Independent Study On Uncertainty Quantification with Pre-trained Models and Single Model

Wang Ma
Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, NY 12180
maw6@rpi.edu

Contents

Preprint.

1 Survey of Uncertainty Quantification with Pre-trained Models and Single Model Methods

1.1 Introduction

1.1.1 The Imperative of Uncertainty Quantification in Modern Deep Learning

Deep Neural Networks (DNNs) have demonstrated remarkable success across a multitude of domains, including computer vision, natural language processing, and scientific discovery. However, their operational mechanism often resembles a "black box," and a significant concern is their propensity to generate predictions that are not only incorrect but also delivered with a high degree of unwarranted confidence Begoli et al. (2021); He et al. (2023). For a given input $\mathbf{x}$, a typical classification model outputs a probability vector $p(y|\mathbf{x}, \theta)$ over classes, where a high confidence prediction corresponds to $\max(p) \approx 1$. This characteristic poses substantial risks, particularly in high-stakes applications such as medical diagnosis, autonomous vehicle navigation, and critical infrastructure management, where erroneous and overconfident decisions can lead to severe, even catastrophic, consequences Amodei et al. (2016).

Uncertainty Quantification (UQ) emerges as a critical scientific discipline to address this challenge. It seeks to distinguish between two primary types of uncertainty Kendall and Gal (2017):

- **Aleatoric Uncertainty** ($\mathcal{U}_A$): This refers to the inherent, irreducible noise or randomness in the data generating process. It captures the notion that even a perfect model cannot resolve ambiguity present in the input data itself. It is often modeled by placing a distribution on the model's output, such as $p(y|f(\mathbf{x}; \theta))$.
- **Epistemic Uncertainty** ($\mathcal{U}_E$): This is the uncertainty in the model parameters θ, stemming from a lack of knowledge or limited training data. It is reducible with more data. In a Bayesian framework, this is represented by the posterior distribution over the weights, $p(\theta|\mathcal{D}_{\text{train}})$.

The total predictive uncertainty is a combination of both. The capacity of models to not only achieve high predictive accuracy but also to articulate their confidence by quantifying these uncertainties is paramount for responsible and safe deployment. The drive towards robust UQ is intrinsically linked to the broader goal of constructing trustworthy and safe Artificial Intelligence (AI) systems. As AI systems assume increasingly critical roles, their ability to "know when they don't know"—that is, to recognize and communicate their own limitations—becomes indispensable for facilitating human oversight and mitigating potential harms.

1.1.2 Challenges and Significance of UQ for Pre-trained and Single Models

The landscape of deep learning is increasingly dominated by large Pre-trained Models (PTMs). This paradigm, while powerful, introduces unique challenges for UQ. Frequently, access to the original, extensive training datasets is restricted, and the computational cost of fully retraining these massive models is prohibitive. Consequently, UQ methodologies must be adapted to operate effectively under these constraints.

Simultaneously, for single models—as opposed to computationally intensive ensemble methods—the development of efficient UQ techniques is a significant research thrust. A prominent baseline, **Deep Ensembles**, involves training an ensemble of M models with different random initializations, $\{f(\mathbf{x}; \theta_m)\}_{m=1}^{M}$, and using the variance of their predictions as a measure of epistemic uncertainty Lakshminarayanan et al. (2017). However, this is often too expensive. Therefore, methods that estimate uncertainty from a single network pass are highly desirable. Two popular examples include:

- **Monte Carlo (MC) Dropout**: This technique approximates a Bayesian neural network by keeping dropout active at inference time. By performing T stochastic forward passes, one obtains a set of predictions $\{\hat{y}_t\}_{t=1}^{T}$. The predictive mean and variance of this set serve as the final prediction and its uncertainty estimate, respectively Gal and Ghahramani (2016). The predictive probability is given by $p(y|\mathbf{x}) \approx \frac{1}{T} \sum_{t=1}^{T} p(y|\mathbf{x}, \hat{\theta}_t)$.
- **Evidential Deep Learning (EDL)**: Instead of outputting a simple probability, an EDL model outputs the parameters α of a Dirichlet distribution, $Dir(p|\alpha)$. This distribution

is placed over the categorical output distribution, allowing the model to directly quantify classification uncertainty based on the evidence collected from the input Sensoy et al. (2018).

Effective UQ for PTMs and single models is crucial for their safe and reliable deployment in diverse downstream applications. Success in this area could significantly democratize access to uncertainty-aware AI.

1.1.3 Scope and Objectives of the section

This independent study aims to provide a comprehensive survey and critical analysis of Uncertainty Quantification methodologies tailored for both single deep learning models and pre-trained models. In alignment with the course syllabus and inspired by large-scale benchmarking efforts such as Nado et al. (2021), the primary objectives are:

- To gain an in-depth understanding of recent advancements in UQ, particularly focusing on techniques applicable to single models (e.g., evidential deep learning, interval-based UQ) and pre-trained models (e.g., last-layer methods, post-hoc UQ).
- To critically evaluate existing UQ methods, analyzing their novel contributions, underlying assumptions, and inherent limitations.
- To compare and contrast different classes of UQ methods, such as those operating in parameter-space versus output-space, and to identify their respective strengths and weaknesses.
- To identify deficiencies or areas for improvement in current UQ approaches and to design and propose innovative solutions to address these gaps, potentially by combining different methods or exploring novel perspectives (Bayesian/non-Bayesian, input/parameter/output space, prior/posterior).
- To conduct rigorous experimental evaluations of any proposed solutions on benchmark datasets, comparing their performance against state-of-the-art methods.

1.1.4 Structure of this section

This report is structured to systematically address the objectives outlined above. Section 1 provides a comprehensive survey of existing UQ methodologies, including fundamental concepts and a categorization of methods. Section III (to be developed) will detail the gap analysis of current techniques and present proposed methodological enhancements. Section IV (to be developed) will describe the experimental design, including datasets, evaluation metrics, and baseline comparisons, followed by a detailed analysis of the results obtained from implemented solutions. Finally, Section V (to be developed) will conclude the report with a summary of key findings, a discussion of limitations, and an outline of potential future research directions in this rapidly evolving field.

1.2 Fundamental Concepts: Aleatoric vs. Epistemic Uncertainty

A cornerstone of UQ in machine learning is the distinction between two primary types of uncertainty, first rigorously defined in the context of modern deep learning by Kendall and Gal (2017); Der Kiureghian and Ditlevsen (2009): aleatoric and epistemic uncertainty. For classification problems, these can be elegantly decomposed using information-theoretic measures Depeweg et al. (2018); **?**.

- **Aleatoric Uncertainty** ($\mathcal{U}_A$), or data uncertainty, reflects the inherent randomness in the data. This is quantified by the *expected predictive entropy*. We calculate the entropy of the predictive distribution for various model parameters θ sampled from the posterior $p(\theta|\mathcal{D})$, and then take the average. This measures the ambiguity present in the data itself.

$$\mathcal{U}_A(\mathbf{x}) = \mathbb{E}_{p(\theta|\mathcal{D})}\left[\mathcal{H}[p(y|\mathbf{x},\theta)]\right] = \mathbb{E}_{p(\theta|\mathcal{D})}\left[-\sum_c p(y=c|\mathbf{x},\theta)\log p(y=c|\mathbf{x},\theta)\right] \quad (1)$$

- **Epistemic Uncertainty** ($\mathcal{U}_E$), or model uncertainty, arises from a lack of knowledge in the model parameters due to limited data. It is reducible and can be quantified by the *mutual information* between the prediction y and the model parameters θ **?**. It is calculated as the difference between the total uncertainty (the entropy of the final predictive distribution) and the aleatoric uncertainty. High mutual information indicates that the model's prediction is highly dependent on the specific choice of parameters, signifying high model uncertainty.

$$\mathcal{U}_E(\mathbf{x}) = \mathcal{I}(y;\theta|\mathbf{x},\mathcal{D}) = \mathcal{H}[p(y|\mathbf{x},\mathcal{D})] - \mathbb{E}_{p(\theta|\mathcal{D})}[\mathcal{H}[p(y|\mathbf{x},\theta)]] \tag{2}$$

Here, the total uncertainty is the entropy of the marginalized predictive distribution, $\mathcal{H}[p(y|\mathbf{x},\mathcal{D})]$, where $p(y|\mathbf{x},\mathcal{D}) = \int p(y|\mathbf{x},\theta)p(\theta|\mathcal{D})d\theta$.

The capacity to differentiate between aleatoric and epistemic uncertainty carries significant practical weight. High aleatoric uncertainty suggests an inherent limit to predictability, while high epistemic uncertainty signals that the model is unsure and its predictions should not be trusted, potentially guiding active learning strategies to gather more data in those regions of the input space Houlsby et al. (2011).

1.3 Categorization of UQ Methods for Pre-trained Models

UQ methods can be categorized based on various criteria. A particularly relevant categorization for PTMs is based on the level of access required to the model and its training data Abdar et al. (2021).

- **White-Box / Full Access Methods:** These methods assume full access to the model architecture, parameters, and often the original training data. They typically involve modifying the training process itself. Examples include:
  - **Bayesian Neural Networks (BNNs):** Fully Bayesian training to learn a posterior distribution over all weights Neal (2012).
  - **Deep Ensembles:** Training multiple models from scratch with different initializations and aggregating their predictions Lakshminarayanan et al. (2017).

  While powerful, these are often impractical for large PTMs due to prohibitive computational costs.
- **Grey-Box / Parameter Access Methods:** These methods do not require the original training data but need access to the pre-trained model's parameters to perform modifications or further training, often on a smaller downstream dataset. This is a common scenario in transfer learning.
  - **Last-Layer Bayesian Methods:** A computationally efficient compromise where a Bayesian treatment is applied only to the final layer(s) of the network, keeping the pre-trained feature extractor fixed Kristiadi et al. (2020).
  - **Laplace Approximation:** This involves finding a Gaussian approximation to the posterior distribution around a pre-trained MAP (Maximum A Posteriori) weight configuration. It is more efficient than MCMC-based BNNs MacKay (1992); Ritter et al. (2018).
- **Black-Box / Post-Hoc Methods:** These methods require only query access to the pre-trained model. They treat the model as a black box, operating on its outputs without needing to modify its internal parameters. This makes them highly versatile and applicable to proprietary models (e.g., via APIs).
  - **Temperature Scaling:** A simple yet effective calibration technique that adjusts the "temperature" parameter of the final softmax layer on a validation set to make the model's confidence scores more representative of the true likelihoods Guo et al. (2017).
  - **Input-Perturbation Methods:** Some methods estimate uncertainty by analyzing the sensitivity of the model's output to small perturbations of the input, without needing access to model weights Wang and Ji (2024).

This categorization provides a useful framework for navigating the landscape of UQ techniques and selecting the most appropriate method based on the constraints of a given application.

1.3.1 Full Access (Model & Data for Retraining/Tuning)

This category encompasses methods that necessitate access to both the model architecture and training data for retraining or fine-tuning. A model $f_\theta(\mathbf{x})$ can be viewed as a composition of a feature extractor

$g_{\theta_{enc}}(\mathbf{x})$ and a decision head $h_{\theta_{head}}(z)$, where $z = g_{\theta_{enc}}(\mathbf{x})$ are the extracted features. Full access methods can modify both θ_{enc} and θ_{head}. While offering comprehensive UQ, they are often less suitable for off-the-shelf PTMs where original data is unavailable or retraining is computationally infeasible.

Last-Layer Methods Last-layer UQ methods are a pragmatic compromise, leveraging powerful pre-trained encoders while localizing UQ modifications to the decision head Kristiadi et al. (2020). The encoder parameters θ_{enc} are kept fixed, and a new, often stochastic, head $h_{\theta_{head}}$ is trained on a downstream task.

One such approach is Wang et al. (2024a), it introduces Credal-Set Neural Networks (CreNets), a novel neural network architecture designed to directly output a credal set. The key innovation lies in the final layers of the network and a custom loss function. In practice, they change the last-layer of neural network to force it output interval output.

For a C-class classification problem, a CreNet has $2C$ output nodes in its final layer. For each class c, there are two outputs: one for the midpoint m_c of the probability interval and one for its half-length h_c. The lower and upper probability bounds are then given by:

$$\underline{p}_c = m_c - h_c \quad \text{and} \quad \overline{p}_c = m_c + h_c$$

Constraints are imposed to ensure that these bounds form a valid credal set, i.e., $\sum_c \underline{p}_c \leq 1 \leq \sum_c \overline{p}_c$.

Another last-layer technique is the **Bayesian Non-negative Decision Layer (BNDL) ?**. BNDL reformulates the final linear decision layer (which computes logits $\mathbf{L} = h_{\theta_{head}}(z)$) as a conditional Bayesian non-negative factor analysis (NFA). This introduces stochastic latent variables into the decision process to provide robust UQ. The emphasis on non-negativity and sparsity aims to learn more interpretable and disentangled decision processes.

A primary limitation of last-layer methods is their restricted scope. The Credal Net appoarch force the model output an interval, but they themselves are still single-model approah which fails to capture accurate EU. The second approach approximates the total epistemic uncertainty, the mutual information $\mathcal{I}(y;\theta|\mathbf{x},\mathcal{D})$, by considering uncertainty only in the head:

$$\mathcal{I}(y;\theta|\mathbf{x},\mathcal{D}) \approx \mathcal{I}(y;\theta_{head}|\mathbf{x},\mathcal{D}) \tag{3}$$

This assumes the feature extractor is deterministic and perfect. Consequently, uncertainty that manifests in the lower-level features processed by the frozen encoder $g_{\theta_{enc}}$ may be inadequately propagated or ignored, representing a trade-off between computational efficiency and a complete uncertainty picture.

Full Retraining Approaches Methods that train all model parameters $\theta = \{\theta_{enc}, \theta_{head}\}$ with UQ considerations fall under this sub-category. The canonical example is a full **Bayesian Neural Network (BNN)** Neal (2012); MacKay (1992), which aims to infer the posterior distribution over all model weights, $p(\theta|\mathcal{D})$. The predictive distribution is found by marginalizing over this complete posterior:

$$p(y|\mathbf{x},\mathcal{D}) = \int p(y|\mathbf{x},\theta)p(\theta|\mathcal{D})d\theta \tag{4}$$

The epistemic uncertainty is then the full mutual information term $\mathcal{I}(y;\theta|\mathbf{x},\mathcal{D})$. While theoretically robust, this approach is the most resource-intensive and often impractical for very large PTMs.

1.3.2 Model-Only Access (Parameters/Weights, No Data for Fine-tuning)

This category is paramount for PTM scenarios where model weights are accessible but the original training data is not. These methods analyze or perturb the existing model parameters θ without requiring retraining.

Gradient-Based Methods A novel gradient-based approach quantifies epistemic uncertainty in PTMs without needing data or model modifications Wang and Ji (2024). The core intuition is that for an out-of-distribution (OOD) input $\mathbf{x}$, the model parameters are far from optimal, leading to large gradients. This uncertainty can be quantified by the norm of the gradient of the log-likelihood with respect to the parameters:

$$\mathcal{U}_{\text{grad}}(\mathbf{x}) = \|\nabla_\theta \log p(y_{pred}|\mathbf{x},\theta_{MAP})\|_2 \tag{5}$$

where y_{pred} is the model's prediction for $\mathbf{x}$ and θ_{MAP} are the fixed pre-trained weights. A larger gradient norm signifies higher epistemic uncertainty. The method is enhanced with techniques like layer-selective weighting and gradient smoothing via input perturbations to produce a more robust uncertainty score.

Parameter-Perturbation Methods These techniques introduce small perturbations to the learned parameters and observe the effect on the output. The variability of the outputs serves as an indicator of epistemic uncertainty. This is often motivated as a simple approximation of a Bayesian posterior. For example, one can sample multiple parameter vectors by adding small Gaussian noise to the MAP estimate:

$$\tilde{\theta}_i \sim \mathcal{N}(\theta_{MAP}, \sigma^2 I) \quad \text{for } i = 1, \dots, T \tag{6}$$

The epistemic uncertainty can then be estimated from the resulting set of predictions, for instance, by computing their mutual information. Proposition 3.5 in Wang and Ji (2024) establishes a mathematical bridge between these approaches, showing the expected gradient norm can upper-bound the uncertainty derived from small perturbations.

Laplace Approximation (LA) The Laplace Approximation is a classic technique to create a post-hoc Gaussian approximation of the true weight posterior $p(\theta|\mathcal{D})$ MacKay (1992); Daxberger et al. (2021). The approximation is centered at the MAP estimate θ_{MAP}:

$$p(\theta|\mathcal{D}) \approx \mathcal{N}(\theta|\theta_{MAP}, H^{-1}) \tag{7}$$

where H is the Hessian of the negative log-posterior evaluated at θ_{MAP}, $H = -\nabla_\theta^2 \log p(\theta|\mathcal{D})|_{\theta=\theta_{MAP}}$. A key advantage is its post-hoc nature, converting a standard NN into an approximate BNN with relative computational ease.

To scale LA to deep architectures, several approximations are essential:

- **Scope of Inference:** LA can be applied to all weights or, more practically, to a subset, such as the last layer only (last-layer LA) Kristiadi et al. (2020).
- **Hessian Approximation:** The exact Hessian is intractable. It is commonly replaced by the Generalized Gauss-Newton (GGN) matrix or further factorized structures like Kronecker-Factored Approximate Curvature (KFAC) Martens and Grosse (2015).
- **Hyperparameter Tuning:** The prior precision is a key hyperparameter that can be tuned post-hoc on a small validation set by optimizing the marginal likelihood Daxberger et al. (2021).
- **Approximate Predictive Distribution:** The integral for the predictive distribution is approximated, often by linearizing the network's output around θ_{MAP} (the delta method) or via Monte Carlo sampling from the Gaussian posterior.

1.3.3 No Model or Data Access (Black-Box / Input-Space Methods)

These methods are designed for the most restrictive scenarios with no access to model parameters or training data. UQ is derived by observing the model's input-output behavior.

Input Clarification Methods Particularly relevant for Large Language Models (LLMs), Input Clarification Ensembling probes uncertainty by generating multiple "clarifications" or rephrasings of a single input query Zhang et al. (2023). Let $\mathcal{C}(\mathbf{x}) = \{\mathbf{x}'_1, \dots, \mathbf{x}'_T\}$ be a set of rephrasings for input $\mathbf{x}$. These are fed to the model, and the resulting predictions are ensembled. This method decomposes uncertainty as follows:

- **Aleatoric Uncertainty**, arising from ambiguity in the original prompt $\mathbf{x}$, is estimated by the mutual information between the prediction Y and the clarification variable C. It measures how much the output depends on the specific rephrasing, indicating input ambiguity.

$$\mathcal{U}_A \approx \mathcal{I}(Y; C|\mathbf{x}) = \mathcal{H}\left[\frac{1}{T}\sum_{i=1}^{T} p(y|\mathbf{x}'_i)\right] - \frac{1}{T}\sum_{i=1}^{T} \mathcal{H}[p(y|\mathbf{x}'_i)] \tag{8}$$

- **Epistemic Uncertainty**, reflecting the model's intrinsic knowledge gaps, is estimated by the expected predictive entropy across the different clarifications. It captures the model's confusion even for specific, clarified prompts.

$$\mathcal{U}_E \approx \mathbb{E}_C[\mathcal{H}[Y|\mathbf{x}, C]] = \frac{1}{T}\sum_{i=1}^{T} \mathcal{H}[p(y|\mathbf{x}'_i)] \tag{9}$$

The flip is reasonable and understandable.

1. **Aleatoric** $\mathbb{I}[Y, C \mid X]$: If X is clear and specific, then it should be low. If Y heavily depends on C, it means the raw input X results in high uncertainty on the output.
2. **Epistemic** $\mathbb{E}_C[\mathbb{H}[Y \mid X, C]]$: If, given X and its different kinds of clarifications $\{C_i\}_{i=1}^{n}$, the entropy $\mathbb{H}[Y \mid X, C]$ is still high, this means the model is not familiar with the topic of X. That is, the model needs more knowledge about this topic.

This approach is a notable UQ technique for PTMs that operates purely in the input space. However, it presents several limitations and points for consideration:

- **Limitations:** The method relies on a single model, lacking the parameter diversity of true ensembles. Furthermore, the decomposition lacks rigorous mathematical proof connecting its terms to formal definitions of uncertainty, and it assumes the input prompt is the only source of aleatoric uncertainty.
- **Further Considerations:** Conceptually, the technique is similar to other input perturbation and test-time augmentation methods. An open question remains as to whether it can be formally integrated into a Bayesian deep learning framework.

Input Perturbation Methods Input perturbation methods apply small, often semantic-preserving changes to the input and measure output sensitivity. High sensitivity indicates high uncertainty. For LLMs, the SPUQ method generates perturbations via paraphrasing or altering inference temperature Chen and Zhao (2023). In computer vision, a similar principle is used in randomized smoothing, where Gaussian noise is added to inputs to analyze prediction stability Cohen et al. (2019). The core assumption is that the local flatness of the function $f(\mathbf{x})$ correlates with model certainty.

Test-Time Augmentation (TTA) TTA applies a set of data augmentations $\{a_1, \ldots, a_T\}$ to a single test input $\mathbf{x}$ and aggregates the predictions on the augmented versions $\{a_1(\mathbf{x}), \ldots, a_T(\mathbf{x})\}$ Shanmugam et al. (2021). This creates a pseudo-ensemble at inference time. The dispersion among the predictions is used as an uncertainty estimate. Mathematically, it is analogous to input perturbation, where the epistemic uncertainty is captured by the mutual information between the output and the specific augmentation applied:

$$\mathcal{U}_E \approx \mathcal{I}(y; a) = \mathcal{H}\left[\frac{1}{T}\sum_{i=1}^{T} p(y|a_i(\mathbf{x}))\right] - \frac{1}{T}\sum_{i=1}^{T} \mathcal{H}[p(y|a_i(\mathbf{x}))] \tag{10}$$

While practical, these black-box methods can be misled. A model might be robustly wrong, yielding low variance for an incorrect prediction. Therefore, the resulting scores measure output sensitivity, which is a useful but potentially uncalibrated proxy for true predictive uncertainty.

1.4 UQ for Single Models

This section focuses on methods for individual neural networks to estimate uncertainty without explicit ensembling. While some techniques, like MC Dropout, have an implicit ensembling nature, they operate on a single model architecture.

Evidential Deep Learning (EDL) Evidential deep learning is a powerful approach for uncertainty quantification that estimates both aleatoric and epistemic uncertainty by directly modeling the evidence derived from the data within the network's output. Unlike traditional Bayesian methods that treat model parameters as random variables, evidential deep learning treats the parameters of the target distribution as random variables that follow a conjugate prior. For instance, in classification tasks, a Dirichlet distribution is used, while a Gaussian-Inverse-Wishart distribution is typically employed for regression tasks.

In classification, the target variable y is assumed to follow a categorical distribution with parameter λ, i.e., $y \sim p(y \mid \lambda) = \text{Cat}(\lambda)$ where $\text{Cat}(\cdot)$ represents the categorical distribution. The parameter λ is treated as a random variable following a Dirichlet distribution, $\lambda \sim p(\lambda \mid \alpha(\mathbf{x}, \theta)) = \text{Dir}(\alpha(\mathbf{x}, \theta))$, where $\alpha(\mathbf{x}, \theta) = [\alpha_1, \alpha_2, \ldots, \alpha_C]^T$ is the output of a deterministic neural network parameterized by θ and C is the number of classes. The conditional probability $p(y \mid \mathbf{x}, \theta)$ is expressed as:

$$p(y \mid \mathbf{x}, \theta) = p(y \mid \alpha(\mathbf{x}, \theta)) = \int p(y \mid \lambda) p(\lambda \mid \alpha) d\lambda = \text{Cat}\left(\left\{\frac{\alpha_k}{\alpha_0}\right\}_{k=1}^{C}\right) \quad (2.44) \qquad (11)$$

where $\alpha_0 = \sum_{k=1}^{C} \alpha_k$ is the total evidence. The aleatoric uncertainty is typically captured by the entropy of $p(y \mid \mathbf{x}, \theta)$. The epistemic uncertainty is captured by the total evidence α_0. Lower total evidence reflects higher uncertainty due to a lack of sufficient knowledge or data. Additionally, epistemic uncertainty can be quantified using the mutual information $\mathcal{I}(y; \lambda \mid \mathbf{x}, \theta)$.

Evidential deep learning offers a computationally efficient approach to uncertainty quantification, as it does not rely on sampling methods like MCMC or variational inference. Instead, it provides direct estimates of uncertainty through the model's output, enabling uncertainty estimation within a single forward pass of the neural network. This makes it highly practical for large-scale applications.

Monte Carlo Dropout (MCD) Monte Carlo Dropout (MCD) leverages dropout for uncertainty estimation Gal and Ghahramani (2016). By keeping dropout active at inference time, one performs T stochastic forward passes for an input $\mathbf{x}$, each with a different dropout mask. This generates an ensemble of predictions $\{p(y|\mathbf{x}, \tilde{\theta}_t)\}_{t=1}^{T}$, where each $\tilde{\theta}_t$ represents a thinned version of the full network parameters θ. This process approximates Bayesian inference over the model's weights. The predictive distribution is the ensemble average, $p(y|\mathbf{x}) \approx \frac{1}{T} \sum_{t=1}^{T} p(y|\mathbf{x}, \tilde{\theta}_t)$. Epistemic uncertainty is quantified by the mutual information across the predictive samples:

$$\mathcal{U}_E \approx \mathcal{H}\left[\frac{1}{T} \sum_{t=1}^{T} p(y|\mathbf{x}, \tilde{\theta}_t)\right] - \frac{1}{T} \sum_{t=1}^{T} \mathcal{H}[p(y|\mathbf{x}, \tilde{\theta}_t)] \qquad (12)$$

While simple to implement, standard MCD can produce poorly calibrated uncertainty estimates. Research has thus focused on improvements, such as employing uncertainty-aware loss functions during training to improve calibration Yoo et al. (2022).

Other Single-Pass Bayesian Approximations Beyond EDL and MCD, other methods aim to approximate Bayesian inference in a single forward pass. One notable example is **Deterministic Uncertainty Quantification (DUQ)** Van Amersfoort et al. (2020). DUQ replaces the final linear layer of a network with a radial basis function (RBF) layer. For an input $\mathbf{x}$, the model produces a feature embedding $g_\theta(\mathbf{x})$, and the final layer computes the similarity to a set of learnable class centroids $\{c_k\}_{k=1}^{C}$. Confidence is derived from these similarities. DUQ uses a two-sided gradient penalty during training to enforce a clear separation in the feature space between in-distribution and out-of-distribution examples, enabling reliable OOD detection in a single deterministic pass.

1.5 Other Categorizations (e.g., Bayesian vs. Non-Bayesian, Output-space vs. Parameter-space vs. Input-space)

The categorization based on model and data access is particularly useful for PTMs, but UQ methods can also be viewed through other lenses, providing a richer taxonomy.[2, 11] These perspectives are not mutually exclusive and often intersect.

- **Bayesian vs. Non-Bayesian:**
  - **Bayesian methods** explicitly model uncertainty over model parameters or predictions using Bayes' theorem. They typically involve defining prior distributions over parameters, likelihood functions for the data, and then computing or approximating posterior distributions. Examples include BNNs (Deep Ensemble), Laplace Approximation, MC Dropout (as a Bayesian approximation), and BNDL.

    - **Non-Bayesian (or Frequentist) methods** define and quantify uncertainty without necessarily relying on explicit prior distributions for parameters. This category includes techniques like direct prediction of uncertainty intervals without Bayesian formulations, and some forms of TTA.

- **Space of Operation:**
    - **Parameter-space methods** define uncertainty primarily with respect to the model's parameters (weights and biases). The goal is often to estimate a distribution over these parameters. BNNs, LA, parameter perturbation techniques, and gradient-based methods that analyze gradients with respect to parameters fall into this category.[9, 11, 20]
    - **Output-space methods** define uncertainty directly on the model's outputs or predictions. EDL, which predicts parameters of a distribution over outputs, and methods that directly output prediction intervals are examples.
    - **Input-space methods** derive uncertainty estimates by analyzing the model's behavior in response to variations or augmentations of the input data. TTA, input perturbation methods, and input clarification techniques operate in this space.

Understanding these intersecting categorizations is crucial. For example, Last-Layer Laplace Approximation can be classified as 'Full Access' (if training data is used for the last layer) or 'Model-Only' (if applied post-hoc to a PTM using only weights for the Hessian approximation), 'Bayesian' in its principle, and 'Parameter-space' in its operation (approximating the posterior of last-layer weights). Similarly, TTA is typically 'No Model/Data Access' (black-box), can have non-Bayesian (simple averaging) or Bayesian interpretations (e.g., modeling acquisition priors), and operates in 'Input-space'. A comprehensive grasp of UQ requires considering these multiple dimensions, as they highlight different facets and assumptions of the various techniques.

1.6 Deep Ensembles

Deep Ensembles are a conceptually simple yet empirically powerful UQ method. The standard approach involves training multiple (typically 5 to 10) neural networks with the same architecture independently. Independence in training is encouraged by using different random initializations for weights and biases, and often by shuffling the training data differently for each member of the ensemble. At inference time, an input is passed through all ensemble members, and their predictions are aggregated, usually by averaging. The variance among the predictions of the ensemble members for a given input is then used as a measure of epistemic uncertainty. If each ensemble member is trained to predict the parameters of a probability distribution (e.g., the mean and variance of a Gaussian for regression tasks, or parameters of a categorical distribution for classification), then aleatoric uncertainty can also be captured by averaging these predicted distributional parameters.

Deep Ensembles are often considered a strong baseline in UQ due to their robust empirical performance across various tasks and datasets. They are known for their ability to capture multi-modality in the function space, as different ensemble members may converge to different local minima in the highly non-convex loss landscape of DNNs. However, their primary drawback is the significant computational cost associated with training and storing multiple independent models, which can be prohibitive for very large networks or resource-constrained environments.The precise reasons for their strong performance are still an active area of research, though exploration of diverse modes in the loss landscape is a commonly cited factor.

From a Bayesian perspective, Deep Ensembles can be viewed as an approximate Bayesian inference method, where the ensemble of networks approximates the Bayesian model average over the posterior distribution of models. Each network in the ensemble can be thought of as a sample from an approximate posterior. An extension, Credal Deep Ensembles (CreDEs), combines the ensemble paradigm with Credal-Set Neural Networks, where each member outputs interval-based predictions that are then aggregated. Deep Ensembles inherently require full model and data access for training the multiple constituent models from scratch.

1.7 Comparative Analysis: Strengths, Weaknesses, and Applicability

The diverse array of UQ methodologies surveyed highlights that there is no single "silver bullet" solution. The choice of an appropriate method is a complex decision contingent on the specific application, available resources, and desired uncertainty granularity. Navigating these trade-offs requires a clear comparison along key axes, as summarized in recent surveys and benchmark studies Abdar et al. (2021); Ovadia et al. (2019).

Table 1 provides a comparative overview of the prominent UQ methods discussed. This analysis considers factors ranging from the type of uncertainty captured to the computational overhead and suitability for pre-trained models.

Table 1: Comparative analysis of major UQ methodologies.

Method	Uncertainty Type	Train Cost	Inference Cost	PTM Suitability	Theoretical Rigor	Primary Limitation
Deep Ensembles Lakshminarayanan et al. (2017)	Epistemic + Aleatoric	Very High (M models)	High (M passes)	Low (costly)	Heuristic	High computational cost
Full BNN (MCMC) Neal (2012)	Epistemic + Aleatoric	Extremely High	Extremely High	Very Low	Very High	Intractable for large models
MC Dropout Gal and Ghahramani (2016)	Epistemic (approx.)	Low (standard train)	Moderate (T passes)	High	Moderate (approx. VI)	Inconsistent calibration
Laplace Approx. Daxberger et al. (2021)	Epistemic (approx.)	Low (standard train)	Low (1 pass + calc)	High (Post-hoc)	Moderate (Gaussian assumption)	Hessian is complex
Evidential DL Sensoy et al. (2018); Meiseles and Rokach (2023)	Epistemic + Aleatoric	Moderate (custom loss)	Very Low (1 pass)	Moderate (retrain head)	Low-Moderate	Theoretical critiques
Black-Box (TTA) Shanmugam et al. (2021)	Epistemic (proxy)	None (post-hoc)	Moderate (T passes)	Very High	Low (heuristic)	Proxy for uncertainty

This landscape underscores a persistent tension in UQ research. Methods with strong theoretical grounding, like full BNNs, remain computationally prohibitive for large-scale models. In contrast, empirically robust and conceptually simpler methods like Deep Ensembles are often the top performers in benchmarks but are equally costly Ovadia et al. (2019). This has motivated the development of approximations. MC Dropout is widely adopted for its simplicity, but its calibration can be inconsistent without enhancements. The Laplace Approximation offers a more principled post-hoc Bayesian treatment but is constrained by its Gaussian assumption and the complexity of the Hessian. Evidential Deep Learning promises efficient decomposition but faces critiques regarding the statistical validity of its uncertainty estimates Meiseles and Rokach (2023). Finally, Black-box methods are universally applicable but provide a less direct measure of model uncertainty, reflecting local input sensitivity rather than a holistic view of the parameter posterior. Selecting a UQ method therefore requires a careful, context-dependent balancing of these multifaceted trade-offs.

The following table (Table 1) provides a summarized comparison of various UQ methodologies discussed.

Table 2: Comparative Overview of Uncertainty Quantification Methodologies (Landscape)

Method Category	Specific Technique(s)	Core Principle	Access Req.	Uncertainty Types	Key Strengths	Key Weaknesses/ Limitations	Cost (Train/Inf)
Bayesian (Full BNN)	Variational Inference, MCMC	Full posterior approx. over weights	Full	Both, Disentangled	Theoretical grounding	High cost, Approx. errors	High/Hig
Bayesian (Approx.)	Laplace Approx. (LA)	Gaussian posterior approx. at MAP	Need Last layer's weights and gradients	Both, Disentangled	Efficient, Post-hoc for PTMs	Gaussian assumption, Hessian approx. needed	Low (post-hoc)/Low
Bayesian (Approx.)	MC Dropout (MCD)	Dropout as Bayesian approx.	Model (trained w/ dropout)	Both, Disentangled	Simple, Widely applicable	Calibration issues, Approx. quality varies	Low (if pretrained)/ Med (mult. passes)
Ensembles	Deep Ensembles	Diversity from independent training	Full	Both, Disentangled	Empirically strong, Conceptually simple	High cost (train & store multiple models)	High/Med-High
Last-Layer	Credal Ensembles (CreNets)	Interval probabilities via robust optimization	Full (for last layer)	Both, Disentangled (credal way)	Good for OOD, Captures epistemic shift	Only sees high-level features, Retrains last layer	Med/Low
Last-Layer	Bayesian Non-negative Decision Layers (BNDL)	Bayesian NFA for decision layer	Full (for last layer)	Both, Disentangled	Interpretability, Robust UQ	Only sees high-level features, Retrains last layer	Med/Low
Gradient-Based (PTM)	REGrad	Output gradients w.r.t. parameters as UQ	Model-Only	Mainly Epistemic	No data/retraining for PTMs, Theoretical links to BNNs	Sensitivity to gradient noise, Focus on epistemic	N/A (no train)/Low
Parameter Perturbation	-	Output sensitivity to weight perturbations	Model-Only	Both, Disentangled	Simple concept, Links to BNNs	Defining perturbation strategy	N/A (no train)/Med

Continued on next page

Table 2 – continued from previous page

Method Category	**Specific Technique(s)**	**Core Principle**	**Access Req.**	**Uncertainty Types**	**Key Strengths**	**Key Weaknesses/ Limitations**	**Cost (Train/Inf)**
Evidential Deep Learning (EDL)	-	Learns parameters of evidential (meta) distribution	Full (training w/ EDL loss)	Both, Disentangled	Single pass, Efficient	Theoretical critiques (non-vanishing epistemic U), Model-dependent aleatoric U	Med/Low
Input-Space (Clarification)	Input Clarification Ensembling (LLMs)	Ensemble over clarified inputs	Black-Box	Both, Disentangled (input ambiguity vs. model knowledge)	Good for LLM ambiguity, Interpretable UQ source	Needs clarification LLM, LLM-specific	Low/Med
Input-Space (Perturbation)	SPUQ (LLMs), ClaudesLens (Vision)	Output sensitivity to input perturbations	Black-Box	Both, Disentangled	Versatile, No model access needed	Sensitivity to perturbation type, May reflect local stability	Low/Med
Input-Space (TTA)	Standard TTA, Intelligent TTA	Aggregate predictions over augmented inputs	Black-Box	Both, Disentangled	Simple, Improves robustness	Cost of multiple inferences, Choice of augmentations	Low/Med-High

1.8 Experimental Evaluation

To empirically validate and compare the practical performance of several key Uncertainty Quantification (UQ) methodologies discussed in this survey, we conducted a series of experiments on standard image classification benchmarks. The evaluation focuses on two primary aspects: predictive accuracy on in-distribution (ID) data and the ability to detect out-of-distribution (OOD) samples, a critical test for the reliability of uncertainty estimates.

1.8.1 Experimental Setup

Datasets and Model Architecture The experiments were performed on the **CIFAR-10** and **CIFAR-100** datasets. For the OOD detection task, the **Street View House Numbers (SVHN)** dataset was used as the source of OOD samples. For all experiments, a **ResNet-18** architecture was used as the base network to ensure a fair comparison across methods.

Methods Compared We evaluated five distinct approaches, representing a spectrum of the techniques surveyed:

- **Single Model (Baseline):** A standard ResNet-18 model trained with maximum likelihood, representing performance without any explicit UQ method.
- **Deep Ensemble (Baseline):** An ensemble of five ResNet-18 models, each trained independently with different random initializations. This serves as a strong, albeit computationally expensive, baseline for UQ performance Lakshminarayanan et al. (2017).
- **LLLA (Last-Layer Laplace Approx.):** A post-hoc application of the Laplace approximation to the final layer of a pre-trained ResNet-18, representing an efficient parameter-space UQ method Daxberger et al. (2021); Kristiadi et al. (2020).
- **Evidential DL:** A ResNet-18 model trained with an evidential loss function to directly output the parameters of a Dirichlet distribution, enabling single-pass uncertainty estimation Sensoy et al. (2018).
- **ABNN (Approximate BNN):** An approximate Bayesian Neural Network, implemented as MakeMe-BNN, which offers a scalable approach to approximate the weight posterior.

Evaluation Metrics Performance was assessed using the following metrics:

- **Test Accuracy (%):** The standard classification accuracy on the test set of the in-distribution dataset (CIFAR-10 or CIFAR-100).
- **AUROC:** The Area Under the Receiver Operating Characteristic curve for the OOD detection task (distinguishing between CIFAR and SVHN). A higher value indicates better separability.
- **AUPR:** The Area Under the Precision-Recall curve, also for the OOD detection task. This metric is particularly informative when there is a class imbalance between ID and OOD samples.

1.8.2 Results and Analysis

The results of our comparative experiments on both CIFAR-10 and CIFAR-100 are summarized in Table 3 and Table 4, respectively.

Analysis of Results The experimental results provide several key insights into the practical trade-offs of the evaluated UQ methods.

- **Predictive Accuracy:** Across both datasets, the **Deep Ensemble** method achieves the highest test accuracy. This confirms its status as a powerful, high-performance baseline, benefiting from the aggregation of multiple diverse models. The single-model UQ methods (LLLA, ABNN) generally outperform the standard single model baseline, demonstrating that introducing diversity in parameter space based on point estimation does not necessarily compromise, and can even enhance, predictive performance. Evidential DL shows a comparable test results in accuracy compared to the baseline on both datasets.

Table 3: Comparative results on **CIFAR-10** (ID) vs. **SVHN** (OOD) using a ResNet-18 architecture. Best performance in each column is in **bold**.

Method	Test Accuracy (%)	AUROC	AUPR
Single Model	94.1	0.8464	0.8586
Deep Ensemble	**96.0**	**0.8842**	**0.9064**
LLLA	95.3	0.8653	0.8875
Evidential DL	94.7	0.8804	0.8972
ABNN	95.4	0.8742	0.8975

Table 4: Comparative results on **CIFAR-100** (ID) vs. **SVHN** (OOD) using a ResNet-18 architecture. Best performance in each column is in **bold**.

Method	Test Accuracy (%)	AUROC	AUPR
Single Model	74.1	0.7754	0.7668
Deep Ensemble	**76.3**	**0.8257**	**0.8193**
LLLA	75.6	0.8077	0.8108
Evidential DL	74.3	0.7967	0.8129
ABNN	75.5	0.8056	0.8097

- **OOD Detection:** The results for AUROC and AUPR clearly demonstrate the primary value of UQ. The standard **Single Model** is the poorest performer in OOD detection, highlighting its tendency to produce overconfident predictions for unfamiliar inputs. In contrast, all specialized UQ methods show a marked improvement. This means, no matter what kinds of diversity are added to the model parameter, they will enhance the model ability in indentifying ood data, showing the effectiveness of introducing uncertainty.

In summary, these experiments confirm that while Deep Ensembles remain a gold standard for raw performance, more efficient single-model techniques like LLLA, Evidential DL, and ABNN provide substantial and crucial benefits for uncertainty estimation, they do not harm the model prediction, but help the model with relialbe uncertainty estimation. They effectively bridge the gap between a naive single model and a full ensemble, offering a practical pathway to deploying safer and more reliable models, which is especially critical in the context of large PTMs where ensembling is often computationally infeasible.

1.9 Historical Development and Trends in UQ for Deep Learning

Uncertainty Quantification as a general scientific discipline has a long history, with Bayesian methods, in particular, tracing their origins back several centuries and undergoing significant evolution throughout the 20th and 21st centuries, profoundly influencing fields including Artificial Intelligence. The application of UQ principles to neural networks, and subsequently to deep learning models, has followed a distinct trajectory.

Early research explored Bayesian Neural Networks (BNNs), but scaling these methods to the increasingly deep and complex architectures of modern DNNs posed significant computational and inferential challenges. This led to the development and popularization of more practical and scalable approximation techniques. Monte Carlo Dropout, proposed by Gal and Ghahramani in 2016, offered an accessible way to obtain uncertainty estimates from standard networks by leveraging dropout during inference. Around the same time, Deep Ensembles, introduced by Lakshminarayanan et al. in 2017, demonstrated strong empirical performance by simply training multiple independent networks. These methods became widely adopted due to their relative ease of implementation and effectiveness.

More recently, with the proliferation of large Pre-trained Models (PTMs), the focus has shifted towards developing UQ techniques that can operate with limited access to training data or model internals, or that can be applied post-hoc without expensive retraining. This includes last-layer methods, Laplace approximation for PTMs, gradient-based approaches, and various input-space techniques.

Several key trends characterize the evolution of UQ in deep learning:

- **Adaptation to PTMs:** A clear shift from methods requiring full retraining towards post-hoc, efficient, or model-access-only techniques suitable for large pre-trained foundational models.
- **Emphasis on Safety and Reliability:** An increasing demand for robust and reliable UQ, driven by the deployment of DNNs in safety-critical applications where understanding model confidence is non-negotiable.
- **Focus on Calibration and Interpretability:** Growing research efforts are dedicated to improving not just the magnitude of uncertainty estimates but also their calibration (i.e., ensuring that a predicted $p\%$ confidence corresponds to $p\%$ accuracy) and interpretability (i.e., understanding what the uncertainty signifies).
- **Standardization and Benchmarking:** The development of standardized benchmarks, datasets, and evaluation metrics for UQ methods is crucial for rigorous comparison and progress in the field.

The historical development of UQ in deep learning often exhibits a cyclical pattern: theoretically elegant but computationally intensive methods (like full BNNs) are proposed, followed by the development of more practical and scalable approximations (such as MCD, LA, and Ensembles). These approximations, in turn, reveal their own limitations (e.g., issues with calibration, residual computational costs, or restrictive assumptions), which then spur further research into refining these techniques or developing entirely new paradigms (like EDL or specialized gradient-based methods for PTMs). This iterative process of proposing theory, developing practical approximations, identifying their shortcomings, and then seeking further refinements or novel approaches signifies a maturing field actively grappling with the dual imperatives of theoretical rigor and practical applicability in the quest for trustworthy AI.

2 PNC Predictor Summary

2.1 Introduction

This part summarizes the methodology and algorithms proposed in the work by Huang et al., which aims at efficient uncertainty quantification (UQ) for over-parameterized neural networks (NNs). The authors leverage neural tangent kernel (NTK) theory and light-computation resampling methods to provide statistically guaranteed UQ at low computational cost.

2.2 Problem Setup

Suppose we have data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, where $(X, Y) \sim \pi$ with $X \in \mathbb{R}^d$, $Y \in \mathbb{R}$. The goal is to learn a predictor $h : \mathbb{R}^d \to \mathbb{R}$ minimizing the risk

$$R_\pi(h) = \mathbb{E}_{(X,Y)\sim\pi}\left[L(h(X), Y)\right],$$

where L is typically the squared error loss:

$$L(y', y) = (y' - y)^2.$$

2.2.1 Sources of Epistemic Uncertainty

The authors decompose epistemic uncertainty into:

$$\begin{aligned} \text{Model approximation error: } UQ_{\text{AE}} &= h^* - h_B^*, \\ \text{Data variability: } UQ_{\text{DV}} &= \hat{h}_n^* - h^*, \\ \text{Procedural variability: } UQ_{\text{PV}} &= \hat{h}_{n,\gamma} - \hat{h}_n^*, \end{aligned}$$

where h_B^* is the Bayes predictor, h^* is the limiting predictor with infinite data, $\hat{h}_n^*$ is the ideal deep ensemble predictor, and $\hat{h}_{n,\gamma}$ is the predictor obtained by a single training run with randomness γ from initialization or stochastic training dynamics.

2.3 Methodology

2.3.1 NTK Characterization of Neural Networks

Under NTK theory, for sufficiently wide networks trained with gradient descent, the predictor admits the form:

$$\hat{h}_{n,\theta^b}(x) = s_{\theta^b}(x) + K(x,X)^\top (K(X,X) + \lambda_n nI)^{-1}(y - s_{\theta^b}(X)),$$

where

- s_{θ^b} is the output of the untrained network at initialization θ^b,
- K is the NTK kernel,
- λ_n is a regularization parameter.

2.3.2 Procedural-Noise-Correcting (PNC) Predictor

The PNC predictor is defined as:

$$\hat{h}_{\text{PNC}}(x) = \hat{h}_{n,\theta^b}(x) - \phi_{n,\theta^b}(x),$$

where $\phi_{n,\theta^b}(x) = \phi'_{n,\theta^b}(x) - \bar{s}(x)$, with ϕ'_{n,θ^b} being an auxiliary network trained on artificial data $\{(x_i, \bar{s}(x_i))\}$ and $\bar{s}(x) = \mathbb{E}[s_{\theta^b}(x)]$.

2.3.3 Confidence Interval Construction

Two efficient methods are proposed:

- **PNC-enhanced batching**: Divide data into m' batches, compute batch predictors $\hat{h}_j$, and form a t-statistic for confidence intervals.
- **PNC-enhanced cheap bootstrap**: Resample data R times, compute predictors on resamples, and form confidence intervals based on their variability.

2.4 Main Theoretical Result

Theorem 1 (Large-sample normality of PNC predictor) *Under suitable regularity conditions, as* $n \to \infty$,

$$\sqrt{n}\left(\hat{h}_{PNC}(x) - h^*(x)\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2(x)),$$

where $\sigma^2(x)$ *is given by*

$$\sigma^2(x) = \int IF^2(z; T_1, \pi)(x) d\pi(z),$$

with IF *denoting the influence function of the functional* T_1 *associated with kernel ridge regression.*

2.5 Conclusion

The paper presents an innovative framework for UQ in over-parameterized NNs that:

- Efficiently eliminates procedural variability using a single auxiliary network.
- Provides asymptotically valid confidence intervals via batching or cheap bootstrap.
- Reduces computational cost compared to deep ensembles while maintaining statistical rigor.

2.6 My Understanding and Derivations

The PNC-predictor framework provides a strong theoretical foundation for efficient UQ. However, its reliance on resampling methods for constructing confidence intervals motivates exploration into more direct, model-based approaches to uncertainty estimation. Building on this work, we propose a direction for future research based on evidential deep learning principles.

2.6.1 Modeling Predictor Output as a Distribution

Instead of treating the network output as a point estimate, we can model it as a probability distribution. For a given input x, let the predictor's output follow a Gaussian distribution:

$$\hat{h}_\theta(x) \sim \mathcal{N}(\mu(x,\theta), \sigma^2(x,\theta)).$$

This allows for a natural decomposition of uncertainty. For the final PNC predictor, $\hat{h}_{\text{PNC}}$, which we assume is also Gaussian, $\hat{h}_{\text{PNC}} \sim \mathcal{N}(\mu_{\text{PNC}}, \sigma^2_{\text{PNC}})$, we can define:

- **Aleatoric Uncertainty**: $\mathbb{E}[\sigma^2_{\text{PNC}}]$, capturing inherent noise in the data.
- **Epistemic Uncertainty**: $\text{Var}[\mu_{\text{PNC}}]$, capturing the model's uncertainty about the true function.

2.6.2 Challenges and Proposed Solutions

A key challenge in this approach is characterizing the distribution of the PNC predictor, $\hat{h}_{\text{PNC}} = \hat{h}_{n,\theta^b} - \phi_{n,\theta^b}$. While both $\hat{h}_{n,\theta^b}$ and the auxiliary network ϕ_{n,θ^b} can be modeled as Gaussians, they are not independent. The variance of the difference is thus:

$$\text{Var}(\hat{h}_{\text{PNC}}) = \text{Var}(\hat{h}_{n,\theta^b}) + \text{Var}(\phi_{n,\theta^b}) - 2\text{Cov}(\hat{h}_{n,\theta^b}, \phi_{n,\theta^b}),$$

where the covariance term is non-trivial.

To address this, we can draw inspiration from evidential deep learning. One promising approach is to model the predictive distribution using a Normal-Inverse-Gamma (NIG) distribution:

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

This models a distribution over the parameters of the output Gaussian, providing a complete solution for UQ, as demonstrated in Deep Evidential Regression. A potential simplification, particularly if epistemic uncertainty is the primary focus, is to only model the mean μ as a random variable while keeping the variance σ^2 deterministic.

2.6.3 Extension to Classification

Finally, the current formulation of the PNC-predictor involves the direct subtraction of network outputs, which is well-defined for regression. Extending this concept to classification tasks is not straightforward, as subtracting probabilities or logits does not have a clear probabilistic interpretation. Future work should investigate equivalent operations in the space of probability measures to adapt the PNC framework for classification uncertainty.

2.7 Derivations: Uncertainty Quantification with PNC-Predictor

Model and Variable Definitions

- **Single Model** $\hat{h}_{n,\theta_b}$: A single neural network model trained on a dataset of size n, with a specific parameter initialization θ_b.
- **Mean Prediction** $\hat{s}(x)$: The expected prediction of a network at a point x, averaged over the distribution of initializations. It is approximated by the sample mean over m different initializations:

$$\hat{s}(x) = \mathbb{E}_{\theta_b}[s_{\theta_b}(x)] \approx \frac{1}{m}\sum_{i=1}^{m} s_{\theta_{b_i}}(x)$$

- **Auxiliary Network** φ_{n,θ_b}: An auxiliary function defined as $\varphi_{n,\theta_b}(x) = \varphi'(x) - \hat{s}(x)$.
- **Final Predictor** h^*: The final, corrected predictor is given by:

$$h^* = \hat{h}_{n,\theta_b} - \varphi_{n,\theta_b}$$

Predictor Formulation

Let $\mathbf{X}$ and $\mathbf{y}$ be the training inputs and outputs. Let $k(x, \mathbf{X})$ be a kernel vector between a test point x and the training inputs, and $K(\mathbf{X}, \mathbf{X})$ be the kernel matrix for the training data. The expression for the final predictor h^* is given as:

$$h^*(x) = \hat{s}(x) + k(x, \mathbf{X})^T (K(\mathbf{X}, \mathbf{X}) + \lambda n I)^{-1} (\mathbf{y} - \hat{\mathbf{s}}(\mathbf{X}))$$

This can be written more compactly by defining a kernel weight vector $\mathbf{k}(x)^T$:

$$\begin{aligned} \mathbf{k}(x)^T &= k(x, \mathbf{X})^T (K(\mathbf{X}, \mathbf{X}) + \lambda n I)^{-1} \\ h^*(x) &= \hat{s}(x) + \mathbf{k}(x)^T (\mathbf{y} - \hat{\mathbf{s}}(\mathbf{X})) \end{aligned}$$

Epistemic Uncertainty Derivation

The epistemic uncertainty is measured by the variance of the predictor's output, $\text{var}(\hat{h}^*)$, over the distribution of model initializations.

$$\begin{aligned} \text{var}(\hat{h}^*(x)) &= \text{var}\left(\hat{s}(x) + \mathbf{k}(x)^T (\mathbf{y} - \hat{\mathbf{s}}(\mathbf{X}))\right) \\ &= \text{var}\left(\hat{s}(x) - \mathbf{k}(x)^T \hat{\mathbf{s}}(\mathbf{X})\right) \\ &= \text{var}(\hat{s}(x)) + \text{var}(\mathbf{k}(x)^T \hat{\mathbf{s}}(\mathbf{X})) - 2\text{cov}(\hat{s}(x), \mathbf{k}(x)^T \hat{\mathbf{s}}(\mathbf{X})) \\ &= \text{var}(\hat{s}(x)) + \mathbf{k}(x)^T \text{var}(\hat{\mathbf{s}}(\mathbf{X})) \mathbf{k}(x) - 2\mathbf{k}(x)^T \text{cov}(\hat{s}(x), \hat{\mathbf{s}}(\mathbf{X})) \end{aligned}$$

Using the relation $\text{var}(\hat{s}) = \frac{1}{m}\text{var}(s)$, the final expression is:

$$\text{var}(\hat{h}^*(x)) = \frac{1}{m}\left[\text{var}(s(x)) + \mathbf{k}(x)^T \text{var}(\mathbf{s}(\mathbf{X})) \mathbf{k}(x) - 2\mathbf{k}(x)^T \text{cov}(s(x), \mathbf{s}(\mathbf{X}))\right]$$

Asymptotic Analysis

The behavior of the uncertainty as the training dataset size $n \to \infty$:

1. The term $\text{var}(s(x))$ does not depend on n.
2. The terms involving $\mathbf{k}(x)$ diminish as n increases, because the matrix inverse term scales as $O(\frac{1}{n})$:

$$\begin{aligned} (K(\mathbf{X}, \mathbf{X}) + \lambda n I)^{-1} &= \left(n\left(\frac{1}{n}K(\mathbf{X}, \mathbf{X}) + \lambda I\right)\right)^{-1} \\ &= \frac{1}{n}\left(\frac{1}{n}K(\mathbf{X}, \mathbf{X}) + \lambda I\right)^{-1} \sim O\left(\frac{1}{n}\right) \end{aligned}$$

For a sufficiently large dataset, the epistemic uncertainty converges to:

$$\lim_{n \to \infty} \text{var}(\hat{h}^*(x)) = \frac{1}{m}\text{var}(s(x))$$

3 Uncertainty Quantification with Credal Ensemble and Credal Wrapper

In predictive modeling, it is often not enough to provide a point estimate; quantifying the uncertainty associated with a prediction is crucial for reliable decision-making. This uncertainty can be broadly categorized into two types: aleatoric uncertainty, which is inherent in the data due to noise or randomness, and epistemic uncertainty, which arises from the model's lack of knowledge or limited training data.

While traditional probabilistic models output a single probability distribution over the possible outcomes, this representation can be insufficient to distinguish between aleatoric and epistemic uncertainty. Credal sets, which are convex sets of probability distributions, offer a more expressive framework. A credal set $\mathcal{P}$ is defined as:

$$\mathcal{P} = \text{conv}\{p_1, p_2, \ldots, p_k\}$$

where $\{p_i\}$ is a set of probability distributions and conv denotes the convex hull. The size or "spread" of the credal set can be interpreted as a measure of epistemic uncertainty: a larger set indicates greater model uncertainty.

For a classification task with C classes, a credal set can be represented by lower and upper probability bounds for each class c:

$$\underline{p}_c = \min_{p \in \mathcal{P}} p(c) \quad \text{and} \quad \overline{p}_c = \max_{p \in \mathcal{P}} p(c)$$

The interval $[\underline{p}_c, \overline{p}_c]$ represents the range of plausible probabilities for class c. The width of this interval, $\overline{p}_c - \underline{p}_c$, reflects the degree of epistemic uncertainty for that class.

3.1 Credal Ensembles

The work by Wang et al. (2024a) introduces Credal-Set Neural Networks (CreNets), a novel neural network architecture designed to directly output a credal set. The key innovation lies in the final layers of the network and a custom loss function.

3.1.1 CreNet Architecture

For a C-class classification problem, a CreNet has $2C$ output nodes in its final layer. For each class c, there are two outputs: one for the midpoint m_c of the probability interval and one for its half-length h_c. The lower and upper probability bounds are then given by:

$$\underline{p}_c = m_c - h_c \quad \text{and} \quad \overline{p}_c = m_c + h_c$$

Constraints are imposed to ensure that these bounds form a valid credal set, i.e., $\sum_c \underline{p}_c \leq 1 \leq \sum_c \overline{p}_c$.

3.1.2 Loss Function

The training of a CreNet employs a composite loss function inspired by Distributionally Robust Optimization (DRO). The loss has two components:

1. **An "optimistic" component** that applies a standard cross-entropy loss to the upper probability vector $\overline{\mathbf{p}} = (\overline{p}_1, \ldots, \overline{p}_C)$. This encourages the model to be confident in its predictions, assuming the test distribution is similar to the training distribution.

$$\mathcal{L}_{\text{optimistic}} = -\sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \log(\overline{p}_{ic})$$

   where y_{ic} is the one-hot encoded true label for sample i.

2. **A "pessimistic" component** based on DRO, which aims to minimize the worst-case loss over a set of possible test distributions. This is applied to the lower probability vector $\underline{\mathbf{p}} = (\underline{p}_1, \ldots, \underline{p}_C)$ and encourages the model to be cautious, reflecting potential shifts in the data distribution. The DRO objective is:

$$\min_{\theta} \sup_{q \in \mathcal{Q}} \mathbb{E}_{x,y \sim q}[\mathcal{L}(f(x;\theta), y)]$$

   where $\mathcal{Q}$ is a set of distributions close to the empirical training distribution.

The total loss is a weighted sum of these two components, which trains the network to produce intervals whose widths reflect the epistemic uncertainty.

3.1.3 Credal Deep Ensembles (CreDEs)

To further improve performance and robustness, multiple CreNets are trained with different random initializations to form a Credal Deep Ensemble (CreDE). The final prediction is obtained by aggregating the output intervals from each member of the ensemble, for example, by averaging the lower and upper bounds.

3.2 Credal Wrapper

In contrast to designing a new network architecture, the Credal Wrapper proposed by Wang et al. (2024b) is a post-processing method that constructs a credal set from the outputs of a standard ensemble of models, such as a Deep Ensemble or a Bayesian Neural Network (BNN).

3.2.1 Constructing the Credal Set

Given an ensemble of M models, for a given input x, we obtain a set of M predictive probability distributions $\{p_1, p_2, \ldots, p_M\}$. The Credal Wrapper constructs a credal set from this collection by defining the lower and upper probability bounds for each class c as:

$$\underline{p}_c = \min_{i=1,\ldots,M} p_i(c) \quad \text{and} \quad \overline{p}_c = \max_{i=1,\ldots,M} p_i(c)$$

This credal set, defined by the convex hull of the ensemble's predictions, captures the disagreement among the models, which is a proxy for epistemic uncertainty.

3.2.2 Intersection Probability

To obtain a single, definite prediction from the credal set, the Credal Wrapper employs the intersection probability. This is a method for mapping a credal set back to a single probability distribution. The intersection probability p^* is the unique probability distribution that is "closest" to all distributions in the credal set, in a specific information-theoretic sense. For the credal set constructed from the ensemble, the intersection probability for class c is given by:

$$p_c^* = \frac{\overline{p}_c - \underline{p}_c + \sum_{j=1}^{C} \underline{p}_j - 1}{C - 1 - \sum_{j=1}^{C} (\overline{p}_j - \underline{p}_j)}$$

This provides a principled way to make a final decision while still having access to the full credal set for uncertainty analysis.

3.3 Empirical Results on OOD Detection

This section presents comprehensive empirical results to compare the practical performance of the discussed methods. The experiments cover two scenarios: CIFAR-10 (with a ResNet-18 backbone) and CIFAR-100 (with a ResNet-50 backbone). In both cases, the Street View House Numbers (SVHN) dataset served as the out-of-distribution (OOD) data. The models were fine-tuned for 100 epochs with early stopping to ensure good convergence. For the credal methods, test accuracy was calculated using the mid-points of the output intervals. It is important to note that the Credal Wrapper uses the exact same pre-trained models as the Deep Ensemble baseline, making it a direct test of the post-processing method's effectiveness.

Table 5: Performance comparison for OOD detection. Epistemic uncertainty is used as the OOD score. The best performance in each column is in bold.

	Cifar 10 (Resnet 18) vs SVHN			**Cifar 100 (Resnet 50) vs SVHN**		
Models	**Test Accuracy**	**AUROC**	**AUPRC**	**Test Accuracy**	**AUROC**	**AUPRC**
Deep Ensemble	95.95%	0.9683	0.9832	85.35%	0.8567	0.9227
Credal Ensemble	**96.43%**	**0.9813**	**0.9924**	**86.02%**	0.8520	0.8838
Single Credal Nets	95.78%	0.8646	0.9495	83.44%	0.7354	0.8186
Credal Wrapper	95.51%	0.9761	0.9877	84.93%	**0.8956**	**0.9488**

The results, summarized in Table 5, reveal interesting performance differences between the methods across the two datasets.

- On the CIFAR-10 vs SVHN task, the **Credal Ensemble** is the unequivocal top performer. It achieves the highest test accuracy and demonstrates superior OOD detection capabilities, leading in both AUROC and AUPRC scores. The Credal Wrapper also performs very well, significantly outperforming the standard Deep Ensemble in OOD detection.

- On the more complex CIFAR-100 vs SVHN task, the results are more nuanced. While the Credal Ensemble again achieves the highest in-distribution accuracy, the **Credal Wrapper** emerges as the clear winner for OOD detection, posting the best AUROC and AUPRC scores by a significant margin.

These findings suggest that the natively-trained Credal Ensemble excels on the simpler CIFAR-10 task, while the post-hoc Credal Wrapper provides a more robust OOD signal on the more challenging CIFAR-100 dataset.

3.4 Analysis of Prediction Strategies within the Credal Set

A practical question when using credal methods is how to derive a single-point prediction from the output interval for classification. While using the interval's mid-point is a common default, other strategies exist. The following experiment compares several prediction strategies for the Credal Wrapper method on in-distribution test data.

Table 6: In-distribution accuracy for different Credal Wrapper prediction strategies. The Deep Ensemble (simply averaging outputs) serves as a baseline. Best performance in each column is in bold.

	Cifar 10 (Resnet 18)		**Cifar 100 (Resnet 50)**	
Strategy	**Ensemble Size of 5**	**Ensemble Size of 30**	**Ensemble Size of 5**	**Ensemble Size of 30**
gray!10 Deep Ensemble (Simply Average)	**95.95%**	**96.10%**	85.35%	86.05%
Credal Wrapper (Mid Point)	95.91%	95.95%	**84.93%**	85.31%
Credal Wrapper (Lowest Entropy Point)	95.91%	95.95%	84.92%	85.31%
Credal Wrapper (Highest Entropy Point)	93.36%	88.35%	79.23%	70.07%
Credal Wrapper (Lower Bound)	95.89%	95.92%	84.84%	**85.44%**
Credal Wrapper (Upper Bound)	**95.98%**	**95.96%**	84.76%	85.22%

From the results in Table 6, we can draw several key conclusions:

1. **Choice has little impact on in-distribution accuracy.** For standard classification on test data from the same distribution as the training data, most strategies (Mid Point, Lowest Entropy, Lower/Upper Bound) yield accuracies that are very close to each other and to the Deep Ensemble baseline. The only exception is the "Highest Entropy Point," which corresponds to the most uncertain prediction and, as expected, performs significantly worse.
2. **Ensemble size affects the credal set.** As the ensemble size grows from 5 to 30, the credal set becomes larger because the minimum of all predictions tends to decrease and the maximum tends to increase. This widening of the probability intervals explains the dramatic performance drop of the "Highest Entropy Point" strategy with a larger ensemble—there is simply more room for an uncertain (and incorrect) prediction.
3. **Implications for robustness.** This analysis leads to a crucial question for future research: while the choice of prediction point seems minor for standard accuracy, how does it affect model robustness? It is plausible that on domain-shifted, corrupted, or adversarial data, one strategy might prove significantly more robust than others. Investigating which point's prediction is most stable in the face of such data perturbations is a valuable direction for future work.

3.5 Technical Conclusion and Summary

The Credal Ensemble and Credal Wrapper methods both successfully apply credal set theory to improve uncertainty quantification, but their approaches and ideal use-cases differ.

- **Credal Ensembles** involve training a specialized network architecture with a custom DRO-inspired loss function. This "natively credal" approach proves highly effective, as shown by its superior combined accuracy and OOD performance on the CIFAR-10 benchmark. It represents a powerful, integrated solution for building models with reliable uncertainty estimates from the ground up.
- **The Credal Wrapper** is a flexible, post-hoc method that constructs a credal set from the outputs of any standard ensemble. Its strength lies in its simplicity and adaptability. The empirical results demonstrate its remarkable effectiveness for OOD detection on the complex

CIFAR-100 dataset, where it outperformed all other methods, including the more complex Credal Ensemble.

In conclusion, both methods are valuable contributions to the field of uncertainty quantification. The empirical evidence suggests that the best choice may be task-dependent. For tasks similar to CIFAR-10, a Credal Ensemble offers state-of-the-art performance in a single package. For more complex tasks or when working with existing model ensembles, the Credal Wrapper provides a simple yet powerful tool to enhance OOD detection capabilities. Ultimately, both techniques underscore the utility of credal sets for creating more robust and reliable AI systems.

4 Contrastive Learning Learns Semantic Epistemic Uncertainty

4.1 Introduction

Deep neural networks have achieved remarkable success across various domains, yet their deployment in high-stakes environments like autonomous driving and medical diagnosis requires not only accurate predictions but also reliable estimates of their confidence. Uncertainty quantification (UQ) addresses this need by allowing models to express their own uncertainty. A principled approach, rooted in Bayesian modeling, decomposes the total predictive uncertainty into two fundamental types: **aleatoric uncertainty (AU)**, which captures inherent noise or ambiguity in the data itself, and **epistemic uncertainty (EU)**, which represents the model's ignorance due to limited training data (Kendall and Gal, 2017).

Ideally, these two sources of uncertainty should be disentangled, providing distinct and interpretable signals. For example, a model viewing a corrupted but otherwise familiar image should report high aleatoric uncertainty (due to data quality) but low epistemic uncertainty (as the content is familiar). Conversely, an image from a completely novel class should elicit high epistemic uncertainty. However, recent work has shown that in standard training regimes, these two uncertainties are often highly correlated, limiting their practical utility (Nayman et al., 2024).

We hypothesize that this entanglement stems from the fact that standard models conflate semantic content with nuisance variations (e.g., blur, noise, lighting). Epistemic uncertainty, which should ideally track a lack of knowledge about the core *content*, becomes inflated by superficial variations the model was not explicitly trained on.

In this paper, we propose to address this issue by learning a **semantic epistemic uncertainty**. Our core idea is that if a model's representations are invariant to nuisance variations while remaining sensitive to changes in semantic content, its epistemic uncertainty will naturally align with content-level shifts. To achieve this, we employ contrastive learning (Chen et al., 2020; He et al., 2020), which is renowned for learning such invariant representations. We treat clean data as an "anchor" and its corrupted versions as "positive" pairs, forcing the model to learn representations that focus on the shared, underlying content.

Our contributions are threefold:

1. We provide a theoretical framework that formally links contrastive learning to the disentanglement of content and nuisance information in a model's learned representations.
2. We prove that by minimizing nuisance information, our approach aligns the model's observation-level epistemic uncertainty with a more desirable content-level epistemic uncertainty.
3. We empirically validate our approach on CIFAR-10, showing that our contrastively trained model produces more meaningful uncertainty estimates. It correctly identifies corrupted in-distribution data as having high aleatoric but low epistemic uncertainty, and it significantly outperforms a standard baseline in detecting out-of-distribution data based on its superior epistemic signal.

4.2 Related Work

Our work is situated at the intersubsection of three key areas: uncertainty quantification, representation learning, and out-of-distribution detection.

Uncertainty Quantification. The decomposition of predictive uncertainty into aleatoric and epistemic components is a cornerstone of Bayesian deep learning (Depeweg et al., 2018). Practical methods for estimating these quantities often rely on approximations of the Bayesian posterior, such as Monte Carlo Dropout (Gal and Ghahramani, 2016) or Deep Ensembles (Lakshminarayanan et al., 2017), which we use in this work. While powerful, these methods do not inherently guarantee that the estimated EU will be semantically meaningful or disentangled from AU. Several works have noted the high correlation and proposed post-hoc recalibration methods, but few have addressed it at the training and representation level (Nayman et al., 2024).

Disentangled Representation Learning. The goal of disentanglement is to learn representations where distinct latent units correspond to distinct, interpretable factors of variation in the data (Bengio et al., 2013). Early work focused on generative models like VAEs, but later research revealed fundamental challenges in achieving unsupervised disentanglement without inductive biases (Locatello et al., 2019). Our work approaches disentanglement from a different perspective: instead of isolating all factors, we aim to specifically separate semantic content from all other nuisance variations for the purpose of robust UQ.

Contrastive Learning. Contrastive learning has emerged as a dominant paradigm in self-supervised learning (Oord et al., 2018). Methods like SimCLR (Chen et al., 2020) and MoCo (He et al., 2020) learn representations by maximizing agreement between differently augmented "views" of the same image. The inductive bias imposed by data augmentation forces the learned representations to be invariant to these transformations. Recent theoretical work has shown that this process provably isolates content from nuisance factors defined by the augmentations (von Kügelgen et al., 2021). We build directly on this principle, proposing to use data corruptions as a form of augmentation to explicitly learn representations invariant to them, thereby purifying the resulting epistemic uncertainty signal.

4.3 Preliminaries and Notations

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the training dataset, where $x_i \in \mathcal{X}$ represents an observed input (e.g., an image) and $y_i \in \mathcal{Y}$ is its corresponding label. We postulate a latent variable model where each input x is generated from underlying factors: a *content* variable $c \in \mathcal{C}$ representing the core semantic information relevant to the task, and a *nuisance* variable $n \in \mathcal{N}$ capturing variations irrelevant to the task (e.g., style, lighting, background).

Let θ denote the parameters of our predictive model, typically a neural network. The model aims to approximate the true data distribution and provide a predictive distribution $\hat{p}(y|x,\theta)$. When considering uncertainty, particularly in a Bayesian context or using ensembles, we are often interested in the posterior distribution of parameters $p(\theta|\mathcal{D})$. The overall predictive uncertainty for a new input x can be decomposed using the law of total variance for entropy (**?**):

$$\underbrace{\mathbb{H}\left[\mathbb{E}_{\theta\sim p(\theta|\mathcal{D})}[\hat{p}(y|x,\theta)]\right]}_{\text{Total Predictive Uncertainty}} = \underbrace{\mathbb{I}[y;\theta|x,\mathcal{D}]}_{\text{Epistemic Uncertainty (EU)}} + \underbrace{\mathbb{E}_{\theta\sim p(\theta|\mathcal{D})}\left[\mathbb{H}[\hat{p}(y|x,\theta)]\right]}_{\text{Aleatoric Uncertainty (AU)}}. \tag{13}$$

Here, $\mathbb{H}[\cdot]$ denotes the Shannon entropy, $\mathbb{I}[\cdot;\cdot|\cdot]$ denotes the conditional mutual information, and the expectation $\mathbb{E}_{\theta\sim p(\theta|\mathcal{D})}$ is taken over the parameter posterior distribution. Epistemic uncertainty (EU) captures the model's uncertainty due to limited training data, while aleatoric uncertainty (AU) captures inherent randomness or noise in the data generating process itself.

4.3.1 Factorised Data-Generating Process

We assume the following factorized structure for the joint distribution of labels, latent variables, and observations:

$$p(y, c, n, x) = p(y|c, n, x)p(x|c, n)p(c|n)p(n). \tag{14}$$

We make two simplifying assumptions common in disentanglement and robust representation learning literature (Bengio et al., 2013; Locatello et al., 2019):

1. **Latent Independence:** The content c and nuisance n variables are statistically independent, i.e., $p(c|n) = p(c)$.
2. **Deterministic Generation:** The observation x is generated by a deterministic function $g : \mathcal{C} \times \mathcal{N} \to \mathcal{X}$, such that $p(x|c, n) = \delta(x - g(c, n))$, where $\delta(\cdot)$ is the Dirac delta function.

Under these assumptions, and noting that x is fully determined by c and n, the dependence of y on x becomes redundant given c and n, i.e., $p(y|c, n, x) = p(y|c, n)$. The factorization simplifies to:

$$p(y, c, n, x) = p(y|c, n)\delta(x - g(c, n))p(c)p(n). \tag{15}$$

Often, it is further assumed that the label y depends only on the content c, i.e., $p(y|c, n) = p(y|c)$, although we do not strictly require this for the subsequent analysis unless specified.

4.3.2 Model Inference Process

We consider neural network models for classification, parameterized by θ. These models typically consist of an encoder $f_\phi : \mathcal{X} \to \mathcal{Z}$ with parameters ϕ, which maps the input x to a latent representation $z \in \mathcal{Z}$, followed by a classifier head $h_\psi : \mathcal{Z} \to \Delta^{|\mathcal{Y}|-1}$ with parameters ψ, which maps the representation z to a probability distribution over labels. Here, Δ^K denotes the K-dimensional probability simplex, and the full model parameters are $\theta = (\phi, \psi)$. The model's predictive distribution is thus given by:

$$\hat{p}(y|x, \theta) = h_\psi(z; \psi) \quad \text{where} \quad z = f_\phi(x; \phi). \tag{16}$$

Our primary interest lies in learning an encoder f_ϕ that extracts representations z which are informative about the underlying content c while being invariant to nuisance variations n. We hypothesize that such representations facilitate more meaningful uncertainty quantification. In this work, we leverage contrastive learning (Oord et al., 2018; Chen et al., 2020; He et al., 2020) to train the encoder f_ϕ and investigate its impact on epistemic uncertainty estimation, particularly for challenging tasks like Out-of-Distribution (OOD) detection.

4.4 Contrastive Training Learns Content-Aligned Epistemic Uncertainty

We now establish theoretically how contrastive learning, specifically through the optimization of the InfoNCE loss (Oord et al., 2018), encourages the learned representation $z = f_\phi(x)$ to capture content c while discarding nuisance n, and how this property leads to more robust epistemic uncertainty estimates.

4.4.1 Contrastive Learning Promotes Content Informativeness and Nuisance Invariance

Contrastive learning aims to learn representations by maximizing agreement between representations of different "views" (augmentations) of the same data point (positive pairs), while minimizing agreement with representations of different data points (negative pairs). We operate under the assumption that the positive pairs, denoted (x_1, x_2), are generated such that they share the same underlying content c but possess independent nuisance variables $n_1, n_2 \sim p(n)$. Formally, $x_1 = g(c, n_1)$ and $x_2 = g(c, n_2)$ for some $c \sim p(c)$ and $n_1, n_2 \overset{\text{i.i.d.}}{\sim} p(n)$. The InfoNCE objective, in its idealized population form, encourages the encoder f_ϕ to learn representations $z = f_\phi(x)$ such that the mutual information between representations of positive pairs is maximized.

This learning process implicitly optimizes the information content of the representation z with respect to the latent factors c and n. The following lemmas formalize this relationship, building upon insights from (Tsai et al., 2020; von Kügelgen et al., 2021).

Lemma 1 (Content Informativeness via Contrastive Learning) *Let $x = g(c, n)$ with $c \sim p(c)$ and $n \sim p(n)$ being independent. Let $z = f_\phi(x)$ be the representation learned by minimizing the InfoNCE loss using positive pairs (x_1, x_2) where $x_1 = g(c, n_1)$ and $x_2 = g(c, n_2)$ share the same content c and have independent nuisances n_1, n_2. Optimizing the InfoNCE objective generally encourages an increase in the mutual information between the representation and the content variable:*

$$\mathbb{I}(c; z) \uparrow .$$

Lemma 2 (Nuisance Invariance via Contrastive Learning) *Under the same assumptions as Lemma 1, optimizing the InfoNCE objective encourages a reduction in the mutual information between the nuisance variable and the representation, conditioned on the content:*

$$\mathbb{I}(n; z \mid c) \downarrow .$$

Intuition: Lemma 1 arises because the shared information between positive pairs (x_1, x_2) is primarily the content c. By pulling their representations (z_1, z_2) together, the InfoNCE loss implicitly maximizes the information z retains about c. Lemma 2 follows because, for a fixed content c, the representations $z_1 = f_\phi(g(c, n_1))$ and $z_2 = f_\phi(g(c, n_2))$ are encouraged to be similar despite variations in n_1 and n_2. This forces the representation z to become invariant to nuisance variations n, conditional on knowing the content c. Rigorous proofs adapting arguments from prior work are provided in Appendix A.

4.4.2 Contrastive Representations Enable Content-Aligned Epistemic Uncertainty

A key desideratum for reliable uncertainty quantification, especially in safety-critical applications or OOD detection, is that epistemic uncertainty (EU) reflects genuine model ignorance about the task-relevant content c, rather than being inflated by superficial nuisance variations n. We formalize this by defining two related quantities:

Definition 1 (Observation-level and Content-level Epistemic Uncertainty) *Given an input $x = g(c, n)$, the training data $\mathcal{D}$, and the parameter posterior $p(\theta|\mathcal{D})$:*

- *The standard **observation-level epistemic uncertainty** is:*
$$EU_x := \mathbb{I}(y; \theta \mid x, \mathcal{D}) = \mathbb{I}(y; \theta \mid c, n, \mathcal{D}).$$
- *The desired **content-level epistemic uncertainty** is:*
$$EU_c := \mathbb{I}(y; \theta \mid c, \mathcal{D}).$$

Ideally, we want $EU_x \approx EU_c$, meaning the uncertainty estimate for an observation x is determined by its content c, irrespective of the nuisance n. We now demonstrate that representations learned via contrastive methods promote this alignment.

Proposition 1 (Epistemic Uncertainty Alignment via Contrastive Representations) *Let $EU_x = \mathbb{I}(y; \theta \mid c, n, \mathcal{D})$ and $EU_c = \mathbb{I}(y; \theta \mid c, \mathcal{D})$ be the epistemic uncertainties defined above. Assuming the learned representation $z = f_\phi(x)$ captures the relevant information pathways, the absolute difference between these two uncertainty measures can be bounded in terms of the nuisance information retained by the representation:*
$$|EU_x - EU_c| \leq \mathbb{I}(n; \theta \mid c, \mathcal{D}) \approx \mathbb{I}(n; z \mid c). \tag{17}$$
The approximation $\mathbb{I}(n; \theta \mid c, \mathcal{D}) \approx \mathbb{I}(n; z \mid c)$ holds under the assumption that the variation in θ (especially the encoder parameters ϕ) relevant to predicting y given n (conditional on c) is primarily channeled through the representation z.

Proof Sketch: The relationship $|EU_x - EU_c| \leq \mathbb{I}(n; \theta \mid c, \mathcal{D})$ follows from properties of conditional mutual information (see Appendix A for details). The crucial step is linking $\mathbb{I}(n; \theta \mid c, \mathcal{D})$ to $\mathbb{I}(n; z \mid c)$. Since the model's prediction depends on n primarily through $z = f_\phi(x)$, the information that the parameters θ (specifically ϕ) hold about n (given c) is closely related to the information z holds about n (given c).

Implication. Proposition 1 demonstrates that the discrepancy between the standard epistemic uncertainty EU_x and the more desirable content-grounded uncertainty EU_c is bounded by the amount of nuisance information encoded in the representation z, conditioned on the content c. According to Lemma 2, optimizing the InfoNCE loss actively minimizes this conditional mutual information $\mathbb{I}(n; z \mid c)$. Therefore, contrastive pre-training naturally encourages the learned epistemic uncertainty (computed using the learned representation z) to align better with the content-level uncertainty EU_c. This provides a theoretical grounding for the empirical observation that contrastively trained models often yield more robust uncertainty estimates, particularly for distinguishing in-distribution nuisance variations from genuine out-of-distribution samples based on content mismatch.

4.5 Experiments

We conduct a comprehensive set of experiments to empirically validate our hypothesis that contrastive learning can produce a more disentangled, semantic epistemic uncertainty. We compare our proposed method against a standard supervised baseline across three benchmark datasets: MNIST, CIFAR-10, and CIFAR-100.

4.5.1 Experimental Setup

Datasets and Data Splits. For each benchmark, we define four categories of test data to evaluate model performance and uncertainty under different distribution shifts:

- **Clean In-Distribution (ID)**: The standard, unmodified test set of the dataset (e.g., CIFAR-10 test set).
- **Nuisance Shift (Seen)**: ID images corrupted by a set of nuisance transformations (e.g., blur, noise from CIFAR-10-C) with a fixed severity. These specific corruption types are used during the training phase.
- **Nuisance Shift (Unseen)**: ID images corrupted by a *different*, held-out set of nuisance transformations at a higher severity, representing a novel distributional shift in the nuisance variable.
- **Semantic Shift (OOD)**: A true out-of-distribution dataset with different semantic content. We use FashionMNIST, KMNIST, and EMNIST for MNIST; SVHN for CIFAR-10; and SVHN for CIFAR-100.

Models and Training. Our experiments are based on a **ResNet-18** architecture. For each experiment, we train a deep ensemble of 5 models to obtain robust uncertainty estimates. We compare two training paradigms:

- **Standard (Baseline)**: A ResNet-18 model trained with a standard cross-entropy loss on a mixture of Clean ID and Nuisance Shift (Seen) data.
- **Contrastive (Ours)**: We first train the ResNet-18 encoder using a contrastive loss, where clean images serve as anchors and their Nuisance Shift (Seen) counterparts act as positive samples. Subsequently, the encoder is frozen, and a linear classifier is trained on top using the labeled Clean ID data.

Evaluation. We evaluate models on: (1) Test accuracy; (2) Mean uncertainty scores (Aleatoric, Epistemic, and Total); and (3) OOD detection performance, measured by AUROC and AUPR.

4.5.2 Uncertainty Under Nuisance and Semantic Shifts

We first analyze how the uncertainty estimates from each model behave across the different data splits. Table 7 summarizes the accuracy and mean uncertainty scores.

Across all three datasets, the **aleatoric uncertainty (AU)** behaves as expected for both models, increasing monotonically as the data quality degrades from Clean ID to Nuisance Shift (Seen) and Nuisance Shift (Unseen). This correctly reflects the increasing noise and inherent difficulty in the input data.

The primary distinction emerges in the **epistemic uncertainty (EU)**. For the **Standard** baseline, the EU consistently rises when moving from Clean ID to Nuisance Shift (Seen) data (e.g., from 0.063 to 0.076 on CIFAR-10; 0.300 to 0.380 on CIFAR-100). This indicates the model perceives familiar corruptions as a source of model uncertainty, confounding nuisance with novelty.

In stark contrast, our **Contrastive** model demonstrates remarkable robustness. On CIFAR-100, its EU remains almost constant between Clean ID (0.170) and Nuisance Shift (Seen) (0.186). On CIFAR-10, the EU (0.073 vs 0.086) is also much more stable than the baseline. This provides strong evidence that our training method successfully learns to be invariant to seen nuisances, attributing them correctly to data noise (high AU) rather than model ignorance (low EU). Crucially, for novel shifts—both Nuisance (Unseen) and Semantic (OOD)—the EU of our model increases sharply and significantly, demonstrating its sensitivity to genuine distribution shifts.

4.5.3 Out-of-Distribution Detection

Distinguishing Distributional Shifts from Clean Data

A key test for a well-calibrated uncertainty model is its ability to distinguish shifted data from clean ID data. Table 8 shows this comparison.

Table 7: Accuracy and Mean Uncertainty Scores across all datasets. Our contrastive method maintains low epistemic uncertainty (EU) on familiar nuisance shifts while correctly elevating it for unseen and semantic shifts, unlike the standard baseline which shows increased EU even for familiar corruptions.

Dataset	Model	Clean ID	Nuisance Shift (Seen)	Nuisance Shift (Unseen)	Semantic Shift (OOD)
			Test Accuracy		
CIFAR-10	Standard	0.878	0.822	0.698	—
	Contrastive (Ours)	0.886	0.864	0.635	—
MNIST	Standard	0.997	0.996	0.572	—
	Contrastive (Ours)	0.996	0.994	0.388	—
CIFAR-100	Standard	0.787	0.683	0.395	—
	Contrastive (Ours)	0.713	0.687	0.397	—
			Mean Aleatoric Uncertainty (AU)		
CIFAR-10	Standard	0.313	0.509	0.869	1.317
	Contrastive (Ours)	0.294	—	0.496	0.929
MNIST	Standard	0.014	0.038	0.420	0.529*
	Contrastive (Ours)	0.029	0.145	0.194	0.487*
CIFAR-100	Standard	0.615	0.991	1.607	2.296
	Contrastive (Ours)	1.994	1.985	2.223	2.490
			Mean Epistemic Uncertainty (EU)		
CIFAR-10	Standard	0.063	0.076	0.115	0.112
	Contrastive (Ours)	**0.073**	**0.086**	**0.164**	**0.266**
MNIST	Standard	0.005	0.006	0.133	0.247*
	Contrastive (Ours)	**0.008**	**0.041**	**0.152**	**0.296***
CIFAR-100	Standard	0.300	0.380	0.540	0.547
	Contrastive (Ours)	**0.170**	**0.186**	**0.346**	**0.459**

*For MNIST, OOD uncertainty is averaged across FashionMNIST, KMNIST, and EMNIST.

The results reveal two critical patterns. First, when detecting **Nuisance Shift (Seen)** data, our contrastive model consistently yields a lower AUROC score than the baseline (e.g., 0.539 vs 0.579 EU-AUROC on CIFAR-10). This is a desirable property, confirming that our model correctly perceives these familiar nuisance variations as being "in-distribution" from a semantic standpoint and does not flag them as novel.

Second, for detecting true **Semantic Shift (OOD)**, our model is dramatically superior. On CIFAR-100, the EU-based AUROC for our model is **0.959**, a massive improvement over the baseline's 0.757. Similarly, on CIFAR-10, we see a jump from 0.743 to **0.876**. This demonstrates that the epistemic uncertainty learned via our contrastive method is a far more reliable and pure signal for detecting genuine, content-based novelty. While the baseline's Total or Aleatoric uncertainty sometimes achieves high AUROC, it does so by confounding data noise with semantic shifts, a problem our method mitigates.

Table 8: OOD Detection vs. Clean ID Data (AUROC). Our method is intentionally less sensitive to seen nuisance shifts but vastly superior at detecting true semantic shifts using epistemic uncertainty (EU), highlighted in bold.

Dataset	Uncertainty	vs. Nuisance (Seen)(↓)		vs. Nuisance (Unseen)(↓)		vs. Semantic (OOD)(↑)	
		Standard	Contrastive	Standard	Contrastive	Standard	Contrastive
CIFAR-10	Total	0.618	**0.536**	0.775	**0.672**	**0.925**	0.890
	Aleatoric	0.623	**0.535**	0.782	**0.659**	**0.936**	0.883
	Epistemic	0.579	**0.539**	0.697	**0.691**	0.743	**0.876**
MNIST	Total	**0.712**	0.788	0.922	**0.883**	**0.973**	0.954
	Aleatoric	**0.713**	0.788	0.923	**0.876**	**0.972**	0.950
	Epistemic	**0.698**	0.790	0.917	**0.899**	0.974	**0.978**
CIFAR-100	Total	0.628	**0.502**	0.807	**0.607**	**0.942**	0.704
	Aleatoric	0.634	**0.497**	0.810	**0.564**	**0.950**	0.637
	Epistemic	0.587	**0.542**	**0.726**	0.797	0.757	**0.959**

Distinguishing Between Degraded and OOD Data

Finally, we test the model's ability to perform more nuanced distinctions, as shown in Table 9. A key challenge for robust models is to separate true semantic OOD samples from merely corrupted in-distribution ones.

The results are striking. When tasked with distinguishing **Nuisance (Seen) vs. Semantic OOD**, our model's epistemic uncertainty provides a powerful separative signal. On CIFAR-100, it achieves an AUROC of **0.941**, whereas the baseline model is close to random chance at 0.675. This is a critical result: the baseline model, when presented with a familiar corrupted image, is almost as uncertain as when shown a true OOD image. Our model, however, has learned to be confident about the content of the familiar corruption (low EU) while being appropriately uncertain about the true OOD image (high EU), enabling clear separation.

Similarly, when distinguishing **Nuisance (Unseen) vs. Semantic OOD**, our model consistently maintains a stronger signal (e.g., EU-AUROC of 0.704 vs 0.482 on CIFAR-100), demonstrating its superior ability to disentangle content novelty from nuisance variations even in challenging scenarios.

Table 9: Detection Performance Between Degraded and OOD Data (AUROC). Our model's epistemic uncertainty provides a significantly stronger signal for distinguishing true semantic OOD from both seen and unseen nuisance shifts.

Dataset	**Uncertainty**	**Nuisance (Seen) vs. (Unseen)(↓)**		**Nuisance (Seen) vs. Semantic OOD(↑)**		**Nuisance (Unseen) vs. Semantic OOD(↑)**	
		Standard	Contrastive	Standard	Contrastive	Standard	Contrastive
CIFAR-10	Total	0.677	**0.639**	0.859	**0.871**	0.701	**0.774**
	Aleatoric	0.678	**0.627**	**0.868**	0.864	0.720	**0.784**
	Epistemic	0.683	**0.656**	0.683	**0.850**	0.523	**0.703**
MNIST*	Total	0.851	**0.634**	0.701	**0.766**	0.649	**0.799**
	Aleatoric	0.848	**0.604**	0.689	**0.788**	0.590	**0.817**
	Epistemic	0.855	**0.681**	0.707	**0.714**	0.752	**0.757**
CIFAR-100	Total	0.700	**0.606**	**0.862**	0.703	**0.689**	0.581
	Aleatoric	0.695	**0.568**	**0.870**	0.631	**0.715**	0.560
	Epistemic	**0.656**	0.765	0.675	**0.941**	0.482	**0.704**

*MNIST OOD columns use specific datasets: vs. KMNIST, *vs. KMNIST, Nuisance(Seen) vs FashionMNIST, Nuisance(Unseen) vs EMNIST.

4.6 Conclusion and Future Work

In this paper, we addressed the prevalent issue of entanglement between aleatoric and epistemic uncertainty. We argued that for uncertainty to be interpretable and reliable, a model's epistemic uncertainty (EU) should be grounded in semantic content, reflecting true model ignorance rather than superficial nuisance variations in the data. To this end, we proposed a training framework that uses contrastive learning to build representations that are invariant to known data corruptions. Our theoretical analysis provides a formal justification, showing that this approach minimizes the influence of nuisance variables on the final epistemic uncertainty estimate.

Our comprehensive experiments across MNIST, CIFAR-10, and CIFAR-100 provide strong empirical validation of our claims. The results consistently demonstrate that our contrastively trained model learns a more disentangled and meaningful form of uncertainty. As per our core expectation, the model correctly identifies low-quality in-distribution data (Nuisance Shifts) as being semantically familiar, evidenced by two key findings: (1) its epistemic uncertainty remains low and stable, comparable to that on clean data, and (2) its ability to distinguish these nuisance-shifted samples from clean data is near random chance, the desired outcome for nuisance invariance. Conversely, the model's EU proves to be a powerful and pure signal for detecting true, content-based novelty. This is most evident in the semantic OOD detection tasks, where our method dramatically outperforms the standard baseline (e.g., achieving an EU-based AUROC of 0.959 vs. 0.757 on CIFAR-100). Furthermore, our model excels at the critical task of separating true OOD samples from merely corrupted ones—a scenario where the baseline model fails.

A limitation of our current approach is that the contrastive pre-training of the encoder is entirely self-supervised and does not leverage class labels. This may potentially limit the discriminative power of the learned features for the downstream classification task, which could contribute to the slightly lower clean accuracy observed on CIFAR-100. This opens up a clear avenue for future work: exploring supervised or semi-supervised contrastive learning techniques. By integrating label information into the contrastive objective, it may be possible to enhance classification performance while retaining the robust, semantic uncertainty properties demonstrated in this work. Ultimately, this research paves the way for building more reliable models where the reported uncertainties are more closely and interpretably aligned with their intended real-world meaning.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghasemian, M., Li, J., Zhang, Z., Khosravi, A., Acharya, U. R., et al. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Begoli, E., Bhattacharya, T., and Somen, D. (2021). A survey of uncertainty quantification in machine learning: From models to applications. *arXiv preprint arXiv:2106.13813*.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Chen, Z. and Zhao, B. (2023). Spuq: A simple and effective method for quantifying uncertainty of pre-trained language models. *arXiv preprint arXiv:2311.00288*.

Cohen, J., Rosenfeld, E., and Kolter, Z. (2019). Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR.

Daxberger, E., Kristiadi, A., Immer, A., Hron, R., Eschenhagen, F., and Hennig, P. (2021). Laplace redux-effortless bayesian deep learning. *Advances in Neural Information Processing Systems*, 34:20089–20102.

Depeweg, S., Hernandez-Lobato, J.-M., Doshi-Velez, F., and Udluft, S. (2018). Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International conference on machine learning*, pages 1184–1193. PMLR.

Der Kiureghian, A. and Ditlevsen, O. (2009). Aleatory and epistemic uncertainty in machine learning: An introduction to concepts and methods. In *Computational Methods in Engineering and Health Sciences*, pages 245–273. Springer.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

He, Z., Meng, J., Zhang, W., and Ren, P. (2023). A survey on uncertainty quantification in large language models. *arXiv preprint arXiv:2310.03154*.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. (2011). Bayesian active learning for classification and preference learning.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584.

Kristiadi, A., Hein, M., and Hennig, P. (2020). Being bayesian, even just a little bit, helps. In *International Conference on Machine Learning*, pages 5436–5446. PMLR.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR.

MacKay, D. J. (1992). A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472.

Martens, J. and Grosse, R. (2015). Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR.

Meiseles, E. and Rokach, L. (2023). On the pitfalls of deep evidential learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9936–9944.

Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B., and Snoek, J. (2021). Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning. In *Third Workshop on Uncertainty & Robustness in Deep Learning*.

Nayman, N., Noy, A., Halperin, T., Avidan, S., and Gal, Y. (2024). The unreasonable effectiveness of post-hoc uncertainty quantification. *arXiv preprint arXiv:2402.19460*.

Neal, R. M. (2012). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Snoek, J., Lakshminarayanan, B., and D'Amour, A. (2019). Evaluating scalable bayesian deep learning methods for robust computer vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0.

Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. (2019). On variational bounds of mutual information. In *International conference on machine learning*, pages 5171–5180. PMLR.

Ritter, H., Botev, A., and Barber, D. (2018). A scalable laplace approximation for neural networks. In *International Conference on Learning Representations*.

Sensoy, M., Kaplan, L., and Kandemir, M. (2018). Evidential deep learning to quantify classification uncertainty. In *Advances in neural information processing systems*, pages 3179–3189.

Shanmugam, D., M L, G., et al. (2021). Test-time augmentation for deep learning: A survey. *arXiv preprint arXiv:2107.13279*.

Tsai, Y.-H. H., Wu, Y., Salakhutdinov, R., and Morency, L.-P. (2020). Self-supervised learning from a multi-view perspective. *arXiv preprint arXiv:2006.05576*.

Van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. (2020). Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR.

von Kügelgen, J., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., and Locatello, F. (2021). Self-supervised learning with data augmentations provably isolates content from style. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16451–16467. Curran Associates, Inc.

Wang, H. and Ji, Q. (2024). Epistemic uncertainty quantification for pre-trained neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23485–23494.

Wang, K., Cuzzolin, F., Manchingal, S. K., Shariatmadar, K., Moens, D., and Hallez, H. (2024a). Credal deep ensembles for uncertainty quantification. In *Thirty-eighth Conference on Neural Information Processing Systems*.

Wang, K., Cuzzolin, F., Shariatmadar, K., Moens, D., and Hallez, H. (2024b). Credal wrapper of model averaging for uncertainty estimation in classification. *arXiv preprint arXiv:2405.15047*.

Wang, T. and Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR.

Yoo, S.-H., Jang, H.-O., and Chun, H. (2022). Calibrating bnns with a beta-tempering-based loss function for a more reliable ood detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4552–4561.

Zhang, Z., Shi, J.-C., Wang, Y., Li, H., Yuan, Y., Shi, W., and Zhang, Y. (2023). Input clarification ensembling for large language models. *arXiv preprint arXiv:2311.08718*.

A Proofs of Theoretical Results

A.1 Proof of Lemma 1: InfoNCE Increases $\mathbb{I}(c; z)$

We begin by analyzing the effect of minimizing the InfoNCE loss on mutual information between the content variable c and the learned representation z. Let $x_1 = g(c, n_1)$ and $x_2 = g(c, n_2)$ be two samples sharing the same content c, but with independent nuisance factors n_1, n_2. Let $z_1 = f_\phi(x_1), z_2 = f_\phi(x_2)$.

Following Oord et al. (2018) and Poole et al. (2019), the InfoNCE loss with N negative samples provides a variational lower bound on the mutual information between z_1 and z_2:

$$\mathbb{I}(z_1; z_2) \geq \log N - L_{\text{InfoNCE}}. \tag{18}$$

Now, suppose x_1 is a "super clean" view such that $p(z_1|x_1) = p(z_1|c)$. Since x_2 shares the same content c, we have:

$$\mathbb{I}(z_1; z_2) \leq \mathbb{I}(c; z_2), \tag{19}$$

by the data processing inequality.

Therefore, minimizing L_{InfoNCE} increases a lower bound of $\mathbb{I}(c; z)$, implying that z becomes increasingly informative about the content c.

A.2 Proof of Lemma 2

As shown in Wang and Isola (2020), the InfoNCE loss can be approximated as:

$$L_{\text{InfoNCE}} \approx \frac{1}{2\tau} \mathbb{E}\left[\|z_1 - z_2\|^2\right] + \log Z.$$

As the encoder is trained to minimize this loss, $\mathbb{E}[\|z_1 - z_2\|^2] \to 0$ for (z_1, z_2) generated from (x_1, x_2) sharing the same c.

This implies that the conditional distribution $p(z|c, n)$ becomes indistinguishable from $p(z|c)$. By Pinsker's and Jensen's inequalities:

$$D_{\text{KL}}(p(z|c, n) \| p(z|c)) \to 0.$$

Therefore:

$$\mathbb{I}(n; z \mid c) = \mathbb{E}_{c,n}[D_{\text{KL}}(p(z|c, n) \| p(z|c))] \to 0.$$

A.3 Proof of Proposition 1

We start from the chain rule of mutual information:

$$\mathbb{I}(y;\theta \mid c,n,D) = \mathbb{I}(y;\theta \mid c,D) - \mathbb{I}(y;n \mid c,D) + \mathbb{I}(y;n \mid c,\theta,D).$$

This gives:

$$|EU_x - EU_c| = |\mathbb{I}(y;n \mid c,\theta,D) - \mathbb{I}(y;n \mid c,D)| \leq \mathbb{I}(y;n \mid c,\theta,D).$$

Assuming that y depends on n only through its effect on the latent representation $z = f_\phi(x)$, we obtain:

$$\mathbb{I}(y;n \mid c,\theta,D) \leq \mathbb{I}(z;n \mid c,\theta,D) \leq \mathbb{I}(n;z \mid c),$$

by the data processing inequality. Therefore:

$$|EU_x - EU_c| \leq \mathbb{I}(n;z \mid c).$$

By Lemma 2, this upper bound is minimized during contrastive learning.