Weekly Study Report

Wang Ma

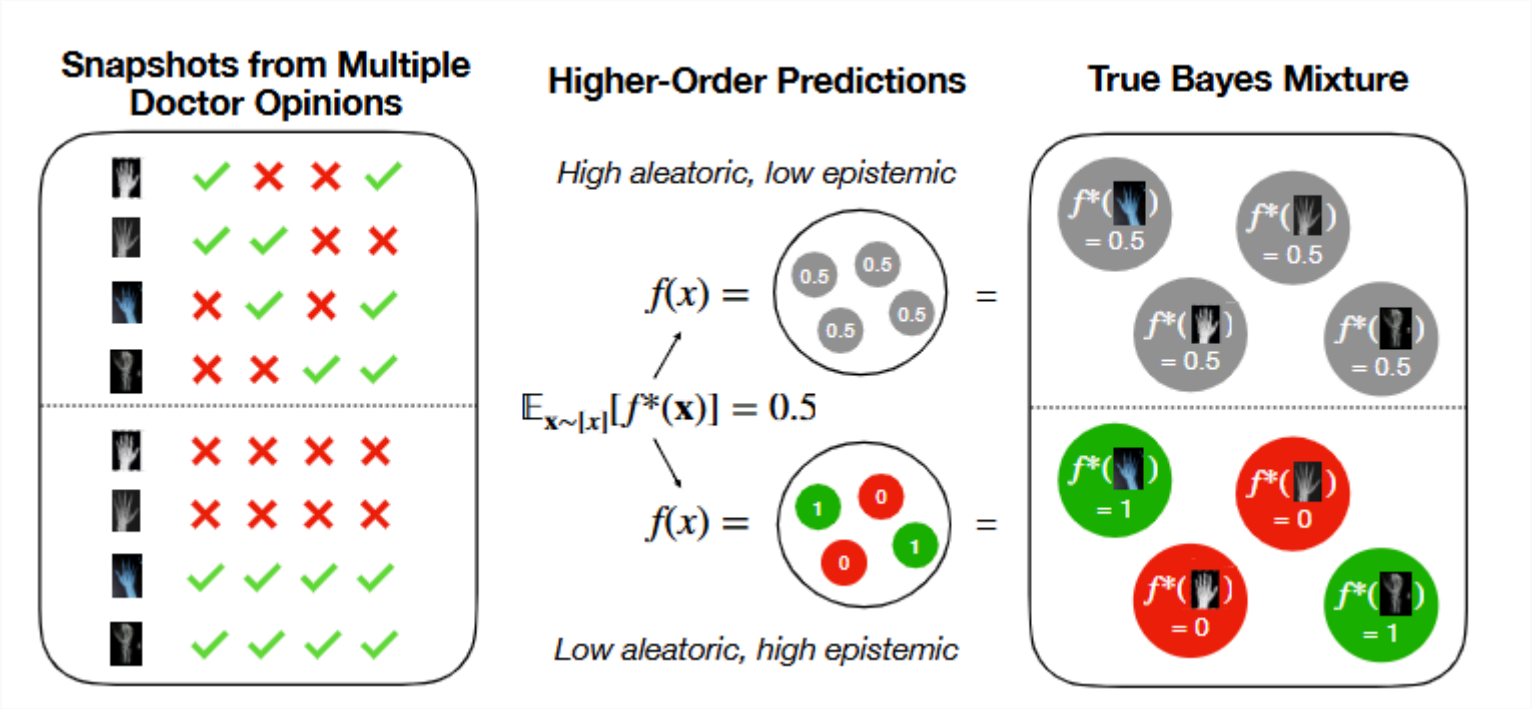
2025-04-15

Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute

Outline

1. [ICLR 2025] Provable Uncertainty Decomposition via Higher-Order Calibration

1. [ICLR 2025] Provable Uncertainty Decomposition via Higher-Order Calibration



- First-order prediction: $f(x) = 0.5$, ambiguous
- Higher-order prediction: $f(x) \sim (,)$, and some samples: $f(x) = 0, 1, 0, 1$

1.1 Calibration and Higher-Order Calibration

- Calibration: If a model predicts a group of instances with 70% probability, then the predicted accuracy on this group of instances should also be around 70%.
- Higher-Order Calibration:
  - First-order predictor: $f^* : X \to \Delta Y$, where ΔY can be a softmax probability.
  - Higher-order predictor: $f : X \to \Delta\Delta Y$, which is the distribution over the first-order prediction, for example, the distribution of a Softmax probability can be a Dirichlet Distribution.

  Let $[x] = \{x' \in X : f(x') = f(x)\}$, which is a level set of x, we say f is higher-order calibrated if for every x, $f(x) = f^*([x])$.

Example, a two-class classification problem, the first order prediction is bernoulli(p):

- Suppose $f(x_1) = 0.5 * \delta_{p=0.2} + 0.5 * \delta_{p=0.8}$, and $[x_1] = \{x_1, x_2\}$, where $f(x_1) = f(x_2)$.
- Then if we have $f(x) = f^*([x_1]) = \text{Average } (f^*(x_1) + f^*(x_2))$, then f is higher-order calibrated.

1.2 Basic Settings

1. In higher-order calibration, we only have the higher-order predictor $f : X \to \Delta\Delta Y$, and we do not have the first-order predictor $f^* : X \to \Delta Y$, which is considered as the ground-truth in the paper, that is, the empirical distribution of the label for x.
2. Uncertainty Quantification: after we get the distribution of the prediction:

$$\mathbb{H}\Big(\mathbb{E}_{p\sim f(x)}[x]\Big) = \mathbb{E}_{p\sim f(x)}(\mathbb{H}(p)) + \Big[\mathbb{H}\Big(\mathbb{E}_{p\sim f(x)}[x]\Big) - \mathbb{E}_{p\sim f(x)}(\mathbb{H}(p))\Big]$$

So, this paper actually proposed a way to train the higher-order predictor based on the higher-order calibration view with theoretical guarantees.

Theoretical Guarantees:

- Under higher-order calibration, the estimated aleatoric uncertainty equals to the average of the real aleatoric uncertainty. (then the EU also is real based on the difference.) And the decomposition has real semantic meanings and are more reliable.

1.3 kth-Order Calibration for Training

- k-snapshot: It is impossible to know the exactly ground-truth $f^*(x)$, then the author propose k-snapshot, that is, for every x, we get k independent labels, and construct the empirical distribution on x to approximate $f^*(x)$.

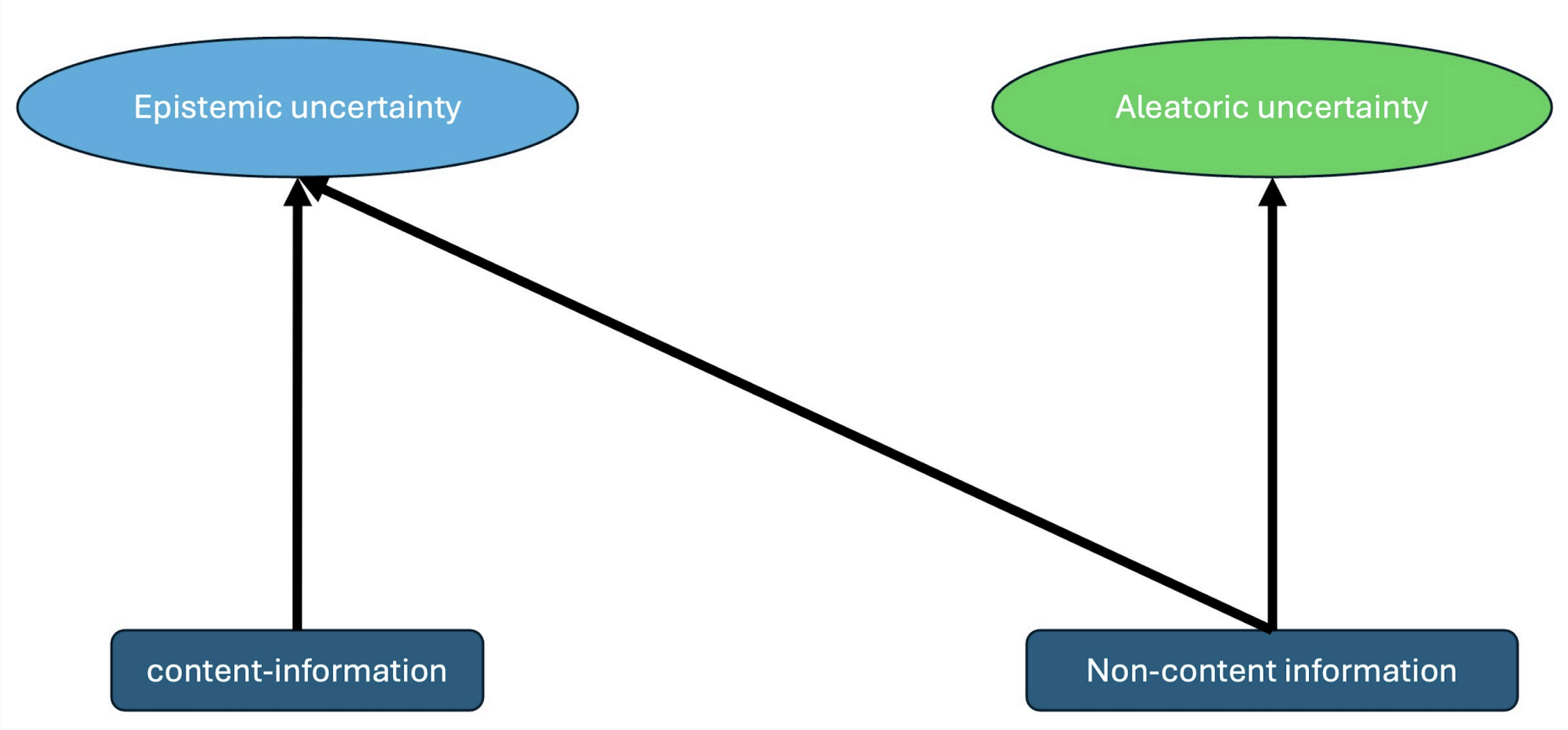
$$\Pr[y] = \frac{1}{k} \sum_{i=1}^{k} \mathbb{1}[y_i = y]$$

- k-projection: We draw k labels(first-order prediction) $\hat{y}_1, \hat{y}_2 ..., \hat{y}_k$ from the higher-order prediction $f(x)$. And here $Y_k = [\hat{y}_1, \hat{y}_2 ..., \hat{y}_k]$

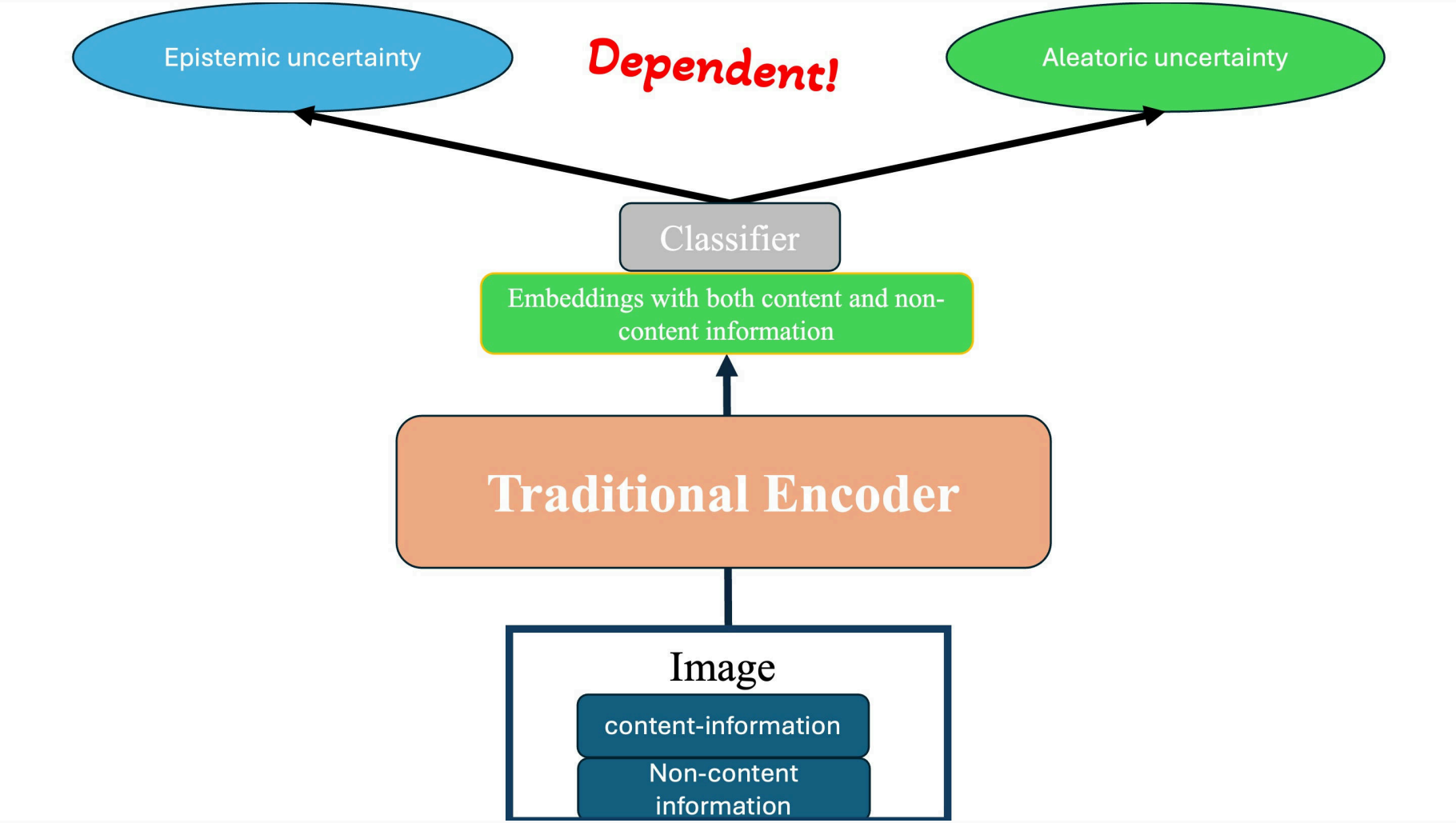
During the training, we measure the distance between the snapshot and projection, possible metrics are Wasserstein Distance, KL Divergence and MSE loss. As $k \to \infty$, the model approaches true higher-order calibration.

Goal of the Training: Let the output higher-order prediction $f(x)$'s projection matches the real snapshot.

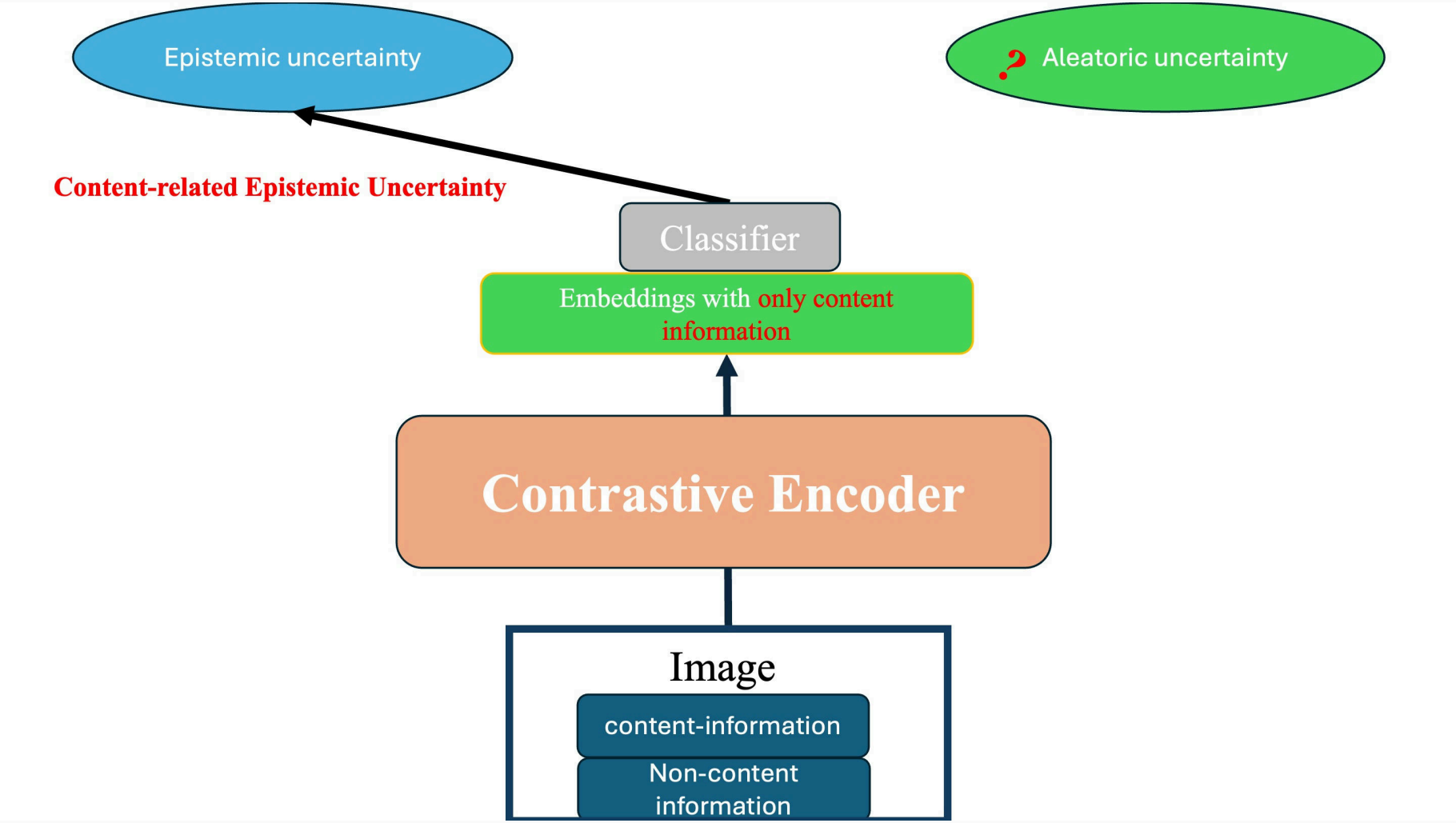
2. [Updated Results] Contrastive Learning to get content-related Epistemic Uncertainty


Epistemic uncertainty
Aleatoric uncertainty
content-information
Non-content information

2.1 Background



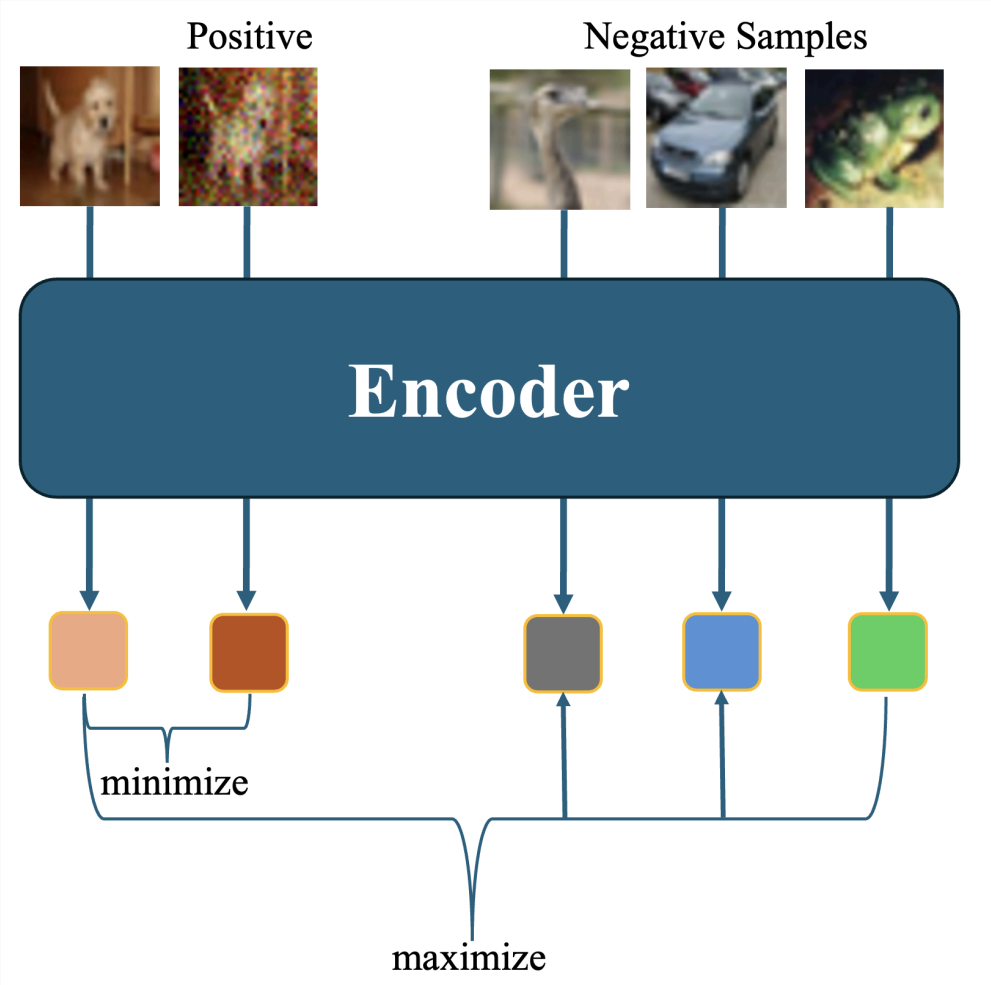
2.1 Background



GOAL.

1. Training a contrastive encoder to learn consistent features/embeddings from high- and low- quality input.
2. We want to minimize the influence of non-content information to the epistemic uncertainty.
3. The final goal is to obtain True content-related Epistemic Uncertainty, which can be used to detect in-lier data when both AU and EU are high.

2.3 The Contrastive Learning Model (Encoder Learning)



- Anchor: the clean (high-quality) image
- Positive Samples: the corrupted image
- Negative Samples: other images in batch

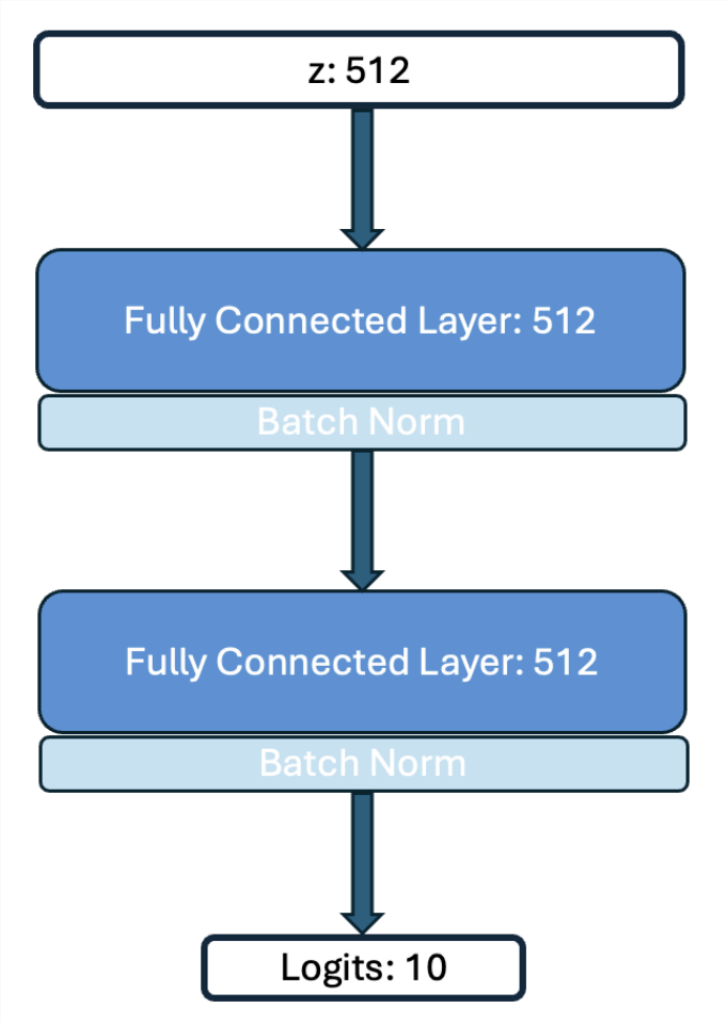
Output of the Encoder: the embedding z

The loss function:

$$L_i = -\log \frac{\exp(z_i z_i^+)}{\sum_{k,k\in z_i^-} \exp(z_i z_k)},$$

where $\|z\|^2 = 1$, so $z_i z_i^j$ is the cosine similarity of the two embeddings.

2.4 The MLP Classifier Model



During the training of the Classifier head, we freeze the encoder part and solely update the parameters in the MLP head.

This training follows a standard Classification task training with cross entropy loss.

2.5 Design of the Experiments

- Training.
  1. Train a contrastive encoder, where the positive samples are corrupted images. (Original contrastive learning uses augmentations as the positive samples, and learns consistent embeddings from the augmentations.)
  2. Add a classification head to the encoder to get prediction results and do uncertainty quantification. (Here we train ensemble models of size 10)
- Testing. We want to test on
  1. Original Clean high-quality images (get uncertainty results)
  2. Generated low-quality images
     - Corrupted images with same corruption types and severity as training (get uncertainty results)
     - Corrupted images with different corruption types
  3. OOD data (SVHN)

2.6 Expectation of the Test Results

1. Original clean high-quality images

   Low uncertainties
2. Generated low-quality images
   - Corrupted images with same corruption types and severity as training. (Severity: 2)

     Low uncertainties as the clean images, hard to use uncertainty to separate from clean images (they are not ood, so can not be separated)
   - **Corrupted images with different corruption types**. (Severity: 4)

     Low uncertainties as before, hard to be separated. Our goal is that the contrastive encoder can learn consistent embeddings from both high- and low- quality data, the experimental results on this setting is the most important.
   - OOD data (SVHN)

     High uncertainty, can be easily separated.

2.7 Current Results

2.7.1 Uncertainty Quantification Performance

Contrastive Leared Encoder_pretrained		Clean_id	Corrupted_trained	Corrupted_not_trained	OOD
Test Accuracy		0.9072	0.8811	0.6688	\
Total Uncertainty	mean	0.3158	0.3655	0.6708	1.1917
	std	0.4399	0.4582	0.5583	0.4603
Aleatoric Uncertainty	mean	0.2498	0.286	0.4981	0.9158
	std	0.3424	0.3621	0.4222	0.3596
Epistemic Uncertainty	mean	0.066	0.0795	0.1727	0.2759
	std	0.0998	0.1108	0.1696	0.1481

ResNet18 Results_pretrained		Clean_id	Corrupted_trained	Corrupted_not_trained	OOD
Test Accuracy		0.9582	0.9165	0.7835	\
Total Uncertainty	mean	0.1223	0.2636	0.6828	1.43
	std	0.2607	0.4162	0.6076	0.4464
Aleatoric Uncertainty	mean	0.0824	0.2035	0.5691	1.2731
	std	0.1752	0.1012	0.5234	0.4233
Epistemic Uncertainty	mean	0.0399	0.0601	0.1138	0.1569
	std	0.0933	0.1012	0.1126	0.0754

2.7 Current Results

Contrastive Leared Encoder_not_pretrained		Clean_id	Corrupted_trained	Corrupted_not_trained	OOD
Test Accuracy		0.8863	0.8635	0.6353	\
Total Uncertainty	mean	0.3667	0.4197	0.6592	1.1948
	std	0.4528	0.4756	0.5338	0.4475
Aleatoric Uncertainty	mean	0.2941	0.3335	0.4956	0.9287
	std	0.3661	0.3821	0.411	0.3492
Epistemic Uncertainty	mean	0.0726	0.0862	0.1636	0.2661
	std	0.0998	0.1097	0.158	0.146

ResNet18 Results_not_pretrained		Clean_id	Corrupted_trained	Corrupted_not_trained	OOD
Test Accuracy		0.8784	0.8224	0.6977	\
Total Uncertainty	mean	0.3766	0.5853	0.984	1.429
	std	0.4611	0.5655	0.6472	0.4831
Aleatoric Uncertainty	mean	0.3132	0.509	0.8692	1.3167
	std	0.3866	0.5034	0.5852	0.4625
Epistemic Uncertainty	mean	0.0634	0.0763	0.1148	0.1123
	std	0.0851	0.082	0.0904	0.0558

2.7 Current Results

1. The Contrastive Learned Encoder seems to disentangle AU and EU from the strong Linear Relationship.
   - AU increases as the corruption severity increases
   - EU is consistent for Clean Data and Corrupted_id data
2. For resnet-18, the measured AU and EU are in strong linear relationship. And it seems the model fails to identify SVHN (?)

2.7.2 Detecting OOD and Low-quality Data

Contrastive Learned Encoder_pretrained		Corrupted_trained	Corrupted_not_trained	OOD
Total Uncertainty	AUROC	0.534	0.699	0.9051
	AUPR	0.5294	0.6939	0.9527
Aleatoric Uncertainty	AUROC	0.5327	0.6887	0.8988
	AUPR	0.5269	0.6691	0.9465
Epistemic Uncertainty	AUROC	0.5367	0.7119	0.8934
	AUPR	0.5347	0.7242	0.9464

Resnet18 Classifier_Pretrained		Corrupted_trained	Corrupted_not_trained	OOD
Total Uncertainty	AUROC	0.63	0.8237	**0.9846**
	AUPR	0.5945	0.6647	**0.9346**
Aleatoric Uncertainty	AUROC	0.634	0.8322	**0.9903**
	AUPR	0.6003	0.6665	**0.937**
Epistemic Uncertainty	AUROC	0.6142	0.7778	**0.8888**
	AUPR	0.568	0.646	**0.8963**

2.7 Current Results

Contrastive Learned Encoder_not_pretrained		Corrupted_trained	Corrupted_not_trained	OOD
Total Uncertainty	AUROC	0.5361	0.6718	0.8901
	AUPR	0.529	0.6532	0.9434
Aleatoric Uncertainty	AUROC	0.5346	0.6593	0.8827
	AUPR	0.5265	0.6277	0.9347
Epistemic Uncertainty	AUROC	0.5391	0.6906	0.8758
	AUPR	0.5355	0.7017	0.9389

Resnet18 Classifier_not_Pretrained		Corrupted_trained	Corrupted_not_trained	OOD
Total Uncertainty	AUROC	0.6183	0.7752	**0.9252**
	AUPR	0.5861	0.653	**0.9105**
Aleatoric Uncertainty	AUROC	0.623	0.7816	**0.9355**
	AUPR	0.5918	0.6554	**0.9147**
Epistemic Uncertainty	AUROC	0.5793	0.697	**0.7433**
	AUPR	0.5393	0.6121	**0.8229**

2.7 Current Results

Contrastive Learned Encoder_pretrained		Corrupted_id vs Corrupted_ood	Corrupted_id vs SVHN (OOD)	Corrupted_ood vs SVHN(OOD)
Total Uncertainty	AUROC	0.6688	0.8866	0.7592
	AUPR	0.661	0.9425	0.8652
Aleatoric Uncertainty	AUROC	0.659	0.8811	0.7702
	AUPR	0.6382	0.9365	0.8725
Epistemic Uncertainty	AUROC	0.6798	0.8689	0.6998
	AUPR	0.6891	0.9319	0.812

Resnet 18 _pretrained		Corrupted_id vs Corrupted_ood	Corrupted_id vs SVHN (OOD)	Corrupted_ood vs SVHN(OOD)
Total Uncertainty	AUROC	0.7241	0.9521	0.8263
	AUPR	0.7235	0.9767	0.9054
Aleatoric Uncertainty	AUROC	0.73	0.9599	0.8402
	AUPR	0.7318	0.9809	0.917
Epistemic Uncertainty	AUROC	0.685	0.8248	0.658
	AUPR	0.631	0.8573	0.7634

2.7 Current Results

Contrastive Learned Encoder_not_pretrained		Corrupted_id vs Corrupted_ood	Corrupted_id vs SVHN (OOD)	Corrupted_ood vs SVHN(OOD)
Total Uncertainty	AUROC	0.6388	0.8709	0.7742
	AUPR	0.6201	0.9325	0.877
Aleatoric Uncertainty	AUROC	0.6268	0.864	0.7844
	AUPR	0.5971	0.9236	0.88
Epistemic Uncertainty	AUROC	0.6563	0.8503	0.7031
	AUPR	0.6653	0.9238	0.8207

Resnet 18 _not_pretrained		Corrupted_id vs Corrupted_ood	Corrupted_id vs SVHN (OOD)	Corrupted_ood vs SVHN(OOD)
Total Uncertainty	AUROC	0.6767	0.8586	0.7008
	AUPR	0.682	0.9263	0.818
Aleatoric Uncertainty	AUROC	0.6776	0.8681	0.7201
	AUPR	0.6782	0.9357	0.8433
Epistemic Uncertainty	AUROC	0.6831	0.6831	0.5228
	AUPR	0.776	0.776	0.6862

2.7 Current Results

Conclusion:

1. The Contrastive Learned Encoder is consistent to Corrupted_id data. And for Corrupted_ood data, the AUROC is low meaning the model does not really see difference in highly-corrupted data.
2. Meanwhile, the model can identify true OOD data well, even the test accuracy is not good. But it seems the EU does not work well on SVHN detection.
3. The Resnet-18 almost failed on the OOD detection, which means it cannot capture significant features. This means I need to check the model training and retrain the resnet18 models.
4. The Contrastive Learned Encoder has the ability to identify corrupted_ood from SVHN, that is, the low-quality version from the true OOD. But Resnet-18 failed to do this, especially for the un-pretrained resnet 18.

2.8 Uncertainty-enhanced Design