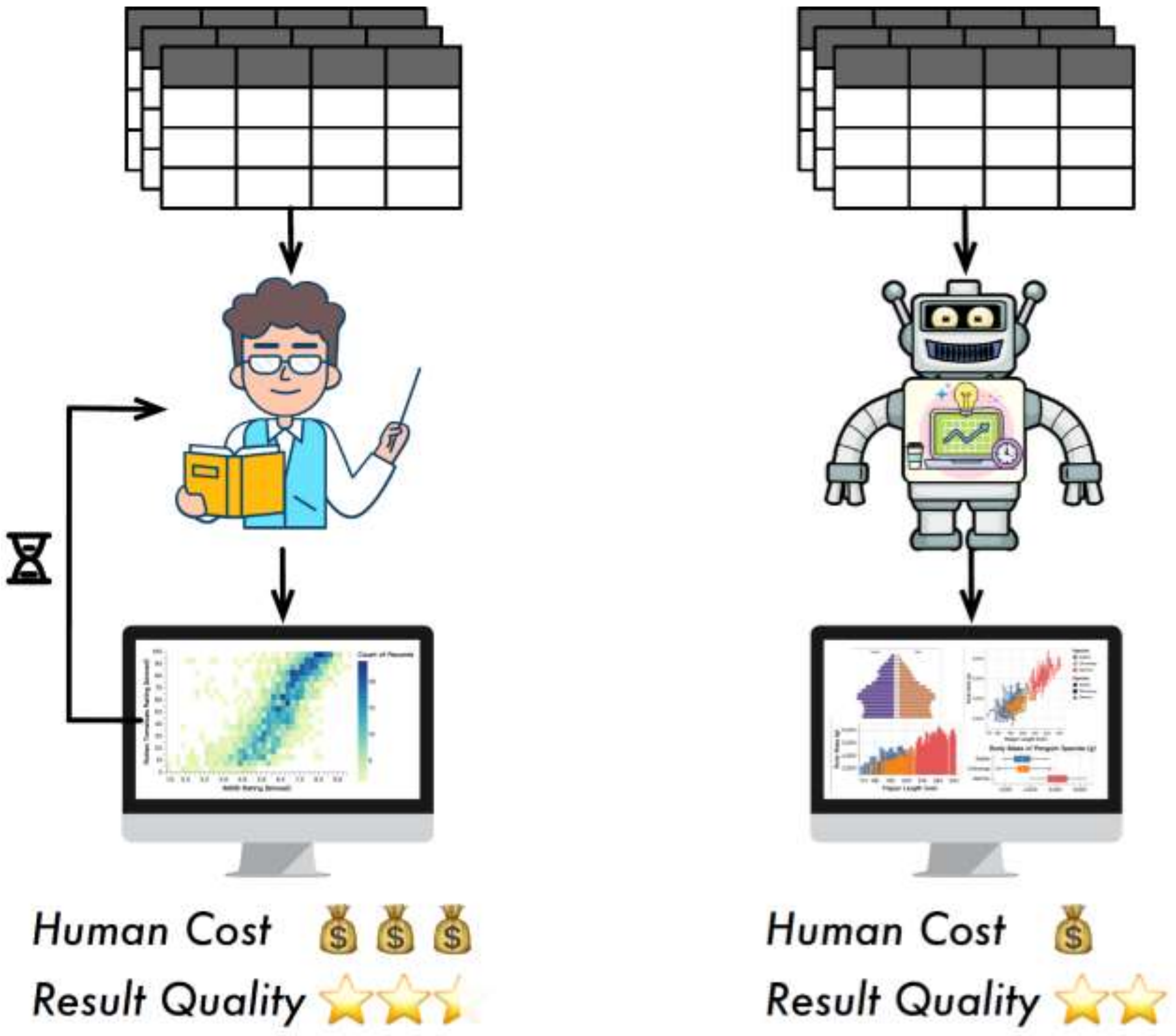
VLDB 2024 - 50th International Conference on Very Large Data Bases

HAIChart: Human and AI Paired Visualization System

Yupeng Xie[1], Yuyu Luo[1,2*], Guoliang Li[3], Nan Tang[1,2]
[1]HKUST (GZ), [2]HKUST, [3]Tsinghua University

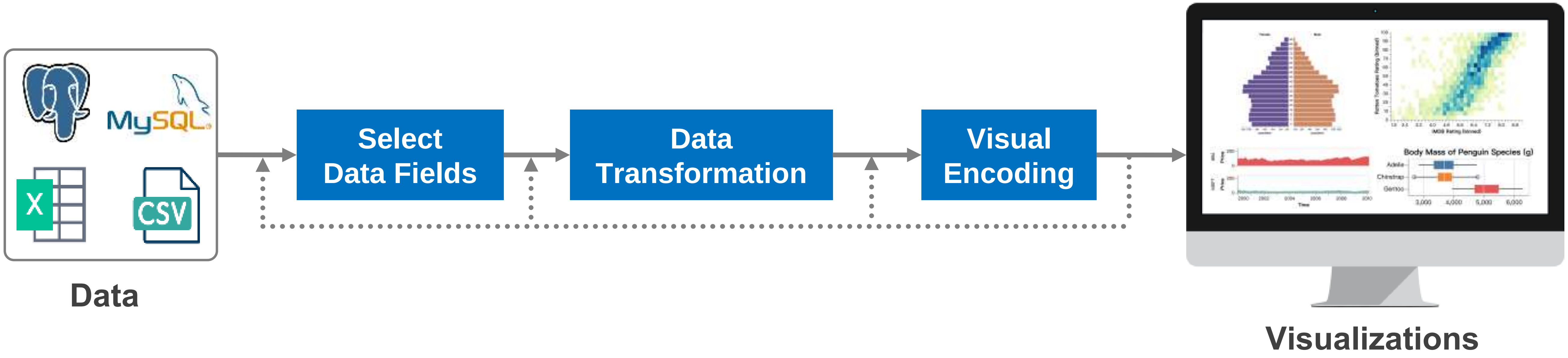
Introduction

- Existing data visualization tools fall into two main categories:



Human-powered VIS **Fully Automatic VIS**

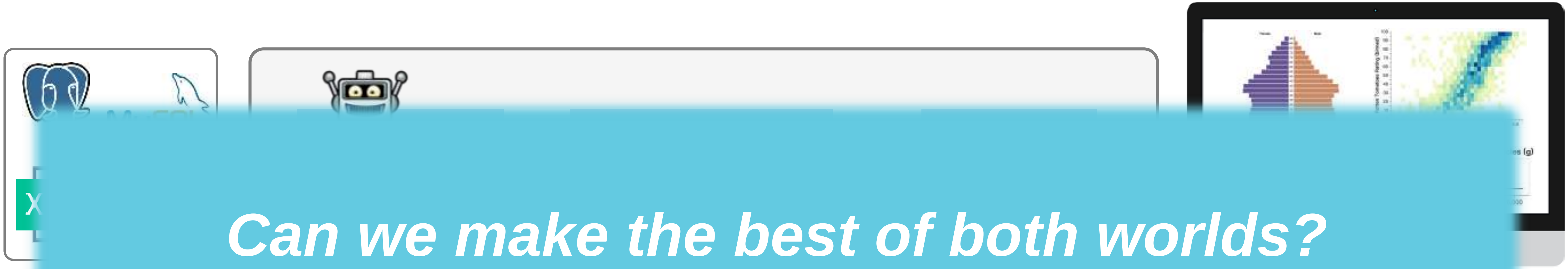
Human-powered Visualization



Manually specify through **Code** **or** **Clicks**

- ***Require*** *human and* ***domain expertise***
- *Tedious and* ***time-consuming*** *(even for experts)*

Fully Automatic Visualization

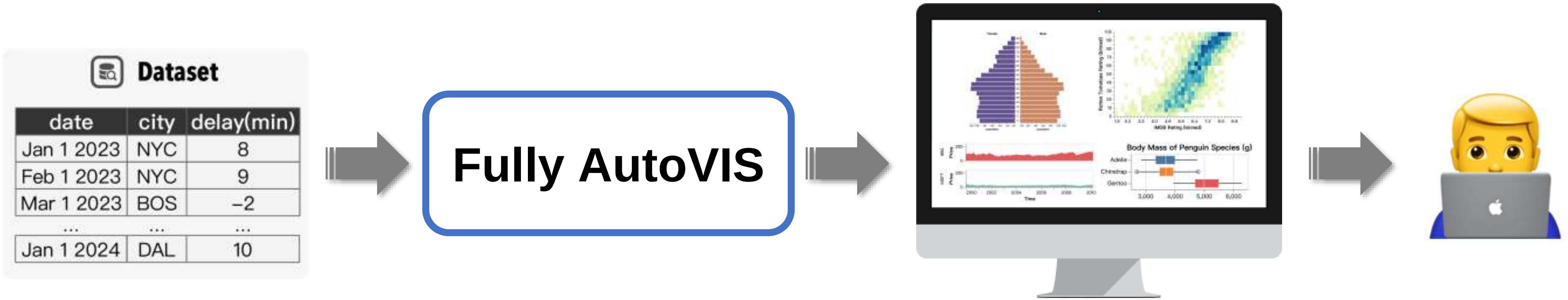


Can we make the best of both worlds?

- Cannot capture **user intent or feedback**.
- Fails to meet specific **user needs**.

HAIChart: Human and AI Paired Visualization System

Phase 1: Recommend **high-quality** visualizations **to minimize manual effort.**



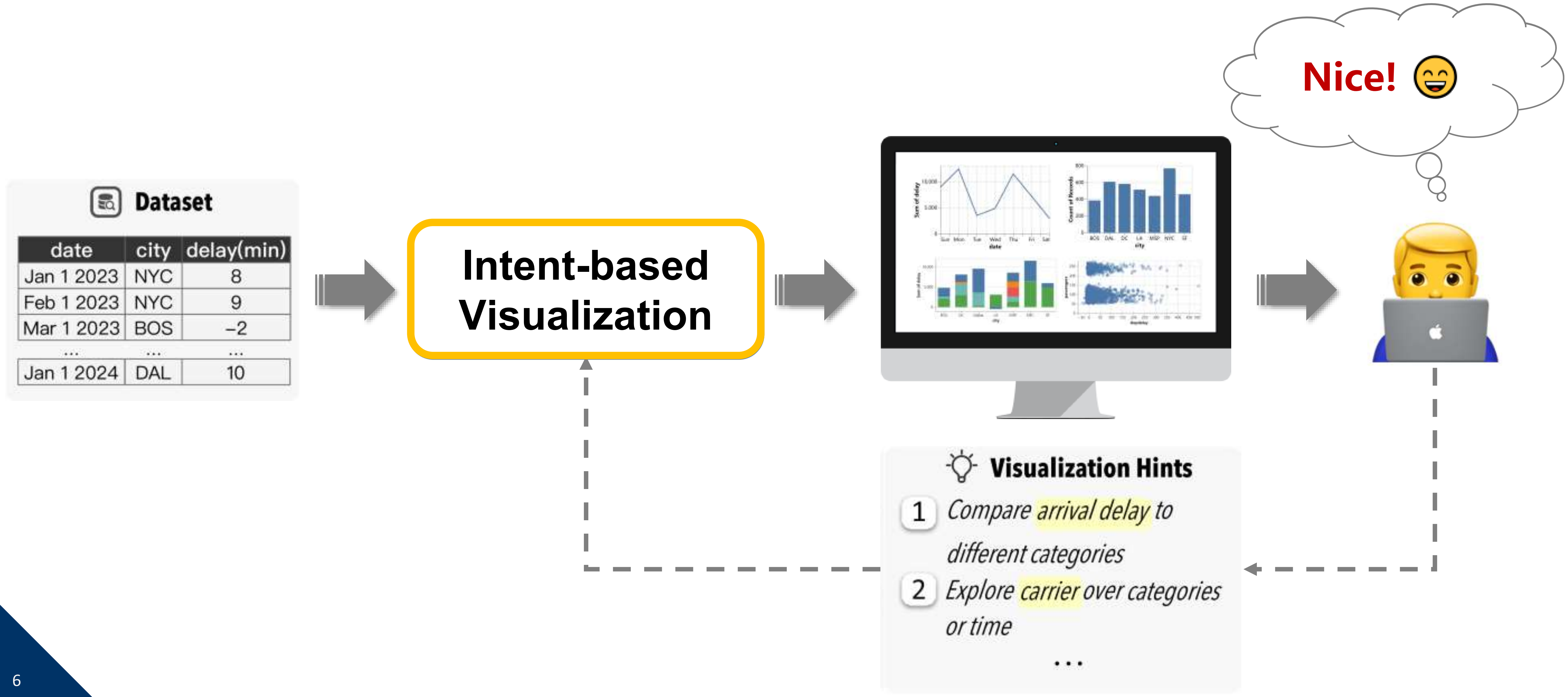
date	city	delay(min)
Jan 1 2023	NYC	8
Feb 1 2023	NYC	9
Mar 1 2023	BOS	−2
...	...	...
Jan 1 2024	DAL	10

HAIChart: Human and AI Paired Visualization System

香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

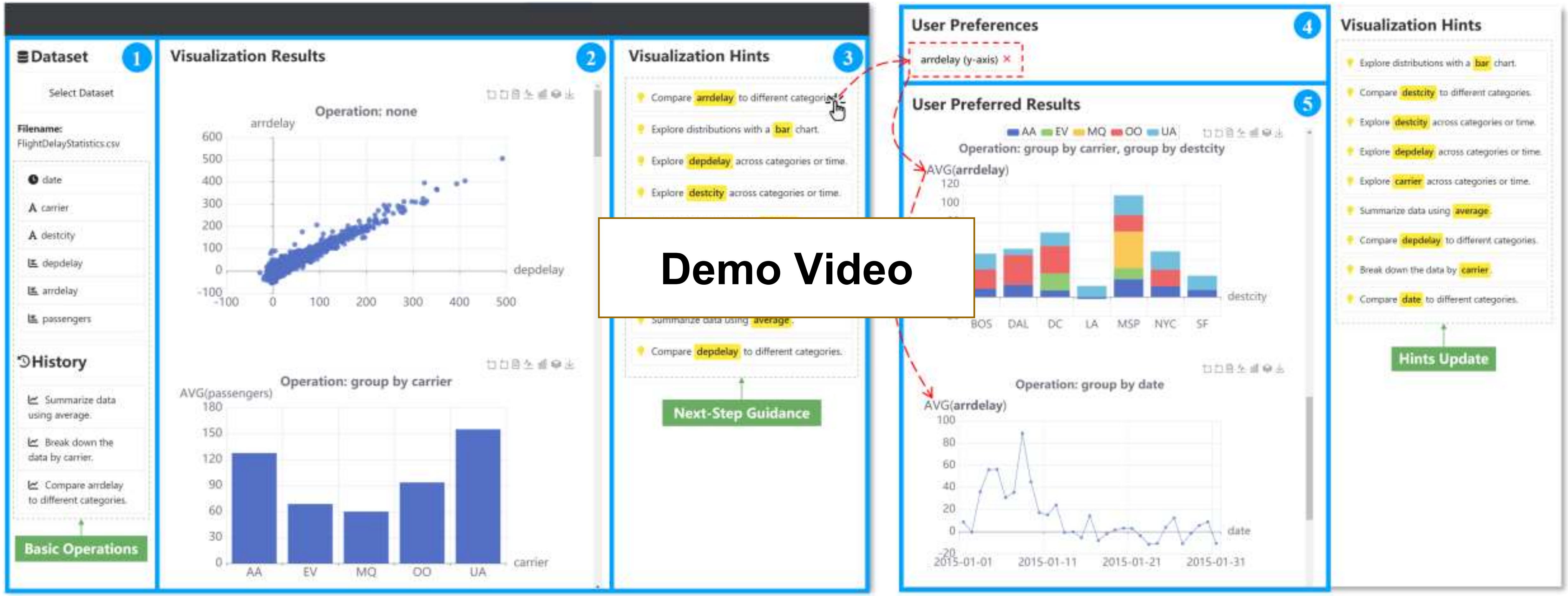
Phase 1: Recommend **high-quality** visualizations **to minimize manual effort.**

Phase 2: **Refine** visualizations **with hints** to more closely **align with user needs**.

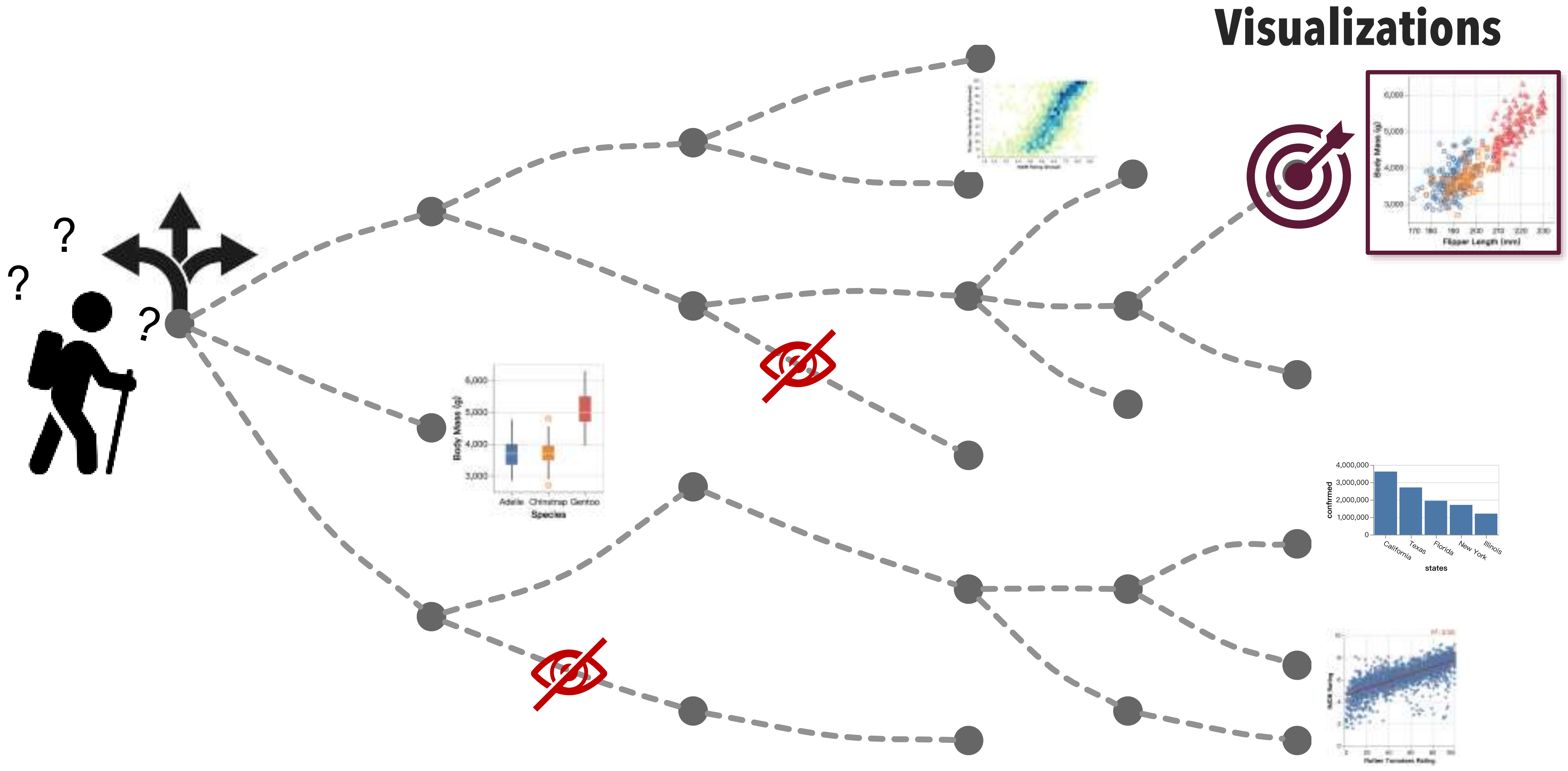


Demonstration Scenarios

- First-Round Visualization Recommendations.
- Multi-Round Visualization Recommendation based on Hints.



Challenge 1: How to Explore Search Space Efficiently?



Visualization Query

We use **visualization queries** to represent all possible visualizations.

Each query is a sequence of operations such as visual encoding and data transformation.

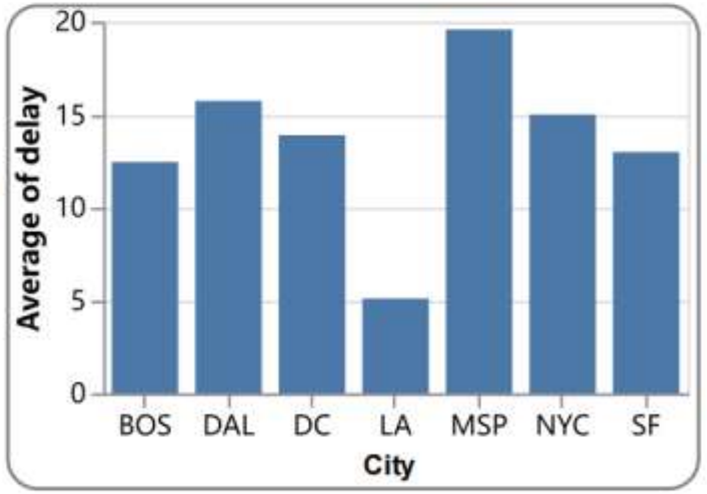
Flight Delay Dataset

Date	City	Delay(min)	...
Jan 1 2023	NYC	8	...
Feb 1 2023	NYC	9	...
Mar 1 2023	BOS	-2	...
...	...	...	...

mark Bar
encoding x City y AVG(Delay)
transform group City

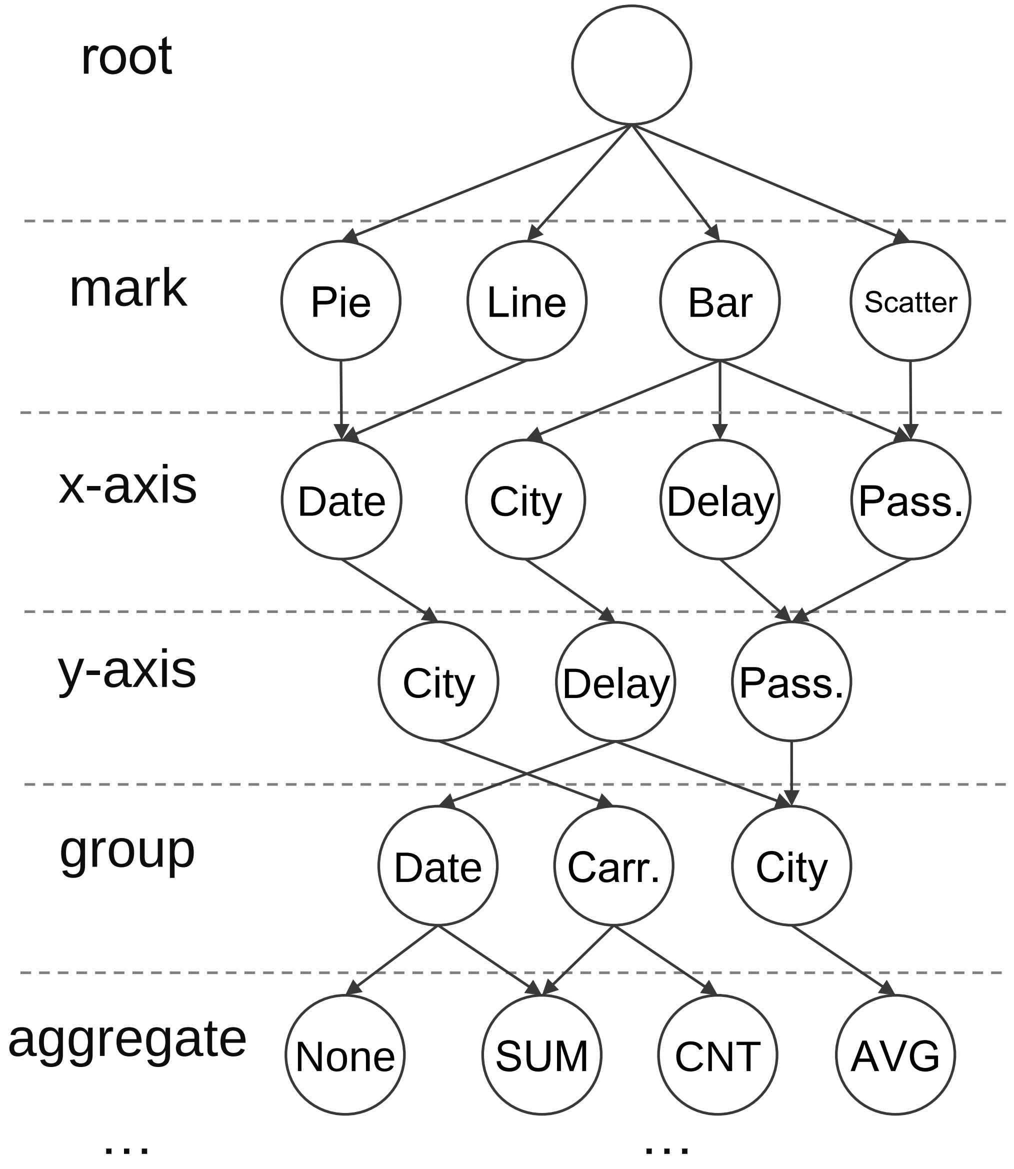
Query

Visualization



Visualization Query Graph

We also introduce the **visualization query graph** to explore various visualizations.

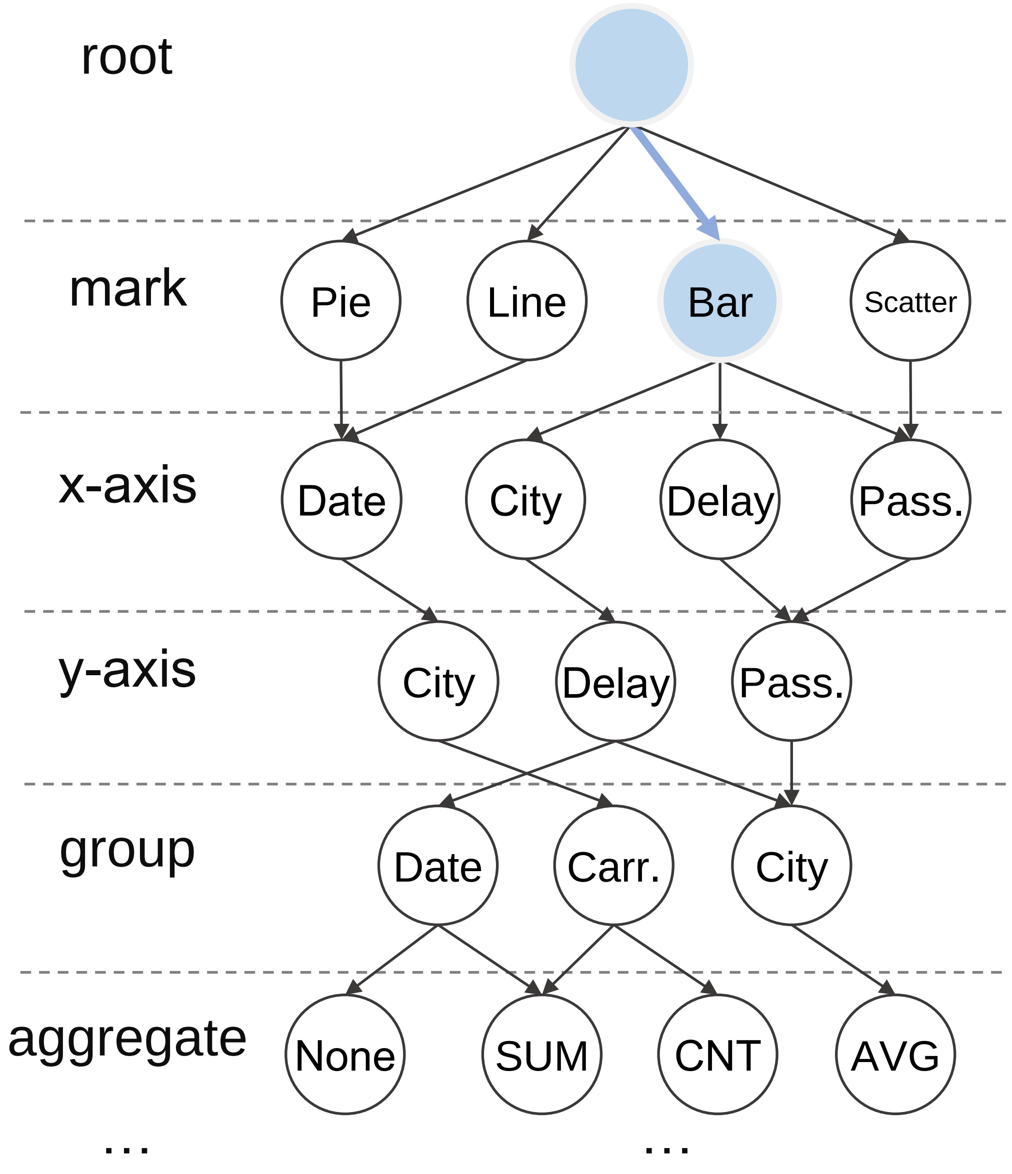


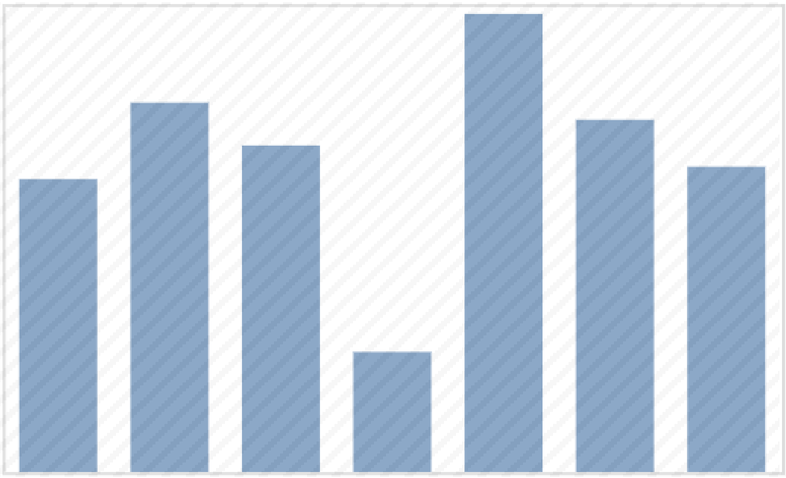
- Each **node** represents a visualization **operation**.
- A **path** is a sequence of these operations, representing a **query**.

Visualization Query Graph

香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

For example, start from the root node. First, select the chart type as a bar chart.

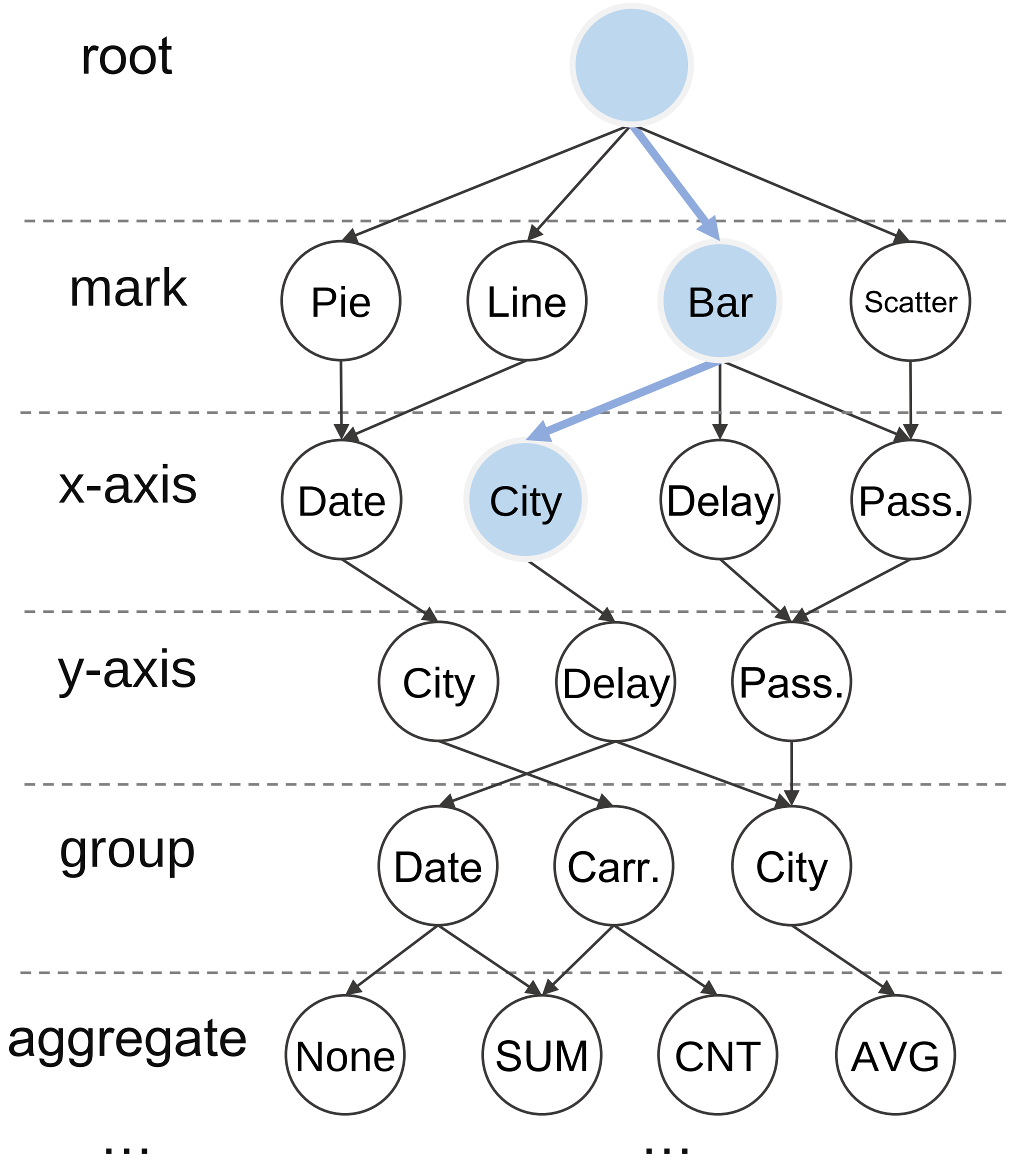


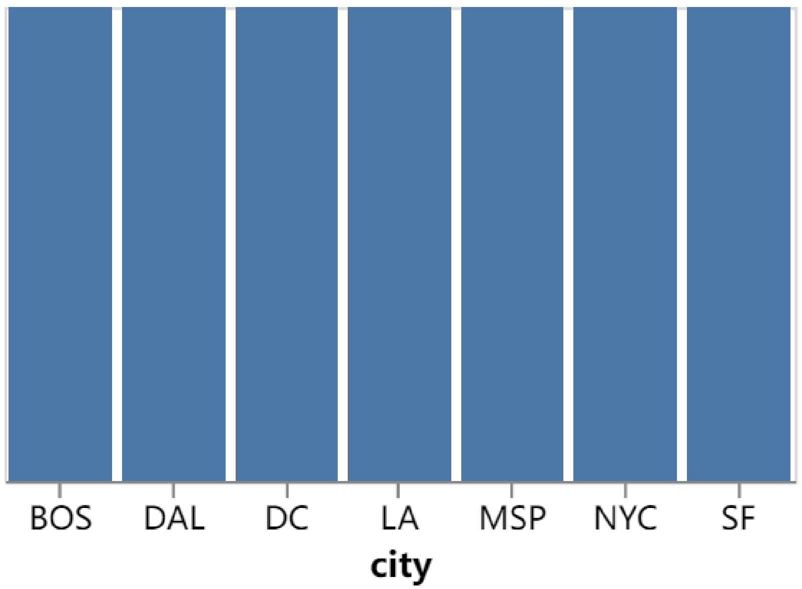


Current Visualization

Visualization Query Graph

Next, set City as the X-axis and Delay as the Y-axis.

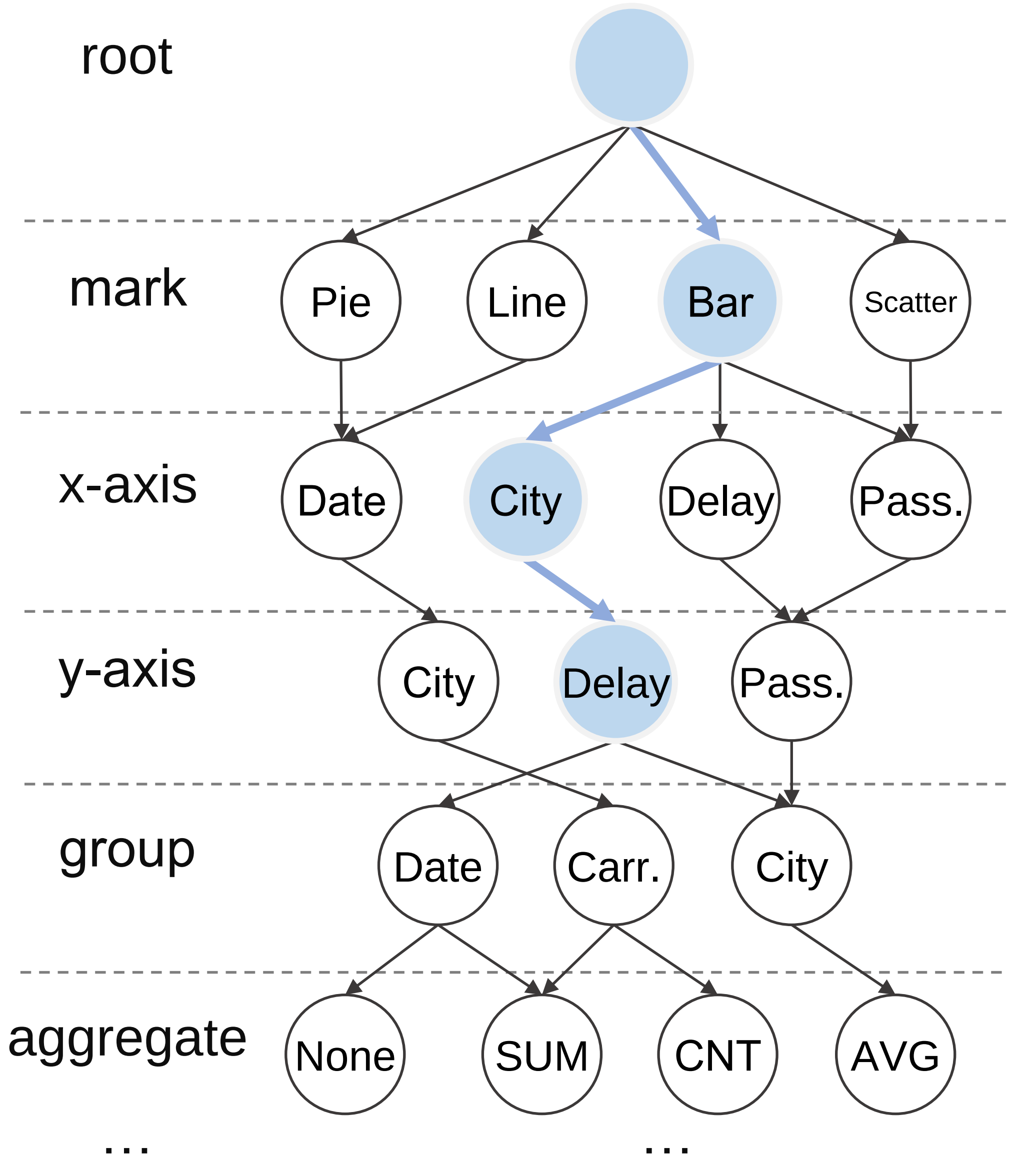


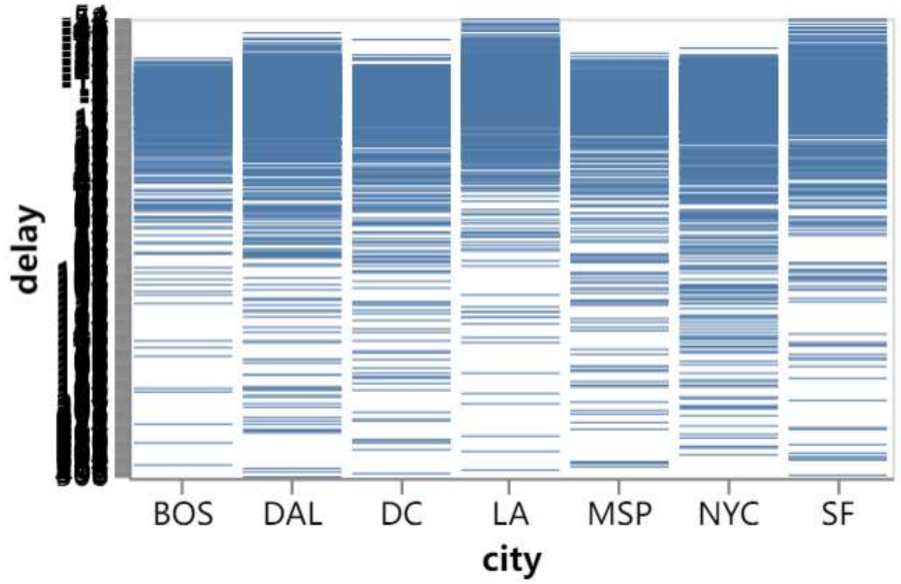


Current Visualization

Visualization Query Graph

Next, set City as the X-axis and Delay as the Y-axis.

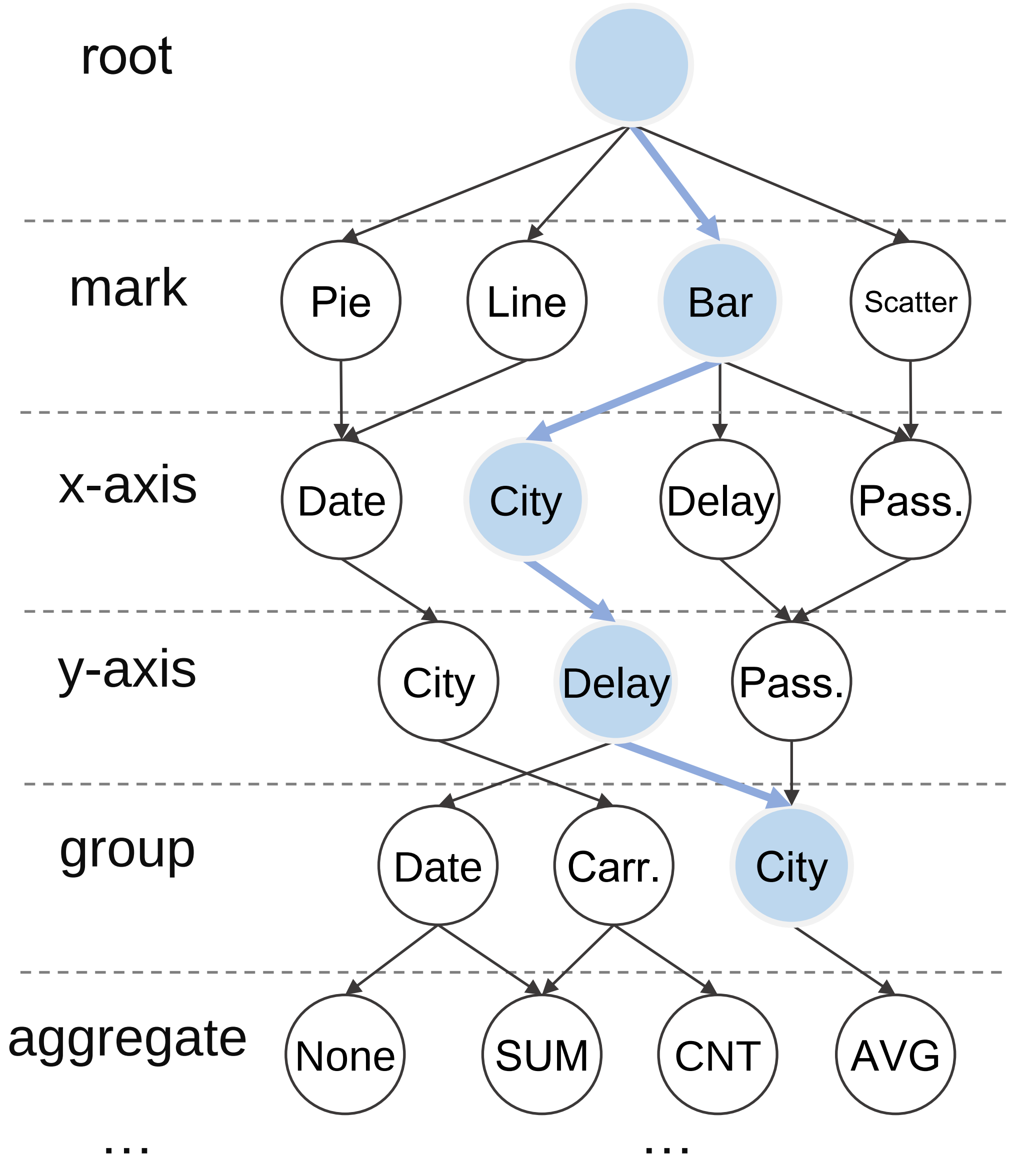


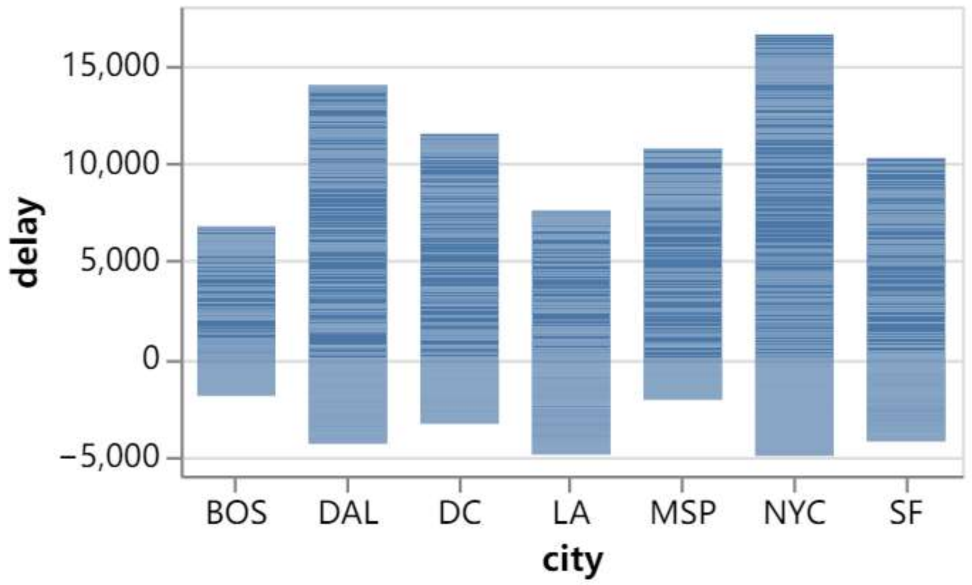


Current Visualization

Visualization Query Graph

Then, apply a data transformation by grouping the data by City.

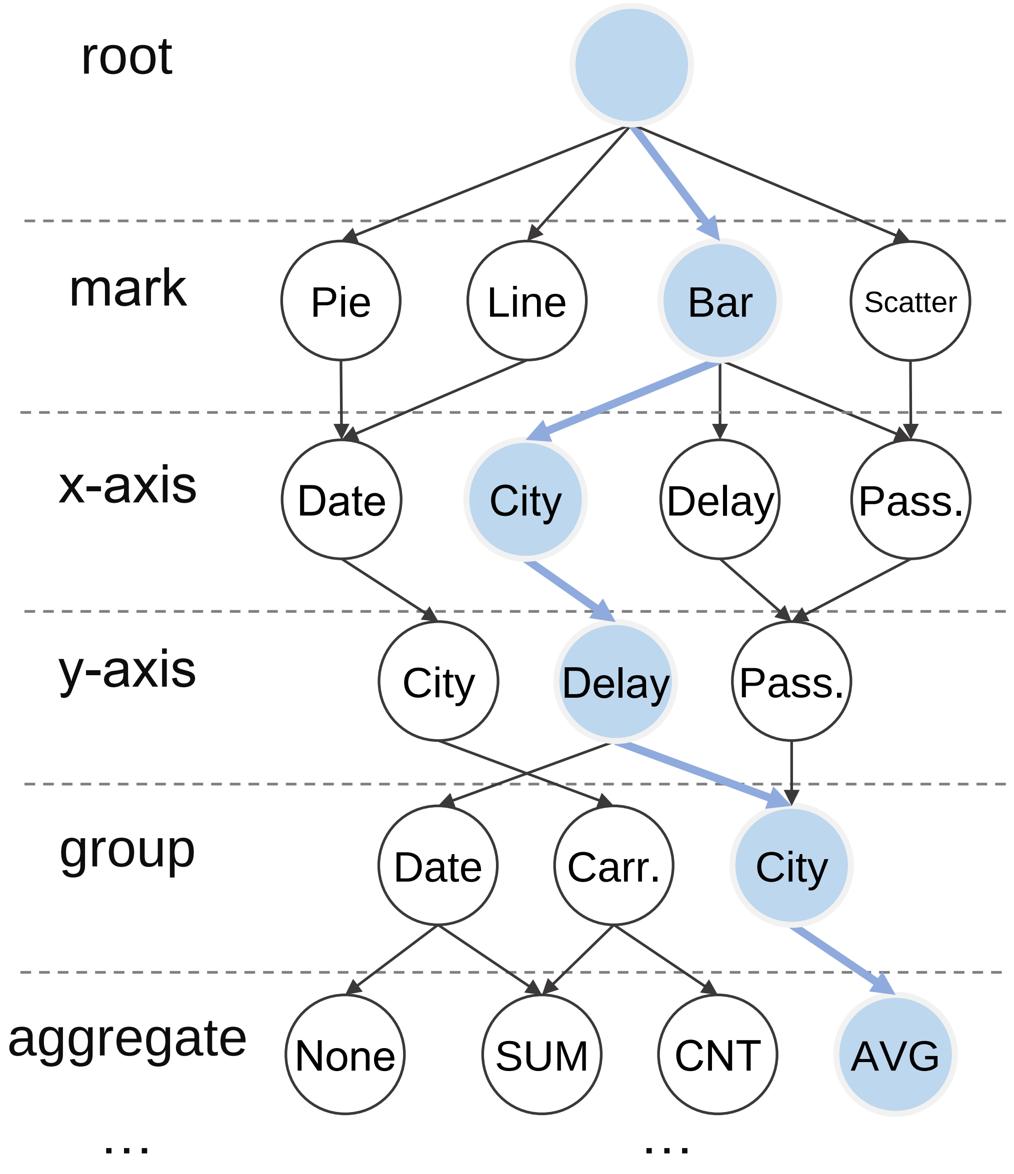


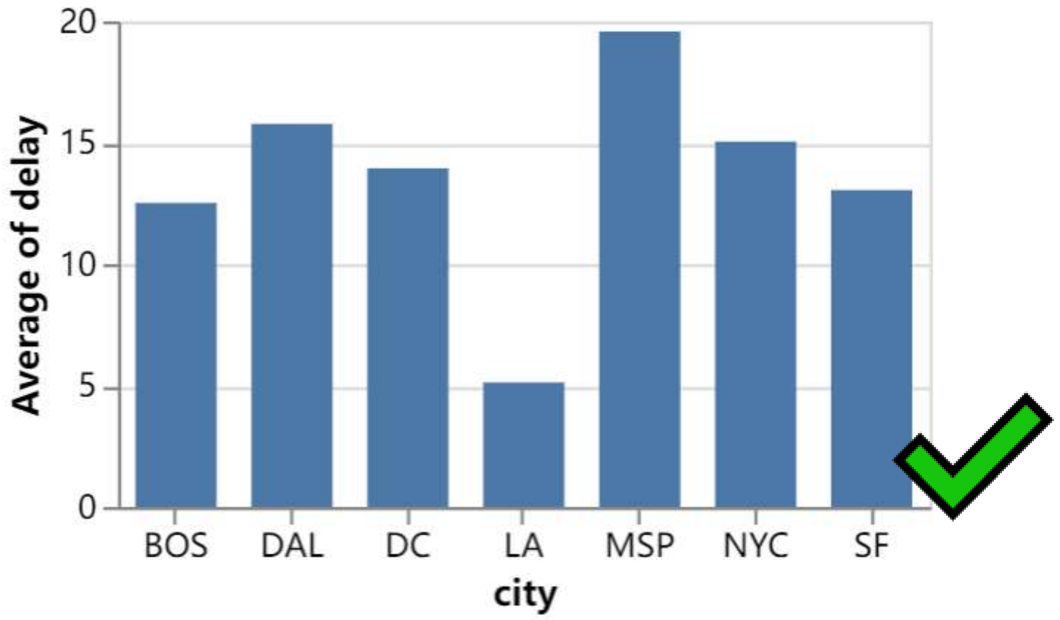


Current Visualization

Visualization Query Graph

Finally, select the average as the aggregation function.



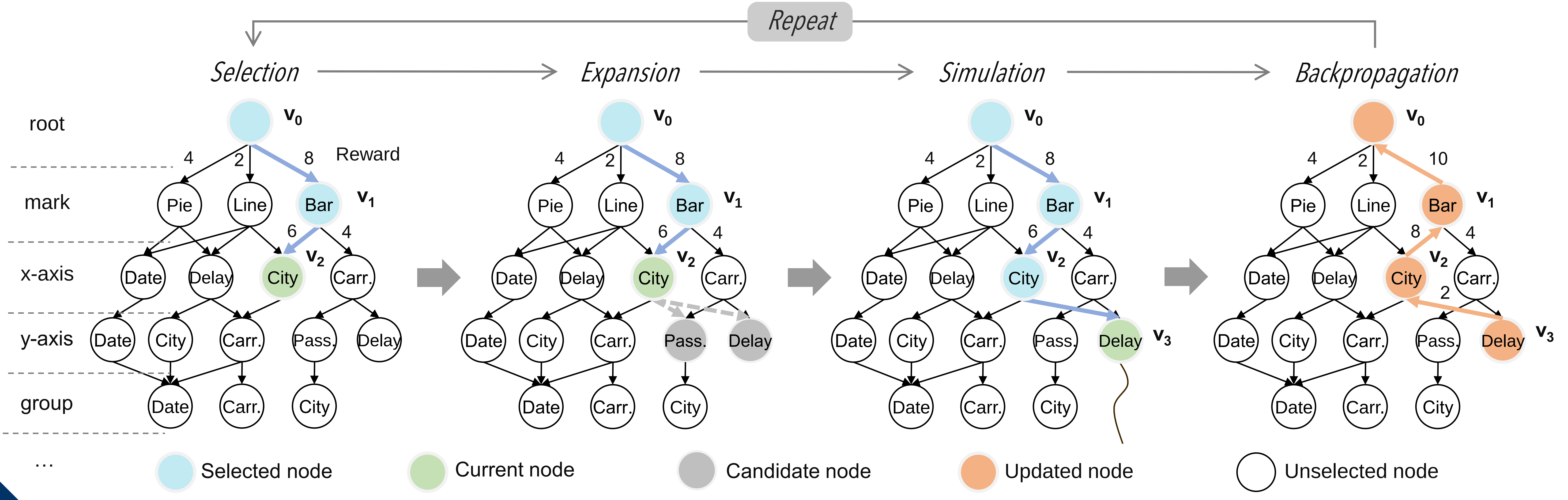


Current Visualization

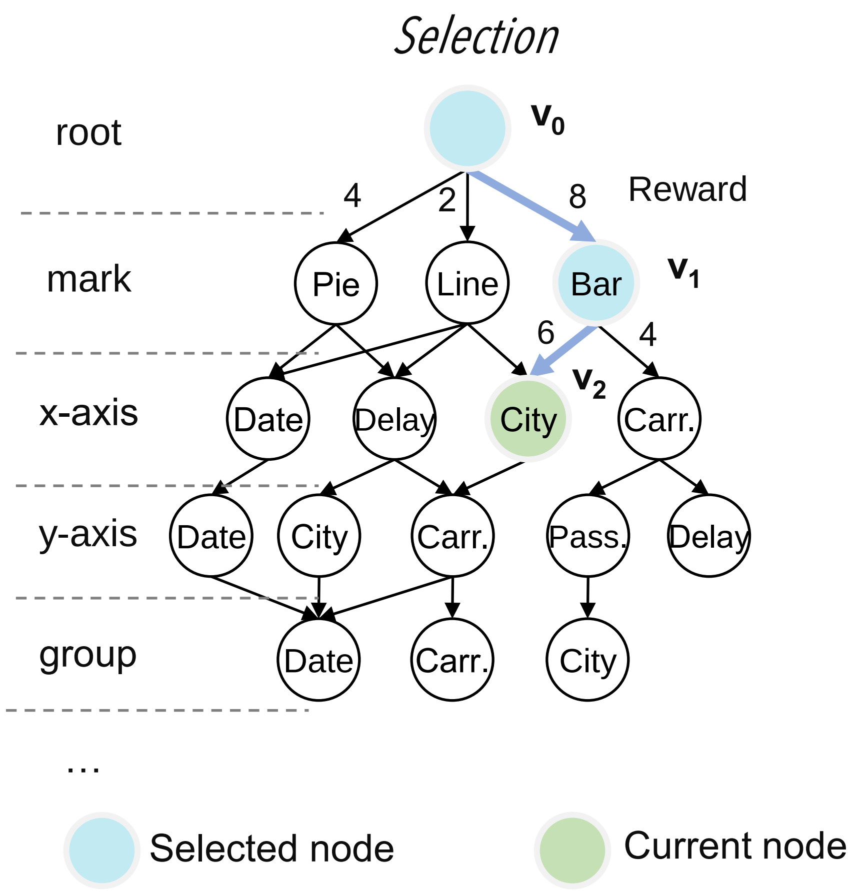
This sequence results in a valid visualization.

Monte Carlo Graph Search-based Visualization Generation

The figure shows the MCGS algorithm's four steps.



MCGS-based Visualization Generation

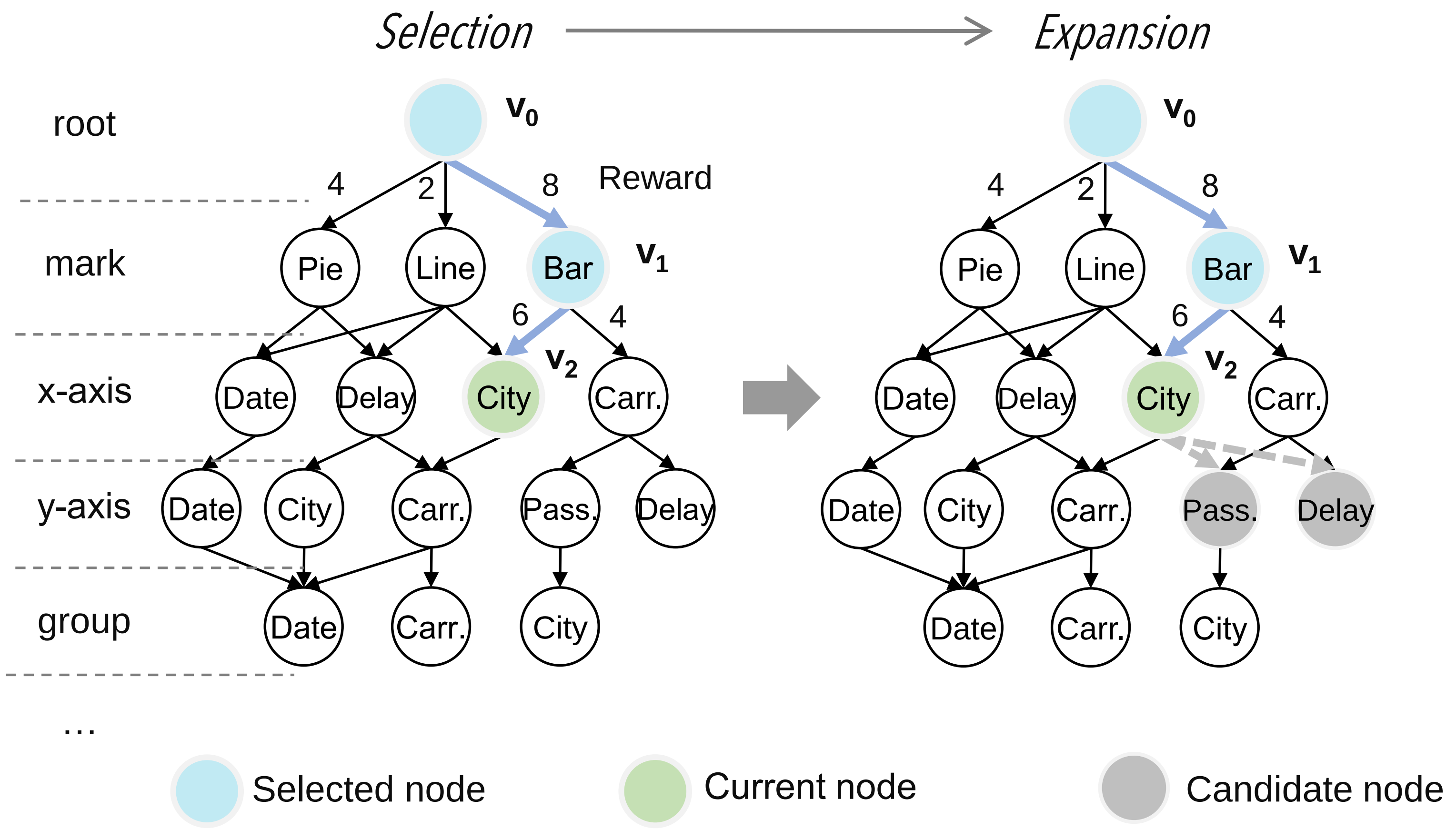


In the selection phase, the algorithm uses the Upper Confidence Bound (UCB) to recursively select optimal child nodes until reaching an unexpanded node.

$$UCB = \underbrace{\bar{X}_i}_{exploitation} + \underbrace{c\sqrt{2\ln n/n_i}}_{exploration}$$

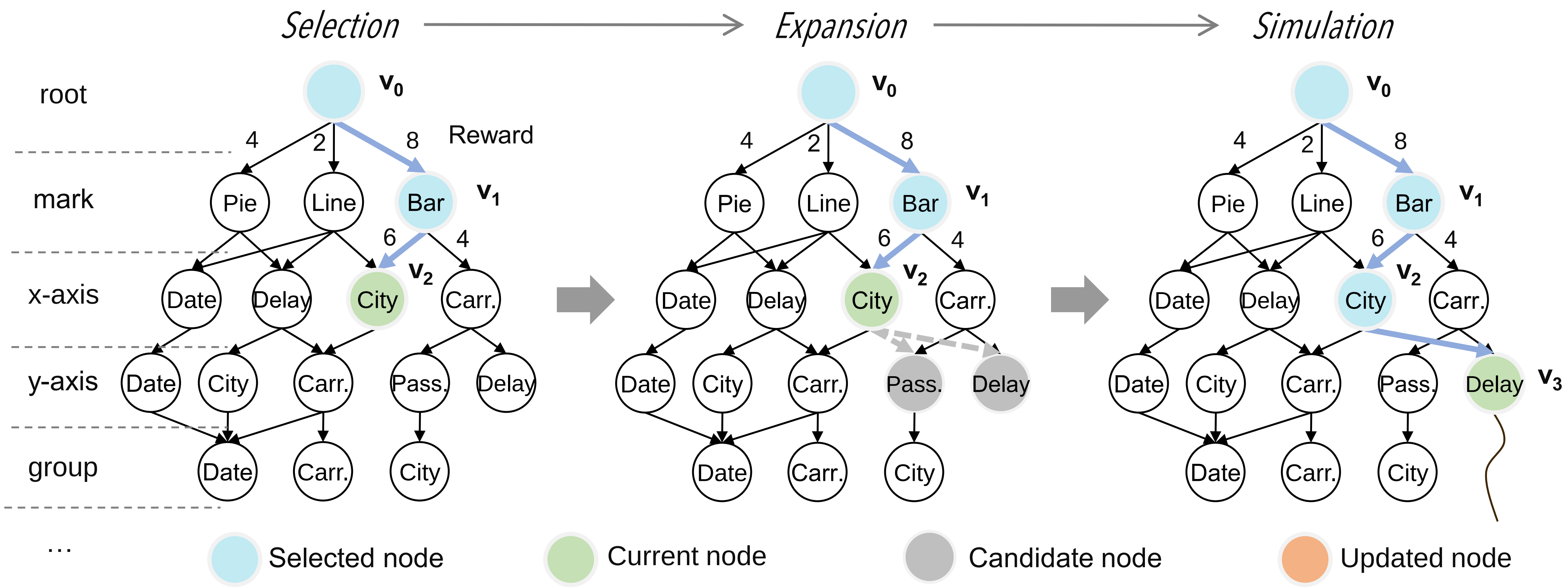
Specifically, the UCB algorithm balances exploration and exploitation during the search process.

MCGS-based Visualization Generation



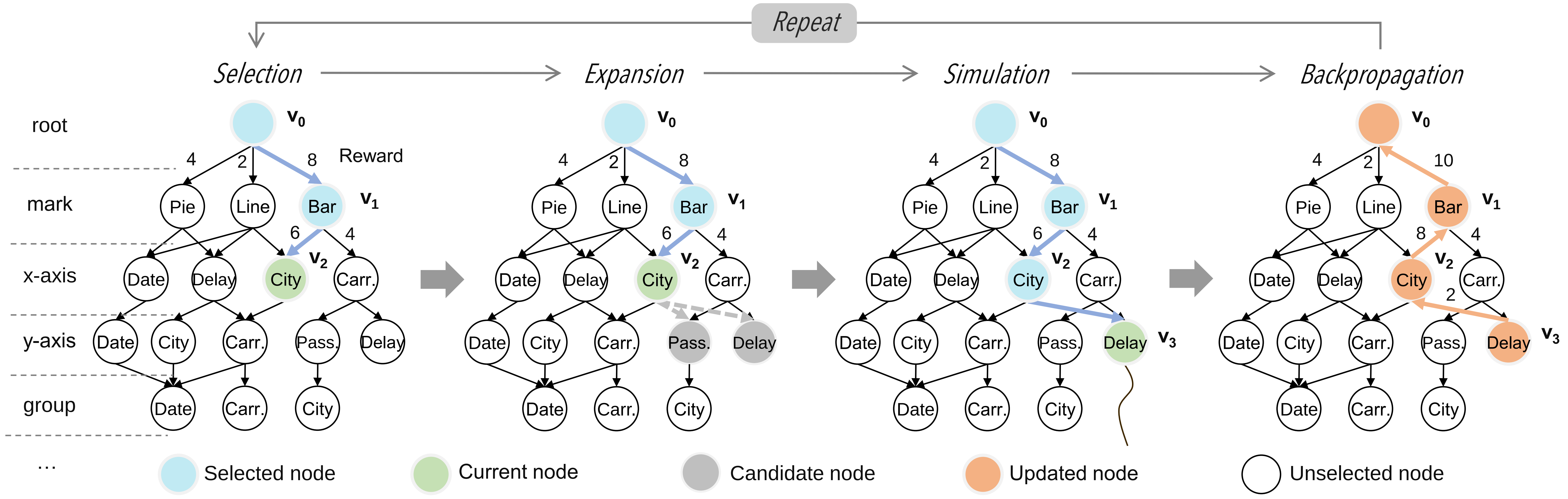
In the expansion phase, the algorithm selects the next valid action by removing low-quality visualizations that are either syntactically incorrect or violate visualization rules.

MCGS-based Visualization Generation



In the simulation phase, the algorithm randomly explores based on the current query until a valid query is found, then assigns a reward using the reward function.

MCGS-based Visualization Generation

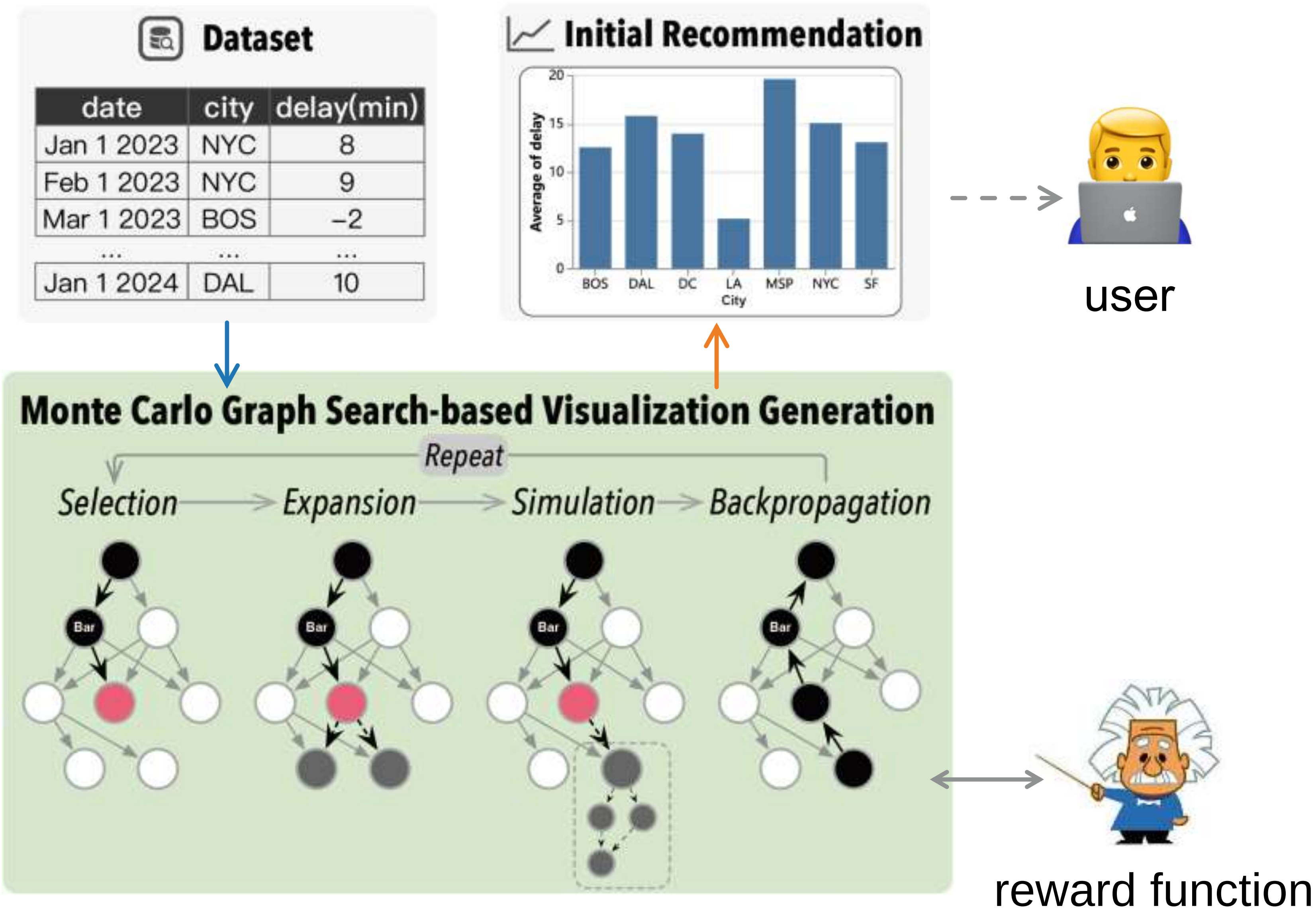


In the backpropagation phase, the reward values from the simulation are used to update the graph, guiding future searches more effectively.

These steps repeat until the maximum iterations are reached.

MCGS-based Visualization Generation

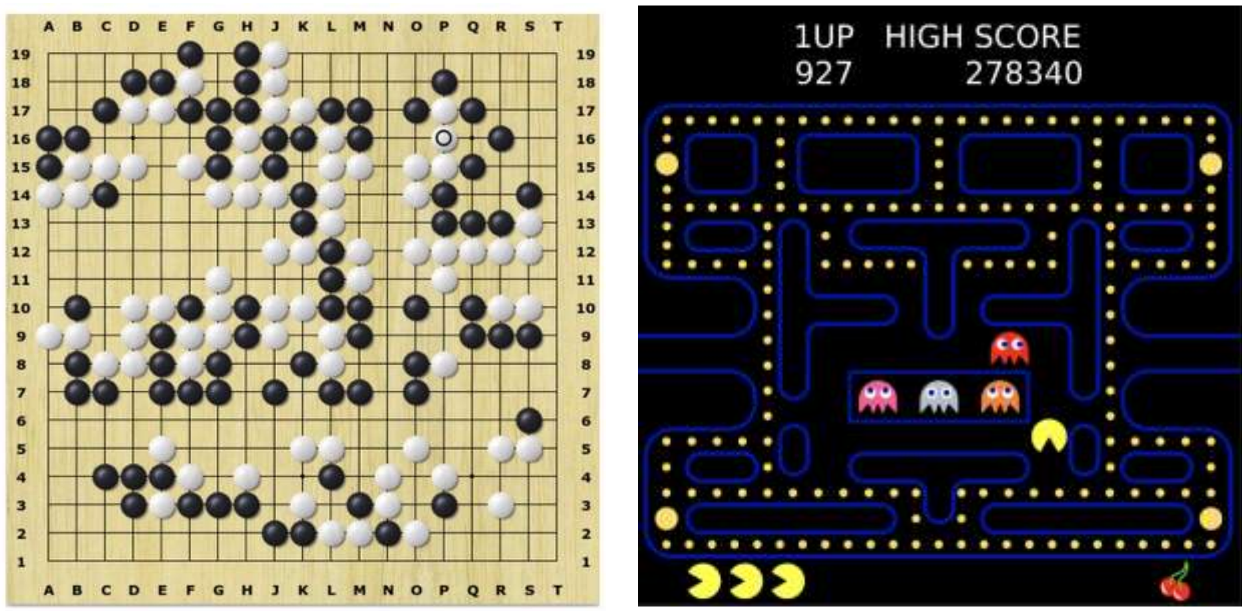
In Monte Carlo Graph Search, the reward function plays a key role in guiding the search process.



Challenge 2: How to Evaluate Visualization Quality?

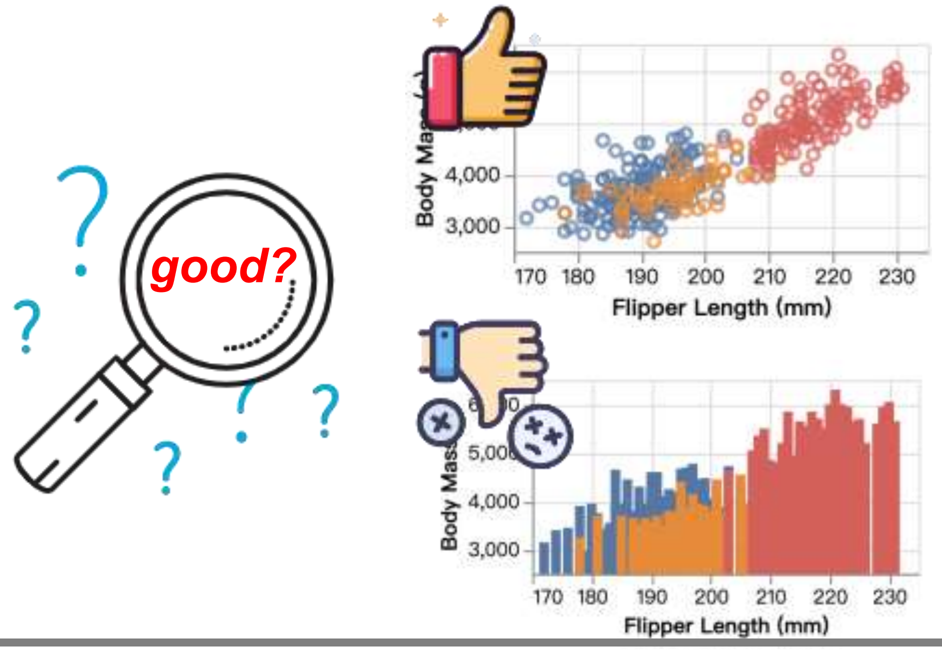
Unlike games such as Go, which have clear rules, visualization evaluation lacks well-defined reward criteria and can be biased by a single metric.

Games have clear rules



Games

How to quantify the goodness?

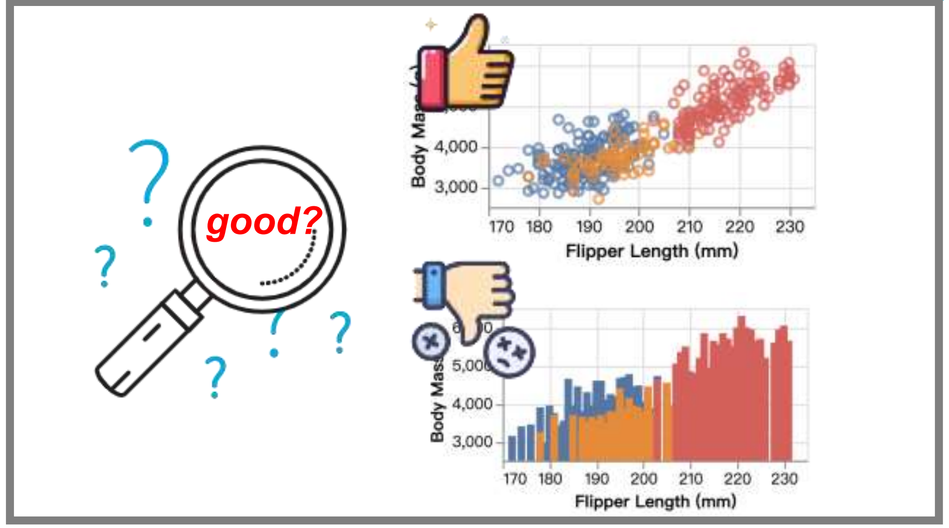


Visualization

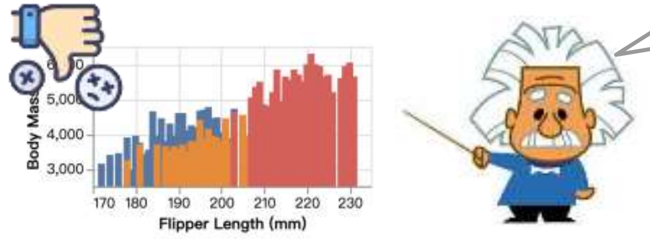
Source: https://x.com/karpathy/status/1821277264996352246
https://medium.com/swlh/a-brief-history-of-reinforcement-learning-in-game-play-d0861b2b74ef

Composite Reward Function

How to quantify the goodness?

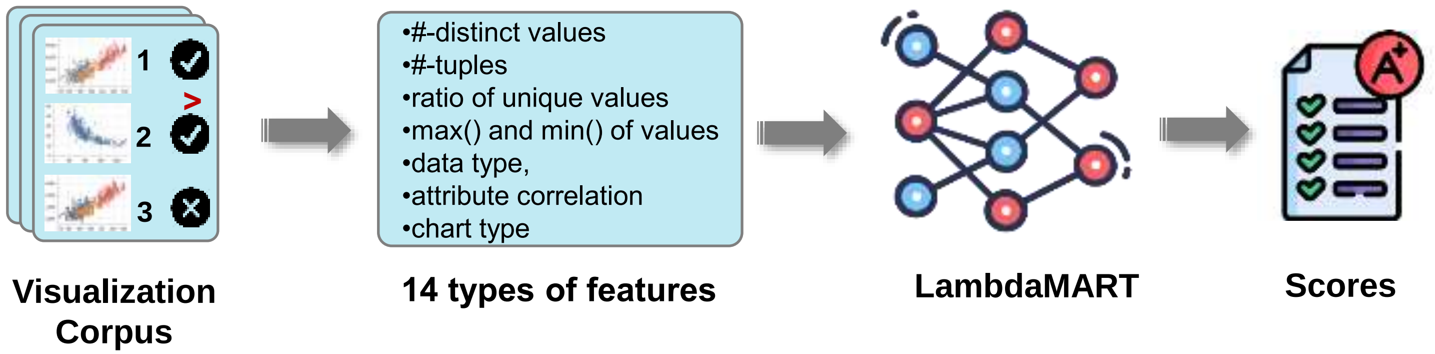


Learning from Domain Knowledge

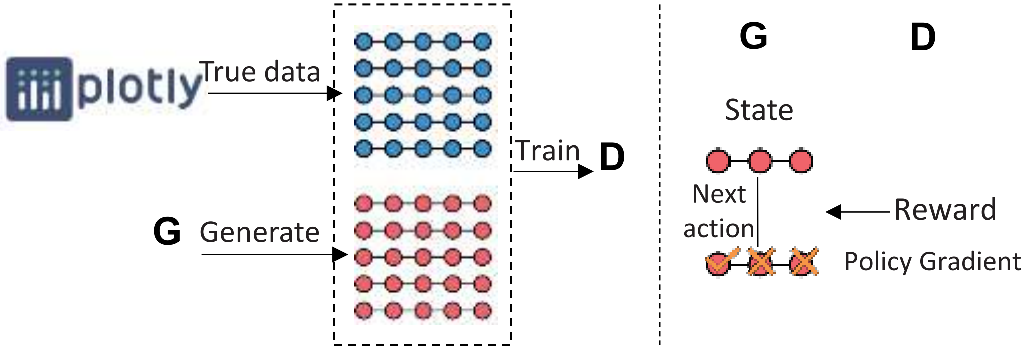


a bar chart with more than 50 bars is hard to get insights.

Capturing Data Features



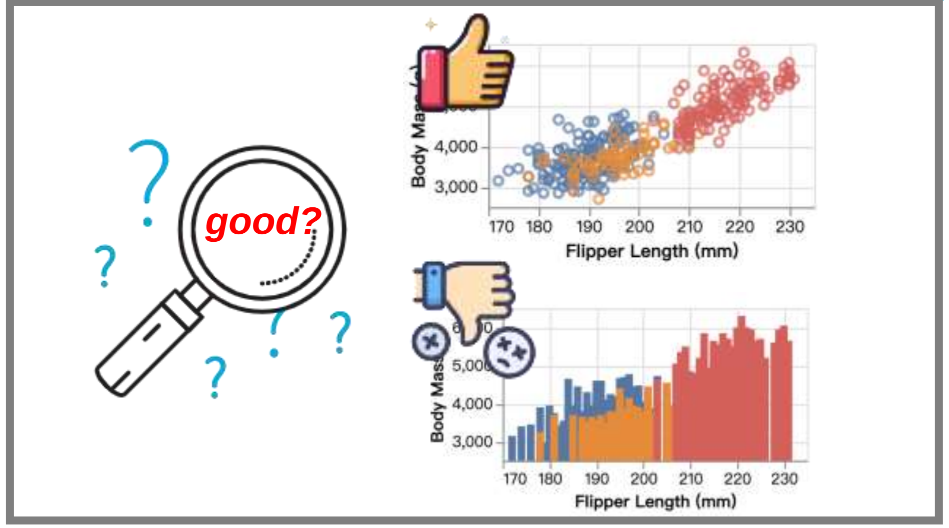
Learning Common User Preferences



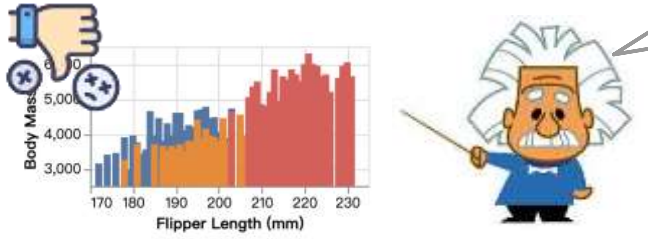
Composite Reward Function

We remove low-quality visualization results based on expert rules.

How to quantify the goodness?

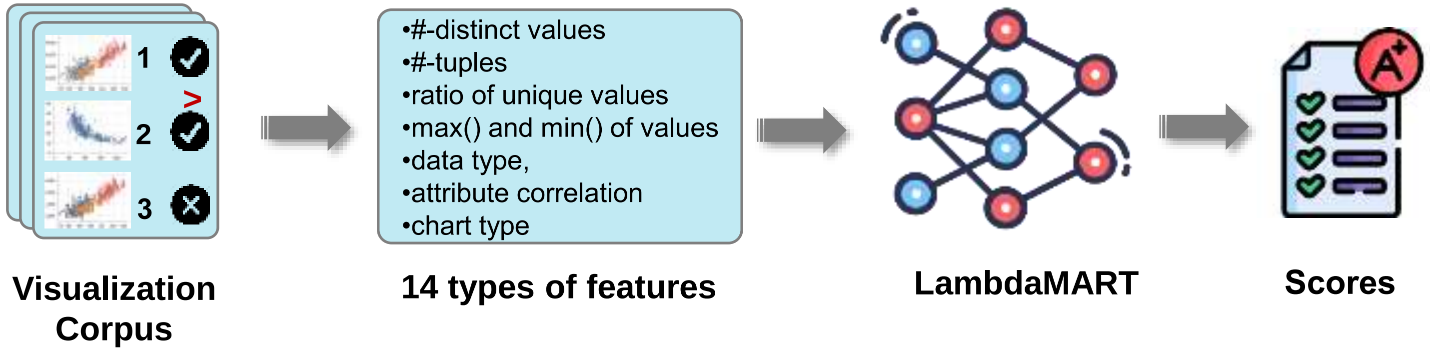


Learning from Domain Knowledge

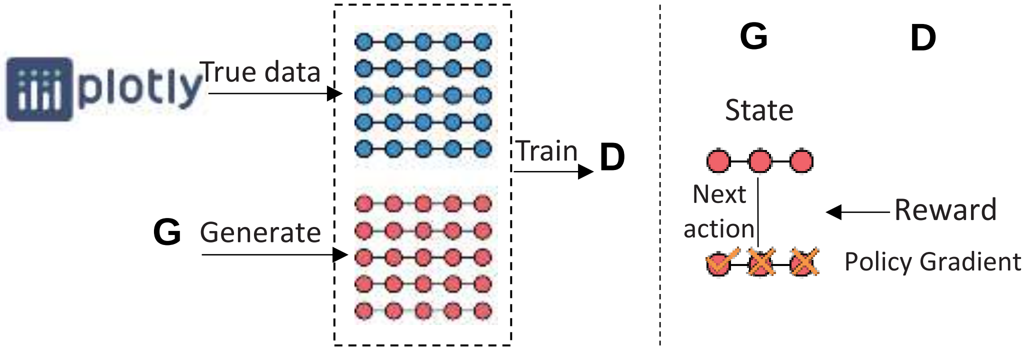


a bar chart with more than 50 bars is hard to get insights.

Capturing Data Features

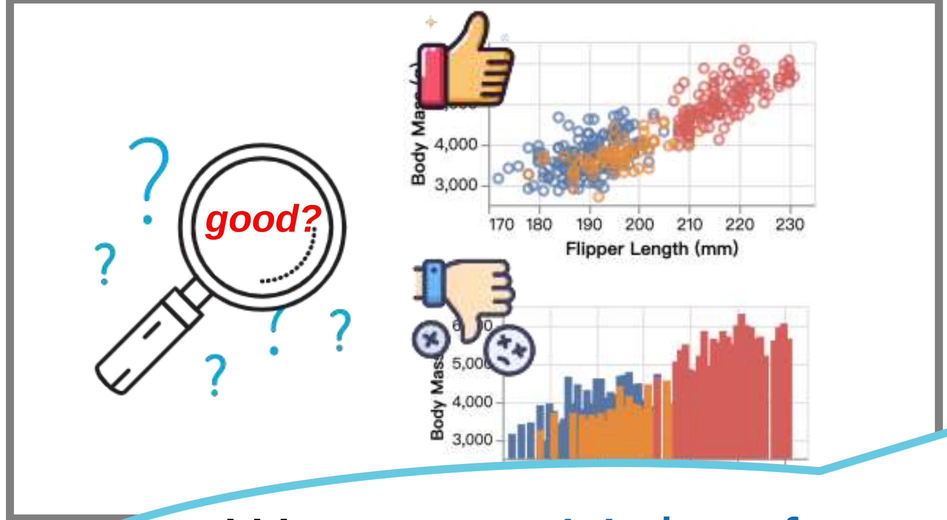


Learning Common User Preferences



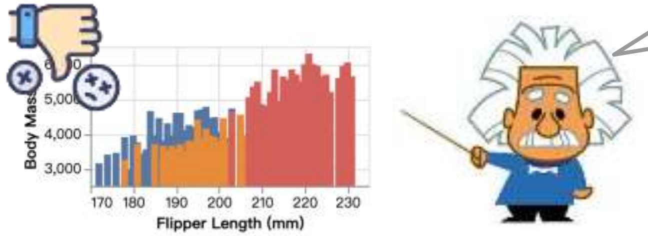
Composite Reward Function

How to quantify the goodness?



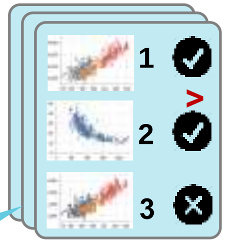
We extract 14 data features and use LambdaMART to evaluate the visualizations.

Learning from Domain Knowledge



a bar chart with more than 50 bars is hard to get insights.

Capturing Data Features



Visualization Corpus

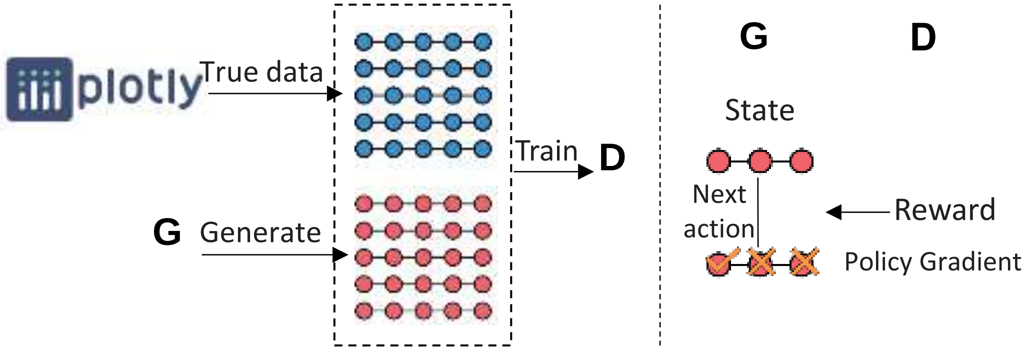
- #-distinct values
- #-tuples
- ratio of unique values
- max() and min() of values
- data type,
- attribute correlation
- chart type

14 types of features

LambdaMART

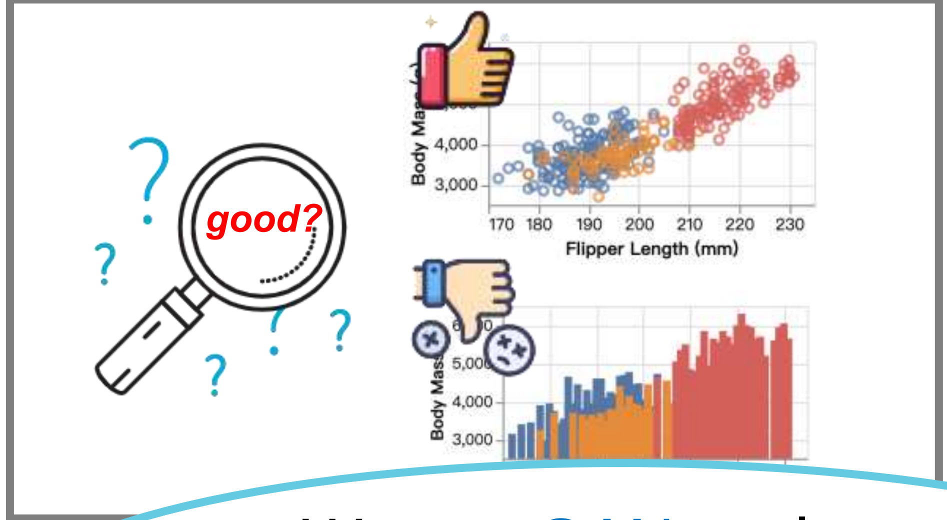
Scores

Learning Common User Preferences



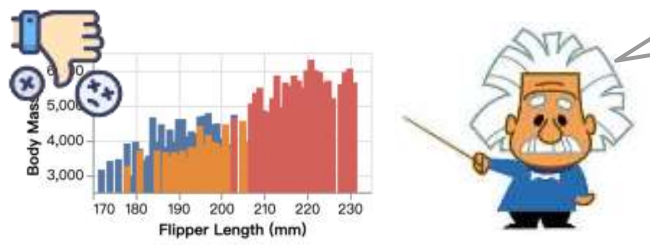
Composite Reward Function

How to quantify the goodness?



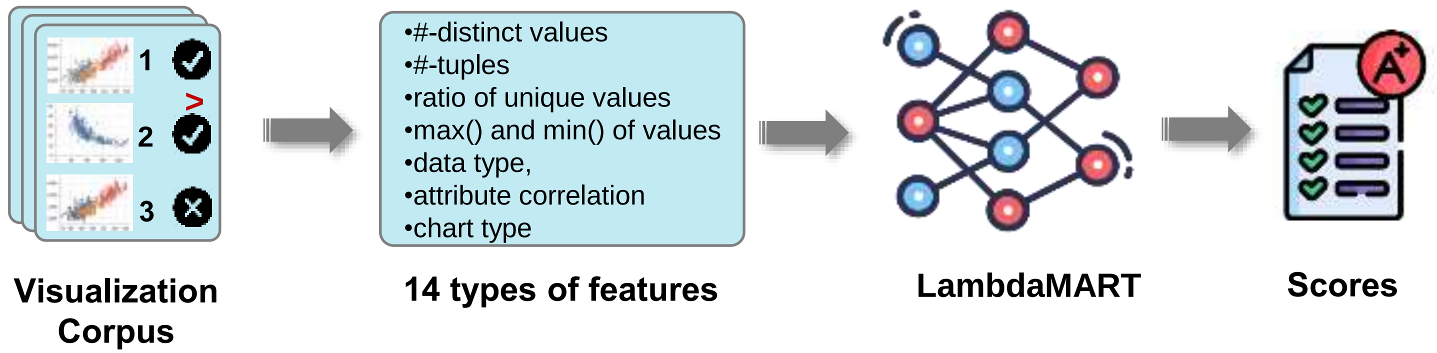
We use GANs to learn common user preferences from the Plotly community.

Learning from Domain Knowledge

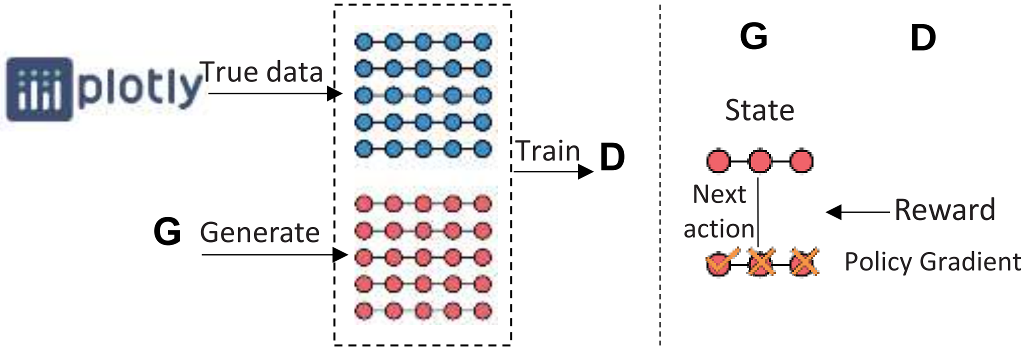


a bar chart with more than 50 bars is hard to get insights.

Capturing Data Features



Learning Common User Preferences



Composite Reward Function

香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

The composite reward function (CRF) is calculated as follows:

$$CRF = S_k \times (\beta S_d + (1 - \beta) S_u)$$

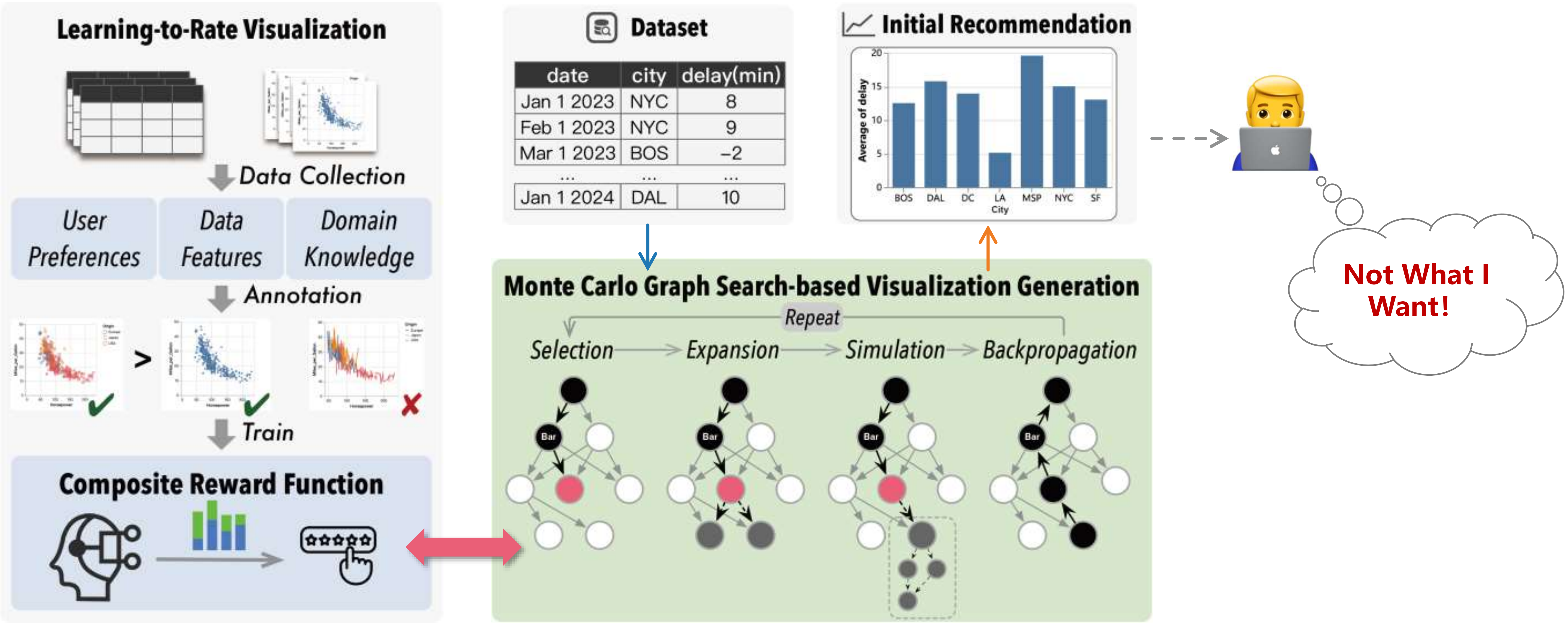
Where S_k is the domain knowledge score, S_d is the data feature score, and S_u is the user preference score. If S_k is 0, the reward CRF is 0; if S_k is 1, CRF is a weighted combination of S_d and S_u. The weight β controls the importance of S_d and S_u.

香港科技大学(广州)
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Challenge 3: How to Integrate User Feedback?

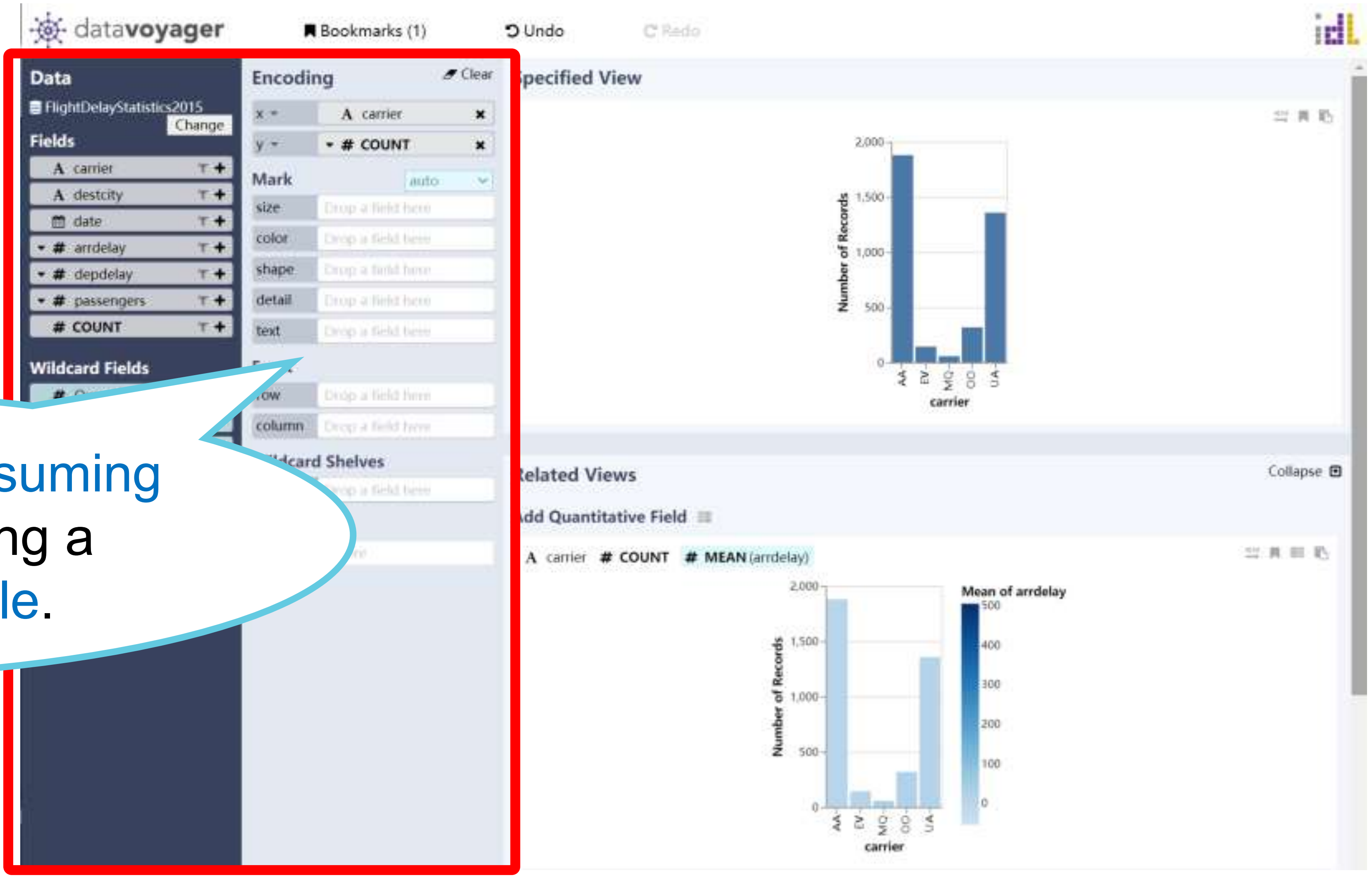
MCGS recommends high-quality visualizations, but they may not match user needs.

How can user feedback be used to align visualizations with user needs?



How to Integrate User Feedback?

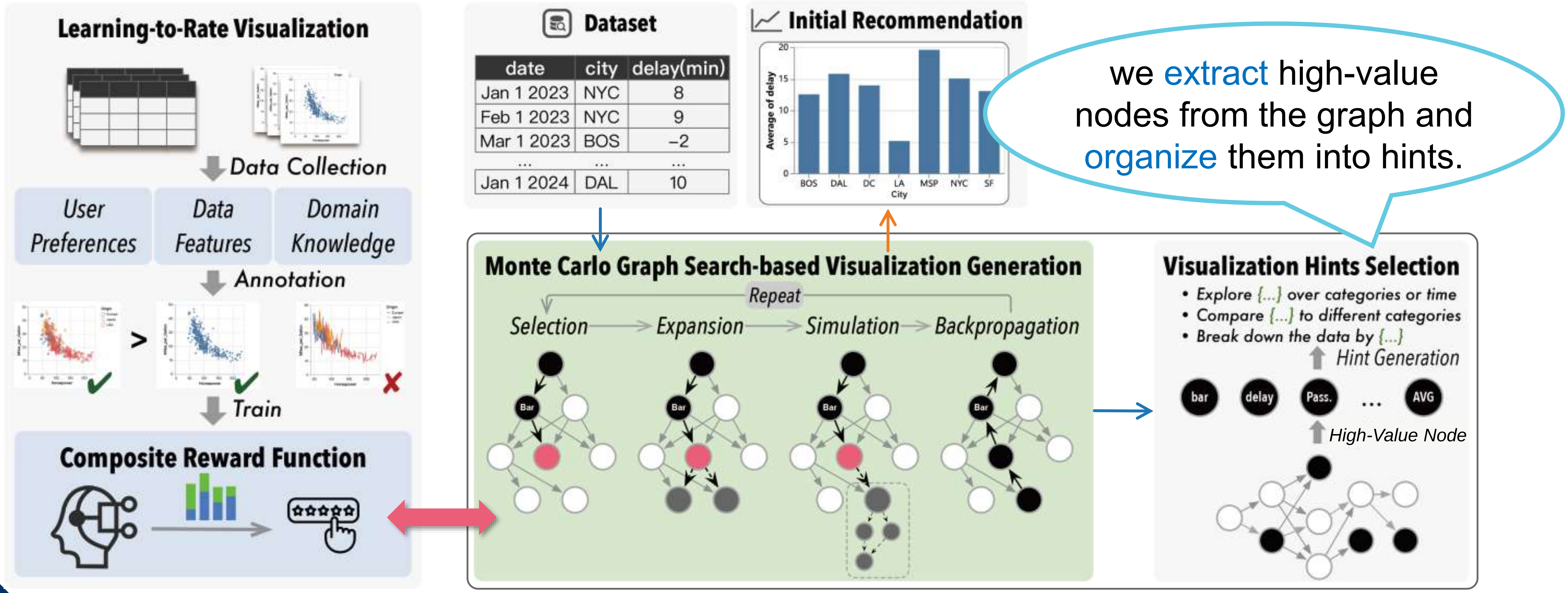
The common method is to provide a control panel for users to manually select data columns and visual encodings.



This process is time-consuming and can feel like solving a fill-in-the-blank puzzle.

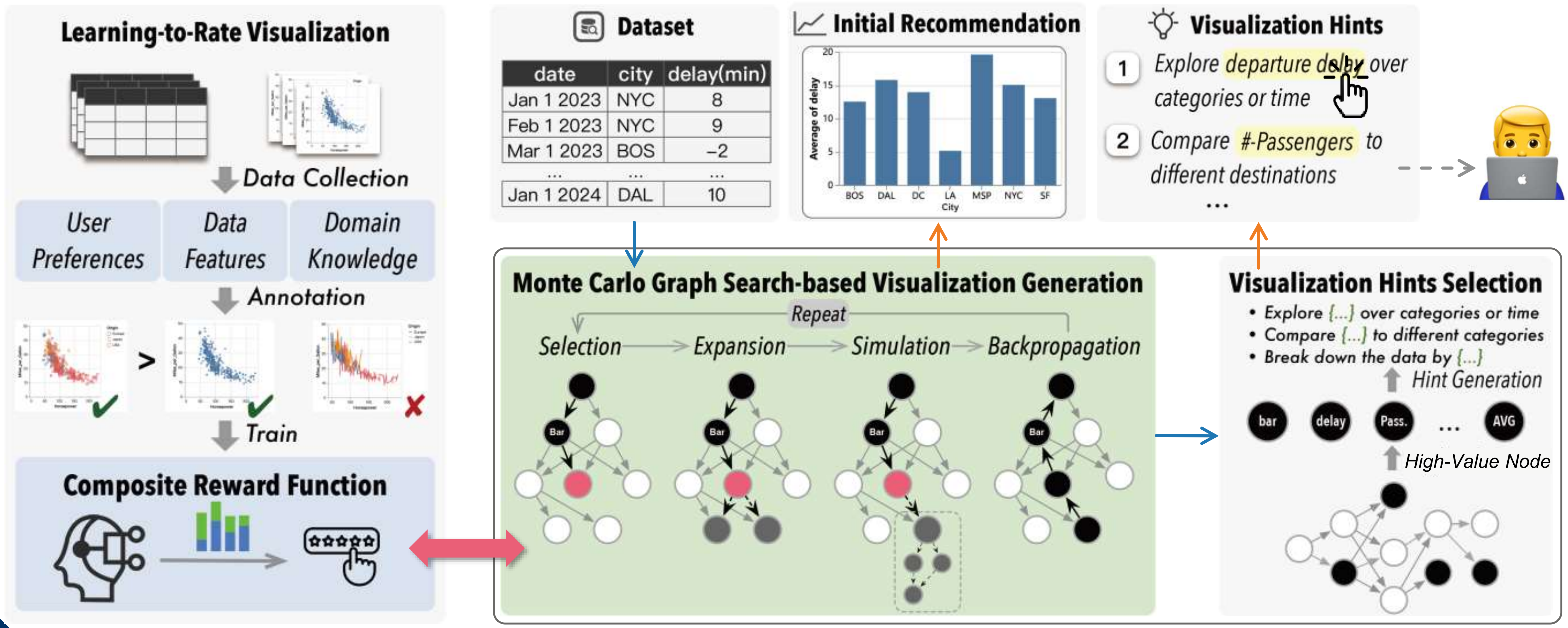
Visualization Hints Module

To address this, we introduce a visualization hints module.



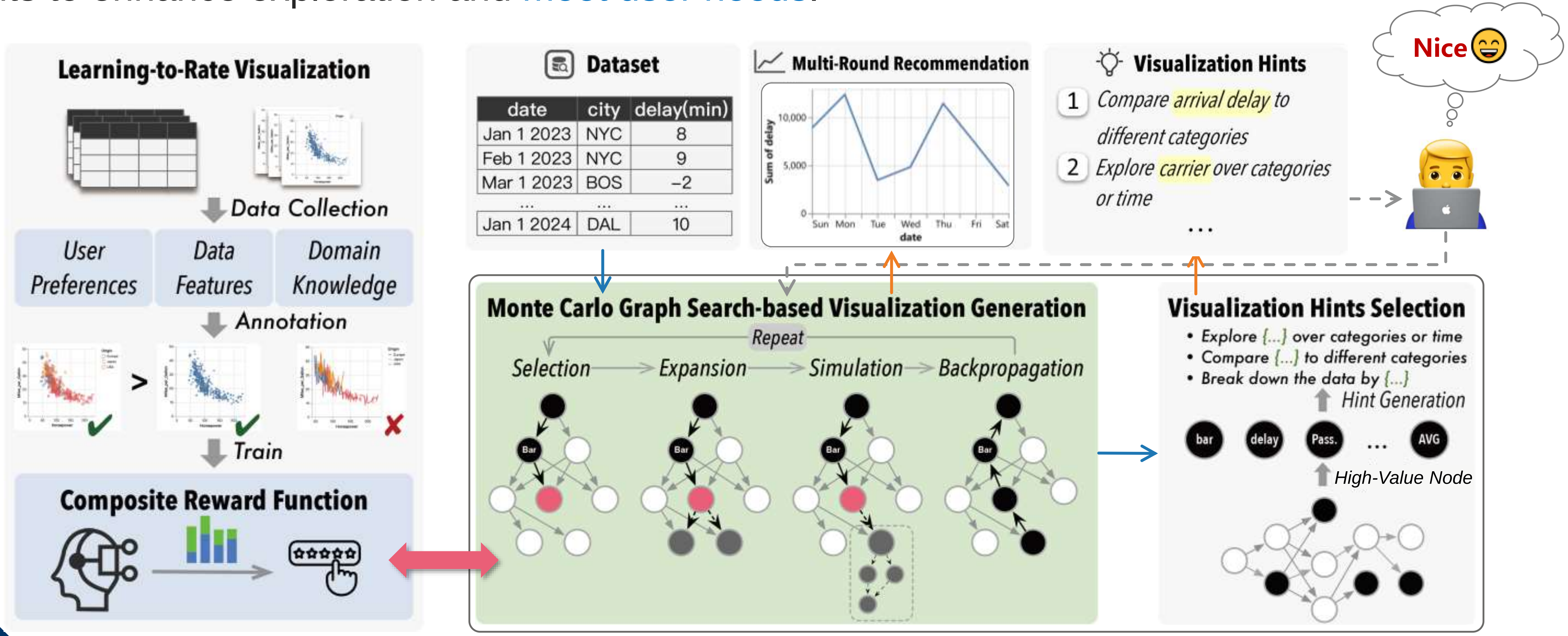
Visualization Hints Module

The user can indicate their analysis intent by clicking a hint.



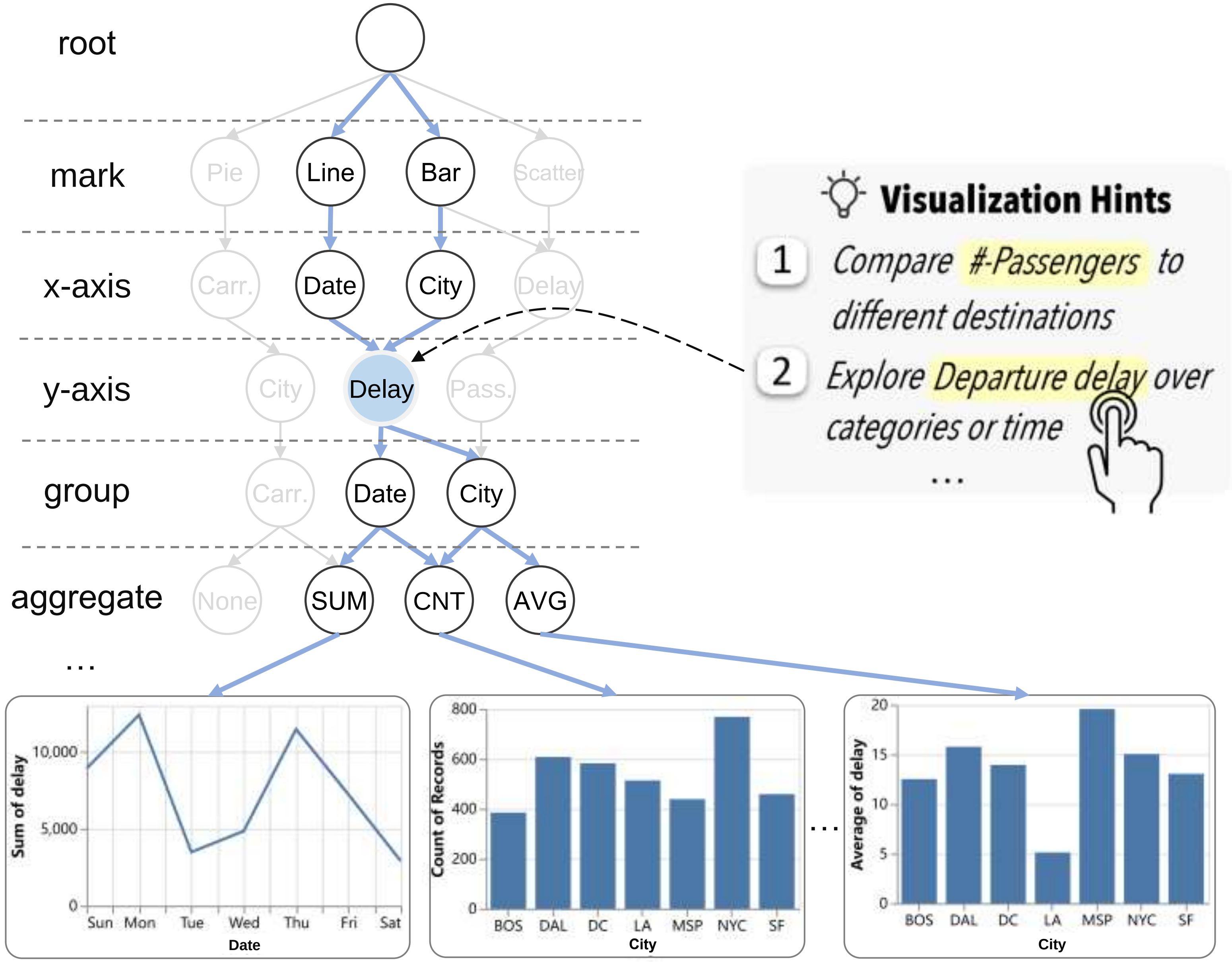
Visualization Hints Module

HAIChart then updates its recommendations, offering relevant visualizations and new hints to enhance exploration and meet user needs.

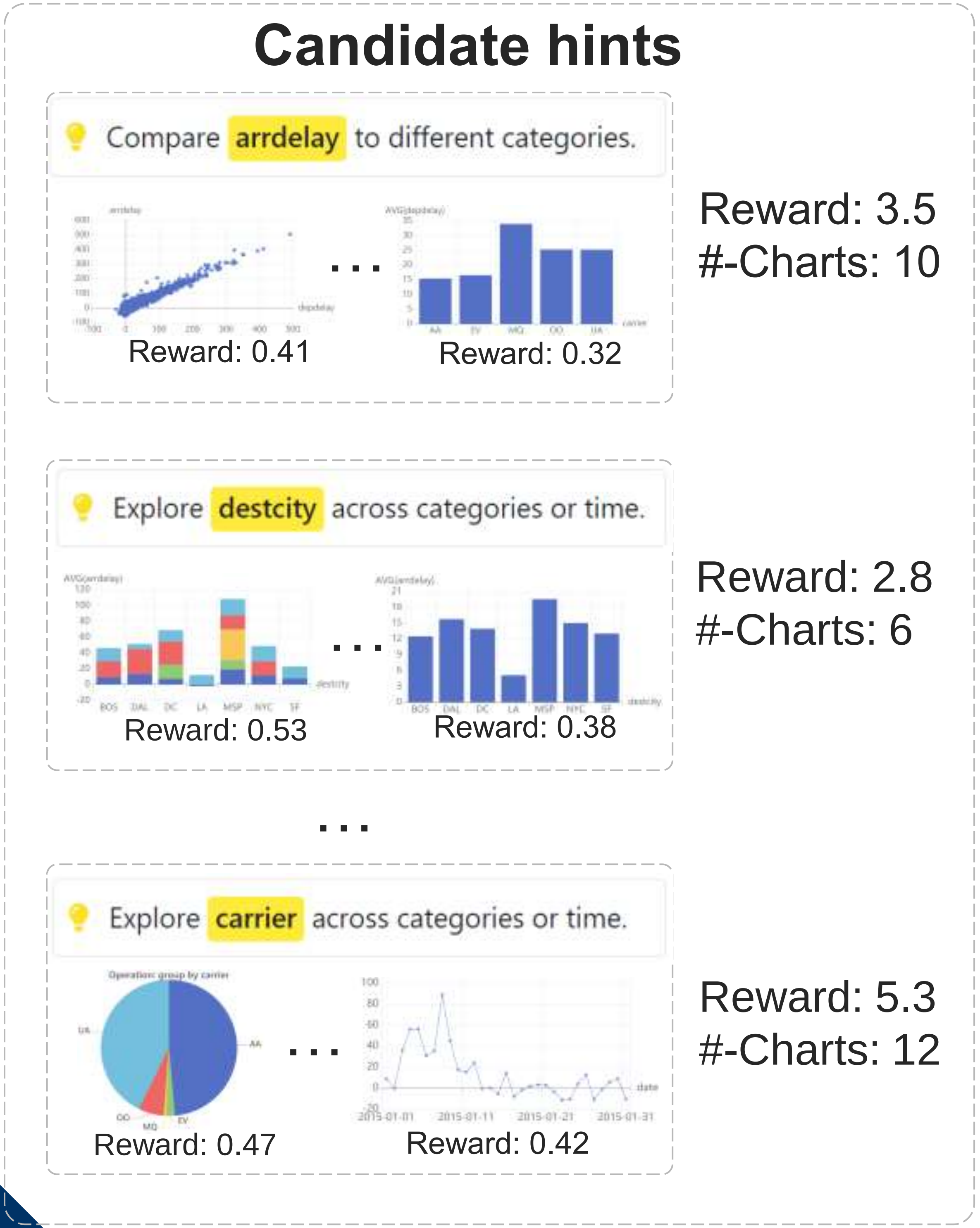


Visualization Hint

Visualization Hint: A visualization hint corresponds to a visualization operation described in natural language. Each hint is associated with a set of visualizations.



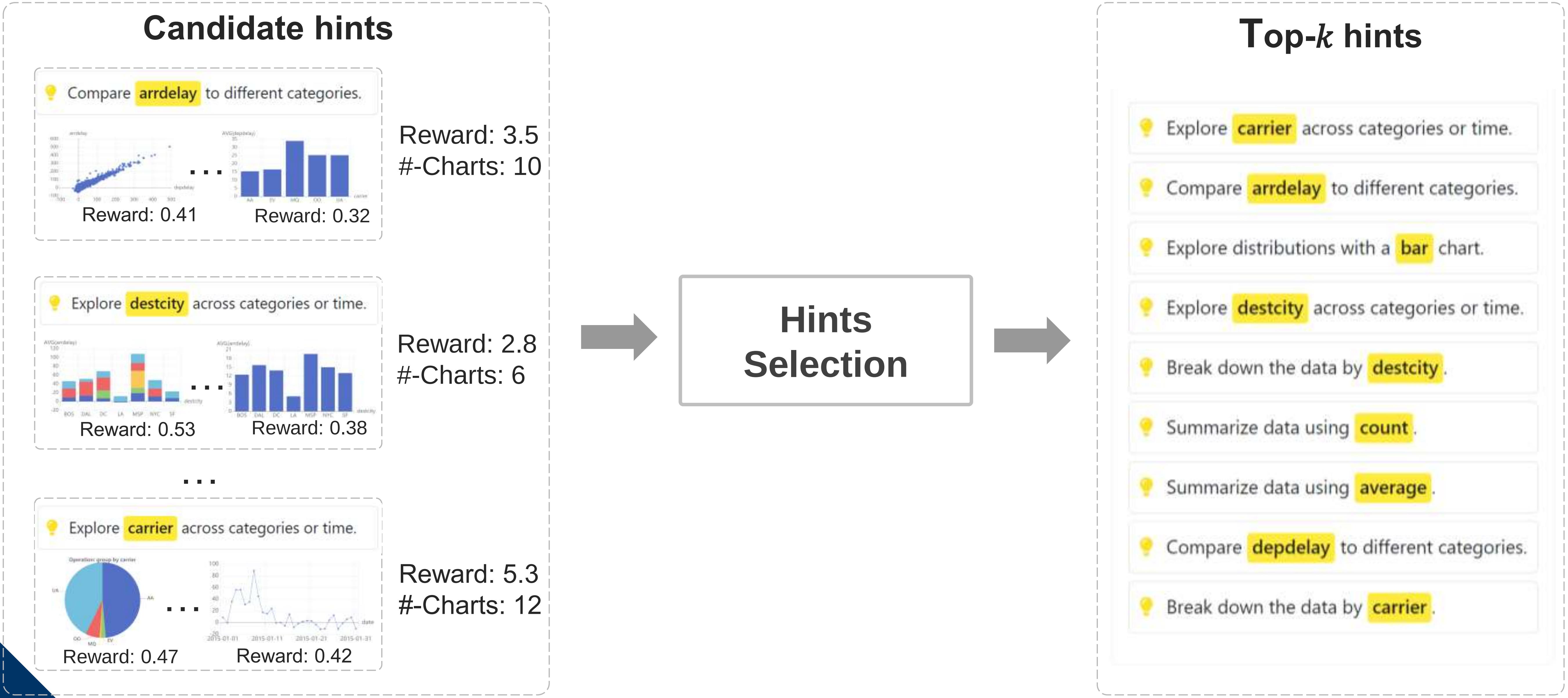
Visualization Hint



Given a set of hints $H = \{h_1, h_2, ..., h_n\}$, where each hint h_i is associated with a set of visualizations V_i and each visualization $v \in V_i$ has a reward value r_v.

Visualization Hints Selection

We aim to select top-k hints that not only cover different aspects but also ensure high-quality visualizations.



Visualization Hints Selection

The problem is defined as follows:

The goal is to select a subset $H' \subseteq H$ of k hints, where the total number of visualizations in H' does not exceed a budget B (#-Charts to users), and to maximize the overall reward value.

$$Maximize \quad F(H') = \sum_{h_i \in H'} \sum_{v \in V_i} r_c$$

$$Subject\ to \quad \sum_{h_i \in H'} |V_i| \le B \quad and \quad |H'| = k$$

This problem is NP-hard by a reduction from well-known the Budgeted Maximum Coverage problem.

Top-*k* Hints Selection

To address this, we propose a top-k visualization hints selection algorithm.

1. Select all hints with costs under budget B to create a candidate set (Line 2).
2. Sort the hints based on their average visualization score (Line 3).
3. Add hints to the final set until either the number of selected hints reaches k or the total cost exceeds budget B (Lines 5-10).

Algorithm 2: Top-k Visualization Hints Selection

Input: Set of hints $\mathbb{H} = \{h_1, h_2, \ldots, h_n\}$, B, k;

Output: Selected top-k set of hints $\mathbb{H}'$;

1 $\mathbb{H}' \leftarrow \emptyset$; $totalCost \leftarrow 0$;

2 $\mathbb{H}_v \leftarrow \{h \in \mathbb{H} \mid |h| \leq B\}$; // 1. Filter valid hints

3 $\mathbb{H}_v \leftarrow$ SortByScore($\mathbb{H}_v$); // 2. Sort hints by score

4 // 3. Selection of top-k hints

5 **for** *each* h_i *in* $\mathbb{H}_v$ **do**

6   **if** $|\mathbb{H}'| < k$ *and* $totalCost + |h_i| \leq B$ **then**

7         $\mathbb{H}'$.append(h_i);

8         $totalCost \leftarrow totalCost + |h_i|$;

9   **if** $|\mathbb{H}'| = k$ **then**

10   break;

11 **return** $\mathbb{H}'$;

Experiments Settings

Table 1: Statistics of the experimental datasets (Vis.: Charts)

Datasets	#-Tables	#-Vis.	Avg(#-Vis.)	Avg(#-Rows)	Avg(#-Col.)	Max(#-Col.)
VizML	79,475	162,905	2	2,817.8	3.3	25
KaggleBench	8	252	31.5	32,585.9	9.1	15

Datasets: We used VizML and KaggleBench benchmarks.

Evaluation Metrics: We used Hit@k, P@k, and Rt@k.

Comparison Methods: We compared HAIChart with 10 state-of-the-art methods: DeepEye, Data2Vis, VizGRank, PVisRec, VizML, LLM4Vis, Voyager2, HAIChart-, LLM4Vis+, and MCTS-based baseline.

Effectiveness of the First-round of Recommendations

D	Tasks	Metrics	The State-of-the-Art Methods						Our Methods		
			Data2Vis [10]	VizGRank [11]	DeepEye [20]	PVisRec [24]	VizML [14]	LLM4Vis [39]	MCTS	HAICHART-	HAICHART
VizML	Data Queries	Hit@1	47.5%	57.6%	52.4%	52.3%	-	-	78.3%	**79.7%**	79.3%
		Hit@3	51.3%	67.2%	67.6%	58.7%	-	-	88.2%	91.3%	**91.9%**
	Design Choices	Hit@1	41.7%	34.9%	34.1%	28.9%	28.7%	47.9%	42.4%	**50.6%**	48.7%
		Hit@3	43.7%	42.9%	40.7%	51.3%	-	-	77.1%	**81.8%**	81.5%
	Overall	Hit@1	24.3%	25.6%	25.7%	21.8%	-	-	33.1%	**37.9%**	36.9%
		Hit@3	26.9%	30.1%	33.9%	42.3%	-	-	64.7%	**68.4%**	67.4%
KaggleBench	Data Queries	P@10	41.2%	58.7%	62.5%	42.5%	-	-	52.2%	60.0%	**63.8%**
		R10@30	25.0%	50.0%	48.7%	67.5%	-	-	73.6%	80.1%	**83.7%**
	Design Choices	P@10	88.7%	87.5%	93.7%	91.9%	Hit@2:78.3%	Hit@2:87.6%	93.8%	**96.3%**	**96.3%**
		R10@30	95.0%	81.3%	95.0%	85.0%	-	-	92.5%	**96.2%**	**96.2%**
	Overall	P@10	28.7%	43.7%	48.7%	36.7%	-	-	45.4%	51.3%	**55.0%**
		R10@30	13.8%	41.3%	33.7%	60.0%	-	-	63.8%	72.5%	**74.9%**

Overall, our methods (HAIChart and HAIChart-) significantly outperform all state-of-the-art methods across all metrics, showing the effectiveness of our framework.

Effectiveness of Multi-round Recommendations

香港科技大学（广州）
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

We design a **user study**:

- **Participants:** 17 participants.12 experts and 5 non-experts.
- **Task:**
  - Participants used HAIChart, DeepEye, LLM4Vis+, and Voyager2 with KaggleBench datasets.
  - Tasks included specific analyses and open-ended explorations.
- **Procedure:**
  - Preparation: Introduced study context, datasets, and tools.
  - Experimentation: Logged interactions and provided new recommendations and hints.
  - User Feedback: Rated tools on ease of use and quality, and collected feedback through interviews.

Effectiveness of Multi-round Recommendations

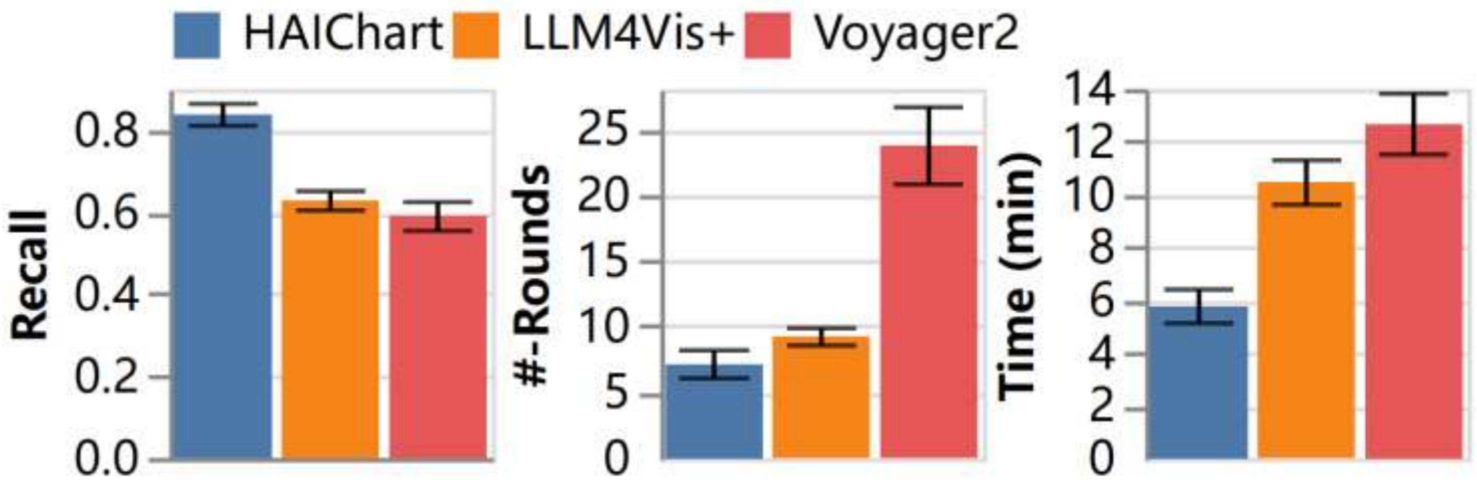


Figure 1: Quantitative analysis on user study

Quantitative Analysis:

Unlike Voyager2's manual exploration and LLM4Vis+'s time-consuming natural language queries. Experiments show that HAIChart meets user analysis needs with lower interaction costs.

Effectiveness of Multi-round Recommendations

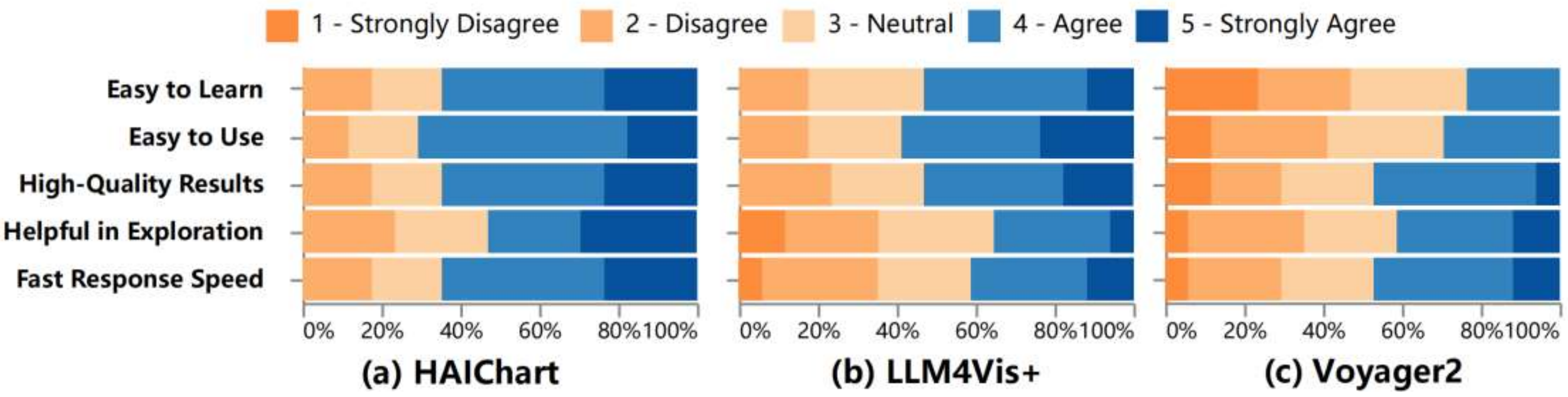


Figure 1: Qualitative analysis on user study

Qualitative Feedback:

User feedback shows that HAIChart is easy to learn and use, helping users explore data and achieve high-quality visualizations efficiently.

香港科技大学(广州)
THE HONG KONG
UNIVERSITY OF SCIENCE AND
TECHNOLOGY (GUANGZHOU)

THANK YOU & QUESTIONS?