LAMDA: Large Language Model as Decision Analyst

Yifan Hong[†]
Department of Industrial Engineering, Tsinghua University, Beijing 100084, China
hongyf23@mails.tsinghua.edu.cn

Sizhong Qin[†]
Department of Civil Engineering, Tsinghua University, Beijing 100084, China
qsz23@mails.tsinghua.edu.cn

Chen Wang[*]
Department of Industrial Engineering, Tsinghua University, Beijing 100084, China
chenwang@tsinghua.edu.cn
[*]Corresponding author

Abstract. Influence diagrams address the challenges of decision making under risk by structuring information, decisions, and values, while clearly depicting uncertainties and probabilistic dependencies. However, constructing an influence diagram requires expertise in decision analysis and is further complicated by the need to process large amounts of contextual information. This work focuses on the construction of influence diagrams from natural language input by leveraging large language models (LLMs). We design a workflow that prompts LLMs to output elements of an influence diagram and resolves issues through verification and regeneration. We also construct a new data set of typical decision problems under risk. Evaluations using this data set demonstrate that our framework effectively identifies key factors and relationships in natural language, making better decisions than standalone LLMs and LLMs enhanced with standard techniques such as chain-of-thought. Finally, we apply the large LAnguage Model as Decision Analyst (LAMDA) to discussions by groups of disease control experts on a hypothetical pandemic to demonstrate its real-world applicability. Overall, the method effectively synthesizes unstructured text into an influence diagram that, although subject to human review and refinement, enhances information processing and supports decision making.

Funding: Financial support from the National Natural Science Foundation of China [Grants 72192824 and 72361137004] is gratefully acknowledged.

Key words: decision under risk, influence diagram, large language model

History: Received June 8, 2025; revised June 16, 2026; accepted June 30, 2026. Published online in Just Accepted July 10, 2026, and in Articles in Advance August 7, 2026.

1. Introduction

Decisions under risk face challenges of multidisciplinary information, high uncertainty, and conflicting objectives. To make informed decisions, decision makers must estimate the probabilities of events and evaluate the outcomes of alternative actions. This complexity becomes particularly evident in public policy decisions, because these choices often affect a wide range of stakeholders, resulting in extensive consequences. Decision makers must integrate diverse stakeholder needs

[†] Yifan Hong and Sizhong Qin contributed equally to the manuscript.
This is the author accepted manuscript; the final published version is available at https://doi.org/10.1287/deca.2025.0438.

and sources of knowledge—for example, by synthesizing insights from epidemiology, economics, and behavioral science when managing a pandemic (Dillon et al. 2023). Here, communication is as essential as computation, enabling both knowledge integration and consensus building (Gregory et al. 2005).

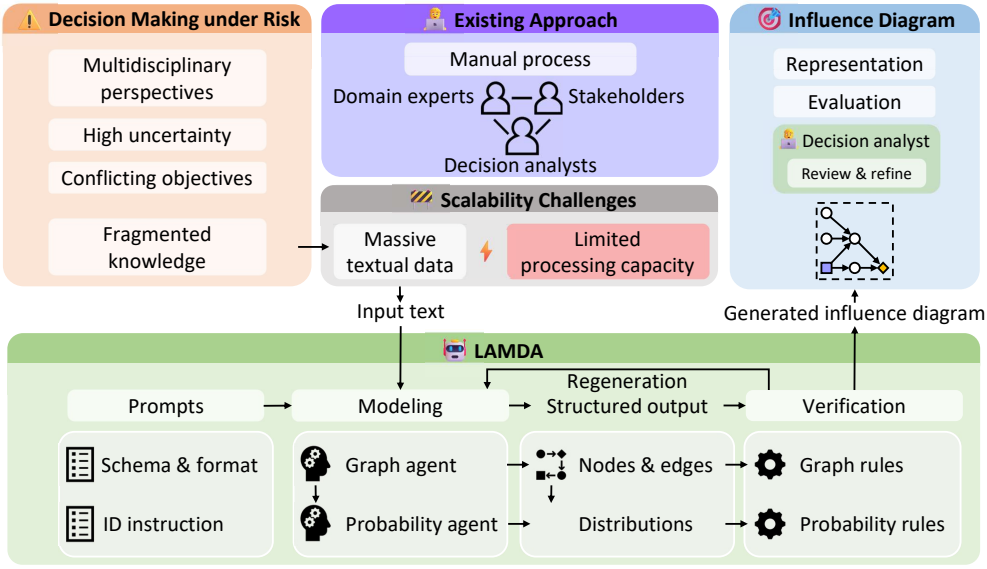


Figure 1 **Overview of the proposed LAMDA framework.**

Decision analysis addresses these challenges with a prescriptive framework, with influence diagrams serving as a powerful representation approach (Howard and Matheson 2005b). Influence diagrams model uncertainties, decisions, and values as interconnected nodes, allowing systematic problem-solving through efficient algorithms (Shachter and Bhattacharjya 2011). Successful applications of influence diagrams span individual choices (Bodily and Furman 2016, Wan et al. 2022), organizational strategies (Bordley 2023, Stonebraker 2023) and public policies (Merrick and McLay 2010, Keating et al. 2023).

However, constructing influence diagrams typically requires specialized modeling expertise and is often carried out manually by analysts collaborating with decision makers. The overwhelming volume of relevant information, much of which is unstructured and difficult to process, poses substantial burdens on human analysts, especially under time pressure. Moreover, the knowledge needed for real-world decision making is often scattered across natural-language sources such as journal articles, scholarly books (von Winterfeldt 2013), technical reports, and expert discussions. Bringing together these dispersed sources of information and expertise is challenging, and relying on human analysts alone may lead to inefficiency, cognitive overload, and biases.

Advances in natural language processing, particularly in large language models (LLMs), have created new opportunities for building influence diagrams from textual data (Brown et al. 2020, Kojima et al. 2022, Wei et al. 2022). In medical decision making, for example, there have been successful applications of LLMs to process unstructured data (e.g., texts and images) and support diagnosis and treatment decisions (Huang et al. 2024, Perlis et al. 2024, Sandmann et al. 2025).

In this paper, we propose a framework that utilizes the large LAnguage Model as Decision Analyst (LAMDA) to construct influence diagrams from natural language input (see Figure 1). Rather than taking the place of human decision analysts, LAMDA is designed to enhance their work by generating a draft diagram that extracts and organizes decision-relevant information from unstructured text. Specifically, an LLM-based graph agent generates nodes and arcs relevant to the decision problem, and an LLM-based probability agent traverses the nodes to assign probabilities. These generation steps are followed by rule-based verification and regeneration to identify and fix emerging issues.

To evaluate the proposed LAMDA framework, we construct a new data set of textual descriptions of typical decision problems, paired with ground-truth influence diagrams drawn from both the decision analysis literature and synthetic scenarios. Results show that LAMDA yields better decisions than standalone LLMs and standard techniques such as chain-of-thought (CoT) by accurately transforming key decision elements and probabilistic relationships into influence diagrams. The resulting structured representations also provide a more transparent and auditable basis for decision support, better aligning with the requirements of normative decision making.

Furthermore, we apply LAMDA to analyze a real-world case study based on discussions among disease control experts about a hypothetical H5N1 influenza outbreak. We show that LAMDA generates influence diagrams that are comprehensive and highly relevant, compared with a manually annotated reference diagram based on standard disease control and treatment manuals and expert input. The case study therefore demonstrates LAMDA's potential to synthesize large volumes of conversation text into a structured representation in practice.

In Section 2, we present the LAMDA framework for eliciting elements of an influence diagram. In Section 3, we introduce a new data set of texts associated with various ground-truth influence diagrams, followed by an experimental evaluation of the proposed framework. Section 4 applies the elicitation method to a practical case study on disease control. Section 5 concludes the paper.

2. Large Language Model as Decision Analyst Framework

The large LAnguage Model as Decision Analyst framework converts unstructured natural-language descriptions of a decision problem into a draft influence diagram. Its goal is to extract decision-relevant variables, relationships, beliefs, and preferences from text, so that the resulting diagram

can support downstream decision analysis. LAMDA is intended to support, not replace, decision analysts. Although the resulting influence diagram can be evaluated to generate decision recommendations, these recommendations require validation and calibration before use. Because decision quality depends strongly on problem framing, important objectives, alternatives, or constraints may be missing from the initial text. We therefore view LAMDA as producing a first-draft influence diagram from the currently available information, which analysts can merge, refine, and simplify further. In subsequent analysis, this draft can also serve as the ''basic influence diagram'' (Matheson and Matheson 2005).

LAMDA generates influence diagrams through an iterative modeling-and-verification framework with two stages: graph generation and probability elicitation. In both stages, it uses structured prompting to generate outputs that can be automatically parsed and verified against predefined validity constraints. Because the initial outputs may contain formatting or structural errors, the framework combines rule-based verification with iterative regeneration, feeding identified issues back to the LLM for refinement. The procedure terminates after a predefined number of iterations. If no valid graph is produced within this limit, it raises an error and terminates. If LAMDA fails to generate a probability distribution for a variable, it raises a warning and defaults to a uniform distribution as an uninformative prior. In the following, we first revisit the basics of influence diagrams and then detail the workflow of the LAMDA framework.

2.1. Influence Diagram

Influence diagrams (IDs) are graphical models for representing and evaluating decisions. They contain *decision nodes* for actions, *utility nodes* for objectives, and *chance nodes* for uncertain variables. Edges into decision nodes indicate available information, whereas edges into utility or chance nodes indicate probabilistic dependencies.

We introduce notations to formalize LAMDA's control logic (see details in Algorithms 1 and 2). An influence diagram is a tuple $M = (G, \mathcal{P})$, where $G = (N, E, D, U)$ is a directed acyclic graph with node set N, edge set E, and subsets $D, U \subseteq N$ denoting decision and utility nodes, respectively. We refer to G as the *influence diagram graph*. The parameterization $\mathcal{P}$ specifies a domain $dom(X)$ for each node $X \in N$, with $dom(X) \subset \mathbb{R}$ for utility nodes $X \in U$, and a probability distribution $P(X \mid \mathrm{PA}_X^G)$ for each nondecision node $X \in N \setminus D$, where $\mathrm{PA}_X^G = \{Y \in N \mid \exists e \in E,\ e = (Y, X)\}$ denotes the parent nodes of X in G. A policy $\pi : \mathrm{PA}_D^G \to \Delta(D)$ specifies actions given the values of the parent nodes of D. Together, π and $\mathcal{P}$ induce a joint distribution P^π over N, and the value of a policy is its expected utility under P^π. An optimal policy maximizes this expected utility.

We next describe how LAMDA automates graph generation and probability elicitation, the two core steps in constructing an influence diagram.

2.2. Graph Generation

The first stage of constructing the influence diagram involves generating the graphical structure, comprising nodes and edges (see Algorithm 1). Each node represents a variable and must specify a name, type (chance, decision, or utility), and a list of valid values. Edges encode conditional-dependence relationships (and are often given a causal interpretation in practice (Everitt et al. 2021, Fox et al. 2021)) and require both endpoint variables to appear in the node list. These constraints are formalized in a JavaScript Object Notation (JSON) schema and included in the prompt p^G. The source text s_0 provides a contextual background for the decision, and the LLM is instructed to incorporate only nodes and edges explicitly mentioned or strongly implied in the source text.

We design a problem- and model-agnostic prompt template that includes (i) requirements for the current generation task, (ii) brief instructions about influence diagrams, and (iii) placeholders for the input text and the current set of nodes and/or edges. This design adapts the LLM in-context to support modeling performance while remaining generalizable across LLMs and problem scenarios. In the joint generation approach, the LLM generates nodes and edges simultaneously (see Figure 2). When modifying an existing graph, this approach includes a postprocessing step to identify and remove redundant nodes and their associated edges. We also explore a sequential generation approach, in which the LLM first generates the nodes, with semantically redundant ones pruned, and then generates edges to represent dependencies among the finalized node set (see Section A.1).

Algorithm 1 LAMDA Graph Generation

Input: Source text s_0, prompts for graph generation p^G, max iterations T_G
Initialize: $s \leftarrow s_0$; $G_0 \leftarrow (N_0, E_0, D_0, U_0)$ with all empty sets, $G \leftarrow G_0$
Output: Influence diagram graph G
for $t \leftarrow 1$ **to** T_G **do**
  $o_t^G \leftarrow \text{LLM}(p^G, s, o_{t-1}^G)$ ▷ Structured output
  $\text{valid}_G, \text{issues}_G \leftarrow \text{VerifyGraph}(o_t^G)$
  if valid_G **then**
    $G \leftarrow \text{Parse}(o_t^G)$ ▷ Parse into ID graph
    break
  else
    $s \leftarrow \text{Concat}(s_0, \text{issues}_G)$ ▷ Append issues to the source text
  end if
end for
return G

The initial LLM outputs may contain formatting or structural issues, such as noncompliant strings or invalid influence diagram components (e.g., cyclic edges). To address these issues, we implement

You are an expert decision analyst specializing in decision-making under uncertainty and influence diagram modeling. You will be provided with a source text. Your goal is to EXTRACT an influence diagram that STRICTLY represents the decision problem AS DESCRIBED in the text.
Steps:

1. Node extraction: Extract ONLY decision-related variables that are EXPLICITLY MENTIONED or CLEARLY IMPLIED in the text. Each variable must correspond to uncertainty, action or utility.
    - For each node, provide a name (using terminology from the source text), a type (decision, chance, utility), and valid values.
    - You must be able to cite or reference the specific part of the text that supports each node.
2. Edge extraction: Extract ONLY influences and causal relations that are EXPLICITLY STATED or CLEARLY IMPLIED in the text.
    - Each edge must be supported by specific text evidence.
    - Edge types: a) Information edges (ending in decision nodes): Shows what information is known when making decisions b) Dependency edges (between other nodes): Shows probabilistic dependencies

Text:
{text}
The original node list:
{node_list}
The original edge list:
{edge_list}
Requirements:

1. Do not include nodes or edges in the original lists or have similar meanings to existing ones.
2. The nodes and edges should be directly mentioned or clearly implied in the text.
3. The node names should be concise and meaningful, and should come from the source text if possible.
4. Utility node values should be numerical. We aim to maximize the utility, so if a utility node is cost-related, the values should be negative. If a utility node is benefit-related, the values should be positive.
5. No cycles are allowed. For example, if the influence with condition 'a' to variable 'b' is present, there cannot be another influence with condition 'b' to variable 'a'.
6. All cause and effect variables (nodes) used in the edge list should be members of the node list, either in the original or the added list.
7. Output using the following format:
   {format_instructions}

Figure 2 Prompt for joint generation of influence diagram graph.

a verification and regeneration process to ensure that the outputs comply with the JSON schemas and are logically valid. Specifically, verification involves several key checks. For nodes, we examine the presence of required fields (variable name, type, and valid values) and verify that each node's type is one of the permitted values (chance, decision, or utility). The graph must include at least one utility and at least one decision node, and each node's value list must contain a minimum of two distinct elements. To ensure edge validity, we verify that all referenced nodes are present in the node list. Furthermore, we check for cycles using a depth-first search to guarantee that the graph is acyclic. To ensure solvability for influence diagrams with multiple decision nodes, the graph must satisfy sufficient recall (Fox et al. 2021). We therefore apply rule-based verification to confirm a complete ordering over decisions and to ensure that information available at earlier decisions is also available to subsequent ones. If any verification fails, the system logs the details of the problems and provides feedback to the graph agent, triggering another round of modeling. This refinement process repeats until all validation criteria are met or the maximum number of iterations is reached.

2.3. Probability Elicitation

Once the nodes and edges of the influence diagram are configured, LAMDA then specifies the subjective probability distributions from the textual input (see Algorithm 2). This step is challenging because natural-language descriptions of decision problems often do not provide probabilities in an explicit or standardized form. In some cases, probabilistic information is left unstated because it is treated as common knowledge, such as the differing likelihood of rain in the Sahara versus the Amazon rainforest. In other cases, probabilities are genuinely unavailable because the underlying uncertainty is not yet well understood, such as the transmission mode of a novel virus in the early stage of an outbreak. Even when uncertainty is expressed verbally, qualitative probability terms can be interpreted inconsistently across individuals (Budescu et al. 2014). These challenges connect to a broader literature on how people represent and communicate uncertainty. Prior work has examined the relationship between mental models and linguistic expressions of uncertainty (Gibson 2011, Dhami and Mandel 2022, Karvetski et al. 2022), highlighting the difficulty of translating natural-language descriptions into precise probabilistic assessments. Recently, LLMs have shown promise in addressing this gap. Existing studies suggest that LLMs can reason about implicit probabilistic information (Zhao et al. 2023, Xia et al. 2024), express uncertainty in structured ways (Xiong et al. 2024), and support the elicitation of probabilities from language (Feng et al. 2025, Schoenegger et al. 2025, Zhu and Griffiths 2025). These findings motivate our use of LLMs to infer probability distributions from text when explicit numerical probabilities are incomplete or absent.

In LAMDA, the variables are processed in a valid topological sequence, ensuring that each variable is considered only after its parent variables have been assessed. For each variable, we derive distributions that describe the probabilistic linkages with its parent nodes, as specified by the graph edges. Utilizing the coding capabilities of LLMs, we prompt them to convert the probability distributions in the text into executable Python lambda functions (see Figure 3). These functions can be automatically transformed into tabular formats for subsequent computations.

The verification process for probability elicitation ensures that variable names are consistent and the code functions correctly. Importantly, this verification is entirely rule-based. It does not call an LLM and, therefore, requires no additional prompts. We verify that each variable referenced in probability expressions precisely matches those defined within the graph structure. Additionally, we assess the Python code generated by the LLM for both syntactic correctness and semantic validity. This involves confirming that the code is executable and checking the coherence of the parsed

Algorithm 2 LAMDA Probability Elicitation

Input: Source text s_0, prompts for probability elicitation p^P, influence diagram graph $G = (N, E, D, U)$, max iterations T_P
Initialize: $s \leftarrow s_0$; $\mathcal{P}_0 \leftarrow \emptyset, \mathcal{P} \leftarrow \mathcal{P}_0$
Output: Probability distribution $\mathcal{P}$
for $n \in \text{TopoOrder}(N)$ **do** ▷ Traverse the nodes in a valid topological order
  for $t \leftarrow 1$ **to** T_P **do**
    $o^P_{n,t} \leftarrow \text{LLM}(p^P, s, n, \text{PA}^G_n, o^P_{n,t-1})$ ▷ Structured output
    $\text{valid}^P_n, \text{issues}^P_n \leftarrow \text{VerifyProbabilities}(o^P_{n,t}, n, \text{PA}^G_n)$
    if valid^P_n **then**
      $\mathcal{P}_n \leftarrow \text{Parse}(o^P_{n,t})$ ▷ Parse into probability distributions
      break
    else
      $s \leftarrow \text{Concat}(s_0, \text{issues}^P_n)$ ▷ Append issues to the source text
    end if
  end for
end for
return $\mathcal{P}$

You are an expert decision analyst specializing in decision-making under uncertainty and influence diagram modeling. You will be provided with:

1. A source text: {text}
2. A target variable: {variable}
3. Evidence variables (parents): {parent_variable_list}

Your task is to:

1. **Extract the conditional probability distribution (CPD)** for {variable} based on the provided **source text**.
2. Assign probabilities **for every possible value** of {variable} given **each possible combination** of the valid evidence values.
3. Adhere to the valid domains of:
   - **Target variable** {variable_domain}
   - **Evidence variables** {condition_domain}
4. Utility node values must be numerical.
5. We aim to maximize the utility, so if a utility node is cost-related, the values should be negative. If a utility node is benefit-related, the values should be positive.
6. **Output** must strictly follow the lambda string format.

Required Output Please produce the **conditional probability distribution (CPD)** using a lambda function. All variables should be in lowercase. Please provide the following Python lambda function example in a code block, and return only the code (no additional text or explanation). The function is: `python lambda ...: ...`

1. **Function Signature**
   - If the variable has no parents (unconditional), write a lambda with no arguments.
   - If the variable has one or more parents (conditional), include them as parameters.
2. **Dictionary Keys:**
   - Must be all valid values that the variable can take (e.g., `0`, `1`, \rain'', etc.).
3. **Dictionary Values:**
   - Must be probabilities (floats) in the range **[0, 1]**.
   - For each parent condition, these probabilities **must sum to 1**.
4. **Conditional Logic:**
   - **All** `if...else` statements must occur **outside** the dictionary, selecting the appropriate dictionary as a whole.
   - **Not** allowed: {{0: 0.5 if parent else 0.2, 1: 0.5 if parent else 0.8}} # Disallowed
   - **Allowed** (top-level branching): lambda evidence_var1, evidence_var2: {{0: 0.1, 1: 0.9}} if evidence_var1 == 1 and evidence_var2 == 0 else {{0: 0.9, 1: 0.1}}
5. **No Escaped Newlines/Characters:**
   - Keep the lambda on one line; avoid using `\n` or other escape sequences within the lambda.
6. **Example:**
   - **Without parents:** lambda: {{0: 0.1, 1: 0.9}}
   - **With parents and multiple conditions:** lambda evidence_var1, evidence_var2: {{0: 0.1, 1: 0.9}} if evidence_var1 == 1 and evidence_var2 == 0 else {{0: 0.9, 1: 0.1}}

Figure 3 **Prompt for elicitation of probability distribution.**

probabilities by transforming them into a tabular form. For example, an error occurs if the probabilities of any variable do not sum up to one. If a check fails, the system logs the error details and

relays them to the probability agent, which then initiates another round of modeling to refine the probability distributions. This process continues until all verification criteria are met or the system reaches its preset iteration number.

3. Experiments on Constructed Data Sets

In this section, we investigate the capability of LAMDA to construct influence diagrams and recommend decisions through controlled experiments. We first construct a new data set of text-based decision problems for the experiments, each paired with the corresponding ground-truth influence diagram. When constructing the data set, we used well-organized text. This facilitates both generating data at scale and evaluating each step (graph, probabilities, decisions) separately. We present here the decision-making experiment that assesses LAMDA's end-to-end performance in solving text-based decision problems under risk.

We also report in Appendix B detailed performance of the two core steps: the graph construction experiment tests LAMDA's ability to extract the influence diagram graph from texts, and the probability elicitation experiment evaluates LAMDA's performance in assigning probability distributions given the ground-truth graph. Code and data set for the experiments are available on GitHub: `https://github.com/lamda-thu/LAMDA.git`.

3.1. Data Set Construction

To evaluate LAMDA's performance on eliciting influence diagrams from textual input, we create a data set of decision problem descriptions, paired with their ground-truth influence diagrams. We developed 15 distinct decision-under-risk scenarios spanning various contexts to ensure comprehensiveness. Five of them were derived from the decision analysis literature (see Table 1). For each, we manually drew the ground-truth graph, specified probability distributions, and wrote up textual problem descriptions. The remaining 10 were synthetically generated by `GPT-4o` following a predefined template, spanning personal (e.g., exam preparation), organizational (e.g., electric vehicle marketing), and public policy (e.g., disaster management) decisions. We varied the parameter values to generate instances of each decision problem. This templated approach enables scalable instance generation and robust evaluation. In total, we produced 302 decision problem instances.

3.1.1. Ground-Truth Influence Diagram Construction. We constructed the ground-truth IDs in three steps (see Figure 4). In Step 1, we annotated the ground-truth influence diagram graphs with node and edge names. For problems in the literature, the graphs were manually annotated.

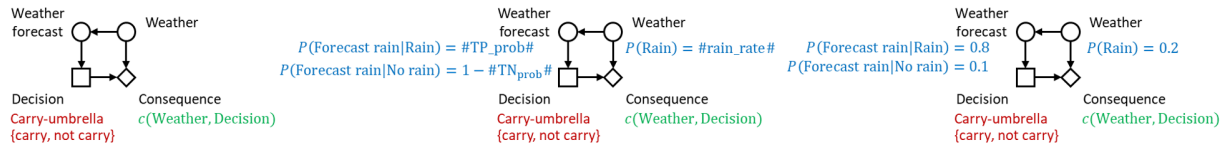


Figure 4 **Three-step ground-truth influence diagram generation, decision problem adapted from Regnier and Feldmeier (2022).**

Table 1 **The decision problems and metadata of the ground-truth influence diagrams.**

Problem name	No. of instances	No. of nodes	No. of edges	Description	Source
Long-term care insurance	32	7	13	Decide insurance coverage for long-term medical care	Bodily and Furman (2016)
Carry-umbrella	16	4	4	Whether to carry an umbrella given weather forecast	Regnier and Feldmeier (2022)
Cargo container inspection	24	9	12	Whether to screen cargo containers for smuggled nuclear threat	Merrick and McLay (2010)
Chemical usage	16	6	6	Decide usage of a chemical with economic value and potential cancer risks	Howard and Matheson (2005b)
Stay at home	54	13	19	Whether to stay at home during pandemic	Wan et al. (2022)

The LLM-generated problems first utilized `GPT-4o` to generate graphs from background descriptions. The ground-truth ID graphs were then derived through manual examination and modification, ensuring the validity of the graphs and their consistency with the problems. In Step 2, we designed symbolic templates for the probability distributions in each problem. A symbolic template specifies a parameterization over the graph, with symbolic placeholders within its specification. It enables us to manipulate the ground-truth distributions systematically. For example, in the carry-umbrella problem, three values are manipulated: the true positive probability of the weather forecast, the true negative probability of the weather forecast, and the base rate of rain. In Step 3, we filled numerical values into the symbolic templates to produce influence diagrams. When the numerical parameters change, the optimal decisions change accordingly. To ensure a comprehensive evaluation, we carefully designed these numerical values so that no single option remains universally optimal in all instances. An example of the carry-umbrella instance is provided in Appendix B (see Figure B.1).

3.1.2. Decision Problem Description Generation. We then employed an LLM (functioning as a ‘‘text generator agent’’) to produce a textual description of each decision problem from the

ground-truth ID. The text generator follows instructions to create descriptions with four information levels:

- **Background information**: Contains basic contexts about the decision problem (see Figure 5).
- **Node information**: Describes all relevant factors (nodes) in the influence diagram without mentioning their relationships (see Figure 6).
- **Graph information**: Includes qualitative information about variable dependencies (edges) in the diagram, excluding details of the probability distributions (see Figure 7).
- **Complete information**: Incorporates all previous elements along with quantitative descriptions of the probability distributions (see Figure 8).

Background:

The decision-making problem revolves around the choice to use a chemical. This decision impacts the economic value derived from the chemical, which in turn influences the overall net value. At the same time, the decision to use the chemical also impacts the cost related to cancer treatment. The cost of cancer is influenced by three factors: the decision of whether or not to use the chemical, the level of human exposure to the chemical, and the chemical's carcinogenic activity. These factors collectively influence the final net value which combines the economic benefits and the cancer-related costs of using the chemical. This decision-making problem seeks to determine the viability of using the chemical by weighing its economic benefits against its health-related costs.

Figure 5 Example of background information.

Factors to be considered:

When making a decision regarding the usage of a particular chemical or product, several factors come into play:

1. **Usage Decision**: This is the primary choice regarding whether or not to use the chemical.
2. **Economic Value**: This refers to the financial benefits or impacts associated with using the chemical.
3. **Cancer Cost**: This involves the potential health costs, specifically relating to cancer, that could result from using the chemical.
4. **Human Exposure**: This factor considers the extent to which people might be exposed to the chemical.
5. **Carcinogenic Activity**: This assesses the potential of the chemical to cause cancer based on its properties.
6. **Net Value**: This represents the overall value, considering both the economic benefits and the health costs.

Figure 6 Example of added node information.

Each higher information level subsumes all lower-level contents. For example, the node information level automatically includes background information, while the complete information level contains all preceding levels. For each decision problem instance, we generated complete information descriptions from the symbolic templates of the influence diagrams. With minor manual adjustments to the generated texts, we derived symbolic templates for the probability distributions. Finally, we specified the numerical values of the distributions to complete the ID construction.

Factors to be considered and their relations:

When making decisions, it's important to consider how different factors interact and influence the outcome. Here's a straightforward way to understand the considerations to be made in a decision-making process relating to the usage of a certain chemical.
First, the decision on whether to use the chemical directly affects its economic value. This economic value is essentially how profitable or beneficial the product is from a financial perspective. Additionally, the decision to use the chemical also impacts the cost associated with potential health risks, specifically cancer in this scenario.
The cost of cancer, which is influenced by the usage decision, is also affected by two other factors: the level of carcinogenic activity of the chemical and the degree of human exposure to it. This means that both the inherent cancer-causing potential of the chemical and how much people are exposed to it will alter the overall cancer-related costs.
The final factor to consider is the net value, which is the overall utility or worth considering all positives and negatives. This net value is determined by balancing the economic value (the financial gains) against the cancer costs (the financial and well-being losses). Thus, the overall decision should aim to maximize this net value, taking comprehensive account of the interplay between economic benefits and health-related costs.
In summary, when deciding on the usage of a potentially carcinogenic product, one should weigh the direct economic benefits against the health-related costs, where the costs are further influenced by both the nature of the chemical and the level of human exposure to it. The decision should ideally be guided by the goal of optimizing overall net value, which incorporates both these financial and health perspectives.

Figure 7 Example of added graph information.

3.2. Performance of LAMDA on Decision Making Under Risk

In the third experiment, we aim to assess LAMDA's end-to-end capability of making the correct decision, building on its separate performance on ID graph generation and probability elicitation. This experiment looks at the pipeline from textual input to decision recommendation, revealing how errors propagate. We use `Qwen2.5-72B` and `GPT-4o-2024-11-20` as the backbone LLMs.

3.2.1. Metrics. For each decision problem instance, we evaluate the quality of decision making by considering the rank order of the recommended option among all possible options and its expected utility. For simplicity, it is assumed that decision makers are risk-neutral.

3.2.1.1. *Normalized Rank.* For each decision instance, we rank all options by their expected utilities in ascending order. The raw ranks are then normalized to the [0,1] interval, where 1 represents the theoretically optimal decision and 0 represents the worst possible choice. The mean normalized rank across all instances quantifies how frequently the decision maker selects the optimal option.

3.2.1.2. *Normalized Utility.* For each decision instance, we compute the expected utility of the action. To enable cross-instance comparison, we normalize this utility relative to its theoretical maximum and minimum, mapping the result to a [0,1] scale. A score of 1 indicates optimality, whereas a score of 0 represents the worst possible outcome. The normalized utility metric offers a more nuanced assessment of decision quality, because it captures the relative performance of near-optimal choices.

3.2.2. Baseline Methods. We consider the following strategies for generating a recommended option, including random choices and multiple methods to improve the reasoning of LLM.

- **Random**: An option is sampled uniformly from all possible options.

Conditional probability distributions:

1. **Economic Value**: The economic value of a chemical depends on the usage decision:
   - If the decision is 'ban', the economic value is -1 with a probability of 1.0.
   - If the decision is 'restrict', the economic value is 1 with a probability of 1.0.
   - If the decision is 'permit', the economic value is 5 with a probability of 1.0.
2. **Cancer Cost**: The cancer cost associated with a chemical is influenced by three factors: usage decision, carcinogenic activity, and human exposure:
   - If the decision is 'ban', the cancer cost is 0 with a probability of 1.0.
   - If the decision is 'restrict'
     - If the carcinogenic activity is 'inactive', the cancer cost is 0 with a probability of 1.0.
     - If the carcinogenic activity is 'moderate', the cancer cost increases with human exposure:
       - cost -0.05 (with probability 1.0) if human exposure is 'low'
       - cost -0.5 (with probability 1.0) if human exposure is 'medium'
       - cost -5 (with probability 1.0) if human exposure is 'high'
     - If the carcinogenic activity is 'very active', the cancer cost increases with human exposure
       - cost -1 (with probability 1.0) if human exposure is 'low'
       - cost -10 (with probability 1.0) if human exposure is 'medium'
       - cost -100 (with probability 1.0) if human exposure is 'high'
   - If the decision is 'permit'
     - If the carcinogenic activity is 'inactive', the cancer cost is 0 with a probability of 1.0.
     - If the carcinogenic activity is 'moderate', the cancer cost increases with human exposure:
       - cost -0.5 (with probability 1.0) if human exposure is 'low'
       - cost -5 (with probability 1.0) if human exposure is 'medium'
       - cost -50 (with probability 1.0) if human exposure is 'high'
     - If the carcinogenic activity is 'very active', the cancer cost increases with human exposure:
       - cost -10 (with probability 1.0) if human exposure is 'low'
       - cost -100 (with probability 1.0) if human exposure is 'medium'
     - cost -1000 (with probability 1.0) if human exposure is 'high'
3. **Net Value**: The net value of a chemical is the sum of the economic value and cancer cost (with a probability of 1.0).
4. **Human Exposure**: The probability distribution for human exposure levels is:
   - 'high' with a probability of 0.5,
   - 'medium' with a probability of 0.45,
   - 'low' with a probability of 0.05.
5. **Carcinogenic Activity**: The probability distribution for carcinogenic activity is:
   - 'inactive' with a probability of 0.5,
   - 'moderate' with a probability of 0.4,
   - 'very active' with a probability of 0.1.

Figure 8 Example of added information on probability distributions.

- **Vanilla**: An LLM is provided with the problem description and directly prompted to select an option for the decision scenario. We also include a state-of-the-art reasoning model `DeepSeek-R1` (Guo et al. 2025) with the same input.
- **Chain-of-thought (CoT)** (Kojima et al. 2022): We guide LLMs to follow the expected-utility framework through structured prompts that generate explicit reasoning steps. Our prompt design systematically instructs the model to: (1) analyze the decision context, (2) assess uncertainties and probabilities, (3) estimate outcome values, and (4) calculate the expected utilities and make the final decision.

- **Self-consistency (SC)** (Wang et al. 2023): We check for self-consistency by sampling multiple reasoning paths and selecting the most frequent recommendation (based on majority voting). Using the same base prompt as the vanilla method, we configure the LLM with temperature = 0.5 to generate five independent reasoning paths.
- **DeLLMa** (Liu et al. 2025): DeLLMa (Decision-making Large Language Model assistant) is a decision-support framework that scaffolds an LLM with expected utility theory: it forecasts uncertain state variables from available texts, elicits a utility function reflecting the decision maker's objectives, and selects the action that maximizes expected utility, yielding a human-auditable rationale. In our evaluation, we use the authors' original instantiation with a factorized state representation, where chance variables are treated as independent factors (i.e., no edges among chance nodes). Consequently, chance nodes connect only to decision and utility nodes, but not to other chance nodes.

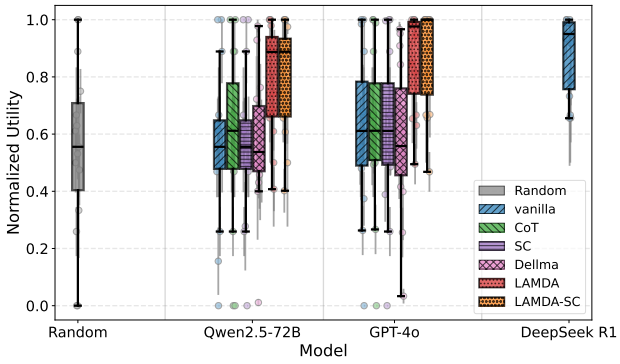


Figure 9 Expected utilities (normalized across instances) of the recommended options.

3.2.3. Results. We present the normalized utility of the options recommended by each method in different decision problems (see Figure 9). The patterns are similar for the normalized rank (see Figure B.7 in Appendix B). The best-performing methods are our LAMDA framework based on `GPT-4o`, followed by the reasoning LLM `DeepSeek-R1` and LAMDA based on `Qwen2.5-72B`. Vanilla LLMs `Qwen2.5-72B` and `GPT-4o` (without any enhancement) produced unreliable results because they missed important quantitative information and failed to perform the correct calculation. None of the CoT and self-consistency enhancements, as well as the basic implementation of DeLLMa, achieved significant improvements over random choices.

The proposed LAMDA framework showed significant performance improvements in terms of decision quality compared with random choices. When we implemented bootstrapped majority

voting across replications (denoted LAMDA-SC), we achieved further gains in decision quality. In general, LAMDA effectively extracted relevant information to construct the key elements of an influence diagram. It is worth noting that LAMDA is designed to be independent of LLMs: although performance varies across LLMs, its overall workflow remains applicable with different choices of backbone models. This flexibility supports different deployment needs. For example, locally deployed open-source models can be used when data are sensitive, whereas more capable closed-source models may be preferable when processing public information.

We also examine LAMDA's performance at each step leading to the recommended decision. LAMDA reliably extracts nodes and edges from the text and assigns node types with high accuracy (see Appendix B, Section B.3). It also translates textual probability statements into executable code, though we observe room for improvement, particularly when expressing complex dependencies, due to the LLM's limited coding reliability (see Appendix B, Section B.4). Overall, LAMDA generates influence diagrams that are consistent with the provided information, which, in turn, supports its observed performance in decision making under risk.

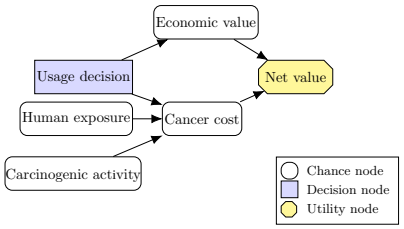


Figure 10 ID graph generated by LAMDA for the chemical usage problem.

`DeepSeek-R1` also identified the correct recommended option for this particular instance. However, its reasoning followed a decision tree-like procedure to derive the expected utility for each possible option (see Appendix B, Figure B.8), which is difficult for humans to audit or understand, especially when conditional probability relationships are coupled for more than three chance nodes and become intractable (Howard and Matheson 2005a, Shachter and Bhattacharjya 2011).

LAMDA also offers better transparency and explainability than the direct application of reasoning LLMs such as `DeepSeek-R1`, because each decision is directly derived from its corresponding influence diagram. When discrepancies occur in decision recommendations, LAMDA supports iterative refinement of the underlying influence diagram. Here, we demonstrate the validity of the LAMDA framework with the chemical usage problem at the complete information level (including texts presented in Figures 5–8). We show the influence diagram generated by LAMDA with `GPT-4o`

- variable: 'carcinogenic activity'

  lambda: {'inactive': 0.5, 'moderate': 0.4, 'very active': 0.1}'

- variable: 'human exposure'

  lambda: {'low': 0.05, 'medium': 0.45, 'high': 0.5}

- variable: 'economic value'

  lambda usage_decision: {-1: 1.0, 1: 0.0, 5: 0.0} if usage_decision == 'ban' else ({-1: 0.0, 1: 1.0, 5: 0.0} if usage_decision == 'restrict' else {-1: 0.0, 1: 0.0, 5: 1.0})

- variable: 'cancer cost'

  lambda usage_decision, carcinogenic_activity, human_exposure: {0: 1.0} if usage_decision == 'ban' else {0: 1.0} if usage_decision == 'restrict' and carcinogenic_activity == 'inactive' else {-0.05: 1.0} if usage_decision == 'restrict' and carcinogenic_activity == 'moderate' and human_exposure == 'low' else {-0.5: 1.0} if usage_decision == 'restrict' and carcinogenic_activity == 'moderate' and human_exposure == 'medium' else {-5: 1.0} if usage_decision == 'restrict' and carcinogenic_activity == 'moderate' and human_exposure == 'high' else {-1: 1.0} if usage_decision == 'restrict' and carcinogenic_activity == 'very active' and human_exposure == 'low' else {-10: 1.0} if usage_decision == 'restrict' and carcinogenic_activity == 'very active' and human_exposure == 'medium' else {-100: 1.0} if usage_decision == 'restrict' and carcinogenic_activity == 'very active' and human_exposure == 'high' else {0: 1.0} if usage_decision == 'permit' and carcinogenic_activity == 'inactive' else {-0.5: 1.0} if usage_decision == 'permit' and carcinogenic_activity == 'moderate' and human_exposure == 'low' else {-5: 1.0} if usage_decision == 'permit' and carcinogenic_activity == 'moderate' and human_exposure == 'medium' else {-50: 1.0} if usage_decision == 'permit' and carcinogenic_activity == 'moderate' and human_exposure == 'high' else {-10: 1.0} if usage_decision == 'permit' and carcinogenic_activity == 'very active' and human_exposure == 'low' else {-100: 1.0} if usage_decision == 'permit' and carcinogenic_activity == 'very active' and human_exposure == 'medium' else {-1000: 1.0} if usage_decision == 'permit' and carcinogenic_activity == 'very active' and human_exposure == 'high' else {}

- variable: 'net value'

  lambda economic_value, cancer_cost: {6: 1.0} if economic_value == 5 and cancer_cost == 1 else {5: 1.0} if economic_value == 5 and cancer_cost == 0 else {4: 1.0} if economic_value == 5 and cancer_cost == -1 else {1: 1.0} if economic_value == 1 and cancer_cost == 0 else {0: 1.0} if economic_value == -1 and cancer_cost == 0 else {-0.5: 1.0} if economic_value == 1 and cancer_cost == -0.05 else {-5: 1.0} if economic_value == 1 and cancer_cost == -5 else {-95: 1.0} if economic_value == 1 and cancer_cost == -100 else {-999: 1.0} if economic_value == 5 and cancer_cost == -1000 else {-100: 1.0} if economic_value == 1 and cancer_cost == -100 else {-1001: 1.0} if economic_value == 5 and cancer_cost == -1000 else {0: 1.0}

Figure 11 Probability distributions represented by Python lambda functions for the chemical usage problem.

as the backbone LLM in Figure 10. The generated distributions are summarized in Figure 11. The generated distributions using LAMDA with `GPT-4o` matched the ground truth exactly.

4. Case Study on Hypothetical Disease Control Decisions

The previous section utilized a controlled experiment to validate LAMDA's capability to extract information from the text and transform it into an influence diagram. In this section, we apply LAMDA to a realistic case study with less organized text input. Compared with the controlled setting, real-world discourse often contains implicit assumptions and shared background knowledge that are not explicitly stated, which makes extraction and formalization more challenging. Four groups of experts independently discussed a hypothetical disease control decision problem. Compared with the controlled experiment, the text inputs here are conversation transcripts, which are poorly organized. LAMDA can extract model components and generate an influence diagram graph for each group. It can also sequentially process each group, building upon an existing graph, to produce an aggregated graph. By comparing the generated graphs and a manually annotated reference ID graph, we demonstrate how opinion aggregation by LAMDA effectively captures decision-related information and transform unstructured text into an influence diagram. The study show-

cased LAMDA's applicability in real-world scenarios with unorganized input. It also highlights LAMDA's potential to synthesize fragmented information for complex decisions.

4.1. Case Background

About 30 young disease-control experts in a training program were gathered to discuss measures for a hypothetical H5N1 influenza outbreak in four groups. The experts were divided into four groups, each comprising 8–10 members, resulting in a unique data set of transcripts that recorded their detailed discussions.

In the hypothetical scenario, the outbreak began on a foreign farm in June with a novel H5N1 virus. By August, the outbreak had spread internationally, with eight countries reporting imported cases and four documenting local transmissions. The World Health Organization (WHO) declared a Public Health Emergency of International Concern. Faced with this evolving situation, public health authorities must decide whether to impose border restriction measures in response to the spread of the virus. Such measures could reduce disease importation, buy critical time for preparedness, and ease pressure on healthcare systems. However, these measures could also cause economic and trade disruptions, strain international relations, and incur costs associated with quarantine and surveillance efforts.

The four groups of disease control experts received instructions on the scenario, which included a description of the epidemic, (partially) available information about the virus, medical resources, and other socioeconomic factors such as population density, age distribution, transportation, and international sports events. Each group of experts independently discussed the hypothetical scenario. Their discussions were recorded and then transcribed into text. We present a sample dialogue in Figure 12. Compared with the previous experiments, the text input is much less organized.

4.2. Method

In this case study, we use LAMDA to translate the experts' discussion transcripts into influence diagrams, using `GPT-4o` as the backbone LLM. We explore both simple input from individual groups and sequential input from all groups. Both the problem description and the transcribed dialogues can be used directly as LAMDA's textual input. For each group, we obtain an influence-diagram graph by running LAMDA on the group's transcript (see Figure 13, left of the right panel). LAMDA also supports aggregation across multiple texts. When processing inputs sequentially, it maintains an evolving influence diagram that integrates information from the current transcript with the existing graph synthesized from earlier inputs (see Figure 13, right of the right panel). To improve graph

A: The thing is, there are just way too many transmission routes.
B: First of all, this disease is pretty infectious, and the severity is also pretty high. And there's community spread. In that situation, if we want to prevent it, then we definitely have to impose restrictions. The earlier you restrict, the better. That's my feeling.
E: But the restriction measures aren't complete. You're only restricting entry and exit at the border.
A: What about animals, poultry, how do you restrict those?
C: You're only restricting people.
D: You have to consider the international influence. We also have friendly diplomacy.
B: That needs to be considered, sure, but we're the national CDC.
D: The CDC also has to consider international relations.
B: From our perspective of epidemic prevention and control, other things need to be considered, but they're not the top priority.
D: You definitely have to think about the country's international image first. Then whether the country's measures, whether those measures are actually necessary. You'd be doing it on the premise of sacrificing the economy, people-to-people exchanges, and international friendliness.
B: We can talk about whether we need to restrict certain key groups to some extent. Or do you think we shouldn't restrict at all, just not manage it whatsoever?
D: It's not "not manage it at all." If you don't do restrictions. You have to follow up with supervision. Measures definitely have to follow.
E: What do you mean by "restriction"? Like, you just don't let people come in? Or you let them in, but with quarantine?
B: What he means by "restriction" is like the kind where you basically cut ties.
C: The restriction is only aimed at certain groups of people.

Figure 12 **Sample dialogue from Group 1 in the hypothetical case.**

quality, after each update, we prompt the LLM to remove redundant nodes and edges, yielding a cleaner aggregated diagram. Because the order of the input sequence can affect LAMDA's processing results, we test with all possible orders in this case study.

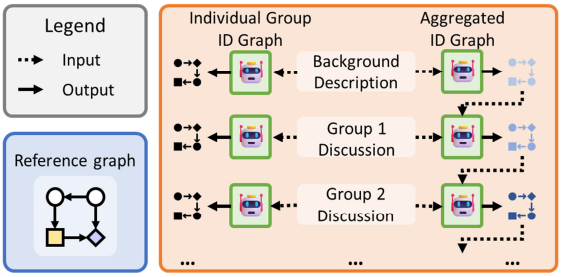


Figure 13 **Application of LAMDA to the case study group discussion transcripts**

To establish a benchmark for evaluation, we engaged an independent coder specializing in decision analysis to construct a reference influence diagram graph (including nodes and edges with probability distributions partially specified according to the problem description). The coder was asked to summarize key information from both the problem description and expert group discussions to develop a comprehensive reference ID graph. She was required to refrain from introducing any nodes or edges based on her personal opinion. We also instructed the coder to align the nodes and edges with the diagnosis and treatment protocol for human avian influenza described in National Health Commission of the People's Republic of China (2024). All disease-related nodes and edges were validated by invited disease control experts. This manually annotated reference ID

graph comprises 27 nodes (including 5 decisions and 7 utilities) and 37 edges. It serves as a reference for comparing different graphs generated from each group's discussions or aggregated using all groups' discussions. See Appendix C, Figure C.1 for the reference ID graph.

For evaluation, we compare the graphs with the reference graph. We investigate whether LAMDA can capture the factors and relations as well as the reference ID graph and, thus, use node/edge recall as the performance metric. We also use the normalized graph edit distance (n-GED) to reflect the similarity between graphs.

4.3. Results

4.3.1. Influence Diagram Graph for Each Group. We first investigate the ID graphs of each group separately. The ID graph of Group 2 omitted several high-impact variables (see Figure 14), including flight reduction (as a decision node) and international relations (as a utility node). The discussions were notably fragmented, where the participants proposed policy suggestions in an uncoordinated manner. Although the members unanimously agreed on the necessity of entry restrictions and converged to a conclusion very quickly, their reasoning lacked depth and substance. In comparison, Group 3 had a larger coverage of decision elements, highlighting the importance of international communication (as a decision node) and humanitarian protection (as a utility node).

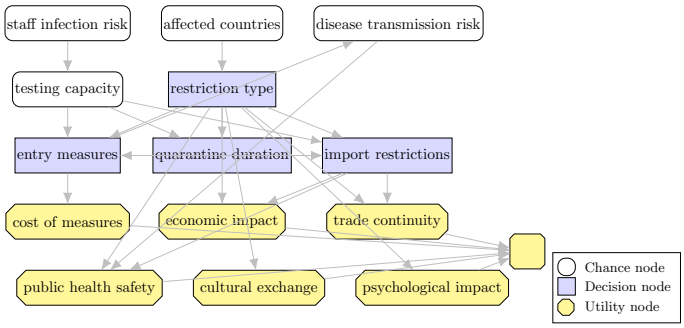


Figure 14 LAMDA-generated influence diagram graph for Group 2.

In contrast, Groups 1 and 4 focused mainly on border restrictions and follow-up measures (see Figure 15). Although addressing similar themes, Group 4 reached a higher level of granularity by considering the cost of medical resources and public acceptance. In addition, experts in Group 4 acknowledged the inherent difficulty of the decision due to the lack of quantitative information. There is a risk of human-to-human transmission, but its severity remains unknown. To address this challenge, they reached some conclusions by ruling out options with unacceptable outcomes.

They emphasized the need for moderation to avoid excessive economic costs. They raised concerns about fairness, particularly regarding travelers from regions without local transmission, and proposed stay-at-home monitoring as a compromise. They mentioned seasonal travel patterns when discussing flight reductions, arguing against blanket restrictions that would prevent students from returning home during peak periods. Instead, they advocated for discouraging nonessential travel.

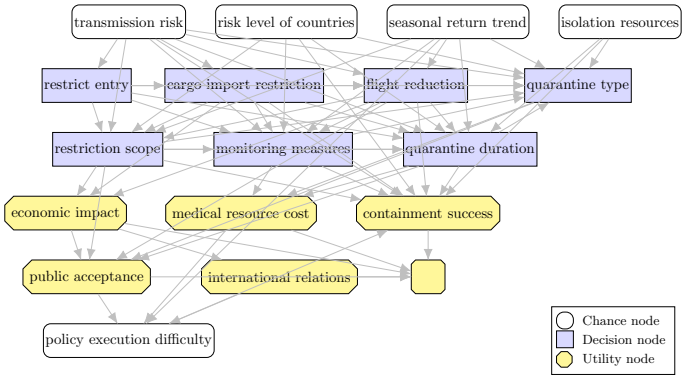


Figure 15 LAMDA-generated influence diagram graph for Group 4.

Here, we also demonstrate how LAMDA performs probability elicitation and decision recommendation without explicit probability statements. For illustration, we use the transcript of Group 2 as the input text, and LAMDA's probability agent elicits the required probability distributions. For example, Table 2 reports the elicited conditional distribution of ''economic impact.'' The elicited distribution suggests that stricter restrictions and broader bans are associated with more negative economic impacts, which is consistent with common expectations. The resulting first-draft influence diagram recommends ''no restriction.'' This outcome arises because the input text did not explicitly specify the relative importance of utility nodes, and, consequently, ''public health safety'' is assigned equal weight to several other considerations in the utility function. When higher weight is assigned to ''public health safety,'' the solution recommends full restriction and a ban on all goods. In practice, the relative importance of attributes should be elicited from stakeholders, with stakeholders' risk attitudes taken into account. Although the graph structure and probability assessments still require careful human validation and calibration, this example illustrates LAMDA's ability to transform less organized text into a structured model that supports subsequent analysis.

Table 2 Conditional Probability of Economic Impact Given Restriction Scenarios

	Import restriction								
	BAN ON ALL GOODS			BAN ON SPECIFIC GOODS			NO RESTRICTIONS		
	Economic impact			Economic impact			Economic impact		
Restriction type	1	0.5	0	1	0.5	0	1	0.5	0
FULL RESTRICTION	0.0	0.3	0.7	0.1	0.4	0.5	0.2	0.5	0.3
CONDITIONAL RESTRICTION	0.3	0.5	0.2	0.4	0.4	0.2	0.6	0.3	0.1
NO RESTRICTION	0.4	0.4	0.2	0.5	0.3	0.2	0.7	0.2	0.1

4.3.2. Aggregated Influence Diagram Graph from All Groups. Previous analysis showed that none of the four individual groups provided a comprehensive view, underscoring the importance of aggregating across groups. Here, we visualize the evolution of the graph when sequentially processing group transcripts, as described in Section 4.2 (see Figure 16). The graph expands in scale at each step, incorporating the nodes and edges newly introduced by each group. The final aggregated ID graph offers a comprehensive understanding of the decision scenario (see Appendix C, Figure C.2).

Now, we compare the aggregated graph to the reference graph. The node recall increased as aggregation proceeded (see Figure 17). The green and red lines indicate the best and worst performance at each step, respectively. The blue line indicates the average performance. Among all possible sequences of aggregation, LAMDA with the best sequence of adding groups covered all nodes of the reference ID graph (see the green line in Figure 17). The variability of performance between different aggregation sequences is pronounced. Therefore, we recommend trying multiple aggregation sequences of the groups in practice to achieve a more diverse perspective. In most cases, LAMDA's output included more nodes than the reference graph, capturing additional nuanced aspects of the decision scenario. This expansive representation offers practical benefits for real-world applications. The more comprehensive graph could facilitate thorough inspection and iterative refinement, while the inclusion of potentially relevant factors can stimulate richer discussions and debates.

On the other hand, edge recall also increased as aggregation progressed (see Figure 18). The green and red lines indicate the best and worst recall levels at each step, respectively. The blue line represents the average along with its standard errors. Edge recall values were generally low for two reasons. First, the generated graphs and the reference graph worked at different levels of abstraction. When a node in the reference graph corresponds to multiple, more nuanced nodes in a generated graph, it is challenging to perfectly match the edges between the nodes in the two graphs. Second, establishing dependency relations is inherently difficult in the absence of detailed quantitative data or causal statements. For practical implementation, we recommend explicitly guiding the discussion

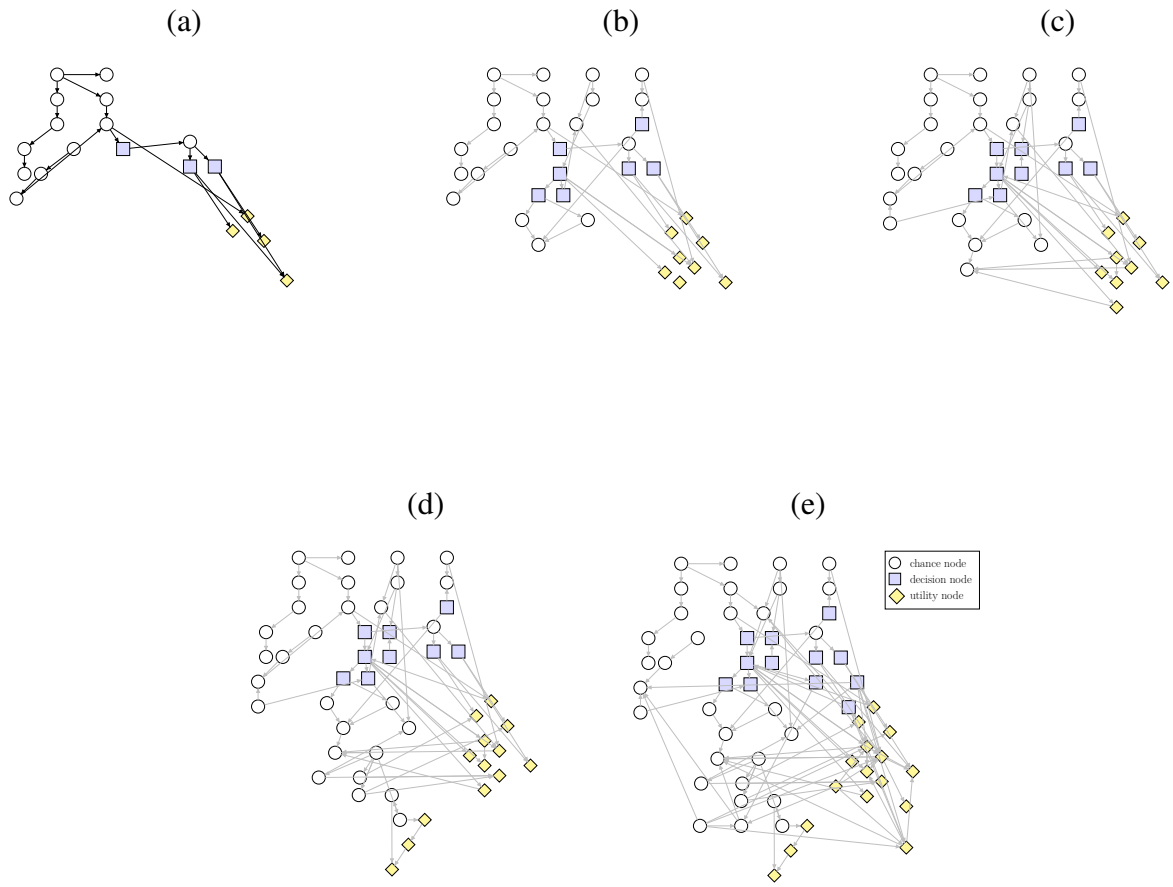


Figure 16 **Updated ID graphs as more groups of discussions were aggregated.**

Notes. (a) Prior graph. (b) Aggregation Step 1 graph. (c) Aggregation Step 2 graph. (d) Aggregation Step 3 graph. (e) Aggregated graph.

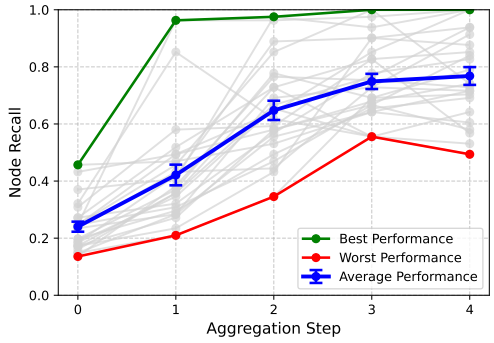


Figure 17 **The changes of node recall as aggregation proceeds.**

to provide detailed descriptions of the dependencies between factors, utilizing the automatically extracted ID graph as a basis for review and refinement.

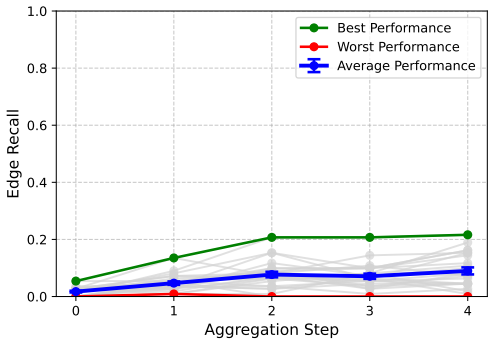


Figure 18 The changes of edge recall as aggregation proceeds.

A critical step in decision analysis is problem framing, which includes the generation of objectives. Here, we specifically focus on the utility nodes generated by LAMDA, comparing them with the set of objectives covered by the reference graph. We use the criteria from Simon and Siebert (2025), designed based on Keeney (1996). For example, the objectives should be essential, controllable, measurable, operational, and understandable. Specifically, the set of objectives should be complete and concise, and nonredundant. The reference graph includes a concise, yet complete, set of utilities, covering essential aspects: public health, international relations, public sentiment, and economic loss, etc. In contrast, the aggregated graph includes more utility nodes, and each is individually viable. However, there are limitations when evaluating them as a whole. The utility nodes are complete, but not concise, in the sense that similar objectives are not grouped together. For example, it includes ‘‘economic impact of travel restrictions’’ and ‘‘economic impact of trade restrictions,’’ which can be merged into a single node. Furthermore, some utility nodes are nonessential for the decision problem. For example, ‘‘impact of restriction on community mobility’’ may not be a major consideration when facing the pandemic control decision.

The first-draft influence diagram generated by LAMDA faces similar challenges in objective generation, as pointed out in a recent study (Simon and Siebert 2025): the generated objectives can be viable individually but flawed as a whole. This highlights the continued importance of human expertise in problem framing, which can substantially affect decision quality. Accordingly, improved prompting practices and systematic human-in-the-loop refinement, as suggested by Simon and Siebert (2025), remain essential.

Last, but not least, for real-world problems, even professional human analysts can miss out on key factors. Therefore, we also utilized open-ended evaluation with LLM-as-a-judge (Gu et al. 2026) to assess the generated graphs. Here, the reference graph serves as a benchmark instead of the

‘‘ground truth.’’ The results show that the aggregated graphs outperformed the reference graph and the graphs generated from individual group transcripts in terms of realism and relevance. This illustrates LAMDA’s potential in processing massive amounts of raw data and generating a comprehensive first-draft influence diagram, which serves as a basis for brainstorming, modeling, and prescription. Details of the assessment are presented in Appendix C, Section C.2.

5. Discussion and Future Work

In this paper, we introduce LAMDA, a novel framework that uses LLMs to assist in constructing influence diagrams from natural language input. To support systematic evaluation, we also develop a new data set that pairs textual decision scenarios with ground-truth influence diagrams. Across controlled experiments, LAMDA reliably identifies key decision elements and probabilistic relationships and yields better decision recommendations than vanilla LLMs and standard prompting-based enhancement techniques. A case study on pandemic control further illustrates its practical value for structuring expert discussions in high-stakes, uncertain settings.

LAMDA can substantially reduce the analyst’s burden by synthesizing large volumes of unstructured text into a candidate influence diagram. However, this output should be treated as a starting point, rather than ground truth. Decision analysts remain essential for orchestrating the ‘‘decision analysis cycle” (Matheson and Matheson 2005), validating assumptions, revising models, and translating model-based insights into recommendations that are relevant and actionable for stakeholders. In this sense, LAMDA is best viewed as a tool that supports, rather than replaces, the broader decision analysis process, as illustrated in Figure 19.

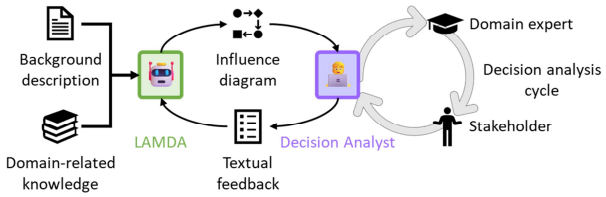


Figure 19 Decision analysis cycle with LAMDA supporting human decision analyst.

Objective generation and problem framing are crucial components of decision analysis and an important research topic. Recent work has examined LLMs’ ability to identify objectives in organizational and policy settings, highlighting limitations such as introducing nonessential considerations while omitting important goals (Simon and Siebert 2025). Abbas et al. (2026) also show that

humans and LLMs have complementary strengths and weaknesses in generating decision inputs. These findings suggest that generative AI tools can support problem structuring, but human knowledge and expertise remain essential. We view this line of research as complementary to LAMDA's contribution to model construction, and an important direction for future work is to integrate these stages into a more end-to-end decision support pipeline.

Probability and preference elicitation is another important direction for future work. Probability elicitation is challenging even for human analysts (Falconer et al. 2022), requiring close collaboration with domain experts, targeted questions about key variables, and careful calibration to mitigate bias. Although LLMs are useful for extracting probabilistic information from large volumes of unstructured text, their outputs may reflect pretraining biases or hallucinations and, therefore, should not be treated as expert priors without scrutiny. A promising next step is to incorporate systematic human-in-the-loop refinement, where decision analysts treat LAMDA's output as an initial draft and provide textual feedback to iteratively revise the model and produce more defensible belief representations. Furthermore, stakeholder preferences are essential for decision analysis. The current framework assumes risk-neutral decision makers, whereas in practice, such preferences also require careful elicitation. In addition, although posterior updating can be performed by standard Bayesian inference once a valid graph and parameterization are obtained, integrating evidence extraction from narrative text for subsequent belief updating remains an important area for future work.

Several technical extensions could further improve LAMDA. The current framework follows a fixed sequence of steps based on standard modeling practice, but a more adaptive workflow may yield better results. For example, difficulties encountered during probability generation may indicate that earlier parts of the graph, such as missing or inappropriate edges, should be revised. A promising direction is to introduce an LLM-based coordinating agent that selects the next modeling step based on the current diagram and the available text, rather than following the fixed procedure (Yao et al. 2023, Ye et al. 2023, Xiao et al. 2024). In addition, the current rule-based checks mainly detect explicit errors. Future work could also use self-reflection and self-refinement to improve the model, even when no clear error is present (Madaan et al. 2023), for example, by identifying mismatches between the generated diagram and the source text and proposing targeted revisions.

Overall, LAMDA takes a step toward making decision modeling more scalable in settings with large volumes of unstructured information. We hope this work encourages further research on integrating LLMs into normative decision analysis, especially through interactive workflows that connect model construction, refinement, and evidence-based updating.

A. Details on LAMDA Framework

A.1. Prompts for Sequential Graph Generation

Here, we present the prompt LAMDA uses for generating influence diagrams in a sequential manner.

Prompt for Node Extraction

You are an expert decision analyst specializing in decision making under uncertainty and influence diagram modeling. For the following text, extract every decision-related variable that supplements the original list. The variables may correspond to uncertainty, action or utility. Output the added list of nodes.
Text:
{text}
The original node list:
{node_list}
Requirements:

1. Do not include nodes in the original node list or have similar meanings to existing nodes in the node list.
2. The variables should be directly mentioned or clearly implied in the text.
3. The variable names should be concise and meaningful and should come from the source text if possible.
4. Utility node values should be numerical. We aim to maximize the utility, so if a utility node is cost-related, the values should be negative. If a utility node is benefit-related, the values should be positive.
5. Output using the following format:

`{format_instructions}`

Prompt for Edge Extraction

You are an expert decision analyst specializing in decision making under uncertainty and influence diagram modeling. For the following text, extract every influence or causal relation (edge) between variables (nodes) that supplements the original edge list. Output the added list of edges.
Understanding edges in influence diagrams:

- An edge ending in a decision node specifies information available when making that decision
- An edge ending in a chance node or a utility node specifies probabilistic dependencies
- The absence of edges between nodes implies conditional independence

Examples:

- Edge {{'cause': 'weather forecast', 'effect': 'take umbrella'}} ends in a decision node 'take umbrella', meaning the value of 'weather forecast' is known when deciding whether to take the umbrella.
- Edge {{'cause': 'weather', 'effect': 'weather forecast'}} ends in a chance node 'weather forecast', meaning the distribution of weather forecast depends on the weather (conditional probability P(weather forecast | weather)).

Text:
{text}
The original edge list:
{edge_list}
Requirements:

1. Do not include edges in the original edge list or edges with similar meanings.
2. The edges should be directly mentioned or clearly implied in the text.
3. No cycles are allowed. For example, if there is an edge from 'a' to 'b', there cannot be another edge from 'b' to 'a'.
4. All cause and effect variables (nodes) used in the edge list must be members of the node list:

`{node_list}`

5. Output using the following format:

`{format_instructions}`

B. Experiment Details

B.1. Data Set Construction

We provide an example heatmap (Figure B.1) that shows that no action is dominated across instances in terms of expected utility.

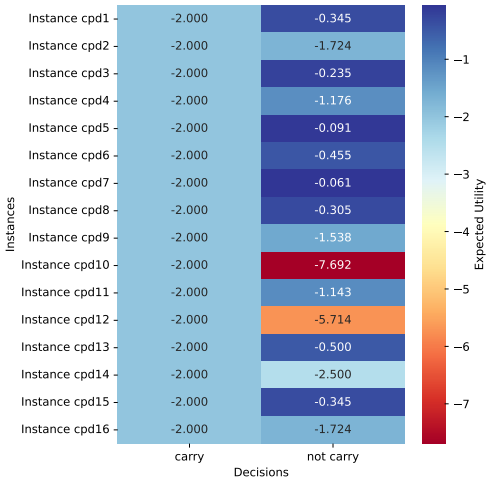


Figure B.1 Expected utility of options when the weather forecast predicts 'no rain'.

B.2. Prompts for Matching Nodes Semantically

Prompts for Matching Nodes Semantically

Your goal is to find the target nodes that are present in the generated node list. Node names may not match exactly, but if they have the same meaning or refer to the same concept, they should be considered the same node.
Based on a target node list and a generated node list, output a list of tuples, each containing a target node name and a generated node name. You should carefully analyze both lists to identify semantic matches even when the wording differs significantly.
Target node list:
{target_nodes}
Generated node list:
{generated_nodes}
Requirements:

1. Output pairs of node names (target node, generated node) that represent the same concept or meaning.
2. Use exact node names as they appear in their respective lists - do not modify them.
3. Two nodes match if they have the same meaning or refer to the same concept.
4. Be strict in your matching criteria - avoid false positives.
5. Each target node and generated node can appear **at most once** in your output.
6. Do not include target nodes that have no match in the generated list.
7. Output using the following format:

`{format_instructions}`

B.3. Performance of LAMDA on ID Graph Generation

In the first experiment, we evaluate LAMDA's performance in generating influence diagram graphs from textual inputs. We utilize LAMDA to generate graphs for the descriptions of decision problems, incorporating background, node, and graph-specific details. For each text sample, we record the results of five independent trials.

B.3.1. Backbone LLMs We use three LLMs as the backbone models of LAMDA: the open-source `Qwen2.5-7B` with 7 billion parameters, `Qwen2.5-72B` with 72 billion parameters (Yang et al. 2024), and closed-source `GPT-4o-2024-11-20` (Hurst et al. 2024). The choice is based on two considerations. Both the `Qwen2.5` series models and `GPT-4o` allow for long context windows

of up to 128k tokens. In addition, these models have outstanding multilingual capabilities, enabling them to also work in diverse language environments.

B.3.2. Metrics We frame influence diagram graph generation as a classification task. The nodes and edges of the ground-truth graph constitute positive samples. Including these nodes or edges in the constructed graph represents ''true positives," whereas omitting them results in ''false negatives." Conversely, including nodes or edges absent from the ground-truth graph yields ''false positives." We employ standard classification metrics to evaluate the performance of node and edge generation. Specifically, we evaluate the precision, recall, and F1-score of the classification task. Because of the presence of synonyms, exact string matching proves inadequate for determining node presence in a list. Instead, we employ `GPT-4o-2024-11-20` to identify nodes that match semantically.

Classification metrics assess only the proportion of ''ground-truth" nodes or edges present in the generated graph. To assess structural discrepancies, we also consider the graph edit distance (GED). GED quantifies the differences between two graphs by counting the edits required to align them. We consider deletion, insertion, and substitution of nodes/edges as basic edit operations. We assign these edit operations equal weights and normalize the GED by the size of the graph, resulting in the normalized graph edit distance (n-GED).

B.3.3. Results We present the performance of node extraction for the 302 decision problems across different information levels, for joint and sequential generation, respectively, in Figure B.2. The points and error bars represent means and standard deviations from five replications per instance. Generally, LAMDA was able to identify key factors and dependencies in these decision problems. The average metric values (including precision, recall, and F1 score) are close to one, and the small variation across replications indicates the stability of the performance. Even smaller open-source LLMs (`Qwen2.5-7B`) could effectively extract graph components from text, whereas larger models (`Qwen2.5-72B` and `GPT-4o`) generally achieved superior performance. Meanwhile, the choice of joint or sequential extraction approach did not significantly affect graph generation performance.

Specifically, the node extraction performance was nearly perfect, especially when the graph information (the third level of information in the problem description, as explained in Section 3.1) was provided. We notice that `GPT-4o` had higher recall and lower precision when extracting nodes compared with `Qwen2.5` series models. This indicates that `GPT-4o` identified a broader set of factors than necessary, including factors that are not considered in the ground-truth IDs.

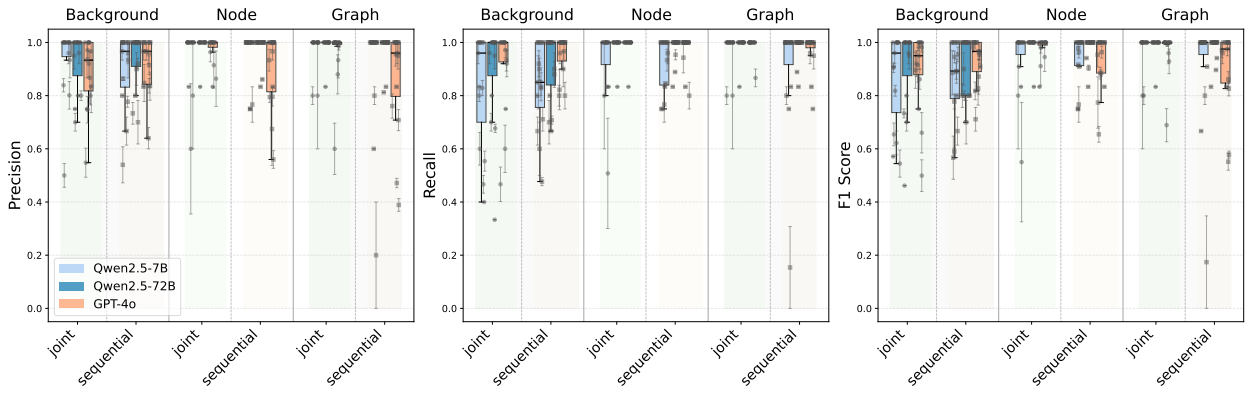


Figure B.2 Node extraction performance (precision, recall, F1 score) across different information levels.

Furthermore, the correct identification of node types—namely, the decision nodes, utility nodes, and chance nodes—is critical for constructing a useful influence diagram. LAMDA achieves high accuracy in classifying node types in most cases. As shown in Table B.1, the average accuracy (± standard deviation (SD)) for correctly classifying the nodes is close to one for all levels of information, with coefficients of variation provided for the five replications. Larger LLMs consistently outperformed `Qwen2.5-7B`, though `GPT-4o` had greater variation in performance compared with `Qwen2.5-72B`.

Table B.1 Node type accuracy and coefficients of variation (CV) (mean±SD).

		Background info.		Node info.		Graph info.	
Method	Model	Acc ↑	CV ↓	Acc ↑	CV ↓	Acc ↑	CV ↓
Joint	Qwen2.5-7B	0.96±0.12	0.01±0.04	0.90±0.17	0.16±0.34	0.94±0.11	0.12±0.23
	Qwen2.5-72B	1.00±0.02	0.01±0.04	1.00±0.02	0.00±0.00	1.00±0.02	0.00±0.00
	GPT-4o	0.96±0.07	0.04±0.06	0.98±0.07	0.01±0.05	0.99±0.03	0.01±0.04
Sequential	Qwen2.5-7B	0.89±0.16	0.13±0.19	0.94±0.11	0.07±0.15	0.87±0.24	0.17±0.58
	Qwen2.5-72B	0.94±0.11	0.01±0.03	0.95±0.10	0.00±0.01	0.95±0.10	0.00±0.00
	GPT-4o	0.97±0.07	0.04±0.06	0.97±0.09	0.01±0.02	0.97±0.09	0.01±0.02

Edge extraction was more challenging for the LLMs we considered. The boxplots in Figure B.3 display the distribution of performance metrics related to edge generation across various decision problems. Points and error bars again represent means and standard deviations from five replications per instance. When provided with graph information, `Qwen2.5-72B` and `GPT-4o` were able to recover most of the edges, whereas `Qwen2.5-7B` performed worse than larger models. Consider the simple ‘‘carry umbrella” example, which involves four key relationships in the ground truth: the real weather influences both the overall outcome and the weather forecast; the decision is based on the weather forecast, and also affects the outcome, because carrying an umbrella on a sunny day

causes inconvenience, whereas not carrying one on a rainy day may lead to greater consequences. When given background information, LAMDA with `GPT-4o` as the backbone LLM accurately constructed the correct graphical model, whereas `Qwen2.5-7B` occasionally makes errors. For example, in some instances, it only accounted for the cost of carrying an umbrella, overlooking the cost of getting caught in the rain and missing the connection between the actual weather and the overall cost.

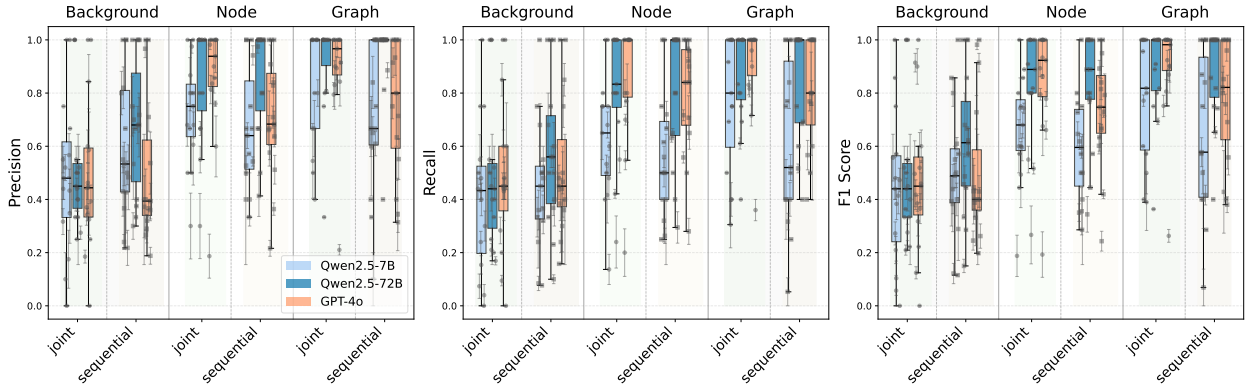


Figure B.3 **Edge extraction performances (precision, recall, F1 score) across different information levels.**

The richness of the input information had a significant impact on overall performance. As more detailed information was provided, from background to node to graph information, we observed substantial improvements in all performance metrics. The boxplots in Figure B.4 show the distribution of n-GED across different decision problems. Explicitly stating the involved nodes and edges reduces ambiguity in the background description and supports precise translation into graphical models.

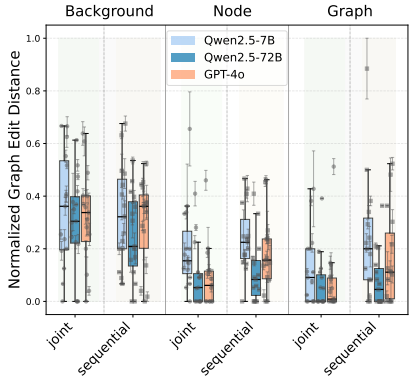


Figure B.4 **Normalized graph edit distance (n-GED) across different information levels.**

Here, we also report the mean performance (± standard deviation) of influence diagram extraction (Table B.2) and the coefficient of variation (± standard deviation) of the five replications (Table B.3). The results demonstrate that the performance is reliable and stable across replications.

Table B.2 Mean performance (± SD) in influence diagram graph extraction.

		Node extraction			Edge extraction			Graph
Method	Model	Precision ↑	Recall ↑	F1 ↑	Precision ↑	Recall ↑	F1 ↑	n-GED ↓
Background information								
Sequential	Qwen2.5-7B	0.90 ± 0.14	0.84 ± 0.16	0.86 ± 0.15	0.60 ± 0.27	0.45 ± 0.24	0.50 ± 0.25	0.33 ± 0.19
Joint	Qwen2.5-7B	0.94 ± 0.14	0.83 ± 0.22	0.87 ± 0.18	0.49 ± 0.29	0.41 ± 0.29	0.44 ± 0.29	0.35 ± 0.22
Sequential	Qwen2.5-72B	0.94 ± 0.11	0.91 ± 0.13	0.92 ± 0.11	0.68 ± 0.25	0.57 ± 0.28	0.61 ± 0.27	0.25 ± 0.17
Joint	Qwen2.5-72B	0.93 ± 0.11	0.89 ± 0.19	0.91 ± 0.16	0.53 ± 0.26	0.49 ± 0.29	0.51 ± 0.27	0.29 ± 0.19
Sequential	GPT-4o	0.92 ± 0.10	0.95 ± 0.08	0.93 ± 0.09	0.50 ± 0.28	0.52 ± 0.26	0.50 ± 0.27	0.29 ± 0.17
Joint	GPT-4o	0.89 ± 0.13	0.91 ± 0.17	0.89 ± 0.15	0.48 ± 0.28	0.49 ± 0.29	0.48 ± 0.28	0.32 ± 0.18
Node information								
Sequential	Qwen2.5-7B	0.95 ± 0.10	0.92 ± 0.10	0.94 ± 0.10	0.66 ± 0.23	0.56 ± 0.23	0.59 ± 0.20	0.24 ± 0.13
Joint	Qwen2.5-7B	0.95 ± 0.12	0.93 ± 0.14	0.94 ± 0.13	0.73 ± 0.20	0.63 ± 0.23	0.66 ± 0.20	0.20 ± 0.16
Sequential	Qwen2.5-72B	0.98 ± 0.05	0.98 ± 0.05	0.98 ± 0.05	0.85 ± 0.20	0.81 ± 0.25	0.82 ± 0.21	0.11 ± 0.12
Joint	Qwen2.5-72B	0.99 ± 0.04	0.99 ± 0.04	0.99 ± 0.04	0.85 ± 0.22	0.81 ± 0.24	0.82 ± 0.22	0.09 ± 0.12
Sequential	GPT-4o	0.90 ± 0.14	0.97 ± 0.06	0.93 ± 0.11	0.71 ± 0.23	0.79 ± 0.22	0.73 ± 0.21	0.18 ± 0.14
Joint	GPT-4o	0.97 ± 0.06	0.99 ± 0.04	0.98 ± 0.05	0.86 ± 0.22	0.85 ± 0.23	0.85 ± 0.22	0.09 ± 0.12
Graph information								
Sequential	Qwen2.5-7B	0.91 ± 0.23	0.90 ± 0.22	0.90 ± 0.22	0.71 ± 0.28	0.61 ± 0.31	0.65 ± 0.30	0.24 ± 0.23
Joint	Qwen2.5-7B	0.97 ± 0.07	0.97 ± 0.07	0.97 ± 0.07	0.83 ± 0.22	0.77 ± 0.25	0.79 ± 0.24	0.12 ± 0.14
Sequential	Qwen2.5-72B	0.98 ± 0.06	0.98 ± 0.05	0.98 ± 0.05	0.94 ± 0.16	0.84 ± 0.21	0.88 ± 0.18	0.08 ± 0.11
Joint	Qwen2.5-72B	0.99 ± 0.04	0.99 ± 0.04	0.99 ± 0.04	0.91 ± 0.18	0.86 ± 0.18	0.88 ± 0.18	0.07 ± 0.11
Sequential	GPT-4o	0.87 ± 0.20	0.97 ± 0.07	0.90 ± 0.15	0.76 ± 0.25	0.81 ± 0.21	0.78 ± 0.23	0.16 ± 0.18
Joint	GPT-4o	0.96 ± 0.11	0.99 ± 0.03	0.97 ± 0.08	0.89 ± 0.20	0.91 ± 0.18	0.89 ± 0.19	0.07 ± 0.13

B.4. Performance of LAMDA on Probability Elicitation

In the second experiment, we evaluate LAMDA's performance in assigning probability distributions based on explicit statements about probabilities in the text. To isolate the performance of extracting probability distributions from the graph-building task, we start with the ground-truth nodes and edges in this experiment. For consistency with the first experiment, we use the same backbone LLMs: `Qwen2.5-7B`, `Qwen2.5-72B`, and `Gpt-4o-2024-11-20`.

Although implicit, ambiguous, or missing probability statements are common in real-world text, in our experiments, we focus on cases with explicit probability statements. This design choice isolates the LLM's capability to formalize stated information into probability distributions, rather than conflating performance with differences in internal knowledge. In practical deployments, LAMDA may begin with prior estimates or fall back to uninformative (e.g., uniform) defaults, after which decision analysts and subject-matter experts review and refine the resulting distributions. Humans can provide interactive feedback in natural language, where explicit statements can be incorporated

Table B.3 Coefficient of variation (± SD) in influence diagram graph extraction.

		Node extraction			Edge extraction			Graph
Method	Model	Precision	Recall	F1	Precision	Recall	F1	n-GED
Background information								
Sequential	Qwen2.5-7B	0.06 ± 0.09	0.10 ± 0.11	0.08 ± 0.09	0.33 ± 0.30	0.29 ± 0.28	0.30 ± 0.28	0.23 ± 0.18
Joint	Qwen2.5-7B	0.04 ± 0.07	0.05 ± 0.08	0.05 ± 0.07	0.46 ± 0.63	0.46 ± 0.62	0.46 ± 0.62	0.13 ± 0.13
Sequential	Qwen2.5-72B	0.04 ± 0.08	0.04 ± 0.08	0.04 ± 0.08	0.15 ± 0.19	0.14 ± 0.19	0.15 ± 0.19	0.20 ± 0.24
Joint	Qwen2.5-72B	0.02 ± 0.04	0.02 ± 0.04	0.02 ± 0.04	0.20 ± 0.36	0.20 ± 0.36	0.20 ± 0.36	0.18 ± 0.20
Sequential	GPT-4o	0.06 ± 0.06	0.05 ± 0.07	0.06 ± 0.06	0.37 ± 0.35	0.34 ± 0.35	0.35 ± 0.35	0.47 ± 0.58
Joint	GPT-4o	0.09 ± 0.10	0.06 ± 0.11	0.08 ± 0.09	0.28 ± 0.19	0.26 ± 0.20	0.28 ± 0.18	0.39 ± 0.38
Node information								
Sequential	Qwen2.5-7B	0.01 ± 0.05	0.03 ± 0.06	0.02 ± 0.05	0.16 ± 0.37	0.18 ± 0.36	0.17 ± 0.36	0.16 ± 0.24
Joint	Qwen2.5-7B	0.10 ± 0.27	0.10 ± 0.27	0.10 ± 0.27	0.18 ± 0.27	0.17 ± 0.27	0.17 ± 0.27	0.26 ± 0.36
Sequential	Qwen2.5-72B	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.01	0.03 ± 0.11	0.04 ± 0.12	0.04 ± 0.11	0.04 ± 0.08
Joint	Qwen2.5-72B	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.08 ± 0.24	0.08 ± 0.24	0.08 ± 0.24	0.07 ± 0.11
Sequential	GPT-4o	0.06 ± 0.12	0.02 ± 0.05	0.04 ± 0.08	0.17 ± 0.14	0.14 ± 0.14	0.15 ± 0.13	0.66 ± 0.76
Joint	GPT-4o	0.04 ± 0.09	0.00 ± 0.00	0.02 ± 0.05	0.14 ± 0.26	0.11 ± 0.26	0.12 ± 0.26	0.59 ± 0.66
Graph information								
Sequential	Qwen2.5-7B	0.16 ± 0.58	0.16 ± 0.58	0.16 ± 0.58	0.26 ± 0.58	0.27 ± 0.59	0.26 ± 0.58	0.26 ± 0.38
Joint	Qwen2.5-7B	0.04 ± 0.14	0.04 ± 0.14	0.04 ± 0.14	0.07 ± 0.17	0.07 ± 0.18	0.07 ± 0.18	0.17 ± 0.24
Sequential	Qwen2.5-72B	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.02	0.00 ± 0.02	0.00 ± 0.02	0.01 ± 0.04
Joint	Qwen2.5-72B	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.02	0.00 ± 0.01	0.01 ± 0.02
Sequential	GPT-4o	0.05 ± 0.06	0.01 ± 0.04	0.04 ± 0.04	0.22 ± 0.27	0.16 ± 0.26	0.19 ± 0.25	0.74 ± 0.81
Joint	GPT-4o	0.05 ± 0.10	0.01 ± 0.02	0.03 ± 0.06	0.08 ± 0.10	0.06 ± 0.09	0.07 ± 0.08	1.04 ± 0.82

by LAMDA using the same elicitation procedure evaluated here. Enabling this human-in-the-loop refinement in a systematic way is an important direction for future work.

B.4.1. Metrics We quantify the deviations between the generated and ground-truth distributions by comparing their joint distributions. Four classical metrics are used to measure the discrepancy between the two distributions, including Brier score, total variation distance, Kullback-Leibler (KL) divergence, and Jensen-Shannon (JS) divergence. We report the means and standard deviations of these metrics for the 302 decision problems.

B.4.2. Results LAMDA successfully translated text statements about distributions into executable Python lambda functions (see Table B.4). Among the three backbone models, `GPT-4o` produced distributions most similar to the ground-truth distributions, whereas `Qwen2.5-7B`'s results are less reliable. For the simple ‘‘carry umbrella" example, the distributions generated by `GPT-4o` exactly matched the ground truth. `Qwen2.5-7B` did not follow correct probabilistic logic, resulting in a flawed probability representation because only one branch was evaluated (see Figure B.5).

Meanwhile, we also observed room for improvement in the accuracy of probability generation, even for large LLMs. In some cases, `GPT-4o` misunderstood algebraic operations, leading to expressions that do not match the given description. See the flawed calculation of the total cost in the long-term care insurance problem in Figure B.6. The total cost depends on whether expenses exceed

lambda carry_umbrella_decision, weather_outcome: {0.0: 1.0 if (carry_umbrella_decision == 'not carry' and weather_outcome == 'rain') else 0.0, -2.0: 1.0 if carry_umbrella_decision == 'carry' else 0.0, -25.0: 1.0 if (carry_umbrella_decision == 'not carry' and weather_outcome == 'no rain') else 0.0}

Figure B.5 Distributions generated by `Qwen2.5-7B` for the "carrying umbrella" problem.

the maximum coverage and requires summing up multiple terms, which `GPT-4o` missed. It underscores the need to develop more robust elicitation methods for LLMs in future work.

Table B.4 Probability distribution extraction performance over various decision problems (Mean ± SD).

Model	Brier score ($\times 10^{-4}$) ↓	Total variation distance ↓	KL-divergence ↓	JS-divergence ↓
Qwen2.5-7B	8.10 ± 23.23	0.50 ± 0.32	9.84 ± 8.95	0.30 ± 0.24
Qwen2.5-72B	1.27 ± 3.26	0.31 ± 0.36	6.45 ± 9.97	0.18 ± 0.27
GPT-4o	0.76 ± 2.16	0.18 ± 0.28	3.76 ± 7.00	0.11 ± 0.19

- If the long-term care expense per day exceeds the maximum coverage per day, the total cost is calculated as (life expectancy - age) * (- 365 * long-term care expense per day - insurance premium per year + 365 * maximum coverage per day).
- Otherwise, the maximum coverage per day can cover the long-term care expense per day. The cost is calculated as (life expectancy - age) * (-insurance premium per year).

Figure B.6 The calculation of total cost in the "long-term care insurance" problem.

B.5. Performance of LAMDA on Decision Making under Risk

Here, we display the results of normalized rank (see Figure B.7).

B.6. DeepSeek-R1 Reasoning Process on an Example Instance

We also present the reasoning process of `DeepSeek-R1` on the chemical usage problem (see Figure B.8).

C. Case Study Details

C.1. Representative Influence Diagram Graphs

Here, we present the reference graph annotated by a trained decision analyst (see Figure C.1). We also present an example of aggregated graph, produced by applying LAMDA to multiple group inputs sequentially (see Figure C.2).

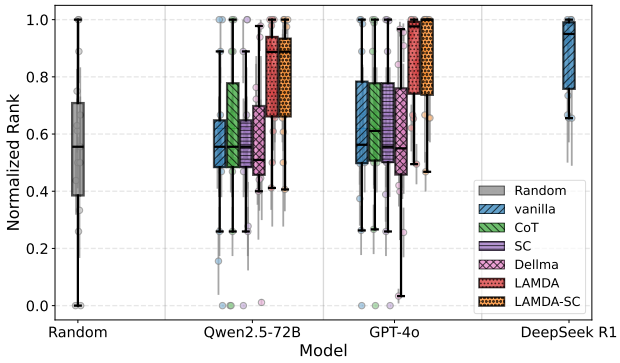


Figure B.7 Normalized rank over decision problems.

Notes. The boxplots show the distribution of performance for each method across decision problems. The scatter points and error bars represent the means and standard deviations of different CPD instances for each problem.

C.2. Open-Ended Assessment of Influence Diagrams

Previously, we evaluated the generated graphs by comparing them to a manually annotated reference graph. However, human analysts may also leave out key considerations, and the reference graph may not be the ‘‘ground truth”. Here, we employ a more open-ended evaluation method using LLM-as-a-judge (Gu et al. 2026). Specifically, we used `GPT-4o`, one of the most powerful LLMs to date, to evaluate the generated graphs in terms of ‘‘realism” and ‘‘relevance”. The graphs we evaluated include the following (in Table C.1, see the metadata of the ID graphs):

- The reference graph.
- Two aggregated graphs were selected from the 24 possible aggregation paths, one with the highest n-GED (the least similar) and the other with the lowest n-GED (the most similar) to the reference graph.
- The two prior graphs initiated the sequential aggregation and resulted in the aggregated graphs that were most and least similar to the reference ID graph, respectively. Notice that all these prior graphs were based on the same problem description. However, the output varied when using `GPT-4o` as the backbone LLM, even with `temperature = 0` (see also Figure 17 for the evident variability).
- Four graphs were extracted separately for each individual group using LAMDA, generated solely from their discussion transcripts.

Notice that the reference graph is no longer the ground truth, but a benchmark in the evaluation instead. We adopt a pairwise comparison paradigm using `GPT-4o` as the judge. In each trial, two graphs were selected, anonymized, and presented in randomized order to the LLM judge. The judge

...
For each decision alternative, I need to compute the expected net value.
Net value = Economic value + Cancer cost
Since net value is simply the sum, and both economic value and cancer cost are in the same units (I assume monetary units), I can work with expected values.
Let's define the decision alternatives:

1. Ban
2. Restrict
3. Permit

For each, I'll find the expected economic value and the expected cancer cost, then sum them to get expected net value.
...

The expected net value for each decision is computed as: - **Expected Net Value** = **Economic Value** + **Expected Cancer Cost**
Where the expected cancer cost is derived by considering all combinations of carcinogenic activity and human exposure, weighted by their probabilities.

1. **Decision: Ban**

- Economic value: -1.0 (fixed)
- Cancer cost: 0.0 (fixed, regardless of activity or exposure)
- **Expected net value = -1.0 + 0.0 = -1.0**

2. **Decision: Restrict**

- Economic value: 1.0 (fixed)
- Expected cancer cost calculation:
  - If carcinogenic activity is inactive (probability 0.50): cancer cost = 0.0
  - If moderate (probability 0.40): expected cancer cost depends on exposure:
  - Low exposure (0.05): cost = -0.05
  - Medium exposure (0.45): cost = -0.50
  - High exposure (0.50): cost = -5.00
  - Expected cost given moderate activity: (0.05 × -0.05) + (0.45 × -0.50) + (0.50 × -5.00) = -0.0025 - 0.225 - 2.5 = -2.7275
  - If very active (probability 0.10): expected cancer cost depends on exposure:
  - Low exposure (0.05): cost = -1.00
  - Medium exposure (0.45): cost = -10.00
  - High exposure (0.50): cost = -100.00
  - Expected cost given very active activity: (0.05 × -1.00) + (0.45 × -10.00) + (0.50 × -100.00) = -0.05 - 4.5 - 50 = -54.55
  - Overall expected cancer cost: (0.50 × 0.0) + (0.40 × -2.7275) + (0.10 × -54.55) = 0 - 1.091 - 5.455 = -6.546
- **Expected net value = 1.0 + (-6.546) = -5.546**

3. **Decision: Permit**

- Economic value: 5.0 (fixed)
- Expected cancer cost calculation:
  - If carcinogenic activity is inactive (probability 0.50): cancer cost = 0.0
  - If moderate (probability 0.40): expected cancer cost depends on exposure:
  - Low exposure (0.05): cost = -0.50
  - Medium exposure (0.45): cost = -5.00
  - High exposure (0.50): cost = -50.00
  - Expected cost given moderate activity: (0.05 × -0.50) + (0.45 × -5.00) + (0.50 × -50.00) = -0.025 - 2.25 - 25 = -27.275
  - If very active (probability 0.10): expected cancer cost depends on exposure:
  - Low exposure (0.05): cost = -10.00
  - Medium exposure (0.45): cost = -100.00
  - High exposure (0.50): cost = -1000.00
  - Expected cost given very active activity: (0.05 × -10.00) + (0.45 × -100.00) + (0.50 × -1000.00) = -0.5 - 45 - 500 = -545.5
  - Overall expected cancer cost: (0.50 × 0.0) + (0.40 × -27.275) + (0.10 × -545.5) = 0 - 10.91 - 54.55 = -65.46
- **Expected net value = 5.0 + (-65.46) = -60.46**

Figure B.8 Reasoning process of `DeepSeek-R1` on the chemical usage problem.

Table C.1 Number of nodes and edges in the graphs evaluated in the pairwise comparison tournament

Graph component	Reference	Most similar aggregated	Least similar aggregated	Most similar prior	Least similar prior	Group 1	Group 2	Group 3	Group 4
No. of nodes	27	53	59	14	19	15	14	17	17
Chance	15	23	31	8	12	5	4	4	5
Decision	5	12	12	3	3	7	4	7	7
Utility	7	18	16	3	4	3	6	6	5
No. of edges	37	96	102	12	19	20	15	29	38

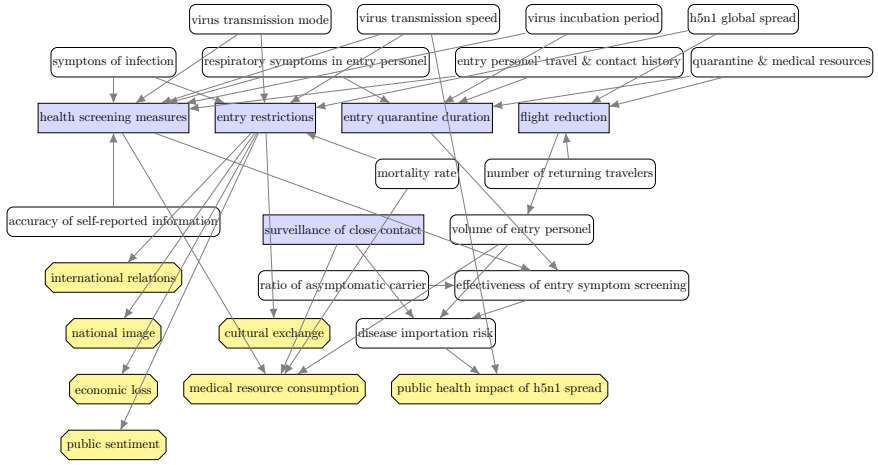


Figure C.1 The reference ID graph.

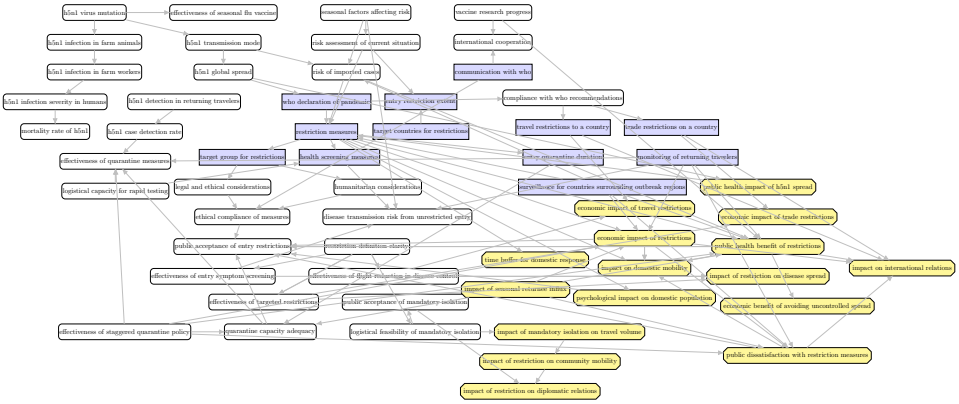


Figure C.2 Example of LAMDA aggregated graph from all group transcripts.

then selected the more ‘‘realistic” and the more ‘‘relevant” graph. Each pair was compared over 10 trials per tournament run, with 5 replicate runs of the tournament. We record the results of each comparison and compare the win rates.

C.2.1. Results The results show that the aggregated graphs outperformed the reference ID graph (see Figure C.3; error bars indicate the standard errors). Meanwhile, the reference graph beat the prior graphs and graphs of individual groups. According to the LLM judge, the LAMDA-generated graphs from expert discussions provided a more in-depth insight than the reference ID graph, including more factors such as WHO recommendations, ethical considerations, psychological impacts, and international cooperation. In addition, the aggregated graphs (including the graphs most and least similar to the reference ID graph) significantly outperformed the prior graphs and the

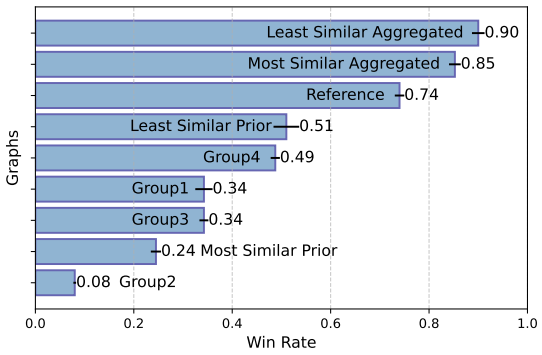


Figure C.3 Average win rates of graphs in pairwise comparison over the five independent runs.

graphs built on individual group discussions, highlighting the added value of synthesizing information from multiple groups of experts.

C.3. Unstructured Summary by DeepSeek-R1

Here, we present a summary of the decision factors provided by `DeepSeek-R1` (see Figure C.4).

Pros and Cons of Restrictive Measures

1. **Arguments Supporting Restrictions**
    1. **Advantages**:
        - **Breaking Transmission Chains**: Reducing imported cases delays local spread, buying time for vaccine development and medical resource stockpiling.
        - **Targeted Prevention**: Focus on Country A and 8 nations with reported imported cases to lower high-risk entry density.
        - **Public Health Security**: Prevents healthcare system overload and mitigates public panic.
    2. **Disadvantages**:
        - **Economic and Trade Losses**: Flight reductions and entry quarantines would impact tourism, education abroad, and international trade.
        - **High Implementation Costs**: Centralized quarantine demands significant medical resources and may spark dissatisfaction (e.g., returning students during summer vacation).
        - **International Relations Risks**: Unilateral restrictions could harm diplomatic image and conflict with WHO recommendations.
2. **Arguments Against Comprehensive Restrictions**
    1. **Limited Effectiveness**:
        - The virus can spread via asymptomatic carriers, animal/goods imports, etc., making personnel restrictions alone insufficient.
        - Short incubation period with transmissibility means short-term quarantine may miss cases (e.g., 5-day isolation cannot cover the full incubation period).
    2. **Alternative Solutions**:
        - Strengthen post-entry controls (e.g., health declarations, symptom monitoring, home quarantine) and animal product inspections.
        - Maintain international exchanges to avoid long-term economic damage from ”isolationist” policies.

Expert Panel Recommended Measures

Balancing pros and cons, a **”Targeted Restrictions + Enhanced Control”** hybrid strategy is advised:

1. Tiered Risk-Based Entry Controls

- **High-Risk Countries (Country A and another four countries with local transmission)**:
    - — Reduce direct flights to minimal essential levels (e.g., 15 weekly flights), prioritizing students and business travelers.
    - — Require 48-hour pre-departure negative PCR tests and 5-day centralized quarantine with testing upon arrival.
- **Medium/Low-Risk Countries (others with imported cases)**:
    - — Implement health declarations, symptom screening, and 5-day home health monitoring with community follow-up.

2. Strengthen Border and Domestic Controls

- **Pre-Entry Measures**: Airlines to conduct health pre-screening and risk notifications for passengers.
- **Symptom Monitoring**: Set up rapid PCR testing at airports/ports for immediate isolation of febrile or respiratory cases.
- **Animal Product Quarantine**: Suspend poultry imports from affected regions and enhance cold-chain disinfection.

3. Supporting Measures

- **Resource Stockpiling**: Expand quarantine facility capacity and ensure emergency readiness of makeshift hospitals.
- **International Cooperation**: Share data with WHO and neighboring countries to coordinate joint prevention.
- **Public Communication**: Transparently release epidemic updates to reduce panic and encourage public compliance.

Figure C.4 `Deepseek-R1` **summary.**

References

Abbas AE, Hupman AC, Monger R (2026) Assessing the quality of large language models and human inputs to a decision: A proposed framework and two benchmark case studies. *Decision Analysis* URL `https://doi.org/10.1287/deca.2025.0467`, ePub ahead of print March 25.

Bodily SE, Furman B (2016) Long-term care insurance decisions. *Decision Analysis* 13(3):173–191.

Bordley RF (2023) Lessons for decision-analysis practice from the automotive industry. *INFORMS Journal on Applied Analytics* 53(3):240–246.

Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, et al. (2020) Language models are few-shot learners. *Advances in Neural Information Processing Systems*, volume 33, 1877–1901 (Red Hook, NY: Curran Associates Inc.).

Budescu DV, Por HH, Broomell SB, Smithson M (2014) The interpretation of IPCC probabilistic statements around the world. *Nature Climate Change* 4(6):508–512.

Dhami MK, Mandel DR (2022) Communicating uncertainty using words and numbers. *Trends in Cognitive Sciences* 26(6):514–526.

Dillon RL, Bier VM, John RS, Althenayyan A (2023) Closing the gap between decision analysis and policy analysts before the next pandemic. *Decision Analysis* 20(2):109–132.

Everitt T, Carey R, Langlois ED, Ortega PA, Legg S (2021) Agent incentives: A causal perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11487–11495 (Washington, DC: AAAI).

Falconer JR, Frank E, Polaschek DL, Joshi C (2022) Methods for eliciting informative prior distributions: A critical review. *Decision Analysis* 19(3):189–204.

Feng Y, Zhou B, Lin W, Roth D (2025) BIRD: A trustworthy Bayesian inference framework for large language models. *International Conference on Learning Representations*, volume 2025, 8961–8989 (Appleton, WI: ICLR).

Fox J, Everitt T, Carey R, Langlois ED, Abate A, Wooldridge MJ (2021) PyCID: A python library for causal influence diagrams. *SciPy*, 65–73.

Gibson DR (2011) Speaking of the future: Contentious narration during the Cuban missile crisis. *Qualitative Sociology* 34:503–522.

Gregory R, Fischhoff B, McDaniels T (2005) Acceptable input: Using decision analysis to guide public policy deliberations. *Decision Analysis* 2(1):4–16.

Gu J, Jiang X, Shi Z, Tan H, Zhai X, Xu C, Li W, Shen Y, Ma S, Liu H, et al. (2026) A survey on LLM-as-a-judge. *The Innovation* 7(6):101253.

Guo D, Yang D, Zhang H, Song J, Wang P, Zhu Q, Xu R, Zhang R, Ma S, Bi X, et al. (2025) DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature* 645(8081):633–638.

Howard RA, Matheson JE (2005a) Influence diagram retrospective. *Decision Analysis* 2(3):144–147.

Howard RA, Matheson JE (2005b) Influence diagrams. *Decision Analysis* 2(3):127–143.

Huang J, Yang DM, Rong R, Nezafati K, Treager C, Chi Z, Wang S, Cheng X, Guo Y, Klesse LJ, et al. (2024) A critical assessment of using ChatGPT for extracting structured data from clinical notes. *npj Digital Medicine* 7(1):106.

Hurst A, Lerer A, Goucher AP, Perelman A, Ramesh A, Clark A, Ostrow A, Welihinda A, Hayes A, Radford A, et al. (2024) GPT-4o system card. *arXiv preprint arXiv:2410.21276* .

Karvetski CW, Meinel C, Maxwell DT, Lu Y, Mellers BA, Tetlock PE (2022) What do forecasting rationales reveal about thinking patterns of top geopolitical forecasters? *International Journal of Forecasting* 38(2):688–704.

Keating LM, Randall L, Stanton R, McCormack C, Lucid M, Seaborn T, Converse SJ, Canessa S, Moehrenschlager A (2023) Using decision analysis to determine the feasibility of a conservation translocation. *Decision Analysis* 20(4):295–310.

Keeney RL (1996) *Value-focused thinking: A path to creative decisionmaking* (Cambridge, MA: Harvard University Press).

Kojima T, Gu SS, Reid M, Matsuo Y, Iwasawa Y (2022) Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems* 35:22199–22213.

Liu O, Fu D, Yogatama D, Neiswanger W (2025) DeLLMa: Decision making under uncertainty with large language models. *International Conference on Learning Representations*, volume 2025, 43850–43886 (Appleton, WI: ICLR).

Madaan A, Tandon N, Gupta P, Hallinan S, Gao L, Wiegreffe S, Alon U, Dziri N, Prabhumoye S, Yang Y, et al. (2023) SELF-REFINE: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, volume 36, 46534–46594 (Red Hook, NY: Curran Associates Inc.).

Matheson D, Matheson JE (2005) Describing and valuing interventions that observe or control decision situations. *Decision Analysis* 2(3):165–181.

Merrick JR, McLay LA (2010) Is screening cargo containers for smuggled nuclear threats worthwhile? *Decision Analysis* 7(2):155–171.

National Health Commission of the People's Republic of China (2024) Diagnosis and treatment protocol for human avian influenza (2024 edition). Technical Report Document 268 (2024), National Health Commission of the People's Republic of China, Beijing, URL `https://www.nhc.gov.cn/ylyjs/gzdt/202407/430f72d43b4c4d70b12f0edbbf9eaf94/files/1734003043606_85634.pdf`.

Perlis RH, Goldberg JF, Ostacher MJ, Schneck CD (2024) Clinical decision support for bipolar depression using large language models. *Neuropsychopharmacology* 49(9):1412–1416.

Regnier ED, Feldmeier JW (2022) D minus months: Strategic planning for weather-sensitive decisions. *Decision Analysis* 19(1):1–20.

Sandmann S, Hegselmann S, Fujarski M, Bickmann L, Wild B, Eils R, Varghese J (2025) Benchmark evaluation of DeepSeek large language models in clinical decision-making. *Nature Medicine* 31(8):2546–2549.

Schoenegger P, Jones CR, Tetlock PE, Mellers B (2025) Prompt engineering large language models' forecasting capabilities. Preprint, submitted June 2, URL `https://arxiv.org/abs/2506.01578`.

Shachter RD, Bhattacharjya D (2011) *Solving Influence Diagrams: Exact Algorithms* (Wiley, New York), ISBN 9780470400630, URL `http://dx.doi.org/10.1002/9780470400531.eorms0808`.

Simon J, Siebert JU (2025) ChatGPT vs. experts: Can GenAI develop high-quality organizational and policy objectives? *Decision Analysis* URL `https://doi.org/10.1287/deca.2025.0387`, ePub ahead of print November 11.

Stonebraker JS (2023) Case article—Bayer new drug development decision making. *INFORMS Transactions on Education* 23(3):154–158.

von Winterfeldt D (2013) Bridging the gap between science and decision making. *Proceedings of the National Academy of Sciences* 110(Suppl 3):14055–14061.

Wan Q, Xu X, Hunt K, Zhuang J (2022) Stay home or not? modeling individuals' decisions during the COVID-19 pandemic. *Decision Analysis* 19(4):319–336.

Wang X, Wei J, Schuurmans D, Le QV, Chi EH, Narang S, Chowdhery A, Zhou D (2023) Self-consistency improves chain of thought reasoning in language models. *The Eleventh International Conference on Learning Representations* (Kigali, Rwanda).

Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D, et al. (2022) Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35:24824–24837.

Xia S, Lu B, Eisner J (2024) Let's think var-by-var: Large language models enable ad hoc probabilistic reasoning. Preprint, submitted December 3, URL `https://arxiv.org/abs/2412.02081`.

Xiao Z, Zhang D, Wu Y, Xu L, Wang Y, Han X, Fu X, Zhong T, Zeng J, Song M, et al. (2024) Chain-of-experts: When LLMs meet complex operations research problems. *International Conference on Learning Representations*, volume 2024, 48519–48537 (Appleton, WI: ICLR).

Xiong M, Hu Z, Lu X, Li Y, Fu J, He J, Hooi B (2024) Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. *International Conference on Learning Representations*, volume 2024, 23650–23678 (Appleton, WI: ICLR).

Yang A, Yang B, Zhang B, Hui B, Zheng B, Yu B, Li C, Liu D, Huang F, Wei H, et al. (2024) Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115* .

Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan KR, Cao Y (2023) ReAct: Synergizing reasoning and acting in language models. *The Eleventh International Conference on Learning Representations* (Kigali, Rwanda).

Ye Y, Cong X, Tian S, Cao J, Wang H, Qin Y, Lu Y, Yu H, Wang H, Lin Y, et al. (2023) ProAgent: From robotic process automation to agentic process automation. Preprint, submitted November 23, URL `https://arxiv.org/abs/2311.10751`.

Zhao Z, Lee WS, Hsu D (2023) Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems* 36:31967–31987.

Zhu JQ, Griffiths T (2025) Eliciting the priors of large language models using iterated in-context learning. *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47 (San Francisco).

Yifan Hong is a PhD candidate in the department of industrial engineering at Tsinghua University. His research interests include behavioral modeling, decision analysis, and online learning and decision making.

Sizhong Qin is a PhD candidate in the department of civil engineering at Tsinghua University. His research interests include intelligent design and construction of building structures, Large Language Models in civil engineering applications, and diffusion models for structural design.

Chen Wang is an associate professor in the department of industrial engineering at Tsinghua University. She obtained her doctoral degree from the Department of Industrial and Systems Engineering at the University of Wisconsin–Madison. Her research interests include decision analysis, risk analysis, and data-driven modeling.