# BIRD: A Trustworthy Bayesian Inference Framework for LLMs

Yu Feng, Ben Zhou, Weidong Lin, Dan Roth

Presenters: Binyue Deng, Abinav Ram Chari

Georgia Tech

# Problem Statement

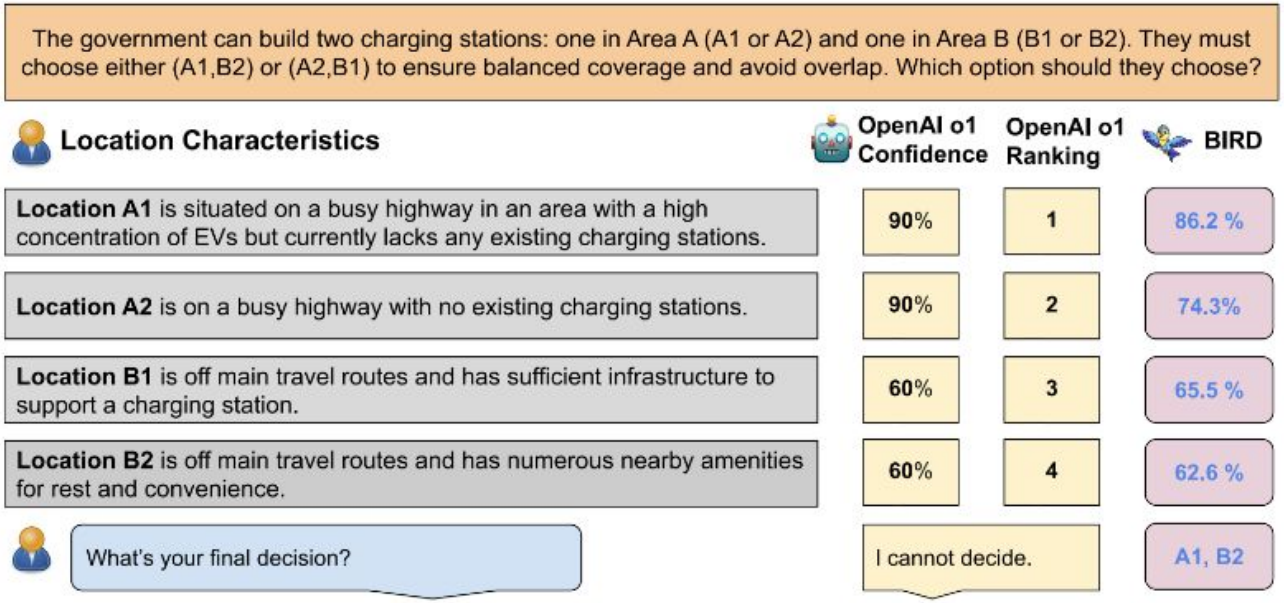
**If an LLM says something is “likely,” how much should we actually trust that number?**

Georgia Tech

# Problem Statement

Estimate a trustworthy probability under partial information: $P(O_i \mid C)$

- LLMs often output the same probability for different conditions, causing ties and unreliable downstream decisions




Georgia Tech

# Why Probability

Probabilities are needed for:

- Optimization under constraints (e.g., choose k locations under budget)
- Evidence strength comparison (avoid frequent ties)
- Uncertainty-aware actions (when to ask follow-up questions)

# Motivation: Why current LLMs fall short

Current LLMs are not ideal for probability estimation because:

- Numerical probabilities can be inaccurate / overconfident
- The estimation process is often not interpretable or controllable
  - LLMs can generate relevant factors + coarse probabilities
  - But cannot produce coherent calibrated probabilities directly

Goal of this paper:

- Infer better probabilities and expose an interpretable process

# Prior Works

**Direct inference in LLMs**

- **Chain-of-Thought prompting:** produces reasoning + verbalized probabilities, but does not enforce calibration/coherence
- **Self-consistency:** sampling multiple CoT paths improves reasoning stability

**Confidence / calibration elicitation**

- **Just ask for calibration:** prompting strategies for calibrated confidence scores
- **Can LLMs express their uncertainty?** empirical study of confidence elicitation

**Probabilistic reasoning with language**

- **Uncertain Natural Language Inference:** extracting uncertainty/probability in NLI settings
- **ThinkSum:** probabilistic reasoning over sets using LLMs
- **From word models to world models:** NL → probabilistic "world model" representations

# Background: Marginalization

When context is incomplete, inference marginalizes over unobserved “complete states” $f$

$$\mathbb{P}(O_i|C) = \sum_{f \in \mathcal{F}} \mathbb{P}(O_i|f) \prod_{j=1}^{N} \mathbb{P}(f_j|C)$$

- $f$ : full assignment of factor values
- This turns the problem into modeling $P(O_i \mid f)$ and estimating $P(f \mid C)$

# Background: Tractability via Conditional Independence

Factorize:

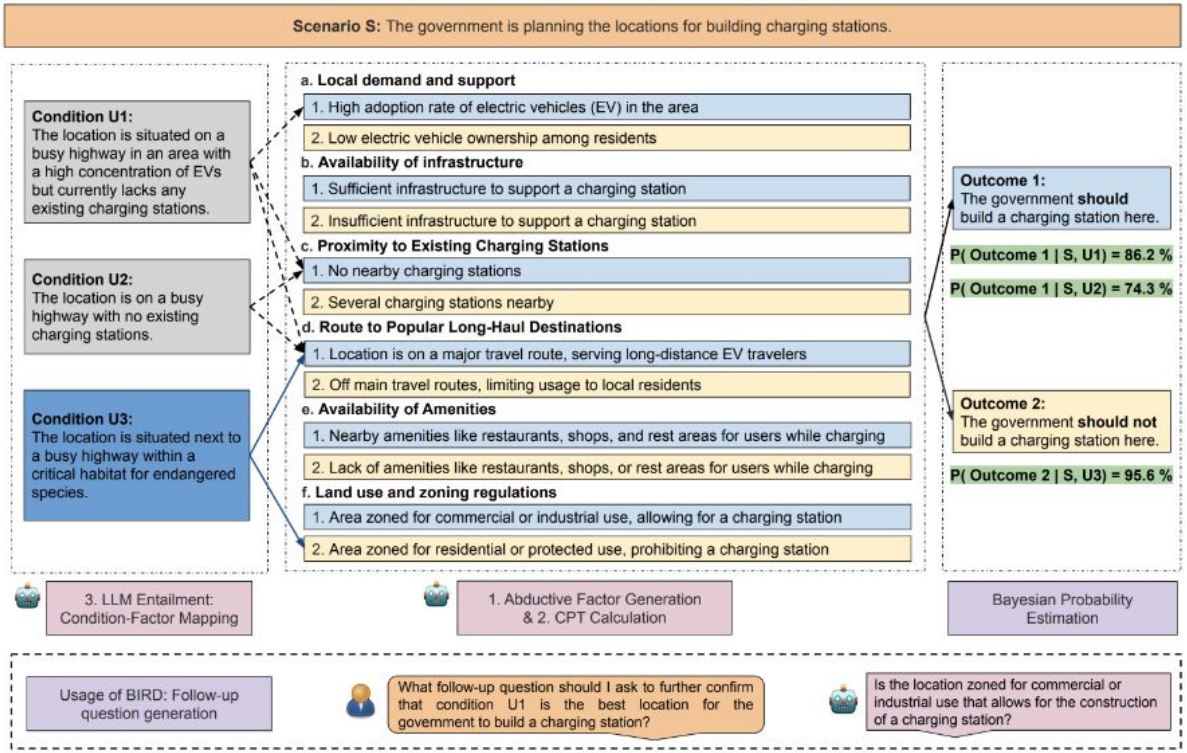
$$P(f \mid C) = \prod_j P(f_j \mid C)$$

- makes inference computable
- introduces a modeling assumption (correlated factors can be challenging)

# Method Overview

BIRD = Bayesian Inference from Abduction and Deduction

- Abduction: generate a comprehensive factor set, then learn Bayesian parameters
- Deduction: map specific conditions to factor values via entailment, then infer $P(O \mid C)$




Georgia Tech

# Method: Abductive Factor generation

Given a scenario $S$, LLM generates:

1. Additional Sentence Sampling
   - Generating sentences describing situations which would increase the likelihood of an outcome.
     - Alternative Exploration
     - Parametric Knowledge
     - Scenario Contextualization

**SYSTEM**
You are given a scenario and an accompanying hypothesis. Generate 5 sentences covering different conditions that would add objective information relevant to the hypothesis such that the hypothesis is more likely to hold true. The information should not definitively imply the hypothesis. You must follow the below structure to just generate sentences with no explanations.

**USER**
Scenario: You want to move around with your cell phone when it is being charged.
Hypothesis: You can move around more freely with your cell phone if it is being charged with a one-foot cord rather than a six-foot cord.

**ASSISTANT**
# The cell phone is being charged with a portable power bank located in your pocket, allowing you to move around without being tethered to a fixed outlet.
# The user is working in a compact space where longer cords could easily snag on furniture or equipment, thus a one-foot cord could minimize this risk.
# The phone is needed for tasks that require frequent handling and close proximity to the user, making a shorter cord more practical to avoid excessive dangling.
# The charging setup includes a small desktop charger that keeps the phone elevated and stable, limiting the practicality of a longer cord.
# The user is in a busy environment like a kitchen or workshop, where shorter cords can reduce the hazard of tripping or catching on moving objects.

Figure 5: Example **Additional Sentence Sampling** Prompt

# Method: Abductive Factor generation

Given a scenario $S$, LLM generates:

2. Factor Summarization
   - Summarizing sampled sentences into discrete factors with corresponding potential values
     - Feature Extraction
     - Discrete Variables
     - JSON Formatting

**EXAMPLE FACTOR SUMMARIZATION FROM ADDITIONAL SENTENCES PROMPT**

**SYSTEM**
From the given sentences for each outcome, identify and list distinct and concrete factors, ensuring each is broad yet specific and focuses on a unique aspect.
Your response should strictly adhere to the JSON format provided, without additional explanations.
For example: {"distinct factor" <ENSURE each factor focuses on a unique aspect>: "factor values" <Each factor MUST cover at least one condition to support the Statement and one condition to support the Opposite statement.>}
1. Ensure that each factor's value MUST directly reference specific elements mentioned in the statements, avoiding vague terms like 'the object'.
2. Ensure the factor values are not too concrete.
3. Do not only mention the common situations.

**USER**
Scenario: You are charging your cell phone and wish to move around with your cell phone.
Outcome 1: You can move around more freely with your cell phone if it is being charged with a six feet cord rather than a one feet cord.
Sentences:
#1 A longer cord provides more flexibility and allows for a greater range of movement while using the cell phone. This is because the additional length of the six-foot cord gives the user a larger radius of movement, enabling them to comfortably use their phone while it is charging without feeling restricted or confined to a specific location.
Outcome 2: You can move around more freely with your cell phone if it is being charged with a one-foot cord rather than a six-foot cord.
Sentences:
#2 If the cell phone is plugged into a portable power bank or a USB port on a computer, a one-foot cord provides greater mobility because it is shorter and less likely to get tangled or caught on objects while moving.
#3 If the cell phone is constantly being used while charging and the user prefers to keep the phone close to the charger at all times, a one-foot cord allows for easier mobility and reduces the risk of tripping over a longer cord."

**ASSISTANT**
{ "The cell phone's charging method":
["The charger is portable", "The charger is unmovable"],
"The user's movement range":
["The user stays very close to the charger", "The user has a large radius of movement"],
"The location of the phone charger":
["The user leaves the charger somewhere", "The user carries the charger"]}

Figure 6: Example **Factor Summarization from Additional Sentences** Prompt

# Method: Abductive Factor generation

Given a scenario $S$, LLM generates:

3. Factor Filtering & Validation
   - Assessing how each factor value affects the decision to eliminate irrelevant signals
     - Outcome Support Classification
     - Pruning
     - Neutral Intermediate Space

**EXAMPLE PROMPT FOR DECIDING THE VALUE BETTER SUPPORTS WHICH OUTCOME**

**SYSTEM**

Decide whether each of the given values supports Statement 1 or Statement 2. The statement and opposite statement are provided, along with a list of values. Determine which statement each value supports. You must follow the below structure to just generate json with an explanation.
For example:
Explanation: <explanations for how each value supports which statement>
List: { "value 1":<output only Statement 1, Statement 2, Both, or Neither>, "value 2":<output only Statement 1, Statement 2, Both, or Neither>}

**USER**

Here is a scenario: ... I'd like you to determine which statement each value supports.
Values: ...
* Statement 1: ...
* Statement 2: ...
Please analyze each value and determine which statement it supports.

**ASSISTANT**

Explanation: [step-by-step analysis here]
List: { "value":"Statement 1" }

Figure 7: Example Prompt for **Deciding the Value Better Supports Which Outcome**

Georgia Tech

# Method: Parameterize P(O | f) via Bayesian Approximation

Avoids exponential CPT growth by using logarithmic opinion pooling instead of a full joint conditional probability table:

- **Logarithmic Opinion Pool**: Instead of a full joint CPT, BIRD uses one learnable parameter $P(O_i \mid f_j)$ per factoper value

Then:

- **Combinatorial Sampling**: BIRD samples full assignments $f \in \mathcal{F}$ from the product space
- **Approximation Formula**: It computes an approximated $\mathbb{P}_{estimated}(O_i|f)$ by aggregating factor-level parameters via the derived log-opinion-pool formula (not a learned joint CPT)

Georgia Tech

# Method: CPT Optimization via Coarse LLM Probabilities

No ground-truth probabilities for $\mathbb{P}(O|f)$
Use LLM coarse estimates as a supervision signal:

- **Verbalized Supervision**: BIRD queries LLMs for coarse probability scores on sampled complete information instances.
- **Constrained Optimization**: Parameters are optimized by minimizing the MSE Loss between $P_{\text{LLM}}$ and $P_{\text{LLM}}(O_i|f)$
- **Directional Consistency**: A Margin Ranking Loss ensures the trained individual probabilities preserve the original support direction of the factors.

High-level benefit: Aligns LLM probability behavior with Bayesian constraints and gives better error tolerance

$P_{\text{estimated}}$

Georgia Tech

# Method: Deduction via Context-to-Factor Mapping

At inference time, each context $C$ is mapped to some factor values $fj$ via entailment

- **LLM Entailment**: Each context $C$ (Scenario $S$ + Condition $U$) is mapped to specific factor values $fj$ via an entailment classification
- **Observed Factors**: Factors directly implied by the context receive a high-confidence assignment
- **Unobserved Factors**: Factors not mentioned in the context are marginalized out under a uniform prior over their possible values
- **Context Sensitivity**: $\mathbb{P}(O|C)$ becomes highly sensitive to nuances because different conditions $U$ activate different sets of factor values

# Method: Bayesian Marginalization Logic

Bridging the gap between partial observations and accurate probability estimation:

- **Handling Incomplete Information**: In real-world scenarios, a condition often only triggers a subset of factors
- **Neutrality Principle**: For any unobserved factor $F_m$ , BIRD assumes an equal probability for all its possible values $f_{j_m}$ to remain unbiased
- **Marginalization Formula**: The final outcome probability is the weighted sum of probabilities across the entire information space $\mathcal{F}$:

$$\mathbb{P}(O_i|C) = \sum_{f \in \mathcal{F}} \mathbb{P}(O_i|f)\mathbb{P}(f|C)$$

- **Contextual Nuance**: By summing over all possibilities, the model captures fine-grained differences that direct LLM prompting typically misses

Georgia Tech

# Method: Final Inference & Trustworthy Outcomes

- Every probability is traceable to specific natural-language factors, providing a transparent "audit trail" for the decision
- By grounding inference in neutral intermediate factors, the framework filters out spurious correlations and inductive biases

# Evaluation

Two evaluation views:

- **Intrinsic**: probability alignment with human preferences
- **Extrinsic**: decision-making accuracy using predicted probabilities
- **Paper headline**: BIRD probabilities are more reliable and aligns with human preference judgments on ~30% more instances compared to direct LLM probabilities

# Experimental Setup

Benchmarking BIRD across diverse reasoning and planning domains:

- **Base Models:** Llama-2-70B and Llama-3.1-70B Instruct versions
- **Core Datasets**:
  - **COMMON2SENSE**: General commonsense reasoning (216 scenarios, 3,822 instances)
  - **PLASMA**: Procedural planning and counterfactual reasoning (279 scenarios, 1,395 instances)
  - **TODAY**: Complex temporal reasoning, identified as the most challenging benchmark
- **Human Gold Standard:** 350 instances annotated via Amazon Mechanical Turk with rigorous quality control
- **Training Hyperparameters**: Learning rate of 1e-02, 20 epochs, and a batch size of 4

# Intrinsic Experimental Setup

- **Setup**: Given scenario S, find two conditions supporting O1 over O2 (making O1 the chosen outcome).
- **Goal**: Find which of the two conditions **better** supports O1 over O2 by predicting conditional probabilities P(Oi|Ci)
- Rather than having two contexts which support different outcomes, this requires a much more fine-grained probability estimation.
- Ablations:
  - 1/2 Assumption: Instead of learning P(Oi|f), set P(Oi|f) = 1 if all factors support Oi or are neutral, else 0
  - 1/n Assumption: Set P(Oi|f) to the fraction of factors which support Oi.
  - Fixed Init: Coarsely assign P(Oi|f) by directing LLM to guess which outcome is more supported by f.

# Intrinsic Results: Human Preference Alignment

Testing BIRD's ability to distinguish subtle probability nuances ($C_1$ vs. $C_2$):

- BIRD (optimized) reaches ~0.59 Average F1 (Llama2 and Llama3.1).
- Baseline Gap: Vanilla and CoT methods show weak discrimination when predicting separate conditions
- Ablations (1/2, 1/n, fixed init) underperform optimized learning, supporting the CPT fitting step

| Model | Different$_1$ | Different$_2$ | Same | Average |
|---|---|---|---|---|
| Random Guessing | 0.333 | 0.333 | 0.333 | 0.333 |
| GPT3.5 CoT | 0.306 | 0.306 | 0.242 | 0.283 |
| GPT4 CoT | 0.312 | 0.357 | 0.216 | 0.289 |
| Llama2-70b Instruct Logit | 0.263 | 0.228 | 0.205 | 0.228 |
| Llama2-70b Instruct Vanilla | 0.375 | 0.333 | 0.243 | 0.311 |
| Llama2-70b Instruct CoT | 0.315 | 0.323 | 0.254 | 0.294 |
| Llama3.1-70b Instruct Logit | 0.300 | 0.282 | 0.242 | 0.269 |
| Llama3.1-70b Instruct Vanilla | 0.365 | 0.301 | 0.251 | 0.303 |
| Llama3.1-70b Instruct CoT | 0.373 | 0.351 | 0.187 | 0.309 |
| Llama2-70b Instruct EC* | 0.530 | 0.529 | 0.207 | 0.503 |
| Llama3.1-70b Instruct EC* | 0.535 | 0.538 | 0.286 | 0.511 |
| GPT4 EC* | 0.588 | 0.533 | 0.300 | 0.540 |
| Llama3.1 BIRD (*ablation w 1/2 assumption*) | 0.527 | 0.532 | 0.196 | 0.480 |
| Llama3.1 BIRD (*ablation w 1/n assumption*) | 0.572 | 0.584 | 0.272 | 0.532 |
| Llama3.1 BIRD (*ablation w fixed initial prob*) | 0.614 | 0.597 | 0.337 | 0.568 |
| Llama2 BIRD (*ours w optimized prob*) | **0.614** | 0.624 | **0.450** | **0.592** |
| Llama3.1 BIRD (*ours w optimized prob*) | 0.612 | **0.625** | 0.382 | 0.588 |

Georgia Tech

# Extrinsic: Decision making

Decision rule: pick argmax$\{P(O_1|C), P(O_2|C)\}$

- BIRD probabilities are accurate enough for direct decision-making.
- With Llama-3.1, BIRD outperforms CoT baseline across the listed benchmarks.
- Gains are strongest on the most challenging dataset (TODAY), where CoT tends to fail more.

| Dataset | BIRD_1/2 | BIRD_1/n | BIRD_fixed_prob | BIRD_optimized (ours) | CoT |
|---|---|---|---|---|---|
| *Llama-2-70b-instruct* | | | | | |
| TODAY | 73.7 | 72.8 | 73.4 | **73.9** | 71.5 |
| PLASMA | 72.8 | 72.3 | 72.7 | 74.0 | **76.8** |
| COMMON2SENSE | 86.9 | 86.8 | 87.5 | 89.0 | **93.8** |
| *Llama-3.1-70b-Instruct* | | | | | |
| TODAY | 65.5 | 68.9 | 65.5 | **74.3** | 72.6 |
| PLASMA | 71.3 | 66.5 | 65.7 | **73.0** | 71.5 |
| COMMON2SENSE | 78.1 | 85.4 | 86.7 | **92.3** | 90.8 |

Table 2: Performance comparisons of BIRD and baselines on decision-making benchmarks. BIRD_optimized is our final model; chain-of-thought with self-consistency (CoT) is our main baseline; others are ablation baselines.

Georgia Tech

# Extrinsic: Soft Supervision Application

BIRD as a "Probability Teacher" for model distillation

- **Soft Labels**: Continuous probabilities provide richer training signals than binary hard labels
- **Distillation Gain**: Fine-tuning T5-large with BIRD probabilities improves cross-domain average by 1.3%
- Generalization: Superior performance across temporal and planning tasks

| Model (Train Data) | TODAY (exp) | TODAY | TRACIE | MATRES | PLASMA | Average |
|---|---|---|---|---|---|---|
| T5_large | 77.8 | 57.9 | **73.0** | **83.5** | 48.1 | 65.6 |
| + hard label | 83.6 | 62.9 | 68.2 | 75.2 | **65.2** | 67.9 |
| + BIRD prob (*ours*) | **84.3** | **63.4** | 71.4 | 77.6 | 63.3 | **68.9** |
| PatternTime | 82.2 | 60.9 | **79.8** | **85.8** | 50.5 | 69.3 |
| + hard label | 83.9 | 61.8 | 76.2 | 83.7 | 56.9 | 69.7 |
| + BIRD prob (*ours*) | **85.1** | **62.6** | 75.7 | 85.1 | **61.5** | **71.2** |

Table 3: System performances under different supervision data across three binary temporal benchmarks and one binary planning benchmark. For simplicity, we use "hard label" representing that we use COMMON2SENSE supervision data with explicit binary labels, and "BIRD prob" representing that we use COMMON2SENSE supervision data with estimated probabilities. TODAY (exp) uses gold explanations during evaluation.

Georgia Tech

# Takeaways

- Direct LLM probabilities can be unreliable; ties make planning fail.
- BIRD: LLM factorization + Bayesian inference → more discriminative, interpretable probabilities.
- Improvements show up in intrinsic preference alignment and decision tasks.

# Reference


- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. NeurIPS 2022. https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html
- Wang, Xuezhi, et al. "Self-consistency improves chain of thought reasoning in language models." arXiv preprint arXiv:2203.11171 (2022). https://arxiv.org/abs/2203.11171
- Kadavath, Saurav, et al. "Language models (mostly) know what they know." arXiv preprint arXiv:2207.05221 (2022). https://arxiv.org/abs/2207.05221
- Lin, Stephanie, Jacob Hilton, and Owain Evans. "Teaching models to express their uncertainty in words." arXiv preprint arXiv:2205.14334 (2022). https://arxiv.org/abs/2205.14334
- Jiang, Zhengbao, et al. "How can we know what language models know?." Transactions of the Association for Computational Linguistics 8 (2020): 423-438. https://direct.mit.edu/tacl/article-abstract/doi/10.1162/tacl_a_00324/96460
- Ozturkler, Batu, et al. "Thinksum: Probabilistic reasoning over sets using large language models." Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2023. https://aclanthology.org/2023.acl-long.68/
- Andreas, Jacob. "Language models as agent models." Findings of the Association for Computational Linguistics: EMNLP 2022. 2022. https://aclanthology.org/2022.findings-emnlp.423/


Georgia Tech