Exploiting LLMs for Automatic Hypothesis Assessment via a Logit-Based Calibrated Prior

Yue Gong*
Amazon Web Services
yuegongy@amazon.com

Raul Castro Fernandez
The University of Chicago
raulcf@uchicago.edu

Abstract

As hypothesis generation becomes increasingly automated, a new bottleneck has emerged: hypothesis assessment. Modern systems can surface thousands of statistical relationships–correlations, trends, causal links–but offer little guidance on which ones are novel, non-trivial, or worthy of expert attention. In this work, we study the complementary problem to hypothesis generation: *automatic hypothesis assessment*. Specifically, we ask–given a large set of statistical relationships, can we automatically assess which ones are novel and worth further exploration? We focus on correlations as they are a common entry point in exploratory data analysis that often serve as the basis for forming deeper scientific or causal hypotheses.

To support automatic assessment, we propose to leverage the vast knowledge encoded in LLMs' weights to derive a prior distribution over the correlation value of a variable pair. If an LLM's prior expects the correlation value observed, then such correlation is not surprising, and vice versa. We propose the *Logit-based Calibrated Prior*, an LLM-elicited correlation prior that transforms the model's raw output logits into a calibrated, continuous predictive distribution over correlation values. We evaluate the prior on a benchmark of 2,096 real-world variable pairs and it achieves a sign accuracy of 78.8%, a mean absolute error of 0.26, and 95% credible interval coverage of 89.2% in predicting Pearson correlation coefficient. It also outperforms a fine-tuned RoBERTa classifier in binary correlation prediction and achieves higher precision@K in hypothesis ranking. We further show that the prior generalizes to correlations not seen during LLM pretraining, reflecting context-sensitive reasoning rather than memorization.

1 Introduction

Generating hypotheses from large data repositories is quickly becoming easier. Modern data discovery systems [2, 8, 25, 21] can enumerate every statistical relationship across datasets, and LLMs can draft thousands of plausible ideas by mining literature and data [33, 31, 29]. What used to take a researcher weeks now happens in minutes. This ease of generation, however, introduces a new bottleneck: assessment. Experts are flooded with machine-suggested relationships–correlations, causal links, trends, anomalies–without a clear signal for which ones merit deeper investigation. Many of these relationships are trivial, redundant, or already well known, forcing human experts to sift through a long list just to find a few that are novel.

For example, as illustrated in Figure 1, a correlation discovery system has surfaced tens of thousands of correlated variable pairs, leaving human experts to manually filter out trivial or expected patterns using their prior knowledge. A strong correlation between `daily temperature` and `ice cream sales`, for instance, is intuitive and quickly dismissed. In contrast, a negative correlation between `household`

*Work done at the University of Chicago

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

`income` and `housing prices` might appear counterintuitive and warrant further scrutiny. This manual triage must be repeated across thousands of pairs to uncover truly novel or surprising correlations, making the process highly labor-intensive. One might hope that ranking correlations by magnitude could alleviate this burden. However, as shown in Figure 1 (left panel where variable pairs are ranked by $|r_{\text{obs}}|$), stronger correlations are not necessarily more surprising—in fact, they often reflect well-known or redundant relationships, a trend we further verify in Section 5.

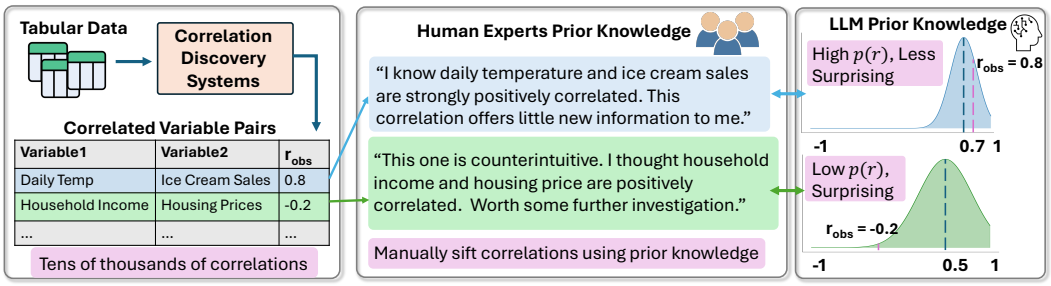

Figure 1: How Human Experts Assess Correlations Manually and How an LLM Can Help

In this paper, we study the complementary problem to hypothesis generation: *automated hypothesis assessment*. Specifically, we ask–given a large set of statistical relationships, can we automatically assess which ones are novel and worth further exploration? We focus on correlation relationships as a starting point, since they are a common entry point in exploratory data analysis and often serve as seeds for forming deeper scientific or causal hypotheses [11, 4].

To tackle this problem, we draw inspiration from how experts reason: they use prior knowledge to form expectations about a correlation's direction and magnitude. If the observed correlation (r_{obs}) matches expectations, it is unsurprising; if it deviates, it may signal something worth exploring. In essence, experts apply an implicit *prior* shaped by their knowledge and the variable context.

Our core idea is to approximate this human prior using the rich, encoded knowledge within LLMs [27, 18]. Specifically, we define the *LLM-elicited correlation prior*, $p_{\text{LM}}(r_{X,Y} \mid \mathcal{C}_{X,Y})$, as a predictive distribution over correlation values $r_{X,Y}$ between a variable pair X, Y conditioned on their context $\mathcal{C}_{X,Y}$, such as the description for each variable. By prompting the LLM with this context, we elicit its belief about the correlation values, treating these beliefs as a proxy for human expectations.

The LLM-elicited correlation prior helps identify which correlations are novel and worth expert attention. For instance, in Figure 1, $p_{\text{LM}}(r_{\text{Daily Temp, Ice Cream Sales}} \mid \mathcal{C})$ centers around 0.7, making an observed value of 0.8 unsurprising. In contrast, a correlation of -0.2 against a prior centered at 0.5 signals high surprise. This surprise-based scoring offers a scalable way to surface potentially insightful correlations. In our later evaluation (Section 5), we show that the LLM prior highlights expert-validated hypotheses from noisy urban data [19].

In this work, we propose *Logit-based Calibrated Prior*, an LLM-elicited correlation prior which transforms the LLM's raw output logits into a calibrated, continuous predictive distribution over correlation values (Section 2). But how do we evaluate its quality?

First, we assess accuracy: if the prior's mode reliably predicts the sign and magnitude of observed correlations, it suggests alignment with empirical patterns. Second, we evaluate information content. A strong prior should assign high likelihood to observed correlations, reducing their information content relative to an uninformative baseline (e.g., a uniform prior). When applied at scale, this indicates the prior captures real-world patterns, easing the burden on analysts. Third, we measure calibration–whether the prior's uncertainty reflects reality–using 95% credible interval coverage. This is crucial for decision-making: overconfident priors exaggerate surprise and risk misdirecting expert attention. Finally, we ask a deeper question: is the prior reasoning from context, or merely recalling memorized correlations based on variable names? To probe this, we introduce a novel evaluation based on *contextual contradiction* to disentangle these possibilities.

To support these goals, we construct a benchmark of 2,096 variable pairs with observed correlations. We evaluate predictive quality and information reduction (Section 4), hypothesis discovery utility (Section 5), and whether the prior reflects contextual reasoning or memorization (Section 6).

Results are promising: our *Logit-based Calibrated Prior* achieves 78.8% sign accuracy and a mean absolute error of 0.26 on Pearson correlation coefficients in the range $[-1, 1]$, with strong calibration–95% intervals covering 89.2% of observed values. It also reduces the average information content from 0.69 (uniform prior) to 0.27. Our method outperforms baselines, including uninformative priors, Gaussian priors from LLM-verbalized parameters, and a fine-tuned RoBERTa classifier [30]. It also achieves higher precision@K when retrieving meaningful correlations in noisy urban data. For instance, it highlights a link between bike dock density and community wealth, a hypothesis studied in prior work [7], while down-ranking obvious patterns like library visitors and book circulation. Finally, we show that the prior generalizes beyond correlations seen during pretraining.

These results show that LLMs encode informative prior beliefs about statistical relationships, demonstrating their potential to serve as proxies for hypothesis assessment–a task that currently relies on human expertise and is highly labor-intensive. Our work highlights a promising direction for leveraging LLMs to help experts navigate large hypothesis spaces and make novel discoveries.

2 Logit-based Calibrated Prior (LCP): Constructing a Continuous Correlation Prior from LLM Logits

In this section, we present the *Logit-based Calibrated Prior (LCP)*, a method for constructing the correlation prior, $p_{\text{LM}}(r_{X,Y} \mid \mathcal{C}_{X,Y})$ –a predictive distribution over correlation values $r_{X,Y}$ between a variable pair X, Y conditioned on their context $\mathcal{C}_{X,Y}$, such as the description for each variable.

One way to elicit a distribution from an LLM is to have it parameterize a fixed form–e.g., modeling its belief over a correlation as a Gaussian by providing a mean and standard deviation. However, this approach relies on the assumption that the model's internal belief distribution conforms to the chosen parametric form, which is not the case in most cases. To test this, we conducted a chi-square goodness-of-fit analysis [28] on the LLM's output distributions for 2,096 correlations. The normality assumption was rejected in 2,095 cases at the 5% significance level, indicating that the LLM's beliefs are poorly approximated by a Gaussian fit (see Appendix A for details). Moreover, this parametric approach introduces additional complexity: While the original goal is to estimate a single correlation value, it requires estimating additional parameters whose values are themselves subject to error.

Our approach. Rather than asking the LLM to estimate parameters of a fixed distributional form, we directly prompt it to predict the correlation coefficient (see prompt in Appendix H.1), and construct a full distribution over possible correlation values. While one could obtain this distribution by sampling from the model, it would be computationally expensive. To address this, we propose a more efficient strategy by constructing the prior directly from the LLM's logits. This approach does not assume the distribution's shape and allows the model to focus on estimating the correlation between the variables.

We begin by constructing a discrete probability distribution from the model's raw token logits (Algorithm 1). Without loss of generality, we assume r denotes Pearson's correlation coefficient, constrained to the range $[-1, 1]$. At each decoding step t, the language model produces a real-valued logit vector ℓ_t, where each entry $\ell_t^{(i)}$ corresponds to a token in the vocabulary. These logits are converted into log-probabilities via the softmax function. For a selected token v_t at position t, its log-probability is given by $\log p_t^{(v_t)} = \ell_t^{(v_t)} - \log \sum_j \exp(\ell_t^{(j)})$.

We design a prompt that elicits a structured scalar response, such as *{"coefficient": "<value>"}*. To extract the correlation value, we first identify the start and end positions of `"<value>"` in the output sequence (line 4). Starting from this token position, we extract the top-k tokens at each subsequent decoding step (line 5). A complete numeric response, such as `"-0.69"`, is composed of a valid sequence of tokens–e.g., a sign, integer part, decimal point, and numeric suffix. For instance, at the numeric suffix token, the model might assign different probabilities to completions like 69, 60, or 70. To prevent length biases, we ensure that positive and negative numbers use the same number of tokens in their representation (see Appendix B).

We enumerate all token sequences (line 5), concatenate them into strings (line 6), cast them to float values (line 8), and compute their joint log-probabilities by summing the log-probabilities of each token in the sequence (line 12). We discard any sequences that produce invalid float values or values outside the valid correlation range $[-1, 1]$ (line 10). When multiple token sequences map to the same numeric value (e.g., `"0.65"` and `".65"`), we aggregate their unnormalized probabilities (line 13-17).

Finally, we normalize across all valid correlation values using the softmax function to obtain a discrete probability distribution, $\{(r_j, p_j)\}_{j=1}^{N}$ (line 19) where r_j is a decoded correlation value, and p_j is its model-assigned probability.

Impact of Single-Path Decoding. Our approach decodes one token at a time conditioned on the most likely token from the previous step. In particular, we first select the most probable sign token (e.g., + or -) and condition all subsequent decoding on that choice. This introduces an approximation: we compute the joint probability of each numeric string under a single sign path, rather than marginalizing over both. While a full multi-beam search would more faithfully capture the true joint distribution by exploring multiple branches at each step, we find that this approximation works well in practice. First, the model achieves *high sign accuracy*: as shown in Section 4, it predicts the correct sign 78.8% of the time, so most sequences are decoded under the correct branch. Second, it exhibits *high sign confidence*: across 2096 correlation predictions, the median probability gap between the two signs is 99.8% (77.7% on average), indicating that the probability mass of the alternative branch is negligible. Finally, our method ensures *scalability*: a full beam search would increase decoding cost exponentially with sequence length, whereas the single-path strategy scales efficiently to tens of thousands of variable pairs while maintaining strong empirical performance.

Smoothing to Obtain a Continuous Prior. The discrete distribution is sparse and limited to discrete values determined by the tokenizer and top-k decoding strategy. However, downstream tasks–such as computing surprise in Figure 1–require probability density at arbitrary values. To support this, we smooth the distribution using a weighted sum of Gaussian kernels centered at each decoded value. Since Pearson correlations lie in $[-1, 1]$, we truncate and renormalize the distribution to ensure it integrates to one. The final LCP density function $f(r)$ is defined as:

$$f(r) = \frac{1}{Z} \sum_{j=1}^{N} p_j \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r - r_j)^2}{2\sigma^2}\right), \quad r \in [-1, 1]$$

where σ is the standard deviation of each kernel and controls the degree of smoothing, and Z is the normalization constant.

Algorithm 1 ConstructDiscretePriorFromLogits

Input: Token logits $\{\ell_t\}_{t=1}^{T}$, structured output template $\mathcal{T}$ such as *{"coefficient": "<value>"}*
Output: Discrete prior $\{(r_j, p_j)\}_{j=1}^{N}$
Initialize empty map: logp_map $\leftarrow \emptyset$
$t_0, t_1 \leftarrow$ `FindValueTokenSpan`$(\{\ell_t\}_{t=1}^{T}, \mathcal{T})$ ▷ Locate start and end positions of the value field
for all sequences $s = (v_{t_0}, \ldots, v_{t_1})$ from top-k tokens at each position **do**
    `str` $\leftarrow$ `concat`(s)
    if `is_valid_float`(str) **and** `float`(str) $\in [-1, 1]$ **then**
        $r \leftarrow$ `float(str)`
    else
        continue
    end if
    $\log p_r \leftarrow \sum_{t=t_0}^{t_0+L} \log p_t^{(v_t)}$
    if $r \in$ logp_map **then**
        logp_map$[r] \leftarrow \log\left(\exp(\text{logp_map}[r]) + \exp(\log p_r)\right)$
    else
        logp_map$[r] \leftarrow \log p_r$
    end if
end for
$\{(r_j, p_j)\}_{j=1}^{N} \leftarrow$ softmax(logp_map) ▷ Normalize log-probs into a valid probability distribution
return $\{(r_j, p_j)\}_{j=1}^{N}$

Selecting an appropriate kernel standard deviation σ is critical to ensure the prior reflects realistic uncertainty. If σ is too small, the resulting distribution will be overconfident and overly spiky; if too large, it will be underconfident and overly diffuse. Standard bandwidth selection rules, such as Scott's [23] or Silverman's rule [26], are not applicable in our setting, as they assume i.i.d. samples from an underlying distribution. In our case, in contrast, the discrete values and their probabilities are derived from LLM output logits and reflect model-specific beliefs, not empirical frequencies.

To address this, we tune σ using a held-out validation set by minimizing the average negative log-likelihood at the observed correlation values:

$$\sigma^* = \arg\min_{\sigma} \ \mathbb{E}_{r_{\text{obs}} \sim \mathcal{D}_{\text{val}}} \left[-\log p_\sigma(r_{\text{obs}}) \right],$$

This objective penalizes priors that assign low probability density to ground-truth correlations, thereby encouraging distributions that place probability mass closer to the observed values. Optimizing σ this way calibrates uncertainty to reflect empirical variability and improves downstream reliability. The validation set $\mathcal{D}_{\text{val}}$ consists of 300 randomly sampled correlations, disjoint from our evaluation dataset. The optimized value $\sigma^* = 0.4$ is used for LCP.

The kernel standard deviation σ does not need to be re-tuned as long as four key elements remain unchanged: the LLM, the prompting strategy, the task (predicting Pearson correlation coefficients), and the kernel function used. This is because σ corrects for the systematic bias in the model's uncertainty–that is, whether the model tends to be consistently overconfident or underconfident in its predictions. When the model architecture, prompt design, task, and the kernel function remain fixed, this bias remains stable across inputs, even if individual predictions vary. In this setting, a single globally tuned σ is sufficient to calibrate the model's uncertainty across a broad range of variable pairs. We further demonstrate in our evaluation (Section 4, 5) that the selected σ generalizes well on the evaluation dataset, demonstrating its robustness. However, if any of these components change–such as switching to a different model, altering the prompt, targeting a different correlation metric, or switching to a different kernel function–the structure of the output distribution may shift, and σ should be re-tuned to ensure proper calibration.

In the evaluation, we compare the Logit-based Calibrated Prior against two baseline methods for constructing correlation priors, highlighting the advantages of avoiding parametric assumptions and applying proper calibration. The first is a Gaussian prior, which assumes the LLM can directly parameterize a normal distribution by predicting its mean and standard deviation. The second is an uncalibrated KDE prior, which is similar to our method, but selects the kernel standard deviation using Scott's rule based on the empirical standard deviation of the discrete probability distribution.

3 Benchmark Construction

We curate a benchmark of 2,096 real-world variable pairs to evaluate correlation priors. Each entry includes two variables, their descriptions, a dataset summary, and the observed Pearson correlation $r_{\text{obs}} \in [-1, 1]$, computed from raw data. The benchmark combines variable pairs from the Cause-Effect Pairs [15] and Kaggle [30] datasets. We have open-sourced our code and data at `https://github.com/TheDataStation/LLM-Prior-for-Correlation-Assessment`.

The Cause-Effect dataset contains 108 variable pairs with known causal relationships. We retain 96 pairs where the correlation is statistically significant ($p < 0.05$). The Kaggle dataset consists of correlations between variable pairs extracted from publicly available tables on Kaggle. The original dataset provides variable names but lacks variable descriptions. To enrich the context for variables, we use the Kaggle API to retrieve dataset summaries and employ GPT-4o (see Appendix H.5) to assess whether the variable names are self-descriptive[2]. We filter out non-informative names (e.g., single characters or generic identifiers like "Unnamed: 0") and retain only those pairs for which both variables are judged meaningful. This further cleaning allows us to isolate and study the model's ability to reason about relationships, rather than its ability to interpret metadata.

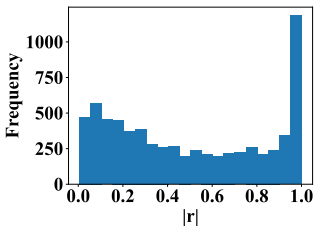

Figure 2: The Bias toward High Correlations in Kaggle dataset

After filtering, we obtain 7045 statistically significant correlations ($p < 0.05$). To mitigate the bias toward extreme correlations (see Fig. 2), we perform stratified sampling by $|r|$: divide the range $[-1, 1]$ into 10 equal-width bins and sample 200 correlations per bin, yielding a balanced set of 2,000.

[2]Kaggle API does not support retrieving variable descriptions

A balanced sample ensures fair evaluation across all correlation strengths, preventing the model's performance from being skewed by overrepresented low or high $|r|$ values.

4 How Well Does LCP Predict Empirical Correlations?

We evaluate LCP by measuring how well it predicts observed correlations. First, we assess *predictive accuracy* using two metrics: *sign accuracy*, the fraction where $\hat{r} \cdot r_{\text{obs}} > 0$, and *absolute error*, $|\hat{r} - r_{\text{obs}}|$, where $\hat{r}$ is the mode of the prior. We calculate the mode using grid sampling. Next, we evaluate differential information content by computing $-\log p(r_{\text{obs}})$, adapting Shannon's self-information [24] to the continuous case. For simplicity, we refer to it as *information content* hereafter. A good prior assigns high likelihood to observed values, reducing the information content of the corpus and easing analyst workload. Finally, we assess *calibration* by 95% credible interval coverage– the fraction of cases where r_{obs} falls within the prior's 95% credible interval. Calibration is critical: an overconfident prior may exaggerate surprise from small deviations, leading to false positives and misleading experts.

We compare LCP with the following baselines. All methods use GPT-4o (2024-08-06) [17] as the underlying model.

• **Uniform Prior:** A non-informative baseline with constant density 0.5 over $[-1, 1]$. The sign accuracy for it is measured by randomly guessing the sign of the correlation.

• **Gaussian Prior:** We adapt the method from Capstick et al. [1], which elicits Gaussian priors via LLM-prompted mean and standard deviation, to model a truncated Gaussian prior over correlations in $[-1, 1]$ (see Appendix H.2).

• **KDE Prior:** A kernel density estimation using Gaussian kernels, where the kernel standard deviation σ is set using a weighted version of Scott's rule: $\sigma = 1.06 \cdot \hat{\sigma} \cdot n_{\text{eff}}^{-1/5}$, where $\hat{\sigma}$ is the weighted standard deviation of $\{(r_j, p_j)\}$, and n_{eff} is the effective sample size.

Results. Fig. 3 reports the average value of each metric across all correlations, positioned in a quadrant plot. Complementarily, Fig. 4 presents the full distributions of absolute error, $p(r_{\text{obs}})$, and information content. Fig. 3 shows that LCP achieves the best balance, matching the highest sign accuracy (78.8%) of KDE while providing significantly better calibration (89.2% coverage). In contrast, the uncalibrated KDE and Gaussian priors are overconfident, assigning low likelihood to r_{obs} and yielding poor coverage (59.9% and 49.1%, respectively). On the other hand, the uniform prior offers high coverage (92.3%) but suffers from poor accuracy and high absolute error ($|\hat{r} - r_{\text{obs}}| = 0.51$).

In addition, LCP significantly reduces the average information content of the correlation corpus—from 0.69 under a uniform prior to 0.27, indicating that it assigns higher likelihood to observed correlations. In contrast, the Gaussian and KDE priors increase the average information content to 4.10 and 1.73, respectively, due to their overconfident predictions. This is reflected in the long tail of low-density values in Fig. 4b. As shown in Fig. 4, LCP yields more concentrated distributions for both likelihood $p(r_{\text{obs}})$ and information content, highlighting better calibration .

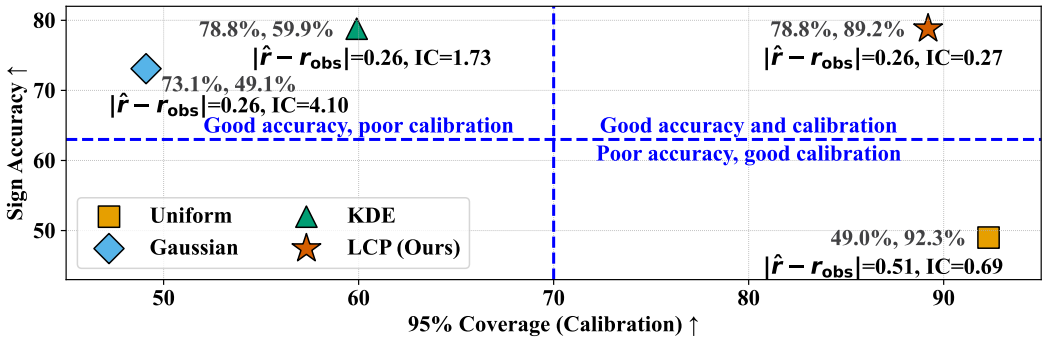

Figure 3: Accuracy vs. Calibration of Correlation Priors (IC=Information Content)

To understand the poor calibration of the Gaussian and KDE priors, we examine their kernel standard deviations. Both produce overly small σ values, leading to sharply peaked densities. The median σ is

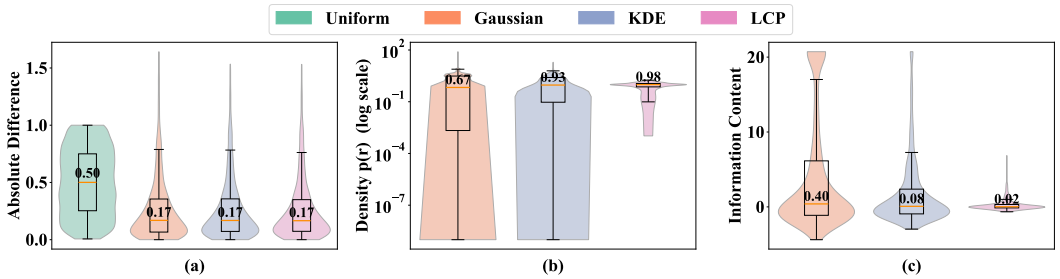

Figure 4: Full Distribution of Metrics over Different Priors

0.10 for the Gaussian prior and 0.08 for the KDE prior–both much smaller than the fixed $\sigma = 0.4$ in LCP. Under the Gaussian prior, the LLM returned $\sigma = 0.1$ in 74% (1,552/2,096) of cases. The full distribution of σ values is shown in Appendix C. As we further examine in Appendix D, this behavior arises because the LLM interprets σ as sampling variability and implicitly assumes a fixed sample size of 100—producing a default value of $\sigma = 0.1$ regardless of context. The predicted σ captures expected variation from random sampling (*aleatoric uncertainty*), but fails to adjust based on the input context or account for uncertainty arising from limited knowledge (*epistemic uncertainty*) [12, 5].

Figure 5 analyzes LCP's behavior across ten bins of observed correlation r_{obs}. The bias $\hat{r} - r_{\text{obs}}$ decreases with r_{obs}: the model overestimates strong negatives and underestimates strong positives. Sign accuracy is lowest when $|r_{\text{obs}}|$ is small, bottoming out near -0.15 and rising sharply beyond $|r_{\text{obs}}| \gtrsim 0.3$, reaching near-perfect accuracy for $|r_{\text{obs}}| \geq 0.7$. In Fig. 5c,d, the prior assigns lowest density (i.e., highest information content) to moderately negative correlations ($r_{\text{obs}} \approx -0.5$), indicating weaker estimation. In contrast, strong positives receive the highest density and lowest information content. Overall, the prior shows asymmetric error: it performs best on strong positives and struggles with moderate negatives, consistently underestimating correlation magnitude—a reflection of the LLM's conservative predictions without direct data access.

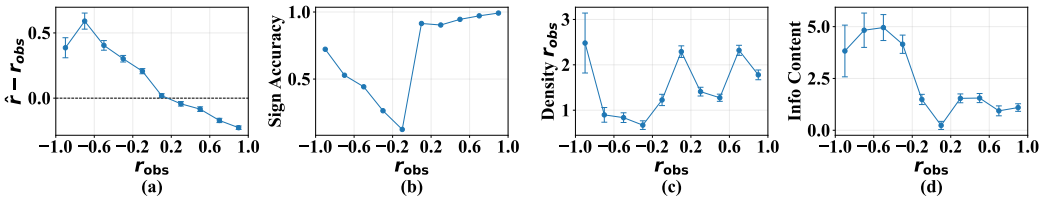

Figure 5: Performance across ten bins of the true correlation r_{obs}

Comparison of LCP and RoBERTa on Binary Correlation Classification. While LCP models a full distribution, BERT- and RoBERTa-based classifiers [14, 6] can be adapted for binary correlation prediction–determining whether a pair of variables is correlated based on a predefined threshold. We adopt the method from Trummer [30], who fine-tune RoBERTa using labeled pairs to build a correlation classifier. LCP is adapted to solve the binary classification task by thresholding its mode, enabling direct comparison with classification-based approaches.

To ensure a fair comparison, we first evaluate RoBERTa in a zero-shot setting, matching LCP, which requires no training. We then fine-tune RoBERTa on 20% of the benchmark, following standard practice for applying RoBERTa to downstream tasks. Figure 6 shows performance across different correlation thresholds. Note that RoBERTa must be re-trained for each threshold.

LCP consistently outperforms both baselines in terms of accuracy, F1, and MCC across all thresholds–achieving up to 0.84 accuracy, 0.79 F1, and 0.53 MCC–despite being entirely zero-shot. This indicates that our method provides the most balanced predictions overall. Zero-shot RoBERTa behaves like a one-class detector: it predicts correlated for every pair, yielding perfect recall but zero MCC and rapidly deteriorating accuracy/precision as the threshold tightens from 0.5 to 0.8. Fine-tuned RoBERTa corrects this imbalance to some extent after seeing 20% of the data, but its gains are threshold-specific and require retraining whenever the decision boundary moves. In contrast, By producing a full predictive distribution over r, LCP naturally adapts to different thresholds.

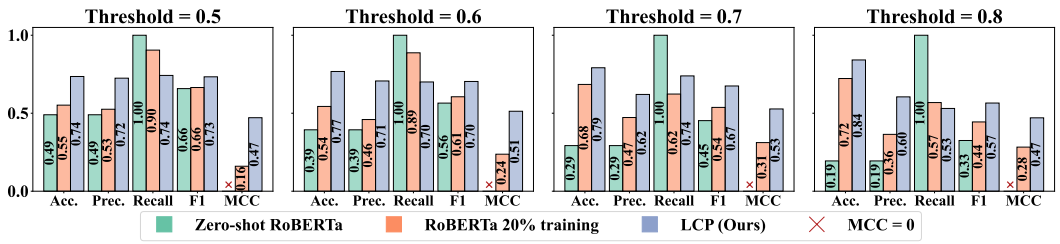

Figure 6: Classification Performance over Different Correlation thresholds

Sensitivity Analysis of LCP. We conducted a sensitivity analysis to assess the effect of two factors on LCP's robustness: (i) the choice of kernel function used for smoothing, and (ii) the validation set size used to calibrate the smoothing parameter σ.

We compared three alternative kernels-Uniform, Epanechnikov, and Triangle-against the Gaussian kernel used in our main results. For each kernel, we tuned the bandwidth parameter σ on a held-out validation set and evaluated performance on 2096 correlated pairs. The results in Table 1 show that LCP maintains strong sign prediction accuracy and calibration across all kernel choices.

Table 1: Sensitivity to kernel function used for smoothing.

Kernel	**Sign Accuracy**	$\lvert\hat{r} - r_{obs}\rvert$	**95% Coverage**
Uniform	76.4%	0.32	82.1%
Epanechnikov	78.8%	0.26	89.7%
Triangle	78.8%	0.31	90.9%
Gaussian (reference)	78.8%	0.26	89.2%

These findings suggest that LCP's performance is not sensitive to the specific choice of kernel function, as long as the bandwidth is appropriately tuned. However, due to differences in kernel shape, the optimal σ should be re-selected using a validation set when switching kernels.

Table 2: Optimal σ across validation set sizes.

Set Size	300	500	1000	2000
σ_{optimal}	0.40	0.38	0.38	0.41

We also evaluated how the optimal smoothing bandwidth σ varies with the size of the held-out validation set. As shown in Table 2, the optimal σ remains stable across different validation sizes, indicating that LCP is robust to sample size variations.

5 Using LCP to Retrieve Expert-Flagged, Hypothesis-Worthy Correlations

Can LCP support hypothesis assessment in noisy, real-world settings? We evaluate it on Nexus [8], a system designed to help domain experts discover correlations in urban data. Nexus computes 40,538 pairwise correlations from Chicago Open Data [19] by aligning and aggregating numeric attributes from different tables–either temporally (by month) or spatially (by census tract). Attributes are summarized (e.g., via mean or sum), joined on a shared key, and then correlated. This pipeline introduces real-world challenges: joins across sources, aggregation choices, and missing values–all of which impact the resulting correlations.

Of the full set, 15 correlations were labeled as hypothesis-worthy by human experts in the original Nexus evaluation. For example, a correlation between bike dock density and community wealth suggests stations are more common in affluent areas, a hypothesis studied in [7]. We use these expert-flagged examples to evaluate how well LCP retrieves hypothesis-worthy correlations in messy, transformed data. Since ground-truth correlation values are unavailable due to data aggregation and imputation, we adopt an information retrieval setup: treating the 15 expert-flagged correlations as targets within a pool of 115, formed by adding 100 random samples from the full Nexus corpus.

We compare five ranking strategies: (i) random, (ii) by absolute correlation $|r|$, (iii) by increasing probability assigned by a RoBERTa model fine-tuned on 20% of the benchmark, where lower

probability of the "correlated" class indicates higher surprise, (iv) by LLM surprise, where the LLM (GPT-4o) is given the full metadata (column names, table names, description, and observed correlation) and prompted to classify the correlation as "surprising" or "not surprising", and (v) by increasing prior likelihood $p(r_{\text{obs}})$ under LCP, treating lower likelihood as more surprising. We report Precision@5, @10, @15, and the average rank of expert-labeled correlations.

Table 3: Retrieval Performance Comparison

Method	Precision@5	Precision@10	Precision@15	Average Rank ↓
Random Ranking	0.13	0.13	0.13	58.0
Ranked by $\|r\|$	0	0	0	95.4
Ranked by RoBERTa	**0.60**	0.60	0.53	30.9
Ranked by LLM Surprise	0.4	0.2	0.13	29.1
Ranked by LCP	**0.60**	**0.80**	**0.60**	**21.5**

As shown in Table 3, ranking correlations by LCP outperforms all baselines–achieving up to 0.80 Precision@10 and reducing the average rank of expert-labeled correlations to 21.5, compared to 30.9, 58.0 and 95.4 for the RoBERTa, random and $|r|$-based rankings, respectively (see Appendix E for a derivation of the expected performance under random ranking). Ranking by $|r|$ performs worst, with the highest average rank and zero precision, as extreme correlations often reflect trivial or redundant relationships (e.g., repeated attributes across years), not meaningful insights in Chicago Open Data.

Using LCP, all four correlations related to the expert-labeled hypothesis—that bike stations are more likely to be located in wealthier areas—are ranked within the top 6. In contrast, an unsurprising correlation–the one between library visitors and library circulation–is ranked much lower at 95th. These results demonstrate that LCP can surface correlations that align with expert judgment, even in the presence of data transformations and noise.

6 Is LCP Reasoning from Context or Relying on Memorization?

We evaluate whether LCP is *reasoning from context* or simply relying on *memorization*, a crucial distinction for generalization beyond the model's pretraining data. This is essential for hypothesis assessment, where many relationships are unseen during training and depend on context. To probe this, we introduce an evaluation based on *contextual contradiction*. For each variable pair, we construct an alternate context that plausibly reverses the original correlation, simulating a counterfactual. We then re-derive the prior by prompting the LLM with this modified context (Appendix H.4). If the model adjusts its belief accordingly, it suggests reasoning from context rather than memorization.

Contradictory Context Generation. We use the Cause-Effect Pairs dataset to construct counterfactual scenarios. From this dataset, we select 84 variable pairs where the model initially predicts the correct correlation sign. For each pair, we prompt Gemini 2.5 Pro to generate a new context that plausibly reverses the original relationship (see Appendix H.3). All 84 generated contexts are manually reviewed by the authors to ensure the reversal is logically sound and free of explicit cues (e.g., phrases like "therefore there should be a negative correlation"). We assign the negated correlation $-r_{\text{obs}}$ as the new observed value. Since these contexts are synthetic and no real data exists, these new r_{obs} values serve as approximations.

Result. Table 4 shows the performance of correlation priors on reversed correlations. LCP achieves 100% sign accuracy on the original contexts, dropping slightly to 95.2% under contradictory contexts. Manual inspection reveals that two of the four errors stem from reasoning failures: the model grasps the high-level logic but fails at the final inference step in multi-hop scenarios (see Appendix F). LCP also maintains strong calibration, with 92.9% coverage at the 95% level, and achieves lower information content (0.25 vs. 0.69) and absolute error (0.30 vs. 0.55) compared to the uniform prior.

Table 4: Performance of correlation priors on correlations with contradictory contexts.

Method	Sign Acc. (↑)	$\|\hat{r} - r_{\text{obs}}\|$ (↓)	Information Content (↓)	95% Coverage (↑)
Uniform	0.464	0.55 ± 0.25	0.69 ± 0.00	92.3%
LCP (ours)	**0.952**	$\mathbf{0.30 \pm 0.28}$	$\mathbf{0.25 \pm 0.98}$	**92.9%**

This experiment shows that LCP is not merely recalling memorized correlations. When given counterfactual contexts, it updates its predictions accordingly, achieving 95.2% sign accuracy with strong calibration and low error. These results suggest that LCP generalizes beyond pretraining and behaves dynamically, a crucial property for real-world hypothesis assessment.

7 Related Work

Elicit Priors from Human Experts. O'Hagan et al. [16] and Gosling [9] introduce the SHELF framework, a structured protocol for eliciting expert judgments and converting them into probability distributions. The process involves training, individual assessments, group discussions, and consensus-building, followed by fitting a statistical distribution to the agreed-upon judgments. This human elicitation process is costly and time-consuming, whereas our approach exploits the rich knowledge encoded in LLM weights to approximate expert priors automatically.

LLMs for Regression Tasks. Several works exploit the knowledge encoded in LLMs for regression. Choi et al. [3] use LLMs for feature selection by prompting whether a variable is predictive of a given target, while others [20, 10, 1] aim to model prior distributions over feature weights. For example, Requeima et al. [20] require training examples to guide the LLM in generating output distributions, and Capstick et al. [1] assume the LLM can directly parameterize a distribution by prompting it to output means and standard deviations given feature and target names. In contrast, our work focuses on constructing a prior distribution over correlation coefficients between variable pairs before observing any data, using raw LLM logits directly—without requiring the model to parameterize a distribution.

Additional Related Work. We include further discussion on data discovery systems and automatic hypothesis generation in Appendix G.

8 Conclusions

In this paper, we propose the Logit-based Correlation Prior, an LLM-elicited prior that transforms raw output logits into a calibrated, continuous predictive distribution over correlation values—paving the way for automatic hypothesis assessment. Our experiments show (i) LCP achieves the best balance between accuracy and calibration for predicting empirical correlations, outperforming Uniform, Gaussian, and KDE priors; (ii) LCP outperforms a fine-tuned RoBERTa classifier on binary correlation classification; (iii) LCP effectively highlights hypothesis-worthy correlations flagged by human experts in noisy urban data; and (iv) LCP goes beyond memorizing correlation values from pretraining, performing contextual reasoning.

9 Limitations

Generating an LCP requires an LLM call per correlation, which can be costly at scale. To improve scalability, preprocessing steps—such as filtering out redundant variable pairs across similar datasets—can help reduce the number of required queries. LLMs may also produce false positives or negatives. An LLM may possess knowledge beyond that of human experts, causing it to dismiss correlations that are actually insightful to the experts (false negatives). It can also misinterpret well-known relationships, incorrectly flagging them as surprising (false positives), as shown in Appendix F.

10 Acknowledgments

We thank the anonymous reviewers for their constructive feedback, which significantly improved the clarity and quality of this work. This work was supported partially by the National Science Foundation (CAREER Award 2340034) and the Data Ecology Research Initiative at the Data Science Institute, University of Chicago.

References

[1] Alexander Capstick, Rahul G. Krishnan, and Payam Barnaghi. Autoelicit: Using large language models for expert prior elicitation in predictive modelling, 2025. URL https://arxiv.org/abs/2411.17284.

[2] Fernando Chirigati, Harish Doraiswamy, Theodoros Damoulas, and Juliana Freire. Data polygamy: The many-many relationships among urban spatio-temporal data sets. In *Proceedings of the 2016 International Conference on Management of Data*, pages 1011–1025, 2016.

[3] Kristy Choi, Chris Cundy, Sanjari Srivastava, and Stefano Ermon. Lmpriors: Pre-trained language models as task-specific priors. *arXiv preprint arXiv:2210.12530*, 2022.

[4] MIT Critical Data, Matthieu Komorowski, Dominic C Marshall, Justin D Salciccioli, and Yves Crutain. Exploratory data analysis. *Secondary analysis of electronic health records*, pages 185–203, 2016.

[5] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112, 2009.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, 2019.

[7] Elizabeth Flanagan, Ugo Lachapelle, and Ahmed El-Geneidy. Riding tandem: Does cycling infrastructure investment mirror gentrification and privilege in portland, or and chicago, il? *Research in Transportation Economics*, 60:14–24, 2016.

[8] Yue Gong, Sainyam Galhotra, and Raul Castro Fernandez. Nexus: Correlation discovery over collections of spatio-temporal tabular data. *Proc. ACM Manag. Data*, 2(3), May 2024. doi: 10.1145/3654957. URL https://doi.org/10.1145/3654957.

[9] John Paul Gosling. Shelf: the sheffield elicitation framework. In *Elicitation: The science and art of structuring judgement*, pages 61–93. Springer, 2017.

[10] Henry Gouk and Boyan Gao. Automated prior elicitation from large language models for bayesian logistic regression. In *AutoML Conference 2024 (Workshop Track)*, 2024. URL https://openreview.net/forum?id=euLzlnU7gz.

[11] Saint John Walker. Big data: A revolution that will transform how we live, work, and think, 2014.

[12] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017.

[13] Anthony ML Liekens, Jeroen De Knijf, Walter Daelemans, Bart Goethals, Peter De Rijk, and Jurgen Del-Favero. Biograph: unsupervised biomedical knowledge discovery via automated hypothesis generation. *Genome biology*, 12:1–12, 2011.

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[15] Joris M Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *Journal of Machine Learning Research*, 17(32):1–102, 2016.

[16] Anthony O'Hagan, Caitlin E Buck, Alireza Daneshkhah, J Richard Eiser, Paul H Garthwaite, David J Jenkinson, Jeremy E Oakley, and Tim Rakow. *Uncertain judgements: eliciting experts' probabilities*. John Wiley & Sons, 2006.

[17] OpenAI. Gpt-4o. https://platform.openai.com/docs/models/gpt-4o, 2024. Accessed: 2025-05-15.

[18] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[19] Chicago Data Portal. Chicago data portal, 2025. URL `https://data.cityofchicago.org/`.

[20] James Requeima, John Bronskill, Dami Choi, Richard Turner, and David K Duvenaud. Llm processes: Numerical predictive distributions conditioned on natural language. *Advances in Neural Information Processing Systems*, 37:109609–109671, 2024.

[21] Aécio Santos, Aline Bessa, Fernando Chirigati, Christopher Musco, and Juliana Freire. Correlation sketches for approximate join-correlation queries. In *Proceedings of the 2021 International Conference on Management of Data*, pages 1531–1544, 2021.

[22] Aécio Santos, Flip Korn, and Juliana Freire. Efficiently estimating mutual information between attributes across tables. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 193–206, 2024. doi: 10.1109/ICDE60146.2024.00022.

[23] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[24] Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[25] Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvarinen, Yoshinobu Kawahara, Takashi Washio, Patrik O Hoyer, Kenneth Bollen, and Patrik Hoyer. Directlingam: A direct method for learning a linear non-gaussian structural equation model. *Journal of Machine Learning Research-JMLR*, 12(Apr):1225–1248, 2011.

[26] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[27] Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

[28] George W Snedecor and Witiiam G Cochran. Statistical methods, 8thedn. *Ames: Iowa State Univ. Press Iowa*, 54:71–82, 1989.

[29] Scott Spangler, Angela D. Wilkins, Benjamin J. Bachman, Meena Nagarajan, Tajhal Dayaram, Peter Haas, Sam Regenbogen, Curtis R. Pickering, Austin Comer, Jeffrey N. Myers, Ioana Stanoi, Linda Kato, Ana Lelescu, Jacques J. Labrie, Neha Parikh, Andreas Martin Lisewski, Lawrence Donehower, Ying Chen, and Olivier Lichtarge. Automated hypothesis generation based on mining scientific literature. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 1877–1886, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623667. URL `https://doi.org/10.1145/2623330.2623667`.

[30] Immanuel Trummer. Can large language models predict data correlations from column names? *Proc. VLDB Endow.*, 16(13):4310–4323, September 2023. ISSN 2150-8097. doi: 10.14778/3625054.3625066. URL `https://doi.org/10.14778/3625054.3625066`.

[31] Guangzhi Xiong, Eric Xie, Amir Hassan Shariatmadari, Sikun Guo, Stefan Bekiranov, and Aidong Zhang. Improving scientific hypothesis generation with knowledge grounded large language models, 2024. URL `https://arxiv.org/abs/2411.02382`.

[32] Yizhen Zheng, Huan Yee Koh, Jiaxin Ju, Anh TN Nguyen, Lauren T May, Geoffrey I Webb, and Shirui Pan. Large language models for scientific discovery in molecular property prediction. *Nature Machine Intelligence*, pages 1–11, 2025.

[33] Yangqiaoyu Zhou, Haokun Liu, Tejes Srivastava, Hongyuan Mei, and Chenhao Tan. Hypothesis generation with large language models. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, page 117–139. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.nlp4science-1.10. URL `http://dx.doi.org/10.18653/v1/2024.nlp4science-1.10`.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim in the abstract and introduction that knowledge encoded in LLMs can support automatic hypothesis assessment. To this end, we contribute the Logit-based Calibrated Prior—a correlation prior derived from an LLM's output logits. Our evaluation shows that this prior effectively highlights correlations considered hypothesis-worthy by human experts. Thus, our main claims accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Section 9.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide all necessary details to enable full reproducibility of our experimental results. Specifically: (1) Section 2 presents a detailed description of our algorithm (see Algorithm 1); (2) Sections 3 and 6 describe our benchmark construction and evaluation procedures; and (3) we release our code and data at `https://github.com/TheDataStation/LLM-Prior-for-Correlation-Assessment`.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide source code for reproducing our experiments at `https://github.com/TheDataStation/LLM-Prior-for-Correlation-Assessment`. The repository includes a README with detailed instructions for installation, configuration, and running the experiments.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all necessary details for people to understand our results. Section 4 outlines the rationale behind the evaluation metrics and describes the setup for each method in detail.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: We report the full distribution of evaluation metrics across all runs (Fig. 4), include error bars in the binned analysis (Fig. 5), and provide standard deviations in Table 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the LLM used in Section 4, including its knowledge cutoff date; its usage cost is publicly available [17].

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed and adhered to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss positive impacts in the introduction, highlighting how our approach supports analysts in identifying hypothesis-worthy statistical relationships. Potential negative impacts are discussed in Section 9, including the risk of false positives and negatives from LLMs. We emphasize that the prior is intended to assist, not replace, human experts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper does not release data or models that have a high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external assets used in this work are properly cited, and we have ensured that their licenses and terms of use are fully respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide well-documented code and benchmark datasets at `https://github.com/TheDataStation/LLM-Prior-for-Correlation-Assessment/`.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Normality Test via Chi-Square Goodness-of-Fit

To assess whether the LLM's output distribution over correlation values conforms to a Gaussian shape, we perform a chi-square goodness-of-fit test. For each correlation prompt, we obtain a discrete probability distribution $\{(r_j, p_j)\}_{j=1}^{N}$, where each $r_j \in [-1, 1]$ is a decoded numeric value and p_j is the associated model-assigned probability mass, derived from token-level logits.

We convert the probability mass function into a set of pseudo-counts by assuming a nominal sample size $M = 1000$, yielding observed counts $O_j = M \cdot p_j$. We then estimate the mean μ and variance σ^2 of the distribution as follows:

$$\mu = \sum_{j=1}^{N} r_j \cdot p_j, \qquad \sigma^2 = \sum_{j=1}^{N} (r_j - \mu)^2 \cdot p_j.$$

Next, we compute the expected count for each support point r_j under a fitted Gaussian:

$$q_j = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_j - \mu)^2}{2\sigma^2}\right),$$

which we normalize to form a probability distribution $\tilde{p}_j = q_j / \sum_j q_j$, and then scale to expected counts $E_j = M \cdot \tilde{p}_j$.

The chi-square test statistic is computed as:

$$\chi^2 = \sum_{j=1}^{N} \frac{(O_j - E_j)^2}{E_j}.$$

The null hypothesis is that the observed distribution comes from the fitted Gaussian. We evaluate the p-value corresponding to the computed χ^2 and reject the null at the 5% significance level.

Applied to the 2,096 correlations in our benchmark, the normality hypothesis was rejected in 2,095 cases, indicating that the LLM's output distributions are poorly approximated by a parametric Gaussian form. This result justifies our non-parametric approach, which avoids imposing a fixed distributional shape.

B Preventing Length Biases

In our setup that uses the GPT-4o tokenizer, both positive and negative correlation values are represented with the same number of tokens. Specifically, both forms include four tokens: sign, integer part, decimal point, and fractional part.

For example, the response:

```
{
    "coefficient": "0.5"
}
```

is tokenized as `' "'`, `'0'`, `'.'`, `'5'`.

While:

```
{
    "coefficient": "-0.6"
}
```

is tokenized as `' "-'`, `'0'`, `'.'`, `'6'`.

In our decoding process, we always treat the token immediately following the colon as the sign token, trimming the prefix (`' "'`) before it. For positive numbers, the sign token can be an empty string or `'+'`, while for negatives it is `'-'`. As a result, negative numbers do not suffer a disadvantage due to extra token length.

C Distribution of Kernel Standard Deviations

Figure 7 shows the distribution of kernel standard deviations σ used in the Gaussian and KDE priors. Both priors tend to produce small σ values, contributing to overconfident and poorly calibrated predictions. The median σ is 0.10 for the Gaussian prior and 0.08 for the KDE prior.

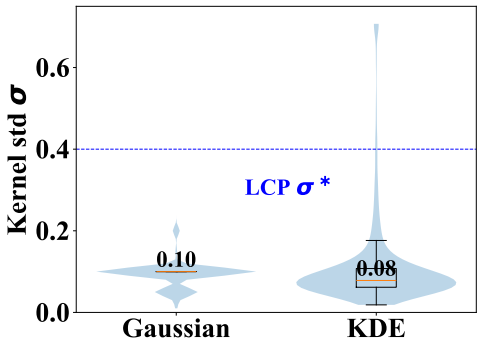

Figure 7: Distribution of kernel standard deviations for Gaussian and KDE priors.

D Understanding the LLM's Behavior in Reporting Standard Deviations

The LLM (GPT-4o) favors the value 0.1 when reporting standard deviations. $\sigma = 0.1$ appears in 74% of cases (1,552 out of 2,096 prompts). To better understand this behavior, we conducted a targeted analysis of the LLM's internal assumptions when predicting σ.

We prompted GPT-4o with 50 column pairs whose names were random strings with no semantic meaning (e.g., abc123, xzy987). This design removes contextual cues, allowing us to observe the model's default behavior under maximum uncertainty. In all 50 cases, the predicted correlation coefficient was exactly zero, and in 46 out of 50 cases, the predicted standard deviation was 0.1. The strong preference for $\sigma = 0.1$ even in the absence of context suggests that, when prompted to express its uncertainty as the standard deviation of a normal distribution, the LLM may default to fixed assumptions—such as an implicit sample size—rather than adjusting its estimate based on contextual information.

To investigate why $\sigma = 0.1$ is so commonly predicted, we analyzed the distribution of Sample Pearson's correlation coefficient r under the assumption that the true correlation $\rho = 0$. When data is sampled from a bivariate normal distribution with zero correlation, the sampling distribution of r has the following form:

$$f_r(r) = \frac{\Gamma\left(\frac{n-1}{2}\right)}{\sqrt{\pi}\,\Gamma\left(\frac{n-2}{2}\right)} \cdot (1-r^2)^{\frac{n-4}{2}}, \quad \text{for } -1 < r < 1,$$

where n is the sample size and $\Gamma(\cdot)$ is the gamma function. This distribution is bell-shaped, and its standard deviation decreases as n increases. Specifically, the variance is given by $\text{Var}[r] = \frac{1}{n-1}$, so the standard deviation is $\text{SD}[r] = \frac{1}{\sqrt{n-1}}$. When $n = 100$, this yields $\text{SD}[r] \approx 0.1$, which aligns with the value most often returned by the LLM.

To test this hypothesis, we asked GPT-4o to explicitly state the sample size it assumes when estimating uncertainty. In all 50 test cases, it responded with $n = 100$, confirming that its predicted standard deviation reflects a fixed assumption about sample size rather than context-specific reasoning.

This result suggests that GPT-4o's predicted σ reflects *aleatoric uncertainty*—uncertainty due to random sampling around a fixed true correlation. The model assumes a fixed value of ρ and estimates how much empirical values of r might vary if repeatedly sampled. However, the type of uncertainty we aim to capture in this work is primarily epistemic uncertainty—uncertainty arising from the LLM's lack of knowledge about the relationship.

For example, if X represents altitude and Y represents precipitation, the correlation might be 0.7 in the U.S. and 0.6 in Germany. If the LLM does not know which country the data comes from, the true correlation is ambiguous—not due to sampling variability, but due to missing contextual information. This is epistemic uncertainty. Unlike aleatoric uncertainty, it can be reduced by providing more metadata or context about the table and dataset.

In summary, GPT-4o's use of $\sigma = 0.1$ reveals a key limitation of the Gaussian prior: it captures a fixed form of uncertainty based on sampling variability but fails to account for uncertainty arising from a lack of knowledge—such as ambiguity in context or missing information. As a result, it often produces overconfident predictions.

E Expected Performance Under Random Ranking

The random ranking baseline yields a constant expected Precision@k of approximately $15/115 = 0.13$, since each of the 15 expert correlations is equally likely to appear in any of the 115 positions. The corresponding average rank is 58.0, which is the expected value of a discrete uniform distribution over ranks 1 to 115: $\mathbb{E}[\text{rank}] = \frac{1+115}{2} = 58$

F Multi-hop Reasoning Failure

We analyze a case involving the variables *Horsepower* and *acceleration time* (measured as 0–100 km/h time), which are originally *negatively* correlated: higher horsepower typically leads to faster acceleration and thus shorter time.

We introduce a counterfactual context: *"Imagine a world where cars are powered by highly unstable, experimental engines. More horsepower means a greater chance of catastrophic failure, forcing drivers to accelerate slowly and cautiously to avoid explosions."* Under this context, we would expect the correlation to flip—*higher* horsepower should now lead to *longer* acceleration time, yielding a *positive* correlation.

However, the LLM still predicts a negative correlation. Its explanation is as follows:

> "In this hypothetical world, cars with more horsepower are more prone to catastrophic failure due to their unstable engines. As a result, drivers of high-horsepower cars are likely to accelerate more slowly and cautiously to avoid triggering an explosion. This cautious driving behavior would lead to longer 0–100 km/h acceleration times. Conversely, cars with lower horsepower are less likely to experience such failures, allowing drivers to accelerate more quickly. Therefore, there is likely a strong negative correlation between horsepower and acceleration time, as higher horsepower results in slower acceleration due to the need for caution."

While the model correctly reasons that higher horsepower leads to slower acceleration, it fails at the final reasoning step—mapping *slower acceleration* to *longer time*. This results in an incorrect correlation prediction, highlighting a multi-hop reasoning failure.

G Additional Related Work

Data Discovery. Modern data discovery systems [2, 25, 21, 22, 8, 30] efficiently compute statistical relationships such as correlations, causality, and mutual information across datasets. They extend beyond analyzing variables within a single dataset to discovering relationships between variables across multiple datasets by automatically transforming and joining different datasets. Specifically, for correlation discovery, Nexus [8] aligns large repositories of spatio-temporal datasets and identifies correlations, while Trummer [30] use a RoBERTa classifier to predict whether two variables are correlated based solely on their names. Data discovery systems surface a large number of potential relationships, but helping analysts identify the ones most relevant to their needs remains a key challenge in this field. Our approach, which uses an LLM-elicited prior to rank relationships, serves as a stepping stone toward addressing this challenge.

Automatic Hypothesis Generation. While data discovery systems identify statistical relationships from structured data that may lead to new hypotheses, a complementary line of work [29, 31,

32, 13, 33] focuses on mining unstructured scientific literature. These methods extract semantic knowledge—such as entities, links, and claims—from text, and store this knowledge for further analysis. Some approaches [29, 13] construct knowledge graphs and use graph analysis to suggest hypotheses, while others leverage language models to analyze the knowledge and suggest hypotheses directly [32, 31]. Zhou et al. [33] explores combining literature-derived insights with structured data.

H Prompts

H.1 Correlation Prediction Prompt to Construct Logit-based Calibrated Prior

Correlation Prediction Prompt for LCP

Task: You are given two attributes from a tabular dataset. Your task is to predict the Pearson's correlation coefficient between the two attributes.

Now, begin to solve the following problem:

Attributes:

– `{attr1}`

– `{attr2}`

Source Table: `{table}`

Descriptions:

- Dataset Description: {tbl_desc}
- Attribute Descriptions:

  `{attr1}`: {var1_desc}
  `{attr2}`: {var2_desc}

Respond with your predictions in the following format:

```
{
  "coefficient": "<predicted correlation coefficient>",
}
```

H.2 Correlation Prediction Prompt to Construct Gaussian Prior

Correlation Prediction Prompt for LCP

Task: You are given two attributes from a tabular dataset. Your task is to predict the Pearson's correlation coefficient between the two attributes and estimate your confidence in the predicted correlation by providing the standard deviation as a measure of uncertainty. Note that the standard deviation cannot be zero.

Now, begin to solve the following problem:

Attributes:

– `{attr1}`

– `{attr2}`

Source Table: `{table}`

Descriptions:

- Dataset Description: {tbl_desc}
- Attribute Descriptions:

  `{attr1}`: {var1_desc}
  `{attr2}`: {var2_desc}

Respond with your predictions in the following format:

```
{
  "coefficient": "<predicted correlation coefficient>",
  "standard deviation": "<predicted uncertainty>",
}
```

H.3 Generate Contradictory Context

Counterfactual Context Generation Prompt

Task:
You are given two attributes and the expected correlation between them from a tabular dataset. Your task is to invent a hypothetical context that *flips* the expected relationship between these attributes.
For example, on Earth, income and education are positively correlated; in an alternate world where education makes people less capable, income and education would be negatively correlated.

Please provide your new context in **2–3 concise sentences**, avoiding any explicit mention of the correlation.

Now, solve the following:

Attributes:

– `{attr1}`

– `{attr2}`

Source Table: `{table}`

Descriptions:

- Dataset Description: {tbl_desc}
- Attribute Descriptions:

  `{attr1}`: {var1_desc}
  `{attr2}`: {var2_desc}

Expected Correlation: `{r_obs}`

Respond in JSON:

```
{
  "new_context": ""
}
```

H.4 Correlation Prediction with Hypothetical context

Correlation Prediction with Hypothetical context

Task: Given two attributes from a tabular dataset and a hypothetical context (which may differ from Earth), predict the Pearson correlation coefficient between them.

Guidelines:

- Use the scenario described under Context to inform your reasoning.
- Return a single floating-point value in the range `[-1, 1]`.

Now, solve the following:

Context:
{context}

Attributes:

- {attr1}
- {attr2}

Source Table: {table}

Descriptions:

- Dataset Description: {tbl_desc}
- Attribute Descriptions:

  {attr1}: {var1_desc}
  {attr2}: {var2_desc}

Format your answer as:

```
{
  "coefficient": "<predicted correlation coefficient>",
  "explanation": "<explanation of the prediction>"
}
```

H.5 Column Semantics Quality Assessment

Column Semantics Quality Assessment Prompt

You are given a column name and the context in which it appears. Your task is to judge whether the column name clearly and accurately conveys its meaning.

Column Name: {col_name}

Dataset Name: {dataset_name}

Dataset Description: {dataset_desc}

Please respond in JSON using exactly this format:

```
{
  "valid": "<yes or no>"
}
```