Rating Composite AI Models for Robustness Through Probabilistic Planning

Kausik Lakkaraju[1], **Sunandita Patra**[2], **Parisa Zehtabi**[3], **Biplav Srivastava**[1]
[1]University of South Carolina, USA
[2]J.P. Morgan AI Research, USA
[3]J.P. Morgan AI Research, UK
kausik@email.sc.edu, sunandita.patra@jpmchase.com, parisa.zehtabi@jpmorgan.com, biplav.s@sc.edu

Abstract

Many real-world AI systems combine several primitive component models, such as translators and sentiment analyzers, into larger composite models, like chatbots. Understanding how these compositions behave under uncertainty and how properties like bias or instability moves through a composite model is increasingly important, yet most evaluation methods still focus on primitive models. We introduce a new use of probabilistic planning to assess the robustness of composite AI models. Each component model call is represented as a stochastic action in RDDL domain, and the reward combines robustness metrics to the cost of components (actions). The planner runs each primitive model on randomly drawn data batches, allowing robustness to be assessed under variation in both the data and the model outputs induced by that data. We demonstrate via case studies and experiments in multilingual sentiment analysis and a synthetic domain, the planner consistently identifies more stable composite configurations than baseline methods, showing that probabilistic planning can serve as a practical, scalable, approach for reasoning about reliability in complex, composite, AI models.

Track: Industry and Applications
Extended Version and Code: https://github.com/kausik-l/composite-rating-rddl

1 Introduction

Although AI systems are achieving impressive performance, a continuing challenge in adopting them for real-world applications is their fragile and unpredictable behavior in the presence of uncertainty. For example, perturbations in input caused by noise or variations due to user demographic diversity can lead to wide swings in model output. This issue is amplified in *composite* AI models, systems that chain together several primitive models such as translators, summarizers, or sentiment analyzers. Each stage influences the next, and a poor choice of primitive model at one step can propagate instability or bias downstream.

Our approach addresses this challenge by helping users *select the right primitive model at every stage* of a composite pipeline so that the resulting system satisfies the user's objectives, which may include robustness and accuracy. In this paper, we define robustness as the model's ability to rely on desirable features (the treatment attribute T) while not relying on undesirable or protected attributes (the variable Z). We quantify these effects using the causal rating methodology introduced in prior work (Lakkaraju, Srivastava, and Valtorta 2024; Lakkaraju et al. 2024a, 2023, 2025), which measures how strongly a model's predictions respond to T and how much they are influenced by Z through statistical or causal pathways.

Prior work has shown that round-trip translation can alter how sentiment models behave: (Christiansen, Gammelgaard, and Søgaard 2021) demonstrate that it can reduce bias in Sentiment Analysis Systems (SASs), and (Lakkaraju et al.

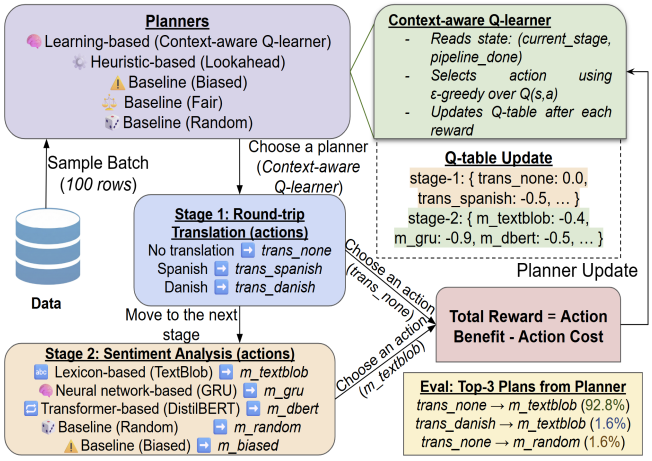


Figure 1: Workflow of our approach illustrated using the sentiment analysis tasks (T1 and T2). At the start of each episode, a batch of data is sampled and the planner selects one primitive model per stage of the composite model. In Stage 1 it chooses a translation action (e.g., *trans_none*, i.e., no round-trip translation), and in Stage 2 a sentiment analysis action (e.g., *m_textblob*, i.e., sentiment analysis using textblob). Each action receives a reward consisting of an *action benefit*, computed from robustness metrics on the sampled batch, and an *action cost* representing the cost of invoking a model. The planner updates its policy using the total reward and outputs a sequence of primitive models, from which the most frequent plans across evaluation episodes can be identified.

2023) further report that it lowers statistical bias on human-generated data while increasing it on synthetic data. To illustrate the types of choices a planner must make, consider a simple two-stage sentiment-analysis composite model. The first stage selects a preprocessing transformation, such as round-trip translation through Spanish (*trans_spanish*) or Danish (*trans_danish*), and the second stage selects a sentiment model (e.g., DistilBERT-based, TextBlob (*m_textblob*), or GRU-based (*m_gru*)). A fixed but suboptimal choice like *trans_spanish* $\rightarrow$ *m_gru* may amplify bias, whereas a learned policy may consistently select a more stable composite model such as *trans_none* $\rightarrow$ *m_textblob*, which our experiments show emerges as the top plan in both T1 and T2 (Section 5.2). **Our approach helps users select the right primitive model at each stage of a composite model to maximize a chosen objective, which in our case is robustness.** The code for our framework and experiments is publicly available.[1] Our contributions are as follows:

1. We develop a rollout-based method that estimates the **robustness of a composite model under stochastic data and model behavior**, allowing informed selection of primitive models at each stage of a composite model.
2. We cast robustness assessment of a **composite model as an RDDL planning problem**, treating each primitive model invocation as an action and using causally grounded robustness metrics as the reward.
3. We evaluate the approach on **two real-world sentiment tasks** (RTS-Small and RTS-Large) and show how differences in data structure, particularly the presence or absence of a confounder, lead to distinct model-selection behaviors.
4. We demonstrate **scalability on a large synthetic composite model** with many stages and model families, allowing assessment of robustness when there are sequential dependencies among primitive models.

2 Problem

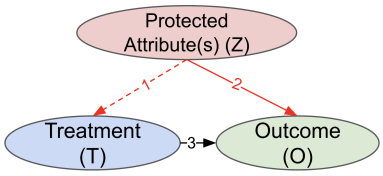


Figure 2: Generalized causal graph used for causal analysis for tasks T1, T2, and T3. The validity of link '1' depends on the data distribution ($T \mid Z$), while the causal effects associated with links '2' and '3' are quantified in our experiments and integrated into the reward.

We consider the problem of selecting *primitive models* inside a *composite AI model*. A composite model is a structured sequence of primitive models that together perform a single task. At each stage, the planner must choose one primitive model from several candidates so that the resulting composite model performs effectively, behaves reliably under uncertainty, and avoids undesirable dependencies on protected attributes.

Causal Setup. Tasks T1, T2, and T3 follow the generalized causal graph shown in Figure 2. The setting involves three types of variables: Z - protected attributes that should not influence model outcomes, T - treatment variables that should influence outcomes, O - the outcome produced by a primitive model on a data instance. Our goal is to choose primitive models that (i) adjust predictions appropriately as T varies, (ii) avoid systematic disparities across groups defined by Z, and (iii) limit the extent to which Z acts as a confounder in the $T \rightarrow O$ relationship. A confounder is any variable that affects both T and O, making it difficult to distinguish true causal effects from spurious associations. In the context of sentiment analysis on sentences such as "$< person\_name >$ is feeling angry," gender (Z) of the person acts as a confounder when it influences both which emotion (T) is assigned (i.e., more negative emotion words associated with one gender compared to others) and how the sentiment model evaluates the sentence (O).

Probabilistic Planning Formulation. We frame primitive model selection as an MDP $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$. A state s records the current stage of the composite model and the primitive models chosen so far. An action a selects a primitive model for the current stage. Since rewards are computed on batches sampled from a large dataset, transitions $P(s' \mid s, a)$ are stochastic: different batches yield slightly different empirical outcomes even for the same action. The reward $R(s, a, s')$ encodes how well the selected primitive model advances the goals described above; γ is the discount factor.

> **Planner Objective.** Select primitive models that maximize the influence of treatment (T) on outcomes (O), minimize the influence of protected attributes (Z), reduce confounding effects of Z, and manage model invocation cost, while accounting for inter-dependencies between stages where the input to each model is shaped by the outputs of earlier models.

This problem formulation captures the core decision-making problem: constructing a composite model that behaves reliably and fairly under uncertainty.

3 Related Work

We situate our work within three areas of prior research: robustness assessment of AI models, planning for AI model reliability and trust, and challenges arising in the composition of multiple AI models.

3.1 Robustness Assessment of AI Models

Assessing the robustness of AI models remains challenging, especially when robustness must account for both model stability and sensitivity to protected attributes. Traditional statistical fairness metrics often overlook causal mechanisms that can lead to unfair or unreliable outcomes (Kusner et al.

[1] https://github.com/kausik-l/composite-rating-rddl

2017; Verma and Rubin 2018). To address this limitation, (Srivastava and Rossi 2019; Bernagozzi et al. 2021; Srivastava et al. 2020) proposed a two-step rating method for AI services, where the ratings of individual services are aggregated to understand a composite system robustness. Causality-based fairness definitions provide a more principled lens by isolating the causal effect of protected attributes on model outcomes (Carey and Wu 2022). Building on this perspective, (Lakkaraju 2022) and (Srivastava et al. 2023) introduced causal rating frameworks that quantify how AI models rely on desirable versus protected attributes. These methods have been applied to sentiment analysis systems (Lakkaraju, Srivastava, and Valtorta 2024), composite models with translators (Lakkaraju et al. 2023), and time-series forecasting models (Lakkaraju et al. 2024a, 2025), allowing robustness to be communicated through ratings. In this work, we adapt the robustness metrics used in these rating approaches by integrating them directly into the reward function of our planner. This allows the planner to score primitive models by how much they rely on desirable attributes, how much they depend on protected attributes, and whether confounding is present.

3.2 Planning for System Reliability and Trust

Probabilistic planning is suited for robustness assessment of composite models because it evaluates sequential decisions under uncertainty and can capture how each primitive model choice can influence later stages. Embedding robustness metrics in the reward helps in end-to-end construction of composite models that remain reliable under stochastic variation. Our formulation uses the Relational Dynamic Influence Diagram Language (RDDL), a framework for modeling probabilistic sequential decision problems (Sanner 2011). Fairness has been studied in sequential decision making (Hu and Zhang 2022). Safe action models, which guarantee that the learned model generates only valid and safe actions, are also required in critical domains (Mordoch, Juba, and Stern 2023). However, none of these approaches combine probabilistic planning with causality-based metrics to quantify how strongly a primitive model should respond to desirable causal attributes while suppressing influence from protected ones.

3.3 Composition of AI Models

Modern systems require guarantees about their behavior under uncertainty, particularly regarding robustness and fairness (Kailkhura et al. 2021). These challenges intensify when multiple AI models are composed, because the output of one primitive model becomes the input to the next. Such dependencies make it difficult to predict how uncertainty, algorithmic bias, or robustness failures will propagate. Similar concerns arise in RDDL planning problems, where early decisions can influence downstream outcomes and generate disparities across subgroups (Hu and Zhang 2022). Even if individual models satisfy fairness criteria in isolation, their combination may fail to do so (Dwork and Ilvento 2018). Verification of composite models is further complicated by scalability issues, debugging difficulty, and feedback loops that obscure end-to-end behavior (Pullum 2021). Moreover, robustness is not compositional: cascading individually robust models may reduce overall reliability due to inter-model interactions (Mangal et al. 2022). Motivated by these challenges, we use probabilistic planning to help users select appropriate primitive models at each stage of a composite AI model, especially when decisions made at one stage influence those downstream. We demonstrate this on two real-world tasks (T1, T2) and a large synthetic task (T3) (Section 5.2).

4 Method

Our method operationalizes the problem in Section 2 by treating primitive model selection as a probabilistic planning problem in an RDDL domain. Each stage of the composite model is a decision point, each admissible action corresponds to invoking a primitive model, and the reward incorporates robustness metrics together with an invocation cost.

4.1 RDDL Formulation

We represent a composite model as a finite-horizon RDDL domain where each stage of the pipeline is a decision point and each action invokes a primitive model. The RDDL specification defines the stage ordering, admissible models at each stage, and state variables that track pipeline progress. State transitions move to the next stage based on the selected model. Stochasticity arises from the reward, which is estimated on randomly sampled data batches during rollouts. The reward combines robustness metrics with a fixed invocation cost.

4.2 Robustness Metrics via Rollouts

During learning, the planner interacts with the RDDL model through rollouts. A rollout is a full episode in which the composite model is executed from its first stage to termination under the current policy. Each episode begins by sampling a batch of data from the larger dataset, and all robustness metrics for that episode are computed using only this batch. At each stage of the composite model, the selected primitive model's predictions on the sampled batch allow us to estimate how the primitive model behaves with respect to attributes of interest. Let Z denote protected attributes, T denote treatment attributes that ought to influence the model's outcome, and O denote the output produced by the selected primitive model. Our method does not depend specific evaluation metrics. The reward function can be defined using any metric that quantifies the desired objective. In this work we use causally grounded robustness metrics, but other objectives can be incorporated without changing the planning formulation.

Average Treatment Effect (ATE). ATE measures how strongly the output O responds to changes in T. We estimate

$$\text{ATE} = E[O \mid do(T{=}1)] - E[O \mid do(T{=}0)],$$

using G-computation (Greenland and Robins 1986) to adjust for protected attributes. Concretely, for each treatment value we evaluate the primitive model's predictions while holding T fixed and averaging over the empirical distribution of Z.

This yields an estimate of how the output would change under an intervention on T while blocking indirect influence from Z.

Weighted Rejection Score (WRS). To quantify disparities across groups of a protected attribute (**statistical bias**), we compute the Weighted Rejection Score (WRS). For each protected attribute Z, we test for differences in mean outcomes (O) between groups at three confidence levels (95%, 70%, and 60%). At each confidence level, a rejection of the null hypothesis contributes a fixed weight to the score (1.0, 0.8, and 0.6, respectively). The WRS is computed by summing these weights across all confidence levels at which the null hypothesis is rejected. Formally, $\text{WRS} = \sum_i w_i \, x_i$, where $x_i \in \{0, 1\}$ indicates whether the null hypothesis is rejected at confidence level i, and w_i denotes the corresponding weight. Higher WRS values indicate stronger group-level disparities in model outputs.

Deconfounding Impact Estimation (DIE). To assess whether Z acts as a confounder in the $T \to O$ relationship, we compare the unadjusted ATE with the adjusted ATE obtained through G-computation. The absolute difference,

$$\text{DIE} = \left|\text{ATE}_{\text{unadj}} - \text{ATE}_{\text{adj}}\right|,$$

captures the degree to which Z alters the true causal effect of T on O. A larger DIE value indicates that part of the observed treatment effect is attributable to imbalances across protected groups (Z).

Reward during rollouts. At each decision point, the selected primitive model affects how the composite model behaves on the sampled batch. We assign a scalar reward

$$R = \eta_1 \, \text{ATE} - \eta_2 \, \text{WRS} - \eta_3 \, \text{DIE} - \eta_4 \, \text{Cost},$$

where Cost is the operational cost defined in the RDDL instance and the coefficients $\eta_1, \eta_2, \eta_3, \eta_4$ determine the contribution of each term. This reward encourages the selection of primitive models that respond appropriately to variation in T, exhibit smaller group disparities across Z, and reduce confounding, while also accounting for fixed model invocation cost. Each episode terminates once the final stage of the composite model has been executed, and the total return for that episode is the objective used for learning.

4.3 Context-Aware Q-Learning

We use a tabular Q-learning agent to learn a policy over RDDL states. The state representation includes the current stage and any context variables that influence later decisions (e.g., a context variable may record whether an earlier stage applied a round-trip translation, since this affects which sentiment model needs to be invoked in the later stage). At each state s, the agent maintains $Q(s, a)$ for each admissible primitive model and selects actions using an ε-greedy strategy. After observing reward r and transitioning to s', the agent performs the update

$$Q(s, a) \leftarrow Q(s, a) + \alpha \, [\, r + \gamma \max_{a'} Q(s', a') - Q(s, a) \,].$$

Terminal states omit the maximization term. Over many episodes, the Q-table converges to a deterministic policy selecting one primitive model per stage. We evaluate three variants of the context-aware Q-learning agent in our experiments. The first is trained to reduce statistical bias (WRS) and is denoted Q-L (WRS). The second is trained to reduce confounding bias (DIE) while increasing the causal impact of the treatment on the outcome (ATE), simply denoted Q-L (DIE). The third combines all three objectives, reducing WRS and DIE while increasing ATE, and we refer to it as Q-L (Both). All variants also include the model invocation cost in the reward. Interestingly, our experiments show that Q-L (DIE) often attains WRS values that are lower than or comparable to those of Q-L (WRS) and Q-L (Both), despite not being trained to optimize WRS directly (Table 2). The training times for the three variants are reported in the extended version (Lakkaraju et al. 2026).

4.4 Overall Workflow

Figure 1 summarizes the workflow. Each episode begins by sampling a batch of data and using pre-generated predictions for all primitive models. The planner proceeds stage by stage, selecting a primitive model, applying its predictions to compute robustness metrics, and receiving a reward. After the final stage, the episode ends and a new batch is sampled. Over training episodes, the planner identifies a composite model that yields high expected reward under the combined robustness and cost criteria.

5 Experiments and Results

This section evaluates our approach described in Section 4 across three tasks that differ in structure, data sources, and the primitive models available at each stage. We first describe the baseline policies used for comparison, and then present each task's data, primitive models, experimental setup, and results. Additional results including plots and statistical significance tests are included in the extended version (Lakkaraju et al. 2026).

5.1 Baseline Policies

We evaluate three variants of the context-aware Q-learning agent described in Section 4.3. We compare these Q-learning agents with the following baselines:

1. **Random Policy:** Selects a primitive model uniformly at random at each stage.
2. **Fixed Policies:** Select the same primitive model at every stage. We use two variants: one that always selects the lowest-bias model in the pool, and another that selects a consistently higher-bias model.
3. **Heuristic Policy:** At each stage, scores admissible primitive models using a one-step objective combining the model's invocation cost and its WRS on the current batch, and selects the model with the lowest score. This policy considers only immediate effects.

5.2 Tasks

We evaluate our method on three tasks that differ in structure, data sources, and the primitive models available at each

stage. Each task section follows the same layout: task description, data, primitive models, experimental setup, and results.

T1: Round-Trip Translated Sentiment–Small (RTS-Small) This task studies how different primitive models behave on short, helpdesk-style dialogue utterances. The goal is to evaluate how model outputs change with respect to the treatment variable (different conversations) T while limiting disparity across groups defined by the protected attribute Z. In the generalized causal graph (Figure 2), RTS-Small corresponds to the setting where link 1 ($T \mid Z$) is absent, and links 2 and 3 are present: the protected attribute is present but does not act as a confounder.

Data. We use the *Unibot* dataset (Lakkaraju et al. 2023), which contains 31 campus helpdesk conversations (1,517 utterances). The logs do not include demographic metadata, so we introduce a simple gender proxy ("Hey boy", "Hey girl", "Hey") to define the protected attribute Z. The sentiment-model output serves as O. Because user gender does not influence the assignment of T in this dataset, Unibot does not instantiate a confounding path $Z \rightarrow T \rightarrow O$ in Figure 2. A sample of this dataset is provided in the extended version (Lakkaraju et al. 2026).

Primitive Models. The composite model has two stages. Stage 1 applies a round-trip translation, translating text from one langauge to the same through different intermediate languages, using Google Translate through Spanish or Danish (`trans_spanish`, `trans_danish`) or applies no translation (`trans_none`). Prior work has shown that round-trip translation can alter how sentiment models behave: (Christiansen, Gammelgaard, and Søgaard 2021) demonstrate that it can reduce bias in Sentiment Analysis Systems (SASs), and (Lakkaraju et al. 2023) further report that it lowers statistical bias on human-generated data while increasing it on synthetic data. Stage 2 applies a sentiment-analysis model to the resulting text. We use the sentiment models from (Lakkaraju et al. 2023): DistilBERT-based SAS, lexicon-based TextBlob, a lightweight GRU-based SAS, and two baselines (random and biased).

RDDL State. The RDDL state captures the progress of the pipeline using two state-fluents: `current_stage(stage)`, which indicates the active stage of the composite model, and `pipeline_done`, which marks termination after the final stage. The planner selects primitive models using the action-fluent `select_component`. This same state schema is used for both T1 and T2.

Experimental Setup. At each stage, the planner selects one primitive model from the admissible set. Each training episode samples a batch of Unibot utterances, applies the chosen translation and sentiment model, and computes the WRS along with invocation cost (0.5 for translation and 0.1 for invoking a sentiment model). The reward is accumulated over the two stages and used to update the Q-learning agent. The reward uses coefficients $\eta_1 = \eta_3 = 0$ (no confounder) and $\eta_2 = \eta_4 = 1$ (Section 4.2).

Results. Evaluation results across 500 episodes are shown in Table 1 (and Figure 3a in the Extended Version (Lakkaraju et al. 2026)). Statistical comparisons of Q-L (WRS) against the other agents is provided in the extended version (Lakkaraju et al. 2026). The Q-Learning (WRS) agent achieves a mean reward of -0.36, substantially outperforming both Fixed (Biased) (-7.65) and Random (-2.30). One-way ANOVA test followed by Tukey HSD confirm these improvements are highly significant for both Fixed (Biased) ($p = 0.00^{***}$) and Random ($p = 0.00^{***}$). The only policy that surpasses Q-L (WRS) in reward is the heuristic (-0.25), but this difference is small and the heuristic incurs the slowest inference time (1.22ms vs. 0.01ms). Top-3 plans generated by each policy are provided in the extended version (Lakkaraju et al. 2026).

RTS-Small					
Agent	Reward ↑	Cost ↓	WRS ↓	DIE ↓	Time (ms) ↓
Q-L (WRS)	-0.36	0.13	0.02	–	0.01
Heuristic (Lookahead)	-0.25	0.10	0.02	–	1.22
Fixed (Biased)	-7.65	0.10	0.76	–	0.00
Random	-2.30	0.45	0.18	–	0.00
RTS-Large					
Q-L (WRS)	-0.50	0.12	0.03	0.01	0.03
Q-L (DIE)	-0.65	0.12	0.04	0.01	0.02
Q-L (Both)	-0.55	0.11	0.03	0.01	0.02
Heuristic (Lookahead)	-2.24	0.10	0.17	0.05	1.29
Fixed (Biased)	-11.95	0.10	0.90	0.29	0.00
Random	-3.50	0.44	0.23	0.07	0.00

Table 1: Performance of Q-Learning (Q-L) policies and baselines on the two sentiment tasks: RTS-Small (T1, no confounder) and RTS-Large (T2). Reward and cost are averaged over stages, and Time (ms) reports average inference time per decision. Green cells indicate best-performing values within each task; red cells indicate worst-performing values.

Conclusion. Q-Learning provides a stable and computationally efficient alternative to heuristic selection. It significantly outperforms random and biased baselines while achieving performance close to the heuristic at a fraction of the inference cost (Table 1).

T2: Round-Trip Translated Sentiment–Large (RTS-Large) RTS-Large builds on the RTS-Small (T1) setup and follows the causal model of (Lakkaraju et al. 2023), in which the protected attribute Z acts as a confounder between the utterances (treatment T) and the sentiment model outcome O. In the generalized causal graph (Figure 2), this corresponds to the presence of the confounder path $Z \rightarrow T \rightarrow O$, in contrast to T1 where this path is absent. Here, the treatment variable T is denoted by the conversation index (`C_num`).

Data. We use the *ALLURE* dataset (Lakkaraju et al. 2023, 2024b, 2022), collected through three controlled studies in which participants interacted with a multimodal tutoring

system designed to teach them how to solve the white cross on a Rubik's Cube. The dataset contains 18 user-participant conversations (3,543 utterances) with self-reported gender metadata (9 male, 8 female, 1 not disclosed). A snapshot of the dataset is provided in the extended version (Lakkaraju et al. 2026).

Primitive Models. The composite model uses the same two-stage structure and the same set of primitive models as in T1: round-trip translation in Stage 1 and sentiment analysis in Stage 2.

RDDL State. The RDDL state includes `current_stage(stage)`, which tracks the active stage, `last_used_family(family)`, which records the family of the previously selected model (biased or fair) to estimate switching costs between model families, and `pipeline_done`, which marks termination.

Experimental Setup. As in T1, each episode samples a batch of utterances, applies the selected translation and sentiment model, and computes robustness metrics (WRS and DIE) together with model invocation cost. The available translation actions (`trans_spanish`, `trans_danish`, `trans_none`) and sentiment models are identical to T1. Because T2 contains a confounder (backdoor) path, we use reward coefficients $\eta_1 = 0, \eta_2 = 1, \eta_3 = 10$ and $\eta_4 = 1$ (Section 4.2) to balance WRS, DIE, and invocation cost.

Results. Evaluation results across 500 episodes are shown in Table 1 (and Figure 3b in the Extended Version (Lakkaraju et al. 2026)). Statistical comparisons of Q-L (WRS) against the other agents is provided in the extended version (Lakkaraju et al. 2026). Q-L (WRS) attains the highest reward (-0.24). Q-L (Both) and Q-L (DIE) perform similarly to Q-L (WRS) in reward. Compared to baselines, Q-L (WRS) shows large and statistically significant gains over Fixed (Biased) (Diff $= +11.45$, $p = 0.00^{***}$), Heuristic ($+1.74$, 0.00^{***}), and Random ($+2.99$, 0.00^{***}) policies. The heuristic again suffers from high inference cost (1.29 ms) and higher disparity (WRS $= 0.17$). The top-3 plans generated by Q-L (WRS) are shown below, with plans generated by the other policies provided in the extended version (Lakkaraju et al. 2026).

Top-3 Plans (from Q-Learning (WRS), RTS-Large).

- `trans_none → m_textblob` (93%)
- `trans_danish → m_textblob` (2.2%)
- `trans_spanish → m_textblob` (1.6%)

Conclusion. Q-L (WRS) attains the highest reward. Across all comparisons, the Q-learning agents substantially outperform the fixed and random baselines, and do so with far lower inference-time cost and higher reward than the heuristic policy (Table 1).

T3: Synthetic Task We created a synthetic sequential decision-making environment to investigate how different planning strategies navigate the trade-off between costs and fairness. The environment models a multi-stage composite AI model in which the objective is to maximize total utility by increasing the causal effect of the treatment, minimizing the cost of switching between model families, and simultaneously reducing both statistical and confounding bias.

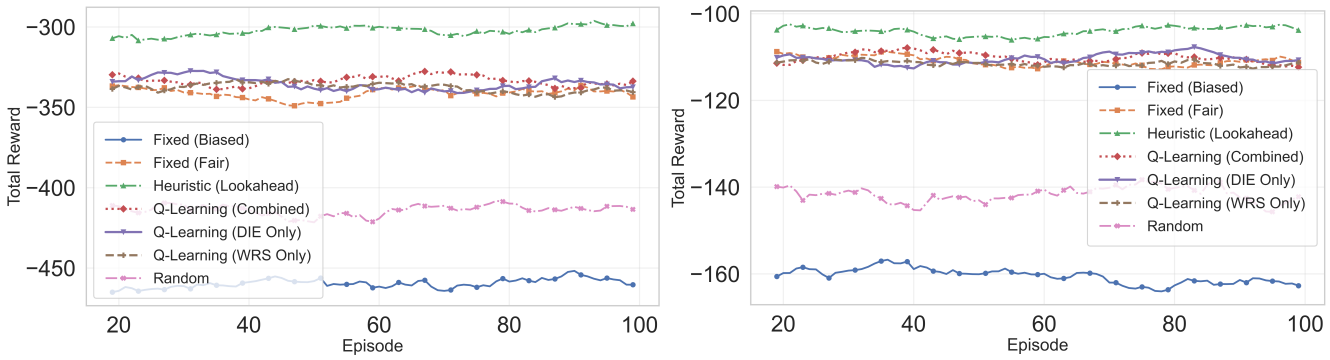
Data. We generate two synthetic scenarios: one with a 30-stage composite AI model with 15 primitive models available at each stage, and another with 10 stages and 5 primitive models per stage, to evaluate the scalability of our method. Each dataset contains 30,000 samples and encodes both:

- **Systemic bias**: protected attribute Z causally influences treatment variable (or merit) T,
- **Algorithmic bias:** model outputs depend on both T and Z.

At each stage, a set of primitive models is available. In this synthetic setting, we use a fixed pool of M primitive models across all N stages, with each stage selecting from the same model set. This differs from our previous tasks (T1 and T2), where the number of admissible primitive models can vary by stage, but our planning formulation applies to both cases. The generated dataset includes one output column for every stage–model pair, allowing evaluation of any composite model. We construct the synthetic data using a Structural Causal Model (SCM) (Figure 2) that induces both systemic and algorithmic bias. Let N denote the number of sequential stages in the composite model and M the number of primitive models available at each stage. The data generation process for each individual i follows a causal graph:

Systemic Bias (Input Level) We first sample a binary protected attribute $Z_i \in \{0, 1\}$ (e.g., demographic group) and a merit score $T_i \in [0, 1]$. To model systemic inequality, where historical factors create disparities in observed qualifications prior to any algorithmic assessment, we define T_i as causally dependent on Z_i: $T_i = \text{clip}_{[0,1]}\left(\mu_T + \gamma Z_i + \epsilon_{T,i}\right)$
where:
- μ_T is the baseline merit intercept, representing the average qualification level before group effects are applied (set to 0.4).
- γ represents the magnitude of *systemic bias*. A positive γ (set to 0.2) introduces a statistical advantage for group $Z = 1$.
- $\epsilon_{T,i} \sim \mathcal{N}(0, \sigma_T^2)$ represents random individual variation. This term ensures that merit is not just determined by the protected attribute group; instead, individuals within the same group Z exhibit a distribution of qualifications around the group mean.
- $\text{clip}_{[0,1]}$ constrains the resulting merit score to the range $[0, 1]$.

Algorithmic Bias (Model Level) The input for the first stage is initialized as the merit, $\hat{Y}_{0,i} = T_i$. At each stage $s \in \{1, \ldots, N\}$, a selected model m processes the incoming input $\hat{Y}_{s-1,i}$ to produce an output score $Y_{s,m,i}$. We model the output of model m as a linear combination of merit and the protected attribute groups, passed through a sigmoid function $\sigma(\cdot)$ to bound scores to the range $[0, 5]$:



(a) Synthetic (30 stages, 15 models)

(b) Synthetic (10 stages, 5 models)

Figure 3: Comparison of synthetic task evaluations for two (stage, model) configurations.

$$Y_{s,m,i} = 5 \cdot \sigma\left(\alpha_m \hat{Y}_{s-1,i} + \beta_m Z_i + \epsilon_{Y,i}\right) \quad (1)$$

where: - α_m is the *utility coefficient*. It governs the model's sensitivity to the merit propagated from the previous stage. We sample $\alpha_m \sim \mathcal{U}(0.8, 1.2)$ to simulate variations in model accuracy.
- β_m is the *algorithmic bias coefficient*. It governs the direct effect of the protected attribute Z on the outcome, independent of merit.
- $\epsilon_{Y,i} \sim \mathcal{N}(0, \sigma_Y^2)$ represents aleatoric uncertainty in the model's prediction.

Primitive models. We partition the M models into two latent families based on the parameter β_m from equation 1:
- **Biased Family ($\mathcal{F}_{biased}$):** Models sample $\beta_m \in [0.1, 0.3]$. A positive β explicitly rewards individuals with $Z = 1$ regardless of their merit, amplifying the existing systemic bias.
- **Fair Family ($\mathcal{F}_{fair}$):** Models sample $\beta_m \in [-0.05, 0.05]$. With $\beta \approx 0$, these models base decisions primarily on the merit $\hat{Y}_{s-1}$.

The input propagated to the next stage, $\hat{Y}_{s,i}$, is the mean output of the models in stage s, creating an inter-dependency between the models chosen at different stages.

Experimental Setup. The problem is formalized as a stochastic planning domain in RDDL. The state space includes the current stage index and the *family context* (the family of the previously selected model). In this task, the reward includes all three robustness components: ATE (to be maximized), WRS, and DIE, so we set the coefficients to $\eta_1 = 10$, $\eta_2 = 10$, $\eta_3 = 20$, and $\eta_4 = 1$ (Section 4.2). The action cost consists of a base cost of 0.5 and an additional switching cost of 0.5 whenever the policy changes model families.

Results. Efficiency and Scalability. As shown in Table 2 and Figure 3, the Heuristic planner attains the highest total reward in both synthetic settings, but this comes with substantially higher inference cost (16.57 ms and 4.39 ms per decision). In contrast, all Q-Learning variants execute via a constant-time table lookup (≈0.13 ms), making them far more efficient at inference time.

Utility Maximization. Across both configurations, the Q-Learning agents consistently outperform the Fixed (Biased), Fixed (Fair), and Random baselines. For the larger setting (30,15), Q-L (Combined) achieves a reward of -11.06, significantly better than Fixed (Biased) and Random. Although the Heuristic obtains the best overall reward (-10.14), its plan time at inference is much higher. Q-L (DIE) yields the highest ATE and exhibits WRS lesser than or closer to Q-L (WRS) in both cases. The top plans generated by the different policies in both configurations are provided in the extended version (Lakkaraju et al. 2026).

Conclusion. Across both synthetic settings, the heuristic approach achieves the highest reward but incurs substantially higher inference-time than all other methods. The Q-Learning variants shift this computational burden to training, allowing fast inference while still performing significantly better than the fixed and random baseline policies. Although the different reward modes (WRS, DIE, Combined) lead to small shifts in behavior, their overall performance remains similar, demonstrating that planning-based selection provides a practical path toward robust composite model construction at scale.

6 Discussion and Conclusion

There is a critical need for AI assessment methods that scale with the growing complexity of real-world AI systems. Our approach shows that probabilistic planning provides a practical way to assess robustness when assembling composite AI models from multiple primitive models. By treating each model invocation as an action with downstream consequences, the planner anticipates how decisions at one stage affect robustness later in the pipeline. This is particularly important in multi-stage systems where early choices influence downstream behavior. Several trends emerged across our real-world (RTS-Small (T1), RTS-Large (T2)) and synthetic (T3) tasks.

First, the heuristic lookahead strategy achieves strong reward performance but incurs high inference latency because

Agent	Total Reward ↑	Switch Cost ↓	WRS ↓	ATE ↑	DIE ↓	Time (ms) ↓
(30, 15)						
Q-L (WRS)	-11.25	0.02	1.07	0.03	0.14	0.12
Q-L (DIE)	-11.28	0.02	0.93	0.04	0.14	0.13
Q-L (Comb.)	-11.06	0.02	0.98	0.03	0.15	0.14
Heuristic (WRS)	-10.14	0.02	0.79	0.03	0.12	16.57
Fixed (Biased)	-15.45	0.02	1.17	0.03	0.18	0.02
Fixed (Fair)	-11.37	0.02	0.93	0.02	0.15	0.02
Random	-13.85	0.26	0.97	0.03	0.16	0.03
(10, 5)						
Q-L (WRS)	-11.14	0.05	1.01	0.06	0.14	0.06
Q-L (DIE)	-10.93	0.05	1.01	0.06	0.16	0.06
Q-L (Comb.)	-10.97	0.05	0.99	0.06	0.13	0.05
Heuristic (WRS)	-10.45	0.05	0.98	0.04	0.12	4.39
Fixed (Biased)	-16.07	0.05	1.20	0.04	0.18	0.01
Fixed (Fair)	-11.04	0.05	0.79	0.05	0.13	0.01
Random	-14.09	0.27	1.16	0.06	0.19	0.01

Table 2: Performance of Q-Learning (Q-L) policies and baselines on the two synthetic settings: 30-stage composite AI with 15 primitive models available at each stage, and 10 stages and 5 primitive models per stage. Reward and cost are averaged over stages, and Time (ms) reports average inference time per decision. Green cells indicate best-performing values within each task; red cells indicate worst-performing values.

robustness metrics must be recomputed online at each decision. In contrast, the Q-learning variants shift this computation into training, reducing inference to a constant-time lookup. Planning-based selection is most useful when decisions at one stage affect what becomes optimal later. In our setting this occurs in two ways: early transformations (e.g., translation or paraphrasing) alter downstream model behavior and therefore final robustness metrics, and in the synthetic chain an explicit switching cost makes the best model at stage n depend on the family chosen at stage $(n-1)$. In such cases, greedy rules that optimize immediate reward can incur switching penalties or select locally beneficial steps that degrade end-to-end robustness, whereas planning accounts for these longer-horizon effects. When stage decisions are largely independent (e.g., no switching penalties), greedy strategies can coincide with the planner's selection.

Second, the three Q-learning reward formulations produce broadly similar policies despite optimizing different objectives. Q-Learning (Combined) achieves the lowest WRS and DIE, indicating that joint optimization is stable. Notably, Q-Learning (DIE) often matches, and in the deeper synthetic configuration even improves on, the WRS achieved by Q-Learning (WRS). This suggests that controlling for confounding can also reduce group-level disparities. Across most comparisons, statistical tests show no significant differences among the Q-learning variants, suggesting practitioners can choose objectives aligned with their application without sacrificing reward or fairness.

Overall, our experiments show that probabilistic planning provides a feasible approach for robustness-aware model selection in larger operational settings. The planner remains stable across small, large, and synthetic environments, scales to deeper chains with many primitive models, and exposes interpretable cost–fairness trade-offs (e.g., switching cost vs. robustness in T3). This approach is relevant wherever complex AI systems are assembled from multiple components, including decision-support workflows, document-processing pipelines, and conversational systems where robustness must be evaluated end-to-end.

Future work may extend this framework in several directions. One direction is to replace the tabular learner with function approximation or hierarchical planning to handle composite models with hundreds of stages or richer branching structures. Another is to incorporate online feedback so the planner can adapt when primitive model behavior or robustness requirements change over time. Finally, the formulation could be extended to richer causal models with multiple path-specific effects or more complex pipeline structures, such as parallel branches or partially shared stages, allowing robustness-aware planning in larger production-scale composite AI systems.

Acknowledgements

This work is partially supported by NSF Awards #2454027 and Faculty Award by JP Morgan Research.

References

Bernagozzi, M.; Srivastava, B.; Rossi, F.; and Usmani, S. 2021. VEGA: a Virtual Environment for Exploring Gender Bias vs. Accuracy Trade-offs in AI Translation Services. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(18): 15994–15996.

Carey, A. N.; and Wu, X. 2022. The causal fairness field guide: Perspectives from social and formal sciences. *Frontiers in Big Data*, 5.

Christiansen, J. G.; Gammelgaard, M.; and Søgaard, A. 2021. The Effect of Round-Trip Translation on Fairness in Sentiment Analysis. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 4423–4428. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Dwork, C.; and Ilvento, C. 2018. Fairness under composition. *arXiv preprint arXiv:1806.06122*.

Greenland, S.; and Robins, J. M. 1986. Identifiability, exchangeability, and epidemiological confounding. *International journal of epidemiology*, 15(3): 413–419.

Hu, Y.; and Zhang, L. 2022. Achieving long-term fairness in sequential decision making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 9549–9557.

Kailkhura, B.; Chen, P.-Y.; Lin, X.; and Li, B. 2021. Safe and Trustworthy Machine Learning.

Kusner, M. J.; Loftus, J.; Russell, C.; and Silva, R. 2017. Counterfactual Fairness. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Lakkaraju, K. 2022. Why is My System Biased?: Rating of AI Systems through a Causal Lens. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '22, 902. New York, NY, USA: Association for Computing Machinery. ISBN 9781450392471.

Lakkaraju, K.; Gupta, A.; Srivastava, B.; Valtorta, M.; and Wu, D. 2023. The Effect of Human v/s Synthetic Test Data and Round-Tripping on Assessment of Sentiment Analysis Systems for Bias. In *2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 380–389. Los Alamitos, CA, USA: IEEE Computer Society.

Lakkaraju, K.; Hassan, T.; Khandelwal, V.; Singh, P.; Bradley, C.; Shah, R.; Agostinelli, F.; Srivastava, B.; and Wu, D. 2022. Allure: A multi-modal guided environment for helping children learn to solve a rubik's cube with automatic solving and interactive explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 13185–13187.

Lakkaraju, K.; Kaur, R.; Zehtabi, P.; Patra, S.; Valluru, S. L.; Zeng, Z.; Srivastava, B.; and Valtorta, M. 2025. On Creating a Causally Grounded Usable Rating Method for Assessing the Robustness of Foundation Models Supporting Time Series. *arXiv preprint arXiv:2502.12226*.

Lakkaraju, K.; Kaur, R.; Zeng, Z.; Zehtabi, P.; Patra, S.; Srivastava, B.; and Valtorta, M. 2024a. Rating Multi-Modal Time-Series Forecasting Models (MM-TSFM) for Robustness Through a Causal Lens. *arXiv preprint arXiv:2406.12908*.

Lakkaraju, K.; Khandelwal, V.; Srivastava, B.; Agostinelli, F.; Tang, H.; Singh, P.; Wu, D.; Irvin, M.; and Kundu, A. 2024b. Trust and ethical considerations in a multi-modal, explainable AI-driven chatbot tutoring system: The case of collaboratively solving Rubik's Cube. *CoRR*.

Lakkaraju, K.; Patra, S.; Zehtabi, P.; and Srivastava, B. 2026. Assessing the Robustness of Composite AI Models via Probabilistic Planning. https://github.com/kausik-l/composite-rating-rddl/blob/main/ICAPS2026_Paper_Extended_Version.pdf.

Lakkaraju, K.; Srivastava, B.; and Valtorta, M. 2024. Rating Sentiment Analysis Systems for Bias Through a Causal Lens. *IEEE Transactions on Technology and Society*, 1–1.

Mangal, R.; Wang, Z.; Zhang, C.; Leino, K.; Pasareanu, C.; and Fredrikson, M. 2022. On the perils of cascading robust classifiers. *arXiv preprint arXiv:2206.00278*.

Mordoch, A.; Juba, B.; and Stern, R. 2023. Learning safe numeric action models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 12079–12086.

Pullum, L. 2021. Verification and Validation of Systems in which AI is a Key Element. Technical report, Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States).

Sanner, S. 2011. Relational Dynamic Influence Diagram Language (RDDL): Language Description. Technical report, ICAPS 2011 Planning Competition. Technical Report.

Srivastava, B.; Lakkaraju, K.; Bernagozzi, M.; and Valtorta, M. 2023. Advances in Automatically Rating the Trustworthiness of Text Processing Services. arXiv:2302.09079.

Srivastava, B.; and Rossi, F. 2019. Towards Composable Bias Rating of AI Services. arXiv:1808.00089.

Srivastava, B.; Rossi, F.; Usmani, S.; and Bernagozzi, M. 2020. Personalized Chatbot Trustworthiness Ratings. *IEEE Transactions on Technology and Society*, 1(4): 184–192.

Verma, S.; and Rubin, J. 2018. Fairness definitions explained. In *2018 ieee/acm international workshop on software fairness (fairware)*, 1–7. IEEE.