Weekly Study Report

Wang Ma

2025-02-18

Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute

Outline

(ICLR 2025) Enhancing Uncertainty Estimation and Interpretability with Bayesian Non-negative Decision Layer

1. Enhancing Uncertainty Estimation and Interpretability with Bayesian Non-negative Decision Layer

1.1 Basic Deep Neural Networks

In the multi-classification problem:

$$p(y_i|x) = \frac{\exp(\boldsymbol{z}w_i^T + b_i)}{\sum_i \exp(\boldsymbol{z}w_i^T + b_i)},$$

where $\boldsymbol{z}w_i^T + b_i$ is called *logits*. In Last-Layer Lapalce Approximation, they model the weights matrix W of last layer to be random variables; this paper does more.

The classification process with softmax can be seen as the special case of generative process of label y (when $p(z|x) \sim \delta$ distribution):

$$p(y|x) = . \int_z p(y|z)p(z|x)dz.$$

1.2 Bayesian Non-negative Descision Layer

If we model $\boldsymbol{z}$ as a latent variable θ, which follows some distribution, the above equation can be improved to

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

Notice, that θ models the output of previous layers.

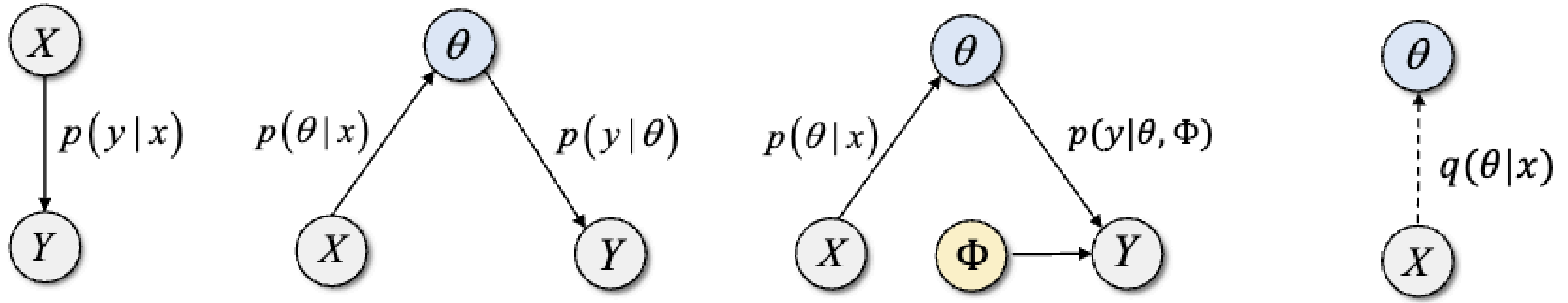
To further account for epistemic uncertainty, we also model the last-layer's weight W to follow some distributions Φ. The we can have the following model:

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

1.2 Bayesian Non-negative Descision Layer

theta: output of previous layers

Phi: the weight matrix of last-layer



(a) DNNs

(b) Extend DNNs with θ

(c) Generative Model of BNDL

(d) Inference Model of BNDL

1.3 The choice of Distributions

Gamma distribution: non-linearity and non-negativity (also conjugate).

$$y_j \mid \theta_j \sim \text{Categaory } \left(\theta_j, \Phi\right),$$
$$\theta_j | x_j \sim \Gamma(1, 1),$$
$$\Phi \sim \Gamma(1, 1).$$

This choice allows for a more comprehensive capture of complex relationships within the model, while also accommodating the characteristics of sparse non-negative variables, thereby enhancing the disentanglement and interpretability of the learned representations θ and decision layers Φ.

About learning the distributions:

1. Weibull approximate posterior instead of direct Gamma distribution
2. Variational Inferecen to estimate poesterir

1.4 Personal Analysis

1. Uncertainty Quantification

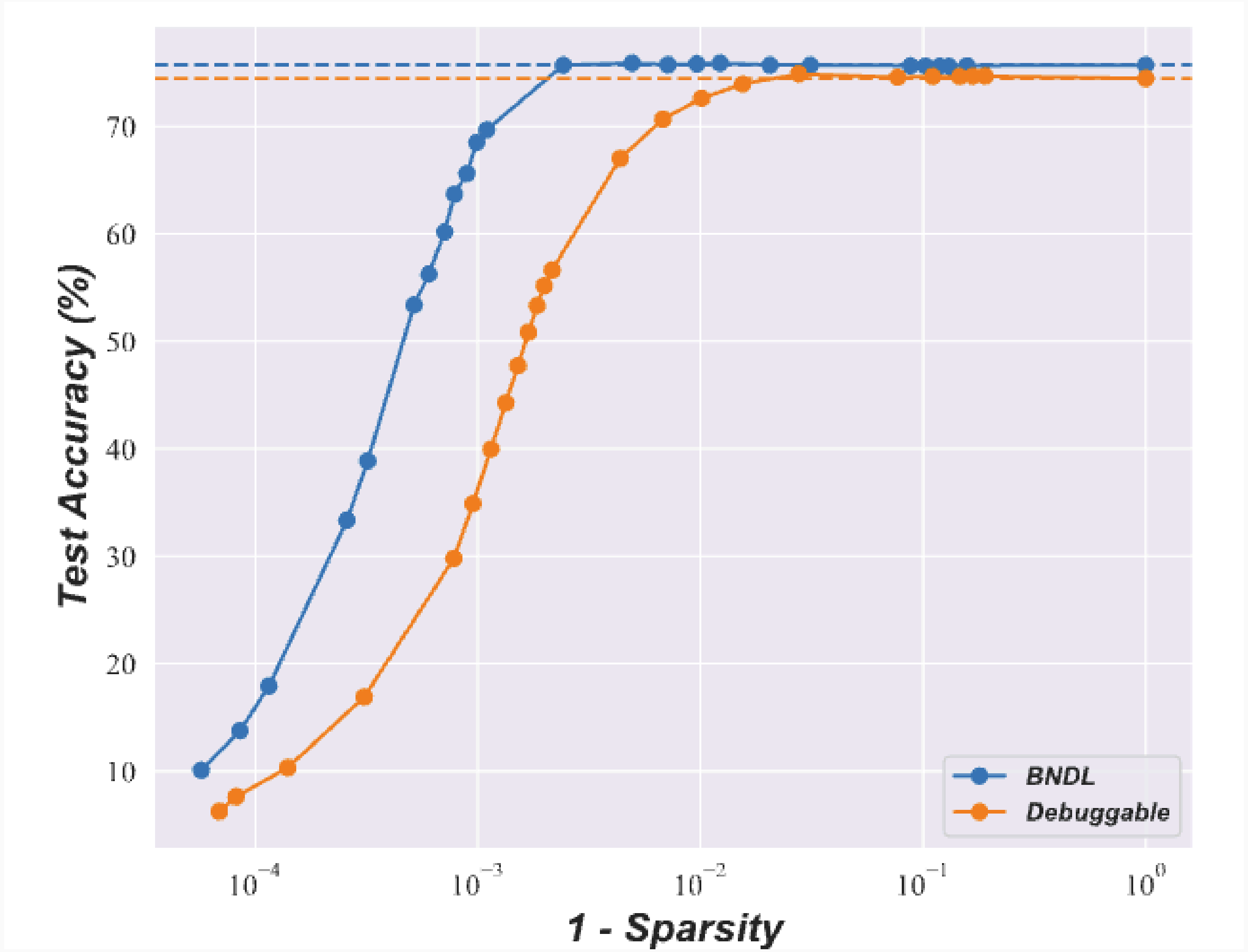
   Like traditional single-model ways, we just sample many times to get an ensemble of results to computer uncertainty. But this model allows to only pass 1 forward process and get the distribution of θ and Φ, we just sample from them to generate multiple resutls.

2. The behavior of θ and Φ

   θ: the output of previous layers; Φ the weight matrix of last layer.

   - **Φ: A learned map: feature $\rightarrow$ Class**. If the model is well-trained, then the variance of Φ's every column should be small. If the model has a high variance of a column, then it means the model does not have enough knowledge of the corresponding class.
   - **θ: the output of previous layers, $\theta_{f(x)}$.**
     - *in-distribution and high quality data*: more sharp at the mean and variance is small
     - *ood data*: θ is flat and has a high variance
     - *low-quality data*: θ will have higher variance than id data

1.5 Sparsity and Interpretability



1.5 Sparsity and Interpretability

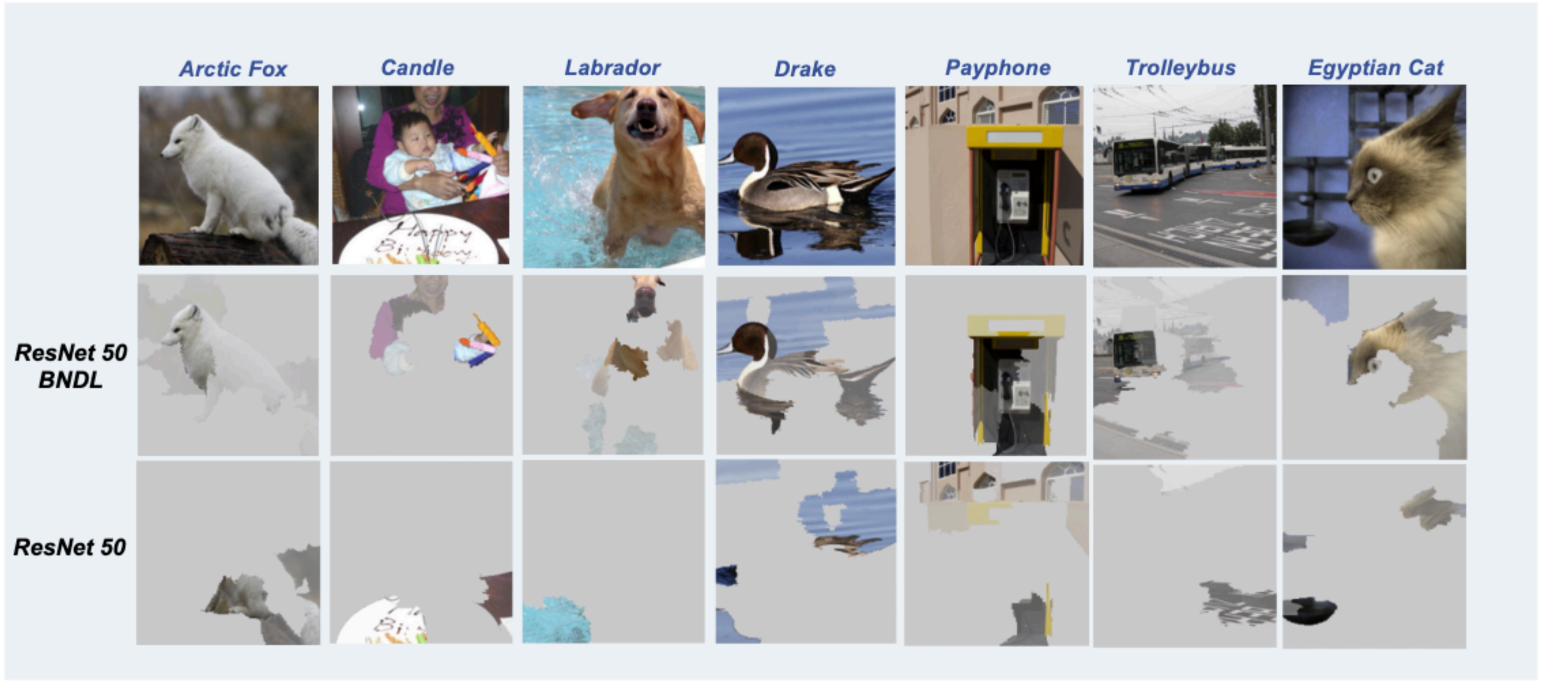
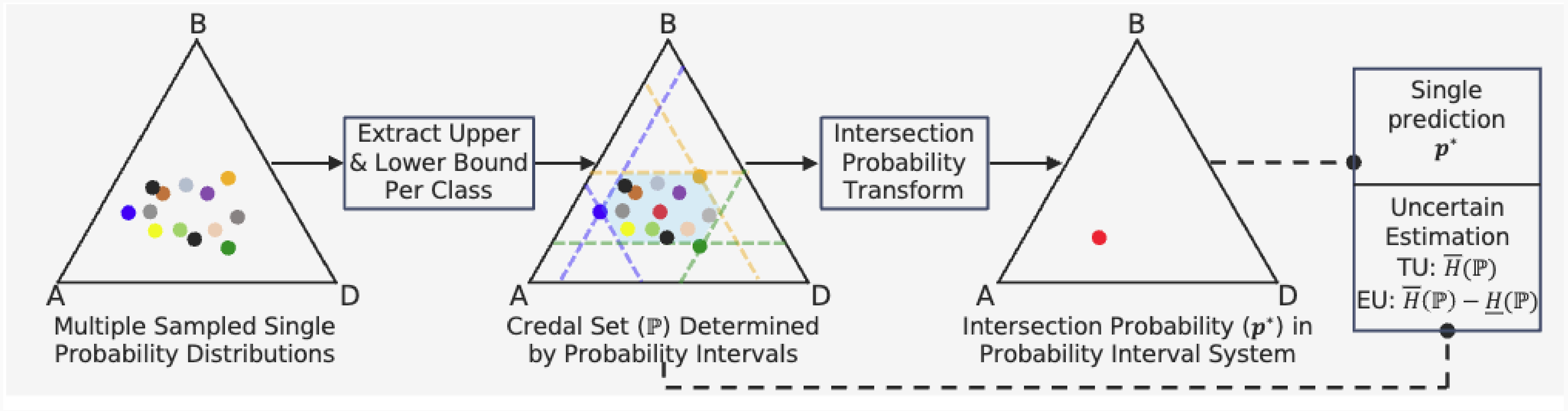


Figure 4: The LIME visualizations for BNDL and ResNet-50, focusing on the largest $\boldsymbol{\theta}$ for each image, show that BNDL's features align more closely with the semantic meaning of true labels, suggesting more disentangled representations. As we selected the top-10 super-pixels for visualization, the results may include some less significant super-pixels; this issue is alleviated when we reduce the number of top-k super-pixels.

(ICLR 2025) Credal Wrapper of Model Averaging for Uncertainty Estimation on Out-Of-Distribution Detection

2. Credal Wrapper of Model Averaging for Uncertainty Estimation on OOD Detection

2. Credal Wrapper of Model Averaging for Uncertainty Estimation on OOD Detection


B
A
D
Multiple Sampled Single Probability Distributions
Extract Upper & Lower Bound Per Class
Credal Set ($\mathbb{P}$) Determined by Probability Intervals
Intersection Probability Transform
Intersection Probability (p^*) in Probability Interval System
Single prediction p^*
Uncertain Estimation
TU: $\overline{H}(\mathbb{P})$
EU: $\overline{H}(\mathbb{P}) - \underline{H}(\mathbb{P})$

Generation of Credal Sets:

$$p_{U_k} = \max_{n=1,\dots,N} p_{n,k} \text{ and } p_{L_k} = \min_{n=1,\dots,N} p_{n,k}$$

2.1 Credal Sets and Uncertainty Quantification

Credal Sets.

$$\mathbb{P} = \left\{\boldsymbol{p} \mid p_k \in \left[p_{L_k}, p_{U_k}\right], \forall k = 1, 2, ..C\right\} s.t. \sum_{k=1}^{C} p_{L_k} \leq 1 \leq \sum_{k=1}^{C} p_{U_k}.$$

Uncertainty in Credal Nets.

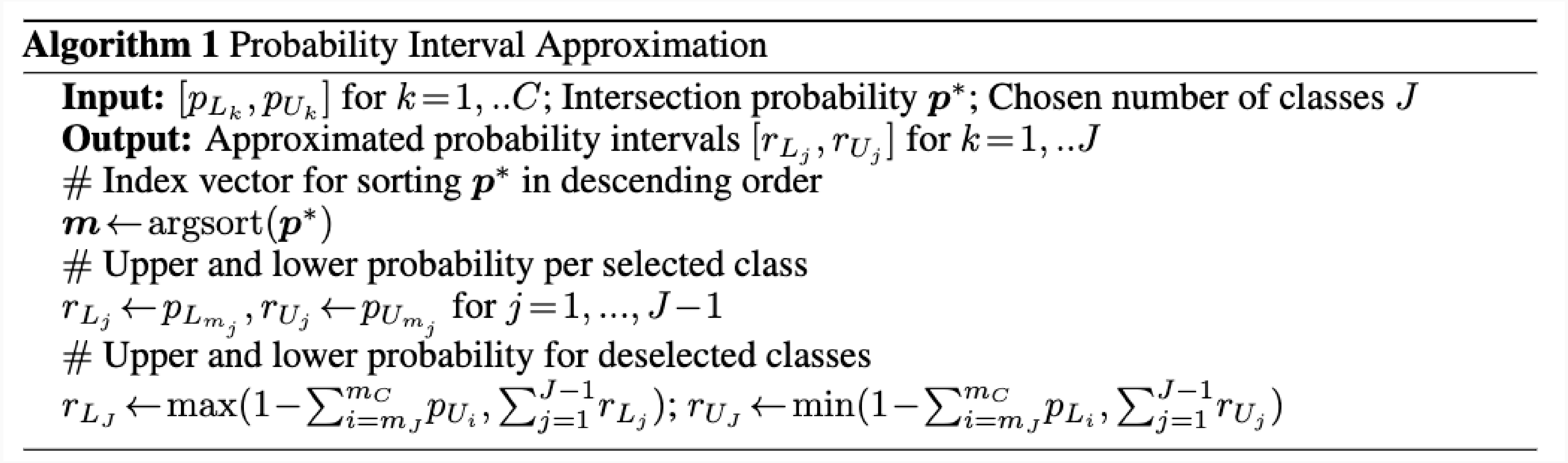
- Total Uncertainty: $\overline{H}(\mathbb{P}) = \max \sum_{k}^{C} -p_k \log p_k$
- Aleatoric Uncertainty: $\underline{H}(\mathbb{P}) = \min \sum_{k}^{C} -p_k \log p_k$
- Epistemic Uncertainty: $\text{EU} = \overline{H}(\mathbb{P}) - \underline{H}(\mathbb{P})$

where $\sum_{k}^{C} p_k = 1.$

2.2 Complexity Reduction and Intersection Probability

2.2.1 Computational Complexity Reduction Method

Key idea: consider the unimportant classes together



Algorithm 1 Probability Interval Approximation

```
**Input:** $[p_{L_k}, p_{U_k}]$ for $k=1,..C$; Intersection probability $\boldsymbol{p}^*$; Chosen number of classes $J$
**Output:** Approximated probability intervals $[r_{L_j}, r_{U_j}]$ for $k=1,..J$
# Index vector for sorting $\boldsymbol{p}^*$ in descending order
$\boldsymbol{m} \leftarrow \text{argsort}(\boldsymbol{p}^*)$
# Upper and lower probability per selected class
$r_{L_j} \leftarrow p_{L_{m_j}}, r_{U_j} \leftarrow p_{U_{m_j}}$ for $j=1,...,J-1$
# Upper and lower probability for deselected classes
$r_{L_J} \leftarrow \max(1-\sum_{i=m_J}^{m_C} p_{U_i}, \sum_{j=1}^{J-1} r_{L_j})$; $r_{U_J} \leftarrow \min(1-\sum_{i=m_J}^{m_C} p_{L_i}, \sum_{j=1}^{J-1} r_{U_j})$
```

2.2.2 Intersection Probability

Key idea: to get a overall predictive probability from ensemble models, and do not use the average directly.

Normally, we know the true probability has a following expression:

$$p_k^* = p_{L_k} + \alpha_k \cdot \left(p_{U_k} - p_{L_k}\right) \quad \forall k = 1, ..., C.$$

The author wants to find a unified α which can be applied to every class:

$$p_k^* = p_{L_k} + \alpha \cdot \left(p_{U_k} - p_{L_k}\right) \quad \forall k = 1, ..., C.$$

2.2 Complexity Reduction and Intersection Probability

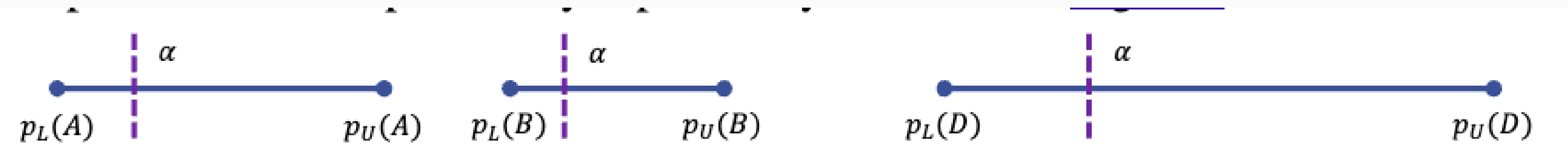


Figure 2: An illustration of the intersection probability for a probability interval system of three classes {A, B, D}[13;12].

$$p_k^* = p_{L_k} + \alpha \cdot \left(p_{U_k} - p_{L_k}\right) \quad \forall k = 1, \ldots, C.$$

By just confirming that $\sum_k^C p_k^* = 1$, we can get that $\alpha = \frac{1 - \sum_k^C p_{L_k}}{\sum_k^C \left(p_{U_k} - p_{L_k}\right)}$.

–

My idea: this way is problematic, because if α is same, then the model tends to have higher prediction on the unsure classes!

Pretrained Visual Uncertainties

3. Pretrained Visual Uncertainties

3. Pretrained Visual Uncertainties

- Not real “Uncertainty” — they consider the loss as the “uncertainty”
- Strength: they can train the *uncertainty head* from the trained backbone NN, and the *uncertainty head* can transfer zero-shot.

3.1 Pretrained Visual Uncertainties: The Methods

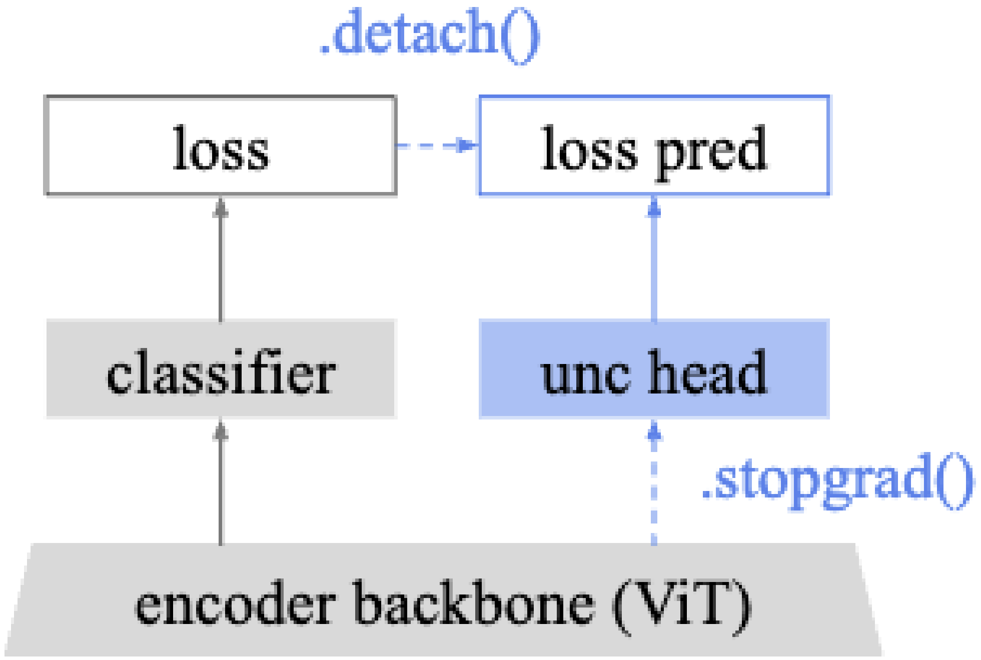


Figure 2. Pretrained uncertainties are returned by an auxiliary head (blue) that is trained to predict the classification loss of each image.

3.1 Pretrained Visual Uncertainties: The Methods

If trained jointly, the loss function:

$$L = L_{\text{task}}(y, f(x)) + \left(u(e(x)) - L_{\text{task}}^{\text{det}}(y, f(x))\right)^2.$$

The Uncertainty Head module u is implemented as a small MLP, and $e(x)$ is the model representation(feature) outputed by the backbone.

If trained separately , the loss function:

$$L = \left(u(e(x)) - L_{\text{task}}^{\text{det}}(y, f(x))\right)^2,$$

which can be optimized efficiently because we only optimize a 3-layer MLP.

3.2 To transfer

With the current L_2 loss, the uncertainty head module can match the scale of the current backbone. However, a downstream user might switch to a different loss on a different scale ◦

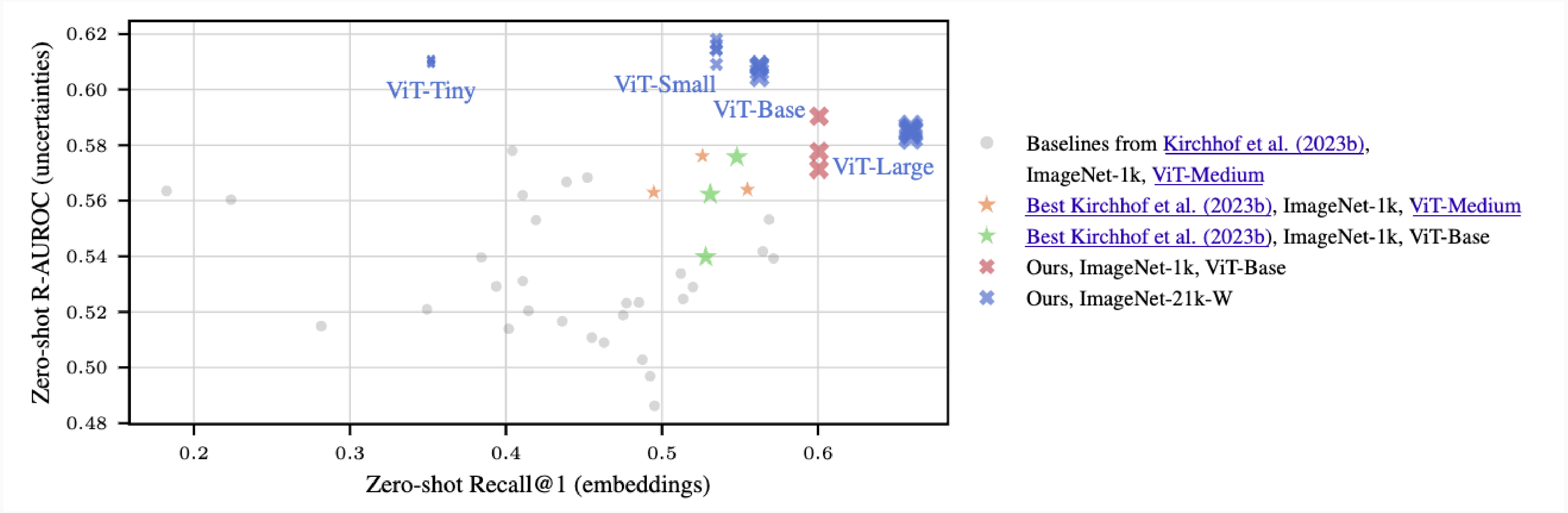
Solution: Ranking-based objective

$$L = \max(0, \mathbb{1}_L \cdot (u(e(x_1)) - u(e(x_2)) + m))$$

$$s.t. \mathbb{1}_L := \begin{cases} +1 \text{ , if } L(x_1) > L(x_2) \\ -1 \text{ , else} \end{cases}.$$

For every pair of images x_1 and x_2, the indicator function compares which images has the higher primary task loss. The the uncertainty u of that sample is forced to be higher than that of the other sample, by a margin of at least $m = 0.1$.

Implementation: Sampling pairs from a mini-batch, and summing up all the ranking-based loss


0.62
0.60
0.58
0.56
0.54
0.52
0.50
0.48
Zero-shot R-AUROC (uncertainties)
0.2
0.3
0.4
0.5
0.6
Zero-shot Recall@1 (embeddings)
ViT-Tiny
ViT-Small
ViT-Base
ViT-Large
● Baselines from Kirchhof et al. (2023b), ImageNet-1k, ViT-Medium
★ Best Kirchhof et al. (2023b), ImageNet-1k, ViT-Medium
★ Best Kirchhof et al. (2023b), ImageNet-1k, ViT-Base
✖ Ours, ImageNet-1k, ViT-Base
✖ Ours, ImageNet-21k-W

3.2 To transfer

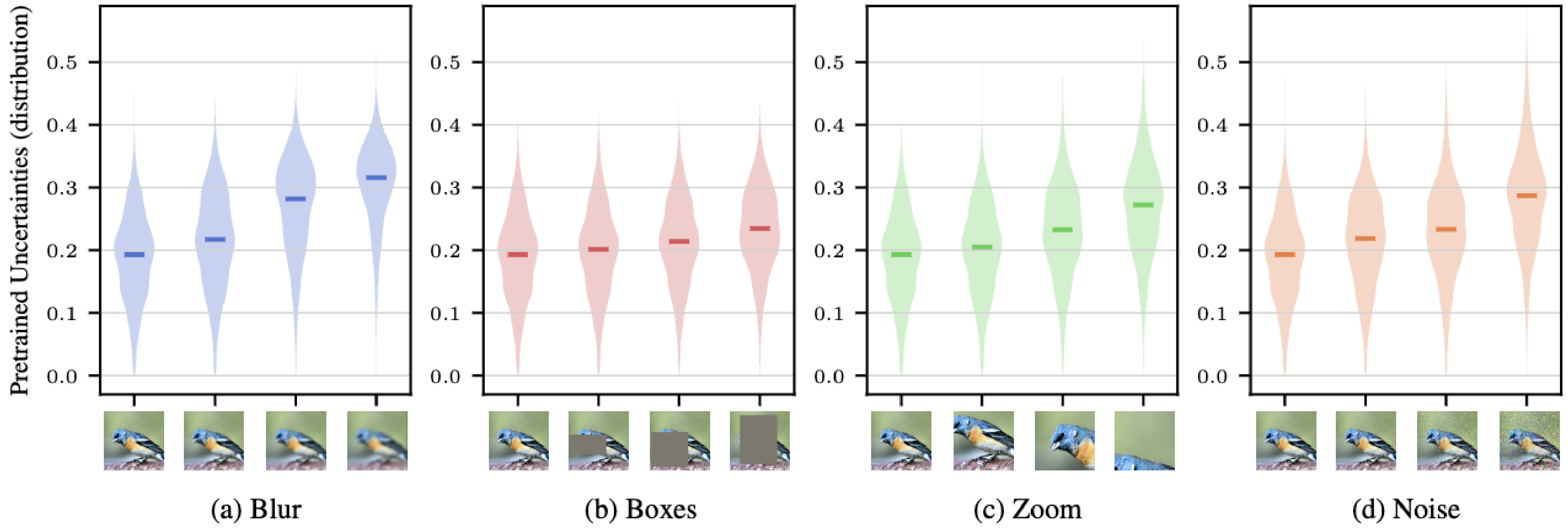


Figure 6. Pretrained uncertainties grow as images are deteriorated. Distributions and medians over unseen datasets (CUB, CARS, SOP). Note that except zooming, the pretrained model is not exposed to these data augmentations during training.

3.2 To transfer

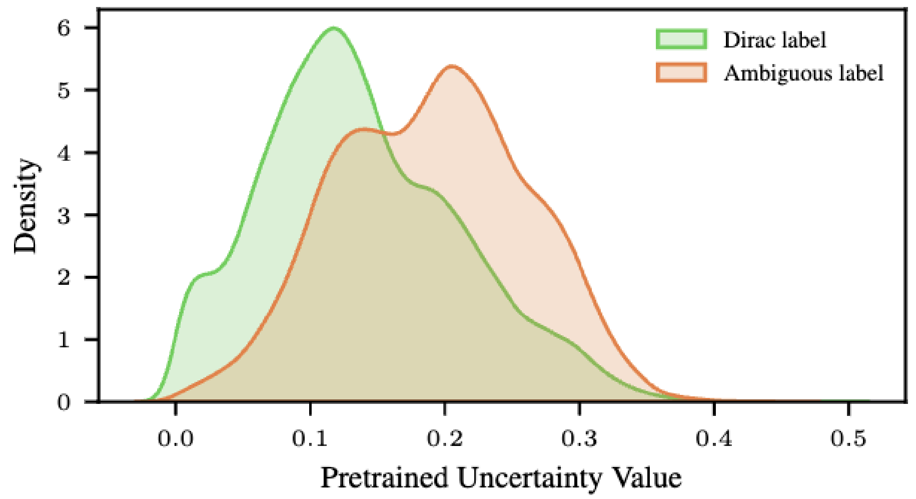


Figure 7. Pretrained uncertainties are systematically higher for ImageNet ReaL-H images with multiple possible labels.

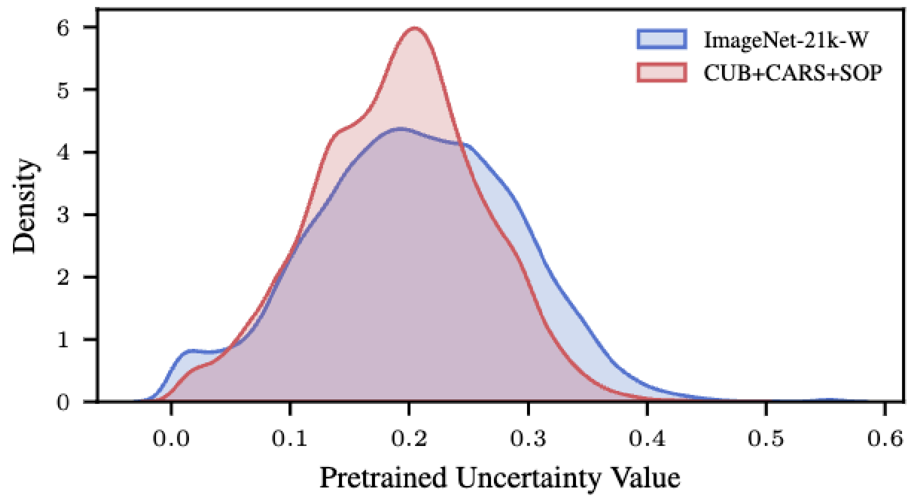


Figure 8. Pretrained uncertainties are consistent between train and unseen datasets, indicating the absence of epistemic uncertainty.

3.3 My idea: combine with real uncertainty?

If we have a model (single model or ensemble) which can do uncertainty quantification, for every training x_i, we have the uncertainty $\mathrm{unc}(x_i)$, then we can train a real "uncertainty head":

$$L = (u(x_i) - \mathrm{unc}(x_i))^2,$$

while we can also use the ranking-based objective to confirm the transferability.