BAYESIAN FORECASTING OF LLM OUTPUTS FROM PRE-GENERATION HIDDEN STATES

Anonymous authors
Paper under double-blind review

ABSTRACT

The stochasticity of next-token prediction in large language models (LLMs) implies that LLM outputs should be represented probabilistically. Yet for problem settings in which outputs take on a structured format, obtaining such probabilities can be costly, as this requires repeated generation in response to a user's query. In this paper we propose a computationally cheap alternative to approximate LLM-derived probabilities. Using pre-generation hidden state representations of an LLM, our method forms predictive distributions that govern predefined concepts of the output space, aiming to forecast empirically-derived LLM probabilities. The approximating nature of the approach, a consequence of limited information and lack of foresight in hidden states, implies that a proper treatment of uncertainty is essential if we wish to place confidence in the resulting distributions. To this end, we utilize Bayesian classifiers in order to decompose a predictive distribution into aleatoric uncertainty, namely the LLM stochasticity that we intend to model, and epistemic uncertainty, or the model uncertainty arising from a lack of evidence in the LLM hidden states. We ground our work in the domain of visualization specification generation, representative of a task for which outputs adhere to a predefined grammar, and output concepts, such as graphical marks and visual channels, are well-defined. Experimental results illustrate that hidden states encode substantial information required for forecasting visualization design concepts, where forecasting difficulty increases towards composite tasks that require both context understanding and response planning. Hidden states also provide meaningful evidence for estimating uncertainty, correlated with inherent difficulty of the given query, suggesting that uncertainty can act as a meaningful criteria for evaluating forecasting quality.

1 INTRODUCTION

The analytical use of large language models (LLMs) is often an iterative process. Users iterate on input prompts by exploring variations in analysis objectives and problem framing Zhao et al. (2024), and even when fixing the input, it can be helpful to generate multiple responses from the LLM for assessing consistency or exploring alternative outcomes (Wang et al., 2023). The variations in both input and output can make LLM generation rather expensive, namely in computational cost and latency (Chen et al., 2026). This has motivated research into making LLM generation more efficient, and in particular, what can be predicted about an LLM's outputs from its internal representations. Specifically, prior works have studied the extent to which pre-generation hidden states are predictive of what an LLM will generate. This ranges from future token prediction (Pal et al., 2023; Dong et al., 2025; Piskorz et al., 2026; Gurnee et al., 2026), response length (Merzouk et al., 2026), answer correctness (Cencerrado et al., 2025), output variability Kossen et al. (2024), and instruction following (Ashok & May, 2026). We refer to this type of prediction task as *forecasting*, which can estimate aspects of the generator's behavior without autoregressively decoding complete responses.

In using a forecasting model for analytical problems, we can gain the benefits of computation efficiency: the ability to rapidly iterate on inputs, and quickly assess the diversity of LLM responses. However, two problems arise in this setting. First, we anticipate that the LLM does not produce just a single response but a distribution of responses, owing to the stochasticity of next-token prediction in LLMs. Some precedence exists in handling output variability, whether using conformal prediction for discretely-valued outputs Ashok & May (2026), or quantile-based regression for numeric-valued

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

outputs Piskorz et al. (2026). Second, as is common to most prior works on forecasting, the building of predictive models necessitates the collection of training data on which to optimize models. For complex problem domains, ensuring good generalization might require large training datasets that are prohibitively expensive to collect. As such, understanding model uncertainty is important when using the forecasting model, e.g., predictions made on certain inputs might be more reliable than other inputs. We thus have two sources of uncertainty to contend with – uncertainty regarding the output, and uncertainty in the learned forecaster. Whether we can tease out these different uncertainty sources is important for reliable downstream use, yet to the best of our knowledge this remains an open question.

To this end, we propose the use of Bayesian models built on LLM hidden states for uncertainty estimation in forecasting. Specifically, we employ Bayesian linear classifiers – a distinct classifier constructed for each categorical dimension of the output – for which the resulting predictive distributions aim to approximate the LLM's empirically-derived distributions over its outputs. A Bayesian formulation is attractive in our scenario as it admits a decomposition of uncertainty into *epistemic uncertainty*, or model uncertainty that, typically, is due to limited evidence in training data, and *aleatoric uncertainty*, or uncertainty that cannot be reduced with additional data. Aleatoric uncertainty corresponds to the stochasticity in LLM outputs, thus rather than simply treated as label noise, in our setting this is precisely what we wish to represent in the forecaster.

We ground forecasting in the context of visualization authoring, where datasets and analytical requests can lead to multiple plausible visualization designs (Luo et al., 2026), and categorical dimensions correspond to well-defined visualization design choices such as graphical marks and encoding types. Moreover, the combination of analysis goals and data descriptions leads to a complicated space of inputs, and thus nontrivial model uncertainty exists. We evaluate both the recovery of output distributions and the interpretation of Bayesian uncertainty estimates against repeated-generation references. We find that estimated output variability agrees well with observed generator variability across many design dimensions, while epistemic uncertainty is associated with larger distributional error. We also identify design dimensions, particularly data channel-to-visualization channel bindings, for which substantial mismatch remains due to the complex nature of the task required for the LLM. Together, these analyses characterize the usefulness of uncertainty estimates for interpreting forecast reliability.

2 Related Work

Our work is most closely related with research on probing and forecasting using LLM hidden states, as well as work that aims to estimate the uncertainty of LLM responses.

Probing and Forecasting with LLM representations In recent years, numerous work has been devoted to understanding what information is contained in the hidden state representations of LLMs, both from the perspective of probing a given input, as well as forecasting what the LLM will subsequently output. Probing for predetermined concepts in a given input has received significant attention, see AxBench (Wu et al., 2025) for an overview of detection and steering methods. The utility of such approaches has been demonstrated in the context of activation-based monitoring (McKenzie et al., 2026; Kramár et al., 2026), in order to detect misaligned model behaviors. In contrast, the problem of *forecasting*, where the goal is to predict properties of LLM-generated output given pre-generation hidden states, has seen relatively less attention. Numerous works have studied what is possible to predict just from the LLM's representation, ranging from: the number of tokens to generate (Merzouk et al., 2026), answer correctness (Cencerrado et al., 2025), instruction adherence (Ashok & May, 2026), and named entities within generated storylines (Dong et al., 2025). The primary benefit of theses approaches is that the costly process of LLM evaluation is replaced with a computationally cheap surrogate, often just a linear model, if not a simple MLP. Yet these approaches are far from perfect, as one should expect that to infer *exactly* what the LLM will generate requires using its full set of parameters. Recent work has studied more precisely what an LLM's representation contains regarding what it plans to generate (Xu et al., 2026; Hanna & Ameisen, 2026; Ustaomeroglu et al., 2026), identifying the limits of inferring an LLM's plan strictly from its representations. In our work we wish to study the predictability of categorical dimensions found in the output, given an LLM's representation computed on the input query. Simultaneously, we acknowl-

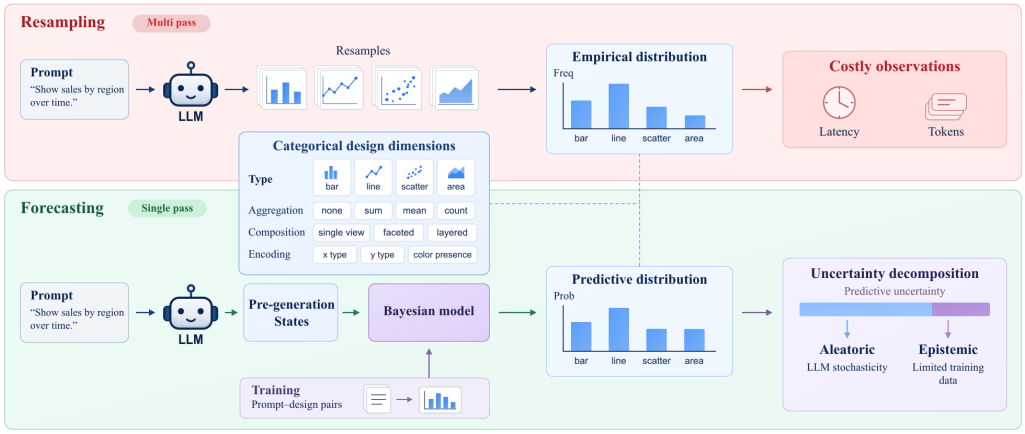


Figure 1: Overview of the benefit from forecasting. LLM makes visualization design decisions on discrete design dimensions. In naive approach, the frequency of design decisions should be collected under repeated prompting of an LLM, which is costly. On the other hand, with a forecasting model that uses pre-generation hidden states of the LLM, the same distribution over design dimensions can be predicted in an inexpensive manner. To forecast, we train a Bayesian model over a corpus of prompt-generated design pairs. Furthermore, because our model is Bayesian, to identify limitations in the approximated distribution, we decompose predictive uncertainty into epistemic and aleatoric parts that separate uncertainty arising from lack of evidence in training data and the LLMs' own stochasticity.

edge that there might exist limitations in what can be predicted, and as such, a proper treatment of predictive uncertainty becomes essential.

LLM uncertainty A considerable amount of work has been devoted to quantifying the uncertainty of LLM-generated output; please see (Xia et al., 2025) for a survey. Works can, roughly, be divided into (1) measuring output diversity (Kuhn et al., 2023; Farquhar et al., 2024), and (2) defining a suitable notion of variability in LLM representations (Kossen et al., 2024). These works are principally concerned with problems for which it is presumed the ground-truth is deterministic, and thus any amount of variability (though typically semantic variability) is treated as uncertainty. Yet for our setting, we do not treat a single LLM response as being the sole ground-truth for a given input query. Rather, we presume multiple LLM responses are equally plausible in addressing the query's request, and it is this variability – the aleatoric uncertainty – that we wish to predict. Yet to confidently state that a model's predictive distribution accurately represents the LLM's distribution conceals the epistemic uncertainty of the forecasting model. As such, Bayesian uncertainty decomposition (Houlsby et al., 2011; Kendall & Gal, 2017; Depeweg et al., 2018) becomes an essential tool to tease out different forms of uncertainty. Recent work has studied the ability of an LLM to capture ground-truth aleatoric uncertainty (Tomov et al., 2025), finding there exists limitations. Yet our problem is different in that it is precisely the LLM's uncertainty—a function of its stochasticity in next-token prediction—that we aim to capture. This aligns our work most closely with (Piskorz et al., 2026), yet our problem setting is different in that we are not concerned with numerical predictive distributions, but rather, a collection of discrete distributions from a single input query, under which we anticipate model uncertainty due to the complexity of the problem domain.

3 Methods

3.1 Problem Setup

In our problem setting, we assume access to an LLM denoted by p_θ which we can both prompt, and collect hidden states from the model's computed representations. We presume that a user is issuing a query to the LLM, a request to generate an output that adheres to a predefined structure, for instance

a context-free grammar or a domain-specific language. This structure admits a decomposition of the output into a set of categorical dimensions, a subset of which reference aspects of the user's query. Moreover, we assume that there is not a single optimal response, but rather the user's query can be satisfied with a set of outputs, each consisting of various trade-offs.

Our setting most resembles design problems for which multiple plausible solutions exist for a single input, for instance shape modeling, user interface design, and procedural layout generation. We limit our study to the problem of visualization design, though extensions to other domains would be straightforward. We assume the user's input contains the description of a given dataset that they wish to visualize, and a description of analytical goals that the visualization should satisfy. The dataset is expressed through its set of field names, along with a summary of its data items, expressed either as a subset of items in tabular form, or as summary statistics. A user's analytical goals correspond to high-level tasks that a visualization is capable of addressing, while also making reference to the problem domain implied by the dataset via its field names. No references are made to specific visualization design choices, rather, tasks such as trends, correlation, or outliers in the data are requested, and it is the responsibility of the LLM to choose relevant visualization designs. The LLM is instructed to generate a visualization specification that adheres to a predefined structure, namely a visualization grammar, from which we can extract categorical dimensions that correspond to essential properties of visualization design, for instance graphical marks (point, bar, line), visual channels (x position, y position, size, color), channel types (numeric, ordinal, nominal), field aggregation (averaging, binning), and view composition. Moreover, we are concerned with the problem of mapping fields mentioned in the input to visual channels.

Given this problem setting, in prompting the LLM for a visualization specification, we anticipate that the LLM can generate different outputs for the same input prompt, namely different designs that satisfy the user's query in different ways. This is a consequence of the stochasticity in token generation – sampling from a distribution of tokens at each generation step. Thus in order to faithfully represent the LLM's output, for each categorical dimension we are interested in forming a probability distribution over its constituent elements. This can be empirically estimated via repeated prompting of the LLM:

$$P_x(f) \;=\; \Pr_{y \sim p_\theta(\cdot|x)} \big[\, \phi(y) = f \,\big], \tag{1}$$

where x represents the user's prompt, y the structured output generated by the LLM, and $\phi(y)$ represents the mapping of the output onto an element of the categorical dimension.

Although straightforward to perform, it can be prohibitively expensive to obtain reliable distribution estimates in this manner. The number of times necessary to prompt the LLM depends on the complexity of the distribution, itself a function of the user query, and the category's cardinality. This motivates alternative, computationally cheaper methods for approximating the LLM-derived distribution. In particular, we consider methods which eschew LLM token generation, and instead form the distribution based on information collected from the input prompt, namely the LLM's hidden states computed on input tokens. Our problem can thus be seen as an instance of *forecasting*: given the input, we aim to forecast what would be generated by the LLM.

Much prior work has studied what information is carried in LLM hidden states, whether probing for knowledge about the input, or predicting what the LLM will subsequently output. In particular, recent work has studied the prediction of probability distributions over discrete and numeric variables, relying on repeated sampling of an LLM to obtain targets for, respectively, conformal prediction Ashok & May (2026) and quantile-based regression Piskorz et al. (2026). Yet our problem setting is distinct in that the space of input prompts is considerably complex: the input space spans both datasets, as well as data-dependent descriptions of a user's analytical goals. Given a fixed budget of samples used for training a forecasting model, a trade-off exists in obtaining good coverage over the input, and good estimates of the LLM's distribution governing the output. We thus view uncertainty quantification of the forecasting model as an important problem to be addressed: we would like to have the model not only provide the predictive distribution for a given input, but also report the uncertainty in its prediction.

3.2 Bayesian Forecasting of LLM Outputs

To this end, we propose the use of Bayesian methods for obtaining uncertainty estimates in forecasting LLM outputs. Specifically, denote the training dataset for a single categorical dimension as

$\mathcal{D} := \{(\mathbf{z}_1, c_1), \ldots, (\mathbf{z}_N, c_N)\}$ where $\mathbf{z}_n \in \mathbb{R}^D$ are vector-based representations corresponding to LLM hidden states, and c_n the corresponding label generated by the LLM. As we are concerned with classification, we define the likelihood in our model to correspond to standard multinomial logistic regression, and a place a Gaussian prior on the model parameters $\mathbf{W} \in \mathbb{R}^{K \times D}$ with K the category's cardinality and D the hidden state dimensionality, where the prior consists of a learnable variance hyperparameter. From here we may form the posterior distribution over the model parameters $p(\mathbf{W} \mid \mathcal{D})$, and in turn, utilize the standard uncertainty decomposition (Houlsby et al., 2011; Depeweg et al., 2018; Kendall & Gal, 2017):

$$\underbrace{H[\bar{p}(c \mid \mathbf{z})]}_{\text{total}} = \underbrace{\mathbb{E}_{\mathbf{W}}[H[p(c \mid \mathbf{W}, \mathbf{z})]]}_{aleatoric} + \underbrace{I(c; \mathbf{W} \mid \mathbf{z})}_{\text{epistemic}}. \tag{2}$$

Intuitively, the aleatoric uncertainty is a model-averaged measurement of how much the probability mass is spread. Note that this quantity can be large, despite that fact samples of the posterior $\mathbf{W}$ give roughly the same distribution $p(c \mid \mathbf{W}, \mathbf{z})$, and thus similar entropy. We interpret this case as a *confident* forecasting prediction, and should accurately reflect the LLM's empirically-estimated distribution (c.f. Eq. 1). The epistemic uncertainty, or the mutual information between posterior and predictive distribution, conditioned on pre-generation state, measures the dependency between these quantities. It is minimized when the predictive distribution does not change as we vary the model parameters, indicative of stable predictions under the posterior. Traditionally, this is due to sufficient evidence in the training data, namely $\mathbf{z}$ is "close" to some region of the training data space.

It is important to distinguish the prompt's representation $\mathbf{z}$ from the prompt itself x. Denoting the LLM's empirical distribution as π_x, then ideally, the entropy $H[\pi_x]$ will accurately match $H[p(c \mid \mathbf{W}, \mathbf{z})]$, e.g. the aleatoric bit of the forecaster. Yet, a hypothetical loss of information in the LLM's representation implies the forecaster's entropy should be at least as large as the LLM-induced entropy $H[\pi_x]$. As prior works have studied Xu et al. (2026); Hanna & Ameisen (2026); Ustaomeroglu et al. (2026) this loss of information is to be expected. As such, an important quantity that we study in this work is the so-called *readout gap* (model misspecification, in the language of uncertainty literature): uncertainty that no amount of training data can remove, because relevant information for forecasting is absent from $\mathbf{z}$. The gap is zero only if $\mathbf{z}$ preserves everything about the prompt that the distribution depends on. The results in §5 report these three parts—the generator's conditional entropy, the epistemic term, and the readout gap—for every categorical dimension.

4 Experimental Results

To show the benefits of Bayesian forecasting, we report on experimental results that demonstrate both the accuracy of our models, and the quality of their uncertainty estimates. We first report on details related to the dataset used in our experiments, visualization generation, construction of the Bayesian models, and details on evaluation.

Dataset For our problem setting of visualization authoring, we expect the input prompt for an LLM to be descriptions of an analysis goal and a dataset. Additionally, we want to make sure that datasets cover a variety of domains, and we would like to organize examples according to tasks (Amar et al., 2005; Saket et al., 2019) common to visualization design. As existing datasets (Luo et al., 2021; Srinivasan et al., 2021; Luo et al., 2026) fall short in either structure or number of examples, we have curated our own corpus of text-to-vis examples, giving 113k instances. Each instance pairs a data context with analytical/visualization request that does not involve design decisions. Please see Appendix A for details on the corpus construction.

Visualization Generation Our experiments study three LLMs – Gemma-4 E4B, Qwen3-14, and Gemma-4 31B. An LLM is instructed to generate a visualization for a provided analysis goal and dataset, ensuring that the output adheres to the visualization grammar of Vega-Lite Satyanarayan et al. (2017). The structure of Vega-Lite enables us to identify a set of categorical dimensions that define the classification tasks used for building Bayesian classifiers, organized around context understanding (A: reading), forecasting (B: design), and composite (C: binding) tasks. **Reading** represents a more traditional input probe concerning aspects of the dataset, included as a type of control task to compare against forecasting tasks. **Design** represents essential properties for describing a visualization, for instance graphical marks, encoding channels, encoding types, and aggregation. **Binding**

tasks refer to properties of the visualization which reference the dataset, namely binding fields of the dataset to encoding channels. Please see Appendix B for additional details on generation.

Approximate Inference We use Hamiltonian Monte Carlo Neal (2011) (HMC) to collect posterior samples, used for making predictions and uncertainty decomposition. NUTS Hoffman & Gelman (2014) is used to automate parameter configurations in HMC. The classifiers utilize pre-generation hidden states, where we select layer, pooling scheme, and prior scale according to validation log-loss. Please see Appendix C for additional inference details.

Evaluation To evaluate predictive distributions, we select 2,000 test prompts per generator, stratified by analytical task. We generate 20 responses per prompt to empirically estimate LLM distributions. Classes of maximum likelihood from the forecaster's posterior predictive distribution and the LLM distribution are used to compute micro- and macro-F1. We additionally report forward KL divergence, $D_{\mathrm{KL}}(\hat{\pi}_x \,\|\, \bar{p}_x)$, where $\bar{p}_x$ is the forecasted distribution, in order to directly compare distributions. To interpret F1 scores, *floors* represent prediction without prompt information: majority-class prediction of micro-F1 and uniform random guessing for macro-F1. Reported *ceilings* are empirical oracle, estimated from repeated generations, accounting for the generator's stochastic labels. These bounds are detailed in Appendix D, while comparisons to other baseline features are reported in Appendix E.3.

4.1 Forecasting Accuracy

Fig. 2 depicts model accuracy under the metrics of micro and macro-F1 and KL divergence, organized by read, design, and binding tasks. We find the **read** tasks perform quite well: information about the given context (column types), task prompt length, number of data columns, and number of rows. The exception is the prediction of the data domain, likely because domains can overlap, e.g. a financial dataset can also be considered retail or e-commerce. Inspecting macro and micro-F1 scores for **design** tasks, we find that the forecasters close much of the gap with the ceiling (e.g. highest possible accuracy accounting for stochasticity in LLM generation). In particular, for structural aspects of the visualization (graphical marks, view arrangement), we find forecasters perform rather well, while more fine-grained dimensions, e.g. aggregation and encoding types, the gap with the ceiling starts to widen. The **binding** task is most difficult of all, because it requires understanding of both context and design.

As F1 scores only focus on comparing classes of largest probability, this misses information about the *rest* of the probability distribution, e.g. other probable design options that one might have in authoring a visualization. KL divergence complements these scores to compare the distributions as a whole, shown in Fig. 2(c). As an illustrative example, consider the "mark 2" dimension – graphical marks chosen for the secondary view in a multiple-view design. F1 scores might appear to be rather low across models, yet KL divergence indicates forecasters still learn a good part of the underlying distribution of mark choices. This is notable for the E4B and 14B models, where KL divergence is found to be much lower than the LLM distribution's averaged entropy (gray tick mark). Thus despite the tendency for LLM-derived distributions to spread their mass in mark choices, the forecaster's predictive distributions accurately match against LLM-recommended marks.

These patterns hold across model scale: the same dimensions are easy and hard for E4B, 14B, and 31B, so larger conditioning vectors do not automatically carry richer design information. Looking at the unconditional baseline divergence, the 31B ceilings sit higher on nearly every dimension, while its model entropies are systematically the lowest: the larger model is not less capable of diverse designs, but more decided about which design fits a given data and task.

4.2 Uncertainty Decomposition

Figure 3-(a) reports on the uncertainty estimation of our forecasting models, namely how much uncertainty is reduced relative to a baseline uniform distribution, the amount attributed to epistemic uncertainty, and a decomposition of aleatoric uncertainty into the LLM's stochasticity and the remaining readout gap. We report in units of nats and perplexity (effective number of choices). Overall, with the exception of binding tasks, we find the readout gap amounts to roughly 0.3 – 1 choices across dimensions, indicating that although the aleatoric uncertainty gives overly conservative es-

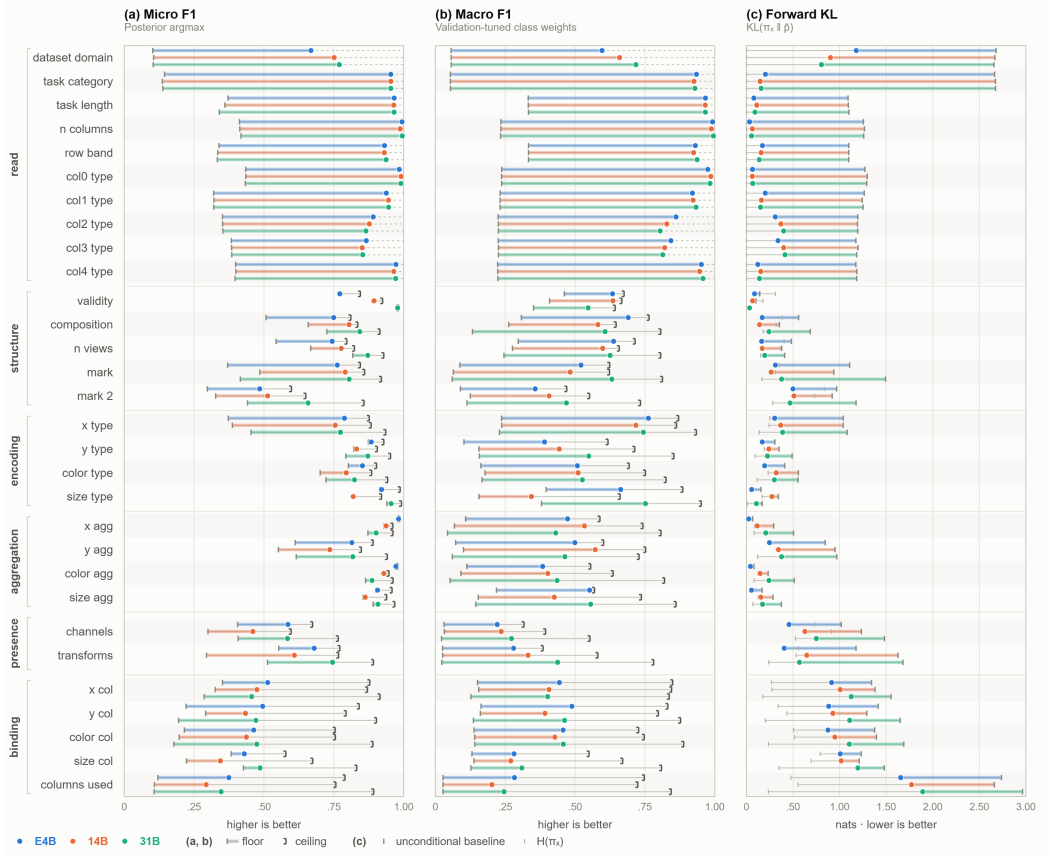


Figure 2: Performance of probes **read** and forecasters **structure, ..., binding** per individual design dimension, under: (a) micro F1, (b) macro F1, and (c) KL divergence from the empirically-estimated distribution to the forecast. Bars encode gain over each metric's reference: the label-only floor in (a, b), a prompt-independent baseline in (c), where zero is optimal. Brackets encode per-model ceilings; floors, ceilings, and baselines are per-model. Panels (a, b) score the full test split, (c) the resampled prompts only. Note the read tasks in (c): the target is deterministic, so their residual divergence is readout error alone. On the read tasks, probing performance is high, mainly because hidden states more readily encode attributes of the context. Performance degrades towards forecasting tasks where binding task struggling due to the tasks' complexity, having to read the context and forecast at the same time.

timates, it remains close to LLM stochasticity. Nevertheless, the excess uncertainty can likely be attributed to the limited information found in the LLM's hidden states, whereas the LLM has access to more information during autoregressive generation (Appendix E.1 shows that a higher-capacity forecaster does not improve on accuracy). We find that the epistemic uncertainty tends to increase as LLM models increase in size, e.g. it is highest for the 31B model. We also find epistemic uncertainty is higher when we have limited data for the task at hand, e.g. designs with multiple views show up less than single view designs, thus secondary marks have higher epistemic uncertainty than primary marks.

On average, the epistemic uncertainty might appear rather small, but importantly, we find that it is highly predictive of error in the forecaster's predictions. This is depicted in Fig. 3-(b), where we plot the correlation between epistemic uncertainty and KL divergence. Interestingly, the most capable model (31B) tends to show the strongest relationship. We argue that this is due to richer information in LLM hidden states as a result of higher dimensionality, whose trend could hold as the model gets larger. In Fig. 3-(c), we sort test examples by quantiles of the epistemic uncertainty, and plot the readout gap in each bin (e.g. 20% lowest epistemic, 40% lowest epistemic, etc..). The intuition is that as model uncertainty decreases, we expect the aleatoric uncertainty to better match

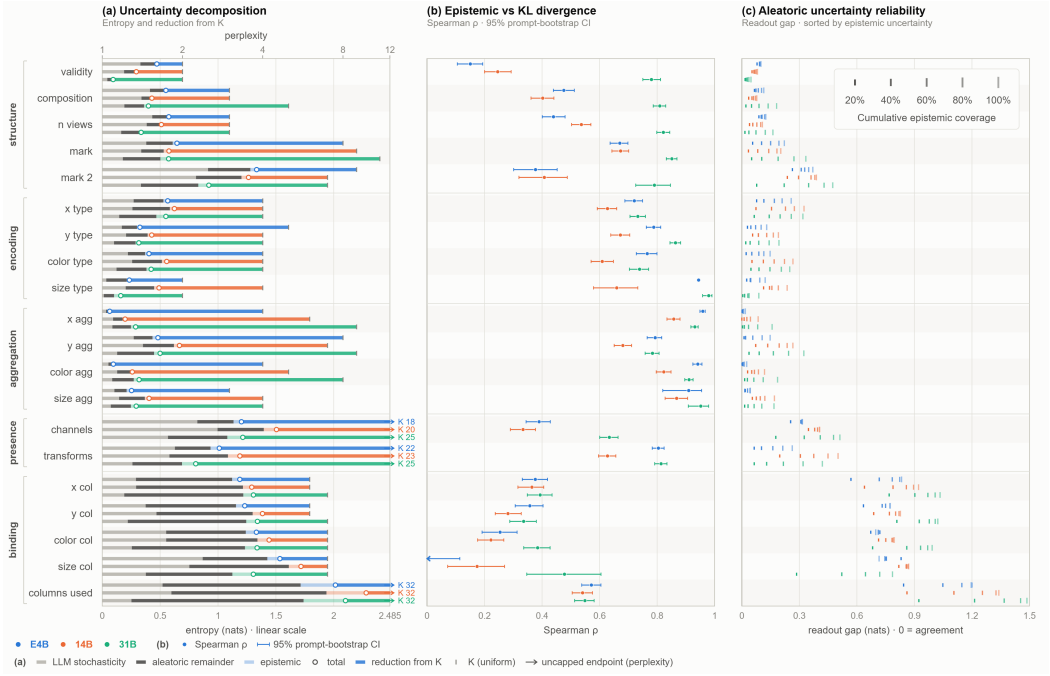


Figure 3: (a) Uncertainty decomposition for forecasters in nats. Each bar runs from the uniform reference of K options to the forecaster's predictive uncertainty (dot). This span is amount of uncertainty conditioning on h removes. Of what remains, the lighter grey segment is the empirical LLM stochasticity, which is irreducible floor. The darker grey segment is a readout gap, a difference between aleatoric uncertainty and empirical stochasticity. The light segment is the epistemic part, which more training data would close. (b) Correlation between epistemic uncertainty and the forecasters' predictive error (KL divergence). This pairing shows that epistemic uncertainty is indicative of forecasters' error, which in turn tells how reliable they are. (c) Agreement between aleatoric uncertainty and empirical LLM stochasticity as readout gap. The agreement is higher in test instances yielding smaller epistemic uncertainty, suggesting that predictions with small epistemic uncertainty also signal reliability of uncertainty decomposition.

LLM uncertainty, and as shown this is indeed the case. Combined, Figs. 3-(b) and (c) suggest that examples reported as low epistemic uncertainty lead to forecasted predictive distributions that are, both, more accurate, and in higher agreement with the spread of the LLM's distribution.

4.3 Visualization Analysis

In this section, we study how the uncertainty estimates are useful from a domain-specific perspective. Visualization/analytical requests differ in how much they constrain design. Some tasks are best done with a single design, such as a scatterplot for a correlation, while others leave several designs reasonable. If the forecaster's aleatoric uncertainty reflects the generator's genuine indecision, it should follow this openness. As every prompt in our corpus is written for one of 17 tasks, we can group the forecasts by tasks and ask: does aleatoric uncertainty agree with the generator's variability across different tasks?

Figure 4 illustrates that there is a variability across different design dimensions across tasks. Figure 4-(a–f) shows that the variability is strong across primary design decisions, some stronger than others. This signifies the usefulness of the aleatoric uncertainty we obtain from a forecaster: before any chart is generated, it grossly tells the uncertainty underlying any *type* of input. So uncertainty decomposition can be used to complement the predictive distribution for a single prompt: the predictive distribution foresees likely decisions by the LLM, while the uncertainty decomposition tells us how reliable that forecast may be. To further substantiate these claims, in Appendix E.4 we show that correlation holds across both data domain types and prompt characteristics.

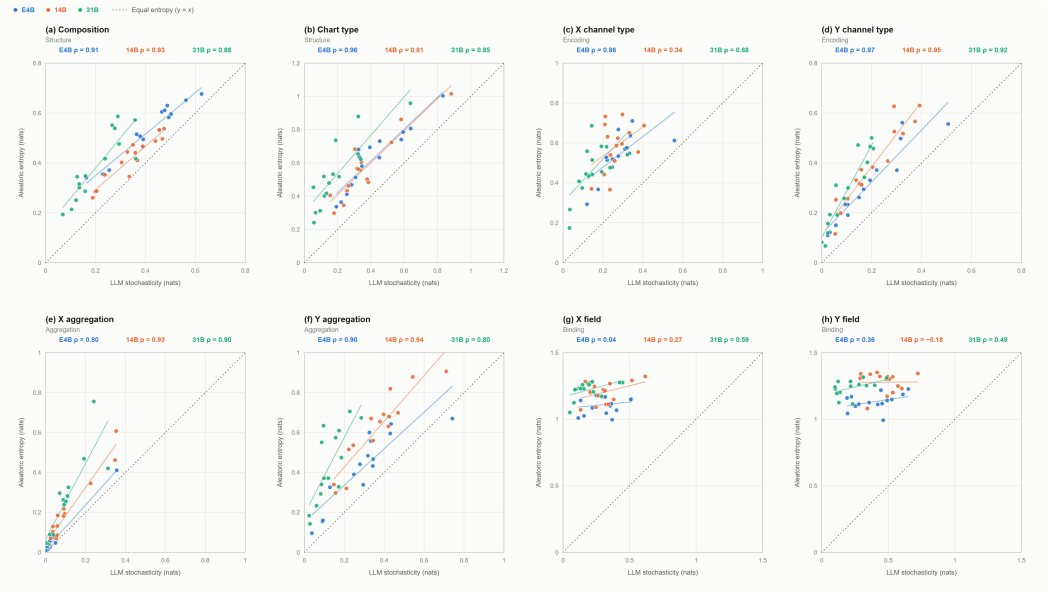


Figure 4: Each panel shows correlation between aleatoric uncertainty and empirically-estimated LLM stochasticity, per different task categories. Each sub-figure encodes three tested LLM models and 17 tasks per model. On design dimensions (a–f), this correlation is clearly observed. On more difficult choices, binding (g–h), the correlation is weaker. This result illustrates that there exists a high variability between different inputs, and that aleatoric uncertainty successfully signals inputs that would likely result in diverse outputs.

5 Conclusion

In this paper, we have shown that structured outputs generated by LLMs, namely LLM-authored visualizations, can be reliably forecasted via their hidden state representations. That is, a predictive distribution whose epistemic uncertainty is low leads to accurate predictions of individual design choices. Similarly, distributions whose epistemic uncertainty is low closely match the LLM's distribution. Although we have focused on visualization design, we note that our approach is generalizable to domains where (1) there exists resampling access to the generator that gives the conditional distribution, and (2) is is possible to abstract the output into a categorical distribution.

A limitation of our work is that we forecast marginal distributions, ignoring the fact that the output adhering to the Vega-Lite grammar gives rise to a joint distribution over design dimensions. A marginal forecast discards the dependencies in a draw. The cost is plausibly largest where our forecasts are weakest: binding is the most dependent task, since columns are exclusive resources across channels and graphical mark constrains encodings. A second limitation is in the hidden states themselves, as they are a lossy summary of the prompt. At different depths and tokens, hidden states could be handling crucial information for making specific decisions, whose effect we marginalize away. We hypothesize that the readout gap is largely impacted by the lossiness incurred in the LLM representation. We believe these limitations open up several avenues for future study—a sequence forecaster, e.g., transformer, could be used in place of marginal prediction to allow joint-distribution forecasting. Our findings also prompt further investigation into what constitutes the readout gap as a part of representation study.

6 Reproducibility Statement

We include full detail necessary for reproducing the result reported in this work. Information regarding dataset generation and processing is outlined in Appendix A and Appendix B. The subsequent LLM representation collection and processing is portrayed in Appendix C, which also includes forecaster fitting details. For reference metric computation, please refer to Appendix D.

7 AI DISCLOSURE

In this work, we used generative AI tools for synthesizing and engineering the large-scale dataset for model training, experiment implementation and modification, gain preliminary insights about our qualitative data. AI tools were used also to translate our figure idea into concrete code for figure generation, and summarize existing literature for identifying relevant work. We have not used AI tools for ideation of the paper. AI generated contents were constantly monitored, fixed, and verified. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Robert Amar, James Eagan, and John Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 111–117. IEEE, 2005.

Dhananjay Ashok and Jonathan May. Language models can predict their own behavior. *Advances in Neural Information Processing Systems*, 38:62708–62739, 2026.

Matthew Brehmer and Tamara Munzner. A multi-level typology of abstract visualization tasks. *IEEE transactions on visualization and computer graphics*, 19(12):2376–2385, 2013.

Iván Vicente Moreno Cencerrado, Arnau Padrés Masdemont, Anton Gonzalvez Hawthorne, David Demitri Africa, and Lorenzo Pacchiardi. No answer needed: Predicting llm answer accuracy from question-only linear probes. *arXiv preprint arXiv:2509.10625*, 2025.

Nan Chen, Yuge Zhang, Jiahang Xu, Kan Ren, and Yuqing Yang. Viseval: A benchmark for data visualization in the era of large language models. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):1301–1311, 2024.

Yuxi Chen, Junming Chen, Chenyu He, Yiwei Li, Yicheng Ji, Yifan Wu, Dingyu Yang, Lansong Diao, Lidan Shou, Hongliang Zhang, et al. Token economics for llm agents: A dual-view study from computing and economics. *arXiv preprint arXiv:2605.09104*, 2026.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning (ICML)*, 2018.

Zhichen Dong, Zhanhui Zhou, Zhixuan Liu, Chao Yang, and Chaochao Lu. Emergent response planning in LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Ce79P8ULPY.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630, 2024.

Siwei Fu, Kai Xiong, Xiaodong Ge, Siliang Tang, Wei Chen, and Yingcai Wu. Quda: Natural language queries for visual data analytics. *arXiv preprint arXiv:2005.03257*, 2020.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, et al. Verbalizable representations form a global workspace in language models. *arXiv preprint arXiv:2607.15495*, 2026.

Michael Hanna and Emmanuel Ameisen. Latent planning emerges with scale. *arXiv preprint arXiv:2604.12493*, 2026.

Matthew D. Hoffman and Andrew Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1593–1623, 2014.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*, 2024.

Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S Dhillon. Consistent binary classification with generalized performance metrics. *Advances in neural information processing systems*, 27, 2014.

János Kramár, Joshua Engels, Zheng Wang, Bilal Chughtai, Rohin Shah, Neel Nanda, and Arthur Conmy. Building production-ready probes for gemini. *arXiv preprint arXiv:2601.11516*, 2026.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*, 2023.

Tianqi Luo, Chuhan Huang, Leixian Shen, Boyan Li, Shuyu Shen, Wei Zeng, Nan Tang, and Yuyu Luo. nvbench 2.0: Resolving ambiguity in text-to-visualization through stepwise reasoning. *Advances in Neural Information Processing Systems*, 38, 2026.

Yuyu Luo, Jiawei Tang, and Guoliang Li. nvbench: A large-scale synthesized dataset for cross-domain natural language to visualization task. *arXiv preprint arXiv:2112.12926*, 2021.

Alex McKenzie, Urja Pawar, Phil Blandfort, William Bankes, David Krueger, Ekdeep S Lubana, and Dmitrii Krasheninnikov. Detecting high-stakes interactions with activation probes. *Advances in Neural Information Processing Systems*, 38:127556–127594, 2026.

Mohamed Amine Merzouk, Dmitri Carpov, Mirko Bronzi, Damiano Fornasiere, and Adam Oberman. How much is left? llms linearly encode their remaining output length. *arXiv preprint arXiv:2607.05316*, 2026.

Harikrishna Narasimhan, Rohit Vaish, and Shivani Agarwal. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. *Advances in neural information processing systems*, 27, 2014.

Radford M. Neal. MCMC using Hamiltonian dynamics. In Steve Brooks, Andrew Gelman, Galin L. Jones, and Xiao-Li Meng (eds.), *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC, 2011.

Koyena Pal, Jiuding Sun, Andrew Yuan, Byron C. Wallace, and David Bau. Future lens: Anticipating subsequent tokens from a single hidden state. In *Conference on Computational Natural Language Learning (CoNLL)*, 2023.

Julianna Piskorz, Katarzyna Kobalczyk, and Mihaela van der Schaar. Eliciting numerical predictive distributions of llms without autoregression. *arXiv preprint arXiv:2603.02913*, 2026.

Bahador Saket, Alex Endert, and Çağatay Demiralp. Task-based effectiveness of basic visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(7):2505–2512, 2019.

Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23 (1):341–350, 2017.

Arjun Srinivasan, Nikhila Nyapathy, Bongshin Lee, Steven M Drucker, and John Stasko. Collecting and characterizing natural language utterances for specifying data visualizations. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–10, 2021.

Tim Tomov, Dominik Fuchsgruber, Tom Wollschläger, and Stephan Günnemann. The illusion of certainty: Uncertainty quantification for llms fails under ambiguity. *arXiv preprint arXiv:2511.04418*, 2025.

Muhammed Ustaomeroglu, Baris Askin, Gauri Joshi, Carlee Joe-Wong, and Guannan Qu. Internal planning in language models: Characterizing horizon and branch awareness. In *International Conference on Learning Representations*, volume 2026, pp. 134732–134766, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2203.11171.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*, 2025.

Zhiqiu Xia, Jinxuan Xu, Yuqian Zhang, and Hang Liu. A survey of uncertainty estimation methods on large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 21381–21396, 2025.

Liyan Xu, Mo Yu, Fandong Meng, and Jie Zhou. How far ahead do llms plan? uncovering the latent horizon in chain-of-thought reasoning. *arXiv preprint arXiv:2602.02103*, 2026.

Yuheng Zhao, Junjie Wang, Linbing Xiang, Xiaowen Zhang, Zifei Guo, Cagatay Turkay, Yu Zhang, and Siming Chen. Lightva: Lightweight visual analytics with llm agent-based task planning and execution. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):6162–6177, 2024.