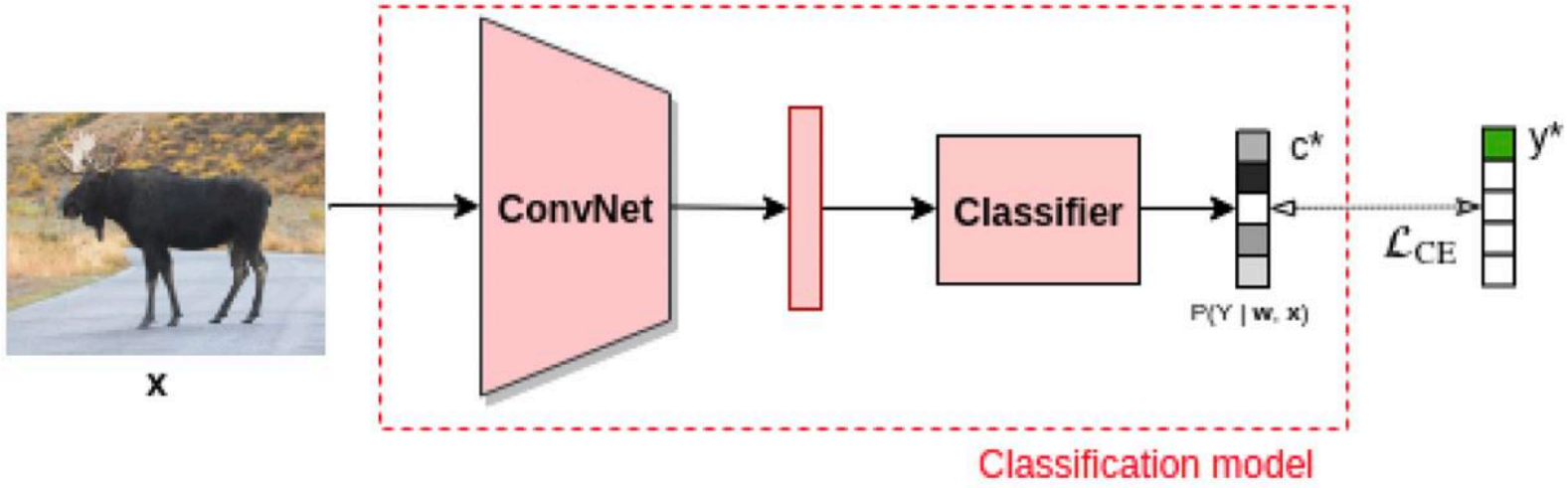
[Murugesan et.al., 2024]Neighbor-Aware Calibration of Segmentation Networks with Penalty-Based Constraints. Balamurali Murugesana, Sukesh Adiga Vasudevaa, Bingyuan Liub, Herve Lombaerta, Ismail Ben Ayeda, Jose Dolz. Medical Image Analysis, 2024

Uncertainty Quantification in deep learning

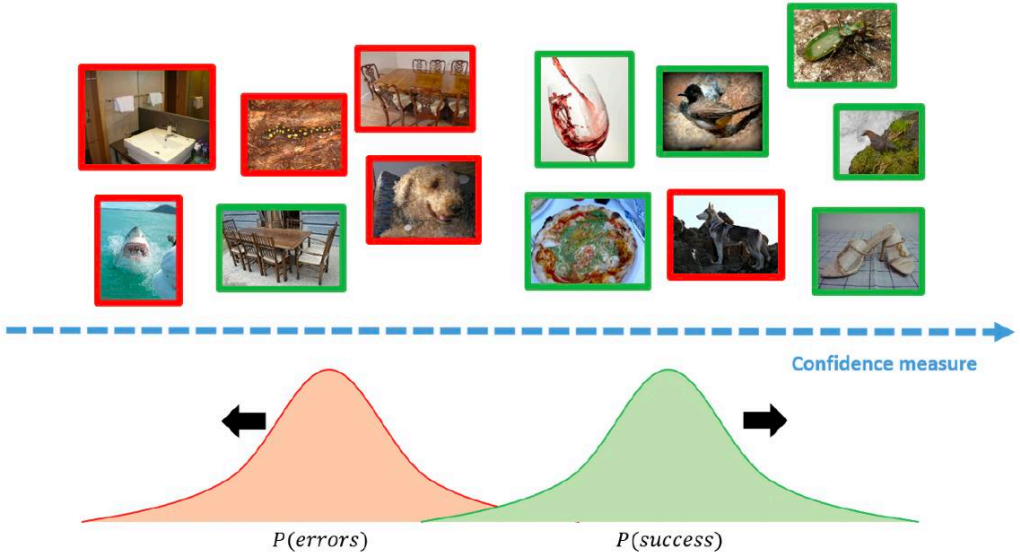
1. Calibration
2. Failure Prediction
3. Out-of-distribution / anomaly detection

Failure prediction in classification

- **<u>Post-hoc task:</u>** UQ of a pre-trained model to detect its own errors



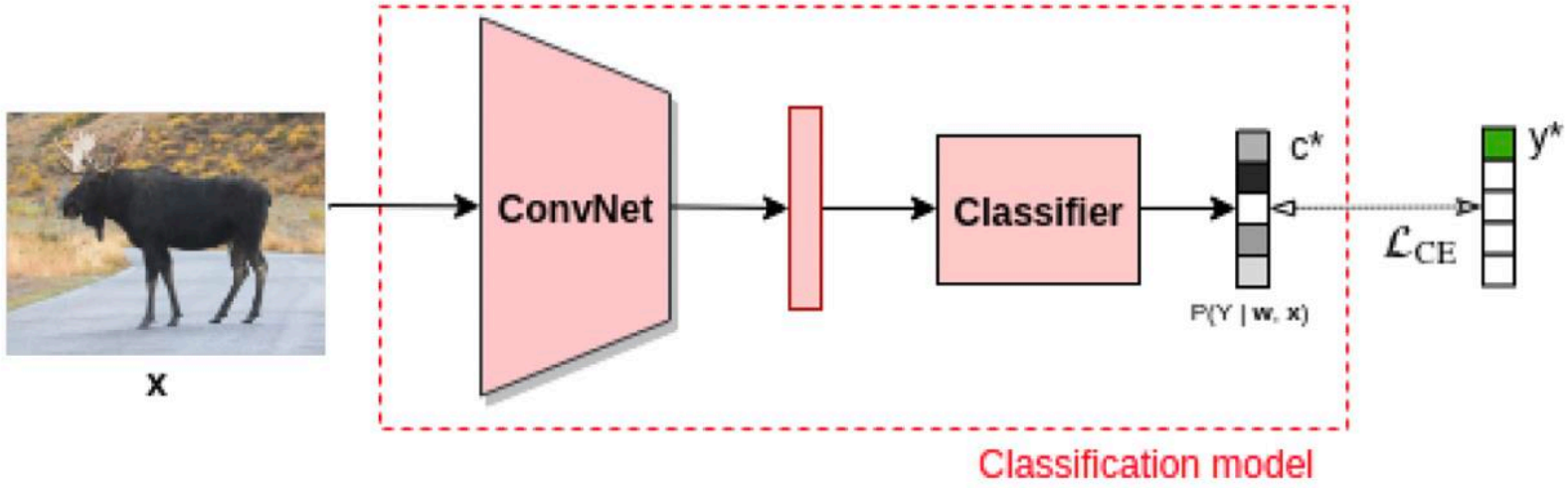
- A binary classification problem: correct vs incorrect prediction



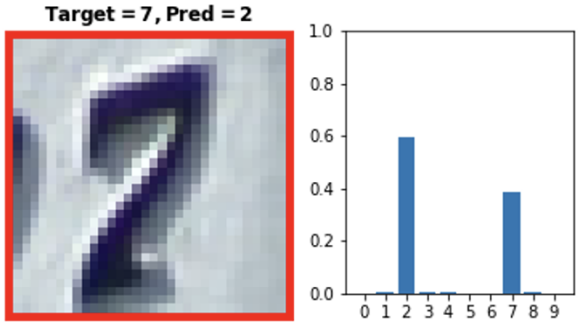
- Imbalance => evaluated by ranking metrics: AUC ROC, AUPR, FPR95, etc

Baseline methods for failure prediction

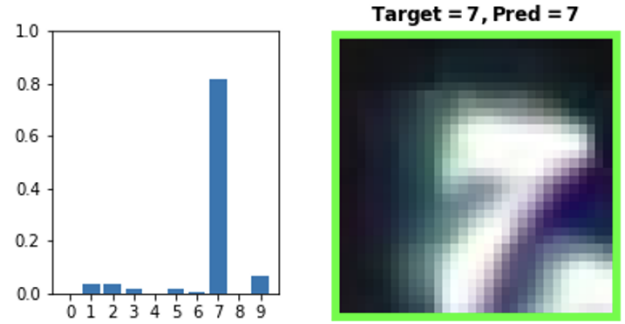
- $MCP(x) = \max_{k \in \mathcal{Y}} p(Y = k | \boldsymbol{w}, \boldsymbol{x})$
  - Overconfident by design



- Entropy: class-permutation invariant
  - Not targeted for failure prediction



Erroneous prediction, $\mathbb{H}(Y \mid \boldsymbol{x}, \boldsymbol{\theta}) = 0.79$



Correct prediction, $\mathbb{H}(Y \mid \boldsymbol{x}, \boldsymbol{\theta}) = 0.79$

$$\mathbb{H}(Y \mid \boldsymbol{x}, \boldsymbol{\theta}) \coloneqq -\sum_{k \in \mathcal{Y}} P(Y = k \mid \boldsymbol{x}, \boldsymbol{\theta}) \cdot \log P(Y = k \mid \boldsymbol{x}, \boldsymbol{\theta})$$

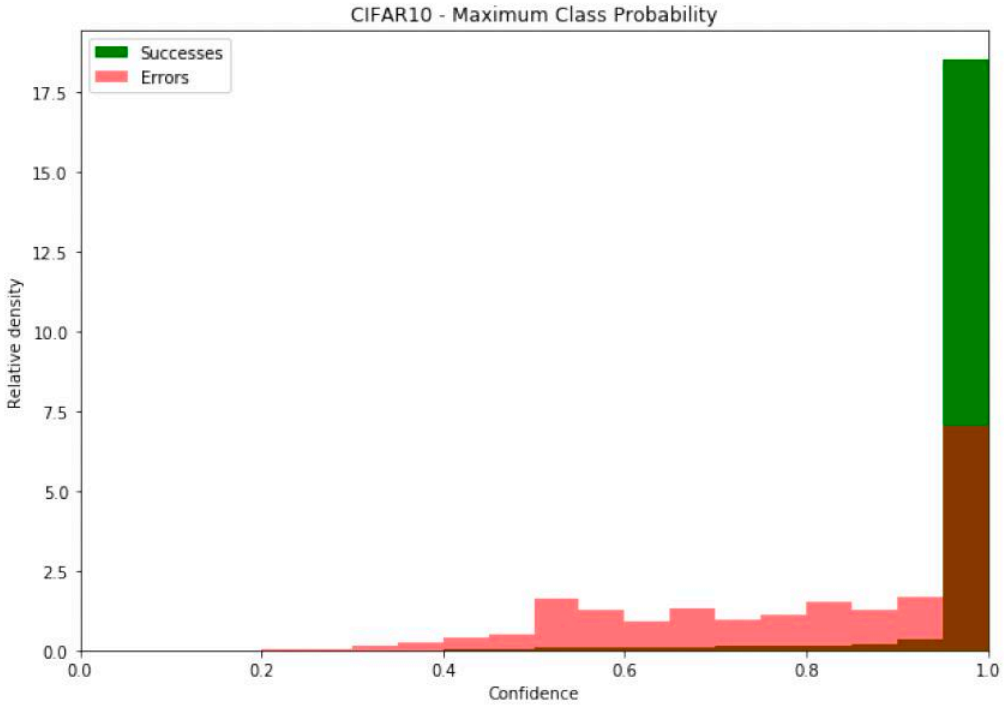
Failure prediction methods

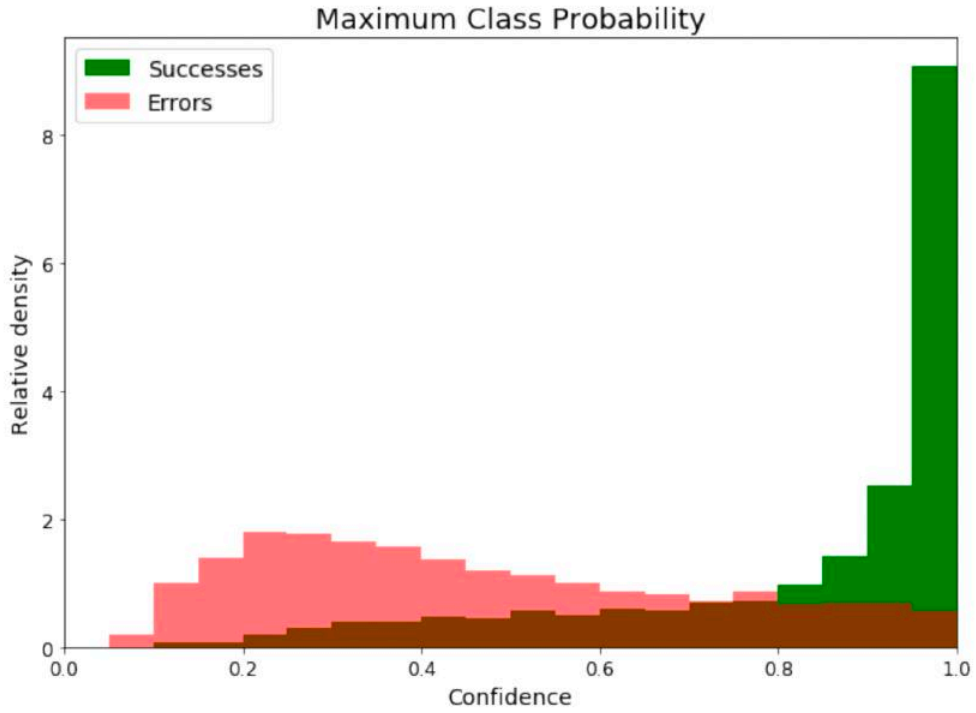
- DOCTOR [GRG+21] :
  - $D_\alpha(\mathbf{x}, \gamma) \sim s_{\text{gini}}(\mathbf{x}) \triangleq 1 - \sum_{y \in \mathcal{Y}} (\hat{\mathbf{p}}(\mathbf{x})_y)^2$ => confidence := $||\hat{P}||^2$
  - $D_\beta(\mathbf{x}, \gamma)$ ~ MCP

- Theoretical bounds for failure prediction
  - Gini $\mathbf{D_\alpha}$: still class-permutation invariant

[GRG+21] F. Granese, M. Romanelli, D. Gorla, C. Palamidessi, P.Piantanida. DOCTOR: A simple method for detecting misclassification errors. NeurIPS'21

Calibration & Failure Prediction



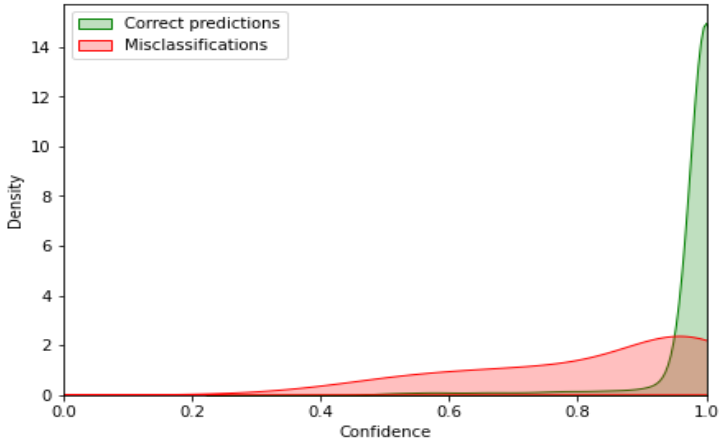


Calibration: absolute score vs

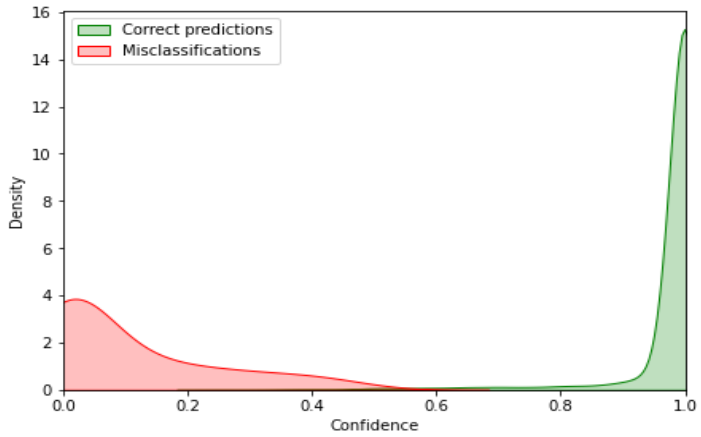
Failure prediction: relative score to distinguish correct from in correct predictions

Criterion for Failure Prediction

- **Idea [CTB+19]: True Class probability (TCP)** instead Maximum Class Probability (MCP) $\mathrm{TCP} : P(Y = y^* | \mathbf{w}, \mathbf{x})$
- TCP better distinguish correct prediction from incorrect predictions MCP for failure prediction
  - Theoretical guarantees: $\mathrm{TCP}(\mathbf{x}, y^*) > 1/2 \Rightarrow \hat{y} = y^* \quad \mathrm{TCP}(\mathbf{x}, y^*) < 1/K \Rightarrow \hat{y} \neq y^*$



MCP



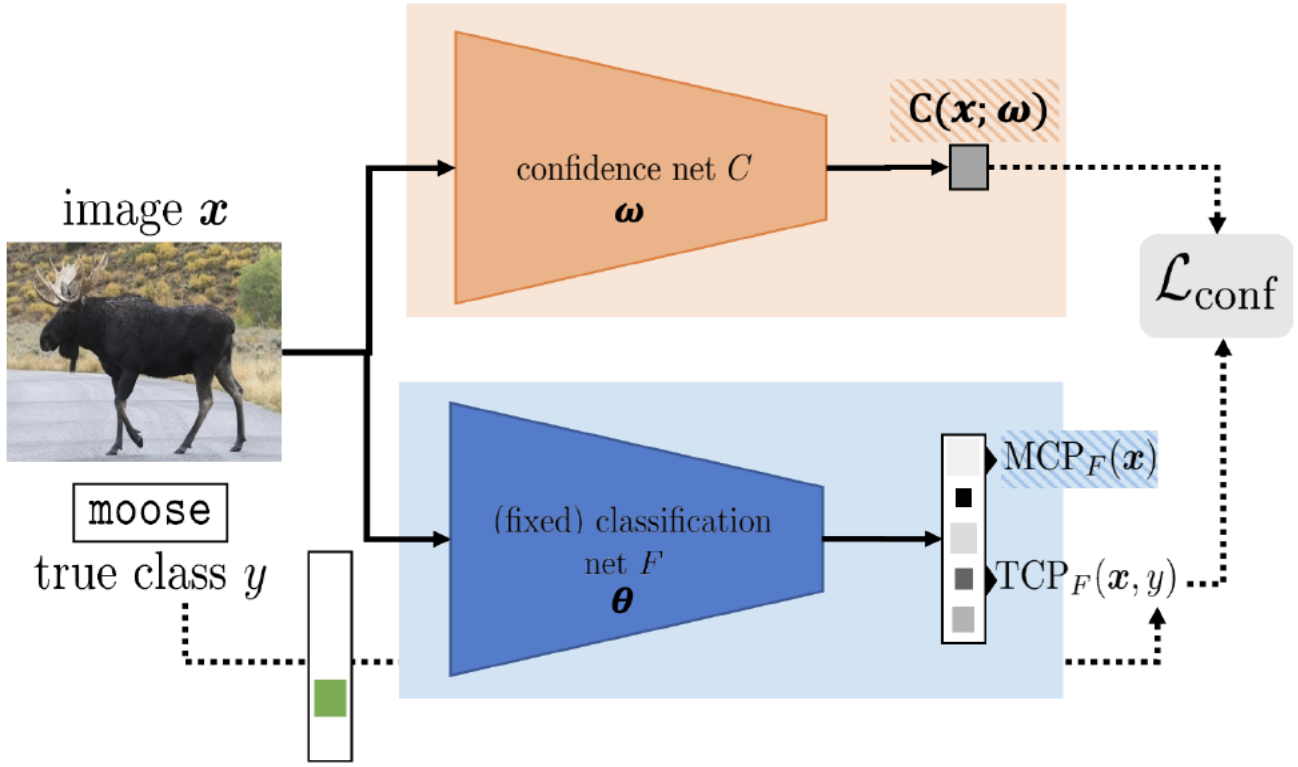
TCP

  - Perfect separation with $m(x) = P(y^* \mid x) - \max_{y' \neq y^*} P(y' \mid x)$

[CTB+19] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, P. Pérez. Addressing Failure Detection by Learning Model Confidence. NeurIPS 2019.

Learning Confidence

TCP unknown at test time: learning it! => ConfidNet



- Pre-trained prediction model (blue)
- Learning to regress TCP with an auxiliary model (orange)

$$\mathcal{L}_{conf}(\theta;\mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}(\hat{c}(\boldsymbol{x}_i,\theta) - c^*(\boldsymbol{x}_i,y_i^*))^2$$

Same idea in different contexts: [YK19, K24] => **Learning Visual Uncertainties (LVU)**

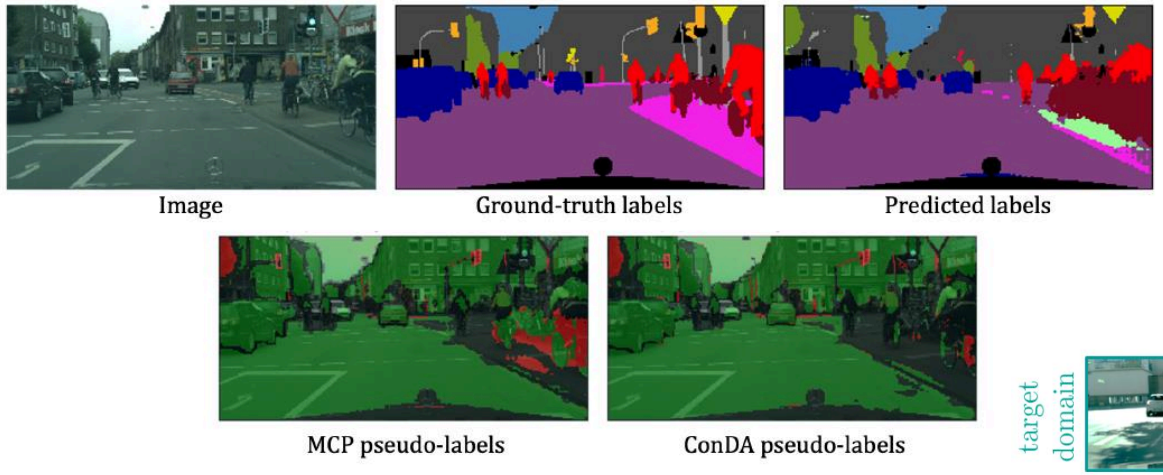
[CTB+19] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, P. Pérez. Addressing Failure Detection by Learning Model Confidence. NeurIPS 2019.

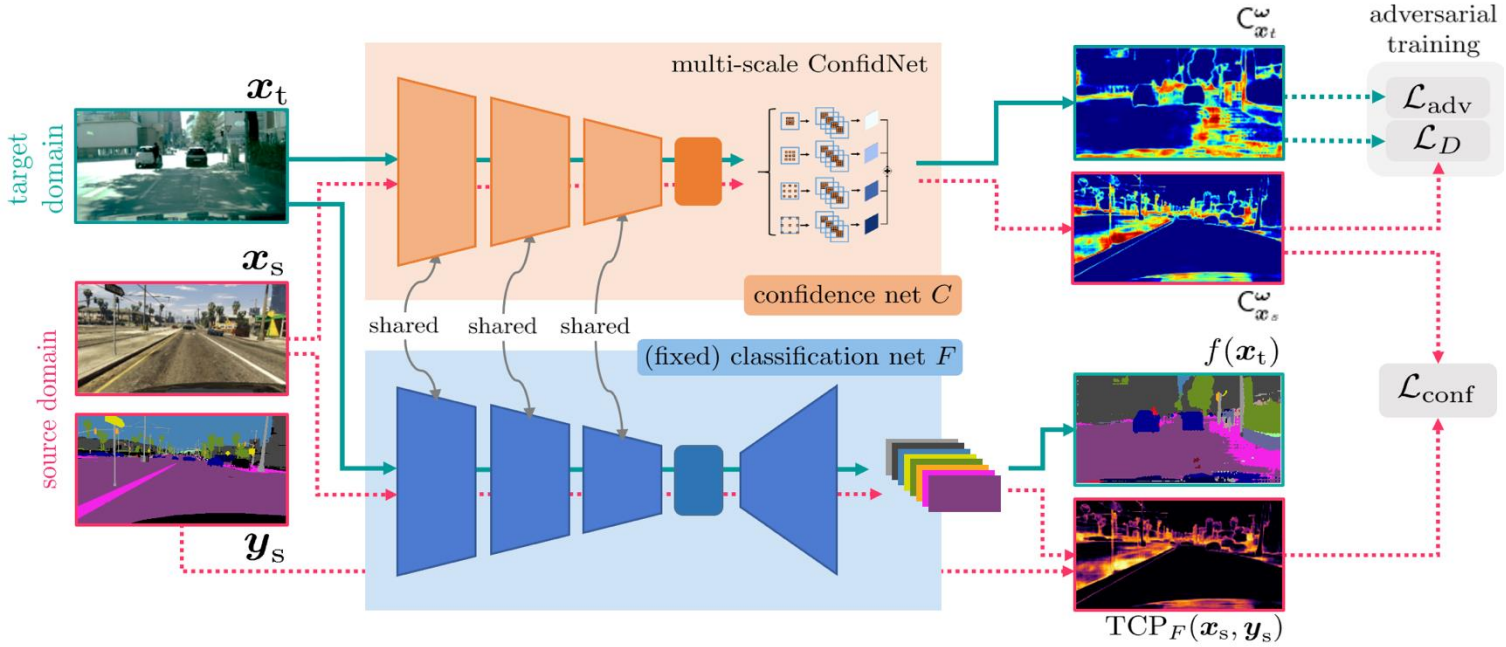
[YK19] D. Yoo and In So Kweon. Learning loss for active learning. CVPR 2019

[K24] Kirchhof, Michael, et al. "Pretrained visual uncertainties." arXiv preprint 2024

Learning confidence for self-labelling

- Extension for domain adaptation [CTS+21]

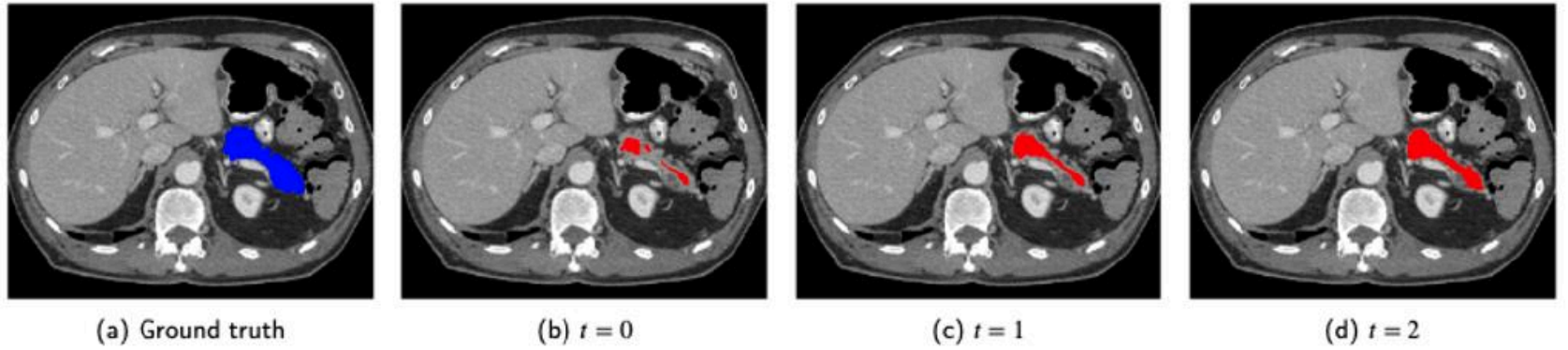




[CTS+21] C. Corbière, N. Thome, A. Saporta, T-H. Vu, M. Cord, P. Pérez. Confidence Estimation via Auxiliary Models. IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI), vol. 44, no. 10, pp. 6043-6055, June 2021.

Learning confidence for self-labelling

- Extension for Medical image segmentation [PTS21]



[PTS21] O. Petit, N. Thome, L. Soler. 3D Spatial Priors for Semi-Supervised Organ Segmentation with Deep Convolutional Neural Networks. International Journal of Computer Assisted Radiology and Surgery, Springer Verlag, In press, 2021.

Learning class distance matrix in classification

- Relative Uncertainty (Rel-U) [DRG+24]:
  - Learn a trainable distance matrix : $s_d(\boldsymbol{x}) = \widehat{\mathbf{p}}(\mathbf{x})\, D\, \widehat{\mathbf{p}}(\mathbf{x})^\top$
  - Optimize to separate errors/correct predictions;

$$\mathcal{L}(D) \triangleq (1-\lambda)\cdot \mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_+)\, D\, \widehat{\mathbf{p}}(\mathbf{X}_+)^\top\right] - \lambda \cdot \mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_-)\, D\, \widehat{\mathbf{p}}(\mathbf{X}_-)^\top\right]$$

$$\text{s.t.} \begin{cases} d_{ii} = 0, & \forall i \in \mathcal{Y} \\ d_{ij} \geq 0, & \forall i,j \in \mathcal{Y} \\ d_{ij} = d_{ji}, & \forall i,j \in \mathcal{Y} \\ \mathrm{Tr}(DD^\top) \leq K \end{cases}$$

[DRG+24] E. Dadalto, M. Romanelli, G. Pichler, and P. Piantanida. A data-driven measure of relative uncertainty for misclassification detection. ICLR'24.

Learning class distance matrix in classification

$$s_d(\boldsymbol{x}) = \widehat{\mathbf{p}}(\mathbf{x})\, D\, \widehat{\mathbf{p}}(\mathbf{x})^\top = \sum_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} d(y, y') \hat{\mathbf{p}}(\mathbf{x})_y \hat{\mathbf{p}}(\mathbf{x})_{y'}$$

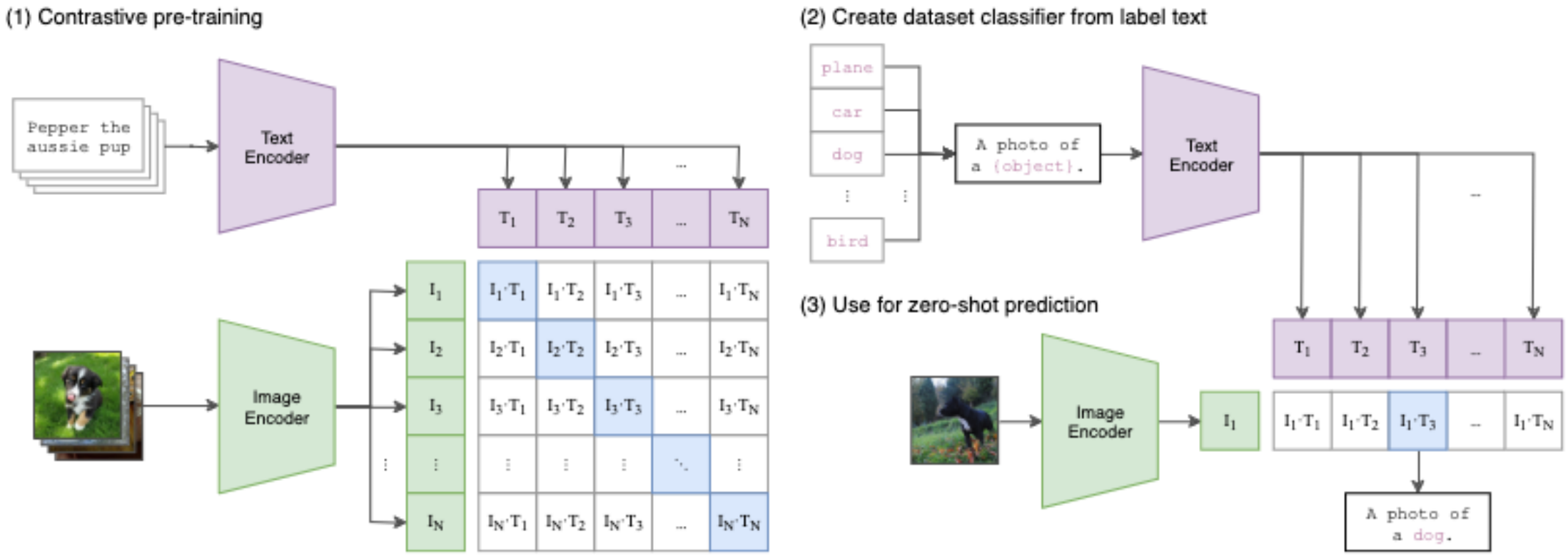
- Closed-form solution

$$D^* = \mathrm{ReLU}\left(\lambda \cdot \mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_-)^\top \widehat{\mathbf{p}}(\mathbf{X}_-)\right] - (1-\lambda) \cdot \mathbb{E}\left[\widehat{\mathbf{p}}(\mathbf{X}_+)^\top \widehat{\mathbf{p}}(\mathbf{X}_+)\right]\right)$$

- Distance matrix: limit the number of trainable parameters, suitable in few-shot context

UQ for Vision-Language Models (VLMs)

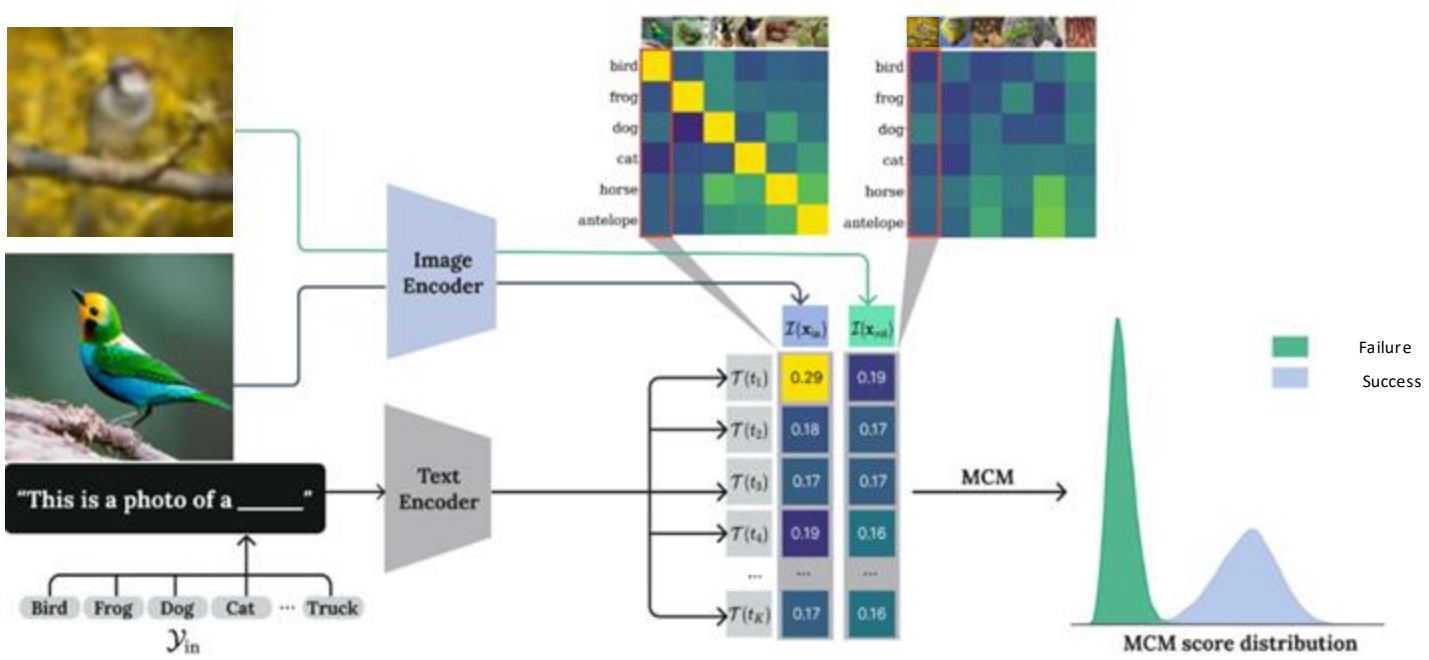
- Vision-Language Models: e.g., text—image (CLIP) [3]



[3] Learning Transferable Visual Models From Natural Language Supervision. 1. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever. ICML 2021

Failure Prediction with Vision Language Models (VLMs)

- **Maximum Concept Matching** (MCM) = Probability of predicted class with VLMs
  - MCP Extension to VLMs



- Pros:
  - Strong baseline
  - No training required
- Cons:
  - Overconfident by design for errors
  - Limited adaptability

- Learning Visual Uncertainties (LVU): don't take into account text uncertainties

[CTB+19] C. Corbière, N. Thome, A. Bar-Hen, M. Cord, P. Pérez. Addressing Failure Detection by Learning Model Confidence. NeurIPS 2019.

[YK19] D. Yoo and In So Kweon. Learning loss for active learning. CVPR 2019

[K24] Kirchhof, Michael, et al. "Pretrained visual uncertainties." arXiv preprint 2024

Taking task complexity into account

What is the uncertainty associated with this image ?

→ It depends on the task

<u>Task 1:</u> cat *vs* dog classification

- Low class confusion
- Low uncertainty

Taking task complexity into account

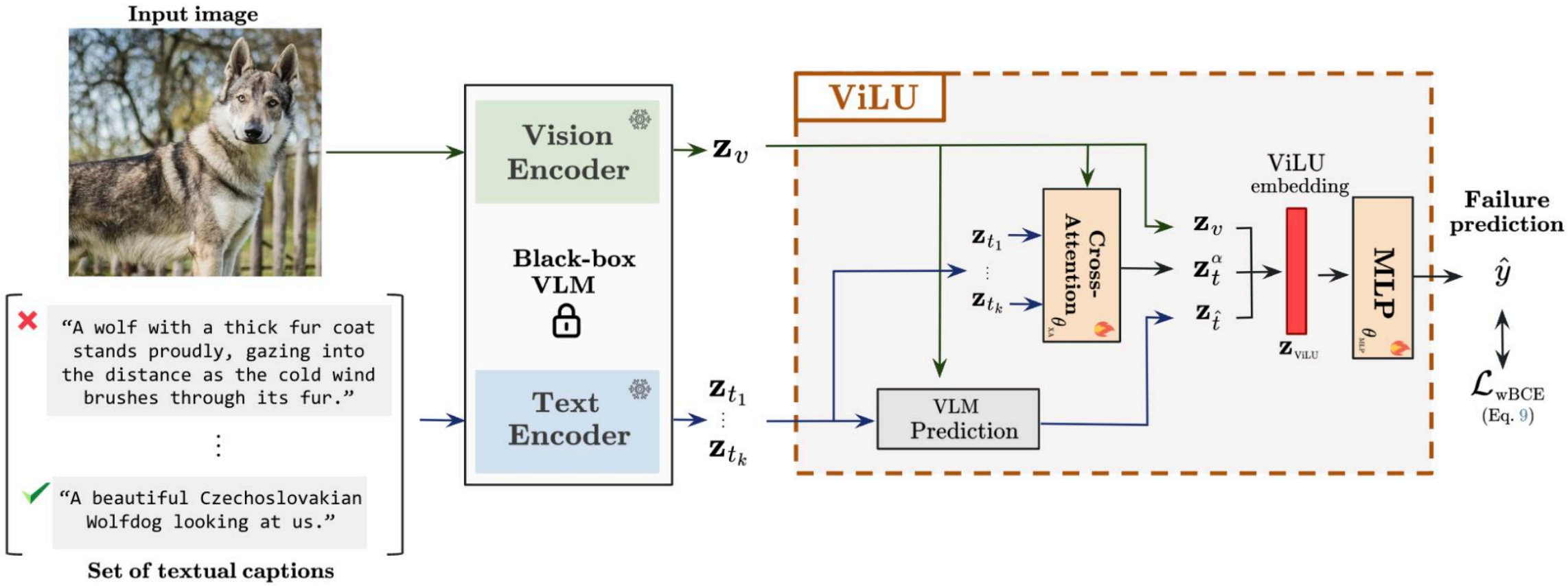
What is the uncertainty associated with this image ?

→ It depends on the task

Task 2: Dog breed classification

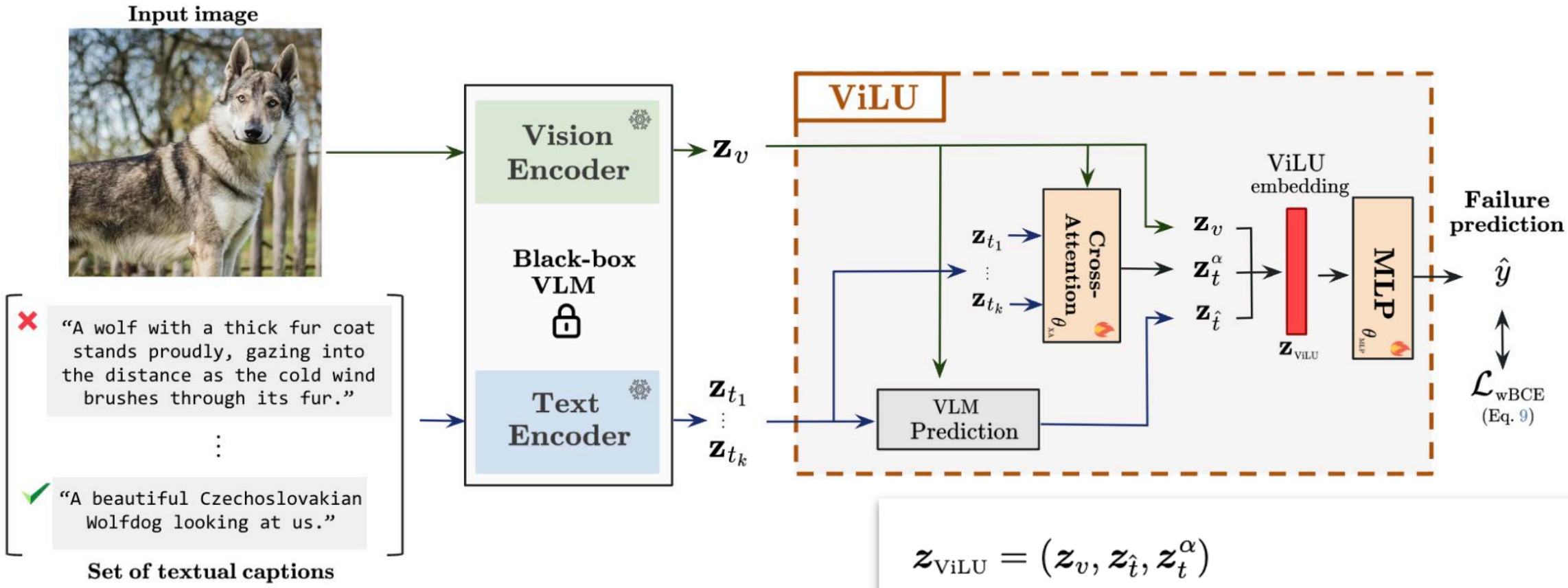
- Higher class confusion
- High uncertainty

ViLU: Learning Vision-Language Uncertainty



- Inputs: visual representation z_v + K concept embeddings $Z_t = \{z_{t_j}\}_{1<j<K}$
- ViLU embedding: $z_{\text{ViLU}} = (z_v, z_{\hat{t}}, z_t^\alpha), z_{\hat{t}}$ pred. Class embedding
  - Image-text cross-attention module => z_t^α
  - Query z_v , keys/values Z_t

ViLU: Learning Vision-Language Uncertainty



- **Failure prediction from ViLU embedding:** binary classification
- Consistent generalization of MCM

$$\boldsymbol{z}_{\text{ViLU}} = (\boldsymbol{z}_v, \boldsymbol{z}_{\hat{t}}, \boldsymbol{z}_t^\alpha)$$

$$g_{\theta_{\text{MLP}}}(\boldsymbol{z}_{\text{ViLU}}) \approx \frac{1}{2}\boldsymbol{z}_{\text{ViLU}}^T \, A \, \boldsymbol{z}_{\text{ViLU}} = z_v^T \, z_{\hat{t}}$$

$$\text{with} \quad A = \begin{pmatrix} \mathbf{0} & \mathrm{I}_d & \mathbf{0} \\ \mathrm{I}_d & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

Conformal prediction & theoretical guarantees

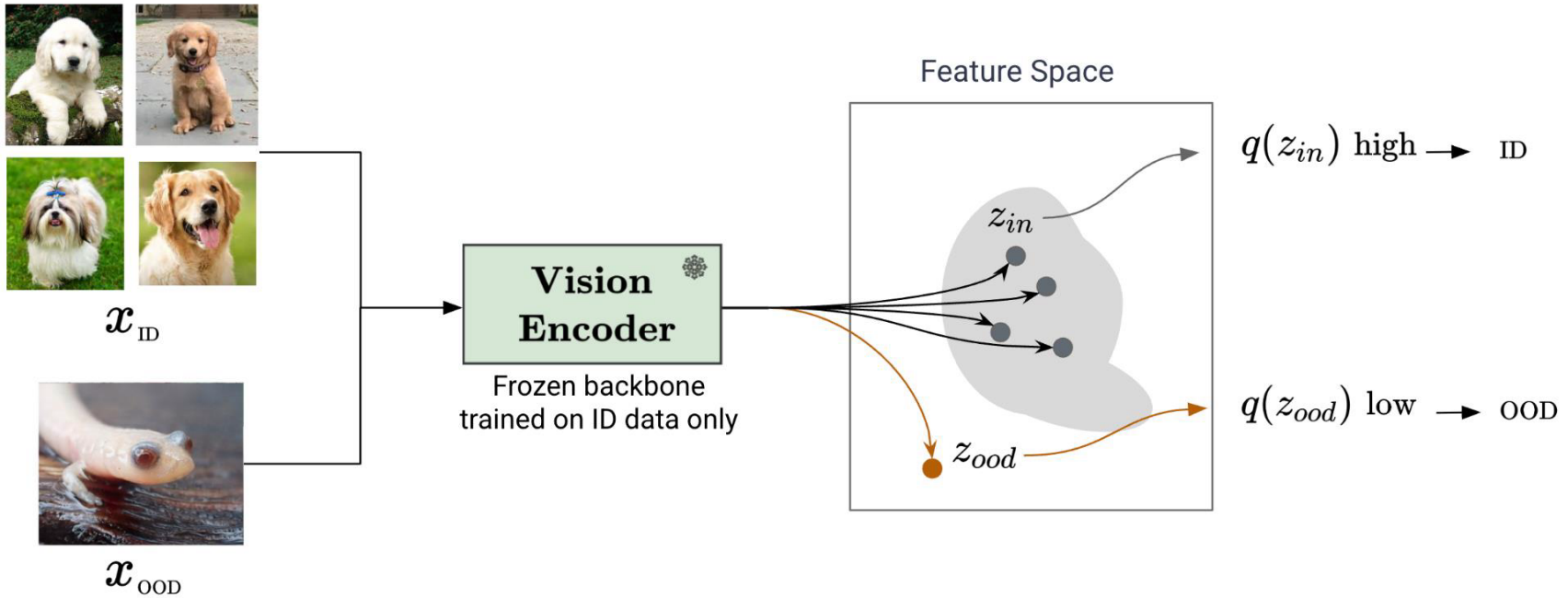
- Matteo’s talk!

Uncertainty Quantification in deep learning

1. Calibration
2. Failure Prediction
3. Out-of-distribution / anomaly detection

OOD/ anomaly detection

- Detecting OODs/anomalies <=> epistemic uncertainty
  - Binary classification problem: in-distribution (ID) vs OOD
- 2 main classes of approaches for OOD
  1. Bayesian methods: estimating the predictive distribution p(y| x*, D)
  2. Estimating ID density p(x)

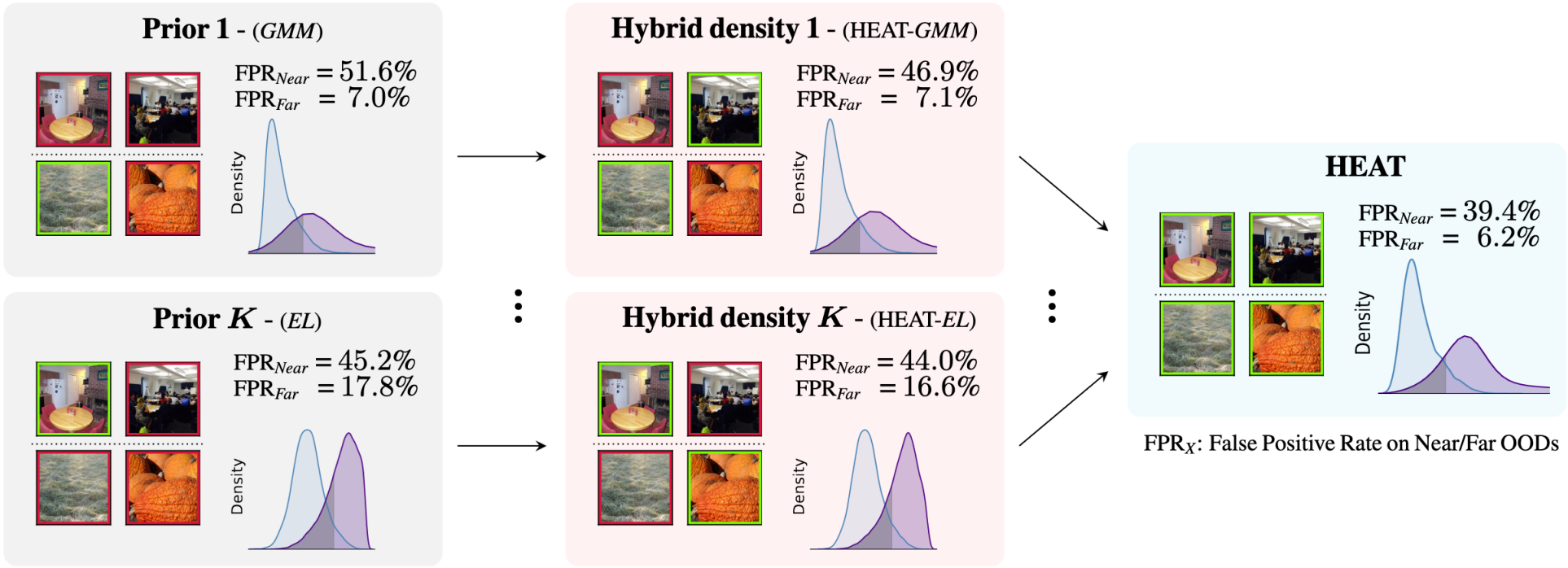


- Density in the feature space of a pre-trained model
- State-of-the-art: GMM, KNN, EBM, diffusion models, etc

Sehwag, V., et al. "SSD: A unified framework for self-supervised outlier detection." ICML 2021.
Sun, Yiyou, et al. "Out-of-distribution detection with deep nearest neighbors." ICML, 2022.
Lafon. et al. "Hybrid Energy Based Model in the Feature Space for Out-of-Distribution Detection." ICML 2023.
A, Heng et.al. "Out-of-Distribution Detection with a Single Unconditional Diffusion Model". NeurIPS 2024

Out-Of-Distribution (OOD) detection

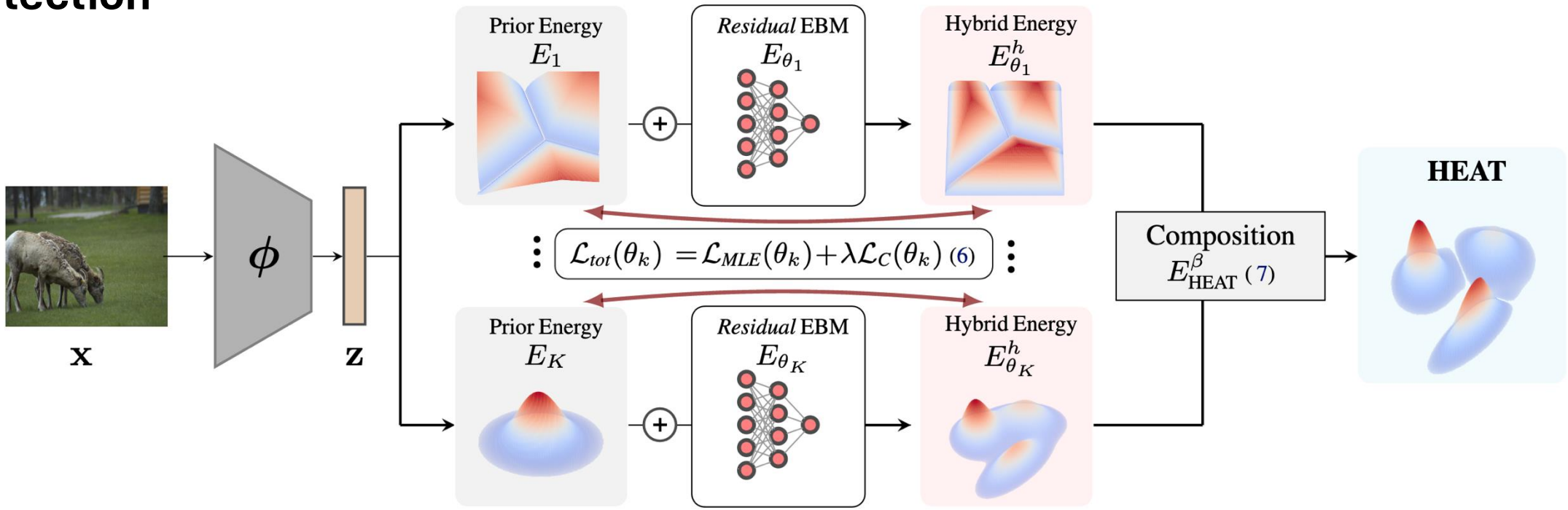
- State-of-the-art ID density estimation: prior densities, *e.g.*, GMM, Energy Logits (EL)



- Prior density: not accurate **=> Energy correction**
- GMM good for far-OOD, EL for near-OOD **=>Energy composition**

OOD detection

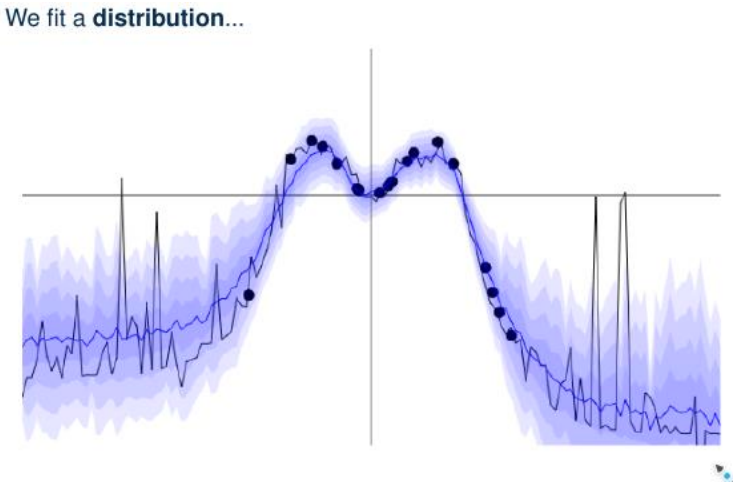
- **HEAT** [LRR+23]: **Hybrid Energy Based Model (EBM) in the feature space for OOD detection**



- **Energy-based correction** of prior energy terms, *e.g.* Gaussians
- **Energy composition** of several terms (Gaussian, Energy Logits, std for style)

[LRR+23] M. Lafon, E. Ramzi, C. Rambour, N. Thome. Addressing Failure Detection by Learning Model Confidence. ICML 2023.

Bayesian methods



- Observed inputs $X = \{x_i\}_{i=1}^N$ and outputs $Y = \{y_i\}_{i=1}^N$
  - $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}^K$ (classification or regression)
  - Model with parameters w: $\hat{y}_i = f_w(x_i)$
- **Bayes rule:** $p(Y, w/X) = p(Y/X, w)p(w) = p(w/X, Y)p(Y/X)$

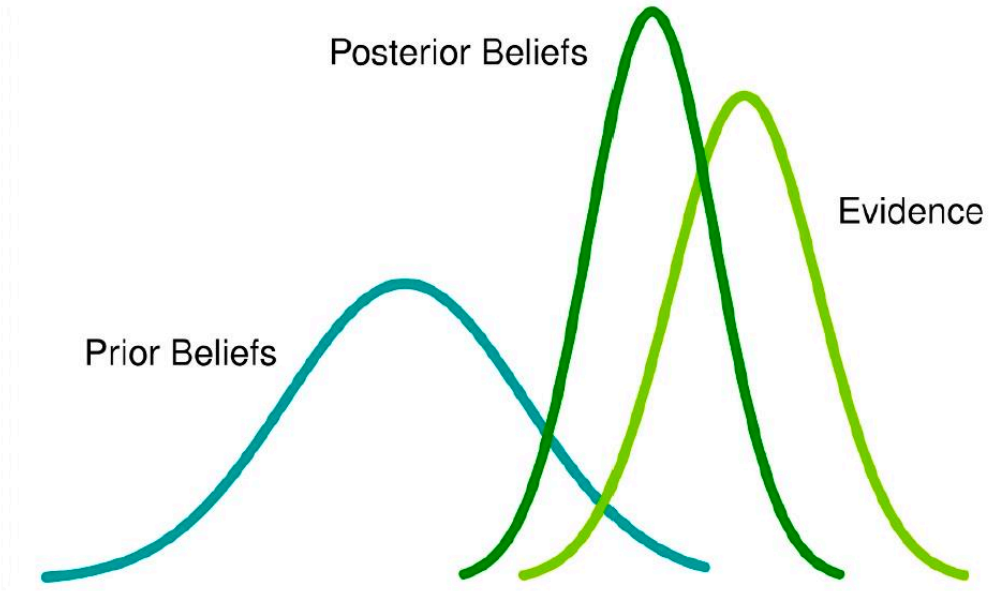
$$\Rightarrow \boxed{p(w/X, Y) = \frac{p(Y/X,w)p(w)}{p(Y/X)} \propto p(Y/X, w)p(w)}$$

- From posterior $p(w/X, Y) \Rightarrow$ compute **predictive distribution** given new input x^* $(\forall y)$: $p(y/x^*, Y, X) = \int p(y, w/x^*, Y, X)dw$

$$\boxed{p(y/x^*, Y, X) = \int p(y/x^*, w)p(w/X, Y)dw} = \mathbb{E}_{p(w|\mathcal{D})}[p(y|x^*, w)]$$

Bayesian methods

- RECAP: for uncertainty estimate with Bayesian models
  1. Define prior $p(w)$ and likelihood $p(Y/X, w)$
  2. Compute posterior distribution $p(w/X, Y)$
  3. Compute predictive distribution $p(y^*/x^*, Y, X)$
- Easy? NO !! $\Rightarrow$ steps 2 and 3 computationally hard in general!
  - Typically no closed form for step 2
  - High-dimensional integration for step 3



Bayesian Linear Regression

With $p(\boldsymbol{w}|\alpha) = \mathcal{N}(\boldsymbol{w}; 0, \alpha^{-1}\boldsymbol{I})$ and $p(y_i/x_i, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{\Phi}_i^T \boldsymbol{w}, \beta^{-1})$, we can show that:

$$p(\boldsymbol{w}/X, Y) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$
$$\boldsymbol{\Sigma}^{-1} = \alpha\boldsymbol{I} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi}$$
$$\boldsymbol{\mu} = \beta\boldsymbol{\Sigma}\boldsymbol{\Phi}^T Y$$

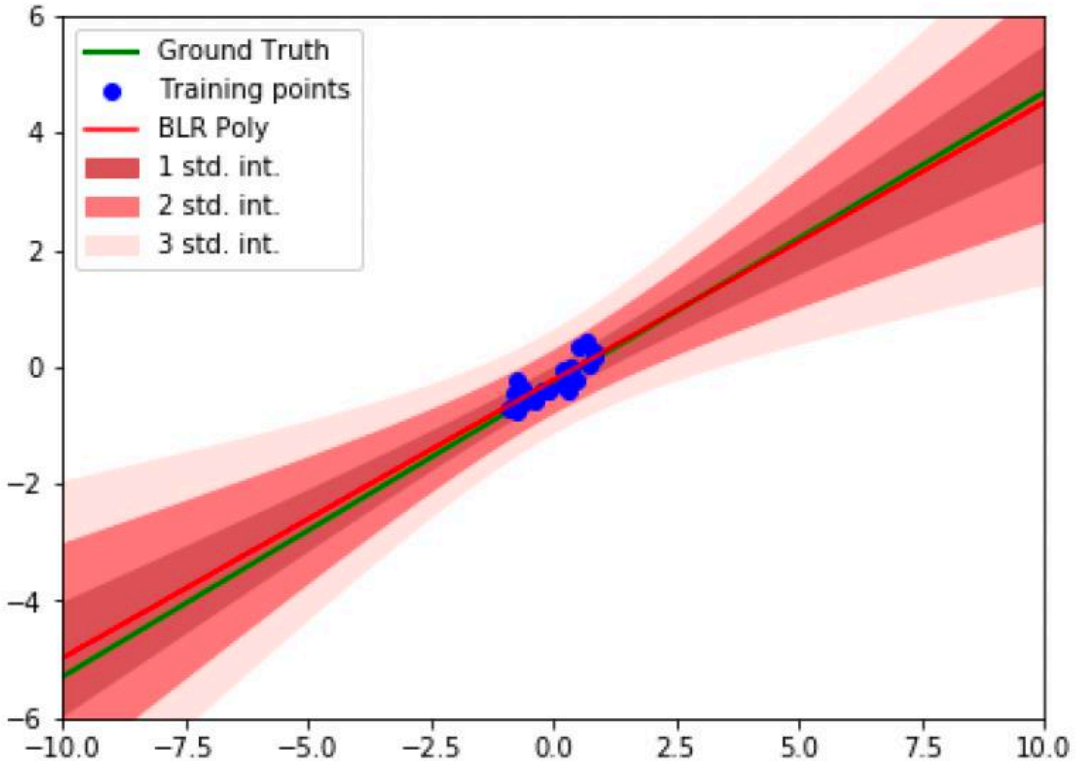
- $p(y|x^*, \mathcal{D}, \alpha, \beta) = \int p(y|x^*, w, \beta)p(w|\mathcal{D}, \alpha, \beta)dw$

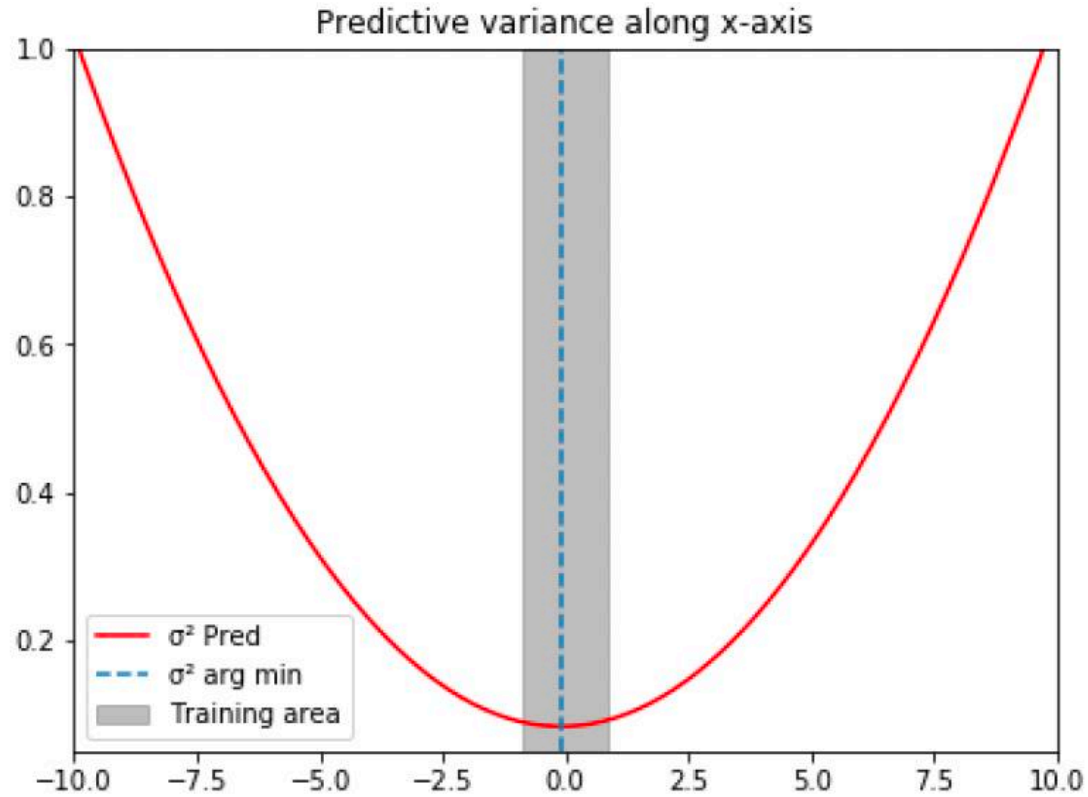
Convolution between 2 Gaussians => Gaussian

  - Mean of predictive distribution $\mu^T\boldsymbol{\Phi}(x^*)$
  - Variance of predictive distribution $\sigma^2_{pred}(x^*) = \frac{1}{\beta} + \boldsymbol{\Phi}(x^*)^T\boldsymbol{\Sigma}\boldsymbol{\Phi}(x^*)$

$$p(y|x^*, \mathcal{D}, \alpha, \beta) = \mathcal{N}(y; \mu^T\boldsymbol{\Phi}(x^*), \frac{1}{\beta} + \boldsymbol{\Phi}(x^*)^T\boldsymbol{\Sigma}\boldsymbol{\Phi}(x^*))$$

Bayesian Linear Regression





Beyond Bayesian Linear Regression

No analytical expression for posterior $p(\boldsymbol{w}|\mathcal{D})$ and $p(y|\boldsymbol{x}^*, \mathcal{D})$ in general

$\Rightarrow$ Approximation nedded!

$p(\boldsymbol{w}|\mathcal{D})$: approximating with a Gaussian distribution $q_\theta(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\theta)$

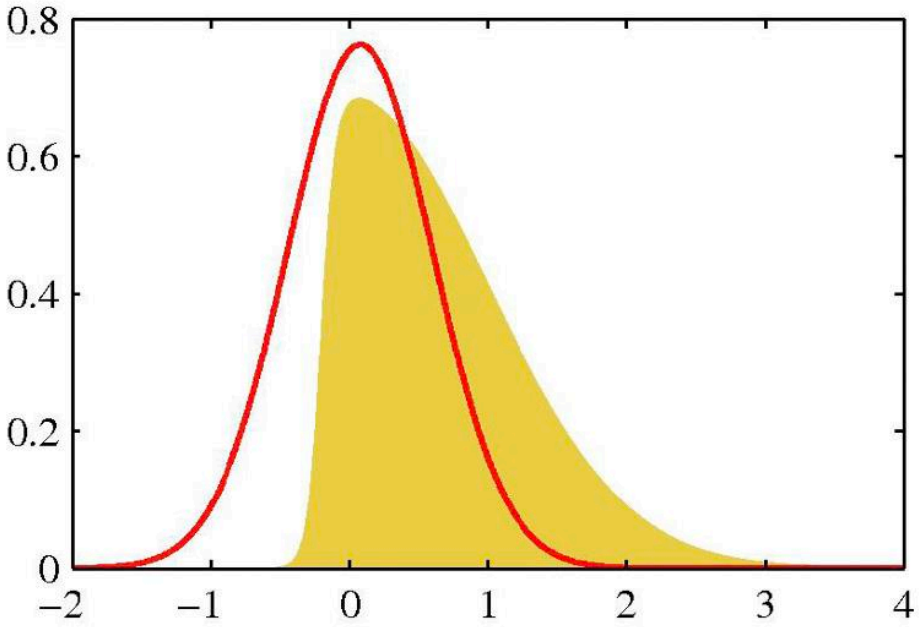
- θ: parameters of the Gaussian distribution, i.e. mean μ, covariance Σ

Approximating posterior with $q_\theta(w)$

- Laplace approximation
- Fit $q_\theta(w)$ on the mode of p(w|D)

$$\boldsymbol{\mu} = \boldsymbol{w}_{MAP} \quad \nabla_{\boldsymbol{w}} p(\boldsymbol{w}) = 0$$

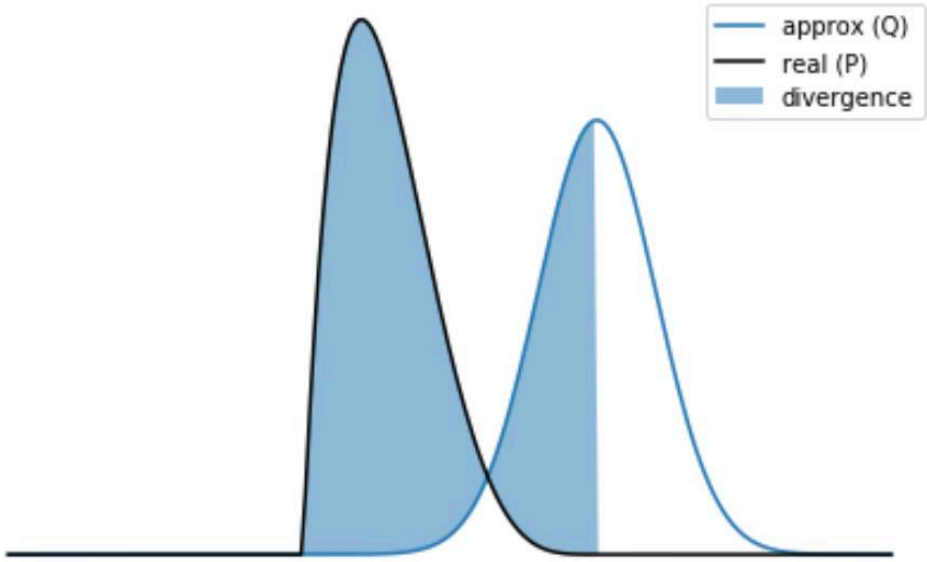
$$\boldsymbol{\Sigma}^{-1} = \nabla\nabla_{\boldsymbol{w}} p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y})\big|_{\boldsymbol{w}=\boldsymbol{w}_{MAP}}$$



- Used in Bayesian Logistic regression

- Variational approximation:
  Minimize $KL(q_\theta(\boldsymbol{w})||p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{Y}))$

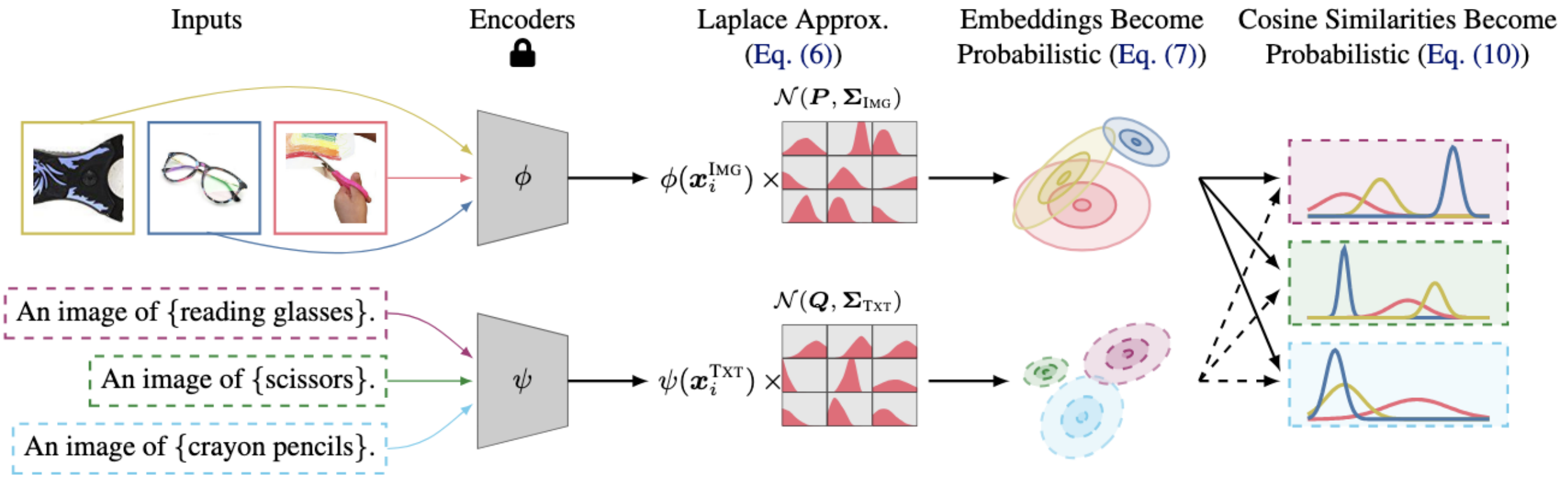
- More global fitting



- Used in Bayesian Neural Networks

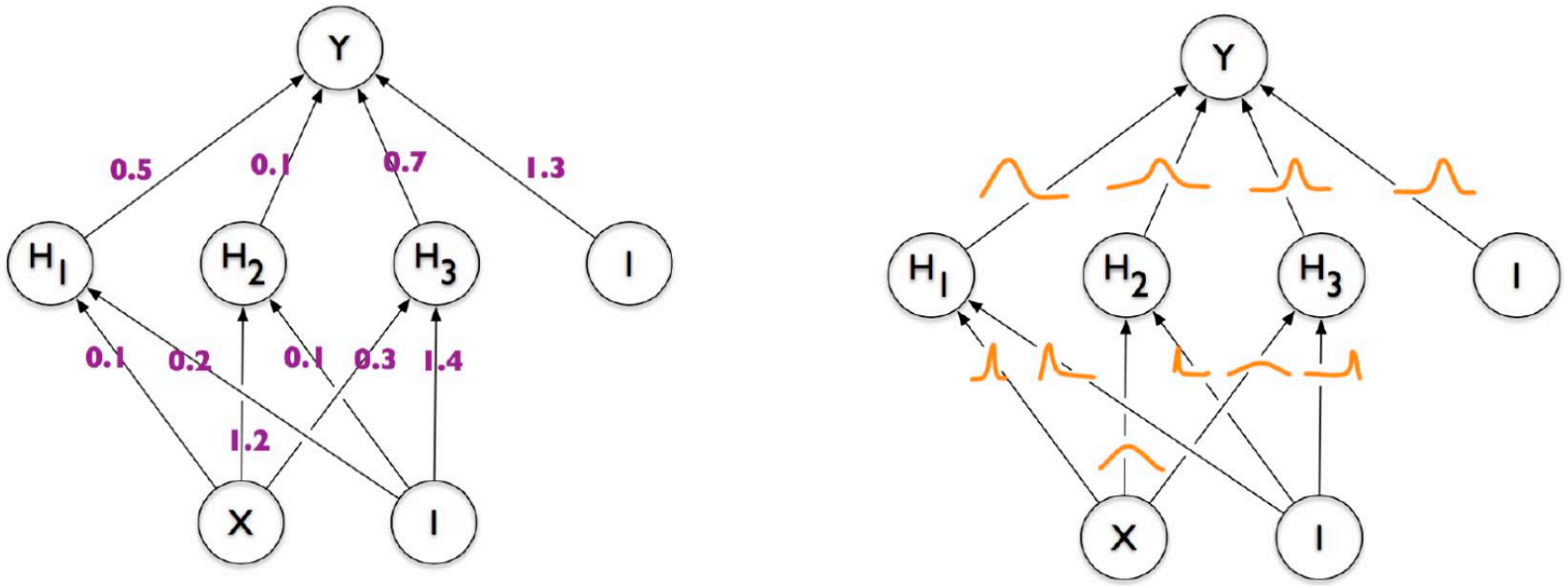
Laplace for Vision Language Models (VLMs)

- BayesVLM [A]: post-hoc Laplace approximation in VLMs, pre-trained e.g., LAION
- Estimate the distribution of similarities between text/image pairs
  - Estimate the distribution of the **last layer** or text/image encoders independently
  - Kronecker-factored (KFAC) Generalized Gauss–Newton (GGN) Hessian approx



[A] Post-hoc Probabilistic Vision-Language Models. ICLR 2026 submission

VI in Bayesian Neural Networks (BNNs)



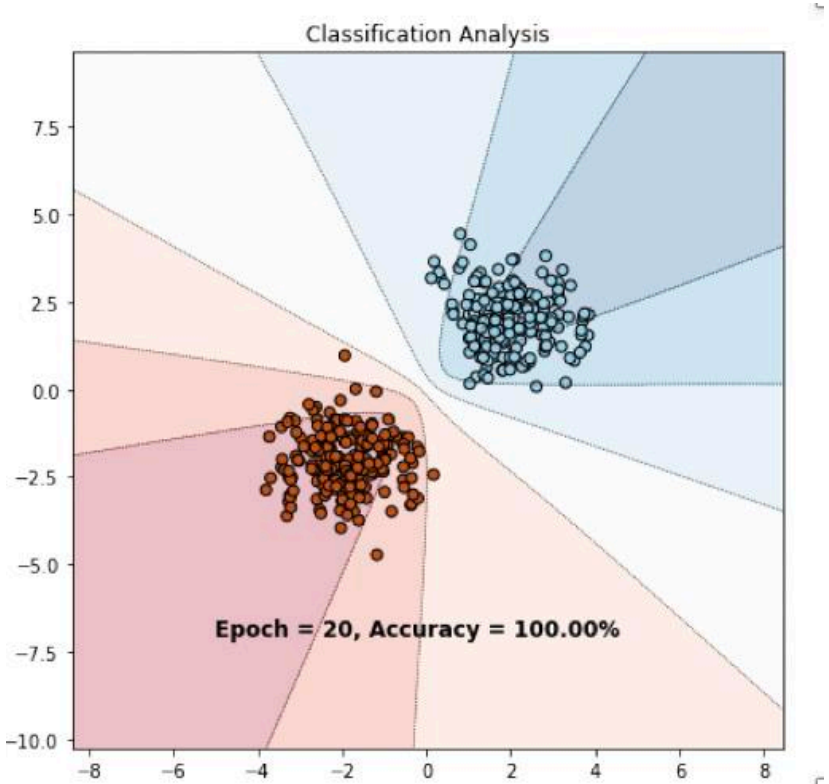
Credit: [Blundell et al., 2015]

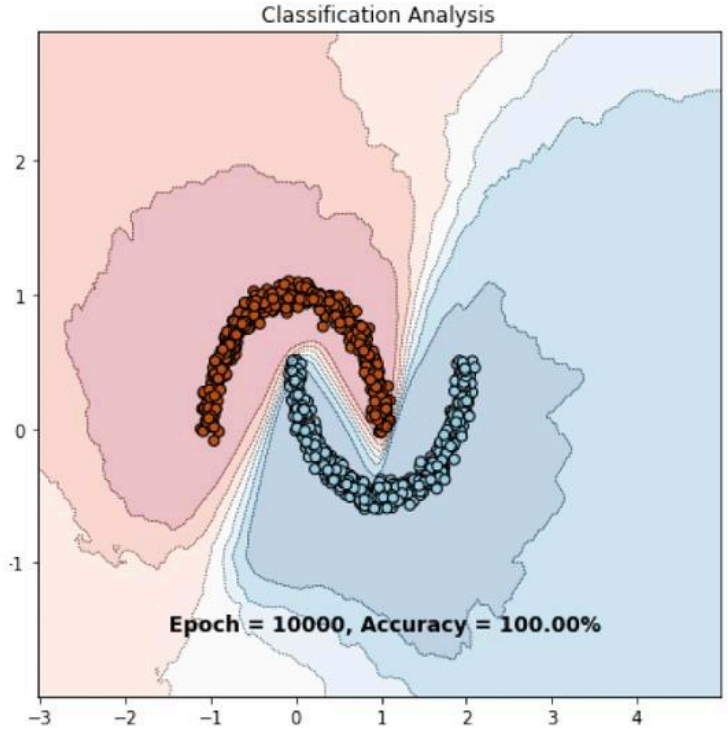
- Generally, independence between weights (mean field)+ simple 1D Gaussians
  - Define prior over weights, *e.g.* $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0, \alpha^{-1}\mathcal{I})$
  - Define likelihood, $p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w})$, *e.g.* for regression $p(\boldsymbol{y}_i|\boldsymbol{x}_i, \boldsymbol{w}) = \mathcal{N}(\boldsymbol{y}_i; f^{w}(\boldsymbol{x}_i), \beta^{-1})$
    - Each weight of the network w_j has its own mean μ_j and variance σ_j
      - $\boldsymbol{\theta} = \{(\mu_j, \sigma_j)\}_{j\in\{1;D\}}$: **variational parameters**

Approximating Predictive Distribution

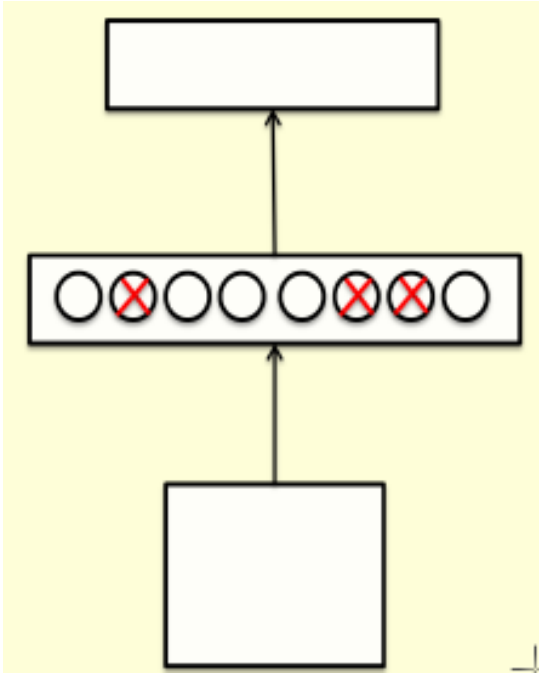
$p(\boldsymbol{y}|\boldsymbol{x}^*, \mathcal{D}) \approx \int p(\boldsymbol{y}|\boldsymbol{x}^*, \boldsymbol{w}) q_\theta(w) d\boldsymbol{w}$ => approximate with Monte Carlo (MC) sampling

$$\int p(y|\mathrm{x}^*, \mathrm{w}) q_{\boldsymbol{\theta}}(\mathrm{w}) dw \approx \sum_{s=1}^{S} p(y|\mathrm{x}^*, w^s) \quad \mathrm{w}^s \sim q_{\boldsymbol{\theta}}(\mathrm{w})$$ Easy to sample from $q_\theta(w)$





Monte Carlo Dropout

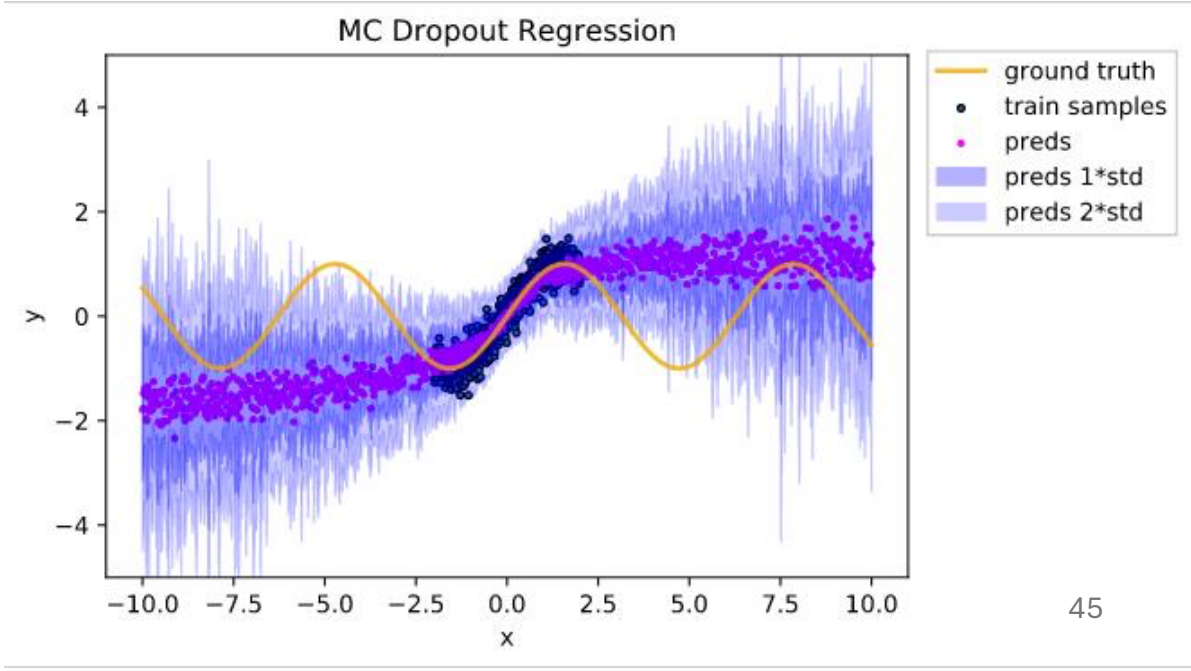


- Dropout as a variational inference [Gal, 2016]
- Big results: training with dropout is a special VI case
  - With some specific approximate posterior distribution

$$q(\boldsymbol{W}_l) = \text{diag}(\hat{\boldsymbol{\varepsilon}}_l)\boldsymbol{M}_l,\ \varepsilon_{l,i} \sim Bernoulli(1 - p_i)$$

  - $\mathbf{M}_l$ deterministic weights, $q(\mathbf{W}_l)$ approximate posterior

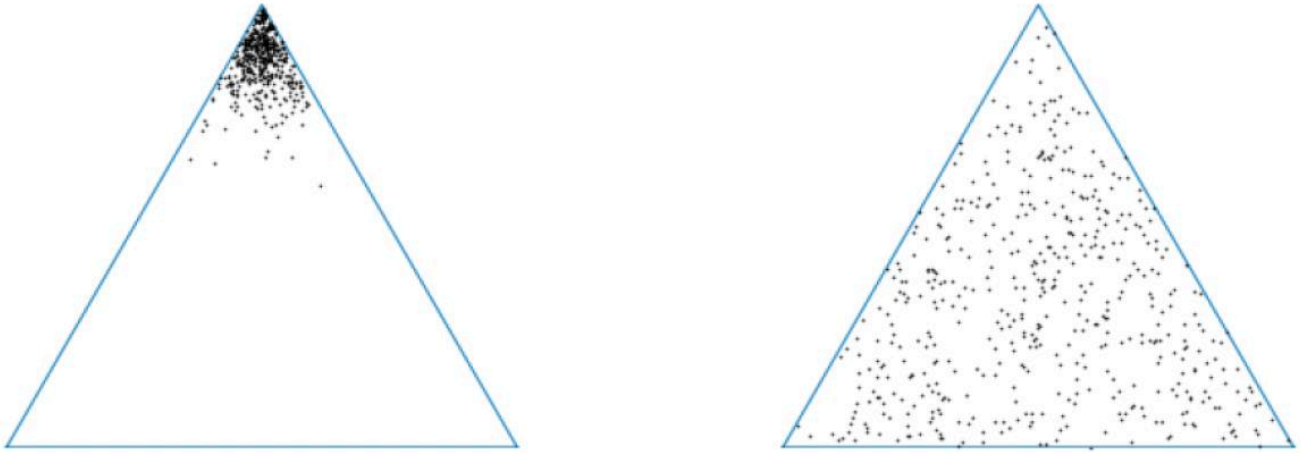
- => MC sampling of model trained with dropout: predictive distribution
  - Mean ~ prediction
  - Std ~ uncertainty



Ensembling

$$\int p(y|\mathrm{x}^*, \mathrm{w}) q_\theta(\mathrm{w}) dw \approx \sum_{s=1}^{S} p(y|\mathrm{x}^*, w^s) \quad \mathrm{w}^s \sim q_\theta(\mathrm{w})$$

- MC sampling ⇔ ensembling different models
- Simple but effective baseline: train several models on the same dataset
  - Use disagreement, e.g. Mutual Information, as epistemic uncertainty



- **<u>Main drawback:</u>** High training & inference cost

Lakshminarayanan, B., Pritzel, A., and Blundell, C.. Simple and scalable predictive uncertainty estimation using deep ensembles. NeurIPS 2017.

Future directions in Uncertainty Quantification

- A unified uncertainty score (epistemic/aleatoric/calibration) able to capture different facets of uncertainties

- UQ for sequential decision tasks, forecasting or NLP?
  - Which uncertainty for LLMs / Generative VLMs?

- Dense/structured prediction tasks, correlations – segmentation
  - How to design UQ score reflecting these correlations?

- Statistical guarantees of UQ?
  - Conformal, but needs conditional guarantees

Thank you for your attention

- Questions?