Rethinking Epistemic and Aleatoric Uncertainty for Active Open-Set Annotation: An Energy-Based Approach

Supplementary Material

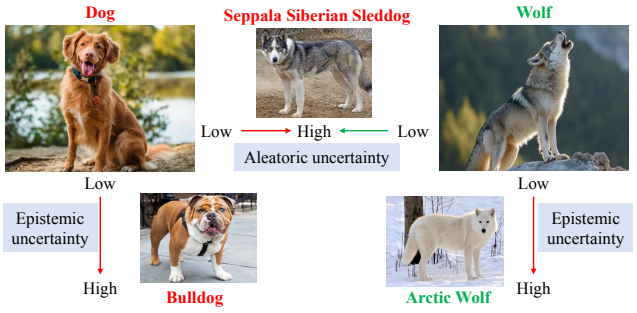

Figure 7. Intuitive examples of aleatoric and epistemic uncertainty in dog-wolf binary classification.

A. Uncertainty Quantification

In deep learning, epistemic uncertainty and aleatoric uncertainty represent two distinct types of uncertainty, commonly used to describe the various sources of uncertainty in a model's predictions:

- **Epistemic uncertainty:**
  - This type of uncertainty arises from a model's lack of knowledge, often due to insufficient training data or the model's complexity. It reflects the model's incomplete or uncertain understanding of the task and can generally be reduced or eliminated with more data or a more effective model.
  - For example, in a deep neural network, if there is little data available for certain classes or the model has not been trained sufficiently, the model may be highly uncertain in its predictions for certain examples.
  - This uncertainty is generally reducible, as it can be mitigated by adding more training data or improving the model architecture.
  - In Figure 7, the "Bulldog" and "Arctic Wolf" exhibit significant feature differences from the "Dog" and "Wolf" in the training set, leading to higher epistemic uncertainty. After these examples are incorporated into model training, predictive performance on them improves, thereby reducing their epistemic uncertainty.
- **Aleatoric uncertainty:**
  - This type of uncertainty stems from inherent noise or variability in the data, i.e., the intrinsic randomness or uncontrollable factors within the data.
  - For instance, in image classification, factors like feature confusion, lighting conditions, or object occlusion may lead to instability in the model's predictions.
  - This uncertainty is generally irreducible because it originates from the intrinsic properties of the data, not from issues with the model or training process.
  - In Figure 7, the "Seppala Siberian Sleddog" resembles the "Wolf" in appearance but belongs to the "Dog" class, leading to higher aleatoric uncertainty. Due to feature confusion, incorporating these examples into model training may not substantially improve performance or reduce their aleatoric uncertainty.

B. Label-Wise Free Energy

EBMs define the probability distribution in multi-label settings through the logits as:

$$\begin{aligned} p\left(y_c|x\right) &= \frac{e^{-E(x,y_c)}}{\int_y e^{-E(x,y_c)}} = \frac{e^{-E(x,y_c)}}{e^{-E(x,y_c)}+e^{-E(x,-y_c)}} \\ &= \frac{e^{-E(x,y_c)+E(x,-y_c)}}{1+e^{-E(x,y_c)+E(x,-y_c)}} = \frac{e^{f_{y_c}(x)}}{1+e^{f_{y_c}(x)}} \\ &= \frac{e^{-E(x,y_c)}}{e^{-E(x)}} \end{aligned} \tag{15}$$

where $y_c = 1$ indicates that instance x belongs to the c-th class while $y_c = -1$ indicates not, $f_{y_c}(x)$ denotes predicted logit of the model f for instance x regarding the c-th class, and $E_{y_c}(x) = -\log\left(1+e^{f_{y_c}(x)}\right)$ is the label-wise free energy for instance x on class y_c.

C. The Pseudocode of EAOA

The pseudocode of EAOA is summarized in Algorithm 1.

D. Comparing Methods

We consider the following AL methods as baselines:

- Random, which selects instances at random;
- Uncertainty, which selects instances with the highest entropy of predictions;
- Certainty, which selects instances with the lowest entropy of predictions;
- Coreset, which uses the concept of core-set selection to choose diverse instances;
- BADGE, which selects instances by considering both uncertainty and diversity in the gradient via k-means++ clustering;
- CCAL, which employs contrastive learning to extract the semantic and distinctive scores of examples for instance querying;
- MQNet, which balances the purity score and informativeness score to select instances through meta-learning;

Algorithm 1 The EAOA algorithm

Input: Labeled data pool $\mathcal{D}_L = \mathcal{D}_L^{kno} \cup \mathcal{D}_L^{unk}$, unlabeled data pool $\mathcal{D}_U$, detector f_{θ_D}, target classifier f_{θ_C}, query budget b, dynamic factor k_t, and target precision tP.

Process: (The t-th AL round)

 # Detector training
Update θ_D by minimizing $\mathcal{L}_{detector}$ in Eq. (13) using all labeled examples from $\mathcal{D}_L$.
 # Epistemic uncertainty estimating
Extract logit outputs and features from f_{θ_D} for examples in $\mathcal{D}_L$ and $\mathcal{D}_U$, respectively.
Based on model outputs, estimate the learning-based epistemic uncertainty score for each example in $\mathcal{D}_U$ using Eq. (1) and Remark 1.
Based on feature similarity, find K-nearest neighbors in $\mathcal{D}_U$ for each example in $\mathcal{D}_L$, and obtain reverse neighbors by class in $\mathcal{D}_L$ for each example in $\mathcal{D}_U$.
Estimate data-centric epistemic uncertainty score for each example in $\mathcal{D}_U$ using Eq. (7) and Remark 1.
For each example, combine the two scores into one final epistemic uncertainty score using GMM and Eq. (8).
 # Target classifier training
Update θ_C by minimizing $\mathcal{L}_{classifier}$ in Eq. (14) using all known class labeled examples from $\mathcal{D}_L^{kno}$.
 # Aleatoric uncertainty estimating
Extract logit outputs from f_{θ_C} for examples in $\mathcal{D}_U$.
Estimate aleatoric uncertainty score for each example in $\mathcal{D}_U$ using Remark 2.
 # Active sampling
$k_t b$ examples with the lowest epistemic uncertainty scores are selected first to form a candidate query set.
b examples with the highest aleatoric uncertainty scores are then queried to form the final query set X^{query}.
 # Oracle labeling
Query labels from Oracle and obtain X_{kno}^{query}, X_{unk}^{query}, and query precision $rP = \left| \frac{X_{kno}^{query}}{X^{query}} \right|$.
Update k_t to k_{t+1} using Eq. (10) based on $tP - rP$.
Update corresponding data pools: $\mathcal{D}_U = \mathcal{D}_U - X^{query}$, $\mathcal{D}_L^{kno} = \mathcal{D}_L^{kno} \cup X_{kno}^{query}$, and $\mathcal{D}_L^{unk} = \mathcal{D}_L^{unk} \cup X_{unk}^{query}$.

Output: $\mathcal{D}_L$, $\mathcal{D}_U$, θ_D, θ_C, and k_{t+1} for next round.

- LfOSA, which selects instances based on the maximum activation value produced by the $(C+1)$-class detector;
- EOAL, which queries instances by calculating the entropy of examples in both known and unknown classes;
- BUAL, which queries instances by adaptively combining the uncertainty obtained from positive and negative classifiers trained in different ways.

Among these methods, EOAL and BUAL are currently state-of-the-art.

E. Additional Ablation Studies

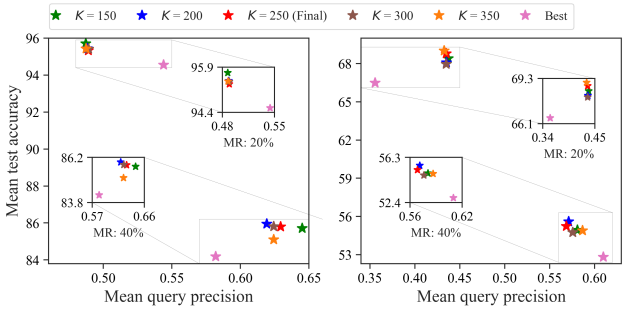

Figure 8. Ablation results for K in reverse k-NN on CIFAR-10 (**Left**) and CIFAR-100 (**Right**). "MR" denotes mismatch ratio. "Best" indicates the top-performing method in the comparisons.

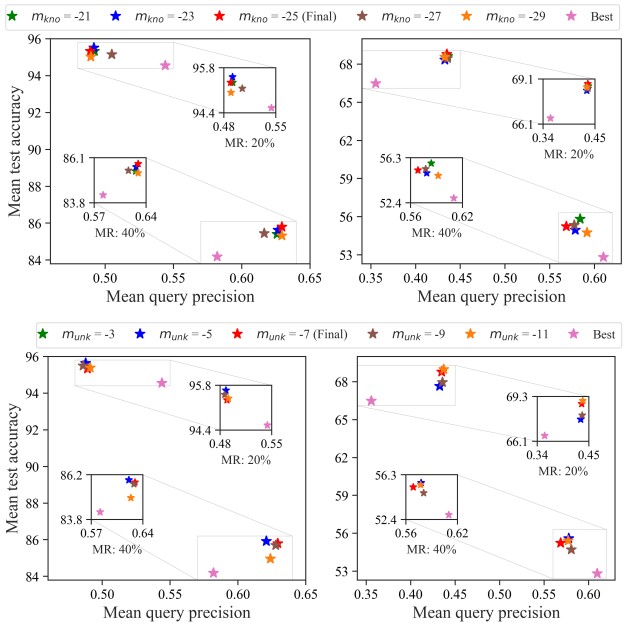

Figure 9. Ablation results for m_{kno} and m_{unk} in margin-based energy loss on CIFAR-10 (**Left**) and CIFAR-100 (**Right**).

Figure 8 illustrates the effect of the hyperparameter K in reverse k-NN on EAOA's performance, with values set to [150, 200, 250, 300, 350]. Figure 9 presents the influence of the known class margin m_{kno} and the unknown class margin m_{unk} in margin-based energy loss $\mathcal{L}_{energy}$ on EAOA's performance, with values set to [-29, -27, -25, -23, -21] for m_{kno} and [-11, -9, -7, -5, -3] for m_{unk}. While the optimal value of K, m_{kno}, and m_{unk} differ across different settings, their overall performance remains relatively stable compared to the top-performing method in the comparisons, with $K = 250$, $m_{kno} = -25$, and $m_{unk} = -7$ consistently achieving strong results.

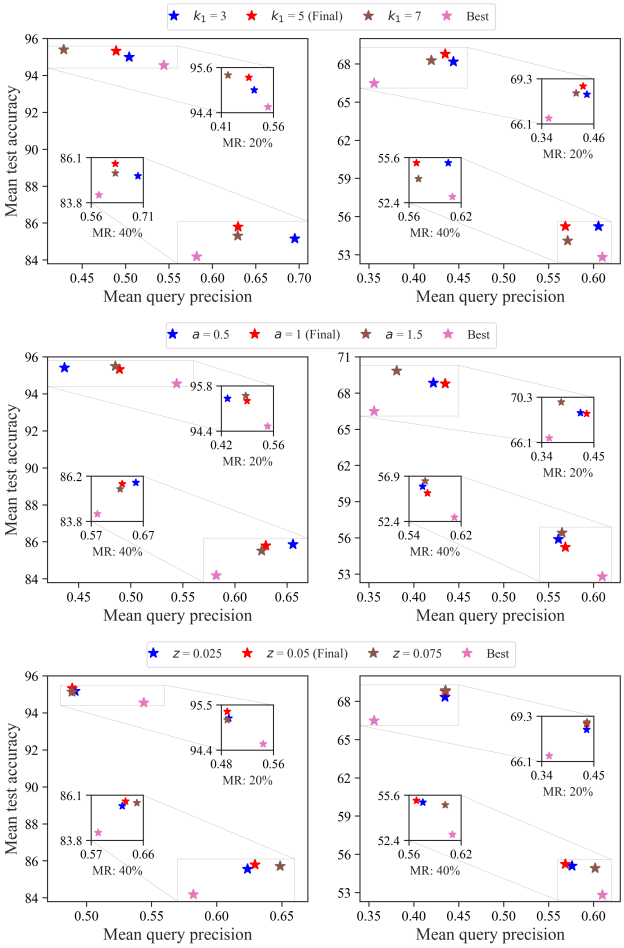

Figure 10. Ablation results for k_1, a and z in target-driven adaptive sampling strategy on CIFAR-10 (**Left**) and CIFAR-100 (**Right**).

Figure 10 shows the impact of initial round k_1, variation amplitude a, and triggering threshold z in Eq. 10 on EAOA's performance, with values set to [-3, -5, -7] for k_1, [0.5, 1, 1.5] for a, and [0.025, 0.05, 0.075] for z. An excessively large k_1 value may lead to initial rounds that prioritize aleatoric uncertainty, beneficial for lower mismatch ratios. Conversely, a small k_1 value emphasizes epistemic uncertainty, making it suitable for higher mismatch ratios. Here, $k_1 = 5$ consistently delivers strong performance across various datasets. In practical applications, prior knowledge about the dataset can be used to further adjust its value. For hyperparameters a and z, their values are simply set to 1 and 0.05 (ensuring no adjustments are triggered when the difference between target and actual query precision is within ± 0.05, or a range of 0.1) respectively to simplify parameter tuning. Although the parameter selection here is intuitive, the results in Figure 10 confirm its suitability.