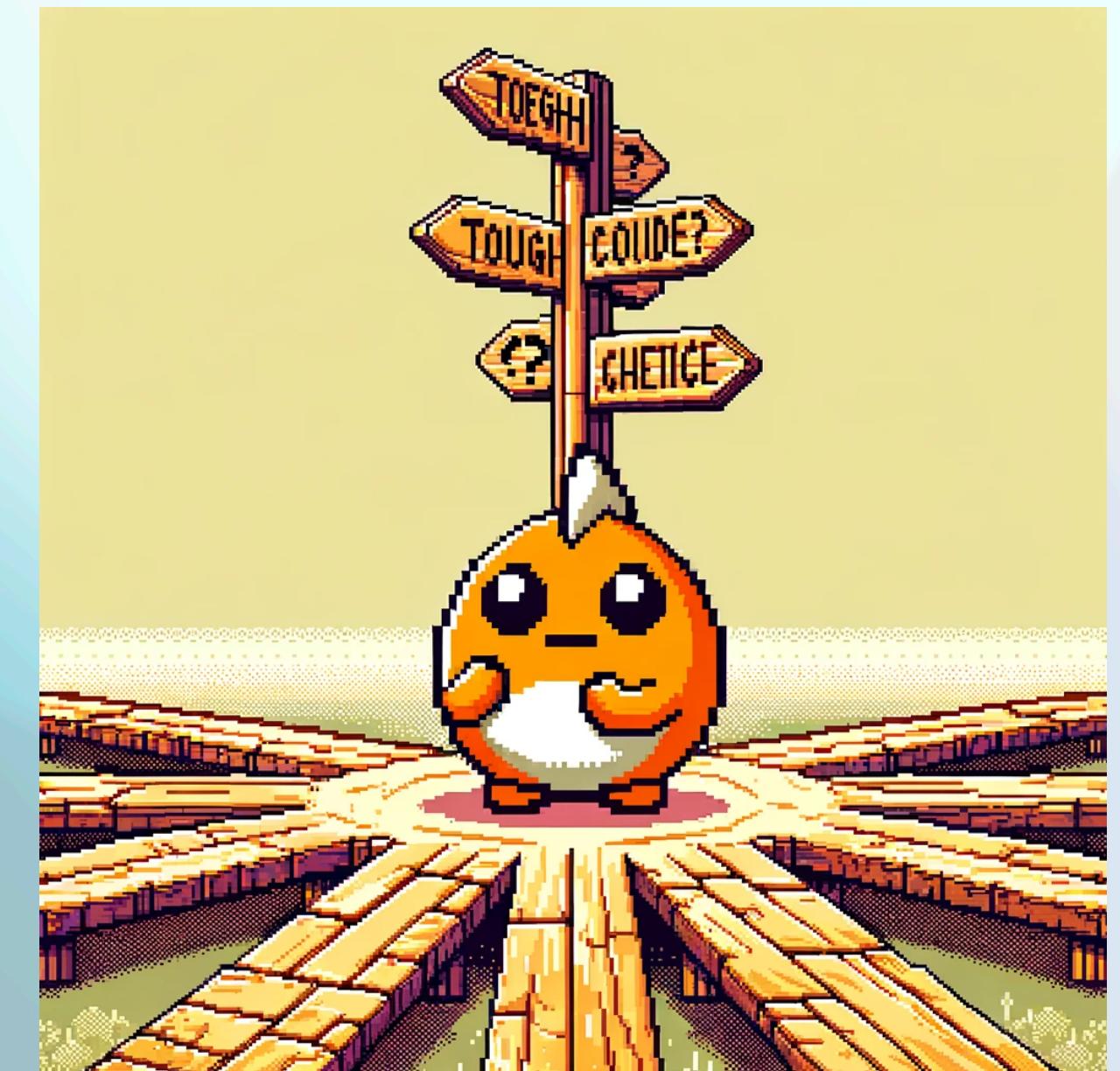


DeLLMa: A Framework for Decision Making Under Uncertainty With Large Language Models

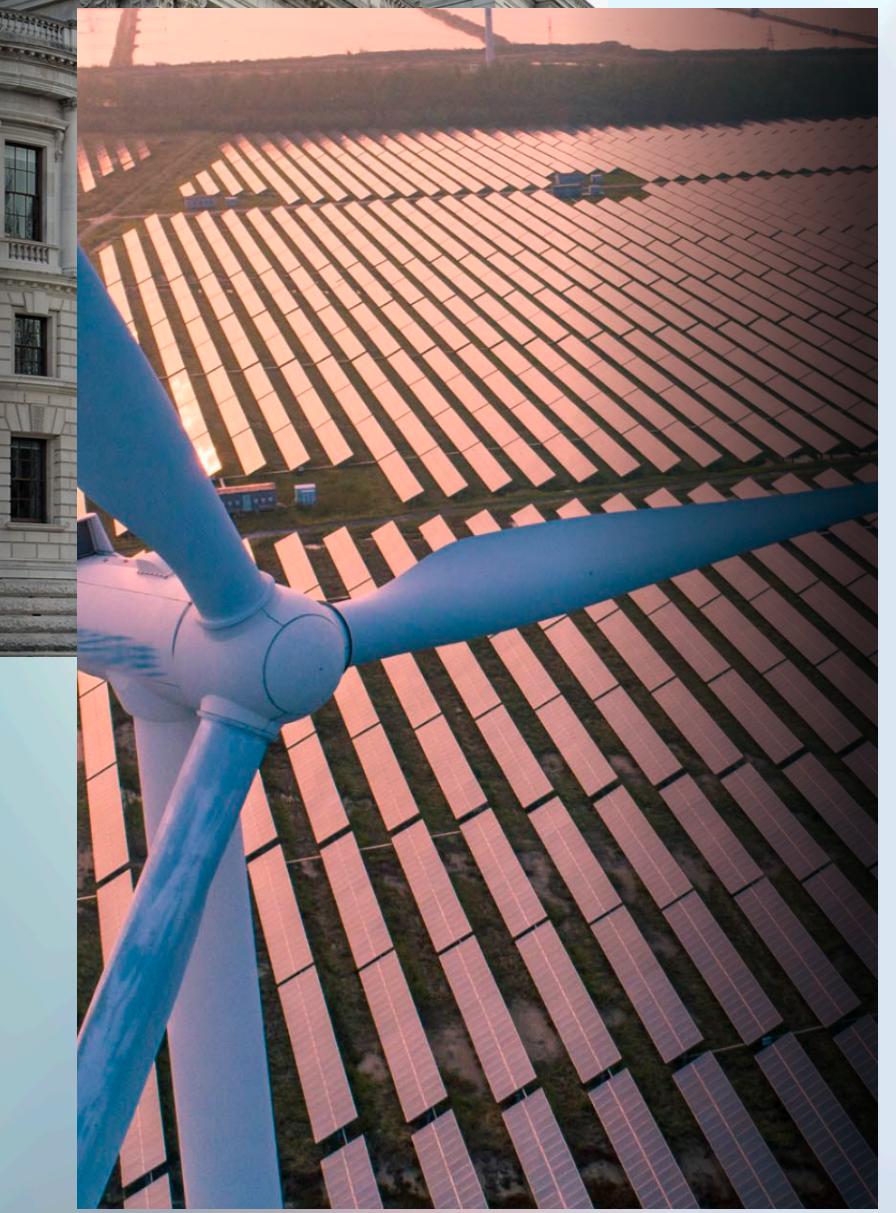
DeLLMa.github.io

Ollie Liu*^{*}, Deqing Fu^{*}, Dani Yogatama, Willie Neiswanger



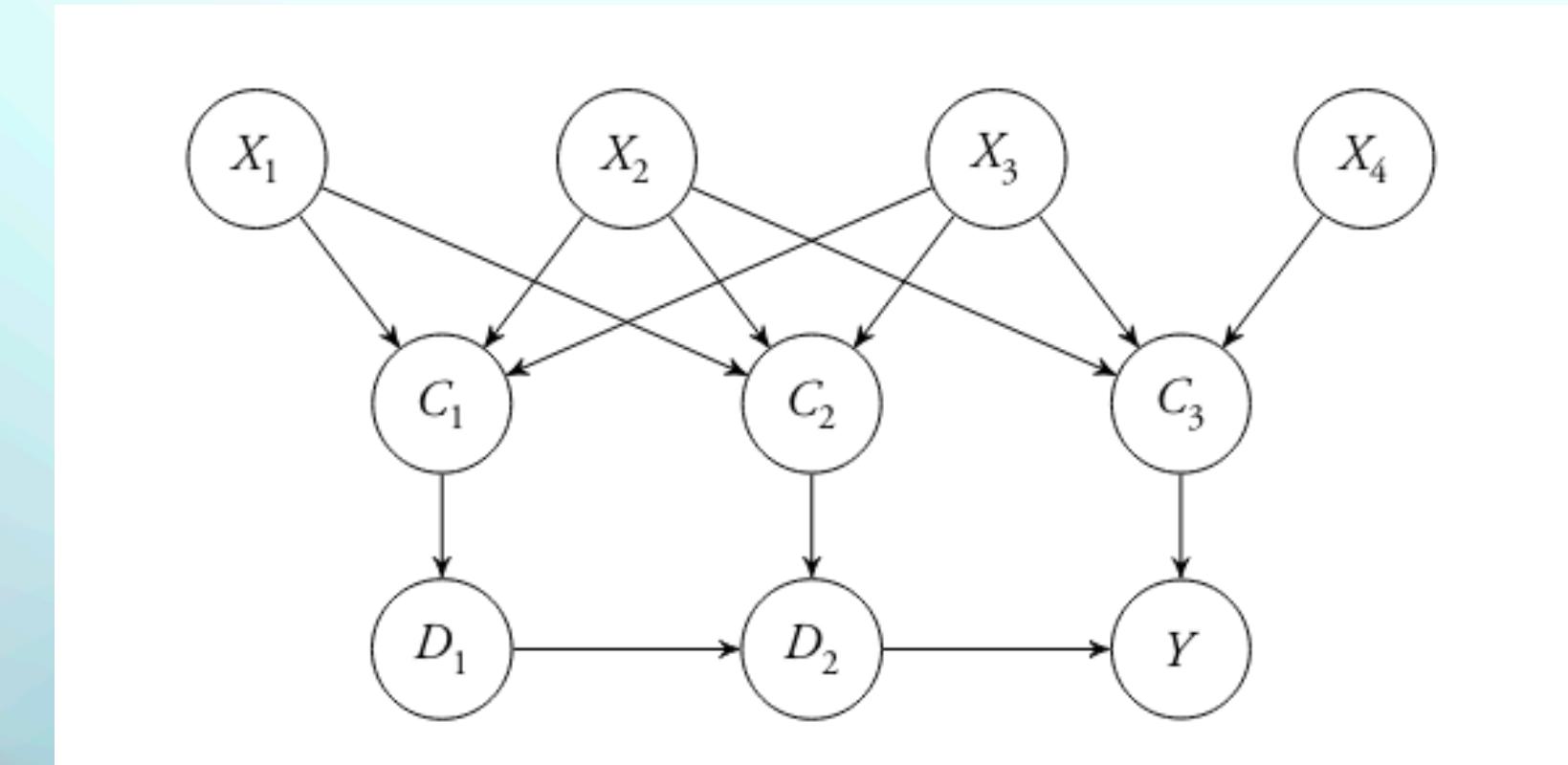
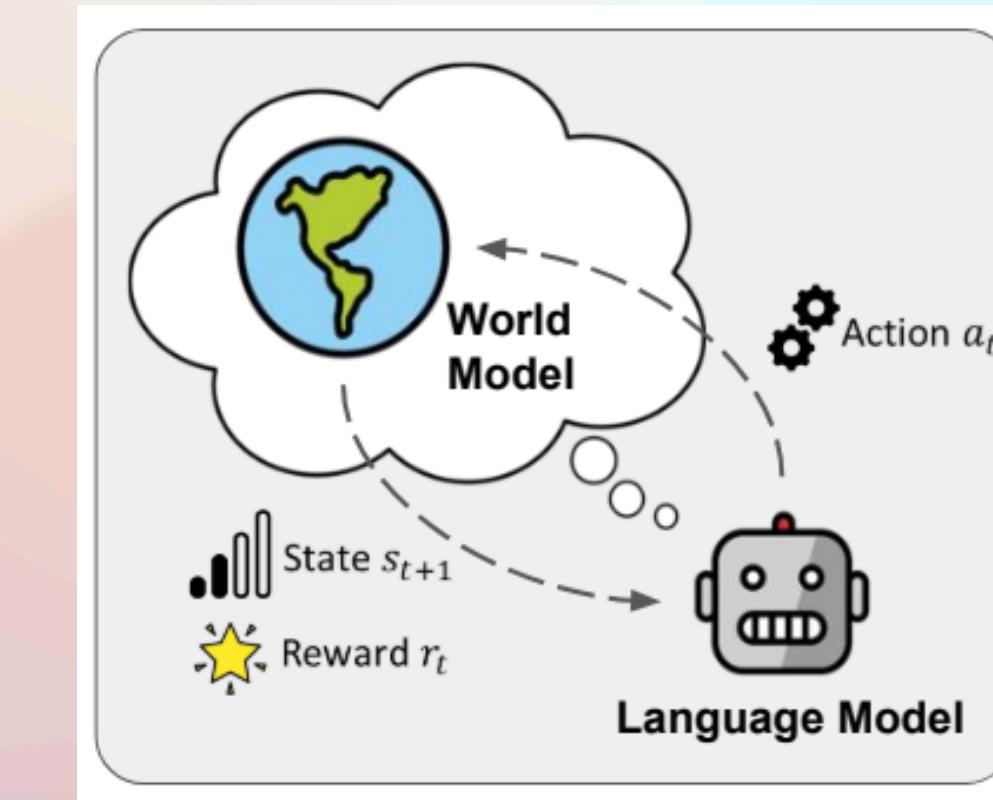
Decision Making Under Uncertainty Is...

- Mathematical models to guide us towards rational and optimal decisions.
- Ubiquitous in our lives:
 - Agriculture planning [[Parsons et al., 2022](#)],
 - Market Investing [[Hallegatte et al., 2012](#)],
 - Medical Treatments [[Wickett et al., 2023](#)],
 - Scientific Discovery [[Lookman et al., 2019](#)].
- Extensively studied in economics, statistics, and philosophy [[Von Neumann & Morgenstern, 1944](#), [Luce & Raiffa, 1989](#); [Berger, 2013](#)].



Why Use LLMs For Decision Making?

- LLMs are powerful common sense reasoners [[Hao et al., 2023](#)].
- They provide a more user-friendly interface than statistical approaches.
- They are “reasonable” out-of-the-box forecasters [[Halawi et al., 2024](#)].
- But they are not out-of-the-box decision makers...



From [Hao et al., 2023](#)

Platform	Train	Validation	Test	Model	Zero-shot	Scratchpad
Metaculus	1,576	230	275	GPT-4-1106-Preview	0.208 (0.006)	0.209 (0.006)
GJOpen	806	161	38	Llama-2-13B	0.226 (0.004)	0.268 (0.004)
INFER	52	50	4	Mistral-8x7B-Instruct	0.238 (0.009)	0.238 (0.005)
Polymarket	70	229	300	Claude-2.1	0.220 (0.006)	0.215 (0.007)
Manifold	1,258	170	297	Gemini-Pro	0.243 (0.009)	0.230 (0.003)
All Platforms	3,762	840	914	Trimmed mean	0.208 (0.006)	0.224 (0.003)

(a) Dataset distribution

(b) Baseline performance of pre-trained models

Forecasting performance from [Halawi et al., 2024](#). Plot (b) reports Brier scores; lower is better. Human crowd performance: 0.149; random baseline: 0.250.

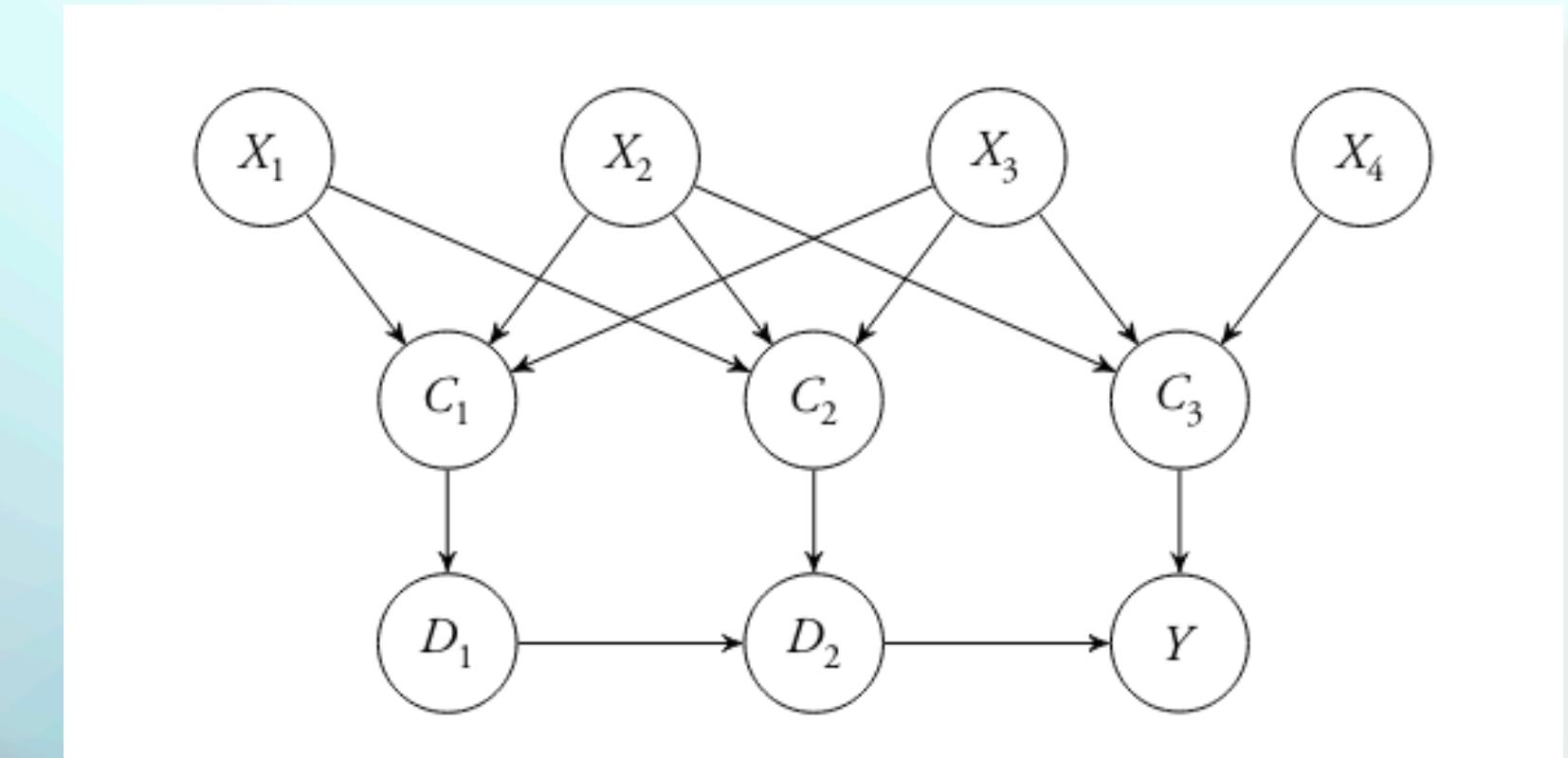
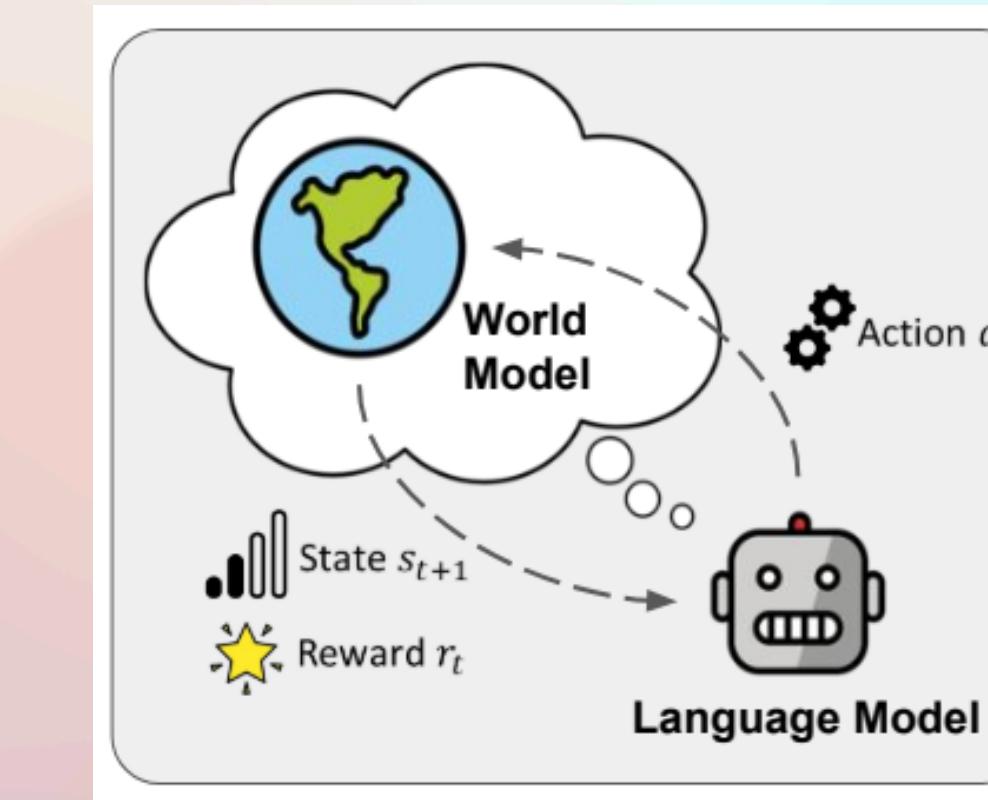
Why Use LLMs For Decision Making? vs.

- **Ground Truth: Avocado**
- **Context (Market Overview Omitted):**
 - **Avocado Product Summary:** California avocado production has decreased, with wildfires and water restrictions impacting yields. However, U.S. avocado consumption has increased significantly, with imports from Mexico and Peru growing substantially. Mexico dominates the U.S. avocado market, with imports peaking from May through July. Peruvian imports compete during the summer months, traditionally a period of lower Mexican imports. The average avocado yield is 2.87 tons/acre and the average price per unit is 2,430 \$/ton.
 - **Grape Product Summary:** Grape production is forecasted to be up 9 percent from 2020, despite drought and heat conditions. California table-type grape production is also expected to increase. High heat has affected the industry, with Coachella valley shipments down and central California shipments up. Imports from Mexico, the main competitor, were down slightly, but overall shipments were higher in 2021 than the previous year. The average grape yield is 6.92 tons/acre and the average price per unit is 908 \$/ton.
- **Instruction**

I'm a farmer in California planning what fruit to plant next year. I would like to maximize my profit with "10" acres of land. Below are the actions I can take: **Action 1.** Avocado: 10 acres; **Action 2.** Grape: 10 acres
- **GPT-4 Prediction: Grape**
- **GPT-4 Response:** ... although avocados have a higher price per ton, the significantly higher yield of grapes per acre makes grape production more profitable on a per-acre basis. Additionally, the grape production is forecasted to be up despite the drought and heat conditions, indicating a resilient crop under current conditions. In contrast, avocado production has decreased due to wildfires and water restrictions, which may pose a higher risk. Therefore, planting grapes is recommended to maximize profit on the 10 acres of land.

Why Use LLMs For Decision Making?

- LLMs are powerful common sense reasoners [Hao et al., 2023].
- They provide a more user-friendly interface than statistical approaches.
- They are “reasonable” out-of-the-box forecasters [Halawi et al., 2024].



From Hao et al., 2023

- But they are not out-of-the-box decision makers...because they:
 - Are too “good” at in-context learning!
 - Cannot reason with stochasticity [Feng et al., 2024].

Platform	Train	Validation	Test	Model	Zero-shot	Scratchpad
Metaculus	1,576	230	275	GPT-4-1106-Preview	0.208 (0.006)	0.209 (0.006)
GJOpen	806	161	38	Llama-2-13B	0.226 (0.004)	0.268 (0.004)
INFER	52	50	4	Mistral-8x7B-Instruct	0.238 (0.009)	0.238 (0.005)
Polymarket	70	229	300	Claude-2.1	0.220 (0.006)	0.215 (0.007)
Manifold	1,258	170	297	Gemini-Pro	0.243 (0.009)	0.230 (0.003)
All Platforms	3,762	840	914	Trimmed mean	0.208 (0.006)	0.224 (0.003)

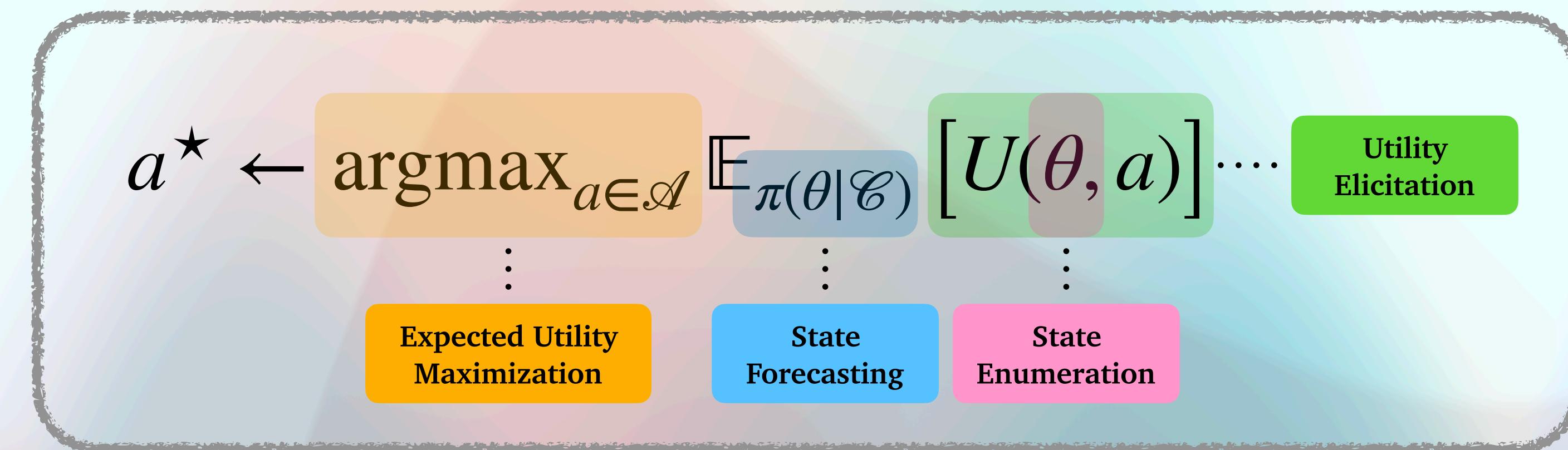
(a) Dataset distribution

(b) Baseline performance of pre-trained models

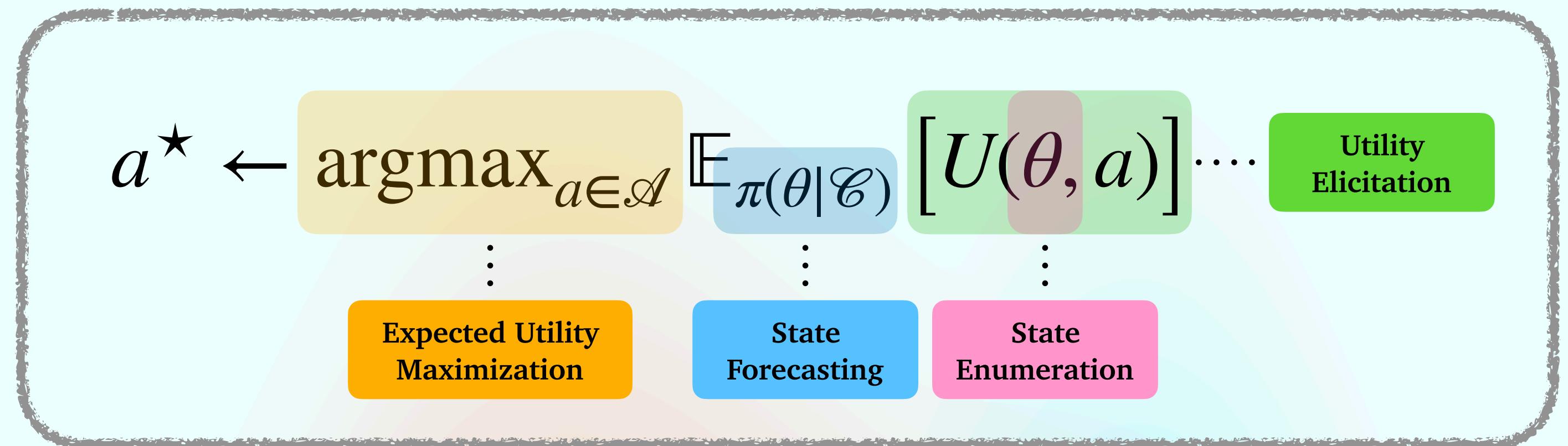
Forecasting performance from Halawi et al., 2024. Plot (b) reports Brier scores; lower is better. Human crowd performance: 0.149; random baseline: 0.250.

Decision Making with LLMs

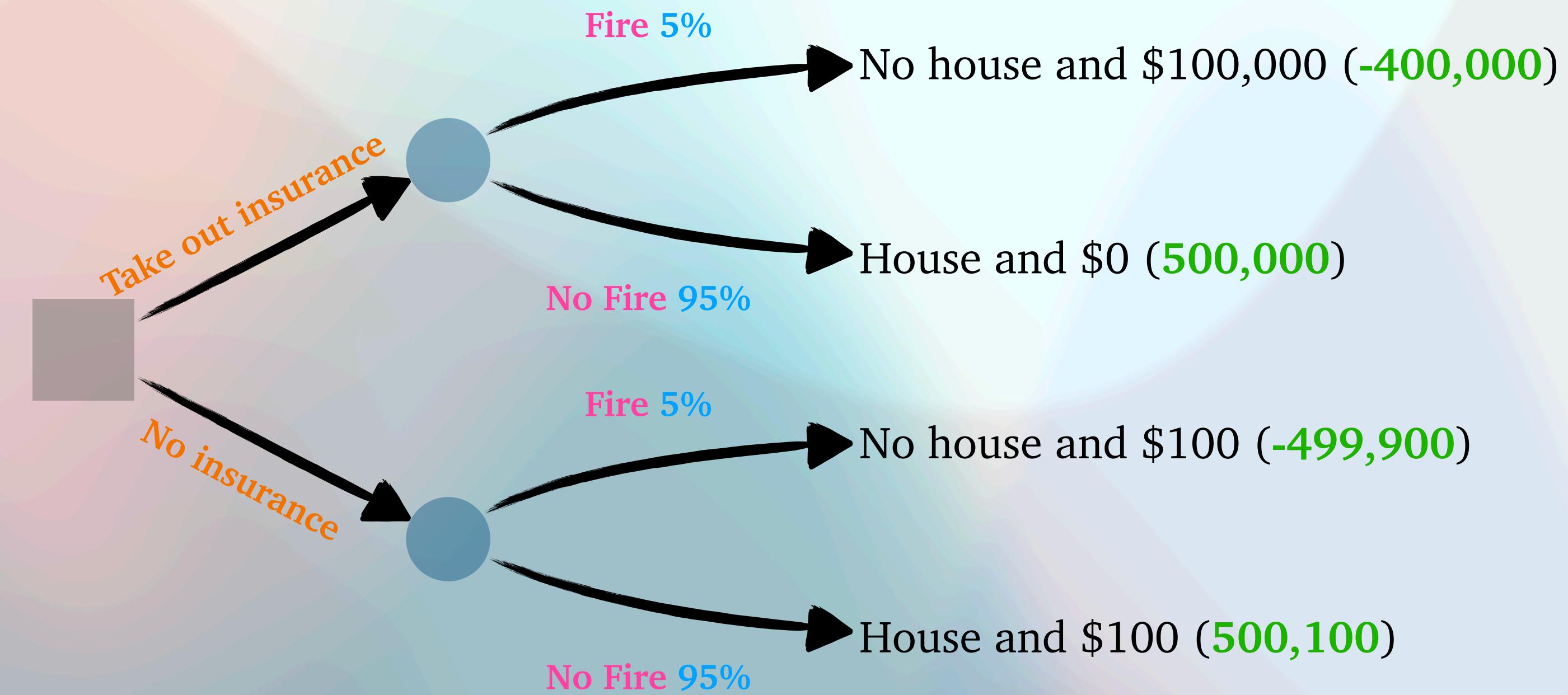
- Goal: search for actions that maximize the expected utility.



- DeLLMa: Decision-making Large Language Model **assistant**



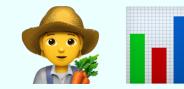
Action/State	Fire	No Fire
Take out insurance	No house and \$100,000	House and \$0
No insurance	No house and \$100	House and \$100



Formal Procedures of DeLLMa



Max Revenue



DELLMA: AN ASSISTANT FOR LLM DECISION MAKING UNDER UNCERTAINTY

Input: A user prompt $\mathcal{P} = (\mathcal{G}, \mathcal{A}, \mathcal{C})$ consisting of a user's goal \mathcal{G} , actions $\mathcal{A} = (a_1, \dots, a_n)$, and context \mathcal{C} .

1. **State Enumeration:** Produce a list of m states $\Theta = (\theta_1, \dots, \theta_m)$, which are unknown quantities whose values are predicted to influence the user's goal \mathcal{G} . ▷ Described in Section 3.1.
2. **State Forecasting:** For each state θ_j , produce a probabilistic forecast $\pi(\theta_j \mid \mathcal{C})$, which describes the probability of different values of this state given context \mathcal{C} . ▷ Described in Section 3.2.
3. **Utility Function Elicitation:** Produce a *utility function* $U : (\theta_j, a_i) \rightarrow \mathbb{R}$, which assigns a real value to each state-action pair (θ_j, a_i) , based on the user's goal \mathcal{G} . ▷ Described in Section 3.3.
4. **Expected Utility Maximization:** For each action a_i , compute the expected utility $U_{\mathcal{C}}(a_i) = \mathbb{E}_{\pi(\theta \mid \mathcal{C})} [U(\theta, a_i)]$, and then return the decision $a^* = \arg \max_{i \in \{1, \dots, n\}} U_{\mathcal{C}}(a_i)$ to the user. ▷ Described in Section 3.4.

State Enumeration

$$a^* \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} \mathbb{E}_{\pi(\theta|\mathcal{C})} [U(\theta, a)]$$

- Goal: produce a list of m states:

$$\Theta = \{\theta_1, \dots, \theta_m\}$$

- Each state θ is a state of k factors:

$$\theta = (f_1, \dots, f_k)$$

- Each factor f may take l plausible values:

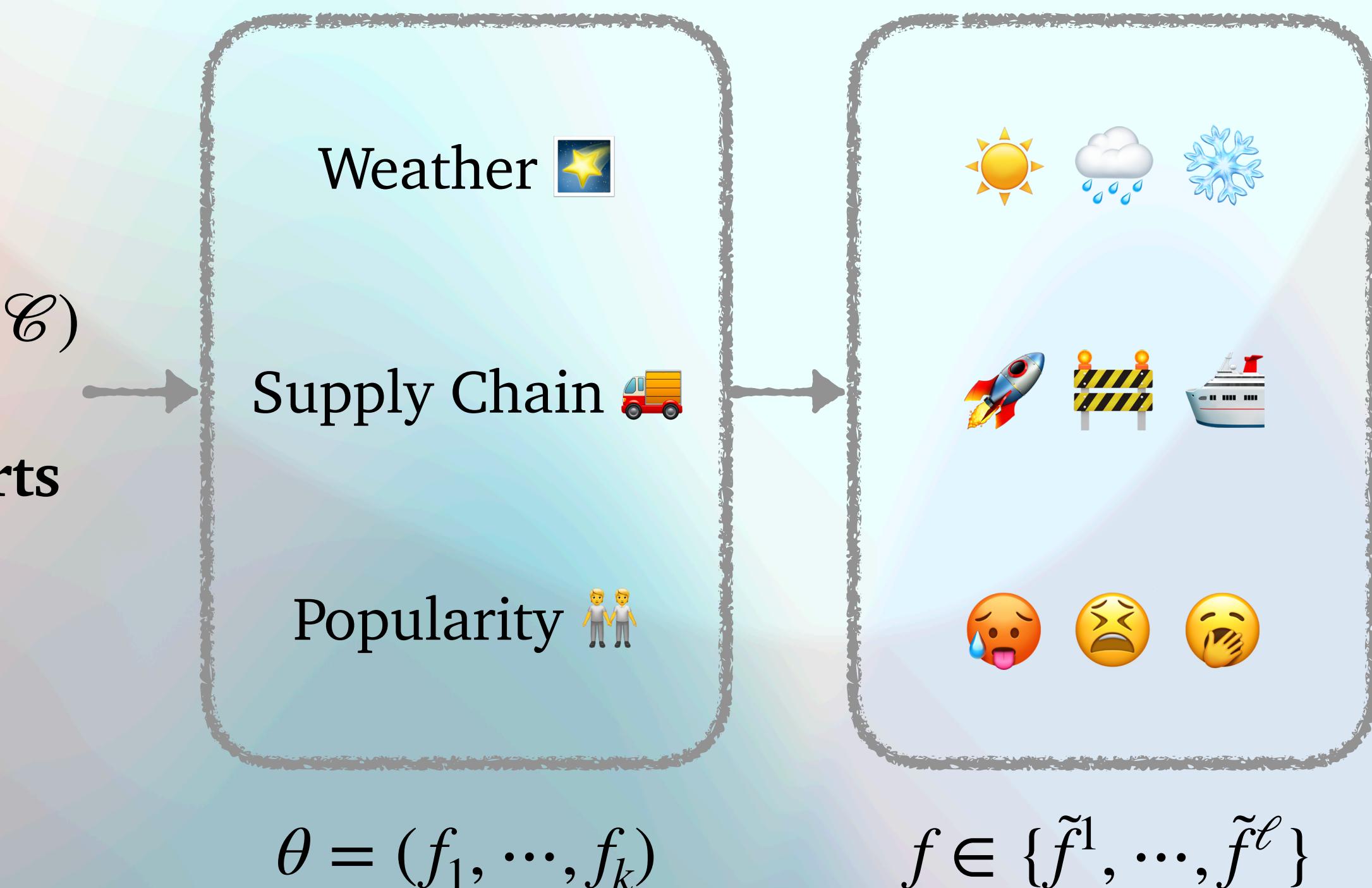
$$f \in \{\tilde{f}^1, \dots, \tilde{f}^\ell\}$$

Generated by an LLM \mathcal{M} :

$$\{f_i, \tilde{f}_i^{1:\ell}\}_{i=1}^k \leftarrow \mathcal{M}(\mathcal{P})$$

$$\mathcal{P} = (\mathcal{G}, \mathcal{A}, \mathcal{C})$$


USDA reports



State Forecasting

$$a^* \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} \mathbb{E}_{\pi(\theta|\mathcal{C})} [U(\theta, a)]$$

- Goal: approximate $\pi(\theta | \mathcal{C})$ with $\pi^{\text{LLM}}(\theta | \mathcal{C})$.
- But there are $m = \ell^k$ unique states!
- We prompt to generate k conditional distributions, one for each f_i .
- Define $\pi^{\text{LLM}}(\theta = (f_1, \dots, f_k) | \mathcal{C})$ as their product distribution.



Apply \mathcal{V}	1	4	3
Normalize	1/8	1/2	3/8

Algorithm 1 STATEFORECAST($\mathcal{M}, \mathcal{P}, \mathcal{V}, \{f_i, \tilde{f}_i^{1:\ell}\}_{i=1}^k$)

Input: LLM \mathcal{M} , user prompt $\mathcal{P} = (\mathcal{G}, \mathcal{A}, \mathcal{C})$, plausibility score mapping \mathcal{V} , latent factors $\{f_1, \dots, f_k\}$, and plausible values $\{\tilde{f}_1^{1:\ell}, \dots, \tilde{f}_k^{1:\ell}\}$.

```

for  $i = 1$  to  $k$  do
    # PMF for the latent factor  $f_i$ 
     $\pi_i(\cdot | \mathcal{C}) \leftarrow \{\}$ 
    # Get verbalized probability scores
     $[v_1, \dots, v_\ell] \leftarrow \mathcal{M}(\mathcal{P}, f_i, \tilde{f}_i^{1:\ell})$ 
    for  $j = 1$  to  $\ell$  do
         $\pi_j(\tilde{f}_i^j | \mathcal{C}) \leftarrow \mathcal{V}[v_j]$ 
    end for
     $\pi_i(\cdot | \mathcal{C}) \leftarrow \text{Normalize}(\pi_i(\cdot | \mathcal{C}))$ 
end for
return  $\pi^{\text{LLM}}(f_1, \dots, f_k | \mathcal{C}) := \prod_{i=1}^k \pi_i(\cdot | \mathcal{C})$ 

```

Plausibility Mapping \mathcal{V} :

LLM Belief	Score
Very Likely	6
...	...
Very Unlikely	1

Utility Function Elicitation

$$a^* \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} \mathbb{E}_{\pi(\theta|\mathcal{C})} [U(\theta, a)]$$

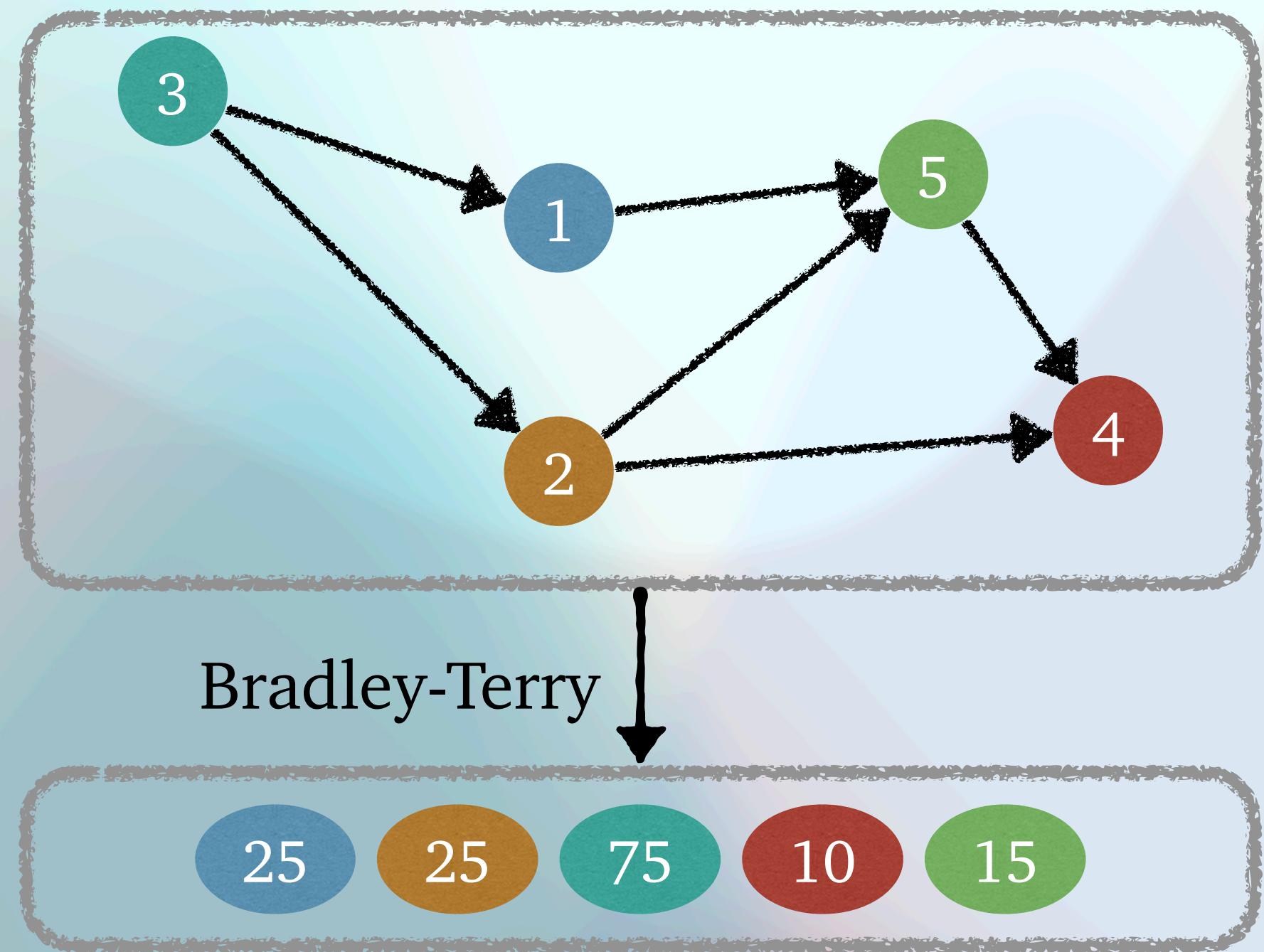
- **Goal:** Produce an utility function $U : \Theta \times \mathcal{A} \rightarrow \mathbb{R}$
- **Key Idea:** Prompt the LLM rank state-action pairs, i.e. is $(\theta, a)_i$ preferred over $(\theta, a)_j$ for some i and j ?
- Collect a dataset of pairwise comparisons:

$$\Omega = \{ (\theta, a)_i \succ (\theta, a)_j \}$$

- Utility elicitation with a Bradley-Terry model!

$$U(\cdot, \cdot) := \text{Bradley-Terry}(\Omega) \in \mathbb{R}^s$$

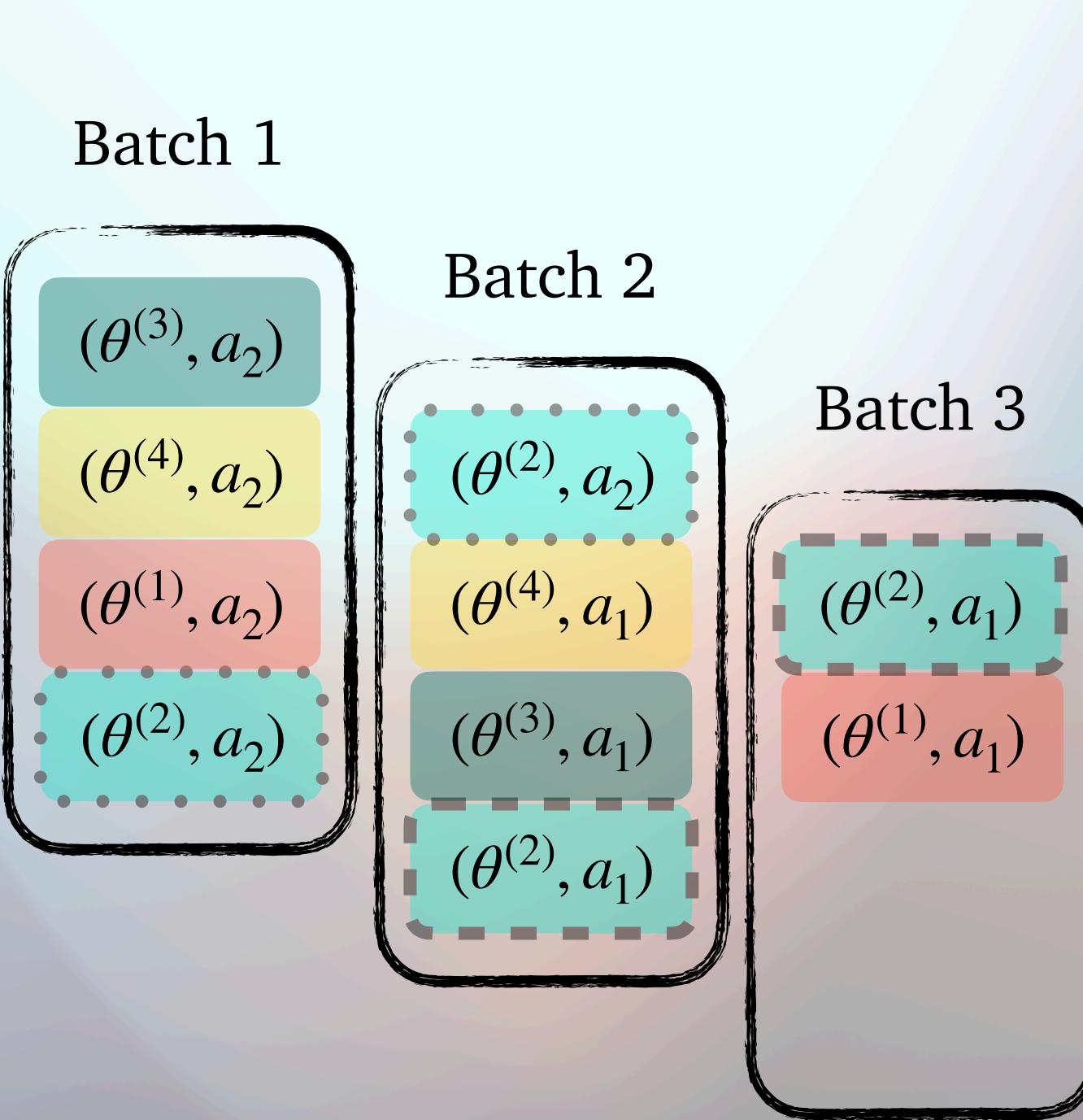
- How can we elicit pairwise comparisons efficiently?



Efficient Utility Elicitation

$$a^* \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} \mathbb{E}_{\pi(\theta|\mathcal{C})} [U(\theta, a)]$$

- Desiderata:
 - Accurate utility (*i.e.* high-quality comparisons);
 - Small number of API queries.



Sampling via a Python program

.....

Convert rank into pairwise comparisons

.....

Algorithm 2 UTILITYELICITATION($\mathcal{M}, \mathcal{P}, \pi^{\text{LLM}}, s, b, q$)

Input: LLM \mathcal{M} , user prompt $\mathcal{P} = (\mathcal{G}, \mathcal{A}, \mathcal{C})$, proposal distribution $\pi^{\text{LLM}}(\cdot | \mathcal{C})$, sample size s , minibatch size b , and overlap proportion q .

```

# Sample fixed states for each action
 $S_A \leftarrow \mathcal{A} \times \{\theta_i \mid \theta_i \sim \pi^{\text{LLM}}(\cdot | \mathcal{C}), 1 \leq i \leq \lfloor s/|\mathcal{A}| \rfloor\}$ 
 $S_A \leftarrow \text{shuffle}(S_A)$ 
 $\Omega \leftarrow \{\}$  # Pairwise comparisons
for  $i = 1$  to  $s$  with step  $\lfloor b \times (1 - q) \rfloor$  do
    # Rank the minibatch
     $\mathcal{R} \leftarrow \mathcal{M}(\mathcal{P}, (\theta_i, a_i), \dots, (\theta_{i+b}, a_{i+b}))$ 
    # Format into comparison & update
     $\Omega \leftarrow \Omega \cup \text{FormatRank}(\mathcal{R})$ 
end for
return  $U(\cdot, \cdot) := \text{BradleyTerry}(\Omega) \in \mathbb{R}^s$ 

```

Maximize Expected Utility

$$a^* \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} \mathbb{E}_{\pi(\theta|\mathcal{C})} [U(\theta, a)]$$

- **Goal:** search for actions that maximize the Monte-Carlo estimates of the expected utility!

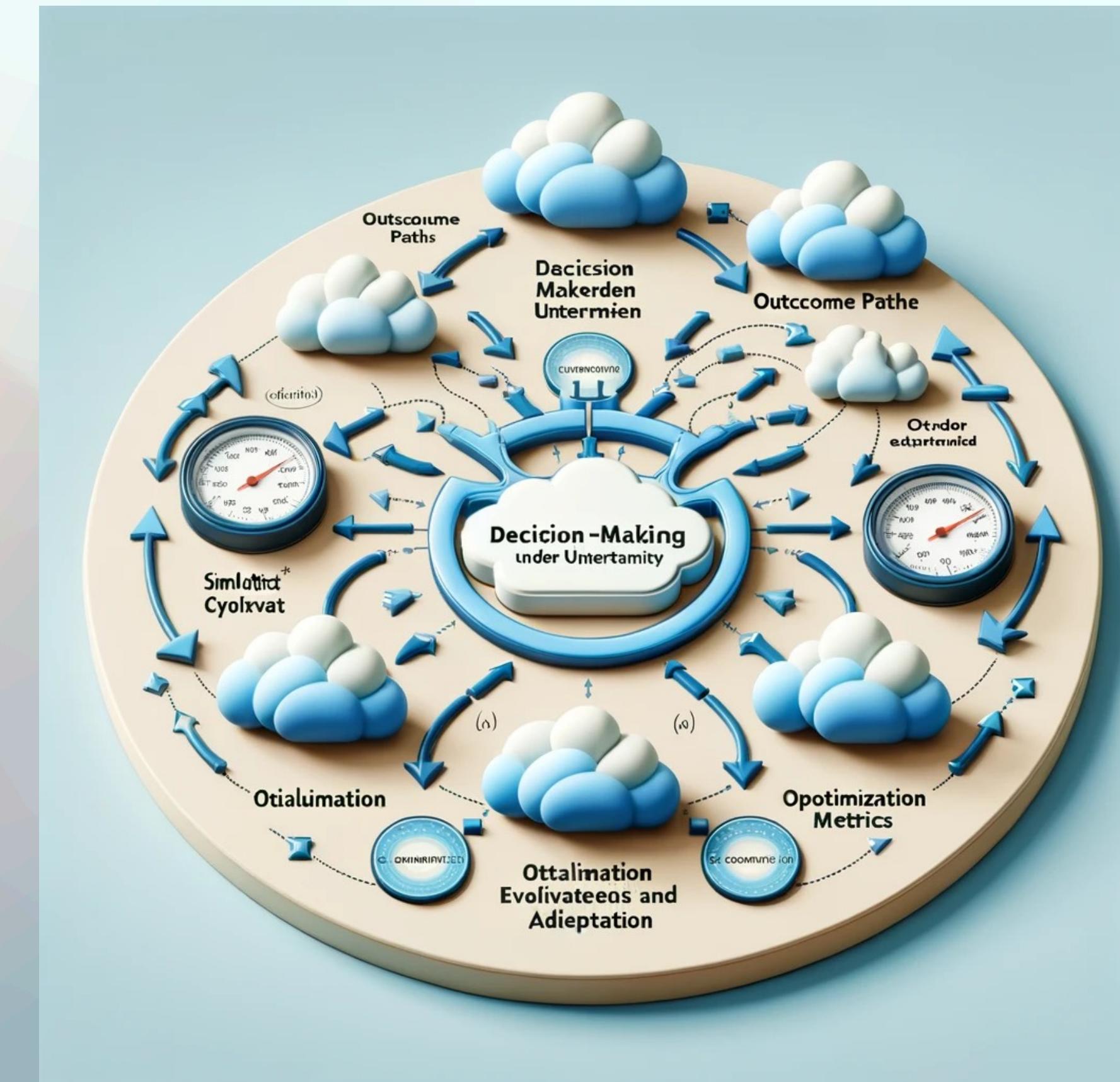
- MC estimates:

$$U_{\mathcal{C}}(a) = \mathbb{E}_{\pi(\theta|\mathcal{C})}[U(\theta, a)] \approx \frac{1}{|S|} \sum_{\theta \in S} U(\theta, a)$$

- Optimization by enumeration:

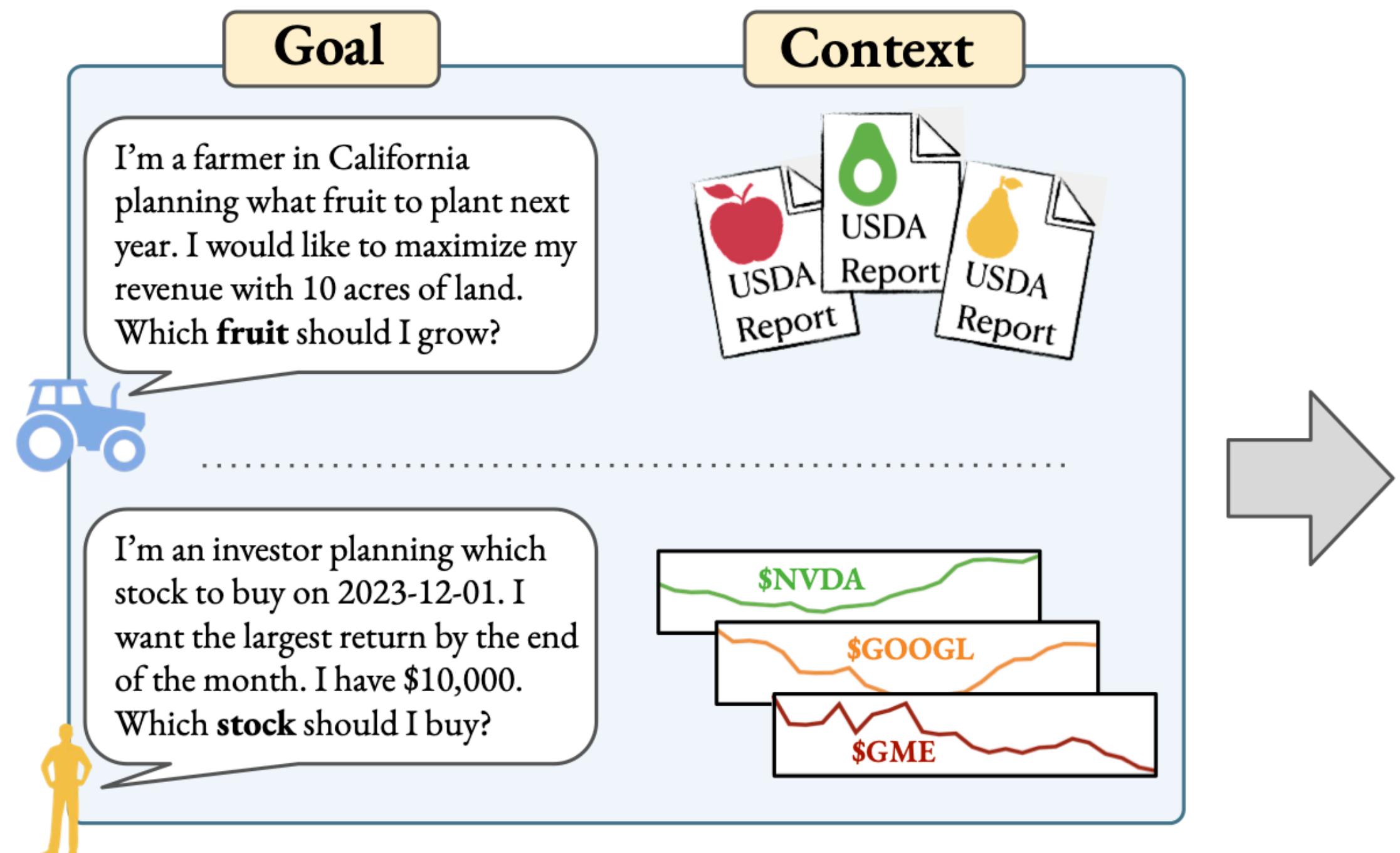
$$a^* = \operatorname{argmax}_{a \in \mathcal{A}} U_{\mathcal{C}}(a)$$

- This step is performed analytically without LLMs!

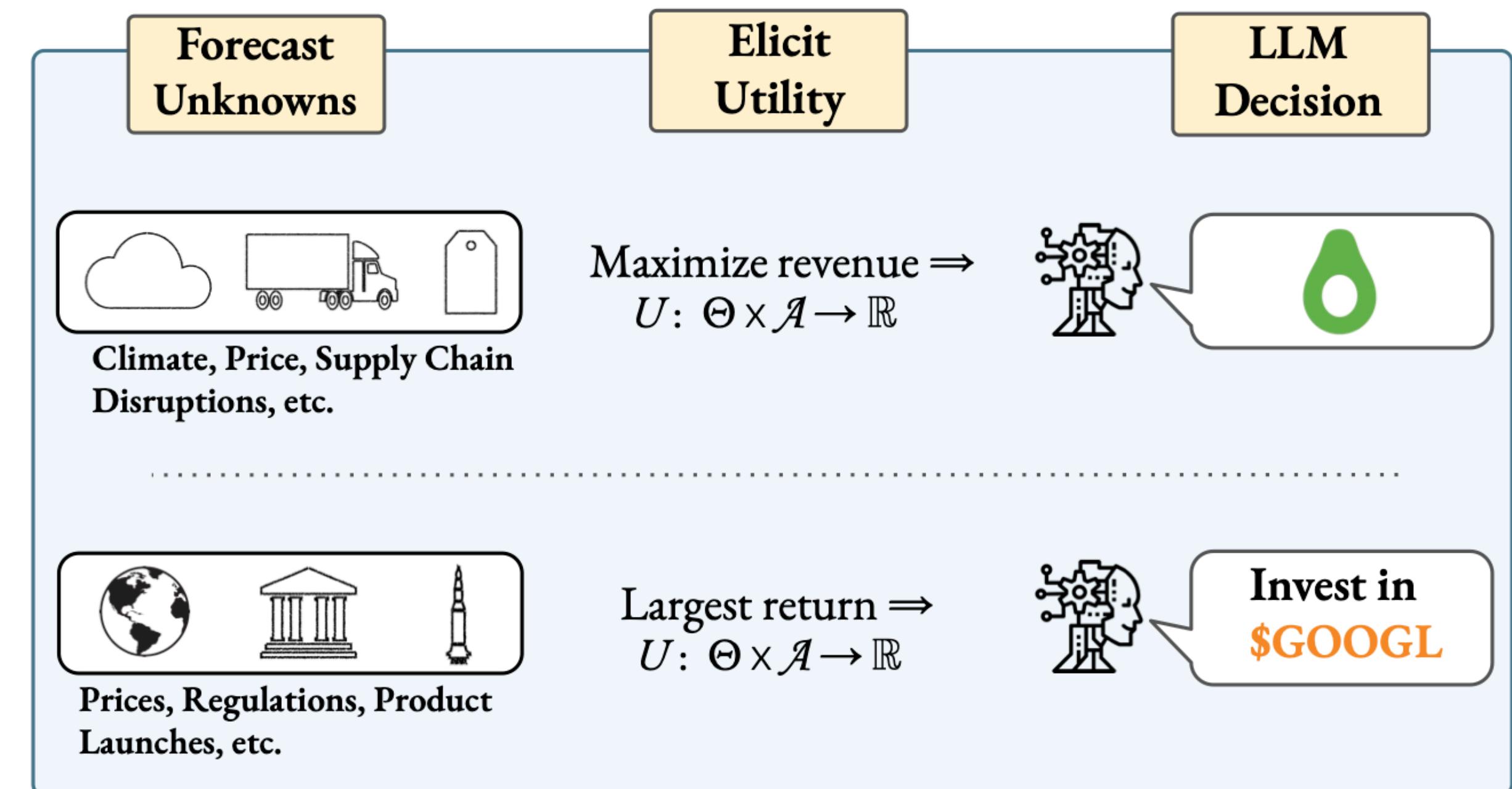


Putting Everything Together

Input: Decision Query from User



DeLLMa: Decision-making LLM assistant



Experiments - Datasets



- Two real datasets curated by the authors:
 - **Agriculture Planning:**
 - **Context \mathcal{C} :** USDA reports & past-year produce price and yield;
 - **Goal \mathcal{G} :** Maximize revenue for next year;
 - **Action \mathcal{A} :** Select one produce to plant (7 in total).
 - **Finance Investing:**
 - **Context \mathcal{C} :** Stock price histories over a 24-month window.
 - **Goal \mathcal{G} :** Maximize return for a one-month trading window.
 - **Action \mathcal{A} :** Select one stock to invest (7 in total).



GPT-4 Preview-1106

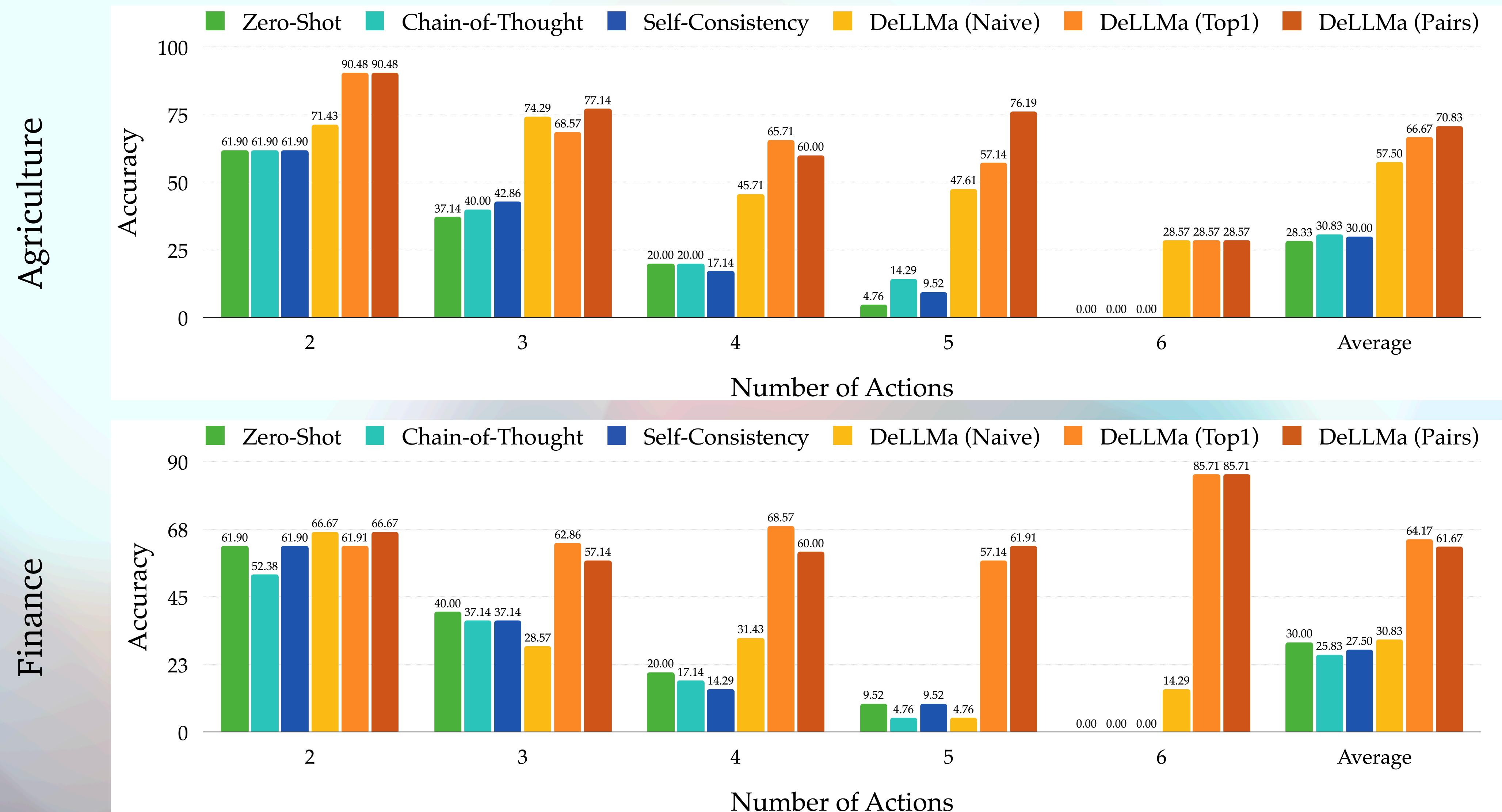
💹 11/2021 — 11/2023

🤔 12/2023

Experiments - Prompting Methods

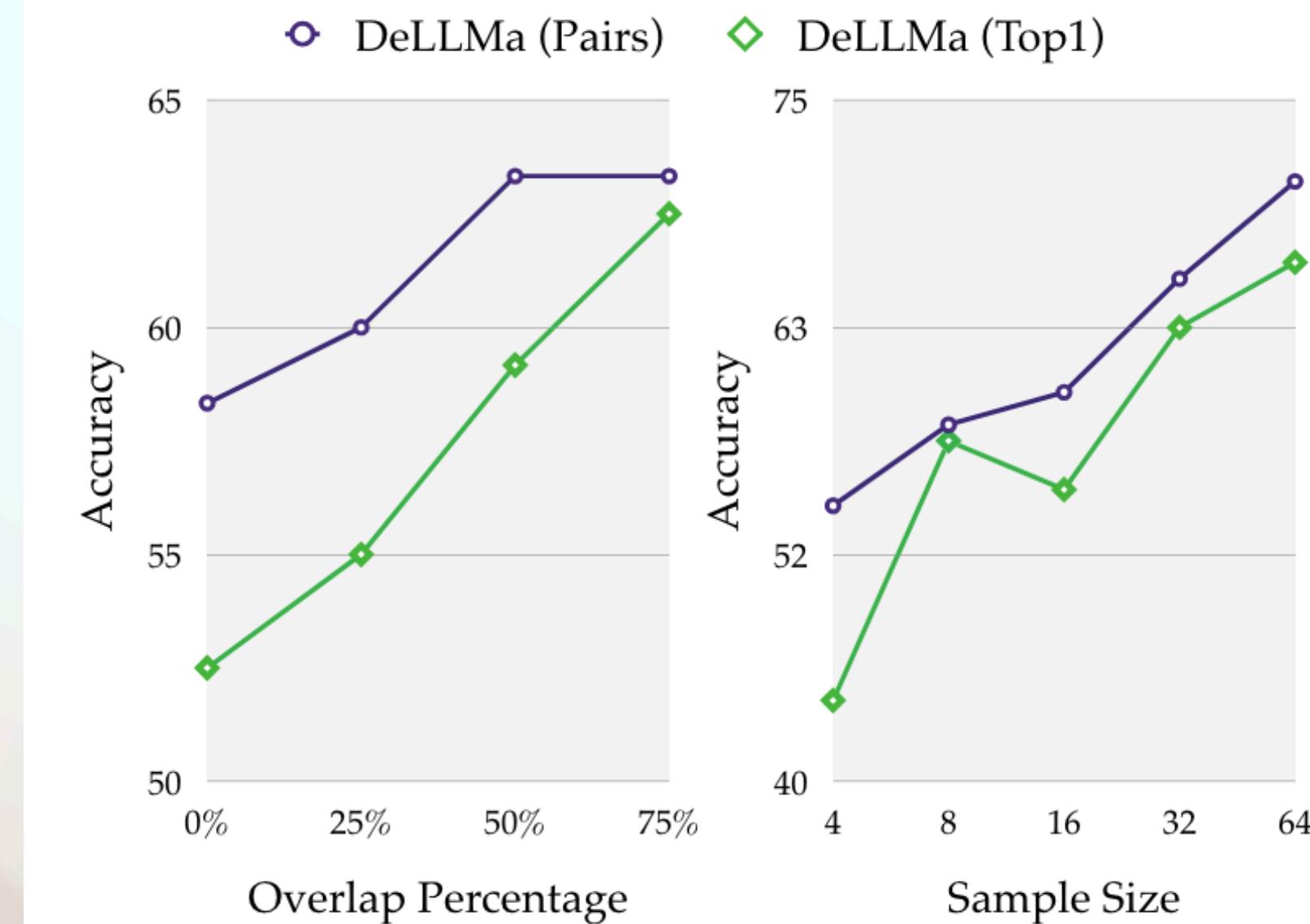
- **DeLLMa Variants:**
 - **DeLLMa-Pairs**: complete DeLLMa with all pairwise comparisons.
 - **DeLLMa-Top1**: complete DeLLMa that only prefers the top-ranked state-action tuple to other state action tuples.
 - **DeLLMa-Naive**: DeLLMa without batching or fixed state samples per action.
- **Baseline Methods:**
 - **Zero-Shot**: direct prompt to infer a^* given \mathcal{P} .
 - **Self-Consistency (SC)**: majority vote of 5 zero-shot predictions.
 - **Chain-of-Thought (CoT)**: multi-prompt chain that emulates DeLLMa prompts, *i.e.* first generating state latent factors, then inferring a^* with step-by-step reasoning on plausible outcomes.

Experimental Results



Observations

- DeLLMa significantly outperforms baseline prompting methods by up to **40%**.
- GPT-4 cannot reason with future scenarios without state samples (*i.e.* external help)!
- Ranking performance is **inconsistent** across domains: DeLLMa-Pairs performs better than Top1 on Agriculture, but worse on Finance.
- We conduct ablation studies and observe:
 - Large **overlap** and **sample size** improve performance.
 - ...but they require more API queries!

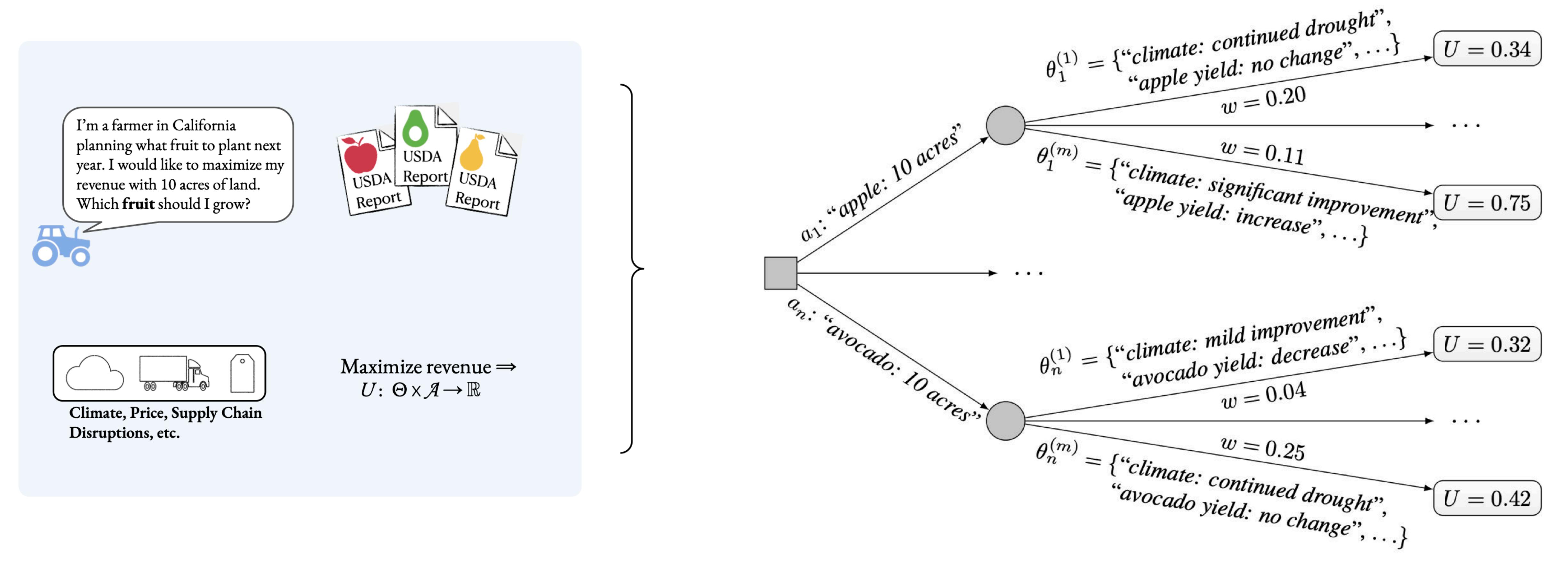


Left: We fix per-action sample size to be 16. Right: We fix overlap percentage to be 25%. All experiments performed on Agriculture data.

	Zero-Shot	CoT	SC	DeLLMa-Naive	DeLLMa-Pairs (16)	DeLLMa-Pairs (64)
API Calls	1	3	5	1	3	10
Word Counts	693.71	2724.2	3468.55	3681.94	7254.46	28895.46

Number of GPT-4 API calls and word counts per decision-making instance (with action set size 4 for the Agriculture dataset) across all methods. For DeLLMa-Naive, we set the total sample size to 50. For DeLLMa-Pairs, we fix the overlap percentage to 25% and vary the per action sample size from 16 to 64. DeLLMa-Top1 has the same statistics as DeLLMa-Pairs since they only differ in post processing.

DeLLMa is Human-Auditable!



Future Works

- Investigate the behaviors of each module:
 - More calibrated forecast with retrieval augmentation & tool usage.
 - Improve efficiency for utility elicitation.
- Expand datasets & application scenarios. Currently investigating airport scheduling, news recommendation, and medical decision making.
- Distill DeLLMa generations to elicit decision making capability on small LLMs.

My Amazing Co-Authors :-)



@olliezliu



@DeqingFu



@DaniYogatama



@willieneis

Thank you!



- Context [Market Overview& Product Summaries]

- Instruction

I would like to adopt a decision making under uncertainty framework to make my decision. The goal of you, the decision maker, is to choose an optimal action, while accounting for uncertainty in the unknown state. The first step of this procedure is for you to produce a belief distribution over the future state. The state is a vector of 16 elements, each of which is a random variable. The state variables are enumerated below:

- **climate condition**: the climate condition of the next agricultural season in California
- **supply chain disruptions**: the supply chain disruptions of the next agricultural season in California
- ...

You should format your response as a JSON object with 16 keys, wherein each key should be a state variable from the list above. Each key should map to a JSON object with 3 keys, each of which is a string that describes the value of the state variable. Together, these keys should enumerate the top 3 most likely values of the state variable. Each key should map to your belief verbalized in natural language. If the state variable is continuous (e.g. changes to a quantity), you should discretize it into 3 bins. You should strictly choose your belief from the following list: “very likely”, “likely”, “somewhat likely”, “somewhat unlikely”, “unlikely”, “very unlikely”. For example, if one of the state variable is “climate condition”, and the top 3 most likely values are “drought”, “heavy precipitation”, and “snowstorm”, then your response should be formatted as follows:

```
{  
    "climate condition":  
        {  
            "drought": "somewhat likely",  
            "heavy precipitation": "very likely",  
            "snowstorm": "unlikely"  
        },  
    ...  
}
```

- Context [Market Overview& Product Summaries]

- Instruction

I'm a farmer in California planning what fruit to plant next year. I would like to maximize my profit with '10' acres of land.

Below are the actions I can take: **Action 1.** avocado: 10 acres. **Action 2.** grape: 10 acres
I would like to adopt a decision making under uncertainty framework to make my decision. The goal of you, the decision maker, is to choose an optimal action, while accounting for uncertainty in the unknown state. Previously, you have already provided a forecast of future state variables relevant to planting decisions. The state is a vector of 16 elements, each of which is a random variable. The state variables (and their most probable values) are enumerated below:

- **climate condition**: {"continued drought": "very likely", "mild improvement": "somewhat likely", "significant improvement": "unlikely"}
...

Below, I have sampled a set of state-action pairs, wherein states are sampled from the state belief distribution you provided and actions are sampled uniformly from the action space. I would like to construct a utility function from your comparisons of state-action pairs

- **State-Action Pair 1.** State: climate condition: continued drought, supply chain disruptions: minor disruptions, avocado price change: no change, avocado yield change: increase, grape price change: increase, grape yield change: increase; Action 1. avocado: 10 acres
...

You should format your response as a JSON object. The JSON object should contain the following keys:

- **decision**: a string that describes the state-action pair you recommend the farmer to take. The output format should be the same as the format of the state-action pairs listed above, e.g. State-Action Pair 5.
- **rank**: a list of integers that ranks the state-action pairs in decreasing rank of preference. For example, if you think the first state-action pair is the most preferred, the second state-action pair is the second most preferred, and so on. For example, [1, 2, 3, 4, 5].
- **explanation**: a string that describes, in detail, the reasoning behind your decision. You should include information on the expected yield and price of each fruit, as well as factors that affect them

Related Works

- Jay Parsons, John Hewlett, Jeff Tranel, “Decision-making Under Uncertainty”, <https://cap.unl.edu/management/decision-making-under-uncertainty>
- Hallegatte, Stéphane, Ankur Shah, Casey Brown, Robert Lempert, and Stuart Gill. "Investment decision making under deep uncertainty-- application to climate change." World Bank Policy Research Working Paper 6193 (2012).
- Wickett, Eugene, Matthew Plumlee, Karen Smilowitz, Souly Phanouvong, and Victor Pribluda. "Inferring sources of substandard and falsified products in pharmaceutical supply chains." IISE Transactions 56, no. 3 (2024): 241-256.
- Lookman, Turab, Prasanna V. Balachandran, Dezhen Xue, and Ruihao Yuan. "Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design." npj Computational Materials 5, no. 1 (2019): 21.
- Von Neumann, John, and Oskar Morgenstern. "Theory of games and economic behavior: 60th anniversary commemorative edition." In Theory of games and economic behavior. Princeton university press, 2007.
- Luce, R. Duncan, and Howard Raiffa. Games and decisions: Introduction and critical survey. Courier Corporation, 1989.
- Berger, Zackary. "Navigating the unknown: shared decision-making in the face of uncertainty." Journal of general internal medicine 30 (2015): 675-678.
- Hao, Shibo, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. "Reasoning with language model is planning with world model." arXiv preprint arXiv:2305.14992 (2023).
- Halawi, Danny, Fred Zhang, Chen Yueh-Han, and Jacob Steinhardt. "Approaching Human-Level Forecasting with Language Models." arXiv preprint arXiv:2402.18563 (2024).