What Language Models Know But Don't Say: Non-Generative Prior Extraction for Generalization

Anonymous ACL submission

Abstract

In domains like medicine and finance, large-scale labeled data is costly and often unavailable, leading to models trained on small datasets that struggle to generalize to real-world populations. Large language models contain extensive knowledge from years of research across these domains. We propose LoID (Logit-Informed Distributions), a deterministic method for extracting informative prior distributions for Bayesian logistic regression by directly accessing their token-level predictions. Rather than relying on generated text, we probe the model's confidence in opposing semantic directions (positive vs. negative impact) through carefully constructed sentences. By measuring how consistently the LLM favors one direction across diverse phrasings, we extract the strength and reliability of the model's belief about each feature's influence. We evaluate LoID on ten real-world tabular datasets under synthetic out-of-distribution (OOD) settings characterized by covariate shift, where the training data represents only a subset of the population. We compare our approach against (1) standard uninformative priors, (2) AutoElicit, a recent method that prompts LLMs to generate expert-style priors via text completions, (3) LLMProcesses, a method that uses LLMs to generate numerical predictions through in-context learning by formatting tabular data as natural language prompts and (4) an oracle-style upper bound derived from fitting logistic regression on the full dataset. We assess performance using Area Under the Curve (AUC). Across datasets, LoID significantly improves performance over logistic regression trained on OOD data, recovering up to **59%** of the performance gap relative to the oracle model. LoID outperforms AutoElicit and LLMProcessesc on 8 out of 10 datasets, while providing a reproducible and computationally efficient mechanism for integrating LLM knowledge into Bayesian inference.

1 Introduction

Machine learning models used in critical domains like medicine and finance often lack access to large, well-balanced, and comprehensively labeled datasets. Clinical datasets may cover only a single hospital or demographic, while credit-risk tables can be skewed toward historical lending patterns. For example, Purushotham et al. (2017) benchmarked several deep learning models on clinical prediction tasks—including mortality and length of stay—using the MIMIC-III intensive care electronic health record dataset from a single hospital site. As a consequence, models trained on small, non-representative samples frequently fail to generalize when confronted with real-world populations that differ from their training distribution (Brigato and Iocchi, 2020; Li et al., 2024; Althnian et al., 2021). Bayesian methods offer a principled way to encode prior knowledge and mitigate such brittleness, but manually eliciting high-quality priors at scale remains a longstanding challenge (O'Hagan et al., 2006).

LLMs, trained on vast corpora of scientific literature and domain-specific data (Brown et al., 2020), offer a promising new source of priors for downstream modeling. Recent efforts have explored LLMs both as zero-shot and few-shot predictors for tabular classification (Hegselmann et al., 2023; Requeima et al., 2024), and as in-context learners capable of producing joint probabilistic predictions (Shysheya et al., 2025). Other work has investigated the integration of LLMs with classical statistical methods: LLM-FE (Abhyankar et al., 2025) uses LLMs to guide feature engineering, LLM-Lasso (Zhang et al., 2025) for feature selection, and LLMCorr (Zhong et al., 2024) for post-hoc model correction. AutoElicit (Capstick et al., 2025) introduced a technique to query LLMs for Gaussian priors, but its reliance on generative sampling introduces run-to-run variability.

Parallel to this, out-of-distribution (OOD) generalization has gained increasing attention, as real-world deployment often involves data that differs from the development environment. Meta-learning approaches like TabPFN (Hollmann et al., 2023) and Drift-Resilient TabPFN (Helli et al., 2024) address OOD robustness through synthetic task generation and causal modeling. Others have proposed logit-based calibration strategies such as UE-NL (Huang and Qiao, 2023), or LLM-guided data augmentation methods like Summary Boosting (Manikandan et al., 2023) and LLMOverTab (Isomura et al., 2025) to cover underrepresented regions of the feature space. However, despite this momentum, few works have systematically explored how to extract priors from the internal structure of LLMs (e.g. hidden states, logits, etc.) for tabular tasks under distribution shift.

We address this gap with **LoID (Logit-Informed Distributions)**, a deterministic framework that elicits informative priors for Bayesian logistic regression directly from an LLM's token-level logits. Instead of relying on generated completions or examples, LoID probes the LLM with semantically paired sentences highlighting positive and negative feature-target relationships. The logit difference between the relational tokens quantifies directional belief, and the variance across paraphrased prompts yields an uncertainty estimate—together forming Normal priors over model coefficients. These priors integrate into Bayesian workflows, without any fine-tuning or in-context learning.

We evaluate LoID on ten publicly available tabular datasets that are widely used benchmarks and have appeared in numerous prior studies, ensuring comparability with existing methods. We compare LoID against uninformative Bayesian inference and LLMProcesses. To provide an upper bound, we additionally train and evaluate a logistic regression model on the *entire* dataset, offering insight into performance when full population data is available. Across extensive out-of-distribution splits, LoID outperforms existing approaches in predictive accuracy (AUC). By deterministically extracting domain knowledge encoded in large language models, LoID enables robust and reproducible inference.

In summary, our contributions include:

- **Logit-based deterministic prior elicitation.** LoID is the first method to extract Bayesian priors directly from the token-level logits of large language models, eliminating the variability associated with text generation and prompt sampling.
- **Baseline comparison across ten OOD datasets.** We evaluate LoID on diverse OOD tabular datasets from healthcare, finance, and social science. Our experiments demonstrate that LoID often outperforms existing approaches across most settings.

2 Related Work

2.1 LLMs for Tabular Data Modeling

The integration of LLMs into tabular data modeling has garnered attention, particularly in addressing challenges such as limited data availability, complex feature interactions, and the necessity for domain-specific feature engineering.

Several studies have explored the use of LLMs as standalone predictors for tabular data. Hegselmann et al. (2023) introduce TabLLM, a framework that applies large language models to zero-shot and few-shot classification of tabular data by converting rows into natural-language strings, enabling effective classification even with minimal labeled examples. Requeima et al. (2024) propose LLM-Processes, a framework that enables LLMs to produce probabilistic regression outputs by conditioning on both numerical data and natural language descriptions of prior knowledge. Shysheya et al. (2025) present JOLT, a method that leverages the in-context learning capabilities of LLMs to define joint probabilistic predictions over tabular data.

Beyond standalone applications, recent research has focused on leveraging LLMs to enhance classical statistical models. For instance, Abhyankar et al. (2025) introduced LLM-FE, a framework that combines evolutionary search with the reasoning capabilities of LLMs to iteratively generate and refine feature transformations for tabular data. Similarly, LLM-Lasso (Zhang et al., 2025) is a domain-informed approach to feature selection and regularization by leveraging large language models to guide the selection process. Capstick et al. (2025) proposed AutoElicit, a method that interacts with LLMs to extract priors for Gaussian-based probabilistic models; however, this approach primarily focuses on Gaussian priors and lacks flexibility in handling other priors that may be more suitable for different data characteristics. In a different vein, Zhong et al. (2024) presented LLMCorr, a training-free framework where LLMs act as post-hoc correctors to propose corrections for the predictions

of arbitrary machine learning models. Liu et al. (2023) investigated the reasoning capabilities of LLMs over tabular data and proposed combining symbolic and textual reasoning to improve performance on table-based question answering tasks.

2.2 Out-of-Distribution Generalization

OOD generalization is a fundamental challenge in real-world applications of tabular machine learning. Models trained on limited or biased datasets often fail to generalize when deployed on test data that differ from the training distribution. A variety of approaches have been proposed to address this issue through data augmentation, distributionally robust optimization, causal modeling, Bayesian inference, and most recently, the use of LLMs.

One line of work focuses on meta-learning for small-data generalization. Hollmann et al. (2023) introduced TabPFN, a foundation model trained on millions of synthetic tabular tasks to approximate Bayesian posterior predictive distributions in one forward pass. The model shows strong generalization across novel tasks and distributions. Building on this, Helli et al. (2024) proposed Drift-Resilient TabPFN, which introduces temporally shifting causal mechanisms during training to improve robustness under distribution shift.

Several recent studies have explored logit-based and uncertainty-aware methods for detecting or adapting to OOD inputs. Huang and Qiao (2023) proposed Uncertainty Estimation with Normalized Logits (UE-NL), a calibration technique that regularizes softmax confidence and improves both in-distribution accuracy and OOD detection.

In the LLM context, OOD robustness has been approached in several creative ways. Summary Boosting (Manikandan et al., 2023) uses GPT-generated feature summaries as weak learners to improve robustness in small or under-specified datasets. LLMOverTab (Isomura et al., 2025) performs data augmentation by prompting LLMs to generate realistic examples, improving classifier performance on underrepresented regions of the feature space. These strategies demonstrate that LLMs can inject external knowledge or expand the training distribution to mitigate generalization failure.

Our work, LoID, complements and extends this growing body of research by proposing the first deterministic logit-based method to elicit Bayesian priors from LLMs for structured tabular tasks. Our method consistently improves OOD generalization across ten tabular datasets, offering a lightweight and almost deterministic integration of LLM knowledge into classical statistical pipelines.

3 Method

We propose LoID (Logit-Informed Distribution), a deterministic framework for extracting priors for tabular dataset features from LLMs using token-level logits, rather than relying on generated samples. Our goal is to derive informative priors for Bayesian logistic regression that generalize robustly in low-data settings.

3.1 Problem Setting

Let $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{n}$ denote a training dataset, where each $x_i \in \mathbb{R}^d$ is a vector of d features and $y_i \in \{0, 1, \ldots, m\}$ is the corresponding label. In Bayesian logistic regression, we model the likelihood as

$$p(y_i = t \mid x_i, \boldsymbol{\beta}) = \sigma(x_i^\top \boldsymbol{\beta}),$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\boldsymbol{\beta} \in \mathbb{R}^d$ represents the regression coefficients. Bayesian inference is performed by specifying a prior distribution $p(\boldsymbol{\beta})$ and computing the posterior $p(\boldsymbol{\beta} \mid \mathcal{D}_{\text{train}})$. In data-scarce regimes, informative priors can substantially improve generalization. LoID constructs such priors by leveraging domain knowledge encoded in large language models, without access to the training data.

3.2 Querying LLM Beliefs via Logit Differences

For each feature x_j, we construct paired prompts that express opposing semantic claims about its effect on the target variable. Each prompt is carefully designed such that all relevant context appears before a sentiment-bearing token (e.g., "positive" or "negative"), ensuring that the token's probability considers the entire context for the task, including the name of the feature and the prediction task (see examples in A.2.1). For example, given a prompt prefix like:

"The impact of f_j on t is "

we use the probabilities from the LLM's softmax over its vocabulary to extract the probability of the next token being "positive" or "negative". We define:

$$\mathcal{P}_j^+ = P(\text{“positive”} \mid \text{prefix})$$
$$\mathcal{P}_j^- = P(\text{“negative”} \mid \text{prefix})$$

where f_j is a natural language description of feature x_j (e.g., "age of the patient") and t describes the prediction task (e.g., "disease risk").

We define the **logit preference score** for a single prompt p_j as:

$$\text{logit}(p_j) = \log\left(\frac{p_j}{1-p_j}\right)$$

Where

$$p_j = \frac{P_j^+}{P_j^+ + P_j^-}$$

The logit preference score quantifies the extent to which the model prefers the token "positive" over "negative" in a given context. While this difference may also reflect other patterns learned during training, our hypothesis is that by constructing sentences that do not grammatically or syntactically favor either token, we reduce the influence of such confounding factors. As a result, the observed preference is driven primarily by the semantic relationship between the feature and the target concept. Finally, applying the logit transformation maps probabilities from the bounded interval (0, 1) to the unbounded real line, making the resulting scores directly appliable as coefficients in logistic regression models.

The **mean prefrence score** μ_j for feature x_j is given by averaging this score across a set of paraphrased prompts T:

$$\mu_j = \frac{1}{|T|}\sum_{t\in T}\text{logit}_t(p_j)$$

This value reflects the LLM's normalized preference for a positive versus negative association with the feature across diverse prompt formulations. Figure 1 demonstrates how LoID extracts informative priors from model logits.

3.3 Constructing Priors from Logits

We model each coefficient β_j with a Gaussian prior: $\beta_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$. The **mean** μ_j is the average logit preference score across paraphrases, which reflects the direction and strength of influence the LLM assigns to feature x_j. To capture uncertainty in the LLM's belief, we define the standard deviation σ_j as a scaled function of variability in the individual logit preference scores:

$$\sigma_j^2 = \alpha + \gamma \cdot \text{std}(logit_1(p_j), \ldots, logit_{|T|}(p_j))$$

Here, $logit_t(p_j)$ denotes the logit preference score for prompt t and feature j. $\alpha > 0$ and $\gamma > 0$ are fixed scaling hyperparameters that can be tuned to optimize LoID's performance on a certain task. We perform a comprehensive hyperparameter tuning in A.2.4 to find the hypermarameters that achieve the highest cumulative AUC. This design assigns higher variance to features for which the LLM expresses inconsistent preferences across paraphrases, thereby reflecting greater uncertainty about the feature's influence in the model.

3.4 Bayesian Inference with LoID Priors

With the LoID priors in place, posterior inference is performed using the No-U-Turn Sampler (NUTS), a variant of Hamiltonian Monte Carlo (HMC). HMC augments the parameter space with auxiliary momentum variables and explores the posterior by simulating Hamiltonian dynamics, enabling efficient, gradient-informed proposals that avoid the random-walk behavior of traditional MCMC methods.

Specifically, HMC introduces a momentum variable $\mathbf{r}$ for the regression coefficients $\boldsymbol{\beta}$ and defines the Hamiltonian

$$\mathcal{H}(\boldsymbol{\beta}, \mathbf{r}) = -\log p(\boldsymbol{\beta} \mid \mathcal{D}_{\text{train}}) + \frac{1}{2}\mathbf{r}^\top\mathbf{r}.$$

Proposals are generated by numerically simulating the resulting Hamiltonian dynamics using a leapfrog integrator, which leverages gradients of the log-posterior to propose distant yet highly probable states.

NUTS eliminates the need to manually tune the trajectory length by adaptively extending the simulation until a U-turn condition is detected, ensuring efficient exploration while maintaining detailed balance. This results in faster mixing and improved effective sample sizes, particularly in high-dimensional parameter spaces.

We run multiple NUTS chains with warm-up steps for step-size and mass-matrix adaptation, and discard burn-in samples before collecting posterior draws. All reported results are computed using posterior expectations estimated from these samples.

3.5 Summary

LoID transforms the implicit domain knowledge encoded in large language models into explicit, feature-level priors suitable for probabilistic modeling. Unlike prior approaches that rely on generated samples, in-context learning, or fine-tuning, LoID

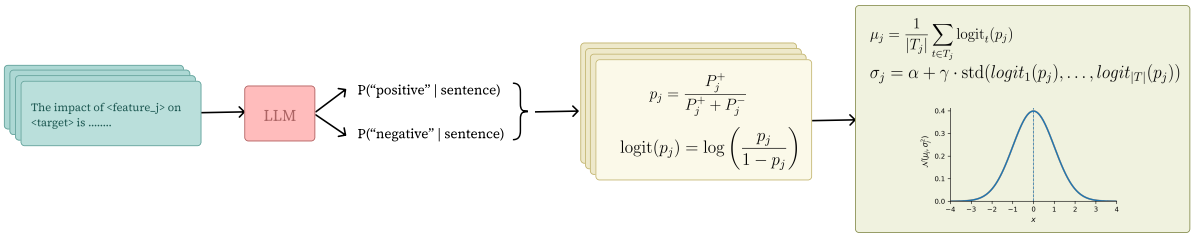

Figure 1: LoID Pipeline: To extract informative priors, LoID uses multiple sentence templates for each feature and computes the probabilities of the tokens "positive" and "negative" as the next token. From these probabilities, we calculate the logit preference score, which quantifies the model's preference for one token over the other. Using the logit preference scores from all sentences, we compute their mean and standard deviation. We then apply a transformation to the standard deviation to reduce overconfidence, allowing the data to have more influence on the resulting distribution.

achieves this deterministically by directly probing token-level logits. This design enables reproducibility and principled integration into Bayesian models.

3.6 Evaluation

To assess the quality of the resulting posteriors, we compute the AUC score on the *entire* dataset using models trained solely on the OOD training split. We then compare this AUC to that of a logistic regression trained on the *entire* dataset. This comparison quantifies how much of the predictive performance gap, caused by training on an OOD distribution, is recovered by our method.

4 Experimental Setup

4.1 Datasets

We evaluate and compare LoID's performance on ten publicly available tabular datasets for binary and multi-class classification: The Heart Disease (Cleveland) dataset (Janosi and Detrano, 1989) predicts the presence of heart disease based on clinical and demographic attributes such as cholesterol level, and blood pressure. The Indian Liver Patient Dataset (ILPD) (Ramana and Venkateswarlu, 2022) classifies whether a patient has liver disease using demographic information and blood test measurements. The Blood Transfusion Service Center dataset (Yeh, 2008a) predicts whether a donor will donate blood again based on historical donation behavior. The Room Occupancy Detection dataset (Candanedo, 2016a) determines whether a room is occupied using time-series environmental sensor readings.

The Diabetes dataset (Islam et al., 2020) predicts the likelihood of diabetes onset based on self-reported symptoms. The Pima Indians Diabetes dataset (Smith et al., 1988) predicts the presence of diabetes using diagnostic measurements such as glucose concentration and BMI. The Give Me Some Credit (Good Credit Score) dataset (Fusion and Cukierski, 2011) assesses an individual's creditworthiness based on income, debt ratio, and delinquency history. The Bank Marketing dataset (Moro et al., 2014) predicts whether a client will subscribe to a term deposit based on demographic attributes and marketing interactions.

The Adult Income (Census Income) dataset (Becker and Kohavi, 1996b) predicts whether an individual's annual income exceeds $50K using demographic and educational features. Finally, the Jungle Chess (Jungle) dataset (Shapiro, 1983) predicts game outcomes based on structured representations of board states, providing a symbolic decision-making task.

For each dataset, we construct synthetic OOD splits as described in Section A.4.2. To verify that the resulting training sets do not reflect the true data distribution, a logitic regression model is trained separately on the out-of-distribution and in-distribution training splits and evaluated on the entire dataset. As shown in Table 1, the model trained on the OOD train set consistently underperforms compared to the model trained on the full dataset. This confirms that generalization is impaired and the OOD train set does not present the full distribution of the dataset as well as the in distribution train set.

Across all datasets, we apply the same preprocessing pipeline: categorical features are one-hot encoded, and missing values are imputed with zeros. For datasets with an excessive number of features—namely *Give Me Some Credit*, and *Adult Income*—we perform feature selection using GPT-4o by providing feature descriptions and prompting

it to return a list of the 15 most relevant features for the prediction task. The same reduced feature set is used consistently across all experimental conditions to ensure comparability.

Dataset	AUC (IID)	AUC (OOD)
Heart Disease	0.93	0.86
Liver Disease	0.77	0.70
Blood Donation	0.75	0.58
Room Occupancy	1.00	0.98
Diabetes	0.84	0.81
Income	0.91	0.78
Credit	0.79	0.71
Jungle	0.81	0.71
Pima Diabetes	0.84	0.81
Bank Marketing	0.89	0.51

Table 1: AUC comparison of logistic regression on IID vs. OOD test splits. Lower performance on OOD indicates distribution shift.

We designate the *Heart Disease*, *Blood Donation*, and *Diabetes* datasets as development datasets for evaluating prompt structures and hyperparameter tuning. The remaining five datasets are reserved for testing the generalization performance of the final LoID method. See (A.2.4) for details on the hyperparameter search process and value ranges explored.

4.2 LLM-Based Prior Extraction

We use the **Gemma-2-27B**, **Gemma-3-27B**, **llama-3-8B**, and **GPT-2** large language models to extract logit-based priors as described in Section 3.2. We use these models in *zero-shot* mode without any fine-tuning or in-context examples. In our experiments, the queried lexical items are represented as single tokens across models, enabling direct logit-based comparisons.

Each dataset is processed independently. For each feature, we issue 10 prompts to extract token-level log-probabilities for the `positive` and `negative` completions. The number of prompts was selected via hyperparameter tuning (Section A.2.4). The resulting feature-wise prior means and variances are then used to define informative Gaussian priors in a Bayesian logistic regression model. Posterior inference is performed using PyMC with a Laplace approximation.

We compare LoID to the following baselines:

- **OOD-LR**: Logistic regression model trained on OOD train set.
- **AutoElicit**: LLM-generated expert-style priors via sampled completions.
- **LLMProc (LLMProcesses)**: In-context learning approach that generates numerical predictions by formatting tabular data as natural language prompts. For fair comparison, we use the OOD training examples as few shot examples for the LLM.
- **Cap**: An oracle-style upper bound achieved by training a logistic regression model on the entire dataset, simulating access to the full population.

We evaluate each method using AUC score obtained from training on the ood train set and testing on the entire dataset.

5 Results

Table 2 summarizes the performance of each method across ten tabular datasets. Based on our experiments in Section A.2.2, Gemma-2-27B achieves the highest cumulative AUC among the evaluated LLMs. Therefore, Table 2 reports results obtained using this model.

5.1 AUC Comparison

LoID outperforms logistic regression trained on the OOD training set for all datasets except Liver Disease. Notably, on the Diabetes and Pima Diabetes datasets, LoID almost completely closes the performance gap induced by OOD training, indicating that it can successfully infer the underlying data distribution. On average, LoID closes 59% of the performance gap introduced by the OOD training setting.

LLMProcesses often underperforms even compared to uninformative priors. Since this approach does not explicitly produce priors, it is difficult to determine the exact cause of its poor performance. One possible explanation is its complete reliance on LLMs without leveraging traditional machine learning models, which are often more robust at pattern extraction in structured data.

LoID also achieves superior performance on 8 out of 10 datasets, achieving an average improvement of 99.7% in gap closed when it outperforms AutoElicit. To further understand these performance differences, we analyze the elicited prior distributions produced by both methods across selected winning and losing datasets. Section A.5.1 presents this comparison.

Dataset	OOD-LR	AutoElicit (baseline)	LLMProc (baseline)	LoID (our method)	Cap (upper bound)	Gap Closed%
Heart Disease	0.87	**0.92**	0.86	0.90	0.93	+50
Liver Disease	**0.70**	0.36	0.60	0.69	0.77	-19
Blood Donation	0.58	**0.71**	0.63	0.67	0.75	+52
Room Occupancy	0.98	0.97	0.80	**1.00**	1.00	+78
Diabetes	0.81	0.83	0.70	**0.84**	0.84	+99
Good Credit Score	0.71	0.71	0.30	**0.76**	0.79	+60
Income	0.78	0.85	0.51	**0.87**	0.91	+57
Pima Diabetes	0.82	0.83	0.74	**0.84**	0.84	+88
Bank Marketing	0.51	0.51	0.51	**0.72**	0.89	+56
Jungle Chess	0.71	0.78	0.62	**0.79**	0.81	+70

Table 2: Comparison of AUC scores across datasets. "OOD-LR" reports the performance of logistic regression trained on an OOD training set. LoID results are obtained using priors from Gemma-2-27B, with $\alpha = 0.2$, $\gamma = 2.0$ and $N_{sent} = 10$. Among Uniformative, AutoElicit, LLMProcesses, and LoID, the highest AUC for each dataset is shown in bold. The Cap condition serves as an oracle-style upper bound. All values are rounded to two decimal places. In some cases, LoID and Cap appear to have similar performance; however, the gap is not fully closed. This is due to rounding. For example, on Diabetes, LoID achieves 0.839 while Cap achieves 0.842.

Finding 1: Non-Monotonic Relationship Between Prior Uncertainty and Performance

- **Optimal range ($\sigma \approx 0.4$–0.5):** Achieves best performance on 7 datasets: Liver, Income, Bank, Creditg, Diabetes, Pima, Occupancy
- **Too weak ($\sigma \gtrsim 0.8$):** Underperforms AutoElicit: Blood, Heart

This shows that moderately weak priors ($\sigma \approx 0.4$–0.5) provide sufficient regularization while allowing the Bayesian posterior to correct systematic biases in the elicited means through the data likelihood. Excessively weak priors ($\sigma \gtrsim 0.8$) fail to leverage LLM knowledge effectively, approaching uninformed priors.

LoID excels when Prior uncertainty is moderate, datasets have sufficient samples to correct sign biases, and AutoElicit's tight priors impose incorrect strong beliefs.

Finding 3: Superior Mean Estimation Despite Directional Bias While LoID exhibits systematic positive bias in sign coverage (0–6% negative priors vs. AutoElicit's 20–50%), the absolute quality of the mean estimates appears superior. Evidence from performance gaps suggests:

1. **Magnitude accuracy over sign accuracy:** On datasets where LoID wins, the improvement over AutoElicit indicates that the LLM-elicited means—despite sign bias—capture feature importance better than AutoElicit's conversational approach.
2. **Comparison to uninformed priors:** LoID achieves 145% better gap closed vs. uninformed priors (see full comparison in A.6.1), demonstrating that the mean estimates contain genuine signal beyond just the regularization effect of uncertainty.
3. **Interaction with uncertainty:** The combination of (a) approximately correct magnitudes in means and (b) moderate uncertainty allows the Bayesian posterior to correct sign errors while preserving magnitude information. In contrast, AutoElicit's tight uncertainties amplify errors in both magnitude and direction.

5.2 Computational Efficiency.

LoID requires a fixed number of LLM queries per feature and performs prior elicitation once per dataset, resulting in a lower computational cost than AutoElicit and LLMProcesses. Across all datasets, LoID achieves a consistent **1.4–1.5× speedup** over AutoElicit during prior elicitation. LLMProcesses on the other hand, scales linearly with test set size and incurs significantly longer runtimes, requiring **15–20 hours** for datasets with 100–400 test instances, compared to **minutes** for prior-based methods. A detailed complexity and runtime analysis is provided in Appendix A.5.2.

6 Discussion

The success of our LoID method challenges the conventional assumption that more accurate prior means necessarily lead to better performance. AutoElicit provides strong, confident priors ($\sigma \approx 0.05$–0.12) with ∼40–50% correct negative signs. However, the incorrect confident priors harm inference by overpowering the data likelihood.

On the other hand, LoID provides weaker and uncertain priors ($\sigma \approx 0.4$–0.5). This allows the weaker means to be overridden by data, providing soft regularization: the priors gently guide inference toward reasonable parameter spaces without rigidly enforcing incorrect beliefs. The result is that even when LoID's directional beliefs are wrong, the posterior can recover because the prior is not insisting on its incorrectness. However, the benefits of this approach diminish when priors become too weak ($\sigma > 0.8$).

7 Limitations and Future Work

While our results demonstrate the effectiveness of LoID, several limitations should be noted, which also point toward directions for future work.

Semantic granularity. Our current approach probes binary directional relationships (positive vs. negative). Extending this framework to multi-token or scalar outputs could capture more nuanced associations, such as non-monotonic effects.

Hyperparameter selection: Although we use fixed hyperparameters ($\alpha = 0.2$, $\gamma = 2.0$) across all test datasets, optimal values vary by task. Developing automated calibration methods or meta-learning approaches for selecting these parameters would improve out-of-the-box applicability.

LLM access requirements: LoID requires access to token-level logits, limiting compatibility with closed-source APIs.

8 Conclusion

In this work, we introduce LoID, a deterministic method for extracting priors from LLMs. We show that priors elicited with LoID substantially mitigate the performance degradation caused by training on small, out-of-distribution datasets, recovering over 59% of the lost performance.

Our comparative analysis with AutoElicit demonstrates that the effectiveness of the LoID approach stems from honest uncertainty quantification, and correct mean magnitude estimation for logistic regression coefficients. By maintaining moderate prior uncertainty ($\sigma \approx 0.4$–0.5), LoID provides soft regularization while allowing the Bayesian posterior to correct systematic biases through the data likelihood.

As a result, LoID outperforms AutoElicit on 80% of benchmark datasets, achieving an average improvement of 99.7% in gap closed when it wins. Against LLMProcesses, LoID achieves superior performance on all 10 datasets, with particularly dramatic gains on datasets where LLMProcesses's end-to-end LLM approach fails to leverage structured data patterns.

References

Nikhil Abhyankar, Parshin Shojaee, and Chandan K. Reddy. 2025. Llm-fe: Automated feature engineering for tabular data with llms as evolutionary optimizers. *Preprint*, arXiv:2503.14434.

Alhanoof Althnian, Duaa AlSaeed, Heyam Al-Baity, Amani Samha, Alanoud Bin Dris, Najla Alzakari, Afnan Abou Elwafa, and Heba Kurdi. 2021. Impact of dataset size on classification performance: An empirical evaluation in the medical domain. *Applied Sciences*, 11(2).

Barry Becker and Ron Kohavi. 1996a. Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

Barry Becker and Ronny Kohavi. 1996b. Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

L. Brigato and L. Iocchi. 2020. A close look at deep learning with small data. *Preprint*, arXiv:2003.12843.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Luis Candanedo. 2016a. Occupancy Detection . UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5X01N.

Luis M. Candanedo. 2016b. Occupancy Detection. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5X01N.

Alexander Capstick, Rahul G. Krishnan, and Payam Barnaghi. 2025. Autoelicit: Using large language models for expert prior elicitation in predictive modelling. *Preprint*, arXiv:2411.17284.

Credit Fusion and Will Cukierski. 2011. Give me some credit. https://kaggle.com/competitions/GiveMeSomeCredit. Kaggle.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. Tabllm: Few-shot classification of tabular data with large language models. *Preprint*, arXiv:2210.10723.

Kai Helli, David Schnurr, Noah Hollmann, Samuel Müller, and Frank Hutter. 2024. Drift-resilient tabpfn: In-context learning temporal distribution shifts on tabular data. *Preprint*, arXiv:2411.10634.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. 2023. Tabpfn: A transformer that solves small tabular classification problems in a second. *Preprint*, arXiv:2207.01848.

Mouxiao Huang and Yu Qiao. 2023. Uncertainty-estimation with normalized logits for out-of-distribution detection. *Preprint*, arXiv:2302.07608.

M. M. Faniqul Islam, Rahatara Ferdousi, Sadikur Rahman, and Humayra Yasmin Bushra. 2020. Likelihood prediction of diabetes at early stage using data mining techniques. In *Computer Vision and Machine Intelligence in Medical Image Analysis*, pages 113–125. Springer, Singapore.

Tokimasa Isomura, Ryotaro Shimizu, and Masayuki Goto. 2025. Llmovertab: Tabular data augmentation with language model-driven oversampling. *Expert Systems with Applications*, 264:125852.

Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. 1988. Heart Disease. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C52P4X.

Steinbrunn William Pfisterer Matthias Janosi, Andras and Robert Detrano. 1989. Heart Disease. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C52P4X.

Kaggle. 2011. Give Me Some Credit. Kaggle Competition.

Guoxin Li, Chang Li, Chun Wang, and Zhenyu Wang. 2024. Suboptimal capability of individual machine learning algorithms in modeling small-scale imbalanced clinical data of local hospital. *PLOS ONE*, 19(2):e0298328.

Tianyang Liu, Fei Wang, and Muhao Chen. 2023. Rethinking tabular data understanding with large language models. *Preprint*, arXiv:2312.16702.

Hariharan Manikandan, Yiding Jiang, and J Zico Kolter. 2023. Language models are weak learners. *Preprint*, arXiv:2306.14101.

Sergio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31.

Anthony O'Hagan, Christopher E. Buck, Alireza Daneshkhah, J. Richard Eiser, Paul H. Garthwaite, David J. Jenkinson, Jeremy E. Oakley, and Tim Rakow. 2006. *Uncertain Judgements: Eliciting Experts' Probabilities*. John Wiley & Sons, Chichester, UK.

Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. 2017. Benchmark of deep learning models on large healthcare mimic datasets. *Preprint*, arXiv:1710.08531.

Bendi Ramana and N. Venkateswarlu. 2012. ILPD (Indian Liver Patient Dataset). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5D02C.

Bendi Ramana and N. Venkateswarlu. 2022. ILPD (Indian Liver Patient Dataset). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5D02C.

James Requeima, John Bronskill, Dami Choi, Richard E. Turner, and David Duvenaud. 2024. Llm processes: Numerical predictive distributions conditioned on natural language. *Preprint*, arXiv:2405.12856.

Alen Shapiro. 1983. Chess (King-Rook vs. King-Pawn). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5DK5C.

Aliaksandra Shysheya, John Bronskill, James Requeima, Shoaib Ahmed Siddiqui, Javier Gonzalez, David Duvenaud, and Richard E. Turner. 2025. Jolt: Joint probabilistic predictions on tabular data using llms. *Preprint*, arXiv:2502.11877.

Jack W Smith, John E Everhart, William C Dickson, William C Knowler, and Richard S Johannes. 1988. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 261–265. American Medical Informatics Association.

I-Cheng Yeh. 2008a. Blood Transfusion Service Center. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5GS39.

I-Cheng Yeh. 2008b. Blood Transfusion Service Center. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5GS39.

Erica Zhang, Ryunosuke Goto, Naomi Sagan, Jurik Mutter, Nick Phillips, Ash Alizadeh, Kangwook Lee, Jose Blanchet, Mert Pilanci, and Robert Tibshirani. 2025. Llm-lasso: A robust framework for domain-informed feature selection and regularization. *Preprint*, arXiv:2502.10648.

Zhiqiang Zhong, Kuangyu Zhou, and Davide Mottin. 2024. Harnessing large language models as post-hoc correctors. *Preprint*, arXiv:2402.13414.

A Appendix

A.1 Reproducibility

Code Availability To ensure reproducibility, we will release our full codebase, including detailed instructions for running the experiments and reproducing all reported results. We also provide the the posterior files generated by our experiments, as well as the scripts used to generate all figures and tables in the paper.

Data Availability All datasets used in this study are publicly available through open-access repositories. Further details and references are provided in Section A.4. All preprocessing steps required to obtain the final train/test splits used in our experiments, along with the preprocessed data, are available in our code release.

Results Variability To assess the stability and determinism of our approach, all experiments were repeated five times. The resulting 95% confidence intervals for AUC were effectively zero (0.000), indicating that our method yields highly consistent results across runs.

A.2 Hyperparameter Optimization

We used three datasets—*Blood Donation*, *Heart Disease*, and *Diabetes*—as development datasets to tune hyperparameters based on AUC score.

A.2.1 Number of Prompt Templates

We tested our method with 1–20 prompt templates analyzed how the number of templates affects LoID's performance. The following are some examples of our prompt templates:

1. The impact of on is . . .
2. The relationship between and is . . .
3. The role of in is . . .
4. When considering , the effect on is . . .
5. The correlation between and is . . .

As shown in Figure 2, performance gains plateau after approximately 10 sentences. Around this point, performance also becomes more stable, while incorporating additional sentences incurs increased computational cost. Accordingly, we use 10 sentences in all experiments.

A.2.2 Model Selection

We compared the performances of LLMs–GPT-2, LLaMA 3.1 8B, Gemma-3-27B, and Gemma-2-27B in terms of AUC. The best averge performance was achieved with Gemma-2-27B, as shown in Table 3.

dataset model	blood	diabetes	heart	average
gemma-3	0.65	0.84	0.89	0.79
gemma-2	0.67	0.84	0.90	0.80
llama-3-8b	0.56	0.80	0.85	0.73
gpt-2	0.53	0.82	0.84	0.71

Table 3: Model comparison across datasets using $\alpha = 0.2$, $\gamma = 2.0$, and 10 sentence formulations.

A.2.3 Variance Estimation Formula

The prior standard deviation σ_j for feature j quantifies the uncertainty in the elicited prior mean. We explored two methods for computing σ_j from LLM token probabilities, each capturing different aspects of elicitation uncertainty.

Method 1: Variance of Logit Differences This method uses the variance across multiple sentence formulations as a measure of prior uncertainty:

$$\sigma_j^2 = \alpha + \gamma \cdot \text{std}(logit_1(p_j), \ldots, logit_{|T|}(p_j))$$

where

$$\text{logit}(p_j) = \log\left(\frac{p_j}{1 - p_j}\right)$$

and

$$p_j = \frac{P_j^+}{P_j^+ + P_j^-}$$

Rationale: Higher variance across sentence formulations indicates inconsistent LLM responses, suggesting greater epistemic uncertainty. The hyperparameters α and γ control the base uncertainty level and sensitivity to variance, respectively.

Method 2: Entropy-Based Uncertainty This method uses Shannon entropy to measure the uncertainty in the LLM's token probability distribution:

$$\sigma_j = \beta \cdot \frac{1}{N_{\text{sent}}} \sum_{i=1}^{N_{\text{sent}}} H(p_{\text{pos}}^{(i)}, p_{\text{neg}}^{(i)})$$

Rationale: Higher entropy (closer to $\log 2 \approx 0.693$) indicates the LLM is uncertain between

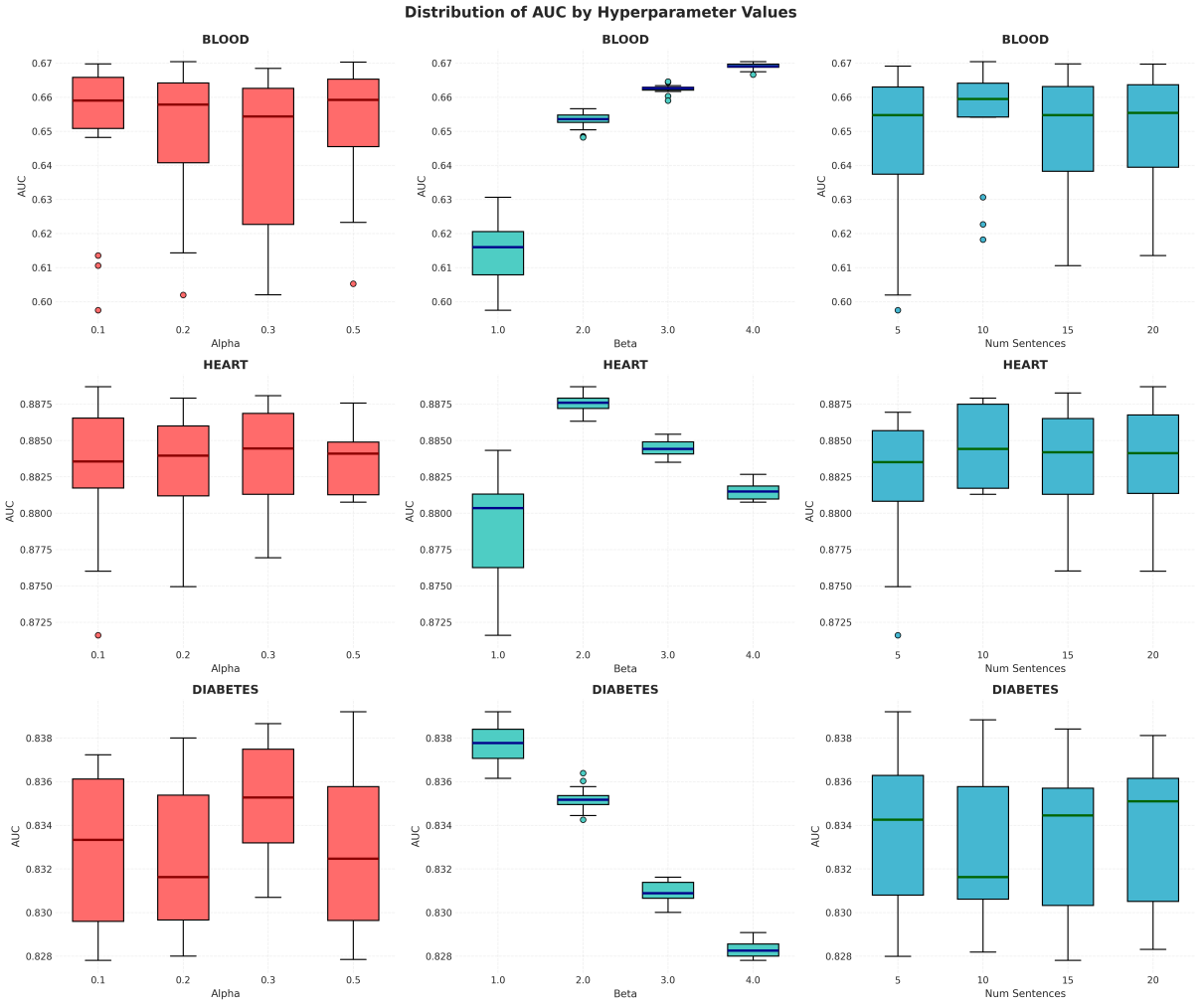

Figure 2: Hyperparameter ablation results demonstrating that prior uncertainty (beta) is the critical factor for performance, with F-statistics of 346.7 (blood), 47.3 (heart), and 583.2 (diabetes), while variance sensitivity (alpha) and number of sentence formulations (num_sentences) show no significant effect.

"positive" and "negative" tokens, reflecting ambiguity about the feature's effect direction. Lower entropy indicates confident predictions. The scaling factor β maps entropy to prior standard deviation.

Results Table 4 compares the variance estimation methods described above. While the overall differences in performance are small, the variance of logit differences consistently performs slightly better across all three datasets. Accordingly, we adopt this method for LoID.

Dataset	variance	Entropy	Δ
Blood	0.67	0.66	+0.01
Heart	0.90	0.89	+0.01
Diabetes	0.84	0.83	+0.01
Average	**0.80**	**0.79**	**+0.01**

Table 4: Comparison of variance of logit differences (used by LoID) vs. entropy-based uncertainty for uncertainty estimation.

A.2.4 Tuning α, γ, and N_{sent}

As detailed in Section 3.3, we use the following formula to calculate standard deviation:

$$\sigma_j^2 = \alpha + \gamma \cdot \text{std}(logit_1(p_j), \ldots, logit_{|T|}(p_j))$$

We searched over $\alpha \in \{0.1, 0.2, 0.3, 0.5\}$, $\gamma \in \{1, 2, 3, 4\}$, and $N_{\text{sent}} \in \{5, 10, 15, 20\}$. Table 5 reports the hyperparameter settings that achieve the best performance for each evaluation dataset. As shown in the table, no single hyperparameter configuration generalizes across all datasets; instead, each dataset requires task-specific tuning to achieve optimal performance. Among the evaluated α–γ combinations, $\alpha = 0.2$, $\gamma = 2.0$, and

$N_{\text{sent}} = 10$ yield the best average performance. For simplicity, we adopt these values for all remaining experiments.

Dataset	Best AUC	α	β	N_{sent}
Blood	0.6704	0.2	4.0	10
Heart	0.8987	0.1	2.0	20
Diabetes	0.8392	0.5	1.0	5
Overall (avg)	**0.8027**	**0.2**	**2.0**	**10**

Table 5: Best hyperparameter configurations identified for each dataset in the ablation study.

Figure 2 illustrates the relationship between AUC and the hyperparameters α, γ, and the number of sentences. Among these, γ has the most pronounced effect on performance, with its influence on AUC appearing approximately linear. For the Blood Donation dataset, AUC increases as γ increases, whereas for the Heart Disease and Diabetes datasets, AUC decreases with larger values of γ. These trends indicate that prior certainty plays a pivotal role in model performance. In contrast, we observe no consistent relationship between α, the number of sentences, and AUC across datasets, suggesting that these factors have a limited impact on performance.

Consistent with the previous findings, analysis of variance (ANOVA) reveals that γ is the only statistically significant predictor of out-of-distribution performance across all datasets (Table 6). The baseline uncertainty parameter γ exhibits strong effects ($p < 0.0001$). In contrast, both α and $_{\text{sent}}$ show no significant impact ($p > 0.05$) on any dataset.

Dataset	Parameter	F-statistic	p-value
Blood	α	0.231	0.8743
	γ	346.743	< 0.0001
	N_{sent}	0.436	0.7280
Heart	α	0.360	0.7820
	γ	47.269	< 0.0001
	N_{sent}	1.174	0.3285
Diabetes	α	1.162	0.3331
	γ	583.151	< 0.0001
	N_{sent}	0.083	0.9691

Table 6: ANOVA results testing hyperparameter effects on out-of-distribution AUC. Only γ (baseline prior uncertainty) shows significant effects across all datasets.

A.3 Additional Experiments

A.3.1 Feature Synonyms

We evaluated the impact of feature synonyms on model performance to determine whether changing feature names could introduce bias based on word selection. To do this, we created three sets of synonyms for the features in our evaluation datasets. Table 7 presents examples of these features and their corresponding synonyms.

Results in Table 8 reveal minimal performance variation across three distinct feature naming schemes, with AUC ranges of 0.007, 0.002, and 0.001 for blood, heart, and diabetes datasets respectively. These negligible differences demonstrate that our elicitation method is robust to superficial linguistic variations in feature descriptions. This stability is significant because it indicates the LLM is capturing genuine semantic understanding of the features' predictive relationships rather than relying on brittle pattern matching or memorized associations with specific phrasings.

A.3.2 Feature Description

We analyzed the impact of adding further description to the features and the target could help the LLM arrive at better estimates of the prior distributions.

Table 9 reveals minimal and inconsistent impact from including feature names alongside descriptions in prompts, with a mean absolute AUC difference of only 0.0084 (0.84%) across all 10 datasets. Performance changes are bidirectional, with 5 datasets showing marginal improvements (range: +0.002 to +0.038) and 5 showing marginal declines (range: -0.003 to -0.016) when feature descriptions are included. This 50/50 split and small standard deviation (0.0135) indicate no systematic benefit or detriment to either prompt format, suggesting the LLM extracts equivalent semantic information regardless of whether features are referenced by name-description pairs or names alone. The method's robustness to this formatting choice demonstrates that its effectiveness stems from semantic understanding rather than sensitivity to linguistic structure.

A.4 Datasets and OOD Splits

A.4.1 Datasets

We evaluate our prior elicitation method on ten binary, and multi-class classification datasets:

Dataset	Original Feature	Synonym Set 1	Synonym Set 2
Blood	recency	Time since last donation	Days elapsed since previous donation
	frequency	Number of donations made	Donation count
Heart	Age	Patient age	Years of life
	Cholesterol	Serum cholesterol level	Blood cholesterol concentration
	MaxHR	Peak heart rate	Highest heart rate reached
	ExerciseAngina	Angina during exercise	Exercise-provoked chest pain
Diabetes	Glucose	Blood glucose level	Plasma glucose concentration
	BMI	Body mass index	Weight-to-height ratio
	DiabetesPedigree	Family diabetes history score	Genetic diabetes risk

Table 7: Representative examples of feature name variations tested in synonym ablation study (Synonym Set 0 = original feature names).

Dataset	Syn0	Syn1	Syn2	Range
Bank	0.692	0.692	0.696	0.004
Blood	0.657	0.654	0.655	0.004
Credit Score	0.755	0.755	0.755	0.000
Diabetes	0.836	0.835	0.835	0.000
Heart	0.887	0.887	0.887	0.000
Income	0.844	0.842	0.844	0.002
Liver	0.667	0.666	0.667	0.001
Occupancy	0.988	0.988	0.988	0.001
Jungle	0.79	0.778	0.778	0.012
Pima	0.836	0.836	0.837	0.000
Mean Range		0.0013		

Table 8: Synonym Ablation Study: Robustness to feature name variations across all datasets. Three synonym sets are tested.

Dataset	w/o Desc	w/ Desc	Δ
Bank	0.692	0.673	-0.019
Blood	0.652	0.650	-0.002
Credit Score	0.756	0.756	+0.000
Diabetes	0.836	0.836	+0.000
Heart	0.886	0.893	+0.007
Income	0.845	0.853	+0.008
Liver	0.668	0.663	-0.005
Occupancy	0.989	0.986	-0.002
Jungle	0.778	0.777	-0.000
Pima	0.837	0.835	-0.002
Mean $\lvert\Delta\rvert$		0.0050	

Table 9: Feature Description Ablation Study: Impact of including feature descriptions in prompts across all datasets. Comparing name-only prompts vs. name+description format.

1. **Bank Marketing** (Moro et al., 2014): Predicts whether a client will subscribe to a term deposit based on direct marketing campaign data from a Portuguese banking institution. Features include client demographics (age, job, marital status, education), campaign information (contact type, number of contacts, previous campaign outcome), and economic indicators (employment variation rate, consumer price index). The binary classification task identifies clients likely to subscribe to term deposits.

2. **Blood Transfusion** (Yeh, 2008b): Predicts whether a blood donor will donate within a given time window using data from the Blood Transfusion Service Center in Hsin-Chu City, Taiwan. Features are based on the RFMTC marketing model: Recency (months since last donation), Frequency (total number of donations), Monetary (total blood donated in c.c.), and Time (months since first donation). The binary classification task identifies donors likely to donate in March 2007.

3. **Give Me Some Credit** (Kaggle, 2011): Predicts the probability of financial distress in the next two years using credit and demographic data from Kaggle. Features include checking account status, credit duration, credit history, credit amount, savings account balance, employment status, and age. The binary classification task evaluates financial risk for 1,000 borrowers.

4. **Diabetes** (Islam et al., 2020): Predicts the

likelihood of diabetes based on self-reported symptoms and demographic information. Features include age, gender, polyuria, polydipsia, sudden weight loss, weakness, polyphagia, genital thrush, visual blurring, itching, irritability, delayed healing, partial paresis, muscle stiffness, alopecia, and obesity. The dataset contains 520 instances collected via questionnaires and is formulated as a binary classification task.

5. **Heart Disease** (Janosi et al., 1988): Predicts heart disease using clinical and demographic attributes such as age, sex, resting blood pressure, cholesterol level, fasting blood sugar, maximum heart rate, exercise-induced angina, ST depression (oldpeak), chest pain type, resting ECG results, and ST slope. The binary classification task identifies individuals with high likelihood of heart disease.

6. **Adult Income** (Becker and Kohavi, 1996a): Predicts whether an individual's annual income exceeds $50K based on 1994 U.S. Census data. Features include age, education level, marital status, occupation, relationship status, workclass, sex, native country, capital gain/loss, and hours worked per week. The binary classification task uses demographic and employment data from 48,842 records.

7. **Chess Endgame** (Shapiro, 1983): Predicts game outcomes in King-Rook vs. King-Pawn chess endgame positions. Features encode piece positions using strength rankings, file coordinates, and rank coordinates for white and black pieces. The binary classification task determines winning positions in this chess endgame with 44,820 board configurations.

8. **Indian Liver Patient** (Ramana and Venkateswarlu, 2012): Predicts whether a patient has liver disease using clinical measurements collected from northeast Andhra Pradesh, India. Features include age, gender, total bilirubin, direct bilirubin, alkaline phosphatase, alanine aminotransferase (SGPT), aspartate aminotransferase (SGOT), total proteins, albumin, and albumin/globulin ratio. The binary classification task distinguishes liver patients from healthy individuals using data from 583 records (416 liver patients, 167 non-liver patients).

9. **Occupancy Detection** (Candanedo, 2016b): Predicts room occupancy status from environmental sensor measurements in an office setting. Features include temperature (Celsius), relative humidity (%), light (Lux), CO2 concentration (ppm), humidity ratio (kg water-vapor/kg-air), and timestamp information. The binary classification task detects occupancy using measurements recorded at one-minute intervals.

10. **Pima Indians Diabetes** (Smith et al., 1988): Predicts the onset of diabetes within five years in females of Pima Indian heritage over age 21. Features include number of pregnancies, glucose level, blood pressure, skin thickness, insulin level, BMI, diabetes pedigree function, and age. The binary classification task uses medical measurements from 768 women from a population near Phoenix, Arizona with high diabetes incidence rates.

A.4.2 OOD Splits

We construct out-of-distribution (OOD) train splits to simulate a training set that does not represent the entire population. For each dataset, we consider numerical features with at least two unique values and apply five shift strategies to induce distributional gaps between training and test data:

1. *extreme_10*, training on the bottom 10%
2. *extreme_5_95*, training on the bottom 5%
3. *moderate_20_80*, training on the bottom 20%;
4. *tail_0_50*, training on the 0–25% range;
5. *tail_50_100*, training on the 50–75% range.

For each split, we designate a different feature as the shift variable (cycling through available features), compute the corresponding quantile thresholds, and construct train–test masks while enforcing a minimum of 50 samples per split. Models are trained on the OOD training set and evaluated on the *entire* dataset to quantify performance degradation under distribution shift. Splits with insufficient class diversity (fewer than two classes in either the training or test set) are discarded. This procedure simulates realistic deployment settings in which models must generalize to regions of the feature space not observed during training.

Dataset	Method	Mean Mag.	Std (Uncertainty)	Std Ratio	Performance
Liver Disease	LoID	0.065	0.430	7.4×	**Win**
	AutoElicit	0.051	0.058	1×	
Income	LoID	0.054	0.426	3.4×	**Win**
	AutoElicit	0.015	0.124	1×	
Bank Marketing	LoID	0.107	0.428	0.3×	**Win**
	AutoElicit	0.017	1.379	1×	
Credit Score	LoID	0.070	0.425	9.0×	**Win**
	AutoElicit	0.008	0.047	1×	
Room Occupancy	LoID	0.071	0.427	4.9×	**Win**
	AutoElicit	0.015	0.087	1×	
Pima DIabetes	LoID	0.077	0.424	9.6×	**Win**
	AutoElicit	0.070	0.044	1×	
Diabetes	LoID	0.213	0.866	14.8×	**Win**
	AutoElicit	0.085	0.059	1×	
Jungle Chess	LoID	0.064	0.416	3.4×	**Win**
	AutoElicit	0.007	0.122	1×	
Heart Disease	LoID	0.296	0.100	0.9×	**Loss**
	AutoElicit	0.025	0.114	1×	
Blood Donation	LoID	0.192	0.863	35.9×	**Loss**
	AutoElicit	0.041	0.024	1×	

Table 10: Comparison of LoID and AutoElicit across datasets in terms of magnitude, uncertainty, and performance.

For *Heart Disease*, we use the cholesterol split (*moderate_20_80*), training on the 20–80% quantile range and testing on extreme values, resulting in an AUC drop from 0.933 to 0.870 ($\Delta = 0.063$). *Liver Disease* uses the alkaline phosphatase split (*tail_0_50*), training on the bottom 50% and testing on the top 50%, with degradation from 0.768 to 0.698 ($\Delta = 0.070$). *Blood Donation* employs the recency split (*extreme_10_90*), training on donors in the bottom 10% and testing on the top 10%, experiencing the largest degradation from 0.755 to 0.579 ($\Delta = 0.176$). *Occupancy Detection* uses the light intensity split (*moderate_20_80*), although this split exhibits insufficient class diversity for full evaluation. *Diabetes* and *Pima* both use the pregnancies split (*extreme_10_90*), training on the bottom 10% and testing on the top 10%, with degradation from 0.839 to 0.813 ($\Delta = 0.027$). *CreditG* partitions on credit amount (*extreme_5_95*), training on the bottom 5% and testing on the top 5%, yielding a drop from 0.786 to 0.710 ($\Delta = 0.076$). *Income* uses the education level split (*extreme_5_95*), with the steepest decline from 0.908 to 0.781 ($\Delta = 0.127$). *Bank Marketing* splits on call duration (*extreme_10_90*), experiencing severe degradation from 0.888 to 0.508 ($\Delta = 0.380$), representing the most extreme covariate shift in our evaluation. Finally, *Jungle Chess* uses the black piece strength split (*tail_0_50*), training on the 0–50% range and testing on 50–100%, with degradation from 0.810 to 0.712 ($\Delta = 0.098$). These splits simulate realistic deployment scenarios in which models trained on one subpopulation must generalize to substantially different data distributions.

A.5 LoID vs Existing Baselines

A.5.1 Prior Performance

Since LoID and AutoElicit employ similar prior elicitation strategies and exhibit more comparable performance than other baselines, Table 10 focuses on analyzing where and why each method performs better. The underlying intuition behind these differences is discussed in detail in Section 5.1.

Dataset	OOD-LR	Prior= $\mathcal{N}(0,1)$	LoID	Prior= $\mathcal{N}(0,0.45)$	Cap	Δ Gap Closed%
Bank	0.51	0.61	0.72	0.61	0.89	+30.0
Blood	0.58	0.63	0.67	0.62	0.75	+31.3
Credit Score	0.71	0.74	0.76	0.76	0.79	+0.0
Diabetes	0.81	0.82	0.84	0.81	0.84	+104.5
Heart	0.87	0.91	0.90	0.92	0.93	-31.8
Income	0.78	0.81	0.85	0.83	0.91	+14.9
Liver	0.70	0.68	0.69	0.70	0.77	-9.9
Occupancy	0.98	0.72	1.00	0.75	1.00	+1155.7
Jungle	0.71	0.72	0.79	0.75	0.81	+40.0
Pima	0.82	0.81	0.84	0.81	0.84	+98.1
Average	**0.74**	**0.75**	**0.80**	**0.76**	**0.86**	**+143.3**

Table 11: Comparison of LoID priors and priors with 0 means and stds in the Goldilock zone on OOD datasets. Δ **Gap Closed shows the percentage point difference in gap closed between LoID and Prior =** $\mathcal{N} = (0, 0.45)$. OOD-LR is logistic regression trained on OOD data (lower bound), Cap is logistic regression trained on full IID data (upper bound).

A.5.2 Computational Efficiency

We compare the computational complexity and runtime of LoID with AutoElicit and LLMProcesses. For a dataset with d features, the LoID method queries the LLM using N_{sent} sentence formulations per feature and extracts the probabilities of two tokens (*positive* and *negative*). Its time complexity is:

$$T_{\text{logit}} = \mathcal{O}(d \times N_{\text{sent}} \times T_{\text{LLM}}),$$

where T_{LLM} denotes the inference time of a single LLM query. In our experiments, $N_{\text{sent}} = 10$, resulting in approximately 20 queries per feature.

AutoElicit employs an interactive multi-round elicitation process with iterative refinement, leading to a higher complexity:

$$T_{\text{AE}} = \mathcal{O}(d \times N_{\text{rounds}} \times N_{\text{queries}} \times T_{\text{LLM}}),$$

where $N_{\text{rounds}} \in [3, 5]$ and $N_{\text{queries}} \in [5, 10]$. This results in more than 28 LLM queries per feature on average.

Empirically, this yields a consistent **1.4–1.5**× **speedup** in prior elicitation for LoID across all evaluated datasets.

LLMProcesses follows a fundamentally different paradigm: it generates predictions for each test instance by sampling from the LLM N_{samples} times using in-context examples. For a test set of size n_{test} with n_{train} training examples in context, its complexity is:

$$T_{\text{LLMP}} = \mathcal{O}(n_{\text{test}} \times N_{\text{samples}} \times T_{\text{LLM}}(n_{\text{train}} \cdot d)),$$

where $T_{\text{LLM}}(n_{\text{train}} \cdot d)$ reflects the increased inference time caused by long prompts containing n_{train} in-context examples. Unlike LoID and AutoElicit, which perform prior elicitation once per dataset, LLMProcesses scales linearly with the test set size and incurs substantially longer runtimes (**15–20 hours** for datasets with 100–400 test instances, compared to **minutes** for prior elicitation methods).

A.6 Prior Impact on Model Performance

A.6.1 LoID Priors vs. $\mathcal{N}$(0, 0.45) vs. $\mathcal{N}$(0, 1)

To isolate the contribution of the prior means inferred by LoID, we evaluate a baseline using normal priors with zero mean and standard deviation 0.45 (the Goldilock setting). This comparison quantifies how much LoID's learned prior means contribute to downstream AUC. As shown in Table 11, incorporating LoID-derived means yields a substantial performance gain, increasing the gap closed by approximately 154% on average. This difference is most prominent in datasets where the uninformed priors catastrophically fail: Occupancy (+1156%), Diabetes (+105%), and Pima (+98%). In these cases, the $\mathcal{N}(0, 0.45)$ prior provides insufficient directional guidance, causing the Bayesian posterior to converge toward coefficients near zero and perform worse than the OOD baseline. Conversely, LoID's informative means—despite systematic positive bias—capture feature importance magnitudes that guide the model toward the cor-

rect solution space. Even on datasets where LoID underperforms (Heart: -32%, Liver: -10%), the performance gap remains small compared to the massive gains elsewhere. This demonstrates that **prior mean quality matters substantially more than perfect directional correctness** when paired with appropriately calibrated uncertainty ($\sigma \approx 0.4$–0.5).

Finally, we include an additional $\mathcal{N}(0, 1)$ baseline to illustrate the improvement achieved over a more weakly informative prior that is commonly used with Bayesian inference.

A.6.2 $\mathcal{N}$(0, 1) vs. $\mathcal{U}$(-1, 1)

To understand the impact of prior distribution choice, we compare two configurations: (1) uninformative Normal priors $\mathcal{N}(0, 1)$, and (2) uninformative Uniform priors $\mathcal{U}(-1, 1)$.

Table 12 shows the catastrophic failure of Uniform priors (average AUC 0.42, with Occupancy at 0.02) and reveals that while Uniform priors ostensibly provide "maximum uninformativeness" by assigning equal probability across a range, they fundamentally fail as regularizers for regression coefficients because:

1. **Lack of soft regularization:** Uniform distributions do not encode the reasonable prior belief that smaller coefficient magnitudes are more plausible than larger ones. This absence of gradient-based regularization toward zero can lead to poor MCMC sampling behavior and overfitting.

2. **Incompatibility with MCMC:** The flat density of Uniform priors provides no gradient information to guide sampling algorithms, resulting in inefficient exploration and convergence issues, particularly visible in the Occupancy dataset collapse.

3. **Prior-likelihood mismatch:** For logistic regression, coefficients naturally follow a scale where values near zero are common and extreme values are rare. Uniform priors violate this structure, treating $\beta = 0.1$ and $\beta = 4.9$ as equally plausible a prior.

Dataset	$\mathcal{N}(0, 1)$	$\mathcal{U}(-1, 1)$
Bank	0.61	0.35
Blood	0.63	0.59
Credit Score	0.74	0.58
Diabetes	0.82	0.32
Heart	0.91	0.84
Income	0.81	0.30
Liver	0.68	0.70
Occupancy	0.72	0.02
Jungle	0.72	0.58
Pima	0.82	0.32
Average	**0.75**	**0.46**

Table 12: Comparison of LoID priors against uninformative Normal and Uniform priors on OOD datasets.

A.7 Computational Resources

1. **Infrastructure:** Experiments were run on a workstation equipped with **2 NVIDIA RTX 6000 Ada Generation GPUs** (each with 50GB VRAM), a **Threadripper PRO 7985WX CPU** with **64 cores (128 threads)** at up to **5.37 GHz**, and **256 GB of RAM**.

2. **Model Inference:** We used **Gemma-2-27B** as our main LLM. All inference was performed using both GPUs in parallel. Each feature prompt required about **50–70 seconds** to complete across 10 paraphrases, leading to approximately **60 seconds * #features** per dataset.

3. **Preprocessing:** Minimal preprocessing was required and was handled efficiently on a single CPU.

4. **Additional Resources:** Exploratory analyses, including feature selection, were performed using **GPT-4o via OpenRouter**. Each call took **2–3 seconds** and used one CPU thread.