Uncertainty in
Dialogue Belief Tracking

Carel van Niekerk

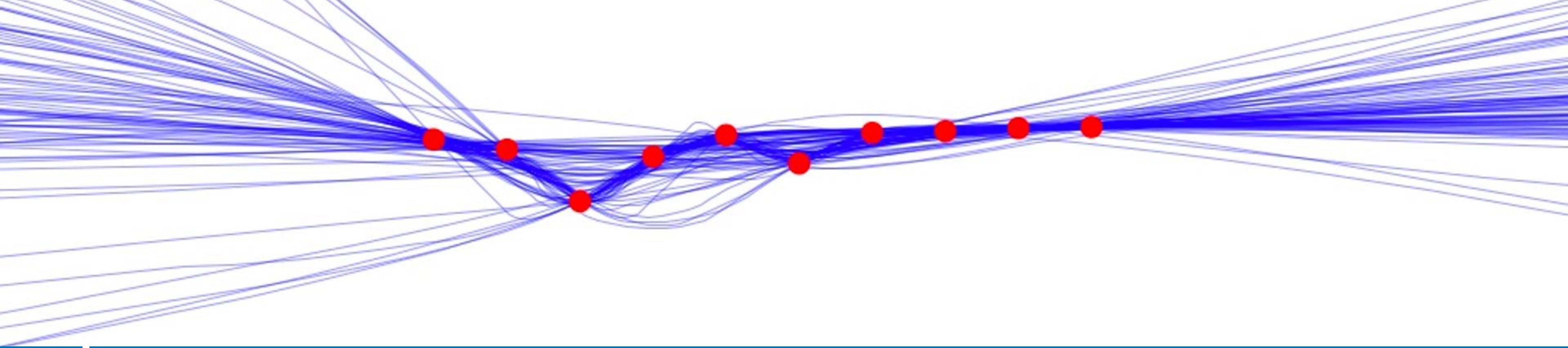
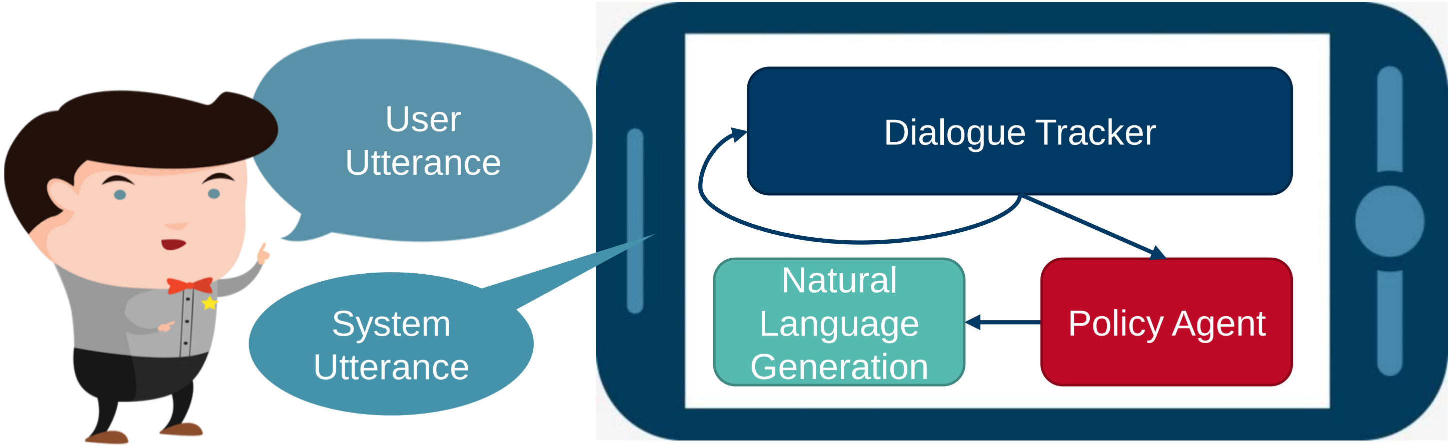


Table of Contents

Heinrich Heine
Universität Düsseldorf

Dialogue Belief Tracking

- Dialogue tracking is the task of tracking the user goal in a dialogue.



Dialogue Belief Tracking

Hey. I need a restaurant near the city centre.

hello(type=restaurant) 0.6
inform(type=restaurant, location=centre) 0.4

Where would you like the restaurant?

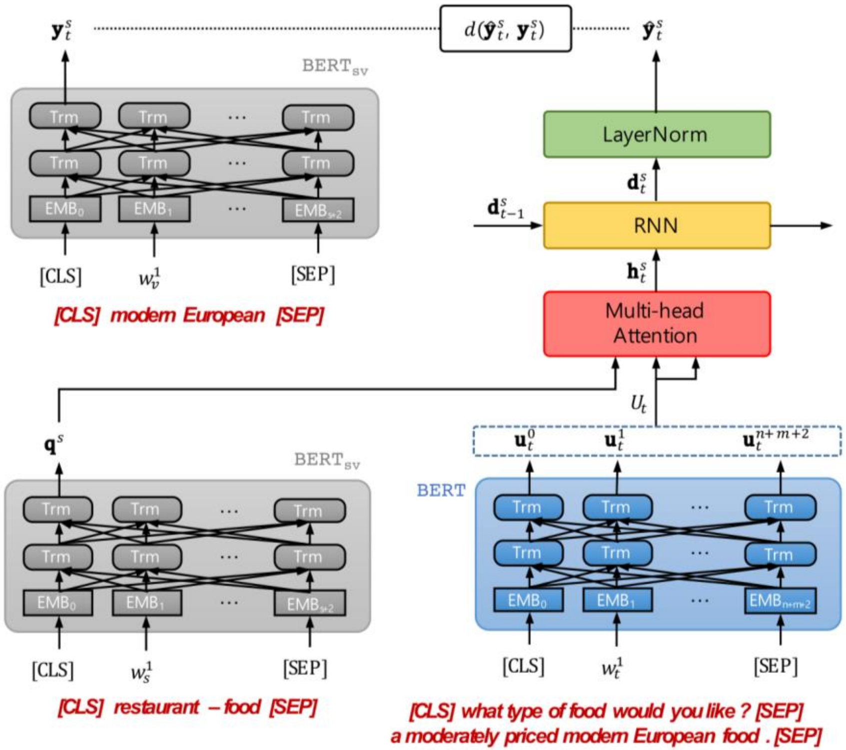
The City Centre!

inform(location=city) 0.6
inform(location=centre) 0.4

To confirm you want a restaurant near the city centre?

Dialogue Belief Tracking - SUMBT

Heinrich Heine
Universität Düsseldorf



What is Uncertainty?

Heinrich Heine Universität Düsseldorf

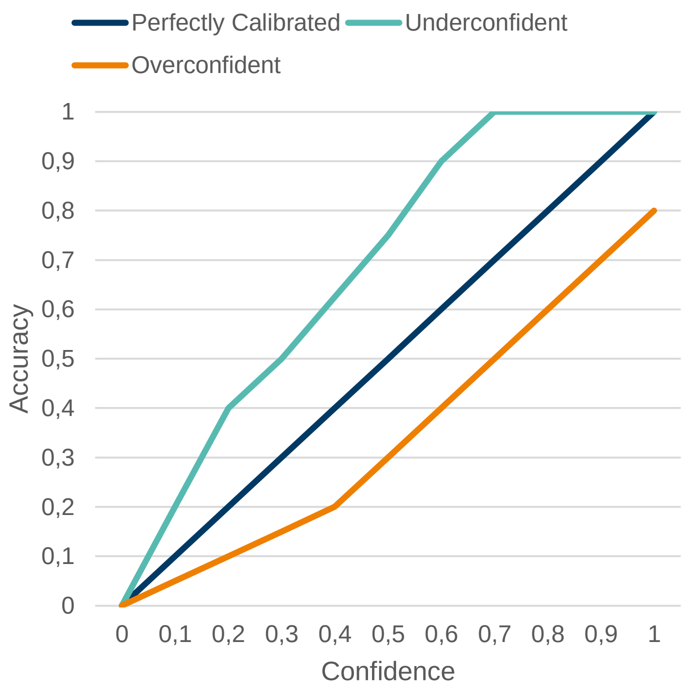
- The ability to say: **“I don‘t know!“**
- A prediction should have **high uncertainty** if the model **cannot accurately** make an prediction for the current observation.

The Problem of Overconfidence

- **Overconfidence** is when a model is **always extremely certain** about its predictions, even when these predictions are incorrect.
- Problems:
  - Users **cannot rely** on the model as it makes incomprehensible mistakes.
  - Predictions by the model is **hard to understand**.

Calibration

- A **well-calibrated** model is one where the **confidence** and **accuracy** of the model is **aligned**.

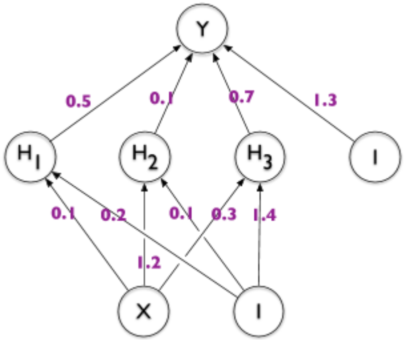


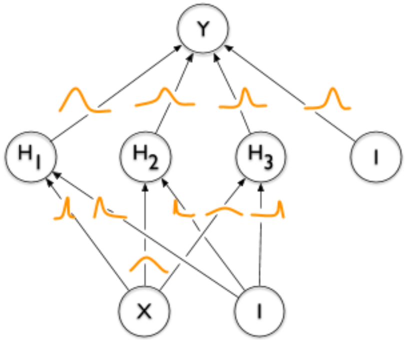
Solutions

- Bayesian Neural Networks
- Loss Functions
- Ensembles
- Post Processing

Bayesian Neural Networks

- Large number of **extra parameters**
- Very difficult to select **suitable priors** for the parameters during training.
- Learns the **posteriors** of the parameters and the **model likelihoods jointly**.

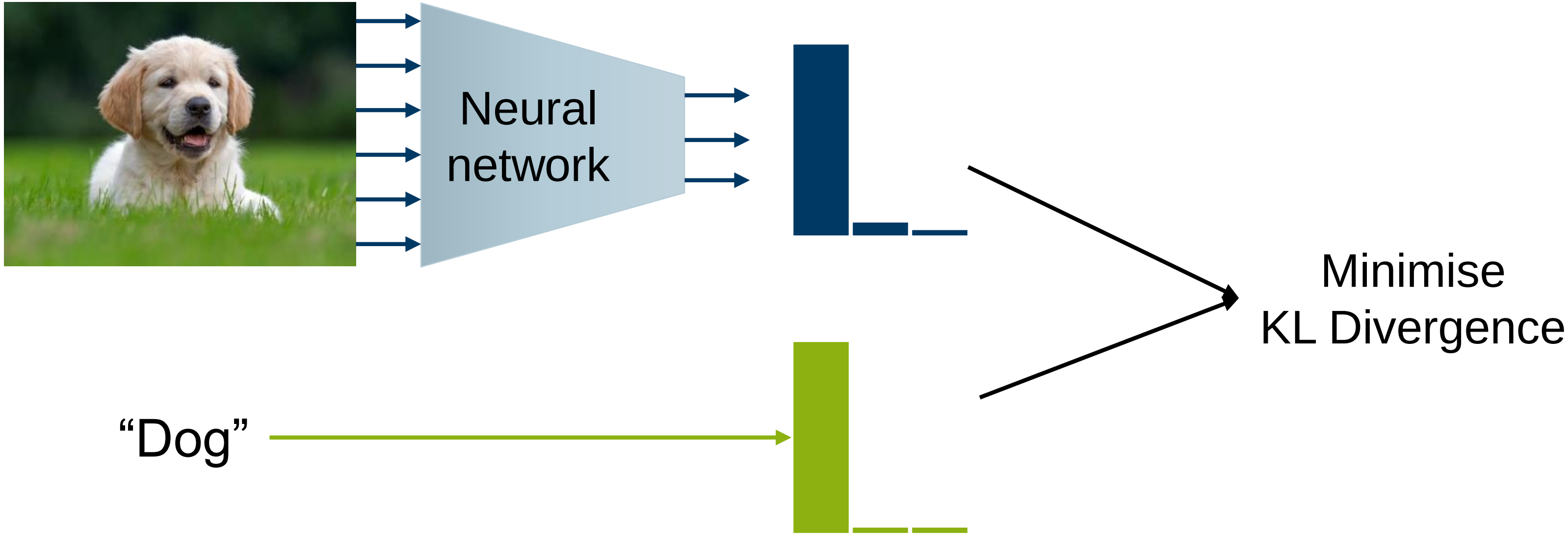




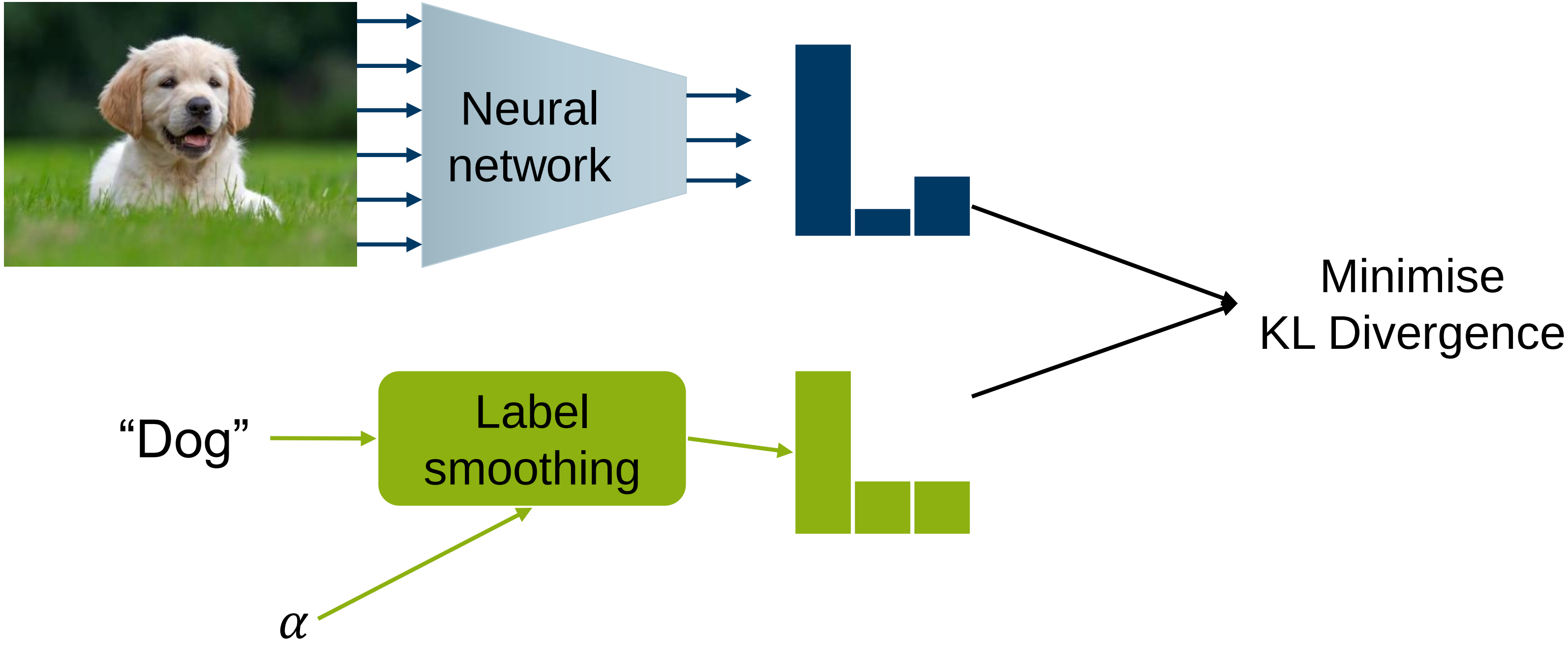
Loss Functions

- SoftMax Cross Entropy
- Label smoothing
- Bayesian matching

SoftMax Cross Entropy



Label Smoothing



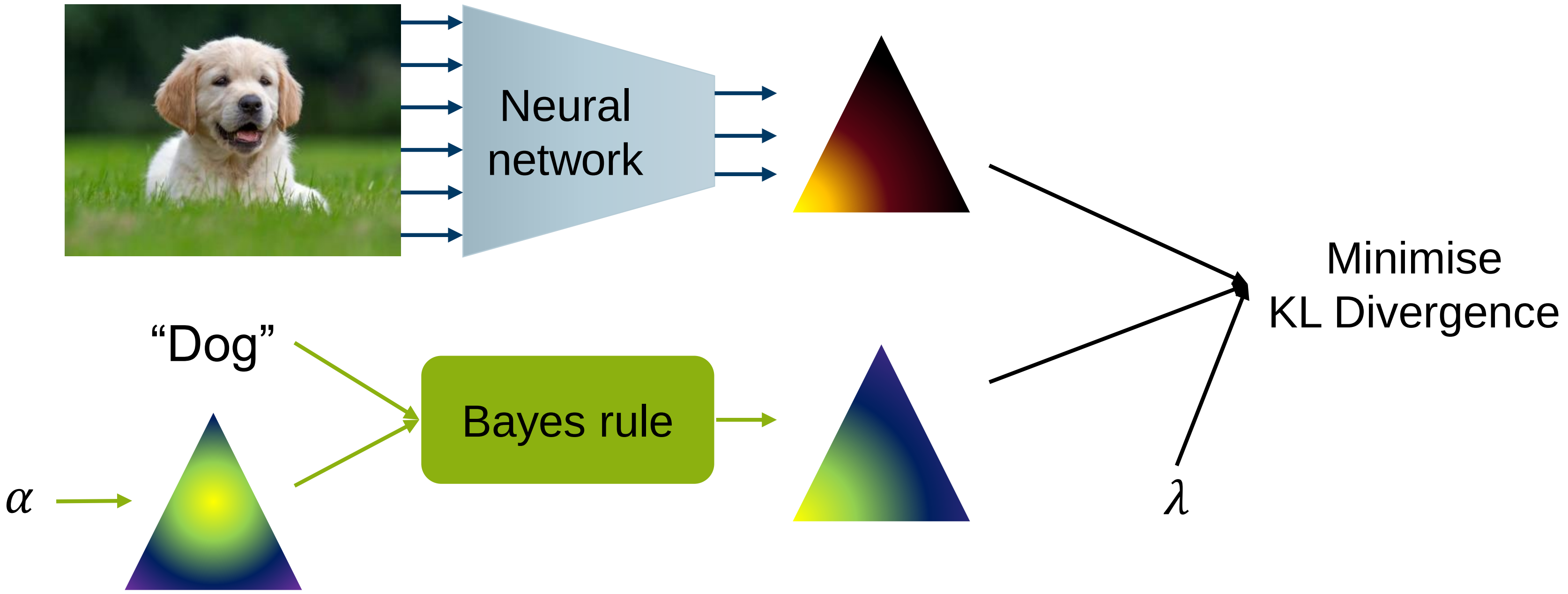
Label Smoothing

Heinrich Heine
Universität Düsseldorf

$$q(y;\alpha) = \begin{cases} 1-(K-1)\alpha & \text{if } y = \text{ true label} \\ \alpha & \text{otherwise} \end{cases}$$

K – Number of possible classes
α – Smoothing parameter

Bayesian Matching



Bayesian Matching

- Likelihood (True Label)

$$y \sim \text{Categorical}(\boldsymbol{z})$$

- Prior (Used to learn uncertainty)

$$\boldsymbol{z} \sim \text{Dirichlet}(\alpha \mathbf{1})$$

- Posterior (Target Distribution)

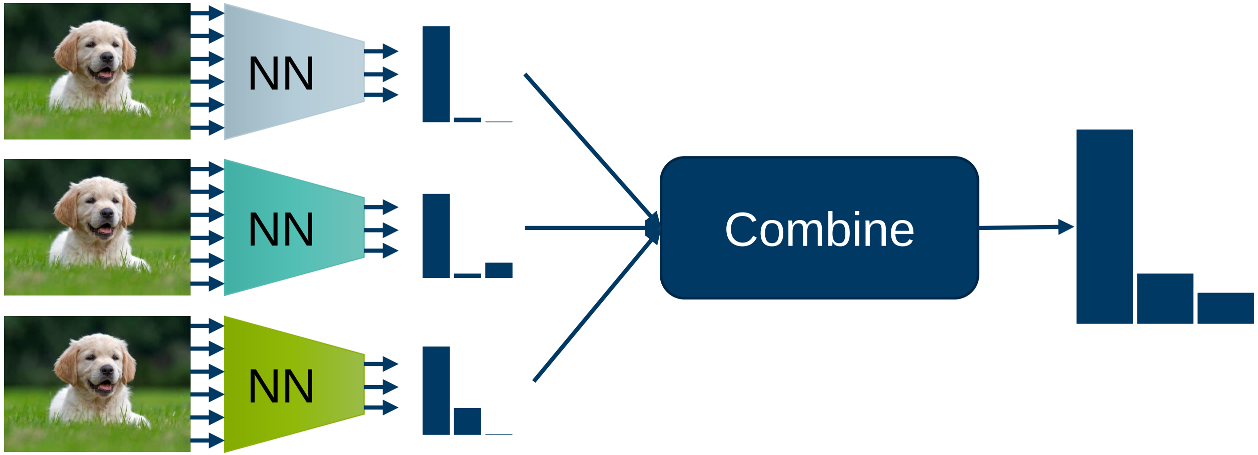
$$\boldsymbol{z} | y \sim \text{Dirichlet}\left(\alpha \mathbf{1} + \frac{\boldsymbol{y}}{\lambda}\right)$$

Loss Functions

Loss function	Joint goal accuracy	Top 3 joint goal accuracy	Expected joint goal calibration error
Cross entropy	**46.7%**	69.9%	1.996
Label smoothing	46.3%	**74.6%**	**1.292**
Bayesian matching	31.0%	45.1%	4.922

- Label smoothing produces better calibration
- Bayesian matching results in under-confidence

Ensemble



Ensemble

$$\mathbb{P}(y|\boldsymbol{x},\mathcal{D}) = \int \mathbb{P}(y|\boldsymbol{x},\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$$

Likelihood given the model (Model predictions)

Posterior of the model (Intractable)

Ensemble

$$\mathbb{P}(y|\boldsymbol{x},\mathcal{D}) = \int \mathbb{P}(y|\boldsymbol{x},\boldsymbol{\theta})\mathbb{P}(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$$

$$\approx \int \mathbb{P}(y|\boldsymbol{x},\boldsymbol{\theta})q(\boldsymbol{\theta})d\boldsymbol{\theta}$$

Approximate the posterior using an ensemble

$$\approx \sum_{i=1}^{N} \mathbb{P}(y|\boldsymbol{x},\boldsymbol{\theta}^{(i)})$$

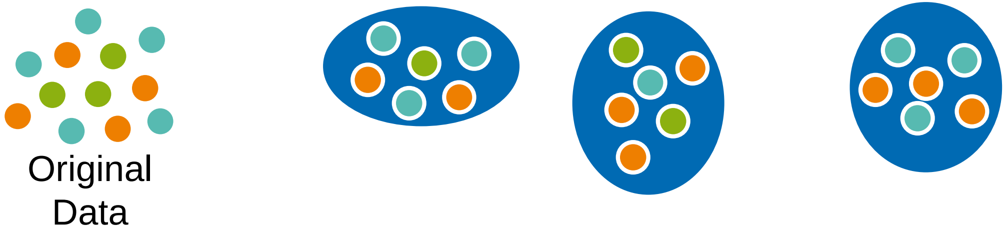
Monte-Carlo Integration

$$\boldsymbol{\theta}^{(i)} \sim q(\boldsymbol{\theta})$$

Ensembles

- Dropout:
  - Collection of models with different **nodes randomly eliminated**.
  - **Single model** trained on all the training data

- Bootstrap:
  - Collection of **training sets resampled** from the original training set
  - Collection of **independent models** trained on the subsets
  - Sampling is done using “**with replacement**”.

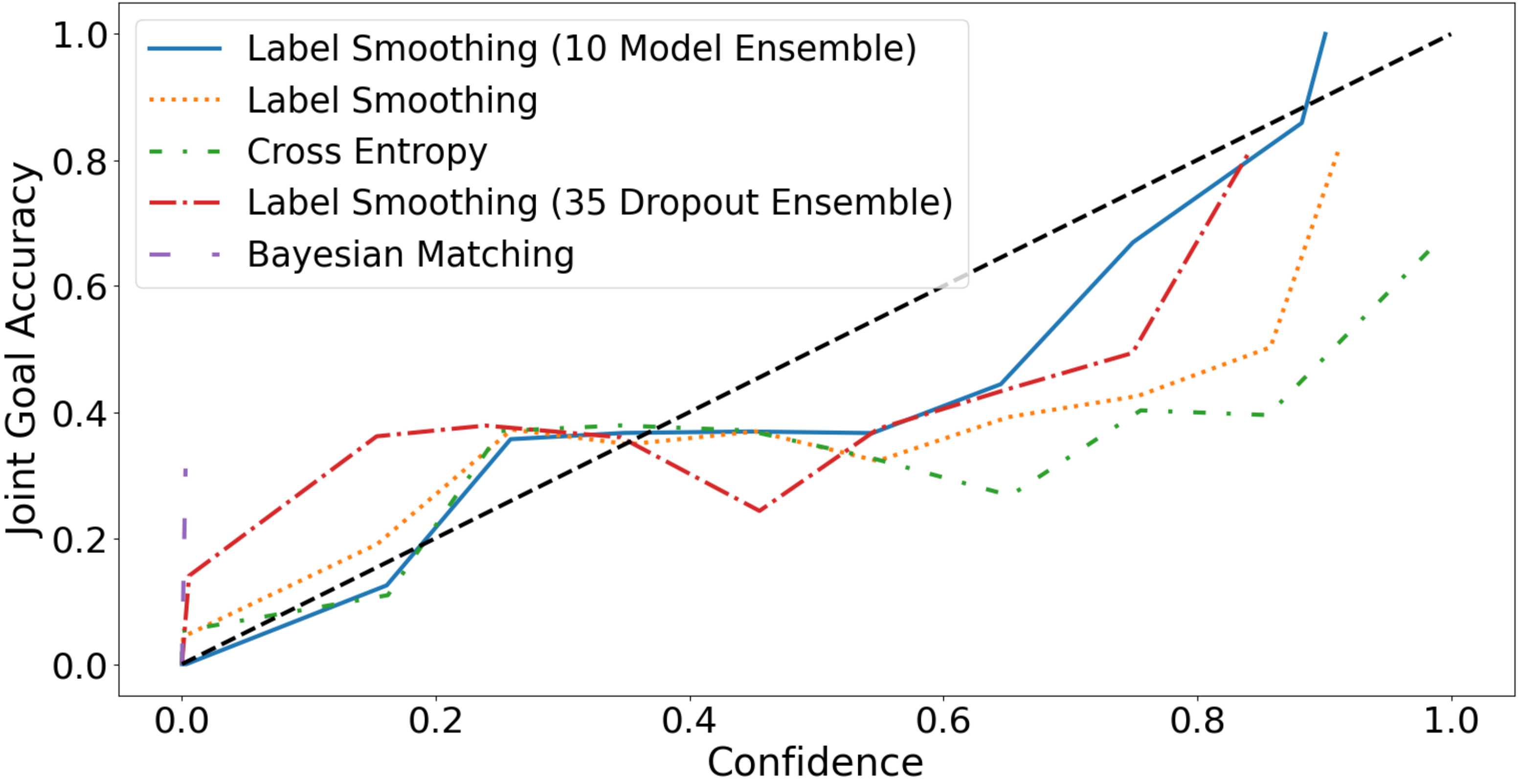


Ensembles

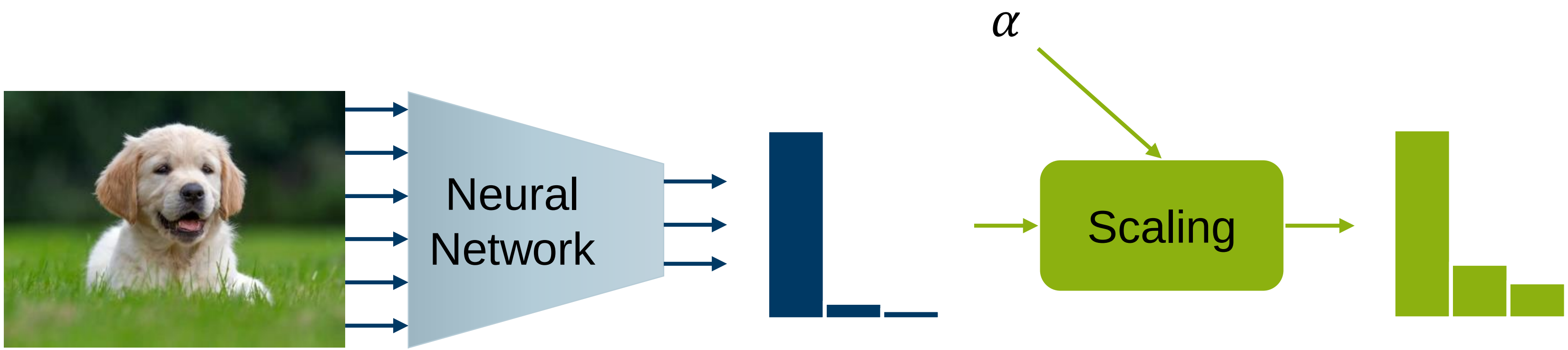
Heinrich Heine
Universität Düsseldorf

Strategy	Joint goal accuracy	Top 3 joint goal accuracy	Expected joint goal calibration error
Baseline	46.3%	74.6%	1.292
Dropout Ensemble	46.6%	76.1%	2.217
Bootstrap Ensemble	**48.4%**	**84.1%**	**0.841**

Reliability Diagram



Post Processing - Temperature Scaling



Post Processing - Temperature Scaling

$$q(y) = \varphi\left(\frac{\mathbf{z}}{\alpha}\right)$$

- **z** – Model output logits
- α – Scaling coefficient
- φ – Activation function

Conclusion

- Using an **appropriate loss** function can improve model calibration.
- **Ensembles** of models provides significant improvement in calibration.
- Post processing is **not very effective** as it applies the **same correction** to every observation.
- It is possible to teach the model to:

  “Know when it does not know.”

Questions

Resources

- Calibration of Pre-trained Transformers
- Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning
- On Calibration of Modern Neural Networks
- Being Bayesian about Categorical Probability
- SUMBT: Slot-Utterance Matching for Universal and Scalable Belief Tracking
- Predictive Uncertainty Estimation via Prior Networks
- Uncertainty in Structured Prediction