CALIBRATING LLMS WITH INFORMATION-THEORETIC EVIDENTIAL DEEP LEARNING

Yawei Li **David Rügamer** **Bernd Bischl** **Mina Rezaei**
Department of Statistics, LMU Munich
Munich Center for Machine Learning (MCML)

ABSTRACT

Fine-tuned large language models (LLMs) often exhibit overconfidence, particularly when trained on small datasets, resulting in poor calibration and inaccurate uncertainty estimates. Evidential Deep Learning (EDL), an uncertainty-aware approach, enables uncertainty estimation in a single forward pass, making it a promising method for calibrating fine-tuned LLMs. However, despite its computational efficiency, EDL is prone to overfitting, as its training objective can result in overly concentrated probability distributions. To mitigate this, we propose regularizing EDL by incorporating an information bottleneck (IB). Our approach **IB-EDL** suppresses spurious information in the evidence generated by the model and encourages truly predictive information to influence both the predictions and uncertainty estimates. Extensive experiments across various fine-tuned LLMs and tasks demonstrate that IB-EDL outperforms both existing EDL and non-EDL approaches. By improving the trustworthiness of LLMs, IB-EDL facilitates their broader adoption in domains requiring high levels of confidence calibration. Code is available at `https://github.com/sandylaker/ib-edl`.

1 INTRODUCTION

Large language models (LLMs) have revolutionized natural language processing, with fine-tuning emerging as a prevalent method to adapt these models for specific tasks or domains (Houlsby et al., 2019; Hu et al., 2022). However, *fine-tuned* LLMs often display overconfidence in their predictions (Jiang et al., 2021; Yang et al., 2024), which compromises their reliability and limits their applicability in critical domains where trustworthiness is essential.

Overconfidence in LLMs often manifests as poor calibration, where the predicted probabilities do not accurately reflect the model's uncertainty about its predictions. Uncertainty-aware methods improve calibration by explicitly quantifying the uncertainty in the model's predictions, allowing the model to produce confidence scores that better correspond to the actual likelihood of correctness. Traditional uncertainty-aware methods, such as MC-Dropout (Gal & Ghahramani, 2016) and Deep Ensemble (Lakshminarayanan et al., 2017; Fort et al., 2019) are commonly used to mitigate overconfidence in neural networks. However, these approaches typically require multiple forward passes, significantly increasing the inference time for LLMs.

Evidential Deep Learning (EDL) (Sensoy et al., 2018; Malinin & Gales, 2018) offers a more efficient alternative by providing uncertainty estimates with a single forward pass. Despite its success in various tasks, recent studies (Deng et al., 2023; Chen et al., 2024) indicate that EDL can still yield overconfident predictions which leads to inaccurate uncertainty estimates, degrading the model's calibration performance. This issue arises from the propensity of vanilla EDL to encourage models to generate excessive evidence (i.e., support for a class) with extremely large magnitudes, leading to overly confident predicted class probabilities.

Motivated by these challenges, we propose a novel regularization approach for EDL using an *information bottleneck*, which we term **IB-EDL**. IB-EDL adaptively distorts the evidence generated by the LLM while maximally preserving the model's performance. In doing so, IB-EDL encourages the model to suppress spurious or uninformative evidence that could lead to overconfident predictions. Our theoretical analysis shows that the information bottleneck effectively penalizes the generation

of disproportionately large evidence, thereby reducing overconfidence. Notably, our method introduces less than 2% computational overhead compared to a pretrained LLM, maintaining the model's inference efficiency while significantly improving its calibration. Our contributions are as follows:

- We introduce IB-EDL, an information-theoretic framework for regularizing EDL. First, we identify a theoretical issue previously overlooked in the IB literature. In the context of EDL, we address this challenge through a novel choice of the IB stochastic variable. Moreover, our solution naturally imposes an ℓ_2 regularization, effectively mitigating the issue of overly large evidence highlighted in prior EDL research.
- We show that several existing EDL methods can be seen as a special case within the IB-EDL framework. This unification offers a cohesive perspective on these approaches.
- We perform extensive experiments on calibrating *fine-tuned* LLMs using EDL, thereby extending its applicability beyond the medium-sized networks commonly used in the EDL literature. Our results across various LLMs and datasets demonstrate that IB-EDL scales effectively.

2 BACKGROUND

2.1 MLE FINE-TUNING OF LLMS

Let $\boldsymbol{x} \in \mathcal{V}^S$ represent the input sequence for an LLM, where $\mathcal{V}$ represents the set of tokens (vocabulary) and S is the sequence length. The target space is denoted by $\mathcal{Y}$, which may be identical to $\mathcal{V}$ (e.g., in next-token prediction) or a different set (e.g., in sentiment analysis). We generally assume that $|\mathcal{Y}| = C$. As we focus on the context of LLMs, *we will use the term "tokens" instead of "classes"* throughout this paper. In tasks like next-token prediction, the target can also be a sequence of tokens. For clarity in our theoretical analysis, we focus on a single-token target, with the understanding that a token sequence can be treated as multiple single-token targets.

Let f be the LLM. The output logits of the LLM, $f(\boldsymbol{x}) \in \mathbb{R}^C$, are passed through a Softmax function, yielding a vector $\boldsymbol{\pi}$ with entries $\pi_j = \exp(f(\boldsymbol{x})_j) / \sum_{j'=1}^{C} \exp(f(\boldsymbol{x})_{j'})$, which represent the probability for each token. Let $\boldsymbol{y} \in \{0,1\}^C$ be the one-hot encoded target. Then, $\boldsymbol{y}|\boldsymbol{x}$ conforms to a categorical distribution $p(\boldsymbol{y}|\boldsymbol{x}) = \mathrm{Cat}(\boldsymbol{y};\boldsymbol{\pi}) = \prod_{j=1}^{C} \pi_j^{y_j}$. Fine-tuning LLMs on downstream tasks typically involves minimizing $-\log p(\boldsymbol{y}|\boldsymbol{x})$ which corresponds to maximum likelihood estimation (MLE). However, fine-tuning LLMs on small downstream datasets can result in overfitting and overconfident predictions. Additionally, MLE yields a *deterministic* model that cannot express uncertainty in the predicted $\boldsymbol{\pi}$.

2.2 UNCERTAINTY-AWARE MODELING VIA EDL

While conventional uncertainty-aware methods like a Deep Ensemble can alleviate overconfidence and improve calibration, they require multiple forward passes during inference. This can be particularly challenging for LLMs due to their already substantial computational demands. EDL provides a more efficient alternative by capturing uncertainty in *a single forward pass*, making it especially suitable for large models. EDL builds on the principles of Subjective Logic (Jøsang, 1997; 2016), which is derived from Dempster-Shafer Theory (DST) (Dempster, 1968; Shafer, 1976).

EDL inference pipeline: Instead of directly predicting the token probabilities $\boldsymbol{\pi}$, EDL uses the model to predict a *Dirichlet prior* on $\boldsymbol{\pi}$. Specifically, the model's output is interpreted as *pre-evidence* $\tilde{\boldsymbol{e}} = f(\boldsymbol{x}) \in \mathbb{R}^C$, which is converted into a non-negative *evidence* vector $\boldsymbol{e} = \mathrm{SoftPlus}(\tilde{\boldsymbol{e}})$ using the SoftPlus activation. Each element e_j of the evidence vector represents the amount of support for token j being the correct prediction. Once the evidence is obtained, we can proceed to predict the Dirichlet prior $\mathrm{Dir}(\boldsymbol{\pi};\boldsymbol{\alpha})$ over the simplex of possible token probabilities $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_C]^\top$ by computing the Dirichlet parameters $\alpha_j = e_j + 1, \forall j \in [C]$. More formally,

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{j=1}^{C} \Gamma(\alpha_j)} \prod_{j=1}^{C} \pi_j^{\alpha_j - 1}, \quad \text{with } \alpha_0 = \sum_{j=1}^{C} \alpha_j, \tag{1}$$

where $\Gamma(\cdot)$ is the *gamma* function. The expected probabilities $\hat{\pi}_j$ and the final predicted token $\hat{y}$ are:

$$\hat{\pi}_j = \mathbb{E}_{\boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\pi};\boldsymbol{\alpha})} [\pi_j|\boldsymbol{\alpha}] = \frac{\alpha_j}{\alpha_0} = \frac{e_j + 1}{\sum_{j=1}^{C}(e_j + 1)}, \quad \hat{y} = \arg\max_j \hat{\pi}_j. \tag{2}$$

In summary, the EDL pipeline can be symbolized as: $\boldsymbol{x} \rightarrow f(\boldsymbol{x}) \rightarrow \tilde{\boldsymbol{e}} \rightarrow \boldsymbol{e} \rightarrow \boldsymbol{\alpha} \rightarrow \boldsymbol{\pi} \rightarrow \boldsymbol{y}$.

Uncertainty estimate: EDL also enables quantifying uncertainty in the model's prediction. This is done through the concepts of *belief mass* b_j and *uncertainty mass* u in the Subjective Logic:

$$b_j = (\alpha_j - 1)/\alpha_0, \quad u = C/\alpha_0. \tag{3}$$

Similar to the evidence, the belief mass b_j indicates the support for token j being the correct prediction, while the uncertainty mass u captures the model's overall uncertainty about the prediction. The sum of all belief masses and the uncertainty mass is normalized: $\sum_{j=1}^{C} b_j + u = 1$.

2.3 Training of EDL networks

In EDL, models are usually trained by minimizing the Bayes risk, which involves the expected loss under the Dirichlet distribution. Given the predicted $\boldsymbol{\alpha}$ from an input $\boldsymbol{x}$, and the target $\boldsymbol{y} \in \{0,1\}^C$, the Bayes risk for the cross-entropy loss is defined as $\mathcal{L}_{\text{CE}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\pi}\sim\text{Dir}(\boldsymbol{\pi};\boldsymbol{\alpha})}\left[-\sum_{j=1}^{C} y_j \log(\pi_j)\right] = \sum_{j=1}^{C} y_j(\psi(\alpha_0) - \psi(\alpha_j))$, where $\boldsymbol{\theta}$ denotes the trainable parameters of the model, and ψ is the *digamma* function. To stabilize the training, Sensoy et al. (2018) introduce the MSE loss as an alternative objective, which can be analytically computed using $\boldsymbol{\alpha}$:

$$\mathcal{L}_{\text{MSE}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\pi}\sim\text{Dir}(\boldsymbol{\pi};\boldsymbol{\alpha})}\|\boldsymbol{y} - \boldsymbol{\pi}\|_2^2 = \sum_{j=1}^{C}\left(y_j - \frac{\alpha_j}{\alpha_0}\right)^2 + \frac{\alpha_j(\alpha_0 - \alpha_j)}{\alpha_0^2(\alpha_j + 1)}. \tag{4}$$

For detailed derivations, we refer to Sensoy et al. (2018). Furthermore, they introduce a regularization term to *suppress evidence for non-target tokens*, i.e., the tokens labeled as $\mathbf{0}$ in $\boldsymbol{y}$. This is achieved by first "removing" the evidence associated with the target token, using $\tilde{\boldsymbol{\alpha}} = \boldsymbol{y} + (\mathbf{1} - \boldsymbol{y}) \odot \boldsymbol{\alpha}$, where $\mathbf{1} = [1, \ldots, 1]^\top$. The regularization term is then defined as

$$\mathcal{L}_{\text{Reg}}(\boldsymbol{\theta}) = D_{\text{KL}}\left(\text{Dir}\left(\boldsymbol{\pi};\tilde{\boldsymbol{\alpha}}\right) \parallel \text{Dir}\left(\boldsymbol{\pi};\mathbf{1}\right)\right), \tag{5}$$

where D_{KL} denotes the Kullback-Leibler (KL) divergence. The total loss for training is given by:

$$\mathcal{L}_{\text{EDL}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{MSE}}(\boldsymbol{\theta}) + \lambda \cdot \mathcal{L}_{\text{Reg}}(\boldsymbol{\theta}), \tag{6}$$

where $\lambda > 0$ is a hyper-parameter. Note that $\mathcal{L}_{\text{MSE}}(\boldsymbol{\theta})$ can be replaced with $\mathcal{L}_{\text{CE}}(\boldsymbol{\theta})$.

However, as the model is trained to minimize the empirical risk, it remains susceptible to overfitting the data and producing overconfident predictions. The objectives in $\mathcal{L}_{\text{CE}}(\boldsymbol{\theta})$ and $\mathcal{L}_{\text{MSE}}(\boldsymbol{\theta})$ drive the learned Dirichlet distribution towards a Dirac delta distribution. Consequently, the trained model may produce α_j with extreme magnitudes for the target token j (Chen et al., 2024). Eq. (5) also does not fully address this issue, as it only suppresses the evidence of non-target tokens.

Recent efforts have sought to mitigate the overconfidence issue in EDL. For instance, I-EDL (Deng et al., 2023) incorporates the Fisher Information matrix into the distribution of $\boldsymbol{y}$. R-EDL (Chen et al., 2024) alleviates overconfidence by relaxing $\alpha_j = e_j + 1, \forall j$ to $\alpha_j = e_j + \eta, \forall j$ with a hyper-parameter $\eta \in \mathbb{R}_+$. Orthogonal to these approaches, we do not alter the assumptions on $\boldsymbol{y}$ or $\boldsymbol{\alpha}$ in the EDL formulation. Instead, we impose regularization on the model using an information bottleneck (IB), which *discourages* the model from relying on irrelevant or spurious correlations that could lead to an overly concentrated Dirichlet distribution, thereby preventing overconfident predictions.

3 Information bottleneck-regularized EDL

We begin by adopting an information-theoretic perspective on neural networks and introduce the IB objective. We then explain how to regularize EDL with IB and why the final IB-EDL objective effectively mitigates overconfidence.

3.1 A Careful examination of the information bottleneck criterion

A high-level view of IB: Let X be the input random variable and Y represent the random variable of the target token. We also introduce an intermediate representation Z, which serves as a stochastic encoding of X. In the context of EDL, Z can take various forms, such as the internal features

of any LLM layer, the pre-evidence $\tilde{e}$ (i.e., model output), evidence $\boldsymbol{e}$, Dirichlet parameters $\boldsymbol{\alpha}$, or token probabilities $\boldsymbol{\pi}$. Choosing different forms of Z corresponds to selecting features that capture different levels of abstraction. We will discuss our selection of Z in Section 3.2. Since input data often contains redundant or irrelevant information, which may hinder the generalization ability of Z, it is essential for Z to retain the most predictive information about Y while discarding irrelevant information from X. This trade-off leads to better generalization. The Information Bottleneck method (Tishby et al., 1999; Tishby & Zaslavsky, 2015) formalizes this principle through the concept of mutual information. Specifically, the IB objective is:

$$\max_{\boldsymbol{\theta}} I(Z, Y; \boldsymbol{\theta}) - \beta I(Z, X; \boldsymbol{\theta}), \tag{7}$$

where $I(\cdot, \cdot)$ represents mutual information (MI), and $\beta > 0$ is a hyperparameter controlling the trade-off between relevance and compression. Since Z is computed by the model f, optimizing the model's parameters $\boldsymbol{\theta}$ is equivalent to optimizing Z. The term $I(Z, Y; \boldsymbol{\theta})$ promotes Z to be predictive of Y, while $I(Z, X; \boldsymbol{\theta})$ encourages Z to ignore irrelevant information from X. For simplicity, we omit the model parameters $\boldsymbol{\theta}$ in the following equations.

The mutual information terms in Eq. (7) are generally intractable. To address this, Alemi et al. (2017) proposed to derive more tractable variational bounds. Following Wieczorek & Roth (2020), we assume the Markov chain $X - Z - Y$[1] to derive the IB objective. Detailed derivations for the following equations are provided in Appendix A.

Upper bound of $I(Z, X)$: To derive a variational upper bound on $I(Z, X)$, we first expand it as:

$$I(Z, X) = \int p(\boldsymbol{x}, \boldsymbol{z}) \log \frac{p(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})} \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{x} = \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{x})}[\log p(\boldsymbol{z}|\boldsymbol{x})] - \mathbb{E}_{p(\boldsymbol{z})}[\log p(\boldsymbol{z})].$$

Computing $p(\boldsymbol{z}) = \int p(\boldsymbol{z}, \boldsymbol{x}) \mathrm{d}\boldsymbol{x}$ is challenging as it involves marginalizing over $\boldsymbol{x}$. Instead, Alemi et al. (2017) suggest approximating it using a predefined prior $r(\boldsymbol{z})$. We will discuss how to choose $r(\boldsymbol{z})$ in Section 3.2. By utilizing the Kullback–Leibler divergence $D_{\mathrm{KL}}\left(p(\boldsymbol{z})||r(\boldsymbol{z})\right) \geq 0$, we obtain:

$$I(Z, X) \leq \int p(\boldsymbol{z}|\boldsymbol{x}) p(\boldsymbol{x}) \log \frac{p(\boldsymbol{z}|\boldsymbol{x})}{r(\boldsymbol{z})} \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{x} = \mathbb{E}_{p(\boldsymbol{x})}\left[D_{\mathrm{KL}}\left(p(\boldsymbol{z}|\boldsymbol{x}) \| r(\boldsymbol{z})\right)\right]. \tag{8}$$

Lower bound of $I(Z, Y)$: Wieczorek & Roth (2020) derive the following lower bound:

$$\begin{aligned} I(Z, Y) &= \mathbb{E}_{p(\boldsymbol{x}, \boldsymbol{y})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{y})}\left[\log p(\boldsymbol{y}|\boldsymbol{z})\right] + H(Y) \\ &\geq \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{y}|\boldsymbol{z})\right], \end{aligned} \tag{9}$$

where $H(\cdot)$ denotes the Shannon entropy. By plugging the upper bound from Eq. (8) and lower bound from Eq. (9) into Eq. (7), and flipping the $\max$ to a $\min$, the IB objective becomes

$$\min_{\boldsymbol{\theta}} \; -\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{y}|\boldsymbol{z})\right] + \beta\, \mathbb{E}_{p(\boldsymbol{x})}\left[D_{\mathrm{KL}}\left(p(\boldsymbol{z}|\boldsymbol{x}) \| r(\boldsymbol{z})\right)\right], \tag{10}$$

where the expectations can be approximated using Monte Carlo samples. In the following paragraph and Section 3.2, we will discuss how to compute $p(\boldsymbol{y}|\boldsymbol{z})$, $p(\boldsymbol{z}|\boldsymbol{x})$, and how to select the prior $r(\boldsymbol{z})$.

Challenges when applying IB to an internal layer of an LLM: Previous works, such as Alemi et al. (2017), typically apply IB to an intermediate layer within the neural network, treating the features at that layer as the hidden variable Z. In this scenario, the earlier layers (a.k.a. the *encoder*) learn $p(\boldsymbol{z}|\boldsymbol{x})$. However, since $\boldsymbol{z}$ represents the features at an intermediate layer, the true distribution $p(\boldsymbol{y}|\boldsymbol{z})$ is unknown in practice. As a result, Alemi et al. (2017) use the later layers (the *decoder*) to learn a distribution $q(\boldsymbol{y}|\boldsymbol{z})$ to approximate $p(\boldsymbol{y}|\boldsymbol{z})$. The approximate distribution $q(\boldsymbol{y}|\boldsymbol{z})$ serves as a substitute for $p(\boldsymbol{y}|\boldsymbol{z})$ in Eq. (9). However, we argue that directly substituting $p(\boldsymbol{y}|\boldsymbol{z})$ with $q(\boldsymbol{y}|\boldsymbol{z})$ in Eq. (9) requires a more careful examination. Specifically, expanding Eq. (9) yields:

$$\begin{aligned} I(Z, Y) &\geq \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{z})] = \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[\log\left(q(\boldsymbol{y}|\boldsymbol{z})\,\frac{p(\boldsymbol{y}|\boldsymbol{z})}{q(\boldsymbol{y}|\boldsymbol{z})}\right)\right] \\ &= \underbrace{\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[\log q(\boldsymbol{y}|\boldsymbol{z})\right]}_{\text{(i)}} + \underbrace{\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{y}|\boldsymbol{z})}{q(\boldsymbol{y}|\boldsymbol{z})}\right]}_{\text{(ii)}}. \end{aligned} \tag{11}$$

[1] Alemi et al. (2017) initially derive the objective using the Markov chain $Z - X - Y$ but implement it with $X - Z - Y$.Wieczorek & Roth (2020) directly derive the lower bound from $X - Z - Y$.

Term (i) is used for training the model in Alemi et al. (2017), including both the encoder and decoder. Term (ii) represents the gap between term (i) and the true lower bound in Eq. (9). Crucially, **term (ii) is not necessarily non-negative**. If term (ii) is negative, it undermines the assumption that term (i) serves as a valid lower bound on $I(Z, Y)$. This calls into question its suitability as a training objective. Term (ii) remains small only when the model is well-trained so that $q(\boldsymbol{y}|\boldsymbol{z})$ closely approximates $p(\boldsymbol{y}|\boldsymbol{z})$. In summary, when introducing IB at an intermediate layer within an LLM and substituting $p(\boldsymbol{y}|\boldsymbol{z})$ with $q(\boldsymbol{y}|\boldsymbol{z})$, we lose control over whether the training objective truly remains a lower bound on $I(Z, Y)$. In the next section, we present our strategy to address this challenge.

3.2 Regularizing EDL with IB

So far we have not introduced how to apply IB to the EDL pipeline: $\boldsymbol{x} \rightarrow f(\boldsymbol{x}; \boldsymbol{\theta}) \rightarrow \tilde{\boldsymbol{e}} \rightarrow \boldsymbol{e} \rightarrow \boldsymbol{\alpha} \rightarrow \boldsymbol{\pi} \rightarrow \boldsymbol{y}$. In this section, we focus on two key questions: **(1) What should the hidden variable Z be in IB-EDL?** In other words, where in the EDL pipeline should we introduce IB? **(2) What prior distribution $r(\boldsymbol{z})$ should we select?**

Choice of hidden variable Z: As highlighted in Section 3.1, a key challenge in applying IB *within* a neural network is the potential violation of the lower bound. We address this by choosing the pre-evidence as the hidden variable Z, i.e., $\boldsymbol{z} = \tilde{\boldsymbol{e}} \in \mathbb{R}^C$. By doing so, we can *exactly evaluate* $p(\boldsymbol{y}|\boldsymbol{z})$ from the pipeline $\tilde{\boldsymbol{e}} \rightarrow \boldsymbol{e} \rightarrow \boldsymbol{\alpha} \rightarrow \boldsymbol{\pi} \rightarrow \boldsymbol{y}$. Specifically, given $\tilde{\boldsymbol{e}}$, we can compute $\boldsymbol{\alpha} = \boldsymbol{e} + \mathbf{1} = \text{SoftPlus}(\tilde{\boldsymbol{e}}) + \mathbf{1}$, from which $\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\pi}; \boldsymbol{\alpha})$ and $\boldsymbol{y} \sim \text{Cat}(\boldsymbol{y}; \boldsymbol{\pi})$ follow. As a result, there is no need to learn $q(\boldsymbol{y}|\boldsymbol{z})$, allowing us to directly use the lower bound in Eq. (9) (and hence Eq. (10)) as the training objective, which only involves $p(\boldsymbol{y}|\boldsymbol{z})$. In other words, by choosing $\boldsymbol{z} = \tilde{\boldsymbol{e}}$, we skip the step of learning an approximated distribution $q(\boldsymbol{y}|\boldsymbol{z})$ and ensure that we are maximizing a valid lower bound of $I(Z, Y)$. Next, we proceed to compute this bound, namely the first term in Eq. (10), which can be analytically computed as:

$$\begin{aligned}\mathcal{L}_{\text{IB-NLL}}(\boldsymbol{\theta}) &:= -\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{z})] \\ &= \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_{j=1}^{C} y_j(\log(\alpha_0) - \log(\alpha_j))\right],\end{aligned} \tag{12}$$

where we have omitted the dependency of $\boldsymbol{\alpha}$ on $\boldsymbol{z} = \tilde{\boldsymbol{e}}$. Furthermore, our method can be further enhanced by integrating the finding of Sensoy et al. (2018), suggesting the MSE loss as a practically more stable objective. This can be analytically computed from $\boldsymbol{\alpha}$ (and $\tilde{\boldsymbol{e}}$) as follows:

$$\begin{aligned}\mathcal{L}_{\text{IB-MSE}}(\boldsymbol{\theta}) &:= \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[||\boldsymbol{y} - \boldsymbol{\pi}||_2^2\right] \\ &= \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_{j=1}^{C}\left(y_j - \frac{\alpha_j}{\alpha_0}\right)^2 + \frac{\alpha_j(\alpha_0 - \alpha_j)}{\alpha_0^2(\alpha_j + 1)}\right].\end{aligned} \tag{13}$$

Choice of prior $r(\boldsymbol{z})$: Having defined $\boldsymbol{z} = \tilde{\boldsymbol{e}}$, we now require a suitable prior $r(\boldsymbol{z})$ for $\boldsymbol{z}$. Notably, $\tilde{\boldsymbol{e}}$ is the output of the LLM, and recent studies (Zhang et al., 2021; Hashemi et al., 2021) suggest that activation distributions in the later layers of neural networks tend to resemble Gaussian distributions more closely than those in earlier layers. Additionally, the pre-evidence $\tilde{\boldsymbol{e}}$ represents the LLM's output values, typically ranging from -2 to 2. Therefore, a standard Gaussian prior, $z_j \sim \mathcal{N}(0, 1)\ \forall j$, represents a reasonable choice.[2] Given that $\boldsymbol{z} = \tilde{\boldsymbol{e}}$ follows a Gaussian distribution, we leverage the LLM f to learn the mean and covariance of this $\boldsymbol{z}$'s distribution. Typically, an LLM comprises a sequence of transformer layers, denoted as g, followed by a linear head h, such that $f = h \circ g$. To model the Gaussian distribution, we double the number of output neurons in h, partitioning it into two equal-sized functional parts. One part, h^{μ}, predicts the Gaussian mean, while the other, h^{σ}, predicts the variances. Since h^{μ} and h^{σ} predictors share the same features from g, both predictions can be computed in a single forward pass. We define $\boldsymbol{\mu} = h^{\mu}(g(\boldsymbol{x}))$ and $\boldsymbol{\sigma} = \text{SoftPlus}(h^{\sigma}(g(\boldsymbol{x})))$, yielding $p(\boldsymbol{z}|\boldsymbol{x}; \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}))$. Then the second term in Eq. (10) becomes:

$$\mathcal{L}_{\text{IB-Info}}(\boldsymbol{\theta}) := \mathbb{E}_{p(\boldsymbol{x})}\left[D_{\text{KL}}\left(p(\boldsymbol{z}|\boldsymbol{x})\|r(\boldsymbol{z})\right)\right] \propto \frac{1}{2}\mathbb{E}_{p(\boldsymbol{x})}\left[\|\boldsymbol{\mu}\|_2^2 + \|\boldsymbol{\sigma}\|_2^2 - 2\sum_{j=1}^{C}\log(\sigma_j)\right]. \tag{14}$$

[2] Alternatively, we can also choose $\boldsymbol{z} = \boldsymbol{e} = \text{SoftPlus}(\tilde{\boldsymbol{e}})$; however, the prior will be a truncated Gaussian.

Algorithm 1 IB-EDL training and inference pseudocode.

Require: Data $(\boldsymbol{x}, \boldsymbol{y})$, LLM $f = h \circ g$, weight β, sample size K, binary flag `IsTraining`.
1: $\boldsymbol{\mu}, \boldsymbol{\sigma} \leftarrow (h \circ g)(\boldsymbol{x})$; and compute $\mathcal{L}_{\text{IB-Info}}(\boldsymbol{\theta})$ with $\boldsymbol{\mu}, \boldsymbol{\sigma}$. ▷ *Predict $p(\tilde{\boldsymbol{e}}|\boldsymbol{x})$ using the LLM.*
2: Draw $\{\tilde{\boldsymbol{e}}^{(k)}\}_{k=1}^{K}$ from $\mathcal{N}(\tilde{\boldsymbol{e}}; \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}))$. ▷ *Parallelized in PyTorch.*
3: **if** `IsTraining` **then** ▷ *At training time.*
4: Compute $\mathcal{L}_{\text{IB-MSE}}(\boldsymbol{\theta})$ for each $\tilde{\boldsymbol{e}}^{(k)}$ and take the **average**. ▷ *Eq. (13). Also parallelized.*
5: Backpropagate **averaged** $\mathcal{L}_{\text{IB-MSE}}(\boldsymbol{\theta}) + \beta\, \mathcal{L}_{\text{IB-Info}}(\boldsymbol{\theta})$.
6: **else** ▷ *At inference time.*
7: Compute **average** $\tilde{\boldsymbol{e}} \leftarrow \frac{1}{K}\sum_{k=1}^{K} \tilde{\boldsymbol{e}}^{(k)}$; and $\boldsymbol{\alpha} \leftarrow \text{SoftPlus}(\tilde{\boldsymbol{e}}) + 1$; and $\hat{\pi}_j \leftarrow \alpha_j/\alpha_0\ \forall j$.
8: Final prediction $\hat{y} \leftarrow \arg\max_j \hat{\pi}_j$. Uncertainty mass: $u \leftarrow C/\alpha_0$.

Overall IB-EDL loss and its interpretation: The final IB-EDL objective is:

$$\min_{\boldsymbol{\theta}}\ \mathcal{L}_{\text{IB-MSE}}(\boldsymbol{\theta}) + \beta\, \mathcal{L}_{\text{IB-Info}}(\boldsymbol{\theta}). \tag{15}$$

Importantly, Eq. (14) imposes a ℓ_2-regularization on $\boldsymbol{\mu}$, i.e. the mean of $\tilde{\boldsymbol{e}}$, and thus on $\boldsymbol{\alpha}$. *Therefore, IB-EDL penalizes the LLM for generating large $\boldsymbol{\alpha}$ that lead to over-confident predictions.*

3.3 Practical implementation

In this subsection, we focus on the implementation of IB-EDL, so *we use $\tilde{\boldsymbol{e}}$ instead of $\boldsymbol{z}$*. Eq. (13) requires sampling $\tilde{\boldsymbol{e}}$ from the predicted distribution $\mathcal{N}(\tilde{\boldsymbol{e}}; \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}))$ and using the sampled $\tilde{\boldsymbol{e}}$ to compute $\boldsymbol{\alpha}$. To handle the non-differentiable sampling operation, we apply the reparameterization trick (Kingma & Welling, 2014), allowing gradients to flow through the LLM parameters $\boldsymbol{\theta}$. For each input $\boldsymbol{x}$, we sample K (e.g. $K = 20$) pre-evidences $\tilde{\boldsymbol{e}}$ and compute the average loss derived from them (see Algorithm 1). At inference time, we also sample K values and compute the average $\tilde{\boldsymbol{e}}$. The extra time cost is minimal compared to the inference time of the LLM (detailed in Section 4.5).

3.4 Variational Bayes-based EDL as a special case of IB-EDL

Some EDL methods (Chen et al., 2018; Joo et al., 2020) are based on a variational Bayes (VB) perspective and can be seen as a special case within the IB-EDL framework. More formally:

Proposition 1. *The VB-based EDL methods, which minimize $\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\left[D_{KL}\left(p(\boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta})||p(\boldsymbol{\pi}|\boldsymbol{y})\right)\right]$, are a special case of IB-EDL when the hidden variable is chosen as $\boldsymbol{z} = \boldsymbol{\pi}$ (i.e. token probabilities) and the prior $r(\boldsymbol{z})$ is chosen as Dir $(\boldsymbol{z}; \boldsymbol{y} \odot \boldsymbol{\alpha} + (\boldsymbol{1} - \boldsymbol{y}))$.*

The proof is detailed in the Appendix B. In the experiments, we will also compare our method with VID (Chen et al., 2018), a representative EDL method from this category.

4 Experiments

4.1 Experimental setups

Models: We fine-tune Llama2-7B (Touvron et al., 2023), Llama3-8B (Dubey et al., 2024), and Mistral-7B (Jiang et al., 2023) using LoRA (Hu et al., 2022) implemented via PEFT (Mangrulkar et al., 2022) and Transformers (Wolf et al., 2020). Details on training configurations and β values are provided in Appendix C. Due to space constraints, we primarily present results for Llama2-7B and Llama3-8B, with Mistral-7B results provided in Appendix E.1.

Datasets: We compare methods on six multiple-choice classification datasets, including five for commonsense reasoning, ARC-C and ARC-E (Clark et al., 2018), OpenbookQA (OBQA) (Mihaylov et al., 2018), CommonsenseQA (CSQA) (Talmor et al., 2019), and SciQ (Welbl et al., 2017), alongside a dataset for reading comprehension, RACE (Lai et al., 2017). For these datasets, we define the target space $\mathcal{Y}$ as the tokens corresponding to the possible options (A/B/C/D). When fine-tuning LLMs, we select next-token logits (pre-evidences) corresponding to these options.

Baselines: We compare our method against a variety of baselines, including standard MAP training, two conventional uncertainty-aware approaches: Deep Ensemble (**Ens**) (Lakshminarayanan et al.,

Table 1: Calibration performance of uncertainty-aware methods on fine-tuned Llama2-7B. Arrows ("↑" or "↓") indicate whether higher or lower values signify better performance, respectively. The best and second-best results are highlighted in **bold** and underlined, respectively. Additionally, ± denotes the standard deviation of 3 runs. IB-EDL consistently achieves comparable accuracy while significantly reducing ECE and NLL, thereby greatly mitigating the overconfidence of the LLM.

Metrics	Method	ARC-C	ARC-E	OBQA	CSQA	SciQ	RACE
Acc ↑	MAP	68.00±1.72	85.07±0.11	79.86±0.12	78.67±0.26	91.30±0.17	81.29±0.29
	MCD	68.01±1.71	85.10±0.09	79.70±0.46	78.65±0.30	**91.33±0.23**	81.26±0.31
	Ens	67.46±0.21	85.16±0.25	79.72±0.61	78.84±0.16	90.83±0.60	81.26±0.18
	LA	66.92±0.55	84.60±0.44	80.15±0.22	78.58±0.50	90.87±0.15	81.19±0.33
	EDL	66.78±1.69	84.56±0.67	79.80±0.40	79.03±0.62	90.07±0.61	81.03±0.29
	VID	67.40±0.90	84.69±0.61	78.93±0.50	78.46±0.25	91.13±0.42	81.33±0.55
	I-EDL	66.98±0.37	84.76±0.69	**81.79±0.53**	**79.22±0.39**	90.43±0.70	78.06±0.32
	R-EDL	**70.37±0.47**	84.53±0.16	81.13±0.51	78.64±0.16	90.49±0.36	79.29±1.57
	IB-EDL	68.17±0.92	**85.67±0.76**	80.17±0.20	78.62±1.07	91.10±0.66	**81.82±0.30**
ECE ↓	MAP	30.08±0.96	13.66±0.14	17.81±0.14	19.06±0.25	6.85±0.21	9.62±0.32
	MCD	30.08±0.97	12.87±0.71	17.14±0.77	19.06±0.25	6.83±0.20	9.62±0.32
	Ens	28.26±2.78	12.64±0.72	15.19±0.94	17.82±0.17	6.77±0.33	9.51±0.43
	LA	8.80±3.77	25.01±1.32	10.71±0.16	11.81±1.56	15.51±2.79	9.62±0.19
	EDL	14.26±1.38	8.87±1.40	9.14±1.98	9.71±1.04	18.69±1.17	7.47±0.72
	VID	13.94±0.87	4.78±0.23	10.06±1.51	8.25±0.27	6.26±0.54	4.56±0.81
	I-EDL	12.39±0.16	11.33±0.27	10.84±1.18	11.66±0.61	15.13±0.08	13.52±0.85
	R-EDL	15.34±2.81	4.68±0.41	7.10±0.68	5.77±0.66	6.13±0.25	4.48±0.59
	IB-EDL	**6.38±0.56**	**2.57±0.57**	**5.84±0.88**	**4.90±1.08**	**5.01±0.43**	**4.03±0.18**
NLL ↓	MAP	2.98±0.11	1.09±0.04	1.14±0.03	1.26±0.01	0.38±0.01	0.61±0.01
	MCD	2.98±0.11	1.09±0.05	1.11±0.04	1.26±0.01	0.38±0.01	0.61±0.01
	Ens	2.90±0.19	1.08±0.03	1.05±0.05	1.12±0.06	0.39±0.02	0.59±0.01
	LA	**1.03±0.01**	0.70±0.01	**0.61±0.00**	0.70±0.03	0.37±0.03	**0.56±0.01**
	EDL	1.07±0.04	0.59±0.03	0.65±0.01	0.75±0.01	0.47±0.01	0.62±0.01
	VID	1.07±0.01	0.60±0.02	0.71±0.02	0.77±0.01	0.31±0.01	0.63±0.03
	I-EDL	1.08±0.01	0.60±0.02	0.62±0.01	0.75±0.01	0.46±0.01	0.71±0.01
	R-EDL	1.07±0.05	0.57±0.01	**0.61±0.01**	**0.74±0.01**	0.35±0.01	0.61±0.03
	IB-EDL	**1.03±0.02**	**0.53±0.03**	0.65±0.01	**0.74±0.02**	**0.29±0.01**	**0.50±0.01**

2017; Fort et al., 2019) and MC-Dropout (**MCD**) (Gal & Ghahramani, 2016), and Laplace-LoRA (**LA**) (Yang et al., 2024), a recent calibration method tailored for fine-tuned LLMs. Additionally, we include four baselines from the EDL family: vanilla **EDL** (Sensoy et al., 2018), **VID** (Chen et al., 2018) from VB-based EDL, and two SOTA methods: **I-EDL** (Deng et al., 2023) and **R-EDL** (Chen et al., 2024). We use their original implementations and hyperparameters where available. Additionally, although PostNet (Charpentier et al., 2020) is also a well-known EDL method, it is omitted here because it requires a specialized Normalizing Flow design to be compatible with Transformers.

4.2 In-distribution calibration

An effective uncertainty-aware method should 1) significantly improve model calibration and 2) show accuracy comparable to standard MAP training. We therefore use Accuracy (Acc), expected calibration error (ECE), and negative log-likelihood (NLL) as metrics for evaluating the fine-tuned LLMs on the six aforementioned datasets. Table 1 and Table 2 present the results for Llama2-7B and Llama3-8B, respectively. The accuracy of IB-EDL and other uncertainty-aware methods is on par with, or even higher than, the MAP baseline, so we focus primarily on analyzing ECE and NLL. MAP exhibits substantially higher ECE and NLL than other methods, suggesting that fine-tuning LLMs on small datasets using MAP (or MLE) leads to significant overconfidence. Overall, IB-EDL shows the lowest ECE and NLL, reducing ECE by several factors compared to MAP, MCD, and Ens. This highlights IB-EDL's ability to effectively mitigate overconfidence in fine-tuned models. The superior performance of IB-EDL, compared to other EDL methods, can be attributed to the ℓ_2 regularization in Eq. (14), which discourages the model from generating excessively large evidences that lead to over-concentrated Dirichlet distributions. Furthermore, the advantages of IB-EDL extend to other architectures, such as Mistral-7B, as demonstrated in Appendix E.1.

Table 2: Calibration performance of uncertainty-aware methods on fine-tuned Llama3-8B.

Metrics	Method	ARC-C	ARC-E	OBQA	CSQA	SciQ	RACE
Acc ↑	MAP	79.74±0.27	92.27±0.17	88.60±0.87	81.51±0.60	93.37±0.12	88.15±0.31
	MCD	79.55±0.21	92.25±0.13	88.63±0.84	81.52±0.60	93.30±0.20	88.12±0.25
	Ens	79.50±0.10	92.26±0.41	88.53±0.59	81.39±0.63	93.27±0.15	88.09±0.08
	LA	77.79±0.39	92.18±0.24	88.34±0.62	81.30±0.44	93.37±0.15	88.09±0.19
	EDL	79.35±1.11	92.31±0.76	87.67±0.31	80.67±0.29	93.13±0.36	87.76±0.21
	VID	79.99±0.13	92.30±0.25	87.57±0.29	80.82±0.94	92.93±0.23	**88.23±0.25**
	I-EDL	80.37±0.48	**92.76±0.47**	88.52±0.50	81.13±0.39	93.47±0.06	85.91±0.40
	R-EDL	80.60±0.93	92.41±0.30	88.00±0.20	80.83±1.00	93.33±0.21	87.62±0.16
	IB-EDL	**81.14±0.09**	92.55±0.15	**89.00±0.40**	**81.71±0.38**	**93.57±0.15**	88.03±0.21
ECE ↓	MAP	19.68±0.43	7.18±0.14	10.52±0.87	17.29±0.57	5.74±0.08	7.95±0.35
	MCD	19.91±0.39	7.10±0.02	10.48±0.86	16.98±0.10	5.74±0.09	7.93±0.35
	Ens	18.20±0.17	3.81±1.60	10.08±0.90	16.11±1.63	5.72±0.24	7.80±0.19
	LA	18.49±0.44	3.33±0.66	5.26±1.30	6.62±0.10	**2.47±0.08**	3.61±0.27
	EDL	6.52±0.12	5.94±0.87	8.28±1.62	7.43±1.48	11.13±1.11	6.51±0.80
	VID	10.96±0.40	3.33±1.40	5.99±1.41	8.38±0.93	2.60±0.13	4.58±0.14
	I-EDL	5.08±1.94	9.69±0.58	7.57±0.52	8.95±0.59	13.07±0.24	14.52±1.15
	R-EDL	10.09±1.01	2.93±1.32	4.68±1.35	6.59±0.78	3.18±0.18	**2.61±0.09**
	IB-EDL	**2.78±0.87**	**2.70±0.58**	**2.34±0.61**	**4.34±0.20**	3.86±0.86	4.47±0.31
NLL ↓	MAP	2.25±0.08	0.61±0.02	0.78±0.05	1.25±0.04	0.36±0.00	0.47±0.01
	MCD	2.27±0.09	0.59±0.00	0.77±0.05	1.22±0.03	0.36±0.01	0.45±0.03
	Ens	2.02±0.07	0.43±0.09	0.74±0.05	1.22±0.12	0.35±0.01	0.46±0.05
	LA	0.80±0.01	0.33±0.04	0.42±0.01	**0.62±0.04**	**0.22±0.00**	0.37±0.01
	EDL	0.74±0.02	0.33±0.01	0.45±0.02	0.68±0.01	0.31±0.01	0.43±0.01
	VID	0.78±0.01	0.35±0.01	0.46±0.02	0.72±0.03	0.28±0.01	0.36±0.00
	I-EDL	0.72±0.02	0.36±0.02	0.43±0.00	0.68±0.01	0.32±0.01	0.52±0.02
	R-EDL	0.74±0.01	**0.32±0.01**	0.43±0.02	0.68±0.02	0.27±0.01	0.43±0.02
	IB-EDL	**0.69±0.01**	**0.32±0.02**	**0.40±0.03**	0.66±0.01	0.25±0.01	**0.35±0.00**

Table 3: OOD detection performance on fine-tuned Llama2-7B and Llama3-8B. $A \rightarrow B$ indicates A as the ID training set and B as the OOD test set. MP and UM are two scores for measuring AUROC.

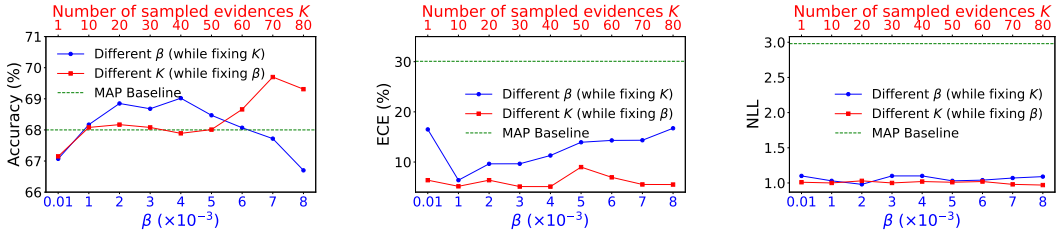
Model	Method	OBQA → ARC-C		OBQA → ARC-E		OBQA → CSQA	
		AUROC ↑		AUROC ↑		AUROC ↑	
		MP	UM	MP	UM	MP	UM
Llama2-7B	MAP	76.07±0.71	–	72.21±0.60	–	72.45±0.20	–
	MCD	76.07±0.71	–	72.20±0.60	–	72.49±0.17	–
	Ens	71.89±2.79	–	70.30±0.43	–	70.50±1.12	–
	LA	72.85±0.86	–	67.46±0.87	–	70.71±0.40	–
	EDL	68.42±1.72	74.89±1.12	78.34±0.45	74.75±0.59	76.46±1.71	72.56±1.99
	VID	86.47±0.26	81.24±0.81	86.18±0.79	83.14±1.54	85.41±0.78	77.27±2.06
	I-EDL	81.34±1.41	75.78±0.94	77.22±0.75	72.10±1.74	75.78±0.75	71.60±1.11
	R-EDL	76.66±1.07	73.80±0.74	72.01±1.28	69.02±0.91	71.01±1.47	68.22±1.43
	IB-EDL	**87.47±0.91**	**88.34±3.07**	**86.42±0.87**	**88.52±2.90**	**86.38±0.53**	**79.79±4.64**
Llama3-8B	MAP	63.12±1.21	–	58.11±1.55	–	64.43±1.63	–
	MCD	62.81±1.00	–	58.30±1.98	–	64.05±2.09	–
	Ens	62.70±1.04	–	58.19±1.72	–	63.79±0.82	–
	LA	62.14±0.55	–	56.11±0.84	–	63.25±1.06	–
	EDL	82.04±0.69	78.99±1.18	78.80±1.47	74.77±2.16	81.07±1.22	77.55±1.69
	VID	**88.85±1.57**	89.95±1.59	87.55±0.94	90.02±1.72	88.66±1.75	84.37±0.20
	I-EDL	81.31±0.52	78.54±0.61	78.29±1.14	74.79±1.43	77.85±3.29	73.80±4.23
	R-EDL	75.79±0.51	73.64±0.49	71.51±0.79	68.81±0.79	70.60±0.82	67.42±1.30
	IB-EDL	**88.85±0.96**	**92.58±0.37**	**88.14±1.10**	**94.77±0.42**	**89.16±1.01**	**85.45±0.55**

4.3 Out-of-distribution detection

In addition to in-distribution (ID) calibration, out-of-distribution (OOD) detection serves as a key benchmark for assessing the performance of uncertainty-aware methods. An effective approach should reliably assign higher uncertainties to OOD samples compared to ID samples. This can

Table 4: Fine-tuning Llama2-7B and Llama3-8B on noisy datasets. In each training dataset, the labels of 30% samples are randomly perturbed. IB-EDL is more robust to noise than other baselines.

Metric	Method	ARC-C	ARC-E	OBQA	CSQA	SciQ	RACE
Llama2-7B Acc ↑	MAP	51.08±2.96	71.39±2.73	73.93±2.21	74.56±0.31	88.63±0.06	76.83±0.39
	MCD	51.08±2.96	71.40±2.72	73.93±2.20	74.56±0.32	88.64±0.06	76.95±0.19
	Ens	56.10±3.11	76.03±1.99	75.23±1.06	74.59±0.23	88.64±0.06	76.94±0.22
	LA	53.38±2.17	74.90±1.76	74.57±2.22	74.59±0.20	88.47±0.21	77.04±0.39
	EDL	57.16±3.33	76.16±1.12	74.46±1.33	74.65±0.05	89.03±0.25	77.69±0.44
	VID	57.56±0.58	76.57±1.79	76.67±1.42	75.97±0.66	89.06±0.21	78.75±0.25
	I-EDL	53.86±1.25	76.08±0.71	77.00±0.88	74.01±1.28	**89.26±0.32**	76.48±0.87
	R-EDL	57.00±0.76	76.96±1.07	73.20±1.44	74.36±0.49	87.33±0.49	78.01±0.53
	IB-EDL	**59.06±2.07**	**80.27±0.48**	**78.13±0.99**	**76.35±0.66**	89.06±0.50	**79.41±0.59**
Llama3-8B Acc ↑	MAP	57.71±0.42	80.34±1.47	78.78±1.00	77.04±0.41	92.60±0.53	86.93±0.11
	MCD	57.71±0.42	80.37±1.46	79.00±0.92	77.05±0.41	92.87±0.12	86.93±0.12
	Ens	63.39±1.09	84.63±0.53	80.61±0.53	77.62±0.41	92.97±0.06	86.93±0.07
	LA	66.62±1.08	82.64±0.50	79.59±0.72	77.80±0.49	92.93±0.21	87.00±0.01
	EDL	69.43±0.98	87.57±0.13	85.60±0.72	79.14±0.41	92.90±0.40	86.26±0.76
	VID	66.58±1.92	87.67±0.99	84.86±1.01	**79.66±1.26**	93.23±0.25	86.86±0.26
	I-EDL	63.87±2.65	84.06±2.80	84.26±0.42	78.02±0.87	92.53±0.37	86.01±0.51
	R-EDL	**71.25±1.20**	84.91±3.91	85.73±0.70	78.57±0.21	**93.56±0.05**	86.16±0.42
	IB-EDL	68.53±0.25	**88.05±0.43**	**86.13±0.51**	79.59±0.79	93.46±0.38	**87.01±0.20**



(a) Accuracy (b) ECE (c) NLL

Figure 1: Ablation study. IB-EDL reduces ECE and NLL compared to MAP across a broad range of β and K values. β controls the regularization strength and balances the calibration and accuracy.

be evaluated by labeling ID samples as class 1 and OOD samples as class 0, and measuring the **AUROC** based on OOD detection scores derived from the fine-tuned model. A higher AUROC indicates better OOD detection performance. Similar to Chen et al. (2024), we use two OOD detection scores: *max probability* (**MP**) and the *reciprocal of uncertainty mass* (**UM**). We fine-tune the LLMs on OBQA (as the ID dataset) and test them on ARC-C, ARC-E, and CSQA (as OOD dataset). Note that non-EDL methods, such as LA, do not provide UM, so we evaluate them using MP only. As shown in Table 3, IB-EDL achieves the highest AUROC across all datasets using both scores, surpassing both non-EDL and EDL competitors. Furthermore, IB-EDL also demonstrates superior OOD detection performance under large distribution shifts (see Appendix E.3). Its calibration performance also generalizes well to OOD datasets (see Appendix E.4).

4.4 Fine-tuning with noisy labels

Datasets used for fine-tuning LLMs can contain a significant portion of mislabeled samples (Wang et al., 2024b; Havrilla & Iyer, 2024), and label verification is often expert-knowledge demanding. Therefore, it is crucial for fine-tuning algorithms to be robust to label noise (Wang et al., 2023a). To assess this robustness, we perturb the A/B/C/D options for 30% of the samples in each training set, fine-tune the models using the aforementioned methods on the noisy datasets, and evaluate them on clean test sets. In this adversarial setting, the primary goal is to maintain accuracy despite label noise, so we primarily evaluate accuracy, with additional metrics in the Appendix E.2. As shown in Table 4, IB-EDL achieves the highest accuracy overall, demonstrating strong robustness to label noise. This suggests that the information bottleneck effectively filters out spurious signals and retains predictive information in the generated evidence.

4.5 ABLATION STUDY

Hyperparameters: We study the weight β and sample size K (used for drawing $\tilde{\boldsymbol{e}}$), using ID calibration for Llama2-7B on ARC-C as the evaluation task. Fig. 1 shows that IB-EDL consistently outperforms MAP across a broad range of values, demonstrating its robustness in reducing overconfidence. While increasing K slightly improves accuracy, it does not necessarily improve ECE or NLL. β plays a key role in balancing regularization and predictive performance. Nonetheless, all β values reduce ECE by at least 40% and NLL by at least 50% compared to MAP. Additionally, we also present a sensitivity analysis on the number bins of ECE in Appendix E.6.

Complexity analysis: Here, we consider Llama2-7B and assume the target space is the vocabulary $\mathcal{Y} = \mathcal{V}$. Compared to the pretrained LLM, IB-EDL introduces an additional linear head h^{σ}, adding only 1.95% more parameters. The computational overhead stems from h^{σ} and the evidence averaging operation (see Algorithm 1), which amount to only 1.98% of the pretrained model's GFLOPs. In Appendix E.5, we provide detailed tests of training and inference time as well as memory usage.

5 RELATED WORK

EDL: EDL leverages models to predict the Dirichlet prior distribution, with training typically done using NLL (Haussmann et al., 2023), ℓ_p-loss (Tsiligkaridis, 2021), or MSE (Sensoy et al., 2018). Chen et al. (2018); Joo et al. (2020); Shen et al. (2023) derive loss functions from a variational Bayesian perspective, while Posterior Networks (Charpentier et al., 2020; 2022) optimize the posterior via Normalizing Flows. EDL methods often incorporate two main types of regularization: (i) encouraging uniformity in non-target Dirichlet parameters (Malinin & Gales, 2018; Sensoy et al., 2018; Chen et al., 2018; Tsiligkaridis, 2021), or (ii) modifying assumptions in the EDL formulation (Deng et al., 2023; Chen et al., 2024). IB-EDL differs by not requiring (i) and taking an alternative approach to (ii), without altering EDL assumptions. Some EDL methods also incorporate OOD data during training (Malinin & Gales, 2018; 2019). Beyond classification, EDL has been extended to regression (Amini et al., 2020) and other applications (Gao et al., 2024; Liu & Ji, 2024). Recent works (Shen et al., 2024; Juergens et al., 2024) analyze EDL's effectiveness and limitations.

Other uncertainty-aware methods: Besides EDL, there are other methods for uncertainty estimation and calibration, including Bayesian Neural Networks via Variational Inference (Graves, 2011; Blundell et al., 2015), MC-Dropout (Gal & Ghahramani, 2016), stochastic gradient MCMC (Welling & Teh, 2011; Ma et al., 2015), and Laplace approximations (Ritter et al., 2018; Kristiadi et al., 2021), recently extended to LoRA fine-tuned LLMs (Yang et al., 2024; Wang et al., 2024a; Li et al., 2024).

IB: The Information Bottleneck (IB) was introduced by Tishby et al. (1999) and later applied in neural networks for learning generalized representations (Tishby & Zaslavsky, 2015; Alemi et al., 2017; Sun et al., 2022), and as a feature attribution method (Schulz et al., 2020; Zhang et al., 2021; Wang et al., 2023b). Other works focusing on theory studied different Markov chains in IB (Wieczorek & Roth, 2020) and the impact of IB on generalization errors (Kawaguchi et al., 2023).

6 CONCLUSION

Summary: We focused on a key challenge in fine-tuning LLMs: mitigating overconfidence and improving calibration. We introduced an information-theoretic regularization to conventional EDL to prevent over-concentrated distributions in predictions. Our method, IB-EDL, introduces minimal computational overhead while significantly improving calibration in fine-tuned LLMs. Additionally, IB-EDL maintains model performance even in the presence of substantial label noise. These results highlight IB-EDL as a promising method for fostering more trustworthy LLMs.

Limitations and future work: Despite IB-EDL's efficiency, there is room for improvement. To reduce the complexity of covariance matrix prediction, we assume the pre-evidences are uncorrelated, but this assumption can be relaxed. Additionally, our evaluations primarily focused on conventional classification tasks, where well-established metrics for calibration and uncertainty estimation are available. In future work, it would be interesting to test IB-EDL on generative tasks. A great challenge is that uncertainty estimation metrics for generative tasks are still an ongoing research topic (Yadkori et al., 2024; Jesson et al., 2024).

ACKNOWLEDGMENT

This work is supported by the Munich Center for Machine Learning (MCML). In addition, we sincerely appreciate the insightful discussions with Emanuel Sommer and Jianfei Li.

REFERENCES

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in neural information processing systems*, 33:14927–14937, 2020.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. *Advances in neural information processing systems*, 33:1356–1367, 2020.

Bertrand Charpentier, Oliver Borchert, Daniel Zügner, Simon Geisler, and Stephan Günnemann. Natural posterior network: Deep bayesian predictive uncertainty for exponential family distributions. In *International Conference on Learning Representations*, 2022.

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.

Wenhu Chen, Yilin Shen, Hongxia Jin, and William Wang. A variational dirichlet framework for out-of-distribution detection. *arXiv preprint arXiv:1811.07308*, 2018.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Arthur P Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(2):205–232, 1968.

Danruo Deng, Guangyong Chen, Yang Yu, Furui Liu, and Pheng-Ann Heng. Uncertainty estimation by fisher information-based evidential deep learning. In *International Conference on Machine Learning*, pp. 7596–7616. PMLR, 2023.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo,

Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang

Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Yibo Gao, Zheyao Gao, Xin Gao, Yuanye Liu, Bomin Wang, and Xiahai Zhuang. Evidential concept embedding models: Towards reliable concept explanations for skin disease diagnosis. *arXiv preprint arXiv:2406.19130*, 2024.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

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

Vahid Hashemi, Jan Křetínskỳ, Stefanie Mohr, and Emmanouil Seferis. Gaussian-based runtime detection of out-of-distribution inputs for neural networks. In *International Conference on Runtime Verification*, pp. 254–264. Springer, 2021.

Manuel Haussmann, Sebastian Gerwinn, and Melih Kandemir. Bayesian evidential deep learning with pac regularization. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2023.

Alex Havrilla and Maia Iyer. Understanding the effect of noise in llm training data with algorithmic chains of thought. *arXiv preprint arXiv:2402.04004*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021a.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021b.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Andrew Jesson, Nicolas Beltran-Velez, Quentin Chu, Sweta Karlekar, Jannik Kossen, Yarin Gal, John P Cunningham, and David Blei. Estimating the hallucination rate of generative ai. *arXiv preprint arXiv:2406.07457*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021.

Taejong Joo, Uijung Chung, and Min-Gwan Seo. Being bayesian about categorical probability. In *International conference on machine learning*, pp. 4950–4961. PMLR, 2020.

Audun Jøsang. Artificial reasoning with subjective logic. In *Proceedings of the second Australian workshop on commonsense reasoning*, volume 48, pp. 34, 1997.

Audun Jøsang. *Subjective logic*, volume 3. Springer, 2016.

Mira Juergens, Nis Meinert, Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. Is epistemic uncertainty faithfully represented by evidential deep learning methods? In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=mxjB0LIgpT.

Kenji Kawaguchi, Zhun Deng, Xu Ji, and Jiaoyang Huang. How does information bottleneck help deep learning? In *International Conference on Machine Learning*, pp. 16049–16096. PMLR, 2023.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Learnable uncertainty under laplace approximations. In *Uncertainty in Artificial Intelligence*, pp. 344–353. PMLR, 2021.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794. Association for Computational Linguistics, September 2017. URL https://aclanthology.org/D17-1082.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. Mixlora: Enhancing large language models fine-tuning with lora-based mixture of experts, 2024.

Pei Liu and Luping Ji. Weakly-supervised residual evidential learning for multi-instance uncertainty estimation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 31262–31292. Proceedings of Machine Learning Research, 2024. URL https://proceedings.mlr.press/v235/liu24ac.html.

Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. *Advances in neural information processing systems*, 28, 2015.

Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31, 2018.

Andrey Malinin and Mark Gales. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. *Advances in neural information processing systems*, 32, 2019.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and B Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. *URL: https://github.com/huggingface/peft*, 2022.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018.

Daniel P. Palomar and Sergio Verdu. Lautum information. *IEEE Transactions on Information Theory*, 54(3):964–975, 2008.

Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *6th international conference on learning representations, ICLR 2018-conference track proceedings*, volume 6. International Conference on Representation Learning, 2018.

Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for attribution. In *International Conference on Learning Representations*, 2020.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

G Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

Maohao Shen, Yuheng Bu, Prasanna Sattigeri, Soumya Ghosh, Subhro Das, and Gregory Wornell. Post-hoc uncertainty learning using a dirichlet meta-model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9772–9781, 2023.

Maohao Shen, J. Jon Ryu, Soumya Ghosh, Yuheng Bu, Prasanna Sattigeri, Subhro Das, and Gregory W. Wornell. Are uncertainty quantification capabilities of evidential deep learning a mirage?, 2024. URL https://arxiv.org/abs/2402.06160.

Qingyun Sun, Jianxin Li, Hao Peng, Jia Wu, Xingcheng Fu, Cheng Ji, and S Yu Philip. Graph structure learning with variational information bottleneck. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 4165–4174, 2022.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5, 2015.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377, 1999.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Theodoros Tsiligkaridis. Information robust dirichlet networks for predictive uncertainty estimation, April 8 2021. US Patent App. 17/064,046.

Song Wang, Zhen Tan, Ruocheng Guo, and Jundong Li. Noise-robust fine-tuning of pretrained language models via external guidance. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 12528–12540, 2023a.

Yibin Wang, Haizhou Shi, Ligong Han, Dimitris Metaxas, and Hao Wang. Blob: Bayesian low-rank adaptation by backpropagation for large language models. *arXiv preprint arXiv:2406.11675*, 2024a.

Ying Wang, Tim GJ Rudner, and Andrew G Wilson. Visual explanations of image-text representations via multi-modal information bottleneck attribution. *Advances in Neural Information Processing Systems*, 36:16009–16027, 2023b.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024b.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106. Association for Computational Linguistics, September 2017. URL https://aclanthology.org/W17-4413.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688. Citeseer, 2011.

Aleksander Wieczorek and Volker Roth. On the difference between the information bottleneck and the deep information bottleneck. *Entropy*, 22(2):131, 2020.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. To believe or not to believe your llm. *arXiv preprint arXiv:2406.02543*, 2024.

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*, 2024.

Yang Zhang, Ashkan Khakzar, Yawei Li, Azade Farshad, Seong Tae Kim, and Nassir Navab. Fine-grained neural network explanation by identifying input features with predictive information. *Advances in Neural Information Processing Systems*, 34:20040–20051, 2021.

A DERIVATION OF THE VARIATIONAL BOUNDS

Upper bound of $I(Z, X)$: We reproduce the derivation steps from Alemi et al. (2017) as follows:

$$I(Z,X) = \int p(\boldsymbol{x},\boldsymbol{z}) \log \frac{p(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})} \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{x} = \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{x})}[\log p(\boldsymbol{z}|\boldsymbol{x})] - \mathbb{E}_{p(\boldsymbol{z})}[\log p(\boldsymbol{z})]. \tag{16}$$

Given a prior $r(\boldsymbol{z})$, we have $D_{\mathrm{KL}}\left(p(\boldsymbol{z}) \| r(\boldsymbol{z})\right) \geq 0$. This indicates that:

$$\begin{aligned} D_{\mathrm{KL}}\left(p(\boldsymbol{z}) \| r(\boldsymbol{z})\right) &= \int p(\boldsymbol{z}) \log \frac{p(\boldsymbol{z})}{r(\boldsymbol{z})} \mathrm{d}\boldsymbol{z} \\ &= \mathbb{E}_{p(\boldsymbol{z})}[\log p(\boldsymbol{z})] - \mathbb{E}_{p(\boldsymbol{z})}[\log r(\boldsymbol{z})] \geq 0. \end{aligned}$$

Therefore,

$$\mathbb{E}_{p(\boldsymbol{z})}[\log p(\boldsymbol{z})] \geq \mathbb{E}_{p(\boldsymbol{z})}[\log r(\boldsymbol{z})]. \tag{17}$$

Plugging Eq. (17) into Eq. (16), we obtain

$$\begin{aligned} I(Z,X) &= \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{x})}[\log p(\boldsymbol{z}|\boldsymbol{x})] - \mathbb{E}_{p(\boldsymbol{z})}[\log p(\boldsymbol{z})] \\ &\leq \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{x})}[\log p(\boldsymbol{z}|\boldsymbol{x})] - \mathbb{E}_{p(\boldsymbol{z})}[\log r(\boldsymbol{z})] \\ &\leq \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{x})}[\log p(\boldsymbol{z}|\boldsymbol{x})] - \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[\log r(\boldsymbol{z})\right] \\ &= \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z}|\boldsymbol{x})}{r(\boldsymbol{z})}\right] \\ &= \mathbb{E}_{p(\boldsymbol{x})}\left[D_{\mathrm{KL}}\left(p(\boldsymbol{z}|\boldsymbol{x}) \| r(\boldsymbol{z})\right)\right]. \end{aligned} \tag{18}$$

Lower bound of $I(Z, Y)$: For clarity, we reproduce the derivation steps from Wieczorek & Roth (2020) here. Unlike Alemi et al. (2017), who assume the Markov chain $Z - X - Y$, Wieczorek & Roth (2020) assume the Markov chain $X - Z - Y$, implying the conditional independence $p(\boldsymbol{y}|\boldsymbol{z}) = p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{x})$. The detailed steps are as follows:

$$\begin{aligned} I(Z,Y) &= \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\log p(\boldsymbol{y}|\boldsymbol{z})] + H(Y) \\ &= \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{x})] + H(Y) \end{aligned} \tag{19}$$

Now, we derive a lower bound on the term $\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{x})]$ in Eq. (19) as follows:

$$\begin{aligned} &\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{x})] \\ &= \int\int p(\boldsymbol{z},\boldsymbol{y}|\boldsymbol{x}) \log p(\boldsymbol{z},\boldsymbol{y}|\boldsymbol{x}) \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{y} \\ &= \int\int p(\boldsymbol{z},\boldsymbol{y}|\boldsymbol{x}) \log \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z},\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})} \mathrm{d}\boldsymbol{y}\mathrm{d}\boldsymbol{z} \\ &= D_{\mathrm{KL}}\left(p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})\right) + \int\int p(\boldsymbol{z},\boldsymbol{y}|\boldsymbol{x}) \log p(\boldsymbol{y}|\boldsymbol{x}) \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{y} \\ &= D_{\mathrm{KL}}\left(p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})\right) + \int p(\boldsymbol{y}|\boldsymbol{x}) \log p(\boldsymbol{y}|\boldsymbol{x}) \mathrm{d}\boldsymbol{y} \\ &= D_{\mathrm{KL}}\left(p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})\right) + \int\int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{z}|\boldsymbol{x}) \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{y} \\ &= D_{\mathrm{KL}}\left(p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})\right) + \int\int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z},\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{z},\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})} \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{y} \\ &= D_{\mathrm{KL}}\left(p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})\right) + D_{\mathrm{KL}}\left(p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x})\right) \\ &\quad + \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{x})] \\ &\geq \mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{x})]. \end{aligned} \tag{20}$$

Plugging Eq. (20) into Eq. (19) and using $p(\boldsymbol{y}|\boldsymbol{z}) = p(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{x})$ again, we obtain:

$$
\begin{aligned}
I(Z,Y) &= \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{y})}[\log p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{x})] + H(Y) \\
&= \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{z})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{z})] \\
&\quad + \mathbb{E}_{p(\boldsymbol{x})}\left[D_{\mathrm{KL}}\left(p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})\right)\right] \\
&\quad + \mathbb{E}_{p(\boldsymbol{x})}\left[D_{\mathrm{KL}}\left(p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{y},\boldsymbol{z}|\boldsymbol{x})\right)\right] + H(Y) \\
&= \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{z})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{z})] \\
&\quad + I(Y,Z|X) + L(Y,Z|X) + H(Y) \\
&\geq \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{z})] + H(Y) \\
&\geq \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{y}|\boldsymbol{z})],
\end{aligned}
\tag{21}
$$

where $I(Y,Z|X)$ is the conditional mutual information, and $L(Y,Z|X)$ is the conditional lautum information (Palomar & Verdu, 2008), respectively.

B PROOF OF PROPOSITION 1

A brief introduction to VB-based EDL methods: Some previous EDL methods (Chen et al., 2018; Joo et al., 2020) derive the optimization objective from the variational Bayes (VB) perspective, where the goal is to optimize the model's parameters $\boldsymbol{\theta}$ such that the *posterior* distribution $p(\boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta})$ aligns with the *true* posterior distribution $p(\boldsymbol{\pi}|\boldsymbol{y})$.

For notation simplicity, we omit the model's parameters $\boldsymbol{\theta}$ in $p(\boldsymbol{\pi}|\boldsymbol{x};\boldsymbol{\theta})$ (or $p(\boldsymbol{z}|\boldsymbol{x};\boldsymbol{\theta})$) and use $p(\boldsymbol{\pi}|\boldsymbol{x})$ (or $p(\boldsymbol{z}|\boldsymbol{x})$) instead.

Remark 1. *The condition in Proposition 1 is that the latent variable $\boldsymbol{z}=\boldsymbol{\pi}$, and the prior $r(\boldsymbol{z})=r(\boldsymbol{\pi})=\mathit{Dir}\left(\boldsymbol{\pi};\boldsymbol{y}\odot\boldsymbol{\alpha}+(\mathbf{1}-\boldsymbol{y})\right)$. In fact, the choice of the prior $r(\boldsymbol{\pi})$ is not unique. For example, Chen et al. (2018) present three options. The correctness of Proposition 1 remains unaffected by the choice of prior. Proposition 1 uses one exemplary prior $r(\boldsymbol{\pi})=\mathit{Dir}\left(\boldsymbol{\pi};\boldsymbol{y}\odot\boldsymbol{\alpha}+(\mathbf{1}-\boldsymbol{y})\right)$ suggested by Chen et al. (2018).*

Proof. The target of VB-based EDL methods is:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[D_{\mathrm{KL}}\left(p(\boldsymbol{\pi}|\boldsymbol{x})\|p(\boldsymbol{\pi}|\boldsymbol{y})\right)\right]. \tag{22}$$

Next, we show that the IB objective in Eq. (10) is an upper bound of Eq. (22) when $\boldsymbol{z}=\boldsymbol{\pi}$ and $r(\boldsymbol{z})=r(\boldsymbol{\pi})$ for a given prior $r(\boldsymbol{\pi})$, e.g., $r(\boldsymbol{\pi})=\mathrm{Dir}\left(\boldsymbol{\pi};\boldsymbol{y}\odot\boldsymbol{\alpha}+(\mathbf{1}-\boldsymbol{y})\right)$. Minimizing the IB objective in Eq. (10) therefore provides a tractable way to approximate the minimization of Eq. (22).

If we choose $\beta=1$, $\boldsymbol{z}=\boldsymbol{\pi}$ and a prior $r(\boldsymbol{z})=r(\boldsymbol{\pi})$, then the IB objective in Eq. (10) becomes

$$\min_{\boldsymbol{\theta}} -\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{\pi}|\boldsymbol{x})}\left[\log p(\boldsymbol{y}|\boldsymbol{\pi})\right]+\mathbb{E}_{p(\boldsymbol{x})}\left[D_{\mathrm{KL}}\left(p(\boldsymbol{\pi}|\boldsymbol{x})\|r(\boldsymbol{\pi})\right)\right]. \tag{23}$$

Expanding Eq. (23), we have:

$$\begin{aligned}
&\mathbb{E}_{p(\boldsymbol{x})}\left[D_{\mathrm{KL}}\left(p(\boldsymbol{\pi}|\boldsymbol{x})\|r(\boldsymbol{\pi})\right)\right]-\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{\pi}|\boldsymbol{x})}\left[\log p(\boldsymbol{y}|\boldsymbol{\pi})\right]\\
&=\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\left[D_{\mathrm{KL}}\left(p(\boldsymbol{\pi}|\boldsymbol{x})\|r(\boldsymbol{\pi})\right)\right]-\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{\pi}|\boldsymbol{x})}\left[\log p(\boldsymbol{y}|\boldsymbol{\pi})\right]\\
&=\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\int p(\boldsymbol{\pi}|\boldsymbol{x})\log\frac{p(\boldsymbol{\pi}|\boldsymbol{x})}{r(\boldsymbol{\pi})}\mathrm{d}\boldsymbol{\pi}\right]-\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\int p(\boldsymbol{\pi}|\boldsymbol{x})\log p(\boldsymbol{y}|\boldsymbol{\pi})\mathrm{d}\boldsymbol{\pi}\right]\\
&\geq\underbrace{\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\int p(\boldsymbol{\pi}|\boldsymbol{x})\log\frac{p(\boldsymbol{\pi}|\boldsymbol{x})}{p(\boldsymbol{\pi})}\mathrm{d}\boldsymbol{\pi}\right]}_{\text{Use } D_{\mathrm{KL}}(p(\boldsymbol{\pi})\|r(\boldsymbol{\pi}))\geq 0 \text{ (Similar to Eq. (18))}}-\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\int p(\boldsymbol{\pi}|\boldsymbol{x})\log p(\boldsymbol{y}|\boldsymbol{\pi})\mathrm{d}\boldsymbol{\pi}\right]\\
&\geq\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\int p(\boldsymbol{\pi}|\boldsymbol{x})\log\frac{p(\boldsymbol{\pi}|\boldsymbol{x})}{p(\boldsymbol{\pi})}\mathrm{d}\boldsymbol{\pi}\right]-\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\int p(\boldsymbol{\pi}|\boldsymbol{x})\log p(\boldsymbol{y}|\boldsymbol{\pi})\mathrm{d}\boldsymbol{\pi}\right]\underbrace{-H(Y)}_{\leq 0}\\
&=\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\int p(\boldsymbol{\pi}|\boldsymbol{x})\log\frac{p(\boldsymbol{\pi}|\boldsymbol{x})}{p(\boldsymbol{y}|\boldsymbol{\pi})p(\boldsymbol{\pi})}\mathrm{d}\boldsymbol{\pi}\right]+\mathbb{E}_{p(\boldsymbol{y})}[\log p(\boldsymbol{y})]\\
&=\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\int p(\boldsymbol{\pi}|\boldsymbol{x})\log\frac{p(\boldsymbol{\pi}|\boldsymbol{x})}{p(\boldsymbol{y}|\boldsymbol{\pi})p(\boldsymbol{\pi})}\mathrm{d}\boldsymbol{\pi}\right]+\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\log p(\boldsymbol{y})\right]\\
&=\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\int p(\boldsymbol{\pi}|\boldsymbol{x})\log\frac{p(\boldsymbol{\pi}|\boldsymbol{x})p(\boldsymbol{y})}{p(\boldsymbol{\pi},\boldsymbol{y})}\mathrm{d}\boldsymbol{\pi}\right]\\
&=\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[\int p(\boldsymbol{\pi}|\boldsymbol{x})\log\frac{p(\boldsymbol{\pi}|\boldsymbol{x})}{p(\boldsymbol{\pi}|\boldsymbol{y})}\mathrm{d}\boldsymbol{\pi}\right]\\
&=\mathbb{E}_{p(\boldsymbol{x})}\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})}\left[D_{\mathrm{KL}}\left(p(\boldsymbol{\pi}|\boldsymbol{x})\|p(\boldsymbol{\pi}|\boldsymbol{y})\right)\right],
\end{aligned} \tag{24}$$

which is the target of VB-based EDL methods. □

Table 5: Loss weight β for IB-EDL.

Model	ID Calibration					
	ARC-C	ARC-E	OBQA	CSQA	SciQ	RACE
Llama2-7B	1×10^{-3}	1×10^{-4}	1.5×10^{-4}	5×10^{-5}	1×10^{-6}	1×10^{-6}
Llama3-8B	9×10^{-5}	2×10^{-6}	1×10^{-5}	3×10^{-5}	5×10^{-7}	1×10^{-6}
Mistral-7B	9×10^{-5}	1×10^{-6}	1×10^{-5}	5×10^{-5}	4×10^{-6}	2×10^{-6}
Model	Learning with Label Noise					
	ARC-C	ARC-E	OBQA	CSQA	SciQ	RACE
Llama2-7B	2×10^{-3}	1×10^{-4}	5×10^{-4}	1.6×10^{-4}	5×10^{-6}	1×10^{-6}
Llama3-8B	5×10^{-6}	2×10^{-5}	6×10^{-5}	3×10^{-5}	5×10^{-6}	2×10^{-6}
Mistral-7B	1×10^{-5}	2×10^{-5}	9×10^{-5}	1×10^{-5}	1×10^{-6}	7×10^{-6}

C Detailed implementation

Models: We fine-tuned three models: Llama2-7B (Touvron et al., 2023), Llama3-8B (Dubey et al., 2024), and Mistral-7B-v0.1 (Jiang et al., 2023).

LoRA hyperparameters: We applied LoRA (Hu et al., 2022) finetuning using the PEFT (Mangrulkar et al., 2022) library. For all models, LoRA adaptors were applied to the `q_proj`, `v_proj`, and `lm_head` modules. Additionally, we used Dropout with a dropout rate of $p = 0.1$, LoRA $\alpha = 16$, rank $r = 8$, and set `bias = "lora_only"`.

Training details: For the MAP baseline and all EDL methods, all models were trained for 30000 steps on the CSQA dataset and 10080 steps on the other datasets. The learning rate was set to 0.00005 and annealed using a cosine schedule. The maximum token length was set to 300 for the RACE dataset and 256 for all other datasets. Training was conducted with `bfloat16` precision. For MCD (Gal & Ghahramani, 2016), we performed 10 forward passes. For Ens (Lakshminarayanan et al., 2017; Fort et al., 2019), we used predictions from 3 models. For LA (Yang et al., 2024), I-EDL (Deng et al., 2023), and R-EDL (Chen et al., 2024), we used the original implementations and hyperparameters; please refer to the respective official codebases. For EDL methods, we follow the previous works to apply gradient clipping with maximal gradient norm of 20 to stabilize the training.

IB-EDL implementation details: Both linear prediction heads (for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$) are initialized using the linear head from the pretrained LLM. Additionally, both heads are equipped with LoRA adapters and are jointly trained alongside the LoRA weights of the transformer encoder layers. By default, we used $K = 20$ for sampling pre-evidences from the predicted Gaussian distribution during both training and inference. Table 5 lists the β values used in different experiments. Note that for OOD detection experiments, the parameters are identical to those used for OBQA, as OBQA is the training set for these experiments. We optimized β values using grid search. We suggest a guideline for selecting β: if the MAP baseline shows lower accuracy and higher ECE, or in the presence of label noise, a larger β may be chosen to encourage stronger compression and forgetting of the input. Otherwise, a smaller β can be selected to allow more information from the input follow through the pre-evidences.

D A post-hoc calibration technique for IB-EDL

In this section, we introduce an additional *post-hoc* calibration technique aimed at further refining the calibration of the IB-EDL finetuned model.

Intuition: The key motivation behind this calibration method is to account for an additional source of uncertainty in IB-EDL: the variance σ_j^2 of the pre-evidences. In IB-EDL, the uncertainty mass and belief mass are normalized by the constraint: $\sum_j b_j + u = 1$, where $b_j = \frac{\alpha_j - 1}{\sum_j \alpha_j}$. To incorporate the additional uncertainty arising from the variance of pre-evidences, we consider the effect of increasing u, which leads to a reduction in $\sum_j b_j$. This suggests the need to adjust α_j by

Table 6: Calibration performance of uncertainty-aware methods on fine-tuned Mistral-7B.

Metrics	Method	ARC-C	ARC-E	OBQA	CSQA	SciQ	RACE
Acc ↑	MAP	80.19±0.68	92.07±0.12	88.06±0.61	83.24±1.47	94.73±0.06	86.61±0.28
	MCD	80.26±0.52	92.13±0.15	88.07±0.61	83.24±1.47	94.73±0.06	86.64±0.22
	Ens	80.48±0.45	92.41±0.12	88.51±0.18	83.41±1.44	**94.80±0.10**	**86.81±0.21**
	LA	78.47±1.10	92.12±0.11	88.19±0.69	83.21±1.01	94.66±0.16	86.34±0.39
	EDL	**80.60±0.47**	92.27±0.44	87.23±1.42	83.31±0.34	94.27±0.23	85.70±0.41
	VID	80.37±0.39	91.85±0.13	88.33±0.81	82.52±0.54	94.17±0.12	86.40±0.21
	I-EDL	80.46±1.63	92.28±0.15	88.06±0.31	82.91±0.20	94.03±0.32	85.11±0.61
	R-EDL	80.12±0.97	92.20±0.15	88.33±0.91	83.07±0.83	94.03±0.31	86.07±0.28
	IB-EDL	80.57±0.68	**92.47±0.12**	**88.73±0.70**	**83.65±0.45**	94.40±0.26	86.40±0.32
ECE ↓	MAP	19.42±0.68	7.63±0.16	11.29±0.31	15.72±1.46	4.98±0.06	8.43±0.46
	MCD	19.26±0.52	7.62±0.16	11.22±0.40	15.72±1.46	4.98±0.06	8.42±0.47
	Ens	17.04±1.58	7.03±0.79	8.87±0.90	14.18±2.52	4.78±0.13	8.21±0.62
	LA	20.04±0.62	**1.57±0.39**	4.49±0.17	15.11±2.95	**1.91±0.46**	**2.94±0.25**
	EDL	6.21±1.18	6.25±0.46	6.23±0.56	7.71±0.86	7.64±0.85	7.75±0.86
	VID	9.38±0.46	2.44±0.16	4.88±0.77	7.32±0.74	5.62±0.52	4.20±0.13
	I-EDL	4.70±2.15	10.94±1.27	9.63±0.75	8.85±0.44	11.00±0.30	13.16±0.87
	R-EDL	11.26±0.18	2.86±1.07	5.43±0.65	6.30±0.65	4.48±0.23	3.69±0.52
	IB-EDL	**3.60±1.10**	3.49±1.24	**2.27±0.67**	**6.02±0.08**	4.99±1.22	3.58±0.22
NLL ↓	MAP	2.18±0.14	0.85±0.03	0.85±0.02	1.18±0.06	0.31±0.01	0.52±0.01
	MCD	2.18±0.13	0.84±0.04	0.84±0.03	1.18±0.07	0.31±0.01	0.52±0.02
	Ens	1.82±0.17	0.78±0.10	0.67±0.04	1.01±0.13	0.28±0.01	0.51±0.03
	LA	0.78±0.02	**0.29±0.01**	**0.38±0.02**	**0.59±0.03**	**0.17±0.01**	0.45±0.03
	EDL	0.71±0.04	0.35±0.02	0.47±0.03	0.63±0.01	0.27±0.01	0.47±0.02
	VID	0.76±0.03	0.36±0.01	0.46±0.01	0.68±0.02	0.24±0.01	0.41±0.01
	I-EDL	0.71±0.05	0.38±0.02	0.45±0.01	0.65±0.01	0.29±0.01	0.50±0.01
	R-EDL	0.77±0.02	0.33±0.01	0.41±0.03	0.64±0.01	0.24±0.01	0.45±0.01
	IB-EDL	**0.70±0.01**	0.32±0.02	0.41±0.01	0.61±0.01	0.23±0.01	**0.40±0.01**

subtracting a term that is proportional to the standard deviation σ_j, which is predicted by IB-EDL. We parameterize this adjustment as $\zeta \cdot \sigma_j$, where ζ is a scalar hyperparameter. Consequently, we update α_j as follows: $\alpha_j \leftarrow \alpha_j - \zeta \cdot \sigma_j$. The intuition behind this update is that if the model exhibits high uncertainty in predicting α_j (i.e., predicting pre-evidences $\tilde{e}_j$), we enforce a more conservative belief representation by reducing α_j accordingly. This adjustment prevents excessively large values of α_j and ensures a more calibrated uncertainty estimation. As a result, the updated uncertainty mass is given by: $u = \frac{C}{\sum_j \alpha_j - \zeta \cdot \sum_j \sigma_j}$. This approach effectively integrates the variance of pre-evidences into the calibration process, leading to improved uncertainty quantification in IB-EDL.

Determining the optimal value of ζ: Theoretically, the EDL model can still be overconfident or underconfident after training, leading to an overestimation or underestimation of the uncertainty mass u. Since u may require adjustment in either direction, we allow the hyperparameter $\zeta \in \mathbb{R}$ to be either positive or negative, enabling calibration in both cases. To determine the optimal ζ, we recommend to analyze the calibration curve (Guo et al., 2017) on the training or validation set, which plots accuracy against confidence for binned samples. If the calibration curve lies below the optimal diagonal line, it indicates that the model is overconfident. In this case, we set $\zeta > 0$ to encourage greater uncertainty and improve calibration. Conversely, if the calibration curve lies above the diagonal, the model is underconfident, and we set $\zeta < 0$ to increase certainty and correct for underconfidence.

Values of ζ used in additional experiments: It is important to note that this post-hoc calibration technique is an optional component of IB-EDL and is not required in all cases. In the experiments presented in the main text, we omit this technique. In the calibration experiment on OOD test sets presented in Table 12, we use $\zeta = -1.0$ for OBQA $\rightarrow$ ARC-C, $\zeta = 3.0$ for OBQA $\rightarrow$ ARC-E, and $\zeta = -5.0$ for OBQA $\rightarrow$ CSQA.

Table 7: OOD detection performance on fine-tuned Mistral-7B. $A \rightarrow B$ indicates A as the ID training set and B as the OOD test set. MP and UM are two scores for measuring AUROC.

Model	Method	OBQA → ARC-C		OBQA → ARC-E		OBQA → CSQA	
		AUROC ↑		AUROC ↑		AUROC ↑	
		MP	UM	MP	UM	MP	UM
Mistral-7B	MAP	60.40±1.36	–	53.30±1.16	–	63.70±1.10	–
	MCD	60.39±1.37	–	53.30±1.16	–	63.70±1.10	–
	Ens	60.67±0.87	–	54.05±0.88	–	63.80±1.09	–
	LA	61.61±1.05	–	53.14±0.93	–	68.31±1.11	–
	EDL	84.17±1.87	77.34±6.97	81.81±2.31	74.18±6.38	83.55±2.67	78.28±5.81
	VID	86.30±3.85	84.74±0.89	88.16±1.87	90.68±0.17	88.93±1.39	81.45±2.04
	I-EDL	85.02±0.90	82.28±1.44	82.67±1.34	79.42±2.33	85.08±0.87	82.07±1.43
	R-EDL	76.26±1.05	72.85±0.61	71.95±0.78	67.56±0.72	75.59±1.32	71.93±2.05
	IB-EDL	**90.28±1.22**	**88.53±1.08**	**89.54±1.16**	**94.29±0.49**	**90.45±1.13**	**83.85±2.55**

E Additional Experimental Results

E.1 ID calibration and OOD detection results of Mistral-7B

In this subsection, we present the ID calibration and OOD detection performance of the fine-tuned Mistral-7B model. Table 6 provides an overview of the calibration performance on ID datasets, while Table 7 reports the OOD detection results.

E.2 Additional results of fine-tuning models on noisy datasets

Table 8 and Table 9, and Table 10 show the results of fine-tuning Llama2-7B, Llama3-8B, and Mistral-7B on noisy datasets, respectively.

Results: In the context of noisy data, the primary objective is to preserve accuracy. Therefore, in Section 4.4 of the main text, we focus primarily on accuracy results. As highlighted in Section 4.2, a good uncertainty-aware method should achieve comparable accuracy while maintaining a low calibration error, a position shared by many other current research efforts (see, e.g., Chen et al., 2024). Here, we thus provide additional results for the ECE and NLL metrics. In the presence of label noise, Non-EDL methods tend to underperform EDL methods significantly in terms of accuracy. In the comparisons of Tables 8 to 10, we therefore focus on the comparison of EDL methods, which manage to strike the right balance of good accuracy and uncertainty quantification. Among these, IB-EDL demonstrates the highest accuracy and achieves the lowest ECE and NLL compared to other EDL approaches, highlighting its robustness against label noise.

E.3 Additional OOD detection results on MMLU-Math dataset

We conduct additional OOD detection experiments using Llama2-7B in a setting characterized by a significant distribution shift. For this purpose, we select a subset of the MMLU dataset (Hendrycks et al., 2021a;b) as the OOD test set, which includes data samples from the math-related topics `college_mathematics`, `high_school_mathematics`, and `abstract_algebra`. We refer to this subset as the MMLU-Math dataset. In addition, we use the OBQA dataset as the ID dataset. While OBQA focuses on common-sense reasoning, the MMLU-Math dataset requires advanced mathematical knowledge to solve its questions. As a result, the distribution shift in this setting is substantially larger compared to settings like OBQA → ARC-C.

As shown in Table 11, IB-EDL consistently achieves the best OOD detection performance when using either MP or UM as the detection score. Additionally, a general trend is observed: the AUROCs of IB-EDL and other baselines improve as the distribution shift increases compared to the setting OBQA → ARC-C.

Table 8: Fine-tuning Llama2-7B on noisy datasets. In each training dataset, the labels of 30% samples are randomly perturbed. A robust uncertainty-aware method should not only maintain accuracy but also exhibit low calibration error. Non-EDL methods tend to significantly underperform EDL methods in terms of Accuracy. Therefore, the comparison for ECE and NLL is limited to EDL methods, with the best and second-best values among them highlighted.

Metrics	Method	ARC-C	ARC-E	OBQA	CSQA	SciQ	RACE
Acc ↑	MAP	51.08±2.96	71.39±2.73	73.93±2.21	74.56±0.31	88.63±0.06	76.83±0.39
	MCD	51.08±2.96	71.40±2.72	73.93±2.20	74.56±0.32	88.64±0.06	76.95±0.19
	Ens	56.10±3.11	76.03±1.99	75.23±1.06	74.59±0.23	88.64±0.06	76.94±0.22
	LA	53.38±2.17	74.90±1.76	74.57±2.22	74.59±0.20	88.47±0.21	77.04±0.39
	EDL	57.16±3.33	76.16±1.12	74.46±1.33	74.65±0.05	89.03±0.25	77.69±0.44
	VID	57.56±0.58	76.57±1.79	76.67±1.42	75.97±0.66	89.06±0.21	78.75±0.25
	I-EDL	53.86±1.25	76.08±0.71	77.00±0.88	74.01±1.28	**89.26±0.32**	76.48±0.87
	R-EDL	57.00±0.76	76.96±1.07	73.20±1.44	74.36±0.49	87.33±0.49	78.01±0.53
	IB-EDL	**59.06±2.07**	**80.27±0.48**	**78.13±0.99**	**76.35±0.66**	89.06±0.50	**79.41±0.59**
ECE ↓	MAP	21.89±0.11	8.95±1.04	8.21±0.78	4.45±1.51	19.05±0.17	15.87±0.30
	MCD	21.87±0.12	8.91±1.10	8.23±0.82	4.47±1.55	19.04±0.16	15.84±0.26
	Ens	8.24±1.18	7.83±0.28	11.63±1.12	4.42±1.52	18.91±0.12	15.90±0.94
	LA	7.12±1.06	27.02±1.39	22.06±2.11	22.57±0.09	26.28±0.04	24.70±0.24
	EDL	**8.38±1.91**	26.16±4.44	35.64±2.97	44.29±2.42	46.47±0.24	30.59±1.14
	VID	19.14±0.99	**13.28±1.84**	15.00±0.98	13.76±0.90	23.49±0.34	21.22±0.63
	I-EDL	13.97±9.04	35.30±1.74	37.56±0.53	38.77±1.59	48.60±0.37	36.34±1.09
	R-EDL	12.25±1.18	20.67±2.30	28.02±2.63	33.64±0.29	39.58±0.56	30.42±0.89
	IB-EDL	11.21±2.27	13.48±1.61	**12.30±1.65**	**10.48±0.82**	**23.47±0.31**	**21.15±0.36**
NLL ↓	MAP	1.94±0.06	1.05±0.09	0.74±0.03	0.84±0.02	0.50±0.01	0.73±0.00
	MCD	1.94±0.05	1.05±0.09	0.74±0.03	0.81±0.02	0.49±0.01	0.72±0.02
	Ens	1.37±0.21	0.73±0.06	0.73±0.03	0.83±0.01	0.49±0.01	0.73±0.01
	LA	1.32±0.06	1.03±0.03	0.85±0.00	0.92±0.01	0.56±0.00	0.81±0.00
	EDL	1.19±0.03	0.97±0.03	1.05±0.02	1.35±0.11	0.92±0.01	0.90±0.03
	VID	1.27±0.01	0.83±0.03	0.76±0.03	0.85±0.01	0.54±0.01	**0.71±0.01**
	I-EDL	1.29±0.12	1.09±0.02	1.03±0.01	1.20±0.01	0.95±0.00	1.02±0.01
	R-EDL	1.24±0.01	0.92±0.01	0.95±0.03	1.13±0.01	0.83±0.01	0.93±0.01
	IB-EDL	**1.18±0.03**	**0.74±0.03**	**0.71±0.03**	**0.81±0.01**	**0.52±0.01**	**0.71±0.01**

E.4 Addition calibration results on OOD test sets

In this section, we evaluate the calibration performance on OOD test sets to assess whether the uncertainty-aware methods can generalize effectively to OOD datasets. Specifically, we fine-tune Llama3-8B on the OBQA dataset and evaluate it on three OOD test sets. As shown in Table 12, IB-EDL achieves the best ECE and NLL on two out of the three OOD test sets, demonstrating that its calibration performance generalizes well to OOD datasets.

E.5 Analysis on training and inference speed, and memory consumption

In this section, we evaluate the complexity of uncertainty-aware methods using three key metrics: (1) the number of samples processed per second during training; (2) the number of samples processed per second during inference; and (3) GPU memory consumption during training. For this experiment, we use Llama3-8B in mixed precision as the model and OBQA as both the training and test dataset. All experiments are conducted on a single NVIDIA H100 GPU. As shown in Table 13, IB-EDL demonstrates comparable training and inference speeds as well as similar memory consumption to MAP and other EDL baselines. Moreover, IB-EDL attains faster inference speeds compared to methods such as MCD and Ens, which require multiple forward passes, and LA, which involves gradient computation during inference. This substantial improvement in speed further highlights the computational efficiency of IB-EDL.

Table 9: Fine-tuning Llama3-8B on noisy datasets. In each training dataset, the labels of 30% samples are randomly perturbed. A robust uncertainty-aware method should not only maintain accuracy but also exhibit low calibration error. Non-EDL methods tend to significantly underperform EDL methods in terms of Accuracy. Therefore, the comparison for ECE and NLL is limited to EDL methods, with the best and second-best values among them highlighted.

Metrics	Method	ARC-C	ARC-E	OBQA	CSQA	SciQ	RACE
Acc ↑	MAP	57.71±0.42	80.34±1.47	78.78±1.00	77.04±0.41	92.60±0.53	86.93±0.11
	MCD	57.71±0.42	80.37±1.46	79.00±0.92	77.05±0.41	92.87±0.12	86.93±0.12
	Ens	63.39±1.09	84.63±0.53	80.61±0.53	77.62±0.41	92.97±0.06	86.93±0.07
	LA	66.62±1.08	82.64±0.50	79.59±0.72	77.80±0.49	92.93±0.21	87.00±0.01
	EDL	**69.43±0.98**	87.57±0.13	85.60±0.72	79.14±0.41	92.90±0.40	86.26±0.76
	VID	66.58±1.92	87.67±0.99	84.86±1.01	**79.66±1.26**	93.23±0.25	86.86±0.26
	I-EDL	63.87±2.65	84.06±2.80	84.26±0.42	78.02±0.87	92.53±0.37	86.01±0.51
	R-EDL	71.25±1.20	84.91±3.91	85.73±0.70	78.57±0.21	**93.56±0.05**	86.16±0.42
	IB-EDL	68.53±0.25	**88.05±0.43**	**86.13±0.51**	79.59±0.79	93.46±0.38	**87.01±0.20**
ECE ↓	MAP	13.26±0.85	8.52±0.63	3.72±1.06	8.43±1.04	16.29±0.43	19.61±0.25
	MCD	12.96±1.15	8.51±0.62	4.17±1.45	8.40±1.07	16.29±0.43	19.64±0.26
	Ens	10.41±0.74	11.09±0.59	4.24±1.91	8.21±0.21	16.15±0.16	19.69±0.23
	LA	19.02±0.78	23.63±1.74	11.61±0.29	18.05±0.86	18.36±0.37	20.99±0.18
	EDL	11.01±0.79	24.29±0.81	41.29±0.62	34.19±1.93	48.36±0.43	34.92±0.46
	VID	**5.90±0.97**	19.58±0.29	19.53±0.77	15.22±1.15	26.30±0.22	**22.25±0.25**
	I-EDL	9.57±2.10	33.57±6.96	42.64±0.36	41.89±1.18	50.13±0.31	42.49±0.20
	R-EDL	14.50±2.55	23.91±2.93	35.09±0.84	30.95±2.78	41.89±0.15	30.74±0.94
	IB-EDL	6.73±1.30	**19.49±0.38**	**17.38±0.52**	**11.73±1.11**	**25.00±0.33**	23.43±0.32
NLL ↓	MAP	1.39±0.05	0.78±0.03	0.64±0.01	0.83±0.01	0.39±0.01	0.56±0.01
	MCD	1.38±0.02	0.78±0.03	0.64±0.03	0.82±0.02	0.40±0.01	0.57±0.01
	Ens	1.15±0.05	0.64±0.03	0.61±0.01	0.78±0.02	0.39±0.01	0.57±0.02
	LA	1.11±0.01	0.80±0.02	0.63±0.02	0.85±0.01	0.39±0.00	0.57±0.01
	EDL	**0.94±0.02**	0.70±0.01	0.92±0.01	1.04±0.03	0.85±0.00	0.78±0.01
	VID	0.98±0.03	0.62±0.02	0.63±0.01	0.80±0.02	0.49±0.01	**0.57±0.01**
	I-EDL	1.03±0.03	0.87±0.06	0.96±0.01	1.16±0.00	0.89±0.01	0.91±0.00
	R-EDL	1.00±0.02	0.76±0.05	0.83±0.02	0.99±0.02	0.72±0.00	0.73±0.01
	IB-EDL	0.99±0.02	**0.60±0.01**	**0.57±0.01**	**0.74±0.02**	**0.46±0.01**	**0.57±0.01**

E.6 Sensitivity analysis on the number of bins of ECE

Following Yang et al. (2024), we use 15 bins by default when measuring ECE. To assess the impact of this hyperparameter, we conduct a sensitivity analysis by varying the number of bins across $\{10, 15, 25, 35\}$. For this experiment, we train and test Llama3-8B on the OBQA dataset and calculate the ECE for each bin setting. As shown in Table 14, although ECE values increase slightly with a higher number of bins, the relative rankings of the methods remain largely consistent.

Table 10: Fine-tuning Mistral-7B on noisy datasets. In each training dataset, the labels of 30% samples are randomly perturbed. A robust uncertainty-aware method should not only maintain accuracy but also exhibit low calibration error. Non-EDL methods tend to significantly underperform EDL methods in terms of Accuracy. Therefore, the comparison for ECE and NLL is limited to EDL methods, with the best and second-best values among them highlighted.

Metrics	Method	ARC-C	ARC-E	OBQA	CSQA	SciQ	RACE
Acc ↑	MAP	50.65±0.30	66.30±2.16	75.72±0.46	69.81±0.55	92.96±0.32	85.47±0.35
	MCD	50.65±0.30	66.21±2.04	75.46±0.24	69.77±0.51	92.93±0.31	85.46±0.35
	Ens	58.16±0.47	76.10±0.86	76.93±2.14	74.18±1.50	92.93±0.39	85.62±0.28
	LA	66.35±0.63	75.48±1.88	78.20±1.25	74.24±1.01	93.02±0.29	85.57±0.06
	EDL	61.63±1.80	85.01±0.23	83.20±1.25	76.63±0.98	93.33±0.06	84.91±0.10
	VID	62.27±2.18	76.98±1.13	82.06±1.51	77.31±1.01	93.20±0.17	85.59±0.06
	I-EDL	68.28±1.19	79.73±0.57	81.06±0.81	77.41±0.09	93.33±0.31	84.44±0.28
	R-EDL	**76.39±0.60**	**86.85±0.69**	82.76±1.73	77.10±1.10	93.09±0.78	85.07±0.34
	IB-EDL	66.44±0.80	85.17±1.39	**84.33±0.51**	**77.44±0.39**	**93.63±0.21**	**85.68±0.20**
ECE ↓	MAP	18.08±0.45	9.95±1.18	7.83±0.62	9.61±0.98	15.11±0.28	17.80±0.33
	MCD	18.24±0.65	9.94±1.18	7.82±0.61	9.60±0.99	15.10±0.27	17.80±0.33
	Ens	10.94±0.21	12.40±0.34	7.02±0.61	8.72±0.29	15.09±0.17	18.58±0.46
	LA	18.50±1.46	17.31±0.23	13.92±1.45	15.04±0.41	18.13±0.19	18.32±0.59
	EDL	12.76±0.41	19.84±0.95	32.57±1.89	25.26±5.63	48.55±0.33	34.06±0.14
	VID	**6.27±0.66**	**13.17±0.76**	**13.92±0.97**	11.66±1.17	26.45±0.16	**22.01±0.08**
	I-EDL	13.14±1.17	21.44±1.54	35.10±1.07	33.29±1.04	50.90±0.44	41.19±0.27
	R-EDL	17.63±2.83	17.57±0.42	18.55±1.68	21.35±1.24	41.48±0.26	29.40±0.31
	IB-EDL	7.46±1.58	19.08±3.68	14.90±0.50	**11.58±0.43**	**25.08±0.27**	22.04±0.06
NLL ↓	MAP	1.67±0.03	0.98±0.06	0.73±0.05	0.98±0.03	0.38±0.01	0.56±0.00
	MCD	1.67±0.03	0.98±0.06	0.70±0.03	0.98±0.03	0.38±0.01	0.56±0.01
	Ens	1.32±0.01	0.73±0.03	0.64±0.04	0.81±0.02	0.39±0.01	0.57±0.02
	LA	1.08±0.03	0.83±0.03	0.70±0.03	0.93±0.01	0.40±0.01	0.56±0.00
	EDL	**0.93±0.05**	0.67±0.00	0.82±0.03	0.99±0.04	0.85±0.00	0.78±0.01
	VID	1.04±0.06	0.74±0.02	0.63±0.01	0.84±0.01	0.50±0.01	**0.60±0.01**
	I-EDL	0.96±0.04	0.76±0.02	0.90±0.01	1.05±0.02	0.89±0.00	0.91±0.01
	R-EDL	**0.93±0.03**	**0.64±0.02**	0.64±0.03	0.97±0.04	0.72±0.01	0.72±0.01
	IB-EDL	0.98±0.02	0.69±0.03	**0.57±0.01**	**0.81±0.01**	**0.45±0.01**	**0.60±0.01**

Table 11: OOD Detection AUROC on Llama3-8B in the setting OBQA → MMLU-Math. IB-EDL achieves the best performance even under significant distribution shifts, such as transitioning from a common-sense reasoning dataset like OBQA to a math-focused OOD dataset.

Method	OBQA → MMLU-Math	
	AUROC ↑	
	MP	UM
MAP	91.36±0.57	-
MCD	90.85±0.33	-
Ens	90.68±0.80	-
LA	91.09±0.41	-
EDL	92.78±0.26	92.86±0.21
VID	91.64±0.79	66.61±4.98
I-EDL	91.48±0.72	90.67±0.88
R-EDL	88.44±2.11	88.22±1.70
IB-EDL	**93.63±0.66**	**93.64±0.56**

Table 12: Calibration performance of uncertainty-aware methods on fine-tuned Llama3-8B in the OOD setting. The model is trained on OBQA and tested on three different OOD test sets.

Metrics	Method	OBQA → ARC-C	OBQA → ARC-E	OBQA → CSQA
Acc ↑	MAP	79.18±0.45	88.06±0.20	69.37±0.67
	MCD	79.16±0.43	88.05±0.20	69.38±0.68
	Ens	79.27±0.25	88.15±0.02	69.14±0.47
	LA	**79.38±0.40**	**88.36±0.24**	69.34±0.58
	EDL	78.27±0.79	86.38±0.87	69.34±0.88
	VID	78.27±0.57	87.41±0.82	69.99±1.07
	I-EDL	78.55±0.30	87.55±0.20	70.49±0.56
	R-EDL	78.32±1.24	87.31±0.93	70.62±1.28
	IB-EDL	78.31±1.14	87.94±0.22	**71.29±0.96**
ECE ↓	MAP	18.04±0.38	9.63±0.47	27.85±0.53
	MCD	18.06±0.36	9.63±0.46	27.86±0.46
	Ens	16.74±0.13	9.19±0.65	27.07±1.32
	LA	6.61±0.41	**3.02±0.15**	11.99±0.72
	EDL	7.65±0.47	10.11±0.85	8.32±1.33
	VID	8.74±1.19	3.19±0.19	16.66±0.73
	I-EDL	7.68±1.19	11.38±0.65	5.96±0.71
	R-EDL	5.03±0.62	5.32±0.46	12.84±1.73
	IB-EDL	**4.67±1.09**	5.03±0.15	**4.51±0.15**
NLL ↓	MAP	1.30±0.02	0.71±0.03	2.06±0.04
	MCD	1.33±0.06	0.70±0.04	2.13±0.13
	Ens	1.19±0.03	0.68±0.05	2.01±0.10
	LA	0.73±0.02	**0.42±0.02**	1.15±0.01
	EDL	0.74±0.01	0.52±0.01	0.98±0.03
	VID	0.78±0.01	0.46±0.02	1.06±0.02
	I-EDL	0.73±0.02	0.53±0.01	0.94±0.01
	R-EDL	0.73±0.04	0.46±0.01	1.00±0.05
	IB-EDL	**0.72±0.02**	0.44±0.02	**0.93±0.02**

Table 13: Comparison of computational efficiency for various methods using Llama3-8B on the OBQA. IB-EDL demonstrates comparable training and inference speeds as well as memory consumption compared to MAP and other EDL methods, confirming its computational efficiency.

Method	Test Samples/s ↑	Training Samples/s ↑	Memory (GB) at Training↓
MAP	69.55±2.86	26.57±1.96	21.21±0.35
MCD (10 forwards)	9.79±1.21	-	-
Ens (3 models)	25.77±3.54	-	-
LA	5.95±0.49	-	-
EDL	68.99±1.59	26.44±1.11	21.23±0.15
VID	69.17±0.99	26.69±1.37	21.29±0.37
I-EDL	68.94±2.18	26.02±0.71	21.33±0.11
R-EDL	68.84±1.09	26.47±1.09	21.27±0.21
IB-EDL	68.08±1.75	26.41±1.04	21.88±0.66

Table 14: Sensitivity analysis of ECE with respect to the number of bins. We train and test Llama3-8B on the OBQA dataset. The results show that while the ECE values increase slightly as the number of bins increases, the relative rankings of the methods remain consistent.

Method	Bins = 10	Bins = 15	Bins = 25	Bins = 35
MAP	$10.45_{\pm 0.52}$	$10.52_{\pm 0.87}$	$10.89_{\pm 0.67}$	$10.99_{\pm 1.01}$
MCD	$10.31_{\pm 0.37}$	$10.48_{\pm 0.86}$	$10.69_{\pm 0.59}$	$10.83_{\pm 0.76}$
Ens	$10.11_{\pm 0.13}$	$10.08_{\pm 0.90}$	$10.91_{\pm 0.40}$	$10.92_{\pm 0.84}$
LA	$5.20_{\pm 1.29}$	$5.26_{\pm 1.30}$	$6.33_{\pm 1.11}$	$6.42_{\pm 0.96}$
EDL	$8.16_{\pm 1.25}$	$8.28_{\pm 1.62}$	$8.78_{\pm 1.27}$	$9.44_{\pm 1.79}$
VID	$5.29_{\pm 0.50}$	$5.99_{\pm 1.41}$	$7.16_{\pm 1.59}$	$7.34_{\pm 1.17}$
I-EDL	$7.31_{\pm 0.33}$	$7.57_{\pm 0.52}$	$8.20_{\pm 0.46}$	$9.01_{\pm 0.50}$
R-EDL	$4.64_{\pm 0.87}$	$4.68_{\pm 1.35}$	$4.81_{\pm 1.09}$	$5.47_{\pm 0.82}$
IB-EDL	$\mathbf{2.77_{\pm 0.61}}$	$\mathbf{2.34_{\pm 0.61}}$	$\mathbf{3.91_{\pm 0.77}}$	$\mathbf{4.54_{\pm 0.52}}$