Uncertainty estimation via Prior Networks: A Tutorial

Andrey Malinin
e-mail - am969@yandex-team.ru
twitter - @AndreyMalinin

22nd January 2021

Joint work with...

(a) Mark Gales

(b) Bruno Mlodozeniec

(c) Ivan Provilkov

(d) Sergey Chervontsev

Overview of the Talk

Overview of the Talk

1. **Motivation and Sources**
2. Uncertainty Estimation via Ensembles
3. Uncertainty Estimation via Prior Networks
4. Ensemble Distribution Distillation

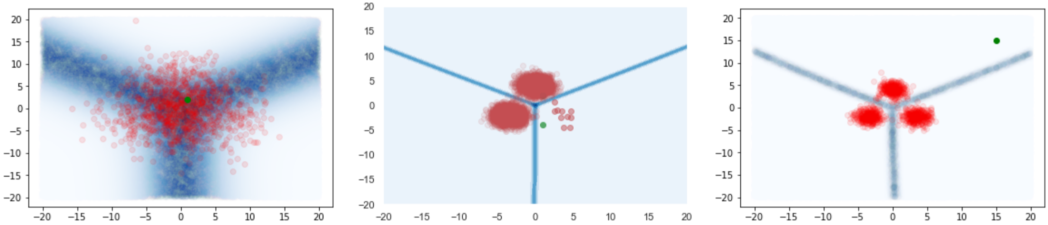
Why is Uncertainty important in practice?

- Machine Learning (ML) systems are being deployed to many applications $\rightarrow$
  - Image Classification, Speech Recognition, Machine Translation, etc...
- In some applications, the cost of a mistake is **high** or consequence **fatal** $\rightarrow$
  - Medical applications, Financial applications and Autonomous vehicles
- Obtaining measures of uncertainty in predictions helps avoid mistakes!
  - Increases safety and reliability of ML system

Scenario

- Given a deployed model and a test input $\boldsymbol{x}^*$ we wish to:
  - Obtain a prediction
  - Obtain a measure of **uncertainty in prediction**
- Take action based estimate of uncertainty
  - Reject prediction / stop decoding sentence
  - Ask for human intervention
  - Use active learning
- Appropriate action depends on source of uncertainty

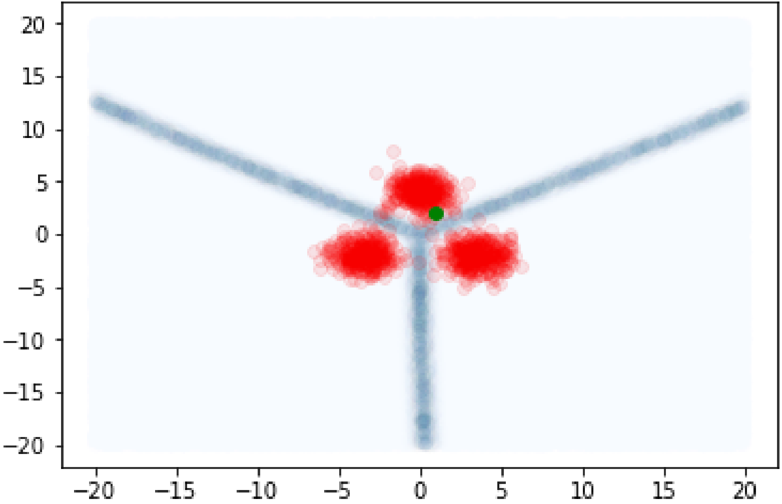
Sources of Uncertainty



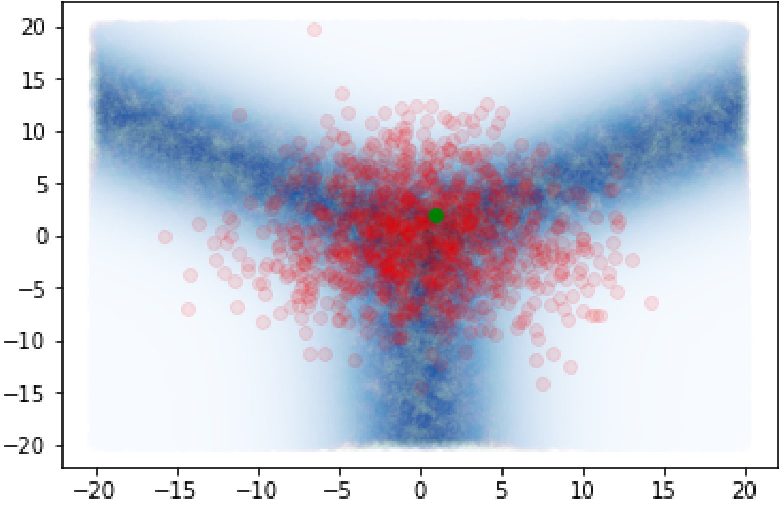
(a) Data Uncertainty (b) Data Sparsity (c) Out-of-Distribution inputs

- Knowledge (epistemic) uncertainty refers to both:
  - Data Sparsity and Out-of-distribution inputs

Data (Aleatoric) Uncertainty

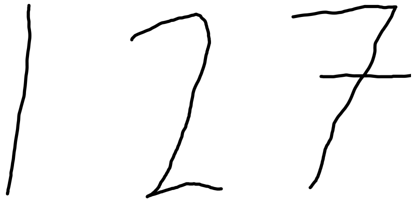


Data Uncertainty



Data Uncertainty

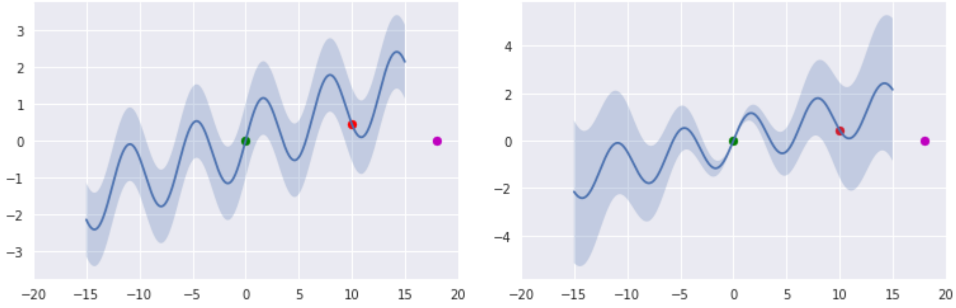
- Distinct Classes



- Overlapping Classes

Data Uncertainty

- In regression tasks data uncertainty takes the form of additive noise



(a) Homoscedastic Noise (b) Heteroscedastic Noise

Data Uncertainty

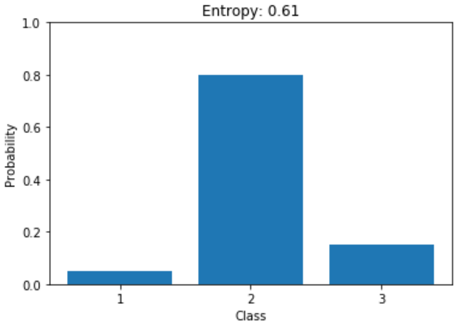
- Data Uncertainty is the *entropy* of the *true data distribution* $\rightarrow$

$$\mathcal{H}[\mathrm{P}_{\mathtt{tr}}(y|\boldsymbol{x}^*)] = -\sum_{c=1}^{K} \mathrm{P}_{\mathtt{tr}}(y=\omega_c|\boldsymbol{x}^*) \ln \mathrm{P}_{\mathtt{tr}}(y=\omega_c|\boldsymbol{x}^*)$$

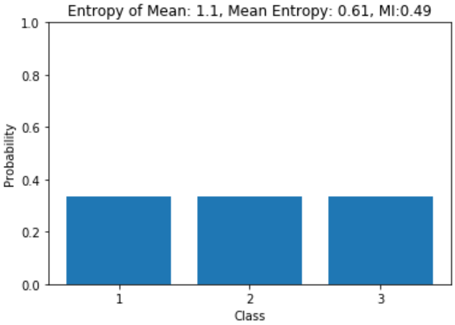
- Captured by the entropy of a model's posterior over classes $\rightarrow$

$$\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})] = -\sum_{c=1}^{K} \mathrm{P}(y=\omega_c|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}}) \ln \mathrm{P}(y=\omega_c|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})$$

Reminder - Entropy



(a) Low Entropy



(b) High Entropy

Data Uncertainty in Regression

- Data Uncertainty is the *differential entropy* of the *true data distribution* $\rightarrow$

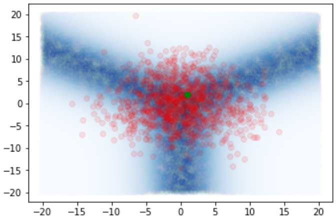
$$\mathcal{H}[\mathrm{p}_{\mathtt{tr}}(\boldsymbol{y}|\boldsymbol{x}^*)] = -\int \mathrm{p}_{\mathtt{tr}}(\boldsymbol{y}|\boldsymbol{x}^*) \ln \mathrm{p}_{\mathtt{tr}}(\boldsymbol{y}|\boldsymbol{x}^*) d\boldsymbol{y}$$

- Captured by the entropy of a model's posterior over classes $\rightarrow$

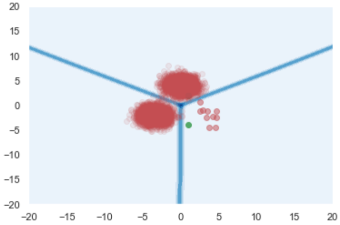
$$\mathcal{H}[\mathrm{p}(\boldsymbol{y}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})] = -\int \mathrm{p}(\boldsymbol{y}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}}) \ln \mathrm{p}(\boldsymbol{y}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}}) d\boldsymbol{y}$$

- Data Uncertainty is captured via Maximum Likelihood Estimation
  - Given sufficient training data, model flexibility and correct output distribution

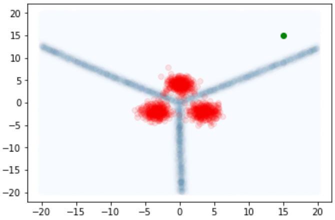
Sources of Uncertainty



(a) Data Uncertainty

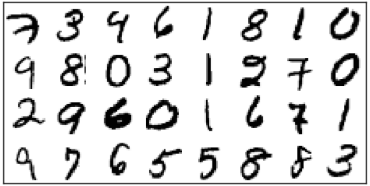


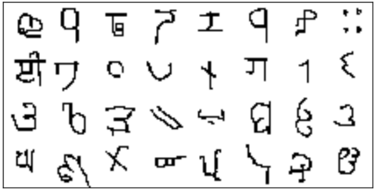
(b) Data Sparsity



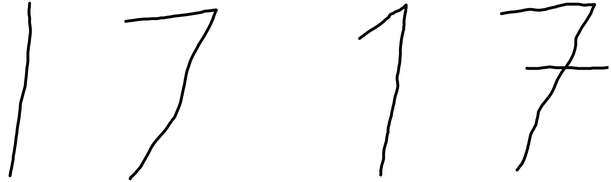
(c) Out-of-Distribution inputs

- Unseen classes





- Unseen variations of seen classes



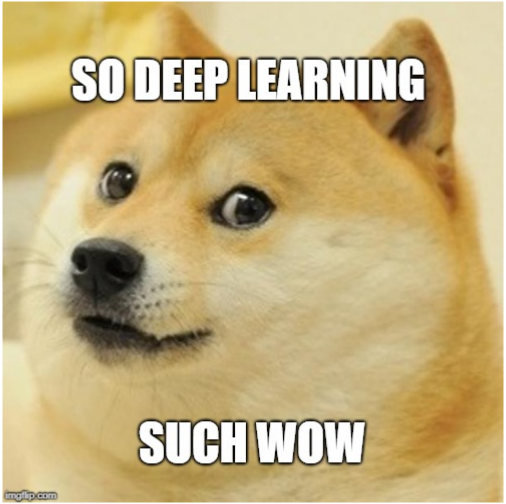
Sources of Uncertainty

- Data Uncertainty $\rightarrow$ **Known-Unknown**
  - Class overlap (complexity of decision boundaries)
  - Homoscedastic and Heteroscedastic noise
- Knowledge Uncertainty $\rightarrow$ **Unknown-Unknown**
  - Test input in out-of-distribution region far from training data
- Appropriate **action** depends on **source** of uncertainty
  - Separating sources of uncertainty requires **Ensemble approaches**

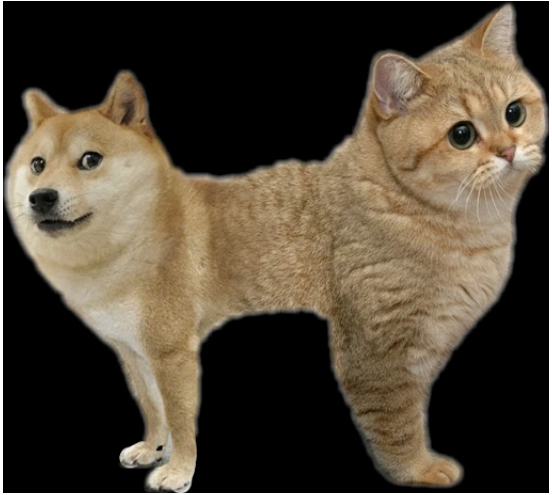
Overview of the Talk

Ensemble Approaches

Ensemble Approaches



Ensemble Approaches



Ensemble Approaches

Ensemble Approaches

- Uncertainty in $\boldsymbol{\theta}$ captured by model posterior $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D}) \rightarrow$

$$\mathtt{p}(\boldsymbol{\theta}|\mathcal{D}) = \frac{\mathtt{p}(\mathcal{D}|\boldsymbol{\theta})\mathtt{p}(\boldsymbol{\theta})}{\mathtt{p}(\mathcal{D})}$$

- Can consider an ensemble of **probabilistic** models $\rightarrow$

$$\{\mathtt{P}(y|\boldsymbol{x},\boldsymbol{\theta}^{(m)})\}_{m=1}^{M},\ \boldsymbol{\theta}^{(m)} \sim \mathtt{p}(\boldsymbol{\theta}|\mathcal{D}),\quad \{\mathtt{p}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}^{(m)})\}_{m=1}^{M},\ \boldsymbol{\theta}^{(m)} \sim \mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$$
$$\mathtt{p}(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{\mu},\boldsymbol{\Lambda}),\quad \{\boldsymbol{\mu},\boldsymbol{\Lambda}\} = \boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta})$$

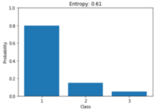
- Bayesian inference of $\mathtt{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta}) \rightarrow$

$$\mathtt{P}(y|\boldsymbol{x},\mathcal{D}) = \mathbb{E}_{\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathtt{P}(y|\boldsymbol{x},\boldsymbol{\theta})] \approx \frac{1}{M}\sum_{m=1}^{M}\mathtt{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta}^{(m)}),\ \boldsymbol{\theta}^{(m)} \sim \mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$$

- $\mathtt{P}(y|\boldsymbol{x}^*,\mathcal{D})$ Is the predictive posterior or **ensemble mean**

Ensemble for certain in-domain input

$$\boldsymbol{\theta}^{(1)} \longrightarrow$$

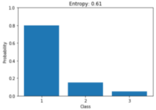


$\rightarrow$

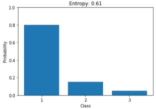
$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(1)})$

$\vdots \qquad \vdots$

$$\mathrm{p}(\boldsymbol{\theta}) + \mathcal{D} \rightarrow \mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \longrightarrow \boldsymbol{\theta}^{(i)} \longrightarrow$$



$\rightarrow$

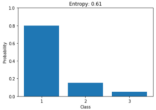


$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)})$

$\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})$

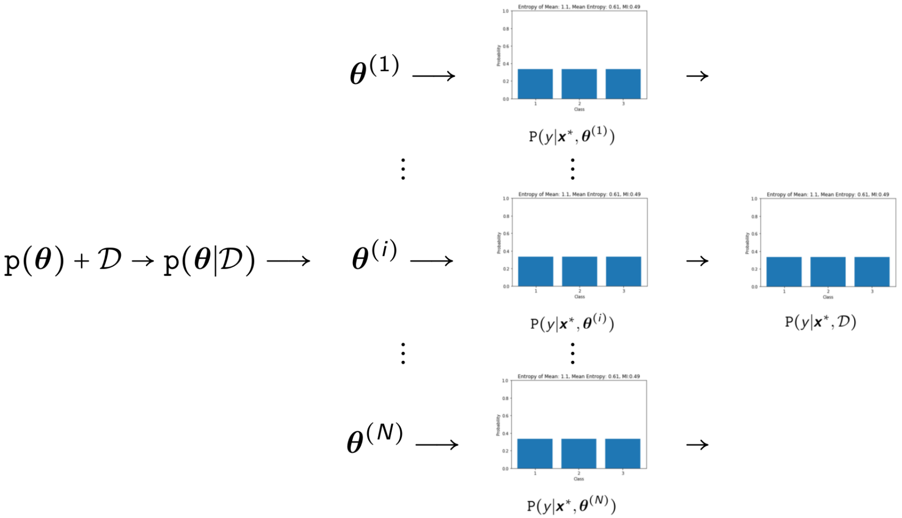
$\vdots \qquad \vdots$

$$\boldsymbol{\theta}^{(N)} \longrightarrow$$



$\rightarrow$

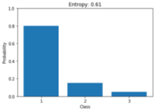
$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(N)})$


$\mathrm{p}(\boldsymbol{\theta}) + \mathcal{D} \rightarrow \mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \longrightarrow$
$\boldsymbol{\theta}^{(1)} \longrightarrow$
$\boldsymbol{\theta}^{(i)} \longrightarrow$
$\boldsymbol{\theta}^{(N)} \longrightarrow$
Entropy of Mean: 1.1, Mean Entropy: 0.61, MI:0.49
Probability
Class
$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(1)})$
$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)})$
$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(N)})$
$\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})$

Ensemble for Out-of-Domain input

$$p(\boldsymbol{\theta}) + \mathcal{D} \rightarrow p(\boldsymbol{\theta}|\mathcal{D}) \longrightarrow$$

$\boldsymbol{\theta}^{(1)} \longrightarrow$

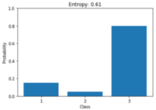


$\rightarrow$

$P(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(1)})$

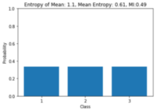
$\vdots$

$\boldsymbol{\theta}^{(i)} \longrightarrow$



$\rightarrow$

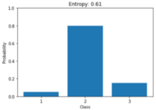
$P(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)})$



$P(y|\boldsymbol{x}^*, \mathcal{D})$

$\vdots$

$\boldsymbol{\theta}^{(N)} \longrightarrow$



$\rightarrow$

$P(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(N)})$

Sources of Uncertainty

- Decompose sources of uncertainty via Mutual Information for classification:

$$\underbrace{\mathcal{I}[y, \boldsymbol{\theta} | \boldsymbol{x}^*, \mathcal{D}]}_{\text{Knowledge Uncertainty}} = \underbrace{\mathcal{H}\big[\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})\big]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\big[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]\big]}_{\text{Data Uncertainty}}$$

- Mutual Information is a measure of ensemble diversity
- Intractable for regression, so use Law of Total Variation:

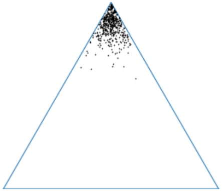
$$\underbrace{\mathbb{V}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\boldsymbol{\mu}]}_{\text{Knowledge Uncertainty}} = \underbrace{\mathbb{V}_{\mathrm{p}(\boldsymbol{y}|\boldsymbol{x},\mathcal{D})}[\boldsymbol{y}]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\boldsymbol{\Lambda}^{-1}]}_{\text{Data Uncertainty}} \tag{1}$$

Overview of the Talk

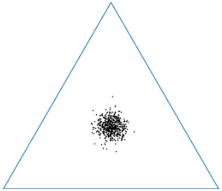
1. Motivation and Sources
2. Uncertainty Estimation via Ensembles
3. **Uncertainty Estimation via Prior Networks**
4. Ensemble Distribution Distillation

Distributions on a Simplex

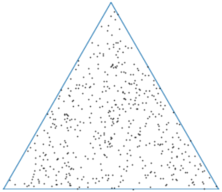
- Ensemble $\{\mathtt{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ can be visualized on a simplex



(a) Confident



(b) Data Uncertainty



(c) Knowledge Uncertainty

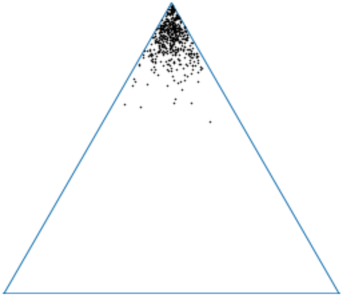
- Same as sampling from **implicit** Distribution over output Distributions

$$\mathtt{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)}) \sim \mathtt{p}(\boldsymbol{\theta}|\mathcal{D}) \equiv \boldsymbol{\pi}^{(m)} \sim \mathtt{p}(\boldsymbol{\pi}|\boldsymbol{x}^*, \mathcal{D})$$

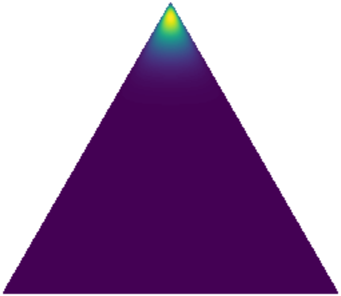
Distributions on a Simplex (cont)

- Expanding out $\boldsymbol{\pi}^{(m)} = \begin{bmatrix} \mathrm{P}(y = \omega_1) \\ \mathrm{P}(y = \omega_2) \\ \vdots \\ \mathrm{P}(y = \omega_K) \end{bmatrix}$, where each $\boldsymbol{\pi}^{(m)}$ is a point on a simplex.

Distribution over Distributions

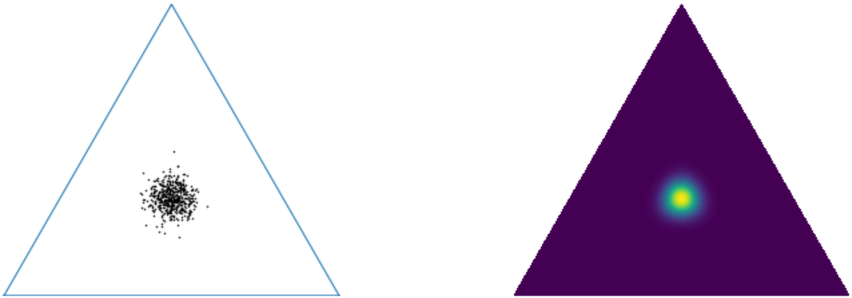


(a) $\{\boldsymbol{\pi}^{(m)}\}_{m=1}^{M}$



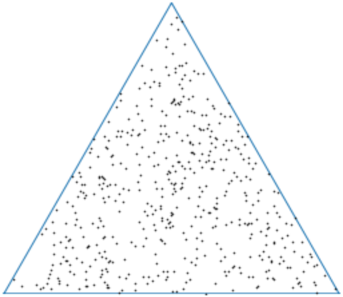
(b) $\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}^*, \mathcal{D})$

Distribution over Distributions

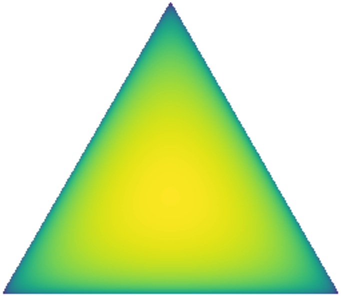


(a) $\{\boldsymbol{\pi}^{(m)}\}_{m=1}^{M}$ (b) $\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}^*, \mathcal{D})$

Distribution over Distributions

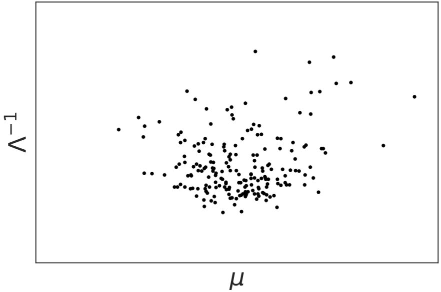


(a) $\{\boldsymbol{\pi}^{(m)}\}_{m=1}^{M}$

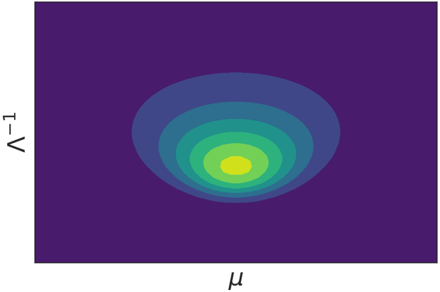


(b) $\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}^*, \mathcal{D})$

Distribution over Distributions - Regression



(c) $\{\boldsymbol{\mu}^{(m)}, \boldsymbol{\Lambda}^{(m)}\}_{m=1}^{M}$



(d) $\mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{x}^*, \mathcal{D})$

Prior Networks [1, 2]

- **Explicitly** model $\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}^*, \mathcal{D})$ and $\mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{x}^*, \mathcal{D})$ using a Prior Network

$$\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) \approx \mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}^*, \mathcal{D})$$
$$\mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{x}^*, \boldsymbol{\theta}) \approx \mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{x}^*, \mathcal{D})$$

- Predictive posterior distribution is given by expected categorical

$$\mathrm{P}(y|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) = \mathbb{E}_{\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})}\big[\mathrm{p}(y|\boldsymbol{\pi})\big] = \hat{\boldsymbol{\pi}}$$
$$\mathrm{p}(\boldsymbol{y}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) = \mathbb{E}_{\mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})}\big[\mathrm{p}(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Lambda})\big]$$

Prior Networks

- A Classification Prior Network parametrizes the **Dirichlet Distribution**

$$\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}^*;\hat{\boldsymbol{\theta}}) = \mathtt{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}), \quad \boldsymbol{\alpha} = \boldsymbol{f}(\boldsymbol{x}^*;\hat{\boldsymbol{\theta}})$$

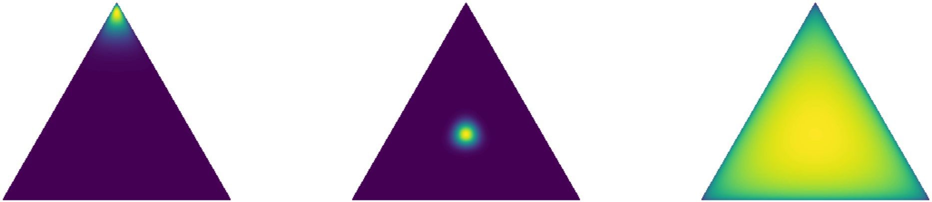
- A Regression Prior Network parameterizes the **Normal-Wishart Distribution**

$$\mathrm{p}(\boldsymbol{\mu},\boldsymbol{\Lambda}|\boldsymbol{x}^*,\boldsymbol{\theta}) = \mathcal{NW}(\boldsymbol{\mu},\boldsymbol{\Lambda}|\boldsymbol{m},\boldsymbol{L},\kappa,\nu), \quad \{\boldsymbol{m},\boldsymbol{L},\kappa,\nu\} = \boldsymbol{\Omega} = \boldsymbol{f}(\boldsymbol{x}^*;\boldsymbol{\theta})$$

- Dirichlet and Normal-Wishart Distributions $\rightarrow$
  - Conjugate priors to Categorical and Normal distributions, respectively.
  - Convenient properties $\rightarrow$ analytically tractable!

Prior Networks

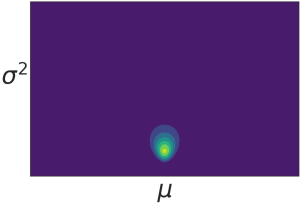
- Construct $\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})$ to emulate classification ensemble



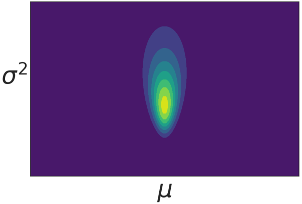
(a) Low Uncertainty (b) Data Uncertainty (c) Knowledge Uncertainty

Prior Networks vs Ensembles

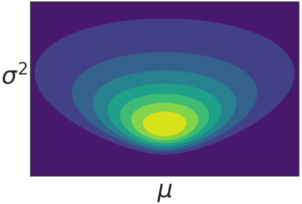
- Construct $\mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda} | \boldsymbol{x}^*, \hat{\boldsymbol{\theta}})$ to emulate regression ensemble



(a) Low uncertainty



(b) Data uncertainty



(c) Knowledge Uncertainty

Prior Networks

- Behaviour of Ensemble distribution over distributions
  - Controlled via prior $\mathrm{p}(\boldsymbol{\theta})$ and inference scheme
- Behaviour of Prior Networks distribution over distributions
  - Controlled via loss function and training data $\mathcal{D}$

Uncertainty Measures for Prior Networks

- Ensemble uncertainty decomposition:

$$\underbrace{\mathcal{I}[y, \boldsymbol{\theta} | \boldsymbol{x}^*, \mathcal{D}]}_{\text{Knowledge Uncertainty}} = \underbrace{\mathcal{H}[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]]}_{\text{Data Uncertainty}}$$

- Prior Network uncertainty decomposition

$$\underbrace{\mathcal{I}[y, \boldsymbol{\pi} | \boldsymbol{x}^*; \hat{\boldsymbol{\theta}}]}_{\text{Knowledge Uncertainty}} = \underbrace{\mathcal{H}\big[\mathbb{E}_{\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}^*;\hat{\boldsymbol{\theta}})}[\mathrm{P}(y|\boldsymbol{\pi})]\big]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}^*;\hat{\boldsymbol{\theta}})}\big[\mathcal{H}[\mathrm{P}(y|\boldsymbol{\pi})]\big]}_{\text{Data Uncertainty}}$$

Uncertainty Measures for Prior Networks

- Ensemble uncertainty decomposition (intractable!):

$$\underbrace{\mathcal{I}[\boldsymbol{y}, \boldsymbol{\theta} | \boldsymbol{x}^*, \mathcal{D}]}_{\text{Knowledge Uncertainty}} = \underbrace{\mathcal{H}[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{p}(\boldsymbol{y}|\boldsymbol{x}^*, \boldsymbol{\theta})]]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathcal{H}[\mathrm{p}(\boldsymbol{y}|\boldsymbol{x}^*, \boldsymbol{\theta})]]}_{\text{Data Uncertainty}}$$

- Prior Network uncertainty decomposition (can be tractable!)

$$\underbrace{\mathcal{I}[\boldsymbol{y}, \{\boldsymbol{\mu}, \boldsymbol{\Lambda}\} | \boldsymbol{x}^*, \mathcal{D}]}_{\text{Knowledge Uncertainty}} = \underbrace{\mathcal{H}\big[\mathbb{E}_{\mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{x}^*, \boldsymbol{\theta})}[\mathrm{p}(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Lambda})]\big]}_{\text{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{x}^*, \boldsymbol{\theta})}\big[\mathcal{H}[\mathrm{p}(\boldsymbol{y}|\boldsymbol{\mu}, \boldsymbol{\Lambda})]\big]}_{\text{Data Uncertainty}}$$

Prior Network Training

$$\mathcal{L}(\boldsymbol{\theta},\mathcal{D}) = \underbrace{\mathcal{L}_{in}(\boldsymbol{\theta},\mathcal{D}_{\texttt{trn}})}_{In\ \ Domain\ \ Loss} + \gamma\cdot\underbrace{\mathcal{L}_{out}(\boldsymbol{\theta},\mathcal{D}_{\texttt{out}})}_{OOD\ \ Loss}$$

(a) In-Domain Target (b) OOD Target

Reverse KL-Divergence Loss and the ELBO

- How to train **Distribution over Distributions** using only $\{y^{(i)}, \boldsymbol{x}^{(i)}\}_{i=1}^{N}$?

Reverse KL-Divergence Loss and the ELBO [3, 2]

- Consider using Bayes' rule as follows:

$$\mathrm{p}(\boldsymbol{\pi}|\hat{\boldsymbol{\alpha}}^{(i)}) \propto \mathrm{p}(y^{(i)}|\boldsymbol{\pi})^{\hat{\beta}}\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0), \quad \mathrm{p}(\boldsymbol{\mu},\boldsymbol{\Lambda}|\hat{\boldsymbol{\Omega}}^{(i)}) \propto \mathrm{p}(\boldsymbol{y}^{(i)}|\boldsymbol{\mu},\boldsymbol{\Lambda})^{\hat{\beta}}\mathrm{p}(\boldsymbol{\mu},\boldsymbol{\Lambda}|\boldsymbol{\Omega}_0)$$

- Minimizing Reverse KL-Divergence induces an ELBO-like loss:

$$\mathtt{KL}[\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x},\boldsymbol{\theta})\|\mathrm{p}(\boldsymbol{\pi}|\hat{\boldsymbol{\alpha}}^{(i)})] = \underbrace{\hat{\beta}\cdot\mathbb{E}_{\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x},\boldsymbol{\theta})}\big[-\ln\mathrm{p}(y|\boldsymbol{\pi})\big]}_{\text{Reconstruction term}} + \underbrace{\mathtt{KL}[\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x},\boldsymbol{\theta})\|\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0)]}_{\text{Prior}} + Z$$

$$\mathtt{KL}[\mathrm{p}(\boldsymbol{\mu},\boldsymbol{\Lambda}|\boldsymbol{x},\boldsymbol{\theta})\|\mathrm{p}(\boldsymbol{\mu},\boldsymbol{\Lambda}|\hat{\boldsymbol{\Omega}}^{(i)})] =$$
$$= \hat{\beta}\cdot\mathbb{E}_{\mathrm{p}(\boldsymbol{\mu},\boldsymbol{\Lambda}|\boldsymbol{x},\boldsymbol{\theta})}\big[-\ln\mathrm{p}(\boldsymbol{y}|\boldsymbol{\mu},\boldsymbol{\Lambda})\big] + \mathtt{KL}[\mathrm{p}(\boldsymbol{\mu},\boldsymbol{\Lambda}|\boldsymbol{x},\boldsymbol{\theta})\|\mathrm{p}(\boldsymbol{\mu},\boldsymbol{\Lambda}|\boldsymbol{\Omega}_0)] + Z$$

- Set $\hat{beta} >> 0$ in-domain and $\hat{beta} = 0$ out-of-domain.

Reverse KL-Divergence Loss and the ELBO

- Prior parameters $\boldsymbol{\alpha}_0$ and $\boldsymbol{\Omega}_0 = \{\boldsymbol{m}_0, \boldsymbol{L}_0, \kappa_0, \nu_0\}$ defined as follows:

$$\boldsymbol{\alpha}_0 = \mathbf{1}$$

$$\boldsymbol{m}_0 = \sum_{i=1}^{N} \frac{\boldsymbol{y}^{(i)}}{N}, \ \boldsymbol{L}_0^{-1} = \frac{\nu_0}{N}\sum_{i=1}^{N}(\boldsymbol{y}^{(i)} - \boldsymbol{m}_0)(\boldsymbol{y}^{(i)} - \boldsymbol{m}_0)^{\mathtt{T}}, \ \kappa_0 = \epsilon, \nu_0 = K + 1 + \epsilon$$

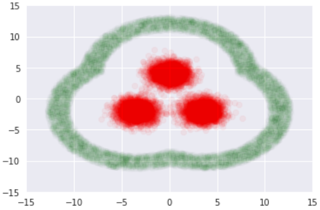
- Prior for classification - uninformative flat Dirichlet Prior
- Prior for regression - semi-informative Prior (uninformative would be improper)

Reverse KL-Divergence Loss and the ELBO

- But how to obtain out-of-domain training data $\mathcal{D}_{OOD} = \hat{\mathrm{p}}_{out}(\boldsymbol{x})$?
  - Use a different dataset, eg: CIFAR10 vs CIFAR100
  - Synthesize using generative model (VAE/GAN)
  - Generate using adversarial attacks
- Choice is highly non-trivial for many tasks (Depth Estimation) $\rightarrow$ main downside!

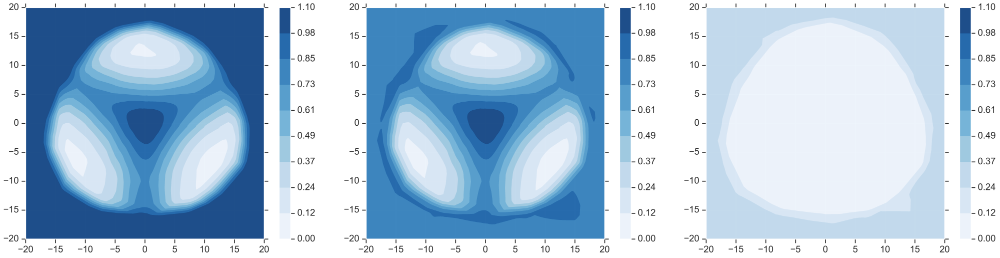
Reverse KL-Divergence Loss and the ELBO

- Out-of-domain (OOD) training data must be on *boundary* on in-domain region $\rightarrow$
  - Too loose $\rightarrow$ Some OOD might be considered in-domain
  - Too tight $\rightarrow$ Some in-domain might be considered OOD



(a) Too Loose (b) Too Tight (c) Good

Prior Networks trained with RKL loss on Artificial Data



(a) Total Uncertainty (b) Data Uncertainty (c) Knowledge Uncertainty

Overview of the Talk

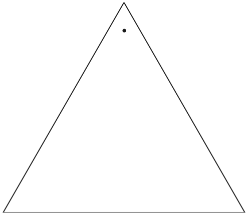
1. Motivation and Sources
2. Uncertainty Estimation via Ensembles
3. Uncertainty Estimation via Prior Networks
4. **Ensemble Distribution Distillation**

Ensemble Distribution Distillation [4, 2]

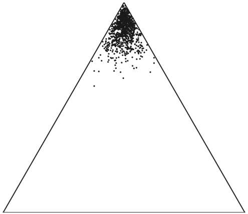
- Ensembles of multiple independently trained models $\{\mathrm{p}(y|\boldsymbol{x}, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$
  - Improved performance
  - Robust uncertainty estimates derived from mean and diversity
  - **Computationally expensive**!
- Ensemble Distillation (EnD) $\rightarrow$ distill ensemble **mean** into a single model
  - Improved performance and low computational cost
  - Lose information about **diversity** $\rightarrow$ cannot separate data and knowledge uncertainty
- **Ensemble Distribution Distillation (EnD^2)** $\rightarrow$
  - Distill mean and diversity of ensemble into single model
  - **Improved performance** and **robust uncertainty** at **low computational cost**

Ensemble Distribution Distillation (EnD^2) for Classification

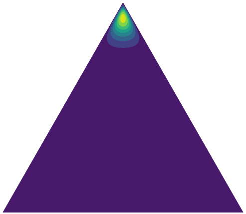
$\frac{1}{M}\sum_{m=1}^{M} \mathrm{P}(y|\boldsymbol{x}, \boldsymbol{\theta}^{(m)}) \quad \longleftarrow \quad \{\mathrm{P}(y|\boldsymbol{x}, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M} \quad \longrightarrow \quad \mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x}; \boldsymbol{\phi})$



(a) EnD



(b) Ensemble



(c) EnD^2

- Distill ensemble distribution (mean and diversity) into a **single** model
  - Fully capture all information about the ensemble

Ensemble Distribution Distillation (EnD^2) for Classification [4]

- Parameterize a Dirichlet distribution using neural network:

$$\mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x};\phi) = \mathtt{Dir}(\boldsymbol{\pi};\boldsymbol{\alpha}), \quad \boldsymbol{\alpha} = \boldsymbol{f}(\boldsymbol{x};\phi), \quad \alpha_c > 0$$

- Training data are ensemble predictions for every input:

$$\mathcal{D} = \left\{\left\{p(y|\boldsymbol{x}^{(i)};\boldsymbol{\theta}^{(m)}), \boldsymbol{x}^{(i)}\right\}_{m=1}^{M}\right\}_{i=1}^{N} \sim \hat{\mathrm{p}}(\boldsymbol{\pi},\boldsymbol{x})$$

- Train via Maximum Likelihood:

$$\mathcal{L}(\phi,\mathcal{D}) = -\mathbb{E}_{\hat{\mathrm{p}}(\boldsymbol{x})}\Big[\mathbb{E}_{\hat{\mathrm{p}}(\boldsymbol{\pi}|\boldsymbol{x})}[\ln \mathrm{p}(\boldsymbol{\pi}|\boldsymbol{x};\phi)]\Big]$$

- Predict using **mean**, derive uncertainty from **mean** and **diversity**

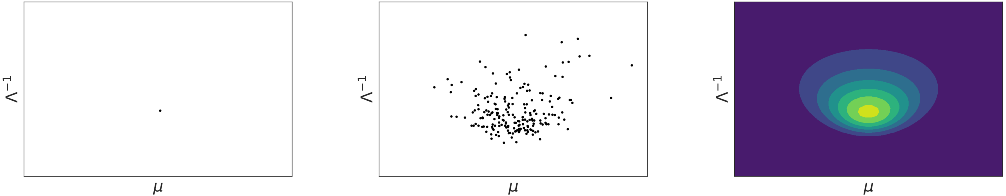
Classification results (% Error and % ROC-AUC)

Method	CIFAR-10	CIFAR-100	TinyImageNet
Single	8.0	30.4	41.8
Ensemble	6.2	26.3	**36.6**
EnD	6.7	28.2	38.5
EnD^2	6.9	28.0	37.3

Model	CIFAR100 vs. LSUN		CIFAR100 vs. TinyImageNet	
	Total Unc.	Knowledge Unc.	Total Unc.	Knowledge Unc.
Ensemble	82.4	**88.4**	76.6	**81.7**
EnD	76.5	-	70.0	-
EnD^2	83.5	86.9	76.4	79.3

Ensemble Distribution Distillation (EnD^2) for Regression [2]

$$\frac{1}{M}\sum_{m=1}^{M} \mathrm{p}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^{(m)}) \longleftarrow \{\mathrm{p}(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M} \longrightarrow \mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{x}; \phi)$$



(a) EnD (b) Ensemble (c) EnD^2

- Distill ensemble distribution (mean and diversity) into a **single** model
  - Fully capture all information about the ensemble

Ensemble Distribution Distillation (EnD^2) for Regression

- Construct a *Regression Prior Network*

$$\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}), \quad \mathcal{NW}(\boldsymbol{\mu}, \boldsymbol{\Lambda} | \boldsymbol{m}, \boldsymbol{L}, \kappa, \nu); \quad \{\boldsymbol{m}, \boldsymbol{L}, \kappa, \nu\} = f(\boldsymbol{x}; \phi)$$

- Training data are ensemble predictions for every input:

$$\mathcal{D} = \Big\{ \Big\{ p(\boldsymbol{y}|\boldsymbol{x}^{(i)}; \boldsymbol{\theta}^{(m)}), \boldsymbol{x}^{(i)} \Big\}_{m=1}^{M} \Big\}_{i=1}^{N} = \Big\{ \{ \boldsymbol{\mu}^{(m,i)}, \boldsymbol{\Lambda}^{(m,i)} \}_{m=1}^{M}, \boldsymbol{x}^{(i)} \Big\}_{i=1}^{N} \sim \hat{\mathrm{p}}(\boldsymbol{\mu}, \boldsymbol{\Lambda}, \boldsymbol{x})$$

- Train via Maximum Likelihood:

$$\mathcal{L}(\phi, \mathcal{D}) = -\mathbb{E}_{\hat{\mathrm{p}}(\boldsymbol{x})} \Big[\mathbb{E}_{\hat{\mathrm{p}}(\boldsymbol{\mu}, \boldsymbol{\Lambda} | \boldsymbol{x})} [\ln \mathrm{p}(\boldsymbol{\mu}, \boldsymbol{\Lambda} | \boldsymbol{x}; \phi)] \Big]$$

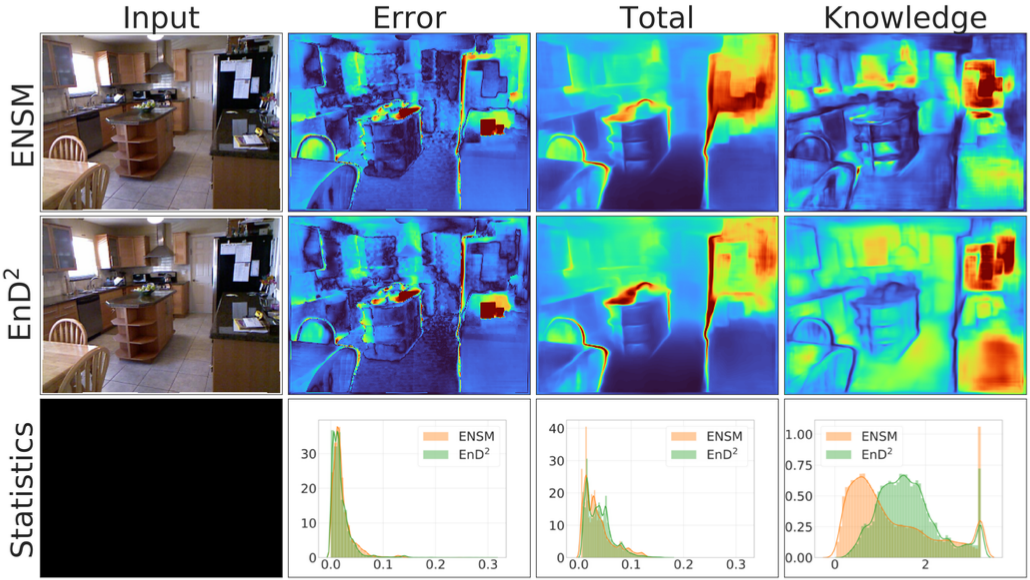
- Predict using **mean**, derive uncertainty from **mean** and **diversity**

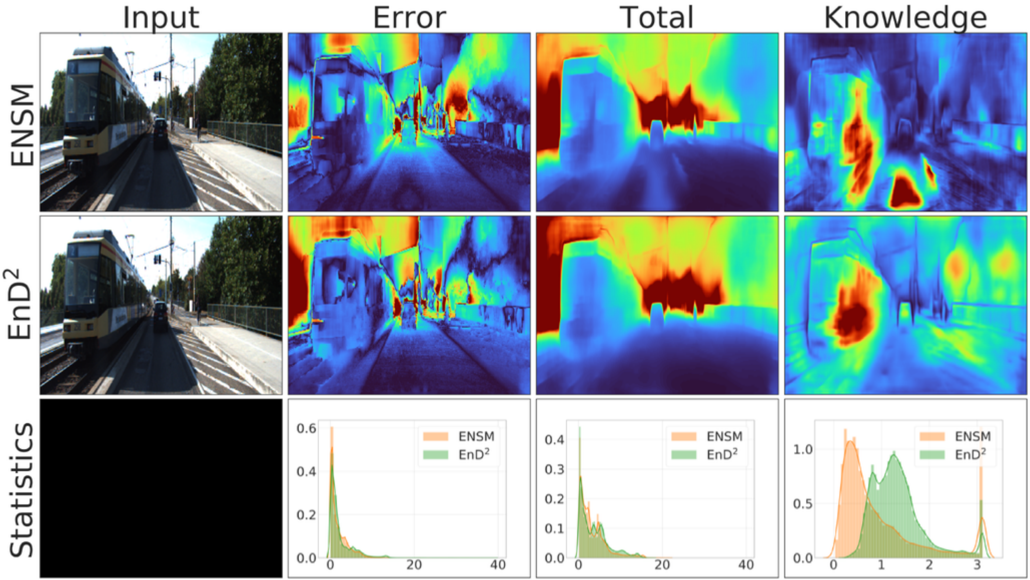
Monocular Depth Estimation - Predictive Performance

Method	NYUv2			KITTI		
	rel(↓)	rmse(↓)	NLL(↓)	rel(↓)	rmse(↓)	NLL(↓)
ENSM 5	**0.117**	**0.438**	0.76	**0.073**	**3.355**	1.94
EnD^2	0.120	0.451	-1.47	0.075	3.367	**1.42**
MD-EnD	0.121	0.451	8.48	0.079	3.446	2.30
DER	0.125	0.464	-1.04	0.078	3.552	1.71

Monocular Depth Estimation - OOD Detection (ROC-AUC)

Method	OOD	NYUv2 vs LSUN		KITTI vs LSUN	
		Total Unc.	Knowledge Unc.	Total Unc.	Knowledge Unc.
ENSM	LSN-B	72.3	74.5	03.2	82.2
EnD^2 (Our)		73.3	**81.7**	1.7	**88.7**
MD-EnD		63.0	50.2	0.4	44.8
ENSM	LSN-C	88.7	88.6	03.6	77.9
EnD^2 (Our)		89.3	**96.4**	2.0	**83.4**
DER		87.7	87.8	03.5	04.
MD-EnD		69.8	42.2	01.2	50.6


Input
Error
Total
Knowledge
ENSM
EnD^2
Statistics
ENSM
EnD^2
30
20
10
0
0.0
0.1
0.2
0.3
40
30
20
10
0
0.0
0.1
0.2
0.3
1.00
0.75
0.50
0.25
0.00
0
2



Input
Error
Total
Knowledge
ENSM
EnD²
Statistics
0.6
0.4
0.2
0.0
0
20
40
ENSM
EnD²
0.4
0.3
0.2
0.1
0.0
0
20
40
ENSM
EnD²
1.0
0.5
0.0
0
1
2
3
ENSM
EnD²

Open questions

- EnD^2 - still a single model $\rightarrow$ evaluate robustness
- Are Dirichlet and Normal-Wishart appropriate?
  - Do we need to model ensemble in model detail?
  - Do we need to only capture bulk properties?
- Do we need auxiliary training data? Mixup?
- Can we use EnD^2 for analysis?
- Can we combine ensemble generation and EnD^2?

Thank you! Questions?

References I

[1] Andrey Malinin and Mark Gales,
"Predictive uncertainty estimation via prior networks,"
in *Advances in Neural Information Processing Systems*, 2018, pp. 7047–7058.

[2] Andrey Malinin, Sergey Chervontsev, Ivan Provilkov, and Mark Gales,
"Regression prior networks,"
arXiv preprint arXiv:2006.11590, 2020.

[3] Andrey Malinin and Mark JF Gales,
"Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness,"
in *Advances in Neural Information Processing Systems*, 2019.

References II

[4] Andrey Malinin, Bruno Mlodozeniec, and Mark JF Gales, "Ensemble distribution distillation," in *International Conference on Learning Representations*, 2020.