My Thesis Title

Liam Heppner

February 26, 2026

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Department of Mathematics,
Informatics and Statistics
Institute of Informatics

Artificial Intelligence and
Machine Learning

Bachelor's Thesis

My Thesis Title

Liam Heppner

Reviewer Eyke Hüllermeier
Institute of Informatics
LMU Munich

Supervisors Eyke Hüllermeier and Paul Hofman

February 26, 2026

Liam Heppner
My Thesis Title
Bachelor's Thesis, February 26, 2026
Reviewers: Eyke Hüllermeier
Supervisors: Eyke Hüllermeier and Paul Hofman

LMU Munich
Department of Mathematics, Informatics and Statistics
Institute of Informatics
Artificial Intelligence and Machine Learning (AIML)
Akademiestraße 7
80799 Munich

Abstract

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Abstract (different language)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

1 Introduction

2 Background

This chapter will discuss the key concepts essential for understanding this thesis. ...

2.1 Large Language Models and Generation

Large Language Models (LLMs) represent a class of deep learning models designed to process, understand, and generate human language. At their core, modern LLMs are built upon the Transformer architecture, utilizing mechanisms such as self-attention to model complex dependencies between words across long sequences [Vas+17; Nav+24]. While they demonstrate capabilities in reasoning [Wei+23] and coding [Tia+23], their fundamental operation remains probabilistic. They predict the likelihood of the next token in a sequence based on the preceding context [Vas+17; BCB16].

2.1.1 Transformers and Next-token prediction

The primary objective of an autoregressive LLM is to model the probability distribution of a target sequence of tokens $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ given an input context or prompt x. The model decomposes the joint probability of the sequence into a product of conditional probabilities using the chain rule of probability [BCB16]:

$$P(\mathbf{y}|x) = \prod_{t=1}^{T} P(y_t \mid \mathbf{y}_{<t}, x) \tag{2.1}$$

Where y_t is the token at step t and $\mathbf{y}_{<t}$ represents all preceeding tokens $y_1, \ldots, y_{t-1}$ in the generated sequence. To compute these probabilities, the Transformer processes the input x and the generated history $\mathbf{y}_{<t}$ to produce a vector of non-normalized scores, known as logits, for every token in the vocabulary $\mathcal{V}$. Let z_i be the logit corresponding to the i-th token $v_i \in \mathcal{V}$. The model converts these logits into a valid probability distribution using the Softmax function [Bri90; Vas+17]:

$$P(y_t = v_i \mid \mathbf{y}_{<t}, x) = \frac{\exp(z_i)}{\sum_{j=1}^{|\mathcal{V}|} \exp(z_j)} \qquad \forall i \in 1, \ldots, |\mathcal{V}| \tag{2.2}$$

This process is repeated iteratively, where the token selected at step t is appended to the input for step $t + 1$ [Vas+17].

2.1.2 Decoding Strategies and the Role of Temperature

Once the probability distribution over the Vocabulary is computed for the next token, a **decoding strategy** determines what token will be selected. Common maximization-based strategies include **Greedy Decoding** or **Beam Search**. **Greedy Decoding** is the simplest deterministic method. It selects the token with the highest probability mass at the current time step t [Ipp+19]:

$$y_t = \arg\max_{y \in \mathcal{V}} P(y \mid \mathbf{y}_{<t}, x) \tag{2.3}$$

While computationally efficient, this strategy often yields sub-optimal results. By focusing solely on local probability maximization, the strategy risks settling into local optima, effectively missing high-probability sequences hidden behind lower-probability initial tokens. Furthermore, greedy decoding is known to produce repetitive and generic text, as it fails to account for the diversity and global coherence of the sequence [Shi+24].

Beam Search extends Greedy Decoding by maintaining multiple hypotheses at each time step, rather than a single sequence. It introduces a hyperparameter k, known as the **beam width**, which determines the number of candidates to keep. At each step t, the algorithm expands all k candidates from the previous step by appending every possible next token from the vocabulary $\mathcal{V}$ [Shi+24].

Formally, let $\mathcal{B}_{t-1} = \{\mathbf{y}_{<t}^{(1)}, \ldots, \mathbf{y}_{<t}^{(k)}\}$ be the set of k hypothesis sequences generated up to step $t - 1$. The algorithm proceeds in three steps [Shi+24]:

1. **Expansion**: Each hypothesis in $\mathcal{B}_{t-1}$ is extended with every possible token v from the vocabulary $\mathcal{V}$, creating a candidate set $\mathcal{C}_t$.

2. **Scoring**: Each candidate sequence is assigned a score based on its cumulative log-probability. The score for a sequence $\mathbf{y}_{1:t}$ (representing $y_1, \ldots, y_t$) is defined recursively:

$$\text{Score}(\mathbf{y}_{1:t}) = \text{Score}(\mathbf{y}_{1:t-1}) + \log P(y_t \mid \mathbf{y}_{<t}, x) \tag{2.4}$$

3. **Selection (Pruning)**: The algorithm selects the k candidates with the highest scores to form the new hypothesis set $\mathcal{B}_t$:

$$\mathcal{B}_t = \underset{\mathbf{y}_{1:t} \in \mathcal{C}_t}{\text{arg-top-}k} \left(\text{Score}(\mathbf{y}_{1:t})\right) \tag{2.5}$$

This approach allows the model to recover from locally sub-optimal decisions if they lead to a highly probable global sequence. However, Beam Search is computationally more expensive than greedy methods and, as noted by Holtzman et al. [Hol+20], can still suffer from repetitive loops or generic responses in open-ended generation tasks.

Sampling represents a fundamental shift from maximization-based decoding strategies towards stochastic generation. Instead of strictly selecting the token with the highest probability, the model randomly draws the next token y_t from the conditional probability distribution $P(y_t \mid \mathbf{y}_{<t}, x)$ [Shi+24].

Formally, at each time step t, the next token is sampled according to:

$$y_t \sim P(y \mid \mathbf{y}_{<t}, x) \tag{2.6}$$

This stochasticity allows the model to generate more diverse and creative outputs for the same input. However, pure random sampling from the full distribution can occasionally lead to incoherence. If the "tail" of the distribution, that contains very low-probability tokens, is sampled, it may disrupt the semantic flow or factual accuracy of the sequence. To mitigate this risk while preserving diversity, two truncation techniques are commonly employed:

1. **Top-k Sampling** [FLD18]: The model samples from the top k most probable tokens. This effectively truncates the unreliable tail of the distribution.

2. **Nucleus (Top-p) Sampling** [Hol+20]: The model samples from the smallest set of tokens whose cumulative probability exceeds a threshold p (e.g., $p = 0.9$). Unlike Top-k, which uses a fixed number of tokens, Top-p adapts dynamically. The set of candidates grows when the model is uncertain (flat distribution) and shrinks when the model is confident (peaked distribution).

While these truncation methods improve coherence, they operate by removing options. A more continuous method for controlling the shape of the probability distribution, and the core mechanism for this thesis, is **Temperature Scaling**.

Temperature Scaling is a technique used to calibrate the confidence of the model's predictions. Unlike Top-k or Nucleus Sampling, which truncate the vocabulary, temperature scaling modifies the probability distribution itself by rescaling the logits before the Softmax function is applied [Shi+24].

Let z_i be the logit for token v_i. We introduce a hyperparameter $\tau > 0$, called **temperature**. The probability of selecting token v_i is then given by:

$$P(y_t = v_i \mid \mathbf{y}_{<t}, x; \tau) = \frac{\exp(z_i/\tau)}{\sum_{j \in \mathcal{V}} \exp(z_j/\tau)} \tag{2.7}$$

The value of τ controls the entropy of the ouptut distribution. When choosing a **low temperature**, $\tau < 1.0$, the distribution becomes sharper, i.e., more peaked. The difference between large and small logits is exaggerated, causing the model to become more confident and deterministic. In the limit, this approximates greedy decoding. Using a **high temperature**, $\tau > 1.0$, flattens the distribution towards a uniform distribution. This increases the likelihood of selecting lower-probability tokens, resulting in more diverse but also potentially more incoherent or hallucinatory outputs. For the temperature $\tau = 1.0$, the distribution remains unchanged, reflecting the model's original logits [Shi+24].

In standard generation tasks, τ is typically a fixed hyperparameter tuned to balance diversity and equality. However, in this thesis, we interpret temperature variation as a mechanism to explore the model's epistemic uncertainty.

2.2 Uncertainty in Machine Learning

As LLMs are increasingly deployed in real-world and high-stakes applications, for example in healthcare [LIH24], their reliability becomes a critical concern. A fundamental challenge with deep neural networks, including LLMs, is that they are prone to overconfidence. They frequently assign high probabilities to incorrect predictions [Guo+17]. Consequently, evaluating a model's performance requires not only measuring its accuracy but also quantifying its **uncertainty**.

2.2.1 Aleatoric and Epistemic Uncertainty

In the field of machine learning, uncertainty can be categorized into two distinct sources: aleatoric and epistemic [KG17; HW21].

Aleatoric Uncertainty arises from the inherent complexity, noise, or randomness in the underlying data generation process. It is irreducable, meaning providing the model with more training data will not eliminate this uncertainty [KG17; HW21]. In the context of Natural Language Processing (NLP), aleatoric uncertainty is highly prevalant due to the inherent flexibility and ambiguity of human language. For a given prompt, there are often multiple semantically equivalent and perfectly valid responses. For example, the answers "It is raining heavily" and "It is pouring outside" describe the same state, distributing the model's probability mass across different valid token sequences.

Epistemic uncertainty, on the other hand, stems from a lack of knowledge or information about the best model. This uncertainty occurs when a model encounters out-of-distribution (OOD) inputs or queries about facts absent from its training data. Unlike aleatoric uncertainty, epistemic uncertainty is reducable. Theoretically it

can be minimized by providing the model with more comprehensive training data [KG17; HW21].

2.2.2 Standard Metrics: Shannon Entropy and Likelihood

To quantify the theoretical concepts of aleatoric and epistemic uncertainty, we rely on established mathematical metrics. In the context of probabilistic sequence modeling, two fundamental measures for evaluating a model's predictive confidence are **Likelihood** and **Shannon Entropy**.

Likelihood and Negative Log-Likelihood (NLL)
The likelihood represents the probability assigned by the model to a specific, observed sequence of tokens. For a given input context x and a target sequence $\mathbf{y}$, the sequence likelihood is the product of the individual token probabilities as defined in 2.1. Because multiplying many small probabilities leads to numerical underflow, it is standard practice to compute the **Negative Log-Likelihood (NLL)** [GBC16].

Taking the negative logarithm transforms the product of the probabilities into a sum. Furthermore, to fairly compare sequences of different lengths, the NLL can be normalized by the total number of tokens T, yielding the average NLL per token [MC18]:

$$\text{NLL}(\mathbf{y} \mid x) = -\sum_{t=1}^{T} \log P(y_t \mid \mathbf{y}_{<t}, x) \tag{2.8}$$

$$\text{average NLL}(\mathbf{y} \mid x) = -\frac{1}{T}\sum_{t=1}^{T} \log P(y_t \mid \mathbf{y}_{<t}, x) \tag{2.9}$$

A lower NLL indicates that the model asssigns a higher probability to the target sequence, implying higher confidence.

Shannon Entropy
While seqeuence likelihood evaluates the probability the model assigns to a *specific* target sequence, **Shannon Entropy** quantifies the fundamental uncertainty inherent in a probability distribution. In the context of autoregressive models, calculating the entropy of the predictive distribution over the entire vocabulary at a given time step t measures the expected "surprise" or uncertainty in predicting the next token [Sha48].

For a discrete random variable Y_t representing the next token, with a probability distribution $P(Y_t \mid \mathbf{y}_{<t}, x)$ over the vocabulary $\mathcal{V}$, the Shannon Entropy H is defined as:

$$H(Y_t) = -\sum_{v \in \mathcal{V}} P(y_t = v \mid \mathbf{y}_{<t}, x) \log P(y_t = v \mid \mathbf{y}_{<t}, x) \tag{2.10}$$

Entropy provides a scalar summary of the distribution's shape.

- **Low Entropy** ($H \to 0$): The distribution is highly concentrated (peaked) on one or a few tokens, indicating that the model is highly certain about its next prediction.

- **High Entropy** ($H \to \log |\mathcal{V}|$): The distribution is flat and uniform across the vocabulary, indicating maximum uncertainty.

Limitations of Standard Metrics
Both NLL and Shannon Entropy are highly effective metrics, provided that the underlying probabiliy distribtuion P is perfectly calibrated and accurately reflects the true state of the world. However, these metrics inherently assume that the model's output distribution is a single, precise, and entirely correct representation of reality [Guo+17; GR07].

As demonstrated by Guo et al. [Guo+17], this assumption of a "precise probability" often breaks down in modern deep leraning models, including LLMs, which frequently suffer from severe overconfidence and calibration. Furthermore, standard precise probabilities mathematically struggle to distinguish between inherent data noise (aleatoric) and a fundamental lack of knowledge (epistemic) [HW21]. This failure to capture true epistemic misconfidence necessitates ore robust mathematical frameworks, such as Imprecise Probabilities and Credal Sets[Wal91].

2.3 Imprecise Probabilities and Credal Sets

Traditionally probability theory operates on the assumption that all forms of uncertainty can be adequately represented by a single, precise probability distribution [Wal91]. Under this paradigm, an agent or model is required to assign an exact numerical probability to every possible event, regardless of the amount of evidence available. While computationally convenient, this precise framework struggles to model severe epistemic uncertainty. When a model lacks sufficient training data or encounters an ambiguous query, forcing it to produce a single probability distribution often results in arbitrary or misleadingly confident predictions. Precise probabilities mathematically conflate the lack of evidence (epistemic uncertainty) with known stochastic noise (aleatoric uncertainty), masking the model's true state of ignorance [HW21].

2.3.1 Introduction to Credal Sets

To address the limitations of precise probabilities, the framework of **imprecise probabilities** replaces the single distribution with a set of plausible distributions [Lev80; Wal91]. A closed, convex set of probability distributions is known as a **credal set** [Coz00], denoted by Q.

2.3.2 Computing Uncertainty on Credal Sets

To quantify the total uncertainty of a model's prediction using a credal set, we must extend standard metrics like Shannon Entropy to operate over sets of distributions. Since we have a set of probability distributions Q, we can determine the **lower entropy** and **upper entropy** by finding the minimum and maximum entropy values across the set [AKM06; HW21]:

$$H_*(Q) = \min_{q \in Q} H(q), \quad H^*(Q) = \max_{q \in Q} H(q) \tag{2.11}$$

We can use these bounds to quantify total uncertainty $U(Q)$, aleatoric uncertainty $AU(Q)$ and epistemic uncertainty $EU(Q)$ [AKM06; HW21]:

$$U(Q) = AU(Q) + EU(Q) \tag{2.12}$$

$$H^*(Q) = H_*(Q) + (H^*(Q) - H_*(Q)) \tag{2.13}$$

Here, upper entropy represents total uncertainty, lower entropy represents aleatoric uncertainty and the difference between them is used as a measure for epistemic uncertainty.

2.4 Relative Likelihood and Valid Temperature Sets

To systematically construct a credal set, we need a rigorous criterion to distinguish between plausubile and implausible probability distributions. Following recent advances in credal prediction [Löh+25], we achieve this by employing the statistical concept of **relative likelihood**.

2.4.1 Relative Likelihood

In standard maximum likelihood estimation, the goal is to find the single best parameter that maximizes the likelihood of the observed data. In the context of our method, we treat the decoding temperature $\tau \in \mathcal{T}$ as the parameter of interest. Let $L(\tau)$ denote the likelihood of the target sequence generated at temperature τ. Furthermore, let τ^{ML} be the optimal temperature that yields the highest likelihood (or equivalently, minimizes the average Negative Log-Likelihood). To characterize the plausibility of any other temperature τ, we compute its relative likelihood $\gamma(\tau)$, defined as the ratio of its likelihood to the maximum likelihood [Bir62; Was90; WM99]:

$$\gamma(\tau) = \frac{L(\tau)}{L(\tau^{ML})} = \frac{L(\tau)}{\sup_{\tau' \in \mathcal{T}} L(\tau')} \tag{2.14}$$

By definition, $\gamma(\tau)$ is bounded in the interval $[0, 1]$, where $\gamma(\tau^{ML}) = 1$. The relative likelihood provides an intuitive measure of how well a specific temperature explains the data copmared to the optimal temperature setting.

2.4.2 Constructing α-Cuts for Temperatures

On the basis of the relative likelihood, a set of valid temperatures can be constructed by including only those that are plausible in the sense of surpassing a specific threshold $\alpha \in [0, 1]$. This set is formally referred to as an α**-cut** [ACC12]:

$$\mathcal{C}_\alpha = \{\tau \in \mathcal{T} : \gamma(\tau) \geq \alpha\} \tag{2.15}$$

According to this definiton, a temperature τ is considered implausible, and is therefore excluded from the valid set, if its likelihood is too small compared to the likelihood of the best temperature, specifically if it is less than α times the optimal likelihood.

3 Related Work

The reliable quantification of uncertainty in LLMs has garnered significant attention in recent years. As researchers seek to mitigate hallucinations and improve model safety, various metrics have been proposed to estimate when a model's generation is likely to be incorrect. This chapter reviews the evolution of these methods, starting from traditional uncertainty quantification (UQ) metrics, and culminating in state-of-the-art approaches like Semantic Entropy, which serves as the primary baseline for the credal prediction methodology developed in this thesis.

3.1 Traditional Uncertainty Quantification (UQ)

Applying traditional UQ techniques to LLMs presents substantial challenges. The computational overhead of Deep Ensembles [LPB17] or Bayesian frameworks like Monte Carlo dropout [GG16] makes them impractical for natural language generation (NLG) in massive models. Initial approaches to estimating LLM uncertainty directly adapted standard classificatoin metrics or relied on heuristic comparisons. These traditional baselines generally fall into three categories: predictive entorpy, lexical similarity and self-evaluation.

Predictive Entropy Methods
The most straightforward methods evaluate the model's confidence using the probability distributions generated during decoding. Token-level entropy evaluates local uncertainty but fails to capture the holistic confidence of a factual claim. To address this, Malinin and Gales [MG21] proposed sequence-level predictive entorpy (often implemented as length-normalized entropy), which estimates uncertainty across all possible output sequences. However, because these methods operate strictly on exact token matches, they artificially inflate uncertainty when an LLM distributes probability mass across multiple valid phrasings of the same answer.

Lexical Similarity
To account for the fact that generated answers might differ slightly in their suface forms, researchsers adapted translation metrics to quantify uncertainty. By sampling multiple answers for a given question, one can compute the average lexical similarity between them using metrics like ROUGE-L or BLEU [Fom+20]. If the sampled answers have high n-gram overlap, the model is deemed certain. While an improvement over strict sequence likelihood, lexical similarity still struggles with open-ended generation. An LLM might generate "Berlin" and "The capital of

Germany", which share zero lexical overlap (low ROUGE-L) but convey the exact same meaning, leading to false uncertainty signals.

Self-Evaluation
Another distinct approach treats the LLM as an independent evaluator of its own uncertainty. Methods like $p(\text{True})$ prompt the model to evaluate its own generated answer or ask it directly if a proposed fact is correct [Kad+22]. While intuitive, self-evaluation baselines are highly sensitive to the specific phrasing of the prompt.

All three of these traditional baseline families fail to reliably distinguish between benign linguistic flexibility and genuine factual ignorance. Traditional metrics treat these lexical variations as disjoint, competing hypotheses. This critical limitation necessitated a paradigm shift from lexical (word-based) probability to semantic (meaning-based) probability, leading to the development of Semantic Entropy [KGF23].

3.2 Semantic Entropy

To resolve the conflation of lexical flexibility and factual ignorance, Kuhn et al. [KGF23] introduced the concept of **Semantic Entropy (SE)**. Semantic Entropy shifts the unit of uncertainty measurement from the exact sequence of words to the underlying meaning, or *semantics*, of the generated text. Because SE serves as both the primary baseline and a core mechanical component of the methodology proposed in this thesis, we detail its formal computation below. The SE itself is computed in three steps:

1. Generating M sequences by the same model
2. Clustering the generated sequences by semantic equivalence
3. Computation of the semantic entropy

3.2.1 Semantic Clustering

The core premise of SE is that the infinite space of possible lexical sequences generated by an LLM can be partitioned into a finite set of discrete meanings.

Let x be the input prompt, and let $\mathcal{S} = \{s^{(1)}, s^{(2)}, \ldots, s^{(M)}\}$ be a set of M sequences generated by the model. We define a bidirectional semantic equivalence relation between sequences, denoted as $E(\cdot, \cdot)$, which holds true if sequence $s^{(i)}$ and sequence $s^{(j)}$ convey the exact same factual information, i.e., they entail, in the context of the prompt x. This equivalence relation groups the generated sequences into distinct

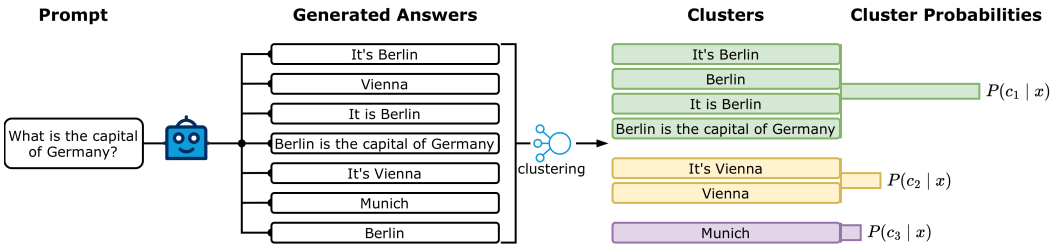

Fig. 3.1.: Visualization of Semantic Clustering

semantic clusters (or equivalence classes), denoted as $C = \{c_1, c_2, \ldots, c_K\}$, where $\forall s^{(i)}, s^{(j)} \in c_k : E(s^{(i)}, s^{(j)})$ and $K \leq M$. In practice, this clustering is performed using an automated evaluation model, such as a Natural Language Inference (NLI) classifier (Kuhn et al. [KGF23] used the DeBERTa-large-MNLI model [He+21]).

3.2.2 Computing the Semantic Entropy

Once the sequences are clustered, the probability of a specific meaning (a cluster $c_k \in C$) is computetd by marginalizing over the probabilities of all individual lexical sequences within that cluster. The cluster probability is given by:

$$P(c_k \mid x) = \sum_{s \in c_k} P(s \mid x) \tag{3.1}$$

By treating the clusters as the new categorical variables, the SE is calculated as the Shannon Entropy over the cluster probabilities:

$$\text{SE}(x) = -\sum_{k=1}^{K} P(c_k \mid x) \log P(c_k \mid x) \tag{3.2}$$

Because we can not know every possible meaning-class c, we can only sample c from the given distribution by the model. Thus Kuhn et al. [KGF23] use Monte Carlo integration over the semantic equivalence classes to estimate the expectation in equation 3.2:

$$\text{SE}(x) \approx -|C|^{-1} \sum_{i=1}^{|C|} \log P(C_i \mid x) \tag{3.3}$$

where

$$P(C_i \mid x) = \frac{P(c_i \mid x)}{\sum_{c \in C} P(c \mid x)} \tag{3.4}$$

SE represents a significant advancement over raw sequence likelihood. But it relies on a single, static probability distribution. In the standard implementation by Kuhn et al. [KGF23], sequences are sampled, and probabilities are evaluated at a single, fixed temperature (Kuhn et al. [KGF23] state that $\tau = 0.5$ is optimal).

As established in Section 2.3, precise probabilities are brittle when a model is miscalibrated or highly ignorant. This gap motivates the primary contribution of this thesis. Instead of computing SE on a single distribution, we propose computing it across a **Credal Set of distributions** generated by a valid range of temperatures. By analyzing how semantic cluster probabilities fluctate across this set, yielding upper and lower cluster bounds, we detect epistemic uncertainty that standard SE is blind to.

3.3 Credal Prediction based on Relative Likelihood

While SE addresses the issue of lexical equivalence in LLMs, it lacks a mechanism to handle severe epistemic uncertainty arising from a poorly calibrated, single probability distribution. To address this, we look to recent advancements in the field of imprecise probabilities, specifically the application of Credal Sets to deep learning.

3.3.1 Relative Likelihood Ensembles

Recently, Löhr et al. [Löh+25] proposed a theoretically grounded approach to credal prediction based on the statistical notion of relative likelihood. Instead of relying on a single maximum likelihood estimator (MLE), their framework defines the prediction target as a credal set induced by all plausible models whose relative likelihood exceeds a specific threshold α (the α-cut). Specifically, they introduce the Credal Relative Likelihood ($CreRL_\alpha$) method, which trains an ensemble of neural networks using an early-stopping strategy. Each model in the ensemble is trained until it reaches a sprecific relative likelihood threshold τ_i, ensuring that the final ensemble spans the entire valid α-cut. To guarantee diversity among the ensemble members, they further employ a novel "ToBias" initialization strategy, which initializes the models to predict degenerate distributions at the vertices of the probability simplex.

3.3.2 Bridging the Gap: From Ensembles to LLM Temperatures

The $CreRL_\alpha$ framework successfully demonstrates that relative likelihood can be used to construct robust credal sets, achieving superior coverage on standard classifiction benchmarks. However, this approach, like other credal deep learning models (e.g., Credal Bayesian Neural Networks), is designed for standard discriminative classification tasks where the model weights are optimized during training.

Applying this weight-space ensemble directly to modern LLMs is computationally prohibitive due to their massive size, and it does not cleanly translate to the autoregressive, open-ended generation of sequences.

4 Methodology

Because calculating the product of raw probabilities for long text sequences inevitably leads to numeraical underflow as shown in Section 2.2.2, the relative likelihood constraint $\gamma(\tau) \geq \alpha$ is implemented in the log domain using the Negative Log-Likelihood (NLL). Taking the negative logarithm of both sides yields the mathematical equivalent:

$$\text{NLL}(\tau) - \text{NLL}(\tau^{ML}) \leq -\log(\alpha) \tag{4.1}$$

5 Conclusion

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.1 System Section 1

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.2 System Section 2

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like "Huardest gefburn"? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look

like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

5.3 Future Work

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Example Appendix

A

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

A.1 Appendix Section 1

And after the second paragraph follows the third paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Alpha	Beta	Gamma
0	1	2
3	4	5

Tab. A.1.: This is a caption text.

A.2 Appendix Section 2

After this fourth paragraph, we start a new paragraph sequence. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like

“Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Alpha	Beta	Gamma
0	1	2
3	4	5

Tab. A.2.: This is a caption text.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Bibliography

[AKM06] J. Abellán, G.J. Klir, and S. Moral. "Disaggregated total uncertainty measure for credal sets". In: *International Journal of General Systems* 35.1 (2006), pp. 29–44. eprint: https://doi.org/10.1080/03081070500473490 (cit. on p. 9).

[ACC12] Alessandro Antonucci, Marco E. G. V. Cattaneo, and Giorgio Corani. "Likelihood-Based Robust Classification with Bayesian Networks". In: *Advances in Computational Intelligence. 14th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2012, Catania, Italy, July 9-13, 2012, Proceedings, Part III*. Ed. by Salvatore Greco, Bernadette Bouchon-Meunier, Giulianella Coletti, et al. Berlin: Springer, 2012, pp. 491–500 (cit. on p. 10).

[BCB16] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL] (cit. on p. 3).

[Bir62] Allan Birnbaum. "On the Foundations of Statistical Inference". In: *Journal of the American Statistical Association* 57.298 (1962), pp. 269–306. eprint: https://doi.org/10.1080/01621459.1962.10480660 (cit. on p. 10).

[Bri90] John S. Bridle. "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition". In: *Neurocomputing*. Ed. by Françoise Fogelman Soulié and Jeanny Hérault. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 227–236 (cit. on p. 3).

[Coz00] Fabio G. Cozman. "Credal networks". In: *Artificial Intelligence* 120.2 (2000), pp. 199–233 (cit. on p. 9).

[FLD18] Angela Fan, Mike Lewis, and Yann Dauphin. *Hierarchical Neural Story Generation*. 2018. arXiv: 1805.04833 [cs.CL] (cit. on p. 5).

[Fom+20] Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, et al. *Unsupervised Quality Estimation for Neural Machine Translation*. 2020. arXiv: 2005.10608 [cs.CL] (cit. on p. 11).

[GG16] Yarin Gal and Zoubin Ghahramani. *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. 2016. arXiv: 1506.02142 [stat.ML] (cit. on p. 11).

[GR07] Tilmann Gneiting and Adrian E Raftery. "Strictly Proper Scoring Rules, Prediction, and Estimation". In: *Journal of the American Statistical Association* 102.477 (2007), pp. 359–378. eprint: https://doi.org/10.1198/016214506000001437 (cit. on p. 8).

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016 (cit. on p. 7).

[Guo+17] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. *On Calibration of Modern Neural Networks*. 2017. arXiv: 1706.04599 [cs.LG] (cit. on pp. 6, 8).

[He+21] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*. 2021. arXiv: 2006.03654 [cs.CL] (cit. on p. 13).

[Hol+20] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. *The Curious Case of Neural Text Degeneration*. 2020. arXiv: 1904.09751 [cs.CL] (cit. on p. 5).

[HW21] Eyke Hüllermeier and Willem Waegeman. "Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods". In: *Machine Learning* 110.3 (Mar. 2021), pp. 457–506 (cit. on pp. 6–9).

[Ipp+19] Daphne Ippolito, Reno Kriz, Maria Kustikova, João Sedoc, and Chris Callison-Burch. *Comparison of Diverse Decoding Methods from Conditional Language Models*. 2019. arXiv: 1906.06362 [cs.CL] (cit. on p. 4).

[Kad+22] Saurav Kadavath, Tom Conerly, Amanda Askell, et al. *Language Models (Mostly) Know What They Know*. 2022. arXiv: 2207.05221 [cs.CL] (cit. on p. 12).

[KG17] Alex Kendall and Yarin Gal. *What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?* 2017. araXiv: 1703.04977 (cs.CV) (cit. on pp. 6, 7).

[KGF23] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. *Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation*. 2023. arXiv: 2302.09664 [cs.CL] (cit. on pp. 12–14).

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles*. 2017. arXiv: 1612.01474 [stat.ML] (cit. on p. 11).

[Lev80] Isaac Levi. *The Enterprise of Knowledge: An Essay on Knowledge, Credal Probability, and Chance*. MIT Press, 1980 (cit. on p. 9).

[Löh+25] Timo Löhr, Paul Hofman, Felix Mohr, and Eyke Hüllermeier. *Credal Prediction based on Relative Likelihood*. 2025. arXiv: 2505.22332 [stat.ML] (cit. on pp. 9, 14).

[LIH24] Timo Löhr, Michael Ingrisch, and Eyke Hüllermeier. "Towards Aleatoric and Epistemic Uncertainty in Medical Image Classification". In: *Artificial Intelligence in Medicine*. Ed. by Joseph Finkelstein, Robert Moskovitch, and Enea Parimbelli. Cham: Springer Nature Switzerland, 2024, pp. 145–155 (cit. on p. 6).

[MG21] Andrey Malinin and Mark Gales. *Uncertainty Estimation in Autoregressive Structured Prediction*. 2021. arXiv: 2002.07650 [stat.ML] (cit. on p. 11).

[MC18] Kenton Murray and David Chiang. *Correcting Length Bias in Neural Machine Translation*. 2018. arXiv: 1808.10006 [cs.CL] (cit. on p. 7).

[Nav+24] Humza Naveed, Asad Ullah Khan, Shi Qiu, et al. *A Comprehensive Overview of Large Language Models*. 2024. arXiv: 2307.06435 [cs.CL] (cit. on p. 3).

[Sha48] C. E. Shannon. "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x (cit. on p. 7).

[Shi+24] Chufan Shi, Haoran Yang, Deng Cai, et al. *A Thorough Examination of Decoding Methods in the Era of LLMs*. 2024. arXiv: 2402.06925 [cs.CL] (cit. on pp. 4–6).

[Tia+23] Haoye Tian, Weiqi Lu, Tsz On Li, et al. *Is ChatGPT the Ultimate Programming Assistant – How far is it?* 2023. arXiv: 2304.11938 [cs.SE] (cit. on p. 3).

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, et al. Vol. 30. Curran Associates, Inc., 2017 (cit. on pp. 3, 4).

[WM99] P. Walley and S. Moral. "Upper probabilities based only on the likelihood function". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 61.4 (1999), pp. 831–847. eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9868.00205 (cit. on p. 10).

[Wal91] Peter Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman & Hall, 1991 (cit. on pp. 8, 9).

[Was90] Larry A. Wasserman. "Belief functions and statistical inference". In: *Canadian Journal of Statistics* 18.3 (1990), pp. 183–196. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.2307/3315449 (cit. on p. 10).

[Wei+23] Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL] (cit. on p. 3).

List of Figures

List of Tables

Colophon

This thesis was typeset with LaTeX 2$_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at `http://cleanthesis.der-ric.de/`.