Evidential Transformation Network: Turning Pretrained Models into Evidential Models for Post-hoc Uncertainty Estimation

Supplementary Material

1. Limitations

While ETN improves the uncertainty estimation performance of pretrained models without harming accuracy and with only minimal additional computational cost, it also has several limitations.

First, the benefits of ETN are largely empirical rather than theoretical. Recent works have raised concerns about EDL from a theoretical standpoint, arguing that current training procedures do not guarantee a faithful modeling of epistemic uncertainty [2, 12, 23]. Our observation that simple scalar scaling is often sufficient to make logits suitable as Dirichlet parameters may reflect inherent limitations in existing EDL training formulations. However, we do not think that this empirical success should be underestimated, as robust and consistent improvements across diverse datasets and architectures are precisely what is required for practical deployment of uncertainty-aware pretrained models, even in the absence of a complete theoretical account.

Second, our method requires access to the logits and the last hidden representation of the pretrained model, which may not be available when using closed-source models exposed only through an API (e.g., recent GPT models). Nevertheless, since ETN depends solely on these two quantities—unlike many uncertainty estimation baselines that require access to the full model architecture or gradients—it remains relatively compatible with *gray-box models* [1, 10].

2. Proofs and Derivations

In this section, we analyze the behavior of logits produced by models trained with cross-entropy and EDL losses. We first define the softmax per-sample (x, y) cross-entropy loss as:

$$\mathcal{L}_{\text{CE}}(\boldsymbol{z}, y) \;=\; -\log \frac{e^{z_y}}{\sum_{j=1}^{C} e^{z_j}} = \log\Big(1 + \sum_{j \neq y} e^{\,z_j - z_y}\Big)$$

Then we define the inter-class margin of an sample as:

$$\gamma(\boldsymbol{z}, y) = z_y - \max_{j \neq y} z_j$$

Given these definitions, we now present two lemmas characterizing the relationship between cross-entropy and EDL models.

Lemma 1 (Zero loss implies infinite margin)**.** *Cross-entropy loss becomes zero if and only if the margin between the logits of the label and other logits become infinite. i,e:*

$$\mathcal{L}_{CE}(\boldsymbol{z}, y) \to 0 \quad \Longleftrightarrow \quad \gamma(\boldsymbol{z}, y) \to \infty.$$

Proof. Suppose $\mathcal{L}_{\text{CE}}(\boldsymbol{z}, y) \leq \varepsilon$. Then the softmax probability of the correct class satisfies

$$\frac{e^{z_y}}{\sum_j e^{z_j}} \;\geq\; e^{-\varepsilon}.$$

Rearranging gives

$$\sum_{j \neq y} e^{z_j} \;\leq\; e^{z_y}(e^{\varepsilon} - 1).$$

Hence, for each $j \neq y$,

$$z_y - z_j \;\geq\; -\log(e^{\varepsilon} - 1).$$

Since $-\log(e^{\varepsilon} - 1) \to \infty$ as $\varepsilon \to 0$, the margin diverges. Conversely, if $\gamma(\boldsymbol{z}, y) \to \infty$, then $z_j - z_y \to -\infty$ for each $j \neq y$, so $e^{z_j - z_y} \to 0$. Therefore

$$\mathcal{L}_{\text{CE}}(\boldsymbol{z}, y) = \log\Big(1 + \sum_{j \neq y} e^{z_j - z_y}\Big) \to 0$$

□

Lemma 2 (Margin of EDL models)**.** *For a sample (x, y), assume there exists η with $0 \leq \eta < \nu - b_y$ such that*

$$\alpha_y \;\geq\; \nu - \eta, \qquad \alpha_j \;\leq\; b_j + \eta \quad \forall j \neq y.$$

Then the inter-class margin of an sample (x, y) of EDL models is defined by:

$$\gamma_{\text{EDL}}(\boldsymbol{z}, y) = f^{-1}(\nu - b_y - \eta) \;-\; f^{-1}(\eta) \qquad (1)$$

Proof. From $\alpha_y \geq \nu - \eta$ we get

$$f(z_y) = \alpha_y - b_y \;\geq\; \nu - b_y - \eta$$
$$\Longrightarrow z_y \geq f^{-1}(\nu - b_y - \eta).$$

For any $j \neq y$, the assumption $\alpha_j \leq b_j + \eta$ gives

$$f(z_j) = \alpha_j - b_j \;\leq\; \eta \quad \Longrightarrow \quad z_j \;\leq\; f^{-1}(\eta).$$

Taking the maximum over $j \neq y$ yields $\max_{j \neq y} z_j \leq f^{-1}(\eta)$, hence

$$\gamma_{\text{EDL}}(\boldsymbol{z}, y) = z_y - \max_{j \neq y} z_j \;\geq\; f^{-1}(\nu - b_y - \eta) - f^{-1}(\eta),$$

which is Equation 1. □

2.1. Proof to Proposition 1

By Lemma 1, zero CE loss is achieved by sending the margins $\gamma(\boldsymbol{z}, y) \to \infty$, which can be done by either pushing the correct logit up or the incorrect logits down. Given this, we provide two explicit cases of logits that both show vanishing cross-entropy loss but lead to bounded and diverging values of α_0, respectively.

Bounded $\tilde{\boldsymbol{\alpha}}_0$. Set $\tilde{z}_y = 0$ and $\tilde{z}_{j\neq y} = -t$. Then $\mathcal{L}_{\mathrm{CE}}(\tilde{\boldsymbol{z}}, y) = \log(1 + (C-1)e^{-t}) \to 0$. Therefore,

$$\begin{aligned}\tilde{\alpha}_0 &= \big(f(0) + b\big) + \sum_{j\neq y} \big(f(-t) + b\big) \\ &\to \big(f(0) + b\big) + (C-1)b \;<\; \infty,\end{aligned} \tag{2}$$

as $t \to \infty$ since $f(-t) \to 0$.

Diverging $\hat{\boldsymbol{\alpha}}_0$. Set $\hat{z}_y = t$ and $\hat{z}_{j\neq y} = 0$. Then $\mathcal{L}_{\mathrm{CE}}(\hat{\boldsymbol{z}}, y) = \log(1 + (C-1)e^{-t}) \to 0$, and

$$\alpha_0(\hat{\boldsymbol{z}}) = \big(f(t) + b\big) + \sum_{j\neq y} \big(f(0) + b\big) \;\to \infty, \tag{3}$$

as $t \to \infty$ since $f(t) \to \infty$.

2.2. Proof of Theorem 1

For a sample (x, y), assume $L := \mathcal{L}_{\mathrm{CE}}(\boldsymbol{z}, y) = \mathcal{L}_{\mathrm{EDL}}(\boldsymbol{z}, y)$. Since $z_j - z_y \leq -\gamma_{\mathrm{CE}}(\boldsymbol{z}, y)$ for all $j \neq y$, we obtain an upper bound on the CE loss:

$$\begin{aligned}L &= \log\left(1 + \sum_{j\neq y} e^{z_j - z_y}\right) \\ &\leq \log\Big(1 + (C-1)\, e^{-\gamma_{\mathrm{CE}}(\boldsymbol{z}, y)}\Big)\,.\end{aligned} \tag{4}$$

Let the lower bound on the EDL margin be

$$\gamma_{\mathrm{LB}} := f^{-1}(\nu - b_y - \eta) \;-\; f^{-1}(\eta).$$

Assume further that L satisfies

$$L \geq \log\big(1 + (C-1)e^{-\gamma_{\mathrm{LB}}}\big)\,. \tag{5}$$

Combining Equation 4 and Equation 5, we obtain

$$\log\Big(1 + (C-1)e^{-\gamma_{\mathrm{CE}}(\boldsymbol{z}, y)}\Big) \;\geq\; \log\big(1 + (C-1)e^{-\gamma_{\mathrm{LB}}}\big)\,.$$

Since the logarithm is monotone increasing and $(C-1) > 0$, it follows that

$$\gamma_{\mathrm{CE}}(\boldsymbol{z}, y) \;\leq\; \gamma_{\mathrm{LB}},$$

and therefore,

$$\gamma_{\mathrm{EDL}}(\boldsymbol{z}, y) \;\geq\; \gamma_{\mathrm{CE}}(\boldsymbol{z}, y). \tag{6}$$

Algorithm 1 Training and Inference of Evidential Transformation Network

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, pretrained model $\theta = h \circ \phi$, number of MC samples M, monotonically increasing function f,
Parameters: Evidential Transformation Network θ_{ETN}, prior belief term $\boldsymbol{b}$
for $(x, y) \in \mathcal{D}$ **do** ▷ Loop over dataset
  $\theta_A \leftarrow \theta_{\mathrm{ETN}}\big(\phi(x)\big)$ ▷ Compute parameters for variational distribution
  $\boldsymbol{z} \leftarrow \theta(x)$ ▷ Compute logits for sample x
  $\mathcal{P} \leftarrow \emptyset$
  for $m \leftarrow 1$ **to** M **do**
    $A^{(m)} \sim Dist(\theta_A)$ ▷ Sample from variational distribution
    $\boldsymbol{\alpha}' \leftarrow f\big(A^{(m)}\boldsymbol{z}\big) + \boldsymbol{b}$
    $p' = \mathbb{E}_{\boldsymbol{\pi}\sim\mathrm{Dir}(\boldsymbol{\alpha}')}[p(y \mid \boldsymbol{\pi})]$
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{p'\}$
  $\bar{p}' \leftarrow \frac{1}{M}\sum_{p'\in\mathcal{P}} p'$
  if TRAINING **then**
    Backprop through $\mathcal{L}_{\mathrm{ETN}}\big(\theta_{\mathrm{ETN}}\big)$
  else
    return $\arg\max_y \; \bar{p}'$

Define event A as the event that Equation 5 holds, and event B as the event that Equation 6 holds. From the derivation above, we have $A \subseteq B$, which implies $P(A) \leq P(B)$ [6]. Thus,

$$\begin{aligned}&P(\gamma_{\mathrm{EDL}}(\boldsymbol{z}, y) \geq \gamma_{\mathrm{CE}}(\boldsymbol{z}, y)) \\ &\quad\geq\; P\Big(L \geq \log\Big(1 + \tfrac{C-1}{e^{f^{-1}(\nu - b_y - \eta) - f^{-1}(\eta)}}\Big)\Big).\end{aligned} \tag{7}$$

2.3. Proof to Corollary 1

With f as *softplus*, $f^{-1}(x) = \log(e^x - 1)$. Plugging into Equation 7, we get:

$$\begin{aligned}&P\big(\gamma_{\mathrm{EDL}}(\boldsymbol{z}, y) \;\geq\; \gamma_{\mathrm{CE}}(\boldsymbol{z}, y)\big) \\ &\quad\geq\; P\Big(L \geq \log\big(1 + (C-1)\tfrac{e^{\eta} - 1}{e^{\nu - b_y - \eta} - 1}\big)\Big).\end{aligned}$$

3. Modeling Transformation Parameterizations

In this section, we describe how the transformation parameter A is modeled when defined as a scalar, vector, or matrix. Specifically, we explain (1) how the variational distribution over A is constructed, and (2) how the prior term $\boldsymbol{b}$ is handled. For clarity, we denote the scalar case by a, the vector case by $\boldsymbol{a}$, and the matrix case by $\boldsymbol{A}$.

Scalar ($a \in \mathbb{R}_+$). We constrain $a > 0$ and model it with a Gamma distribution:

$$a \sim \mathrm{Gamma}(\alpha^{\mathrm{G}},\, \beta^{\mathrm{G}}),$$

where the shape α^{G} and rate β^{G} are predicted by ETN. To strictly preserve accuracy, we set all elements of $\boldsymbol{b}$ to be identical, i.e.,

$$b_1 = b_2 = \cdots = b_C.$$

Vector ($\boldsymbol{a} \in \mathbb{R}_+^C$). We model $\boldsymbol{a}$ as a product of Gamma distributions, one per class:

$$\boldsymbol{a} \sim \prod_{i=1}^{C} \mathrm{Gamma}(\alpha_i^{\mathrm{G}}, \beta_i^{\mathrm{G}})$$

ETN predicts the shape $\boldsymbol{\alpha}^{\mathrm{G}} = (\alpha_1^{\mathrm{G}}, \ldots, \alpha_C^{\mathrm{G}})^\top$ and rate $\boldsymbol{\beta}^{\mathrm{G}} = (\beta_1^{\mathrm{G}}, \ldots, \beta_C^{\mathrm{G}})^\top$. For $\boldsymbol{b}$, we treat each element independently and train them separately.

Matrix ($\boldsymbol{A} \in \mathbb{R}^{C \times C}$). Matrix transformation are a natural choice since they directly operate in Dirichlet space [13]. Although the Wishart distribution[26] would be a natural distribution for positive-definite matrices, in practice we found its parameterization too restrictive and its reparamterization unstable during training. Instead, we model the flattened matrix as a Gaussian with Kronecker-factored covariance:

$$\mathrm{vec}(\boldsymbol{A}) \;\sim\; \mathcal{N}(\boldsymbol{\mu},\; \boldsymbol{\Sigma}),$$

where $\boldsymbol{\Sigma} = \boldsymbol{B} \otimes \boldsymbol{D}$, with $\boldsymbol{B} = \boldsymbol{L}_B \boldsymbol{L}_B^\top$ and $\boldsymbol{D} = \boldsymbol{L}_D \boldsymbol{L}_D^\top$. ETN predicts $\boldsymbol{\mu}, \boldsymbol{L}_B, \boldsymbol{L}_D$. To encourage monotonic behavior, we apply a *softplus* to the diagonal elements of sampled $\boldsymbol{A}$, keeping off-diagonal terms unconstrained. The prior $p(\boldsymbol{A})$ is set as a Gaussian with mode and variance matching the scalar and vector Gamma priors. Additionally, we adopt the ODIR (Off-Diagonal and Intercept Regularization) loss[13] on $\boldsymbol{\mu}$ for stable optimization. As in the vector case, all elements of $\boldsymbol{b}$ are treated independently and trained separately.

4. Experimental Setting

4.1. Training Details

The hyperparameters used for training ETN are summarized in Table 1. For LLM experiments, we employ cosine learning-rate scheduling with warm-up steps. All experiments are performed using three different random seeds, and we report the mean along with 95% confidence intervals.

For post-hoc uncertainty estimation baselines, we select the checkpoint that achieves the highest accuracy on the adaptation dataset. In contrast, for ETN with scalar scaling, we select the checkpoint with the lowest loss on the adaptation dataset.

All training and inference are performed using eight NVIDIA A6000 GPUs.

4.2. Architecture of Evidential Transformation Network

The network is composed of independent modules, each predicting a parameter of the variational distribution. (e.g., for a scalar-prediction case, the network contains two modules to predict two parameters, α^{G} and β^{G}, respectively.). In the image classification case, each module is implemented as a 2-layer MLP with hidden dimension 256. For LLMs, each module is implemented as a 3-layer MLP with hidden dimension 512.

Moreover, the training and inference procedures of ETN are outlined in Algorithm 1.

Setting	VGG16	ResNet50	Llama-3.1-8B	Gemma-2-9B
Pretrain				
Batch size	1024	–	4	4
Learning rate	2.5×10^{-4}	–	2.5×10^{-4}	2.5×10^{-4}
Epochs	200	–	3	3
Uncertainty adaptation				
Batch size	1024	64	8	8
Learning rate	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-3}
Epochs	50	50	5	5
ETN				
Prior mode	10	5	100	100
Prior variance	5	5	5	5
MC samples	20	20	20	20
λ	1	1×10^{-3}	1	1
ν	1×10^4	1×10^4	1×10^4	1×10^4

Table 1. Training and hyperparameter settings for each model.

4.3. Datasets

CIFAR-10. Since CIFAR-10 does not include an official validation split, we use 5% of the original training set for post-hoc uncertainty adaptation and the remaining 95% for pretraining the VGG16 model. Evaluation is conducted on the CIFAR-10 test set, as well as the SVHN and CIFAR-100 test sets for OOD assessment.

ImageNet. Following Minderer et al. [19], we use 20% of the `ILSVRC_2012` validation set for post-hoc adaptation and the remaining 80% for evaluation. For ImageNet-A, ImageNet-S, and ImageNet-R, we use all available samples from each subset.

RACE. To ensure that RACE serves as in-distribution data, we train LLMs on the official training set using cross-entropy loss. The validation set is used to adapt all post-hoc uncertainty estimation methods, and the test set is used exclusively for evaluation.

OBQA. Similar to RACE, we treat OBQA as in-distribution by training LLMs on the official training set with cross-entropy loss. We use the validation set for post-hoc adaptation and the test set for evaluation.

MMLU. We use three domains from MMLU, adopting the same subsets as in Yang et al. [27]. The selected domains and their corresponding subsets are listed in Table 2.

4.4. Models

VGG16. We adopt the VGG16 architecture [24], which is composed of 16 convolutional layers followed by 3 fully

Domain and Subsets of MMLU
Computer Science:
college_computer_science
computer_security
high_school_computer_science
machine_learning
Engineering:
electrical_engineering
Math:
college_mathematics
high_school_mathematics
abstract_algebra

Table 2. MMLU domains and their corresponding subsets.

connected layers. Batch normalization is applied to all convolutional layers. All parameters are updated during the pretraining stage, and for baselines that rely on training the original model (MAP_{CE}, MAP_{EDL} and IB-EDL), all parameters are likewise fully fine-tuned.

ResNet50. We use the ResNet50 architecture [9], a 50-layer convolutional network organized into five *stages*, each containing multiple residual blocks operating at a fixed spatial resolution and channel width. For baselines that require training the original model, all parameters are fully fine-tuned.

Llama-3.1-8B For baselines that require tuning the original pretrained model, we applied LoRA to all attention layers with a rank of 8 and lora alpha value to 16, and trained only the LoRA layers, following the setting in Li et al. [15], Yang et al. [27].

Gemma-2-9B We use the identical setting as Llama-3.1-8B.

4.5. Baselines

Deep Ensemble (DeepEns). We use an ensemble of three models in all settings. Each model is trained on the same dataset with a different random data order.

MC-Dropout (MCD). We set the number of forward passes to 20 for all settings. For image classification setting, we use the dropout layer in the pretrained model with a dropout rate of 0.2, while we use the LoRA dropout layer for LLM with a dropout rate of 0.1.

Laplace Approximation (LA). We utilize the `laplace` library proposed in Laplace-redux [4], which provides a integrated tools for bayesian adaptation of neural networks. As for CIFAR-10, We opted for the best setting proposed in the work, which applies laplace approximation on the last layer of the network with Kronecker-factored Generalized Gauss-Newton (GGN) matrix to the Hessian in a post-hoc manner. For ImageNet setting, we construct GGN matrix with diagonal matrix due to constrained resources. To compute distributional uncertainty, we use Monte Carlo sampling for predictive approximation, with the number of MC samples set to 20.

Laplace LoRA (LL). We build GGN matrix only on all LoRA layers through Kronecker factorization, following Yang et al. [27].

Dirichlet Meta-Model (DMM). For VGG16, we follow the implementation of Shen et al. [22]. For ResNet50, DMM takes the final hidden states from each stage as input, with each module consisting of a max-pooling layer followed by two fully connected layers. For LLMs, DMM receives hidden states from all transformer layers, and each module is composed of three fully connected layers and a max-pooling layer.

MAP_{EDL}. We train the model using the reverse KL formulation of $\mathcal{L}_{\text{EDL}}$, as reverse KL is known to provide more stable optimization than forward KL, primarily due to its mode-seeking behavior [18, 23].

IB-EDL. We follow the original implementation from Li et al. [15] for LLM experiments. For image classification, we modify the architecture by doubling the dimension of the final layer to model both the mean and variance for each class.

Static scaling. We adopt the static scaling approach inspired by Guo et al. [8], Niculescu-Mizil and Caruana [20], Platt et al. [21] for all experimental settings, and train the additional parameters using the reverse KL formulation of $\mathcal{L}_{\text{EDL}}$.

AdaTS. We use the original implementation from Joy et al. [11] for all experimental settings, and train the additional parameters using the reverse KL formulation of $\mathcal{L}_{\text{EDL}}$.

5. Additional Experiments

5.1. OOD-Detection Baselines

In this section, we compare ETN against ODIN [16] and the Mahalanobis distance method (MD) [14]. Although neither ODIN nor MD are strictly uncertainty estimation methods, we include them as they both work in post-hoc manner, and

Method	CIFAR10 → CIFAR10-OOD	ImageNet → ImageNet-OOD	OBQA → MMLU	RACE → MMLU
MD	45.43 1.2 / 56.69 1.06	**87.54** 0.16 / 77.45 0.24	70.5 0.04 / 54.42 0.05	87.28 0.01 / 54.44 0.02
ODIN	**86.41** 0.99 / **87.29** 0.82	79.77 0.00 / 72.11 0.00	61.35 0.00 / 50.11 0.00	81.22 0.00 / 49.65 0.00
ETN	85.93 0.92/ 86.5 0.97	84.78 0.36 / **79.86** 0.49	**91.7** 0.00 / **83.39** 0.01	**96.80** 0.39 / **87.57** 0.01

Table 3. Comparison of ETN to OOD-detection methods. We showcase both AUPR and AUROC scores, respectively. For ETN, we outline the scores based on maximum probability.

Method	CIFAR-10		→ CIFAR-OOD	ImageNet		→ ImageNet-OOD	RACE		→ MMLU	OBQA		→ MMLU
	ACC	UE	UE	ACC	UE	UE	ACC	UE	UE	ACC	UE	UE
DUQ	42.25 9.4	42.05 9.3	54.86 8.0	0.09 0.0	0.18 0.0	69.77 1.9	21.52 0.0	22.54 0.2	74.31 11.7	27.60 0.0	26.74 1.1	50.85 3.0
SNGP	83.62 1.4	90.75 1.3	57.72 4.7	12.83 1.0	12.83 0.9	65.73 0.8	45.73 19.5	49.97 22.9	95.11 1.1	39.53 17.8	43.74 20.8	**94.74** 4.3
ETN	**90.70** 0.0	**98.99** 0.1	**85.93** 0.9	**79.61** 0.0	**88.04** 0.1	**79.86** 0.5	**89.69** 0.0	**97.60** 0.0	**96.80** 0.0	**88.80** 0.0	**97.15** 0.0	91.70 0.0

Table 4. Comparison of ETN with deterministic uncertainty estimation methods in terms of accuracy (ACC) and uncertainty estimation (UE), where UE is measured by AUPR. For ETN, we report UE based on maximum probability.

there exists close relationship between uncertainty estimation and OOD detection [7]. We report both AUPR and Area Under the Receiver Operating Characteristic Curve (AUROC) metrics, and for ETN we showcase scores based on maximum probability. The results are summarized in Table 3.

On CIFAR-10 and ImageNet, ODIN and MD achieve higher AUPR scores than ETN, respectively. However, on OBQA and RACE, ETN outperforms both baselines across AUPR and AUROC. It is also worth noting that MD requires learning class-conditional feature distributions, which becomes resource-intensive as the number of classes grows, while ODIN is highly sensitive to its hyperparameters. By contrast, ETN avoids these limitations by operating directly in logit space, providing a lightweight and broadly applicable alternative.

5.2. Deterministic Deep Neural Network Baselines

In this section, we compare ETN against deterministic deep neural network baselines that estimate uncertainty using a single model and a single forward pass. Specifically, we use DUQ [25] and SNGP [17] as baselines. The comparison results are summarized in Table 4.

Across all image classification and QA settings except OBQA → MMLU, ETN consistently outperforms these baselines in uncertainty estimation without sacrificing accuracy. One possible reason is that DUQ requires a separate learnable weight matrix for each class, while SNGP requires learning a class-wise covariance structure for the Gaussian process. Such additional parameters for post-hoc adaptation can lead to overfitting when only a limited adaptation dataset is available, as also observed for other baselines such as Laplace Approximation and Dirichlet Meta Model.

5.3. More on Comparison of Transformation Methods

In this section, we present additional results on CIFAR-10 and OBQA, comparing different transformation strategies—specifically, static scaling and AdaTS. The results are shown in Figure 1. Since CIFAR-10 is considerably smaller and simpler than the other datasets we evaluate, all three methods—static scaling, AdaTS, and ETN—achieve reasonably strong uncertainty estimation performance. AdaTS performs on par with ETN in terms of mutual information, while static scaling trails ETN by roughly 5%.

On OBQA, however, the differences between methods become more pronounced. Both static scaling and AdaTS exhibit substantially lower mutual information compared to ETN, with margins of approximately 13.6% and 24.7%, respectively. These results highlight two key observations: (1) modeling sample-dependent transformation parameters is crucial for reliable uncertainty estimation, and (2) among sample-dependent approaches, our variational inference framework more effectively transforms logits to produce high-quality evidential uncertainty estimates.

5.4. More on Comparison Across Transformation Dimensionalities

In this section, we take a closer look at how the dimensionality of the transformation parameter A affects uncertainty estimation performance across different transformation methods.

Results of ETN. We first analyze the behavior of ETN. For ImageNet, we exclude the matrix case since the corresponding covariance matrix would contain on the order of 10^{12} entries, which is intractable to store in GPU memory. The results are shown in Figure 2.

On ImageNet, the scalar configuration outperforms the

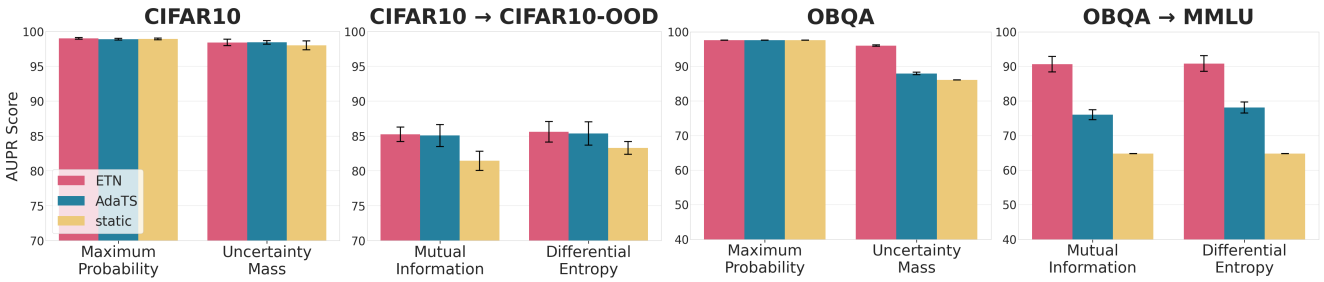

Figure 1. Comparison of uncertainty estimation performance based on different transformation methods on CIFAR-10 and OBQA.

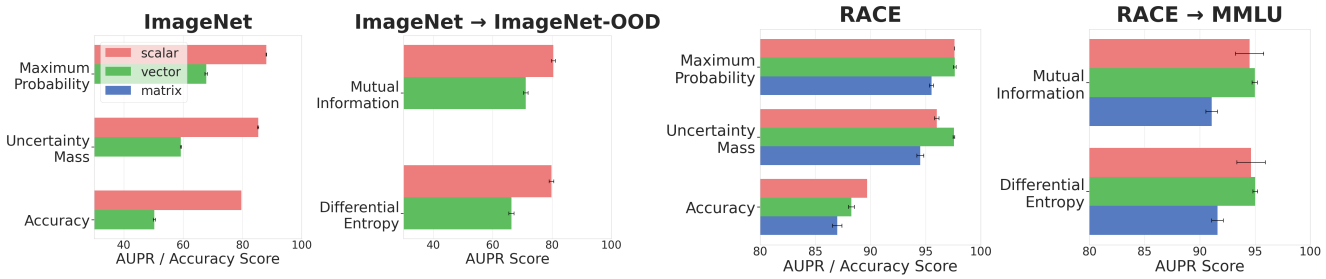

Figure 2. Comparison of uncertainty estimation performance and accuracy across different dimensionalities of the transformation parameter A modeled by ETN on ImageNet and RACE.

vector configuration for both confidence estimation and OOD detection. In contrast, on RACE, the vector configuration achieves the best performance on both ID and OOD metrics.

Results of static scaling. We next consider static scaling with different dimensionalities of A. The results are presented in Figure 4.

For static scaling, all dimensionalities yield broadly similar uncertainty estimation performance on most datasets. An exception is ImageNet, where the maximum predicted probability tends to decrease as dimensionality increases, while OOD detection performance improves.

Results of AdaTS. Finally, we evaluate on AdaTS, with results summarized in Figure 5. In this case, higher-dimensional variants generally improve OOD detection compared to the scalar configuration. However, the behavior in confidence estimation is less consistent: maximum probability typically decreases while uncertainty mass increases, with ImageNet showing particularly irregular trends.

Discussion. Across ETN, static scaling, and AdaTS, a consistent trend emerges: increasing the dimensionality of the transformation parameter A tends to degrade predictive accuracy and introduces a clear trade-off between OOD detection performance and core predictive capability. Moreover, none of the higher-dimensional variants—including the matrix formulation that operates directly in *Dirichlet space*—surpasses scalar-based ETN across all datasets and metrics, with the sole exception of the maximum probability metric on ImageNet. Taken together, these results suggest that a simple scalar-based transformation within ETN offers the most effective and practical balance for adapting pretrained models to the EDL framework.

5.5. AUPR Scores on Gemma-2-9B

To further assess the robustness of ETN across different pretrained architectures, we evaluate its AUPR performance on Gemma-2-9B using OBQA and RACE. The results are shown in Table 5. Consistent with our findings on Llama-3.1, ETN delivers the strongest uncertainty estimation performance while fully preserving the model's predictive accuracy. These results provide additional evidence that ETN generalizes effectively across diverse large-scale pretrained models.

5.6. Uncertainty Estimation Performance Based on AUROC Scores

Although recent works increasingly adopt AUPR as the primary metric for evaluating uncertainty estimation capability [3, 5, 28], we additionally report AUROC scores for image classification in Table 6 and for LLMs in Table 7.

On CIFAR-10, we observe that ETN outperforms all baselines across all AUROC-based metrics, showcasing its robustness across different uncertainty evaluation criteria.

For ImageNet, AUROC trends largely mirror those ob-

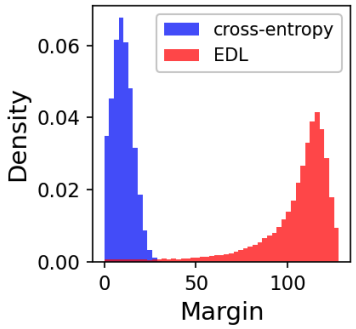

Figure 3. Histograms of logit margins for models trained with EDL and CE.

served with AUPR. ETN remains competitive in OOD detection, while Laplace Approximation attains slightly higher AUROC for mutual information in some settings. In confidence estimation, we observe that DMM attains unusually high AUROC scores relative to its accuracy and AUPR. Upon inspecting its predictions, we find that DMM often produces nearly uniform predictive distributions with low α_0, indicating uniformly high uncertainty across both ID and OOD inputs. This suggests that the inflated AUROC scores do not reflect reliable or informative confidence estimates.

For RACE and OBQA, ETN achieves the strongest OOD detection performance for both Llama-3.1 and Gemma-2. Moreover, in every setting, at least one confidence estimation metric (maximum probability or uncertainty mass) is maximized by ETN.

Although ETN is less dominant under AUROC than under AUPR, it remains competitive with strong baselines across all AUROC metrics while clearly outperforming them on our primary metric, AUPR. Overall, these results support ETN as an effective and practical method for uncertainty estimation in pretrained models.

5.7. Empirical Comparison of Margins between EDL- and CE-Pretrained Models

In this section, we empirically examine the logit margins of EDL- and CE-pretrained models to assess whether enlarging margins during the transformation process, as done by ETN, is a justified approach. For this experiment, we use VGG16 on CIFAR-10. We compare a model trained from scratch with $\mathcal{L}_{\text{EDL}}$, where $\lambda = 0.01$ and $f(\cdot) = \text{softplus}$, against a model trained from scratch with $\mathcal{L}_{\text{CE}}$. The remaining pretraining settings follow those in Table 1. Margins are computed on the CIFAR-10 training set.

Figure 3 shows the resulting margin histograms. The EDL-pretrained model exhibits larger margins than the CE-pretrained model. Together with Corollary 1, this result supports the validity of enlarging logit margins during the transformation process.

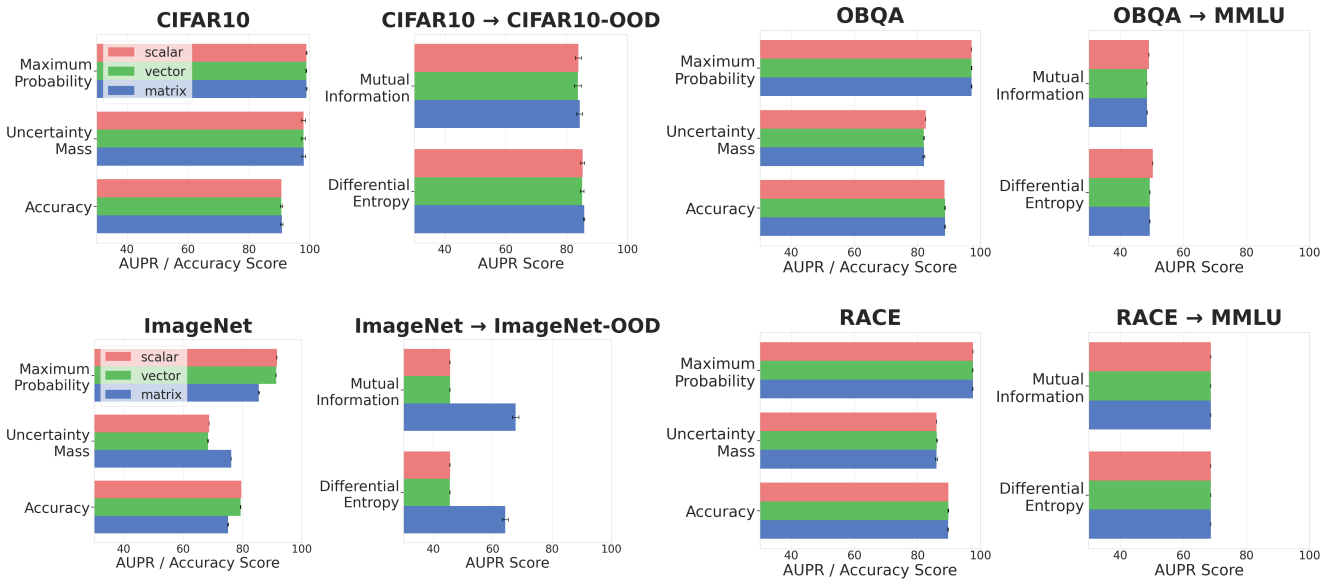

Figure 4. Comparison of uncertainty estimation performance and accuracy across different dimensionalities of the transformation parameter A modeled by **static** scaling.

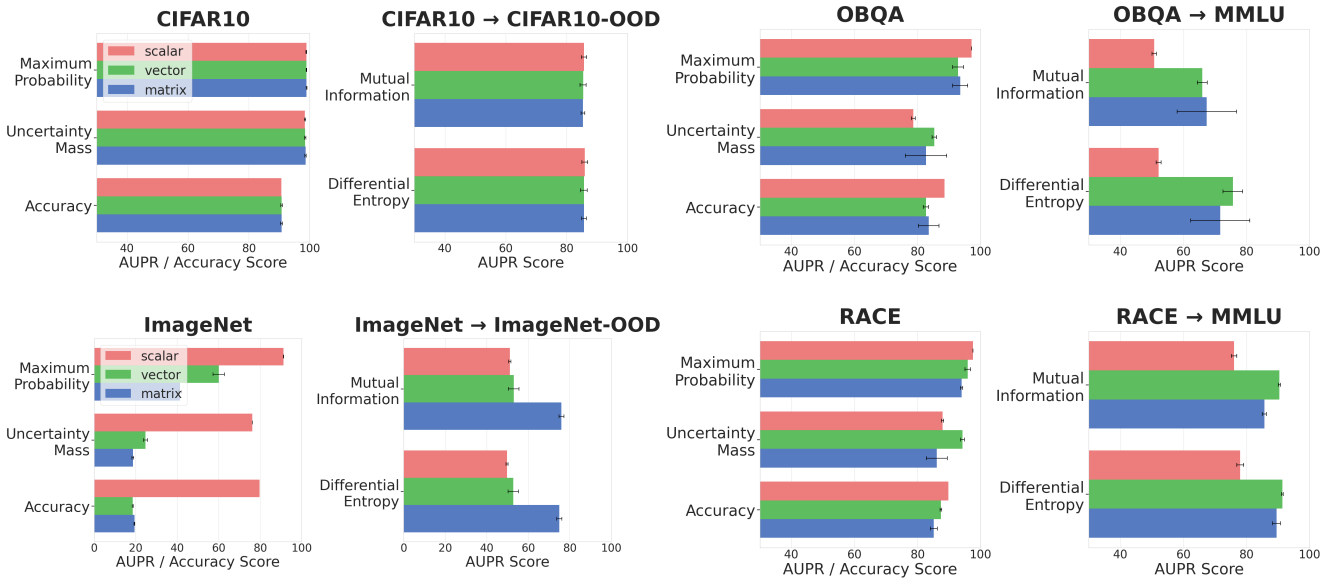

Figure 5. Comparison of uncertainty estimation performance and accuracy across different dimensionalities of the transformation parameter A modeled by **AdaTS**.

Method	RACE			RACE → MMLU			OBQA			OBQA → MMLU		
	ACC	MP	UM	MP	MI	DE	ACC	MP	UM	MP	MI	DE
MAP_{CE}	88.81 0.2	98.16 0.1	–	96.03 0.1	–	–	88.87 0.6	97.59 0.2	–	85.85 0.1	–	–
DeepEns	89.09 0.1	98.26 0.0	–	94.64 0.2	93.79 0.4	–	89.00 0.0	97.84 0.3	–	83.04 0.9	78.82 0.5	–
MCD	89.38 0.4	98.40 0.2	–	96.36 0.3	**95.36** 0.4	–	89.60 0.5	97.87 0.5	–	85.59 1.4	81.81 1.8	–
LL	88.51 0.2	69.05 0.3	–	27.47 1.4	27.03 1.0	–	92.53 0.2	79.45 0.6	–	25.32 0.6	26.25 0.8	–
MAP_{EDL}	87.14 0.2	92.27 0.7	87.60 1.1	93.97 1.5	83.89 1.1	90.36 1.2	87.53 0.4	94.63 0.9	86.32 3.3	80.28 2.7	63.50 1.5	79.78 2.6
DMM	89.42 0.2	97.96 0.1	95.20 0.4	93.84 0.3	92.77 0.6	93.80 0.3	93.00 0.6	96.68 0.0	95.44 0.1	81.48 0.5	80.53 0.4	81.51 0.5
IB-EDL	87.98 0.3	93.72 0.1	88.04 0.4	90.67 0.9	89.43 1.5	90.70 1.1	85.13 0.4	94.60 0.4	92.15 0.1	79.99 0.8	66.58 2.1	74.28 1.7
ETN	**89.48** 0.0	**98.43** 0.0	**95.94** 0.1	**96.70** 0.0	95.21 0.3	**95.29** 0.3	**93.20** 0.0	**98.06** 0.0	**96.00** 0.1	**86.95** 0.1	**82.69** 0.2	**84.39** 0.2

Table 5. AUPR scores of Gemma-2-9B on RACE and OBQA, using MMLU subsets as OOD data.

Method	CIFAR-10		CIFAR-10 → SVHN			CIFAR-10 → CIFAR-100		
	MP	UM	MP	MI	DE	MP	MI	DE
MAP_{CE}	87.71 0.1	–	73.26 5.6	–	–	79.10 2.3	–	–
DeepEns	90.78 0.5	–	87.43 1.1	49.19 0.2	–	83.63 0.8	49.39 0.3	–
MCD	77.67 5.4	–	73.26 4.2	78.56 1.4	–	68.72 6.2	75.62 2.5	–
LA	90.23 0.3	–	84.22 0.1	83.46 0.2	–	82.38 0.2	81.93 0.2	–
MAP_{EDL}	87.91 0.5	86.98 0.8	81.61 0.7	81.58 3.1	81.93 1.9	80.27 0.7	82.24 0.4	81.89 0.4
DMM	90.65 1.2	83.36 1.9	85.68 2.6	80.20 9.7	84.99 6.6	79.25 3.0	79.69 5.5	82.43 4.0
IB-EDL	87.58 1.3	86.19 1.0	78.22 2.9	76.72 3.0	77.64 3.0	79.39 1.7	79.41 1.2	79.76 1.5
ETN	**91.80** 0.9	**87.98** 2.8	**88.60** 1.4	**88.40** 0.9	**88.79** 1.2	**84.40** 0.5	**84.50** 0.6	**84.84** 0.5

Method	ImageNet		ImageNet → ImageNet-A			ImageNet → ImageNet-S			ImageNet → ImageNet-R		
	MP	UM	MP	MI	DE	MP	MI	DE	MP	MI	DE
MAP_{CE}	80.28 0.2	–	70.51 0.2	–	–	67.78 2.9	–	–	67.43 1.0	–	–
DeepEns	63.83 0.4	–	53.39 2.3	49.63 0.1	–	26.96 2.7	50.15 0.3	–	54.25 1.9	50.01 0.1	–
MCD	78.90 0.4	–	67.64 1.6	50.17 0.2	–	60.11 3.5	50.12 0.0	–	66.94 0.8	50.02 0.1	–
LA	49.16 1.5	–	78.39 0.1	**83.55** 0.0	–	75.58 0.1	**81.30** 0.0	–	75.39 0.1	**79.97** 0.0	–
MAP_{EDL}	72.72 0.3	55.10 0.3	70.80 0.8	66.07 1.1	66.49 1.1	68.83 5.3	69.50 4.6	69.90 4.7	68.11 2.3	71.41 2.4	71.55 2.4
DMM	**92.54** 0.4	**91.86** 0.4	48.30 0.4	48.38 0.4	48.36 0.3	43.48 3.6	43.82 3.4	43.77 3.4	48.15 2.0	48.24 1.9	48.25 1.9
IB-EDL	91.05 0.1	48.70 0.1	55.50 1.8	50.03 0.2	48.84 0.3	54.65 1.6	49.00 0.7	49.84 0.2	55.03 1.4	49.63 0.3	49.42 0.1
ETN	69.32 0.3	64.78 0.4	**81.81** 0.2	77.21 0.1	**76.35** 0.9	**78.34** 0.6	74.08 0.9	**73.44** 1.4	**79.42** 0.7	75.63 1.0	**74.88** 1.6

Table 6. AUROC scores on CIFAR-10, SVHN, and CIFAR-100 (top), and on ImageNet, ImageNet-A, ImageNet-S, and ImageNet-R (bottom).

Method	RACE		RACE → MMLU			OBQA		OBQA → MMLU		
	MP	UM	MP	MI	DE	MP	UM	MP	MI	DE
MAP_{CE}	**87.59** 0.4	–	87.02 0.9	–	–	81.94 0.9	–	83.48 1.3	–	–
DeepEns	86.03 0.5	–	80.76 0.8	73.49 2.0	–	80.61 0.5	–	80.33 1.4	75.67 0.8	–
MCD	87.11 0.0	–	86.39 0.01	**85.17** 0.8	–	83.02 0.2	–	81.96 0.0	69.58 0.5	–
LL	45.04 0.8	–	49.82 1.9	53.43 1.3	–	41.32 0.2	–	47.43 0.7	46.37 0.6	–
MAP_{EDL}	70.10 0.7	61.71 2.2	70.22 0.9	65.15 1.5	70.16 0.9	64.44 1.7	48.60 0.2	72.17 1.7	66.99 2.2	72.53 1.9
DMM	82.34 0.4	75.21 2.8	77.88 2.0	76.85 0.7	78.89 1.6	77.12 1.1	69.32 3.7	78.98 0.6	73.51 2.5	78.47 0.4
IB-EDL	82.85 0.4	77.21 0.6	80.48 1.6	66.86 2.9	80.17 1.7	79.98 0.6	63.71 0.9	82.83 1.2	67.89 2.1	83.16 1.2
ETN	84.96 0.0	**77.69** 0.5	**87.57** 0.0	81.42 3.0	**81.81** 3.1	**83.21** 0.0	**73.60** 1.0	**87.09** 0.0	**85.30** 1.2	**86.27** 0.9

Method	RACE		RACE → MMLU			OBQA		OBQA → MMLU		
	MP	UM	MP	MI	DE	MP	UM	MP	MI	DE
MAP_{CE}	89.39 0.3	–	84.74 0.2	–	–	81.35 0.2	–	83.48 1.3	–	–
DeepEns	89.41 0.4	–	80.13 0.7	77.11 1.1	–	**88.03** 1.7	–	75.44 1.2	67.04 1.2	–
MCD	90.45 0.4	–	85.50 1.0	81.61 0.8	–	81.83 1.8	–	78.99 1.5	75.31 1.7	–
LL	36.16 0.8	–	52.36 1.8	51.37 1.8	–	29.94 1.0	–	47.43 0.7	51.54 0.7	–
MAP_{EDL}	71.43 3.0	61.76 2.0	80.65 3.4	64.75 3.9	77.12 3.4	76.90 2.5	56.79 9.8	73.38 3.7	59.13 1.5	73.41 3.9
DMM	88.96 0.4	**84.28** 1.6	79.72 0.7	76.31 1.4	79.48 0.7	77.97 1.5	**75.12** 2.2	78.98 0.6	73.51 2.5	78.47 0.4
IB-EDL	78.58 1.0	62.83 0.6	74.67 1.5	68.19 3.5	72.94 2.4	80.67 0.7	69.30 0.5	73.78 1.0	61.76 2.8	68.66 2.1
ETN	**91.30** 0.0	80.97 1.2	**87.00** 0.0	**83.12** 0.8	**83.31** 0.8	82.68 0.0	64.87 0.7	**85.67** 0.0	**78.82** 0.3	**80.92** 0.3

Table 7. AUROC scores on on RACE and OBQA, using MMLU subsets as OOD data. Results on Llama-3.1-8B is reported (top), and Gemma-2-9B is reported (bottom).

References

[1] Guy Bar-Shalom, Fabrizio Frasca, Derek Lim, Yoav Gelberg, Yftah Ziser, Ran El-Yaniv, Gal Chechik, and Haggai Maron. Beyond next token probabilities: Learnable, fast detection of hallucinations and data contamination on llm output distributions, 2025. 1

[2] Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. On second-order scoring rules for epistemic uncertainty quantification, 2023. 1

[3] Mengyuan Chen, Junyu Gao, and Changsheng Xu. R-edl: Relaxing nonessential settings of evidential deep learning. In *The Twelfth International Conference on Learning Representations*, 2024. 6

[4] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux-effortless bayesian deep learning. *Advances in neural information processing systems*, 34:20089–20103, 2021. 4

[5] Danruo Deng, Guangyong Chen, Yang Yu, Furui Liu, and Pheng-Ann Heng. Uncertainty estimation by fisher information-based evidential deep learning, 2023. 6

[6] M.J. Evans and J.S. Rosenthal. *Probability and Statistics: The Science of Uncertainty*. W. H. Freeman, 2004. 2

[7] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589, 2023. 5

[8] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 4

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 4

[10] Gaurush Hiranandani, Haolun Wu, Subhojyoti Mukherjee, and Sanmi Koyejo. Logits are all we need to adapt closed models, 2025. 1

[11] Tom Joy, Francesco Pinto, Ser-Nam Lim, Philip HS Torr, and Puneet K Dokania. Sample-dependent adaptive temperature scaling for improved calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14919–14926, 2023. 4

[12] Mira Juergens, Nis Meinert, Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. Is epistemic uncertainty faithfully represented by evidential deep learning methods? In *International Conference on Machine Learning*, pages 22624–22642. PMLR, 2024. 1

[13] Meelis Kull, Miquel Perello Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32, 2019. 3

[14] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018. 4

[15] Yawei Li, David Rügamer, Bernd Bischl, and Mina Rezaei. Calibrating llms with information-theoretic evidential deep learning. *arXiv preprint arXiv:2502.06351*, 2025. 4

[16] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. 4

[17] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in neural information processing systems*, 33:7498–7512, 2020. 5

[18] Andrey Malinin and Mark Gales. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 4

[19] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. *Advances in neural information processing systems*, 34:15682–15694, 2021. 3

[20] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632, 2005. 4

[21] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 4

[22] Maohao Shen, Yuheng Bu, Prasanna Sattigeri, Soumya Ghosh, Subhro Das, and Gregory Wornell. Post-hoc uncertainty learning using a dirichlet meta-model, 2022. 4

[23] Maohao Shen, Jongha Jon Ryu, Soumya Ghosh, Yuheng Bu, Prasanna Sattigeri, Subhro Das, and Gregory Wornell. Are uncertainty quantification capabilities of evidential deep learning a mirage? *Advances in Neural Information Processing Systems*, 37:107830–107864, 2024. 1, 4

[24] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015. 3

[25] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020. 5

[26] John Wishart. The generalised product moment distribution in samples from a normal multivariate population. *Biometrika*, 20(1/2):32–52, 1928. 3

[27] Adam X Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. Bayesian low-rank adaptation for large language models. In *The Twelfth International Conference on Learning Representations*. 3, 4

[28] Taeseong Yoon and Heeyoung Kim. Uncertainty estimation by density aware evidential deep learning. *arXiv preprint arXiv:2409.08754*, 2024. 6