Weekly Study Report

Wang Ma

2025-1-17

Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute

Outline

1. Uncertainty Derivation Results for PNC-predictor 2
2. Experiments Reuslts for Causal Saliency Map 9
3. Experiments Attempts for CredalNets 14
4. Plans for next week 19

1. Uncertainty Derivation Results for PNC-predictor

1.1 PNC-Predictor

- $\hat{h}_{n,\theta_b}$: trained single model on dataset with initialization θ_b.
- $\{(x_i, \hat{s}(x_i))\}$, artificial dataset, where $\hat{s}(x) = \mathbb{E}_{\theta_b}\left[s_{\theta_b}(x)\right]$, s_{θ_b} is an θ_b-initialized NN.
- φ'_{n,θ_b}: auxillary network trained on $\{(x_i, \hat{s}(x_i))\}$, denote $\varphi_{n,\theta_b}(x) = \varphi'(x) - \hat{s}(x)$.
- $h^* = \hat{h}_{n,\theta_b} - \varphi_{n,\theta_b}$.

1.2 What we discussed last week

1. $$\begin{aligned}\operatorname{var}(h^*) = \operatorname{var}\left(\hat{h} - \varphi\right) &= \operatorname{var}\left(\hat{h}\right) + \operatorname{var}(\varphi) \quad \text{(independent)} \\ &= \operatorname{var}\left(\hat{h}\right) + \operatorname{var}(\varphi) - 2\operatorname{cov}\left(\hat{h}, \varphi\right)\end{aligned}$$

2. $$f(\theta_h, x) = f(\theta_b, x) + \frac{\partial f}{\partial \theta}(\theta_h - \theta_b)$$

   $$f\left(\theta_\varphi, x\right) = f(\theta_b, x) + \frac{\partial f}{\partial \theta}\left(\theta_\varphi - \theta_b\right)$$

3. For regression $h^* = \hat{h}_{n,\theta_b} - \varphi_{n,\theta_b}$, but for classification, it's weird to directly plus or minus probability, we need to find some equivalent operation in probability measure.

1.3 To Calculate the Uncertainty

In Regression:

- Aleatoric Uncertainty: $\mathbb{E}[\mathrm{var}(y)]$
- Epistemic Uncertainty: $\mathrm{var}(\mathbb{E}(y))$

For a single neural network, it is hard to get variance information or expectation information without perturbation. To effectively get such information, we can do it like evidential deep learning, modeling the output of the NN as a distribution, which allows us to easily get variance/expectation information.

Let h be a NN, suppose $h_\theta(x) \sim N(\mu(x,\theta), \sigma^2(x,\theta))$, which can be optimized use negative log-likelihood.

1.4 If $h_\theta(x) \sim N(\mu(x,\theta), \sigma^2(x,\theta))$

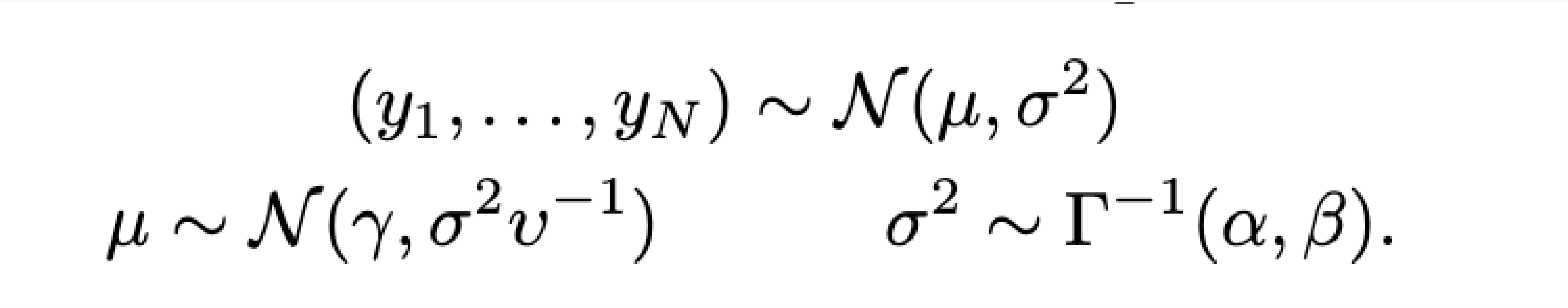
- $\hat{h}_{n,\theta_b}(x) \sim N(\mu(x), \sigma^2(x))$, $\hat{h}_{n,\theta_b}(x) = \mu(x) + \sigma(x) * \varepsilon$, where $\varepsilon \sim N(0,1)$
- $\{(x_i, \hat{s}(x_i))\}$, artificial dataset, where $\hat{s}(x) = \mathbb{E}_{\theta_b}\left[s_{\theta_b}(x)\right]$, s_{θ_b} is a Normal Distribution. Assume different s_{θ_b}s are independent, then $\hat{s}(x)$ is also a normal distribution, with $\mathbb{E}[\hat{s}(x)] = \mathbb{E}\left[\mathbb{E}\left[s_{\theta_b}(x)\right]\right]$, $\mathrm{var}(\hat{s}(x)) = \frac{1}{n}\mathbb{E}\left[\mathrm{var}\left(s_{\theta_b}(x)\right)\right]$ (if we assume different $s_{\theta_b}(x)$) are independent.
- φ'_{n,θ_b}: auxillary network trained on $\{(x_i, \hat{s}(x_i))\}$, denote $\varphi_{n,\theta_b}(x) = \varphi'(x) - \hat{s}(x)$. Here the key is, we know φ is still a gaussian, but since φ' and $\hat{s}$ should be dependent, calculating the variance is a problem.
- $h^* = \hat{h}_{n,\theta_b} - \varphi_{n,\theta_b}$, still a gaussian.

1.5 Under the settings

1. $h^* \sim N(\mu^*, \sigma^{*2})$
2. Aleatoric: $\mathbb{E}[\mathrm{var}(h^*)] = \mathbb{E}[\sigma^{*2}]$
3. Epistemic: $\mathrm{var}(\mathbb{E}[h^*]) = \mathrm{var}(\mu^*)$

So the key is, if μ^* and σ^{*2} follow some distributions, then we can do UQ directly without perturbation.

1.6 Normal Inverse Gamma Distribution



$$(y_1, \ldots, y_N) \sim \mathcal{N}(\mu, \sigma^2)$$

$$\mu \sim \mathcal{N}(\gamma, \sigma^2 \upsilon^{-1}) \qquad \sigma^2 \sim \Gamma^{-1}(\alpha, \beta).$$

It was used to model the output of NN in *Deep Evidential Regression* (NeurIPS 2020), so there is a complete solution to use it.

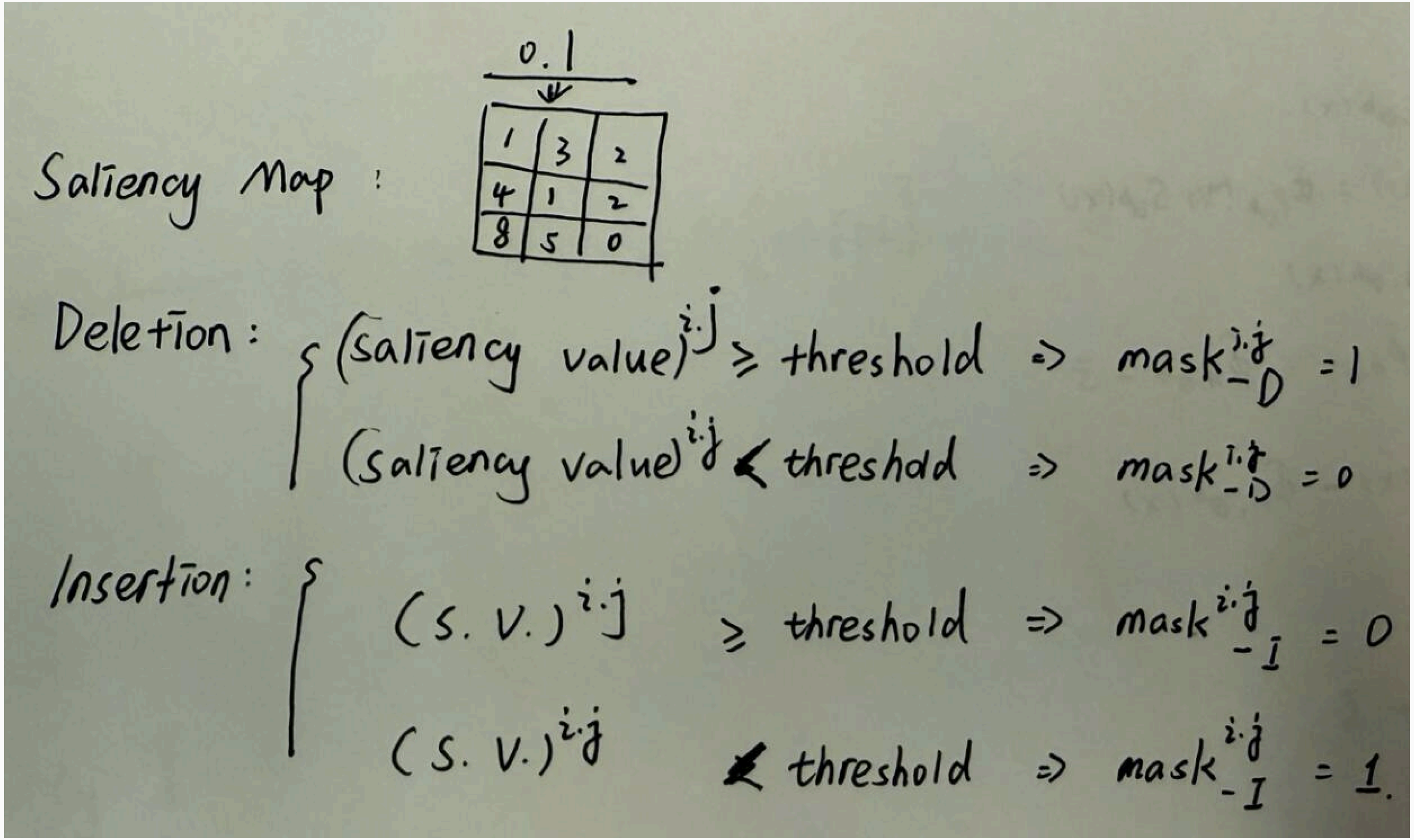
We can Improve.

Calculating AU ($\mathbb{E}[\mathrm{var}(h^*)]$), is sometimes not useful. EU($\mathrm{var}(\mathbb{E}[h^*]) = \mathrm{var}(\mu^*)$) is more useful in downstream tasks. We can still model $y \sim N(\mu, \sigma^2)$, but only model μ as a random variable, and keep σ^2 deterministic.

2. Experiments Reuslts for Causal Saliency Map

2.1 Evaluation Experiments

“Deletion” and “Insertion” (hard to draw slides, I’ll demonstrate it in person)



Saliency Map :

0.1

1	3	2
4	1	2
8	5	0

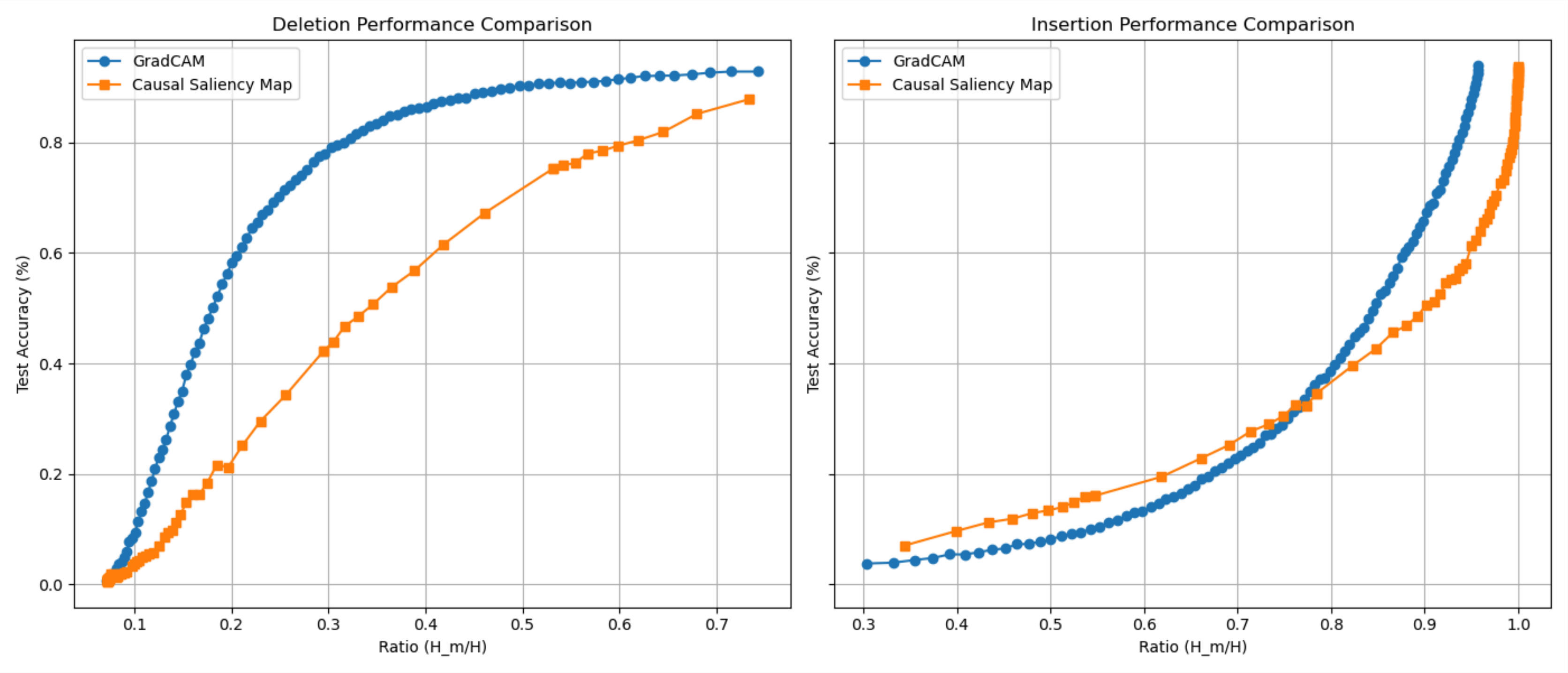
Deletion :

$$\begin{cases} (\text{saliency value})^{i.j} \geq \text{threshold} \Rightarrow \text{mask}^{i.j}_{-D} = 1 \\ (\text{saliency value})^{i.j} < \text{threshold} \Rightarrow \text{mask}^{i.j}_{-D} = 0 \end{cases}$$

Insertion :

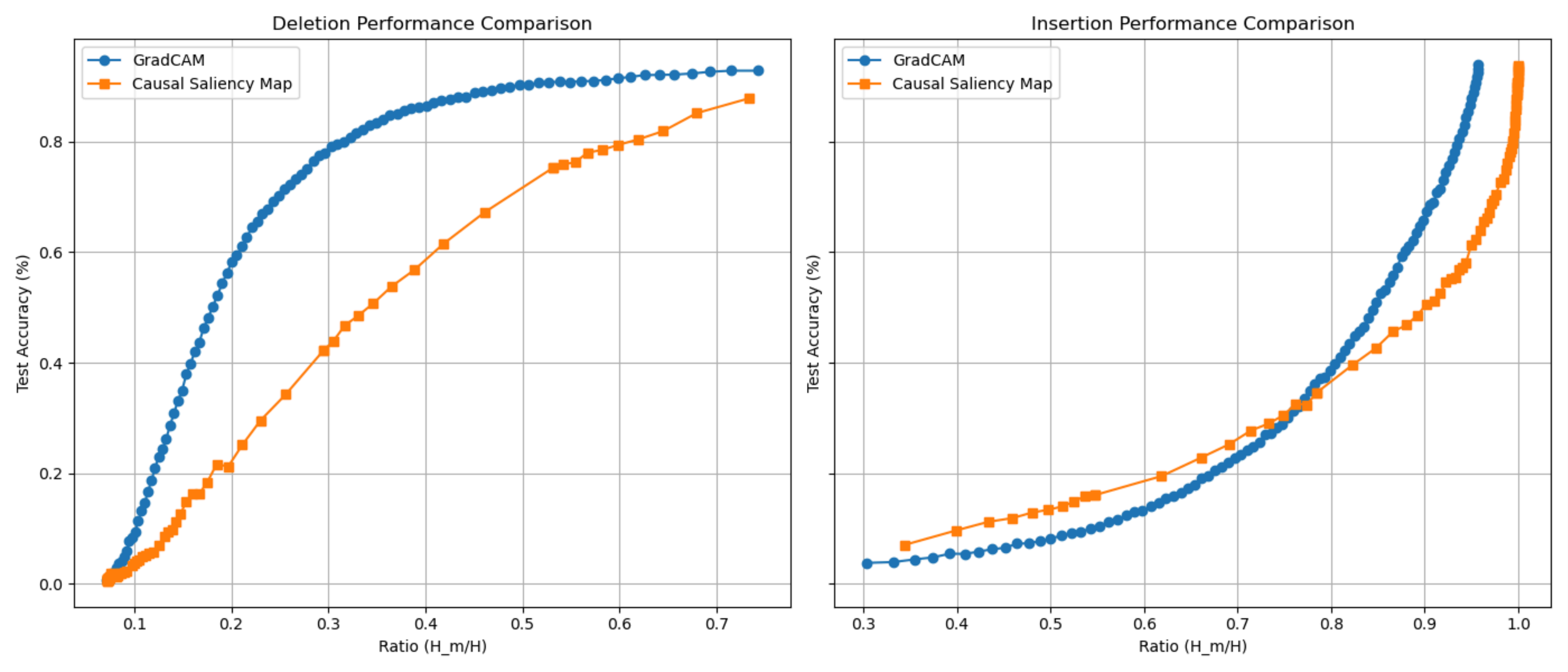
$$\begin{cases} (\text{s. v.})^{i.j} \geq \text{threshold} \Rightarrow \text{mask}^{i.j}_{-I} = 0 \\ (\text{s. v.})^{i.j} < \text{threshold} \Rightarrow \text{mask}^{i.j}_{-I} = \underline{1}. \end{cases}$$

2.2 Current Results



The Insertion performance is reasonable, while the Deletion experiments shows worse results. **Choosing the "threshold" is not fair to our methods**. Next: using proportion instead of numerical threshold.

2.2 Current Results

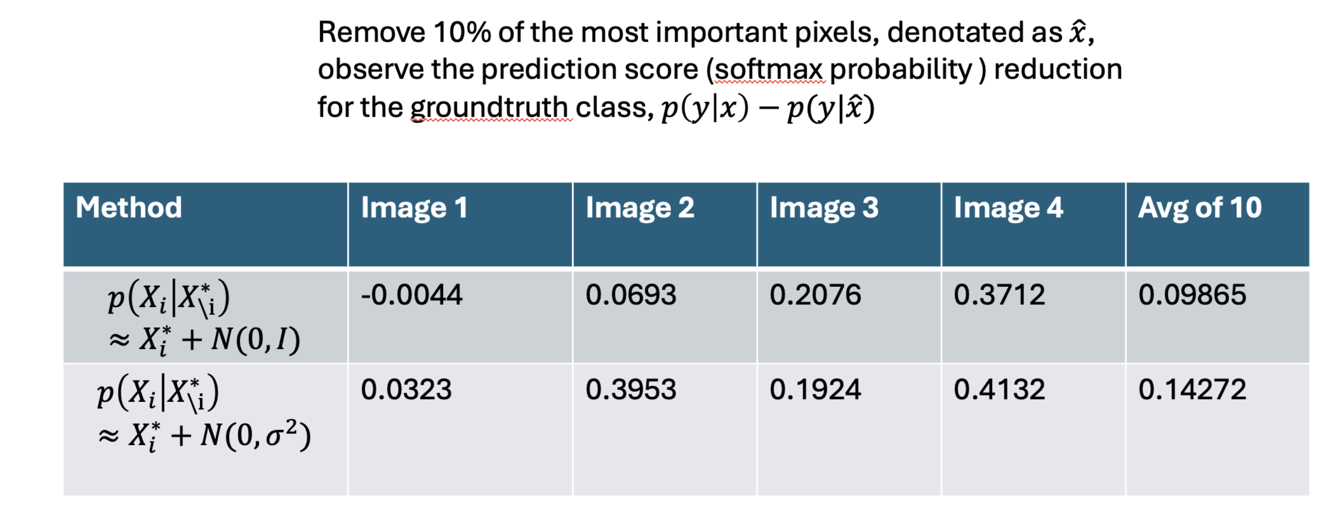


For GradCAM,I chose 100 points evenly between 0.01 and 1. But since our causal saliency map has a different scale, I tried from 0.00001 to 0.0001 then to 0.001 with different levels and draw the above plot. I think I need more experiments between 10^{-5} and 10^{-4}.

2.3 Imporved Estimation

- Previous, $p\left(x_i|x^*_{\setminus i}\right) \approx x^*_i + N(0,1)$
- Improved, $p\left(x_i|x^*_{\setminus i}\right) \approx x^*_i + N(0,\sigma^2)$, where σ^2 is the variance of x_i and its neighbors (total 9 pixels).

Generate 1,200 causal saliency map takes 12 hours, I have generated the improved causal saliency maps, and will do Deletion-Insertion experiments today.



Remove 10% of the most important pixels, denotated as $\hat{x}$, observe the prediction score (softmax probability) reduction for the groundtruth class, $p(y|x) - p(y|\hat{x})$

Method	Image 1	Image 2	Image 3	Image 4	Avg of 10
$p(X_i\|X^*_{\setminus \mathrm{i}}) \approx X^*_i + N(0,I)$	-0.0044	0.0693	0.2076	0.3712	0.09865
$p(X_i\|X^*_{\setminus \mathrm{i}}) \approx X^*_i + N(0,\sigma^2)$	0.0323	0.3953	0.1924	0.4132	0.14272

3. Experiments Attempts for CredalNets

3.1 What we discussed lask week

The main difference in training lower bound and upper bound is “selecting” the data. So we can lower bound and upper bound separately with different data.

A tentative pipline:

1. Train h_D on whole dataset D
2. Sort the training data loss of h_D, generate D_u with good data to train optimistic upper bound; generate D_l with bad data to train pessimistic lower bound.
3. Train h_u and h_l on the two generated dataset to get upper bound and lower bound.
4. During inference, jointly using h_u and h_l to get predictive interval.

3.2 Experimental Design

1. Get pretrained resnet-50 model from torch
2. Raw-tuning a h_D based the pretrained resnet-50 on CIFAR-10, for 5-10 epochs
3. Sort the training data loss of h_D, generate D_u with good data to train optimistic upper bound; generate D_l with bad data to train pessimistic lower bound.
4. Train h_u and h_l based on the raw-tuned h_D on the two generated dataset to get upper bound and lower bound.
5. During inference, jointly using h_u and h_l to get predictive interval.

Outcomes.

1. training h_l is conservative and stable, using only 15% bad data can also give good result (means only using this lower bound to do prediction)
2. Training h_u is highly unstable, if we just use small number of data D_u, the prediction will be too optimistic to be get only accuracy about 80% (expected: at least 90%).

3.3 Problems

- Training two separate model with individual softmax probability cannot meet the requirement of Credal Interval, so it is important to connect the generating of the two bounds, we can think about a more smart way.

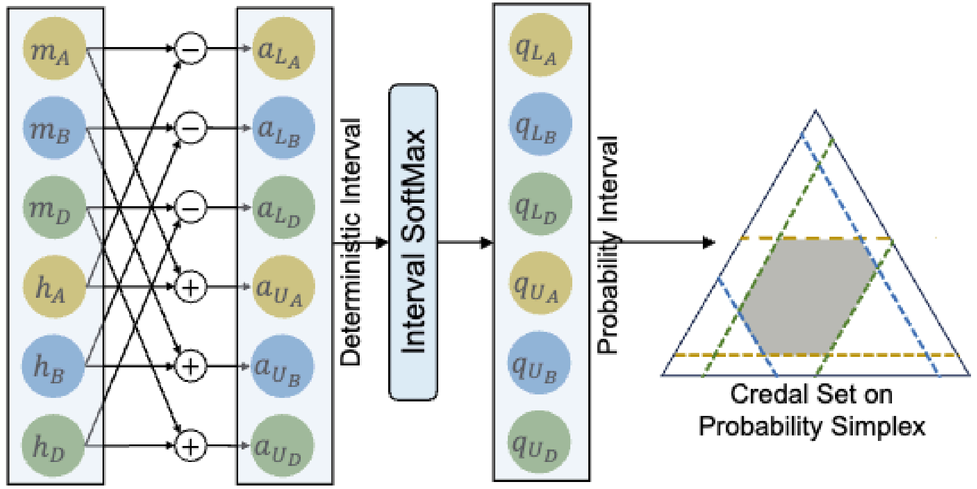


Figure 2: CreNet final layer structure for three classes.

3.3 Problems

1. If we can generate two groups of logits $a_{L_A}, a_{L_B}, a_{L_C}$ and $a_{U_A}, a_{U_B}, a_{U_C}$, and strictly make sure $a_{L_i} \leq a_{U_i}$, then we can use the proposed "Interval Softmax" to generate Credal Interval.
2. Or we can just combine the two prediction from two models directly, for every class, the larger predictive probability is the upper bound, and the smaller is the lower bound (this is similar with their paper "Cradal Wrapper", which was submitted to ICLR 2025, but they used same training policy among the models).

4. Plans for next week

4. Plans for next week

1. Do more experiments on Causal Saliency Map and find better evaluation methods.
2. Try to implement the proposed thoughts for PNC-predictor and do UQ (on some applications).
3. Think and Design experiments to train Credal Nets differently and effectively. (extensive reading and try experiments)