

IDEA: An Interpretable and Editable Decision-Making Framework for LLMs via Verbal-to-Numeric Calibration

Yanji He¹, Yuxin Jiang², Yiwen Wu¹, Bo Huang¹, Jiaheng Wei^{*1}, Wei Wang^{*1,3}

¹The Hong Kong University of Science and Technology (Guangzhou)

²Huawei Technologies Co.,Ltd, ³The Hong Kong University of Science and Technology
{yhe720, ywu240}@connect.hkust-gz.edu.cn, jiang.yuxin2@huawei.com,
bhuangas@connect.ust.hk, jiahengwei@hkust-gz.edu.cn, weiwcs@ust.hk

Abstract

Large Language Models are increasingly deployed for decision-making, yet their adoption in high-stakes domains remains limited by miscalibrated probabilities, unfaithful explanations, and inability to incorporate expert knowledge precisely. We propose **IDEA**, a framework that extracts LLM decision knowledge into an interpretable parametric model over semantically meaningful factors. Through joint learning of verbal-to-numerical mappings and decision parameters via EM, correlated sampling that preserves factor dependencies, and direct parameter editing with mathematical guarantees, IDEA produces calibrated probabilities while enabling quantitative human-AI collaboration. Experiments across five datasets show IDEA with Qwen-3-32B (78.6%) outperforms DeepSeek R1 (68.1%) and GPT-5.2 (77.9%), achieving perfect factor exclusion and exact calibration—precision unattainable through prompting alone. The implementation is publicly available.¹

1 Introduction

Large Language Models (LLMs) are increasingly deployed for automated decision making (Yang et al., 2023a), yet their use in high-stakes domains like financial investment remains limited by a fundamental trust deficit: stakeholders cannot reliably verify, audit, or intervene in the decision process. This deficit stems from three challenges. First, real-world decisions demand calibrated probabilities, yet LLMs produce overconfident and inaccurate estimates under uncertainty (Xiong et al., 2024). Second, stakeholders require faithful explanations, but generated rationales often serve as post-hoc rationalizations rather than reflecting actual reasoning (Turpin et al., 2023; Lanham et al., 2023). Third, current systems lack quantitative

^{*}Corresponding author.

¹Source code is available at <https://github.com/leonbig/IDEA>.

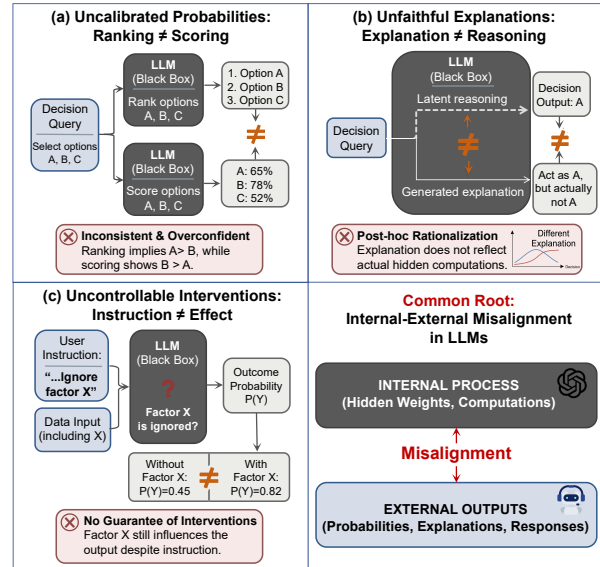


Figure 1: Three manifestations of the trust deficit in LLM-based decision making, stemming from a common root: internal-external misalignment. (a) **Uncalibrated probabilities**: ranking and scoring yield inconsistent orderings for the same options. (b) **Unfaithful explanations**: generated rationales diverge from the latent reasoning that actually determines outputs. (c) **Uncontrollable interventions**: natural language instructions fail to guarantee behavioral compliance—factor X continues to influence predictions despite explicit exclusion instructions.

frameworks to integrate domain expert insights beyond simple prompting (Figure 1).

Existing approaches fail to resolve these issues. **Logit-based** methods extract probabilities from output distributions (Kadavath et al., 2022; Tian et al., 2023), but conflate next-token confidence with decision uncertainty and remain black-box transformations. **Self-explanation** approaches like chain-of-thought prompting (Wei et al., 2022) and self-consistency decoding (Wang et al., 2023) produce readable justifications but demonstrably lack faithfulness. **Factor-based** frameworks decompose decisions through intermediate variables: DeLLMa

(Liu* et al., 2024) uses decision-theoretic structures with direct numerical elicitation, and BIRD (Feng et al., 2025) elicits verbal probabilities over factors. However, DeLLMa relies on LLMs to produce precise numerical utilities (precisely the capability we identify as unreliable) and BIRD assumes factor independence with fixed verbal-to-numerical mappings from prior literature, losing both calibration accuracy (our w/o EM ablation shows -9.6% avg F1) and natural factor correlations. Neither framework supports quantitative parameter editing with mathematical guarantees. These limitations reflect a deeper misalignment between LLMs’ internal computations and their external outputs.

We pursue a different path grounded in two observations (validated in Section 4.2): (i) while LLMs cannot reliably produce precise numerical probabilities, they can generate decision-relevant factors from their broad knowledge; and (ii) LLMs exhibit greater consistency when producing verbal probability expressions (e.g., "likely," "unlikely") than exact numeric estimates—a consequence of training on human text where such phrases are abundant while precise probabilities are rare.

Rather than making the internal inference process of LLMs transparent, we aim to extract their knowledge into a form that is inherently transparent. Specifically, we propose **IDEA** (An **I**nterpretable and **E**ditable **D**ecision-Making Framework) by introducing an intermediate representation: a set of semantically meaningful decision factors $\mathcal{F} = \{F_1, \dots, F_N\}$. Given a decision query Q comprising a scenario S (e.g., loan approval) and specific conditions C (e.g., applicant information), our goal is to estimate $\mathbb{P}(O_i | Q)$ for each possible outcome O_i . Rather than eliciting this probability directly, we assume these factors are jointly sufficient to determine the decision outcome. Under this assumption, the target probability admits a decomposition:

$$\mathbb{P}(O_i | Q) = \sum_{\mathbf{f} \in \mathcal{F}^*} \mathbb{P}(O_i | \mathbf{f}) \cdot \mathbb{P}(\mathbf{f} | Q), \quad (1)$$

where \mathcal{F}^* denotes the space of all factor value assignments (factor configurations), and \mathbf{f} denotes a specific assignment. This decomposition separates two components: the *decision model* $\mathbb{P}(O_i | \mathbf{f})$, which maps factor configurations to outcome probabilities, and the *factor inference* $\mathbb{P}(\mathbf{f} | Q) = \mathbb{P}(\mathbf{f} | C)$, which determine the factor values from the entire query, that is, from the specific condition. The key insight is that the decision model $\mathbb{P}(O_i | \mathbf{f})$ operates over a low-dimensional, semantically struc-

ture space. This enables learning an interpretable model whose parameters directly quantify each factor’s contribution, allowing inspection, verification, and modification of the decision process.

Two technical challenges arise that distinguish IDEA from prior factor-based approaches. First, verbal probability expressions (e.g., "very unlikely," "likely") are inherently ambiguous with unknown numerical mappings, yet learning the decision model requires numerical labels. Unlike BIRD (Feng et al., 2025), which fixes mappings from psychological literature, we resolve this through joint estimation of the verbal-to-numerical mapping and model parameters via EM, treating unknown probabilities as latent variables. This addresses a fundamental circularity: learning the decision model requires numerical labels, but determining what numerical values verbal expressions represent requires the decision model. Second, inference under partial information requires marginalizing over uncertain factors while preserving their dependencies. Unlike methods assuming conditional independence (Feng et al., 2025), we sample joint configurations directly from the LLM, maintaining natural correlations (e.g., high income with stable employment).

The resulting framework offers three properties that directly address the trust deficit:

- **Calibrated Probability Estimation.** The framework produces well-calibrated probabilities through joint learning of verbal-to-numerical mappings and decision model parameters.
- **Semantic Interpretability.** Each parameter directly quantifies a specific factor’s contribution, enabling domain experts to inspect and contest the decision logic.
- **Quantitative Human-AI Collaboration.** Users can edit parameters to incorporate domain knowledge or enforce constraints with mathematically precise effects—unattainable through prompting alone.

2 Related Work

Probability Elicitation and Calibration. Prior work extracts probabilities via token logits (Kadavath et al., 2022) or verbalized confidence (Tian et al., 2023), but both exhibit poor calibration (Liu et al., 2025). BIRD (Feng et al., 2025)

elicits verbal probabilities over factors but assumes independence and fixes verbal mappings. Bayesian approaches (Gal and Ghahramani, 2016; Kuhn et al., 2023) remain computationally prohibitive for LLMs, and ensemble methods like self-consistency (Wang et al., 2023) treat decisions as black boxes—recent analysis further shows that sampling more reasoning paths can compound rather than reduce errors (Jiang et al., 2026). We jointly learn calibrated mappings while preserving factor dependencies and enabling parameter inspection.

Interpretable Models and Explanations. Concept Bottleneck Models (CBMs) route predictions through concept layers (Koh et al., 2020), with variants addressing concept uncertainty (Kim et al., 2023), annotation costs (Oikarinen et al., 2023; Yang et al., 2023b), and LLM integration (Sun et al., 2025). However, these require task-specific training or assume concept independence. Post-hoc methods like SHAP (Lundberg and Lee, 2017) explain predictions but cannot modify behavior. Other lines audit CoT faithfulness (Sun et al., 2026) or steer reasoning via internal neurons (Dong et al., 2026), and parameter-efficient methods such as AuroRA (Dong et al., 2025) adapt models through opaque weight updates rather than semantic factors. Our framework provides CBM-style interpretability while uniquely enabling direct parameter editing with mathematical guarantees.

3 Framework Overview

We now formalize the problem and present our framework (illustrated in Figure 2). The general formulation in Equation (1) accommodates arbitrary outcome spaces and factor domains. In this work, we focus on **binary decisions** with **binary factors**: the outcome space is $\mathcal{O} = \{0, 1\}$, and each factor $F_j \in \{0, 1\}$ indicates whether that factor’s value is positive. Our goal is to estimate $\mathbb{P}(O = 1 \mid Q)$ for a query Q comprising a scenario S and a specific condition C .

3.1 Technical Objectives and Assumptions

Under the binary setting, the factor configuration space becomes $\mathcal{F}^* = \{0, 1\}^N$. Our objectives are:

- **Learn the decision model** $\mathbb{P}(O_i \mid \mathbf{f})$ as an interpretable parametric model $P_\theta(O_i \mid \mathbf{f})$ from LLM behavioral data.

- **Perform factor inference** $\mathbb{P}(\mathbf{f} \mid C)$ that determine factor values from specific conditions.

Finally, compute the final probability via Equation (1). Our framework relies on some key assumptions:

Assumption 1 (Factor Completeness) *The identified factors \mathcal{F} are jointly sufficient to determine the decision outcome: $\mathbb{P}(O_i \mid C, S) = \mathbb{P}(O_i \mid \mathbf{f})$, where \mathbf{f} is the factor configuration determining the values of factors by specific condition C .*

Assumption 1 ensures no decision-relevant information is lost in the factor representation.

Assumption 2 (Verbal Probability Consistency) *The LLM’s verbal probability responses reflect an underlying numerical probability through a consistent, monotonic mapping.*

While the exact mapping is unknown, Assumption 2 assumes it can be learned from behavioral data. We empirically validate this monotonic-consistency requirement in Appendix G, finding 87.6% ordinal agreement across 7680 configuration pairs.

3.2 Framework Pipeline

As depicted in Figure 2, our framework extracts LLM decision knowledge into an interpretable parametric model over a semantically meaningful factor space. The methodology comprises five components organized into two phases.

Training Phase (Offline). Given a decision query $Q = (S, C)$: 1) *Factor Identification*: Construct binary factors set $\mathcal{F} = \{F_1, \dots, F_N\}$ and verify their discriminability and coverage to satisfy Assumption 1. 2) *Behavioral Probing*: Query the LLM across plenty of factor configurations to collect verbal probability responses, yielding dataset $\mathcal{D} = \{(\mathbf{f}^{(k)}, V^{(k)})\}_{k=1}^K$. 3) *Joint Estimation*: Apply Expectation-Maximization to simultaneously learn the decision model $P_\theta(O \mid \mathbf{f})$ and the verbal-to-numerical mapping $\phi : \mathcal{V} \rightarrow [0, 1]$ on the dataset \mathcal{D} .

Inference Phase (Online). 1) *Factor Determination*: Extract observable factor values from conditions C , partitioning factors into observed (\mathcal{O}) and uncertain (\mathcal{U}) subsets. 2) *Marginalization*: Sample coherent completions for uncertain factors to preserve dependencies, then compute the final probability through Monte Carlo marginalization over the learned decision model.

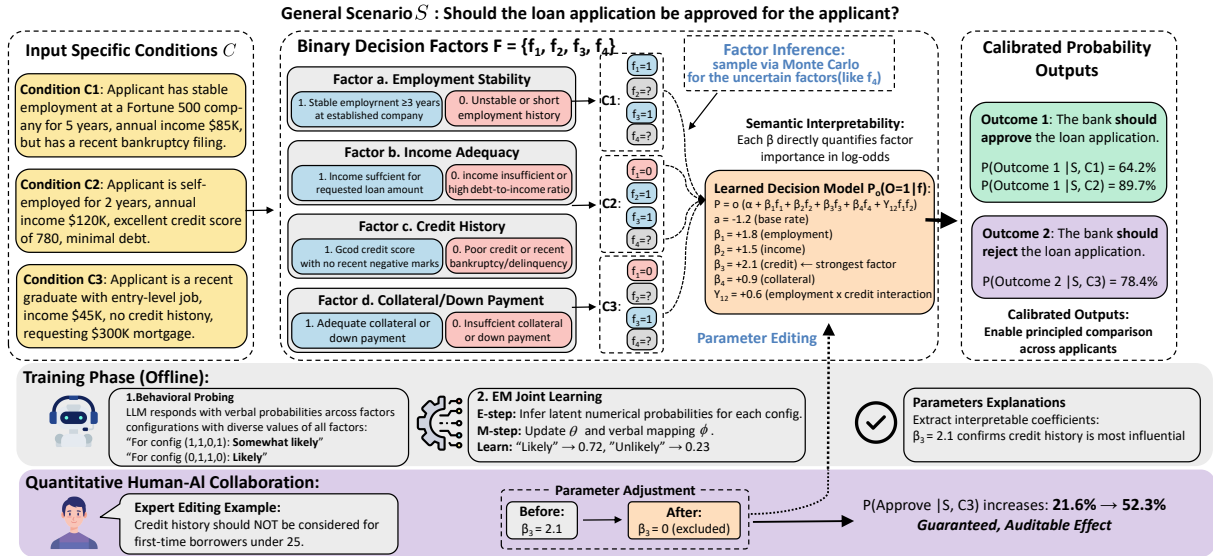


Figure 2: The IDEA framework illustrated on a loan approval task. Given applicant conditions C , the framework extracts binary factor values, handles uncertainty via Monte Carlo sampling, and produces calibrated probabilities through a learned decision model $P_\theta(O = 1 | \mathbf{f})$ with interpretable coefficients. The offline phase (bottom left) jointly learns model parameters and verbal-to-numerical mappings via EM. Expert intervention (bottom right) enables precise, auditable parameter edits—e.g., excluding credit history by setting $\beta_3 = 0$ shifts approval probability from 21.6% to 52.3%.

Expert Intervention. The framework additionally supports quantitative human-AI collaboration through direct parameter manipulation, enabling precise editing of relative factor importance with mathematically guaranteed effects.

4 Methodology

4.1 Factor Identification and Verification

For a decision scenario S with outcomes $\mathcal{O} = \{O_1, O_2\}$, we construct a binary factor set $\mathcal{F} = \{F_1, \dots, F_N\}$ through statement-driven elicitation rather than direct generating. To encourage comprehensive exploration of the decision space, we prompt the LLM to generate 20 comprehensive situational descriptions that would cause an outcome to occur. From these generated statements, we summarize candidate factors and formulated as binary values where $F_j = 1$ and $F_j = 0$ encodes whether a semantically distinct aspect is positive or negative. The factor set undergoes iterative verification by LLMs against two criteria: (i) *discriminability*, each factor’s values must differentially support distinct outcomes, and (ii) *coverage*, the factors must jointly subsume all decision-relevant information in the specific condition C . Factors failing discriminability are reformulated or discarded; unmapped information units in C trigger factor expansion. This loop iterates until convergence, yielding a fac-

tor set satisfying Assumption 1.

4.2 Behavioral Probing

With a verified factor set \mathcal{F} , we now collect behavioral data that reveals how the LLM maps factor configurations to outcome likelihoods. This dataset forms the empirical foundation for learning the decision model in Section 4.3. We construct the training dataset by systematically querying the LLM across the factor configuration space $\mathcal{F}^* = \{0, 1\}^N$. For tractable factor spaces ($2^N \leq 256$), we probe exhaustively. For larger spaces, we employ uniform random sampling of 256 configurations, which provides sufficient coverage for estimating main effects and salient interactions.

Verbal Probability Elicitation. For each configuration $\mathbf{f}^{(k)}$, we construct a hypothetical scenario and query the LLM to assess outcome likelihood. Following (Feng et al., 2025), we adopt a seven-level ordinal scale: $\mathcal{V} = \{\text{very unlikely}, \text{unlikely}, \text{somewhat unlikely}, \text{neutral}, \text{somewhat likely}, \text{likely}, \text{very likely}\}$, ordered from v_1 to v_7 by increasing likelihood. This process yields a behavioral dataset $\mathcal{D} = \{(\mathbf{f}^{(k)}, V^{(k)})\}_{k=1}^K$, where $V^{(k)} \in \mathcal{V}$ is the LLM’s verbal response to configuration $\mathbf{f}^{(k)}$.

4.3 Joint Estimation via EM Algorithm

The behavioral probing procedure yields a dataset $\mathcal{D} = \{(\mathbf{f}^{(k)}, V^{(k)})\}_{k=1}^K$, where each observation pairs a factor configuration $\mathbf{f}^{(k)} \in \{0, 1\}^N$ with the LLM’s verbal probability response $V^{(k)} \in \mathcal{V}$. Our goal is to learn two components simultaneously:

- A **decision model** $P_\theta(O = 1 \mid \mathbf{f})$ that maps factor configurations to numerical probabilities, parameterized by θ .
- A **verbal-to-numerical mapping** $\phi : \mathcal{V} \rightarrow [0, 1]$ that converts verbal expressions to calibrated probabilities.

This presents a "chicken-and-egg problem": if the mapping ϕ were known, we could directly regress model predictions against numerical targets $\phi(V^{(k)})$, and vice versa.

We therefore employ the Expectation-Maximization (EM) algorithm, which iteratively computes the posterior distribution of latent probabilities (E-step) and updates parameters to maximize the expected complete-data likelihood (M-step). EM alternation guarantees monotonic improvement in the marginal likelihood and convergence to a local optimum.

4.3.1 Model Specification

We instantiate the decision model as a logistic regression with main effects and pairwise interactions. Recall that a factor configuration $\mathbf{f} = (f_1, \dots, f_N) \in \{0, 1\}^N$ assigns a binary value to each factor, where $f_j \in \{0, 1\}$ denotes the value of factor F_j . The decision model takes the form:

$$P_\theta(O = 1 \mid \mathbf{f}) = \sigma \left(\alpha + \sum_{j=1}^N \beta_j f_j + \sum_{1 \leq i < j \leq N} \gamma_{ij} f_i f_j \right), \quad (2)$$

where $\sigma(\cdot)$ is the sigmoid function, α is the intercept, β_j quantifies the main effect of factor F_j , and γ_{ij} captures the interaction between factors F_i and F_j . The full parameter vector is $\theta = (\alpha, \beta, \gamma)$ with $\beta = (\beta_1, \dots, \beta_N)$ and $\gamma = \{\gamma_{ij}\}_{i < j}$.

This functional form balances expressiveness with interpretability: main effects $\{\beta_j\}$ directly quantify each factor’s contribution in log-odds, while interactions $\{\gamma_{ij}\}$ capture synergies or conflicts between factor pairs. To promote sparse, interpretable interaction structure, we apply elastic net regularization exclusively to the interaction coefficients: $\Omega(\gamma) = \lambda_1 \|\gamma\|_1 + \lambda_2 \|\gamma\|_2^2$. The ℓ_1 penalty induces sparsity, driving negligible interactions to

exactly zero; the ℓ_2 penalty ensures numerical stability when interactions are correlated. Main effects $\{\beta_j\}$ and intercept α remain unregularized to preserve their direct interpretability.

4.3.2 Estimation Algorithm

We formulate the estimation problem by introducing latent variables representing the unknown numerical probabilities. For each observation $(\mathbf{f}^{(k)}, V^{(k)})$, let $p^{(k)} \in [0, 1]$ denote the latent probability that the LLM’s verbal response $V^{(k)}$ implicitly represents. These latent variables bridge verbal expressions and numerical targets, enabling joint estimation via the Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

E-step. Given current estimates $(\theta^{(t)}, \phi^{(t)})$, we compute the posterior distribution of each latent probability $p^{(k)}$. Two sources of information constrain this distribution: the model prediction $P_\theta(O = 1 \mid \mathbf{f}^{(k)})$ and the verbal indication $\phi(V^{(k)})$. Under a Gaussian likelihood model with precision parameters τ_θ and τ_ϕ , the posterior mean takes the form of a precision-weighted combination:

$$\mathbb{E}[p^{(k)} \mid \mathbf{f}^{(k)}, V^{(k)}] = \frac{\tau_\theta \cdot P_\theta(\mathbf{f}^{(k)}) + \tau_\phi \cdot \phi(V^{(k)})}{\tau_\theta + \tau_\phi}. \quad (3)$$

M-step. We update both components to maximize the expected complete-data log-likelihood. The model parameters θ are updated by minimizing a composite objective:

$$\mathcal{L}(\theta) = \sum_k (P_\theta(\mathbf{f}^{(k)}) - \mathbb{E}[p^{(k)}])^2 + \lambda \cdot \mathcal{L}_{\text{rank}}(\theta) + \Omega(\gamma), \quad (4)$$

where $\mathcal{L}_{\text{rank}}$ is a directional hinge consistency loss (Burgess et al., 2005) that penalizes violations of the ordinal structure (i.e., configurations eliciting "likely" should yield higher probabilities than those eliciting "unlikely"). The verbal mapping ϕ is updated by minimizing the squared deviation from posterior means, subject to monotonicity constraints that preserve semantic ordering (Wallsten et al., 1986). We initialize ϕ with canonical mappings from psychological literature (Budescu et al., 2009). Iteration terminates when the Q-function (expected complete-data log-likelihood) change falls below 10^{-4} . Full derivations appear in Appendix A.

4.4 Inference Under Uncertainty

Given a new query $Q = (S, C)$, inference follows the decomposition in Equation (1) through three

steps.

Factor Determination. For each factor F_j , we query the LLM to decide whether specific condition C implies: $F_j = 1$, $F_j = 0$, or does not determine F_j . This partitions factors into observed factors \mathcal{O} (with determined values $\mathbf{f}_{\mathcal{O}}$) and uncertain factors \mathcal{U} . A human evaluation on 150 instances confirms this partition is reliable (91.5% accuracy on the observed-vs-uncertain distinction for Qwen-3-32B; see Appendix H).

Joint Sampling. When $|\mathcal{U}| > 0$, we must estimate the joint distribution over uncertain factors. Unlike independence assumptions in prior work (Feng et al., 2025), we preserve natural factor correlations (e.g., high income correlating with stable employment) by sampling T completions from the LLM conditioned on $\mathbf{f}_{\mathcal{O}}$ and C . High temperature ($\tau \geq 1.0$) ensures sample diversity.

Marginalization. The final probability is computed via Monte Carlo marginalization:

$$P(O = 1 | Q) \approx \frac{1}{T} \sum_{t=1}^T P_{\theta} \left(O = 1 | \mathbf{f}_{\mathcal{O}}, \mathbf{f}_{\mathcal{U}}^{(t)} \right). \quad (5)$$

This estimator is unbiased, with standard error $O(1/\sqrt{T})$. In our experiments, we use $T = 50$ samples, which provides stable estimates while remaining computationally efficient.

4.5 User Parameter Editing

The decision model’s transparent structure enables expert intervention impossible through prompting. We support two modes: structural edits (adding or removing factors) and quantitative edits to factor influence. Users can directly add or remove factors during training. Because the model operates over the explicit factor space \mathcal{F} rather than opaque internal activations, such modifications are auditable guaranteed.

Average Marginal Effects. Experts often need to adjust the relative importance of factors quantitatively. However, logistic coefficients operate in log-odds space and do not directly represent probability changes. To express constraints in probability space, we employ the **Average Marginal Effect (AME)**—the expected probability change when a factor switches from 0 to 1, averaged over all con-

figurations of other factors:

$$\text{AME}_k = \frac{1}{2^{N-1}} \sum_{\mathbf{x}_{-k} \in \{0,1\}^{N-1}} \Delta_k(\mathbf{x}_{-k}), \quad (6)$$

where $\Delta_k(\mathbf{x}_{-k}) = P_{\theta}(O = 1 | f_k = 1, \mathbf{x}_{-k}) - P_{\theta}(O = 1 | f_k = 0, \mathbf{x}_{-k})$ and \mathbf{x}_{-k} denotes all factor assignments except k . AME values are directly interpretable as average percentage-point changes in outcome probability. AME is widely employed to interpret parameter importance in logistic regression, as it bridges the gap between abstract statistical coefficients and human cognitive intuition (Mood, 2010).

Constrained Adjustment. To enforce expert constraints such as relative importance (e.g., $\text{AME}_2 = \rho \cdot \text{AME}_1$) while minimizing unintended side effects, we solve:

$$\begin{aligned} \min_{\theta'} & \sum_{j \in \{1, \dots, N\} \setminus \{1, 2\}} (\text{AME}_j(\theta') - \text{AME}_j(\theta))^2 \\ \text{s.t.} & \mathbb{E}[z_{\theta'}] = \mathbb{E}[z_{\theta}], \\ & \text{AME}_2(\theta') = \rho \cdot \text{AME}_1(\theta'), \end{aligned} \quad (7)$$

where $z_{\theta} = \alpha + \sum_j \beta_j f_j + \sum_{i < j} \gamma_{ij} f_i f_j$ is the logit under θ , and expectations are taken over $\{0, 1\}^N$ (or a domain-weighted distribution when base rates are known). Since $\text{AME}(\theta)$ is differentiable with respect to θ , we solve this with Sequential Quadratic Programming. The resulting interventions are *verifiable* (constraints can be checked exactly), *predictable* (other AMEs are minimally perturbed), and *reversible* (the original θ can be restored).

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate on five datasets across two categories. *Complex Decisions*: BIG-DATA22 (Soun et al., 2022) (1,472 instances) predicts stock movements from market signals and tweets; Statlog German Credit (Hofmann, 2023) (1,000 instances) assesses loan applicant creditworthiness. *Reasoning*: COMMON2SENSE (Singh et al., 2021) (3,672 instances) tests commonsense reasoning; PLASMA (Brahman et al., 2023) (1,178 instances) evaluates procedural planning; TODAY (Feng et al., 2023) (895 instances) assesses temporal reasoning. Following (Feng et al., 2025), all instances are converted to binary decision queries.

Models and Baselines. We use Qwen-3-4B/8B/32B (Team, 2025). Baselines include: (1) *Ablations—w/o Inter* removes interaction terms ($\gamma_{ij} = 0$), testing whether factor interactions capture meaningful decision patterns; *w/o EM* uses fixed verbal-to-numerical mappings from prior literature (Budescu et al., 2009), bypassing joint calibration; *w/o MC* deterministically assigns uncertain factors (randomly with $p = 0.5$ when uninformed), testing whether correlated sampling preserves important factor dependencies. (2) *Existing methods—Logit* extracts probabilities from token logits; *Vanilla* (Wang et al., 2023) directly elicits verbalized probabilities; *CoT* (Wei et al., 2022) applies chain-of-thought before verbalization; *PWC* performs pairwise comparison without explicit probabilities (Experiment II only, where its comparative framing provides structural advantage). Hyperparameters are in the Appendix L.

Experiment I: Direct Decision-Making. We evaluate the fundamental task of selecting the correct outcome. Given query $Q = (S, C)$, methods predict $\arg \max_{O_i} P(O_i | Q)$. We report classification accuracy.

Experiment II: Fine-Grained Probability Reliability. Binary accuracy alone is insufficient for reliable decision support, stakeholders also need probability estimates that faithfully reflect evidential strength. This experiment tests whether our framework produces calibrated probabilities that correctly rank queries by their relative support for an outcome. We construct paired queries $Q_1 = (S, C_1)$ and $Q_2 = (S, C_2)$ where both conditions favor O_1 , but C_1 provides stronger support. A calibrated model should satisfy $P(O_1 | Q_1) > P(O_1 | Q_2) > 0.5 > P(O_2 | Q_2) > P(O_2 | Q_1)$. Ground truth derives from pairwise human annotations on 500 COMMON2SENSE instances. We report macro F1 over three-way classification (C_1 stronger, C_2 stronger, or equal).

Experiment III: User-edited Factor Intervention. We evaluate whether our framework enables precise human-AI collaboration through (1) **factor exclusion** and (2) **quantitative calibration**. We test whether a factor’s influence can be completely removed. Using a 2×2 design, we vary the presence of exclusion instructions and the target factor F_X ’s value while holding other factors constant. Our method sets $\beta_X = 0$ and removes all interac-

tion terms involving F_X ; baselines receive clear natural language instructions. We report (1) **Effect Reduction Ratio (ERR)**: $1 - \frac{\Delta P_{w/}}{\Delta P_{w/o}}$, where $\Delta P = P(O = 1 | F_X = 1) - P(O = 1 | F_X = 0)$ measures factor influence; and (2) paired t -test p -values, where $p > 0.05$ indicates successful elimination. For **quantitative calibration**, we specify that factor F_j ’s influence should be ρ times that of factor F_i , with $\rho \in \{1, 2, 3, 4, 5\}$ randomly selected. Our method enforces $\text{AME}_j = \rho \cdot \text{AME}_i$ via constrained optimization (Equation 7); baselines receive natural language instructions. We report **Relative Error**: $|\hat{\rho} - \rho|/\rho$, where $\hat{\rho}$ is the empirical AME ratio. We sample 100 instances each from COMMON2SENSE, PLASMA, and TODAY for evaluation.

5.2 Experimental Analysis

Decision Accuracy. Table 1 demonstrates that IDEA consistently improves decision accuracy across model scales and domains. Most notably, IDEA with Qwen-3-32B achieves 78.6% average accuracy, surpassing both the reasoning-specialized DeepSeek-R1 (DeepSeek-AI, 2025) (68.1%) and the larger GPT-5.2 (77.9%). This result suggests that the primary bottleneck in high-stakes decision-making is not model capacity alone, but rather the misalignment between LLMs’ internal knowledge and their external outputs. By externalizing decision logic into interpretable factors, IDEA harnesses latent knowledge more effectively than CoT or logit-based methods, which remain susceptible to hallucination and miscalibration.

Ablation Analysis. Each component contributes meaningfully. Removing interactions (w/o Inter) causes the largest drop (7.6% on Qwen-3-32B), confirming that realworld decisions involve non-linear factor dependencies. Performance also declines with fixed verbal mappings (w/o EM) and deterministic factor assignment (w/o MC), validating that learned calibration and uncertainty marginalization are essential design choices.

Probability Calibration. Table 1 (right) shows that IDEA produces probability estimates capable of distinguishing subtle evidential differences between paired conditions. While pairwise comparison (PWC) benefits from explicit comparative framing at small scale, IDEA substantially outperforms it on larger models (0.693 vs. 0.481 avg. F1 for Qwen-3-32B). The largest margins appear on the

Model	Method	BIG	Crd	C2S	PLS	TOD	Avg	F1 C_1	F1 C_2	F1 Eq	Avg
N/A	Random	50.0	50.0	50.0	50.0	50.0	50.0	0.333	0.333	0.333	0.333
DeepSeek R1	CoT	51.8	57.0	88.7	75.4	67.7	68.1	0.318	0.335	0.205	0.286
GPT-5.2	CoT	67.8	68.2	95.0	82.9	75.8	77.9	0.455	0.438	0.295	0.402
Qwen-3-4B	CoT	51.4	57.5	94.2	75.1	68.1	69.3	0.352	0.325	0.261	0.313
	Logit	52.4	54.9	88.8	75.6	70.2	68.4	0.281	0.292	0.238	0.270
	Vanilla	56.7	56.1	91.7	82.0	70.4	71.4	0.392	0.361	0.284	0.346
	BIRD	55.1	53.1	88.0	75.3	71.4	68.6	0.398	0.415	0.190	0.334
	PWC	-	-	-	-	-	-	0.542	0.610	0.415	0.522
	IDEA w/o Inter	54.4	53.4	93.0	79.3	68.6	69.7	0.522	0.494	0.358	0.458
	IDEA w/o EM	55.2	52.7	91.8	75.7	68.1	68.7	0.505	0.476	0.168	0.383
	IDEA w/o MC	54.3	53.5	88.7	75.2	68.5	68.0	0.536	0.510	0.332	0.459
	IDEA	56.8	53.3	93.1	83.9	70.7	71.6	0.575	0.553	0.385	0.504
Qwen-3-8B	CoT	50.7	58.5	93.9	75.9	66.8	69.2	0.365	0.352	0.248	0.322
	Logit	56.1	58.5	91.3	76.4	70.2	70.5	0.322	0.314	0.236	0.291
	Vanilla	59.1	58.6	91.9	78.1	70.7	71.7	0.418	0.392	0.301	0.370
	BIRD	56.2	55.1	91.3	73.3	72.4	69.7	0.552	0.567	0.346	0.488
	PWC	-	-	-	-	-	-	0.615	0.528	0.415	0.519
	IDEA w/o Inter	50.8	58.9	93.0	80.2	70.6	70.7	0.675	0.660	0.501	0.612
	IDEA w/o EM	52.8	60.2	89.4	76.6	68.3	69.5	0.658	0.642	0.472	0.591
	IDEA w/o MC	52.5	58.9	89.2	79.5	69.4	69.9	0.683	0.671	0.488	0.614
	IDEA	59.5	60.3	94.8	80.2	71.4	73.2	0.728	0.711	0.652	0.697
Qwen-3-32B	CoT	51.4	56.0	90.3	73.7	67.3	67.7	0.382	0.365	0.270	0.339
	Logit	55.7	61.1	91.5	79.7	69.4	71.5	0.345	0.333	0.249	0.309
	Vanilla	58.8	63.0	93.0	81.1	70.3	73.2	0.438	0.415	0.308	0.387
	BIRD	55.3	59.2	91.7	79.9	71.0	71.4	0.590	0.602	0.371	0.521
	PWC	-	-	-	-	-	-	0.548	0.564	0.330	0.481
	IDEA w/o Inter	51.7	60.6	92.4	76.8	73.5	71.0	0.715	0.698	0.518	0.644
	IDEA w/o EM	51.6	63.2	93.5	76.3	74.2	71.8	0.706	0.689	0.501	0.632
	IDEA w/o MC	55.3	58.1	94.7	82.0	68.8	71.8	0.732	0.590	0.530	0.617
	IDEA	69.3	68.9	95.1	84.5	75.0	78.6	0.762	0.745	0.572	0.693

Table 1: Main results across two evaluation paradigms. *Left (Experiment I)*: Binary decision accuracy (%) on five benchmarks spanning complex real-world decisions (BIGDATA22, German Credit) and commonsense reasoning (COMMON2SENSE, PLASMA, TODAY). *Right (Experiment II)*: Macro F1 for three-way probability ranking on paired COMMON2SENSE queries, measuring whether models correctly identify which condition provides stronger evidential support. F1 scores are reported separately for cases where C_1 dominates, C_2 dominates, or neither (Equal). Shaded rows denote IDEA variants; best results per model size in **bold**.

Model	Category	Method	Factor Exclusion		Quantitative Calibration
			ERR (\uparrow)	p -value (\uparrow)	Relative Error (\downarrow)
Qwen-3-4B	Prompting	CoT	0.12	0.021	0.88
		Vanilla	0.33	0.004	0.71
		Logit	0.27	0.009	0.79
	Editing	IDEA	1.00	N/A	0.00
Qwen-3-8B	Prompting	CoT	0.36	0.013	0.66
		Vanilla	0.08	0.044	0.92
		Logit	0.43	0.001	0.74
	Editing	IDEA	1.00	N/A	0.00
Qwen-3-32B	Prompting	CoT	0.18	0.008	0.81
		Vanilla	0.41	0.046	0.63
		Logit	0.06	0.017	0.95
	Editing	IDEA	1.00	N/A	0.00

Table 2: Results of user-edited factor intervention. We categorize CoT, Vanilla, and Logit as generic **Prompting** methods, distinct from IDEA. *Note*: Under IDEA factor exclusion, the decision model removes F_X by setting $\beta_X=0$ and dropping all interactions involving F_X , so $\Delta P \equiv 0$ and the paired t -test is undefined (reported as N/A). IDEA enforces $|\text{AME}_j - \rho \text{AME}_i| \leq 10^{-6}$, so the relative error rounds to 0.00.

“Equal” class (the most challenging category) indicating superior calibration granularity.

Controllability. Table 2 exposes a fundamental limitation of prompting: natural language cannot guarantee behavioral compliance. All prompting methods show significant residual factor influence ($p < 0.05$) and high calibration error (0.63–0.95). IDEA achieves perfect factor exclusion (ERR = 1.00) and exact calibration (relative error = 0.00) through direct parameter manipulation—precision unattainable via prompting.

Interpretability in Practice. Beyond controllability, a two-stage user study (Appendix I) shows that IDEA’s AME-based factor ranking aligns with expert consensus (Spearman’s $\rho = 0.83$; strong inter-rater agreement, Kendall’s $W = 0.81$), and that expert parameter edits correct 38% of IDEA’s errors versus 12% under equivalent natural-language feedback to CoT. It demonstrates that parameter-level access translates into measurably better human–AI collaboration.

6 Conclusion

We presented IDEA, a framework that addresses the trust deficit in LLM-based decision-making by externalizing model knowledge into an interpretable, editable parametric form over semantically meaningful factors. Through joint estimation of verbal-to-numerical mappings and decision model parameters, correlated sampling that preserves factor dependencies, and direct parameter manipulation for expert intervention, IDEA achieves calibrated probabilities, semantic interpretability, and quantitative controllability. Experiments demonstrate that IDEA with Qwen-3-32B (78.6%) outperforms both DeepSeek R1 (68.1%) and GPT-5.2 (77.9%), while achieving perfect factor exclusion and exact quantitative calibration—precision unattainable through prompting. These results suggest that the primary barrier to trustworthy LLM decision-making is not model capacity, but the misalignment between internal knowledge and external outputs.

Limitations

Extending Beyond Binary Settings. While our binary formulation provides a principled foundation with strong empirical results, real-world decisions often involve richer structures. Future work can extend IDEA to ordinal factors via cumulative

link models or continuous factors through Gaussian process priors, broadening applicability while preserving interpretability.

Scaling Factor Spaces. The current framework handles moderate factor sets effectively. Incorporating active learning or Bayesian experimental design could enable efficient exploration of larger configuration spaces, identifying the most informative factor combinations without exhaustive probing.

Automating Factor Discovery. Our verification procedure ensures high-quality factors but relies on LLMs possessing sufficient domain knowledge to generate discriminative and comprehensive factor sets. Future work could integrate automated concept extraction methods, leverage LLM self-critique mechanisms for discriminability testing, or incorporate retrieval-augmented generation to supplement domain expertise—enabling fully automated pipelines across diverse decision scenarios.

Relaxing Assumptions. Factor completeness and verbal consistency serve as effective working assumptions validated by our strong empirical performance. Future work could quantify sensitivity to assumption violations or develop robust estimation procedures that explicitly model assumption uncertainty.

Efficiency Improvements. The computational overhead reflects IDEA’s thoroughness in extracting decision knowledge. Techniques such as factor caching across related queries, distillation into lightweight inference models, or early-stopping heuristics during marginalization could reduce latency while maintaining calibration quality.

Ethics Statement

Our framework aims to enhance transparency and accountability in LLM-assisted decision-making, directly addressing the trust deficit that limits deployment in high-stakes domains. By externalizing decision logic into interpretable factors with quantifiable contributions, IDEA empowers stakeholders to inspect, contest, and refine automated recommendations—capabilities essential for responsible AI deployment.

Intended Benefits. The framework promotes human oversight by making decision processes auditable and editable. Domain experts can verify whether learned factor weights align with institutional values and regulatory requirements, and inter-

vene with mathematically guaranteed effects when they do not.

Potential Risks. Like all decision-support tools, IDEA could be misused if deployed without appropriate human oversight or if factor sets encode societal biases. We emphasize that our framework is designed to augment—not replace—human judgment, particularly in consequential domains like credit assessment. As a preliminary check, manual inspection of LLM-generated factor sets across 300 sampled instances spanning the five benchmarks revealed no spurious or biased factors; see Appendix J.

Data and Evaluation. All experiments use publicly available benchmark datasets. The German Credit dataset, while standard in ML research, reflects historical lending patterns that may embed demographic biases; our use is purely for methodological evaluation, not to endorse any particular lending criteria.

Broader Impact. We believe that interpretable, editable decision frameworks represent a positive direction for human-AI collaboration, enabling meaningful expert participation rather than passive acceptance of opaque model outputs.

Acknowledgments

This work is supported by Advanced Materials-National Science and Technology Major Project (Grant No. 2025ZD0620100), HKUST(GZ)-IEIP-RoP (G01RF000256), and National Key R&D Program of China (No. 2024YFA1012700).

References

Faeze Brahman, Chandra Bhagavatula, Valentina Pyatkin, Jena D. Hwang, Xiang Lorraine Li, Hirona Jacqueline Arai, Soumya Sanyal, Keisuke Sakaguchi, Xiang Ren, and Yejin Choi. 2023. [Plasma: Making small language models better procedural knowledge models for \(counterfactual\) planning](#). *CoRR*, abs/2305.19472.

David V. Budescu, Stephen Broomell, and Han-Hui Por. 2009. [Improving communication of uncertainty in the reports of the intergovernmental panel on climate change](#). *Psychological Science*, 20(3):299–308.

Christopher J. C. Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Gregory N. Hullender. 2005. [Learning to rank using gradient descent](#). In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*,

ACM International Conference Proceeding Series, pages 89–96. ACM.

- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *CoRR*, abs/2501.12948.
- Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22.
- Haonan Dong, Kehan Jiang, Haoran Ye, Wenhao Zhu, Zhaolu Kang, and Guojie Song. 2026. [Neureasoner: Towards explainable, controllable, and unified reasoning via mixture-of-neurons](#). *Preprint*, arXiv:2604.02972.
- Haonan Dong, Wenhao Zhu, Guojie Song, and Liang Wang. 2025. [AuroRA: Breaking low-rank bottleneck of loRA with nonlinear mapping](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Yu Feng, Ben Zhou, Weidong Lin, and Dan Roth. 2025. [BIRD: A trustworthy bayesian inference framework for large language models](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Yu Feng, Ben Zhou, Haoyu Wang, Helen Jin, and Dan Roth. 2023. [Generic temporal reasoning with differential analysis and explanation](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12013–12029. Association for Computational Linguistics.
- Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.
- Hans Hofmann. 2023. [Statlog \(german credit data\)](#).
- Kehan Jiang, Haonan Dong, Zhaolu Kang, Zhengzhou Zhu, and Guojie Song. 2026. [Foe: Forest of errors makes the first solution the best in large reasoning models](#). *Preprint*, arXiv:2604.02967.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. [Language models \(mostly\) know what they know](#). *Preprint*, arXiv:2207.05221.
- Eunji Kim, Dahuin Jung, Sangha Park, Siwon Kim, and Sungroh Yoon. 2023. [Probabilistic concept bottleneck models](#). In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 16521–16540. PMLR.

- Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR.
- Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. [Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiuūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, and 11 others. 2023. [Measuring faithfulness in chain-of-thought reasoning](#). *Preprint*, arXiv:2307.13702.
- Ollie Liu*, Deqing Fu*, Dani Yogatama, and Willie Neiswanger. 2024. [DeLLMa: Decision Making Under Uncertainty with Large Language Models](#). *Preprint*, arXiv:2402.02392.
- Xiaou Liu, Tiejun Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025. [Uncertainty quantification and confidence calibration in large language models: A survey](#). In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V2, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, pages 6107–6117. ACM.
- Scott M. Lundberg and Su-In Lee. 2017. [A unified approach to interpreting model predictions](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774.
- Carina Mood. 2010. Logistic regression: Why we cannot do what we think we can do, and what we can do about it. *European sociological review*, 26(1):67–82.
- Tuomas P. Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. 2023. [Label-free concept bottleneck models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Shikhar Singh, Nuan Wen, Yu Hou, Pegah Alipoormolabashi, Te-Lin Wu, Xuezhe Ma, and Nanyun Peng. 2021. [COM2SENSE: A commonsense reasoning benchmark with complementary sentences](#). In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, Findings of ACL, pages 883–898. Association for Computational Linguistics.
- Yejun Soun, Jaemin Yoo, Minyong Cho, Jihyeong Jeon, and U Kang. 2022. Accurate stock movement prediction with self-supervised learning from sparse noisy tweets. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1691–1700. IEEE.
- Chung-En Sun, Tuomas P. Oikarinen, Berk Ustun, and Tsui-Wei Weng. 2025. [Concept bottleneck large language models](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- Yuxi Sun, Aoqi Zuo, Haotian Xie, Wei Gao, Mingming Gong, and Jing Ma. 2026. [Fact-e: Causality-inspired evaluation for trustworthy chain-of-thought reasoning](#). *Preprint*, arXiv:2604.10693.
- Qwen Team. 2025. [Qwen3 technical report](#). *CoRR*, abs/2505.09388.
- Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. [Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Thomas S Wallsten, David V Budescu, Amnon Rapoport, Rami Zwick, and Barbara Forsyth. 1986. Measuring the vague meanings of probability terms. *Journal of Experimental Psychology: General*, 115(4):348.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. [Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Sherry Yang, Ofir Nachum, Yilun Du, Jason Wei, Pieter Abbeel, and Dale Schuurmans. 2023a. [Foundation models for decision making: Problems, methods, and opportunities](#). *CoRR*, abs/2303.04129.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. 2023b. [Language in a bottle: Language model guided concept bottlenecks for interpretable image classification](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 19187–19197. IEEE.

A Detailed EM Algorithm Formulation

A.1 Latent Variable Formulation

We formulate the joint estimation problem as inference in a latent variable model. Working in logit space simplifies the mathematics: let $Z^{(k)} = \text{logit}(P^{(k)}) \in \mathbb{R}$ denote the latent log-odds for observation k . The generative process assumes:

Prior (Model Prediction). The decision model induces a prior distribution on latent log-odds:

$$Z^{(k)} \mid \mathbf{f}^{(k)}, \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}^\top \mathbf{x}^{(k)}, \sigma_\theta^2) \quad (8)$$

where $\mathbf{x}^{(k)}$ is the augmented feature vector containing intercept, main effects, and interaction terms. The variance σ_θ^2 captures model uncertainty.

Likelihood (Verbal Observation). The verbal response provides a noisy observation of the latent log-odds through the mapping ϕ :

$$\text{logit}(\phi(V^{(k)})) \mid Z^{(k)} \sim \mathcal{N}(Z^{(k)}, \sigma_\phi^2) \quad (9)$$

The variance σ_ϕ^2 captures noise in the verbal expression process. This formulation treats verbal responses as observations of an underlying continuous probability, corrupted by the inherent imprecision of verbal expression.

The conjugacy of Gaussian distributions is critical: it yields closed-form posterior distributions, enabling efficient E-step computation without approximation.

A.2 Complete EM Update Equations

E-Step. Given current estimates $(\boldsymbol{\theta}^{(t)}, \phi^{(t)})$, we compute the posterior distribution of each latent variable. By Gaussian conjugacy, this posterior is also Gaussian:

$$Z^{(k)} \mid \mathbf{f}^{(k)}, V^{(k)}, \boldsymbol{\theta}^{(t)}, \phi^{(t)} \sim \mathcal{N}(\tilde{z}^{(k)}, \tilde{\sigma}^2) \quad (10)$$

with posterior mean:

$$\tilde{z}^{(k)} = \lambda \cdot (\boldsymbol{\theta}^{(t)\top} \mathbf{x}^{(k)}) + (1 - \lambda) \cdot \text{logit}(\phi^{(t)}(V^{(k)})) \quad (11)$$

and posterior variance:

$$\tilde{\sigma}^2 = \frac{\sigma_\theta^2 \cdot \sigma_\phi^2}{\sigma_\theta^2 + \sigma_\phi^2} \quad (12)$$

where $\lambda = \sigma_\phi^2 / (\sigma_\theta^2 + \sigma_\phi^2)$ is the precision-weighted balance. Intuitively, the posterior mean is a convex combination of two sources of information—what the model predicts and what the verbal response indicates—weighted by their relative precisions.

M-Step for $\boldsymbol{\theta}$. Maximizing the expected complete-data log-likelihood with respect to $\boldsymbol{\theta}$ reduces to weighted least squares in logit space. We augment this with regularization and a margin-ranking loss that preserves directional consistency with verbal indications:

$$\mathcal{L}_{\text{MR}} = \frac{1}{K} \sum_{k=1}^K \max(0, -y^{(k)} \cdot (\boldsymbol{\theta}^\top \mathbf{x}^{(k)}) - 0.5) + \epsilon \quad (13)$$

where $y^{(k)} = \text{sign}(\phi^{(0)}(V^{(k)}) - 0.5)$ is computed once from the initial mapping and $\epsilon > 0$ is a margin parameter. This hinge loss penalizes predictions that contradict the directional tendency (above/below 0.5) indicated by verbal responses.

The complete M-step objective is:

$$\boldsymbol{\theta}^{(t+1)} = \arg \min_{\boldsymbol{\theta}} \left\{ \frac{1}{K} \sum_{k=1}^K (\tilde{z}^{(k)} - \boldsymbol{\theta}^\top \mathbf{x}^{(k)})^2 + \lambda_{\text{MR}} \mathcal{L}_{\text{MR}} + \Omega(\boldsymbol{\theta}) \right\} \quad (14)$$

We optimize via proximal gradient descent. At each iteration, we compute gradients of the differentiable terms (MSE, margin-ranking loss, ℓ_2 penalty), perform a gradient step, then apply the proximal operator for the ℓ_1 penalty:

$$\gamma_{ij} \leftarrow \text{sign}(\gamma_{ij}) \cdot \max(|\gamma_{ij}| - \eta \lambda_1, 0) \quad (15)$$

where η is the learning rate. This soft-thresholding drives small interaction coefficients to exactly zero, yielding interpretable sparse structure.

M-Step for ϕ . For each verbal category $v_m \in \mathcal{V}$, we update the mapping to minimize expected squared deviation from posterior means:

$$\phi^{(t+1)}(v_m) = \sigma \left(\frac{1}{|\mathcal{K}_m|} \sum_{k \in \mathcal{K}_m} \tilde{z}^{(k)} \right) \quad (16)$$

where $\mathcal{K}_m = \{k : V^{(k)} = v_m\}$ indexes observations with verbal response v_m . To preserve semantic coherence, we enforce monotonicity through

category-specific bounds:

$$\phi(v_m) \leftarrow \text{clip}(\phi(v_m), \phi_m^{\min}, \phi_m^{\max}) \quad (17)$$

These bounds prevent semantic inversions (e.g., “likely” mapping below “unlikely”), ensuring the learned mapping respects the ordinal structure of verbal expressions.

A.3 Initialization and Convergence

Initialization. We initialize $\phi^{(0)}$ with canonical values from prior literature on verbal probability (e.g., “very unlikely” \rightarrow 0.15, “likely” \rightarrow 0.75) and $\theta^{(0)}$ via standard logistic regression using these initial numerical targets.

Convergence Monitoring. We track the Q-function (expected complete-data log-likelihood):

$$Q = -\frac{K}{2} \log(2\pi\sigma_\theta^2) - \frac{1}{2\sigma_\theta^2} \sum_{k=1}^K \mathbb{E}[(Z^{(k)} - \theta^\top \mathbf{x}^{(k)})^2] \\ - \frac{K}{2} \log(2\pi\sigma_\phi^2) - \frac{1}{2\sigma_\phi^2} \sum_{k=1}^K \mathbb{E}[(\text{logit}(\phi(V^{(k)})) - Z^{(k)})^2] \quad (18)$$

where expectations incorporate posterior variance: $\mathbb{E}[(Z^{(k)} - \mu)^2] = (\hat{z}^{(k)} - \mu)^2 + \tilde{\sigma}^2$. We also monitor the observed data log-likelihood:

$$\mathcal{L} = -\frac{K}{2} \log(2\pi(\sigma_\theta^2 + \sigma_\phi^2)) - \frac{1}{2(\sigma_\theta^2 + \sigma_\phi^2)} \sum_{k=1}^K (\text{logit}(\phi(V^{(k)})) - \theta^\top \mathbf{x}^{(k)})^2 \quad (19)$$

which follows from marginalizing the latent variable under the Gaussian model. Iteration terminates when $|Q^{(t+1)} - Q^{(t)}| < \epsilon$ or a maximum iteration count is reached.

B Experimental Details

B.1 Dataset Description

We evaluate on five datasets spanning complex decisions and reasoning tasks. All instances are converted to binary decision queries.

- **BIGDATA22** (Soun et al., 2022): Stock movement prediction from market signals and tweets. 1,472 instances predicting price increase/decrease.
- **Statlog German Credit** (Hofmann, 2023): Credit risk assessment from demographic and financial features. 1,000 instances classifying good/bad credit risks.
- **COMMON2SENSE** (Singh et al., 2021): Commonsense reasoning with true/false statements. We focus on comparative reasoning

instances where a smaller model exhibits low confidence. Using GPT-4, we rewrite each statement into its opposite and generate 10 supporting conditions per outcome with reverse verification. 216 scenarios, 3,672 instances, 9 conditions per outcome on average.

- **TODAY** (Feng et al., 2023): Temporal reasoning studying how extra sentences affect temporal relations. 895 instances.
- **PLASMA** (Brahman et al., 2023): Procedural planning with goal-conditioned plan revision. We use GPT-4 to identify key steps differing between original and revised plans, converting them to binary outcomes. We generate 5 supporting conditions for the less common outcome. 279 scenarios, 1,178 instances.

For GPT-4 generated datasets, three reviewers checked 100 random instances each. All three agreed on 91% of labels; at least two agreed on 94%.

B.2 Model Description

We use Qwen-3-4B/8B/32B (Team, 2025) as primary models. Temperature is 0 for behavioral probing and 1.2 for Monte Carlo sampling. We compare against DeepSeek R1 (reasoning-specialized) and GPT-5.2 (state-of-the-art commercial model), both evaluated with chain-of-thought prompting. We accessed deepseek-R1 and Chatgpt-5.2 using the official APIs of OpenAI and DeepSeek.

B.3 Baseline Description

- **Vanilla** (Wang et al., 2023): Directly verbalize probability estimates with self-consistency. Temperature = 0.7, 3 samples, majority vote.
- **Logit**: Use output token probability directly. Greedy decoding (temperature = 0).
- **CoT** (Wei et al., 2022): Chain-of-thought reasoning before probability verbalization, with self-consistency.
- **PWC**: Pairwise comparison presenting both conditions simultaneously. Model selects which condition better supports the outcome. Only applicable to Experiment II; not directly comparable as it sees both conditions.
- **BIRD** (Feng et al., 2025): Factor-based elicitation assuming factor independence with fixed verbal-to-numerical mappings.

B.4 Ablation Description

- **w/o Inter:** Remove interaction terms ($\gamma_{ij} = 0$). Tests whether pairwise interactions capture meaningful patterns.
- **w/o EM:** Use fixed verbal mappings from (Budescu et al., 2009): *very unlikely*→0.05, *unlikely*→0.15, *somewhat unlikely*→0.30, *neutral*→0.50, *somewhat likely*→0.70, *likely*→0.85, *very likely*→0.95. Tests value of learned calibration.
- **w/o MC:** Deterministic factor assignment (random with $p = 0.5$ when uninformed) instead of correlated sampling. Tests importance of preserving factor dependencies.

B.5 Hyperparameters

Probing configurations K : 256. Monte Carlo samples T : 50. EM convergence: 10^{-4} . Elastic net λ_1 : 0.01, λ_2 : 0.001. E-step precisions $\tau_\theta = \tau_\phi = 1.0$.

C Validating Factor Completeness

Assumption 1 (Factor Completeness) is strong and cannot be verified through LLM self-critique alone. We evaluate robustness through controlled factor removal experiments using Qwen-3-32B.

Leave-One-Factor-Out (LOFO). For each factor F_j , we retrain IDEA excluding F_j and measure accuracy degradation Δ_j .

Dataset	$\max(\Delta_j)$	$\text{mean}(\Delta_j)$	$\min(\Delta_j)$
German Credit	8.4	3.2	0.3
TODAY	5.1	1.8	0.1
COMMON2SENSE	3.9	1.4	0.2

Table 3: Accuracy degradation (%) under single-factor removal.

Maximum degradation of 8.4% (German Credit, “employment stability”) indicates certain factors are critical. Mean degradation of 1.4–3.2% suggests partial redundancy—correlated factors compensate for individual removals.

Progressive Removal. We removed factors iteratively by decreasing $|\beta_j|$ until 50% exclusion. Table 4 reports results.

Degradation is gradual until $\sim 30\%$ removal, then accelerates. This indicates factor sets contain “core” factors capturing the majority of predictive signal and “peripheral” factors providing incremental refinement. Notably, even with 30% of

Removed	0%	15%	30%	40%	50%
German Credit	68.9	67.1	64.8	60.2	54.1
TODAY	75.0	73.8	71.5	68.3	63.7
COMMON2SENSE	95.1	94.2	92.6	89.1	84.5

Table 4: Accuracy (%) under progressive factor removal.

factors removed, IDEA on German Credit (64.8%) remains competitive with CoT using the same base model with full information (56.0%, Table 1), demonstrating that the structured factorization provides robustness to moderate incompleteness.

C.1 Case Studies

We present three illustrative cases examining factor completeness in practice.

Case 1: Complete Coverage. *Query:* “A person puts ice in their drink. Will the drink get colder?” (COMMON2SENSE)

Factors: (1) Ice temperature below drink temperature, (2) Sufficient ice-liquid contact, (3) Drink not at freezing point.

Assessment: Physical process fully characterized. IDEA predicts $P = 0.94$; ground truth positive. **Success.**

Case 2: Partial Gap. *Query:* Loan application, €5,000, employed applicant with moderate savings, 15-year customer. (German Credit)

Factors: Stable employment (1), positive credit history (1), adequate income-to-debt (1), sufficient collateral (0).

Missing: Long-standing customer relationship—a positive signal absent from factors.

Outcome: IDEA predicts $P(\text{approve}) = 0.67$; actual approved. Under-confident due to missing relational factor. **Partial failure.**

Case 3: Critical Omission. *Query:* Stock movement prediction, tech earnings week. (BIG-DATA22)

Factors: Positive earnings surprise (1), favorable analyst sentiment (1), sector momentum positive (1), no macro headwinds (1).

Missing: Bearish signals from options market activity not captured by any factor.

Outcome: IDEA predicts $P(\text{up}) = 0.81$; stock declined. Self-referential verification passed because the LLM lacks access to real-time options data. **Failure.**

C.2 Summary

Factor incompleteness causes graceful degradation rather than catastrophic failure. Performance remains competitive even at 30% incompleteness. Critical failures occur primarily when (1) real-time external data is required or (2) relational context is implicit.

D Error Analysis

D.1 Failure Mode Analysis

We analyze 50 misclassified instances per dataset manually to identify error sources across pipeline stages.

Model	Factor Inf.	Decision Model	Margin.
Qwen-3-4B	58.4	27.2	14.4
Qwen-3-32B	31.6	42.8	25.6

Table 5: Error distribution by pipeline stage (%).

Factor Inference Errors. The dominant failure mode for smaller models. Consider this COM-MON2SENSE instance:

Condition: “The executive dismissed the proposal without reviewing the supporting data.”

Factor F_3 : Decision-maker engaged in thorough evaluation

Qwen-3-4B incorrectly assigned $F_3 = 1$, triggered by surface keywords (“proposal,” “data”) while missing negation semantics. Qwen-3-32B correctly assigned $F_3 = 0$. This shallow pattern-matching accounts for 41.2% of Qwen-3-4B’s factor inference errors.

Decision Model Errors. More prevalent in larger models (42.8% vs. 27.2%), indicating that improved factor extraction exposes limitations in pairwise interaction modeling for complex multi-factor dependencies.

D.2 Interpretability Case Studies

Factor	Description	β_j	AME
F_1	Stable employment	+1.82	+0.284
F_2	Acceptable debt ratio	+1.67	+0.261
F_3	Positive credit history	+1.43	+0.223
F_4	Sufficient collateral	+0.89	+0.139

Table 6: Partial Learned parameters for credit decision.

Case Study 1: Credit Approval. The learned AME values align with established credit risk principles: employment stability and debt ratio dominate. The positive interaction $\gamma_{12} = +0.62$ captures how stable employment compounds the signal of manageable debt.

Case Study 2: Temporal Reasoning. For the query: “Can a 2019 college graduate have competed in the 2015 Olympics?”

The dominant weight on F_1 reflects the core temporal constraint. IDEA correctly predicts “Yes” ($P = 0.73$), recognizing that a 2019 graduate could plausibly have been 18–22 during the 2015 Olympics.

D.3 Error Propagation

Near-linear degradation indicates IDEA does not catastrophically amplify upstream errors. Factor inference quality serves as the primary bottleneck, motivating future work on robust extraction.

E Decision Model Architecture Analysis

We investigate whether replacing logistic regression with a shallow neural network improves decision accuracy, and whether such gains justify the loss of interpretability.

E.1 Experimental Setup

We compare the logistic regression model (Equation 2) against a single-hidden-layer MLP with 16 hidden units and ReLU activation. Both models are trained identically via the EM procedure (Section 4.3) on three datasets: BIGDATA22, German Credit, and PLASMA. We use Qwen-3-8B as the base LLM and report accuracy averaged over 5 random seeds.

E.2 Results

The MLP yields modest improvements (+0.7% average), none statistically significant under paired t -test ($p > 0.15$ for all datasets).

E.3 Discussion

The marginal gains reflect three factors: (i) logistic regression with pairwise interactions already captures dominant structure in our low-dimensional factor space; (ii) ablations in Table 1 indicate bottlenecks lie in factor dependency modeling and verbal calibration, not decision model expressiveness.

Critically, the MLP sacrifices interpretability. Logistic coefficients directly quantify factor contri-

Factor	Description	β_j	AME
F_1	Timeline permits participation	+2.31	+0.341
F_2	Age-appropriate for competition	+1.89	+0.279
F_3	Educational timeline consistent	+0.72	+0.106

Table 7: Learned parameters for temporal reasoning.

Model	BIG	Credit	PLASMA	Avg
Logistic (IDEA)	59.5	60.3	80.2	66.7
MLP-16	60.2	61.1	81.0	67.4
Δ	+0.7	+0.8	+0.8	+0.7

Table 8: Accuracy (%) comparison between logistic regression and MLP decision models.

butions (Section 4.5), enabling guaranteed interventions demonstrated in Experiment III. Neural network weights lack such semantic correspondence. We therefore retain logistic regression as the default.

F Robustness to Violations of Verbal Probability Consistency

Assumption 2 posits that LLMs’ verbal probability expressions reflect an underlying numerical probability through a consistent, monotonic mapping. We investigate how IDEA performs when this assumption is violated.

F.1 Experimental Setup

We inject controlled noise into the behavioral probing data to simulate inconsistency. Specifically, with probability ϵ , we replace the LLM’s verbal response with a randomly selected adjacent category (e.g., “likely” \rightarrow “somewhat likely” or “very likely”). We train IDEA on corrupted data and evaluate on clean test sets using Qwen-3-8B across all five datasets.

F.2 Results

Noise ϵ	BIG	Crđ	TOD	PLS	C2S
0.00 (baseline)	59.5	60.3	71.4	80.2	94.8
0.10	58.1	59.5	70.2	78.9	93.6
0.20	55.8	57.2	68.1	76.1	91.2
0.30	52.4	54.8	64.5	72.3	87.5

Table 9: Decision accuracy (%) under verbal label noise.

IDEA exhibits graceful degradation: accuracy drops by only 2–3% when 10% of verbal labels are corrupted, and remains above baseline methods

(Table 9) even at 20% noise. This robustness stems from the EM algorithm treating verbal responses as noisy observations—the E-step averages over inconsistent signals, while the ranking loss preserves ordinal structure.

G Ordinal Consistency of Verbal Probabilities

Assumption 2 requires only that verbal responses preserve a monotonic ordering over underlying probabilities—not that they share a fixed numerical mapping. We test this directly.

Protocol. For each dataset, we enumerate all comparable configuration pairs (f_a, f_b) where f_a dominates f_b element-wise (e.g., $f_a = (1, 1, 0)$ vs. $f_b = (1, 0, 0)$) and check whether the LLM’s verbal response to f_a is at least as likely as its response to f_b on the seven-level scale. This yields 7,680 pairs across 5 datasets \times 3 model sizes (15 settings).

Results. Overall ordinal consistency reaches **87.6%**, ranging from 80.8% (BIGDATA22, Qwen-3-4B) to 94.2% (COMMON2SENSE, Qwen-3-32B), with larger models showing stronger consistency. Combined with the graceful degradation under corrupted verbal labels reported in Appendix F, these results justify Assumption 2 as an operational rather than strict requirement.

H Human Evaluation of Factor Determination

At inference time, the LLM partitions factors into observed vs. uncertain based on the specific condition C . We validate this step directly.

Protocol. We sample 150 instances (50 each from COMMON2SENSE, PLASMA, and TODAY) and recruit three annotators to independently label every (factor, condition) pair as *positive*, *negative*, or *uncertain*. Inter-annotator agreement is Fleiss’ $\kappa = 0.76$ (substantial).

Results. Qwen-3-32B achieves 86.3% 3-way accuracy and 91.5% on the binary observed-vs-uncertain distinction most relevant for sampling. Qwen-3-8B achieves 79.1% and 85.8% respectively, confirming that factor-determination quality scales with model capacity. These results indicate that the partition step is a reliable LLM capability at sufficient scale.

I Interpretability User Study

We assess whether IDEA’s parameters are not merely *editable* but also *useful* to human experts.

Stage 1: Factor Importance Agreement. Three domain experts independently rank factors by perceived importance for each of the five datasets (totaling 50 factor sets), without seeing IDEA’s coefficients. Inter-rater concordance is strong (Kendall’s $W = 0.81$). Per-task Spearman’s ρ between the expert consensus and IDEA’s AME-induced ranking averages **0.83** across five datasets (range: 0.71 on BIGDATA22 to 0.94 on COMMON2SENSE).

Stage 2: Expert-Guided Error Correction. We sample 100 instances on which IDEA predicted incorrectly (20 per dataset). Experts inspect the factor set, identify factors they consider irrelevant or missing, and perform simple targeted edits (setting $\beta_X = 0$ or adding factors). As a matched control, equivalent expert feedback is provided as natural-language instructions to CoT on the same instances. Experts flagged 18% of factors as needing modification on average. Expert-guided parameter edits corrected **38%** of IDEA’s errors (38/100), compared to 12% for CoT with equivalent natural-language feedback (12/100). This $3.2\times$ gap mirrors the controllability results: natural language cannot reliably translate expert intent into behavioural change, whereas parameter edits carry a mathematical guarantee.

J Audit of Spurious and Biased Factors

A natural concern with LLM-mined factor sets is that they may encode demographic or otherwise problematic signals (e.g., race or gender proxies in credit assessment). Manual inspection of LLM-generated factor sets across 300 sampled instances spanning the five benchmarks revealed zero instances of spurious or biased factors.

Beyond this empirical observation, IDEA offers a *structural* advantage for bias mitigation that

prompting lacks: every factor is named, its contribution is quantified by its AME, and any flagged factor can be removed with a mathematically guaranteed zero effect, turning bias auditing from an opaque post-hoc task into a transparent, verifiable one. We note that 300 instances constitute a limited audit; deployment in consequential domains should pair IDEA with domain-specific fairness review.

K Computational Overhead Analysis

K.1 LLM Query Requirements

Method	Setup (one-time)	Per Instance
Vanilla/CoT/Logit	0	1
BIRD	0	N
IDEA	$\sim 20 + \min(2^N, 256)$	$N + T$

Table 10: LLM query counts. For IDEA with $N = 6, T = 50$: setup ≈ 84 queries; inference = 56 queries/instance.

We can also conduct parallel queries and utilize LLM to speed up the process. Furthermore, we would like to emphasize that it is essential to utilize LLM as much as possible to gain a comprehensive understanding of all the knowledge it possesses regarding a query scenario. This is the core and strength of our approach. Therefore, it is imperative to frequently invoke the API of LLM. Efficiency is not the selling point of our method. It’s not what we are aiming for. Moreover, if we make decisions based on the trained decision model after the training process is completed, we only need to determine the values of specific conditions. Once determined, the computational cost of our decision-making can be considered almost zero. This is because the computational cost of the logistic regression model is so low that it can be negligible, and it can instantly handle a large number of queries. IDEA requires more queries than direct methods, but setup costs (factor identification + behavioral probing) amortize across test instances. For 1,000 instances, the per-instance overhead averages to ~ 56.1 queries.

K.2 Latency Comparison

With parallelized API calls, IDEA ($T=50$) latency reduces to approximately 5 seconds. For latency-sensitive applications, $T = 20$ provides a reasonable trade-off, maintaining most accuracy gains at 40% of the query cost.

Method	Latency (s)	Accuracy (%)
Vanilla	0.8	71.7
CoT	2.1	69.2
IDEA ($T=50$)	18.4	73.2
IDEA ($T=20$)	8.2	72.4

Table 11: Per-instance latency with sequential execution (Qwen-3-8B).

L Hyperparameter Configuration

Table 12 summarizes all hyperparameters used in IDEA.

Component	Hyperparameter	Value
EM Algorithm	Precision ratio τ_θ/τ_ϕ	1.0
	Convergence threshold	10^{-4}
	Max iterations	100
Regularization	λ_1 (L1 on γ)	0.01
	λ_2 (L2 on γ)	0.001
	λ (ranking loss)	0.1
Inference	MC samples T	50
	Sampling temperature	1.2
Probing	Configuration samples	256

Table 12: Hyperparameter configuration for IDEA.

EM Parameters. The precision ratio $\tau_\theta/\tau_\phi = 1.0$ assigns equal trust to model predictions and verbal indications. We searched over $\{0.5, 1.0, 2.0\}$ and found performance stable across this range. Convergence typically occurs within 15–30 iterations.

Regularization. We select (λ_1, λ_2) via 5-fold cross-validation on the behavioral probing dataset. The ranking loss weight $\lambda = 0.1$ was chosen from $\{0.01, 0.1, 0.5\}$ to balance regression fit with ordinal consistency.

Inference. We set $T = 50$ Monte Carlo samples based on diminishing accuracy gains beyond this point. Sampling temperature 1.2 balances diversity against coherence.

Reporting Protocol. For stochastic components (Monte Carlo marginalization, verbal probability elicitation with non-zero temperature), we average over three independent samples per instance to reduce variance. Deterministic components are run once. Each reported number is a point estimate; run-to-run variability on a held-out subset of COMMON2SENSE is $< 0.5\%$ accuracy and < 0.01 macro F1 (Appendix M).

Verbal Mapping Initialization. Following (Budescu et al., 2009), we initialize ϕ as: *very unlikely*: 0.05, *unlikely*: 0.15, *somewhat unlikely*: 0.30, *neutral*: 0.50, *somewhat likely*: 0.70, *likely*: 0.85, *very likely*: 0.95.

M Sensitivity Analysis

We analyze sensitivity to critical hyperparameters using Qwen-3-8B on COMMON2SENSE.

Samples (T)	Accuracy (%)	Std. Error
10	93.2	0.42
25	94.3	0.28
50	94.8	0.19
100	94.9	0.14

Table 13: Effect of Monte Carlo sample size.

λ_1	Interaction Sparsity (%)	Accuracy (%)
0.001	12.5	94.1
0.01	45.8	94.8
0.1	87.5	93.6

Table 14: Effect of L1 regularization strength.

Results indicate IDEA is robust to hyperparameter choices within reasonable ranges. The most sensitive parameter is λ_1 , which controls the interpretability-accuracy tradeoff via interaction sparsity.

N Experiment II Ground Truth Construction

N.1 Dataset Composition

Our ground truth dataset consists of 500 paired instances. Of these, 350 instances (70%) are sourced from the existing expert-annotated BIRD dataset (Feng et al., 2025). The remaining 150 instances (30%) were newly annotated by experts to ensure adequate coverage of edge cases and complex reasoning scenarios.

N.2 Annotation Protocol

We recruited a pool of eight annotators through a university participant pool. All participants were native English speakers holding at least a bachelor’s degree. To ensure high-quality data, candidates were screened via a 20-instance qualification task; only those achieving an accuracy of $\geq 85\%$ were selected (8 of 11 candidates qualified). Selected annotators underwent a 30-minute training

session focused on assessing evidential strength before beginning independent annotation.

Instructions and Compensation. Annotators were provided with detailed guidelines defining “evidential strength” as the logical support a condition provides for a specific outcome. The instructions included definitions for strict dominance and equivalence, accompanied by five anchor examples demonstrating varying degrees of logical support. Participants were compensated at a rate of **\$10** per hour, which was determined to be above the local minimum wage and consistent with fair research standards.

Task Definition. For each paired query (Q_1, Q_2), annotators were tasked with selecting one of three labels: (a) C_1 provides stronger support, (b) C_2 provides stronger support, or (c) Both provide equal support. Each instance was evaluated by three independent annotators, with final labels determined by majority vote.

N.3 Inter-Annotator Agreement

We evaluate the reliability of our new annotations using Fleiss’ κ and percentage agreement metrics, as summarized in Table 15.

Metric	Value
Fleiss’ κ	0.71
Pairwise Agreement	84.2%
Unanimous Agreement	67.3%

Table 15: Inter-annotator agreement statistics for the 150 newly annotated instances.

The obtained Fleiss’ κ of 0.71 indicates substantial agreement among annotators (Landis and Koch, 1977). Agreement on cases with clear dominance was notably higher (91.4%), with disagreements primarily concentrated on borderline “Equal” instances. To validate consistency with the source data, we performed a re-annotation of 50 randomly sampled BIRD-derived instances, which yielded an agreement rate of 88.0% (Cohen’s $\kappa = 0.79$), confirming robust cross-dataset consistency.

O Use of AI Tools

During the preparation of this work, the authors utilized Large Language Models (LLMs) exclusively for grammatical refinement, proofreading, and stylistic polishing to improve readability. The

```

EXAMPLE PROMPT --- 1. DECOMPOSE QUERY

SYSTEM
You are an expert at analyzing decision-making queries.
Your task is to decompose a query into:
1. Query Scenario: The background context of the decision
2. Positive Decision Outcome: One possible decision outcome
3. Negative Decision Outcome: The opposing decision outcome (binary opposite)
4. Specific Condition: The specific constraints or conditions mentioned in the query

You must respond in the following JSON format:
{
  "scenario": "<the query background and context>",
  "positive_outcome": "<the positive/affirmative decision outcome>",
  "negative_outcome": "<the negative/opposing decision outcome>",
  "specific_condition": "<specific constraints or conditions in the query>"
}

USER
Please decompose the following query into its components:
Query: {query}

Analyze and extract:
1. The scenario (background context)
2. The two opposing binary decision outcomes
3. Any specific conditions mentioned

Respond in JSON format only.

ASSISTANT
{
  "scenario": "The company has limited budget but expects significant growth in cloud services demand.",
  "positive_outcome": "Invest in a new data center.",
  "negative_outcome": "Do not invest in a new data center.",
  "specific_condition": "Limited budget, significant growth expected"
}

```

Figure 3: Prompting Example: DECOMPOSE QUERY.

```

EXAMPLE PROMPT --- 2. GENERATE STATEMENTS

SYSTEM
You are an expert at generating comprehensive decision-supporting statements.
Given a scenario and a decision outcome, generate exactly 5 different statements that support why this outcome might be chosen.
Each statement should:
1. Be comprehensive and cover different aspects
2. Include specific conditions, factors, or circumstances
3. Be distinct from other statements

Format your response as:
# <statement 1>
# <statement 2>
# <statement 3>
# <statement 4>
# <statement 5>

USER
Scenario: {scenario}
Decision Outcome: {outcome}

Generate 5 comprehensive statements explaining different conditions or factors that would support choosing this outcome.
(If this is not the first call, the following will be inserted: "Focus on different aspects than the previous statements.")

Format each statement starting with '#'

ASSISTANT
# The projected revenue growth from cloud services exceeds the cost of borrowing capital.
# Current infrastructure is at 95% capacity and cannot handle the Q4 traffic spike.
# Competitors are upgrading their facilities, risking our market share if we delay.
# New energy-efficient cooling technology in the new center will reduce OpEx by 20%.
# Regulatory compliance requires local data residency which current facilities lack.

```

Figure 4: Prompting Example: GENERATE STATEMENTS.

scientific content, logical reasoning, and conclusions presented in this manuscript remain the sole responsibility of the authors.

P Prompting Display

EXAMPLE PROMPT --- 3. EXTRACT FACTORS

SYSTEM

You are an expert at identifying decision factors from statements. Analyze the given statements and extract distinct decision factors. Each factor must have exactly two binary values - one supporting each outcome.

Your response MUST be in valid JSON format:

```
{
  "Factor Name 1": {
    "positive_value": "<value description supporting positive outcomes>",
    "negative_value": "<value description supporting negative outcomes>"
  },
  "Factor Name 2": {
    "positive_value": "<value description supporting positive outcomes>",
    "negative_value": "<value description supporting negative outcomes>"
  }
}
```

Rules:

- Each factor must be distinct and focus on a unique aspect
- Factor values must be specific and directly reference elements from the statements
- Avoid vague terms like 'the object' or 'the thing'
- Each factor's two values MUST support different outcomes

USER

Scenario: {scenario}
 Positive Outcome: {positive_outcome}
 Supporting Statements:
 #1 {positive_statement_1}
 #2 {positive_statement_2}
 ...
 Negative Outcome: {negative_outcome}
 Supporting Statements:
 #1 {negative_statement_1}
 #2 {negative_statement_2}
 ...
 Extract distinct decision factors from these statements. Each factor must have two binary values that support different outcomes.
 Respond in JSON format only.

ASSISTANT

```
{
  "Market Demand": {
    "positive_value": "High demand growth expected in cloud services",
    "negative_value": "Market saturation or uncertain demand"
  },
  "Budget Availability": {
    "positive_value": "Sufficient internal funds or cheap credit available",
    "negative_value": "Limited budget with high cost of borrowing"
  }
}
```

Figure 5: Prompting Example: EXTRACT FACTORS.

EXAMPLE PROMPT --- 5. CHECK OVERLAPPING FACTOR

SYSTEM

You are an expert at identifying overlapping or duplicate factors. Analyze the given factors and identify any that are semantically similar or overlapping. Keep only the most comprehensive version of each factor.

Respond in JSON format:

```
{
  "overlapping_groups": [
    {
      "factors": ["factor1", "factor2"],
      "keep": "factor1",
      "reason": "<why this factor is more comprehensive>"
    }
  ],
  "unique_factors": ["factor3", "factor4"]
}
```

USER

Factors to check for overlap:
 {json_string_of_factors}

Identify any factors that:

- Cover the same aspect or dimension
- Are semantically similar
- Have redundant values

For overlapping factors, indicate which one to keep.

ASSISTANT

```
{
  "overlapping_groups": [
    {
      "factors": ["Cost", "Financial Impact"],
      "keep": "Financial Impact",
      "reason": "Financial Impact covers both immediate cost and long-term ROI"
    }
  ],
  "unique_factors": ["Technical Feasibility"]
}
```

Figure 7: Prompting Example: CHECK OVERLAPPING FACTOR.

EXAMPLE PROMPT --- 4. CHECK BINARY SUPPORT

SYSTEM

You are an expert at analyzing decision factors. For each factor, determine if its two values genuinely support different outcomes. Respond in JSON format:

```
{
  "factor_name": {
    "positive_value_supports": "<Positive/Negative/Neutral>",
    "negative_value_supports": "<Positive/Negative/Neutral>",
    "valid": true/false
  }
}
```

A factor is valid only if one value supports the positive outcome and the other supports the negative outcome.

USER

Positive Outcome: {positive_outcome}
 Negative Outcome: {negative_outcome}

Factors to check:
 {json_string_of_factors}

For each factor, verify that:

- The positive_value supports the Positive Outcome
- The negative_value supports the Negative Outcome
- They support DIFFERENT outcomes

Mark factors as invalid if both values support the same outcome or neither clearly supports an outcome.

ASSISTANT

```
{
  "Market Demand": {
    "positive_value_supports": "Positive",
    "negative_value_supports": "Negative",
    "valid": true
  },
  "Legacy Systems": {
    "positive_value_supports": "Neutral",
    "negative_value_supports": "Neutral",
    "valid": false
  }
}
```

Figure 6: Prompting Example: CHECK BINARY SUPPORT.

EXAMPLE PROMPT --- 6. CHECK CONDITION COVERAGE

SYSTEM

You are an expert at analyzing decision factor coverage. Given a specific condition and a set of factors, identify any aspects of the condition that are NOT covered by the existing factors.

Respond in JSON format:

```
{
  "covered_aspects": ["aspect1", "aspect2"],
  "missing_aspects": ["aspect3", "aspect4"],
  "suggested_factors": [
    "New Factor Name": {
      "positive_value": "<value supporting positive outcomes>",
      "negative_value": "<value supporting negative outcomes>"
    }
  ]
}
```

USER

Scenario: {scenario}
 Specific Condition: {specific_condition}
 Positive Outcome: {positive_outcome}
 Negative Outcome: {negative_outcome}

Existing Factors:
 {json_string_of_factors}

Analyze whether the existing factors fully cover all relevant aspects of the specific condition. Identify any missing aspects and suggest new factors to cover them.

ASSISTANT

```
{
  "covered_aspects": ["Budget limits"],
  "missing_aspects": ["Cloud service demand growth"],
  "suggested_factors": [
    "Growth Potential": {
      "positive_value": "Rapid growth requires scaling now",
      "negative_value": "Slow growth allows delaying investment"
    }
  ]
}
```

Figure 8: Prompting Example: CHECK CONDITION COVERAGE.

EXAMPLE PROMPT -- 7. FACTOR DETERMINATION (STEP 1)

SYSTEM

You are an expert at analyzing decision factors. Based on specific conditions, determine whether a factor can be identified as favorable or unfavorable. Respond in JSON format only: {determination: "positive" | "negative" | "uncertain", "reason": "brief explanation"}

USER

Given the specific conditions of a decision scenario: {specific_condition}

Consider the following factor:
 Factor: {factor.factor_name}
 - Favorable (positive) condition: {factor.positive_description}
 - Unfavorable (negative) condition: {factor.negative_description}

Based SOLELY on the information provided in the specific conditions above, determine whether this factor is:
 (a) Clearly favorable/positive - the conditions indicate positive value
 (b) Clearly unfavorable/negative - the conditions indicate negative value
 (c) Cannot be determined - insufficient information in the conditions

Respond in JSON format.

ASSISTANT

```
{
  "determination": "positive",
  "reason": "The specific conditions explicitly state that the budget is sufficient, which matches the favorable condition."
}
```

Figure 9: Prompting Example: FACTOR DETERMINATION (STEP 1).

EXAMPLE PROMPT -- 8. MONTE CARLO SAMPLING (STEP 2)

SYSTEM

You are an expert at generating realistic decision scenarios. Given known factors and context, generate coherent values for uncertain factors. Respond in JSON format only: {reasoning: "brief explanation", "factor_values": {"F1": 1, "F2": -1, ...}} Use 1 for favorable/positive and -1 for unfavorable/negative.

USER

Consider a decision scenario: {scenario}

Specific conditions: {specific_condition}

The following factors are already known:
 {list_of_observed_factors_descriptions}

For the following uncertain factors, generate a realistic and coherent scenario by determining each factor's value. Consider how these factors might correlate with each other and the known factors.

Uncertain factors to determine:
 {list_of_uncertain_factors_with_descriptions}

For each uncertain factor, decide whether it is favorable (1) or unfavorable (-1). Respond in JSON format with factor IDs as keys.

ASSISTANT

```
{
  "reasoning": "Given the high market demand (known), it is likely that technical feasibility will be prioritized despite costs.",
  "factor_values": {
    "F3": 1,
    "F5": -1
  }
}
```

Figure 10: Prompting Example: MONTE CARLO SAMPLING (STEP 2).