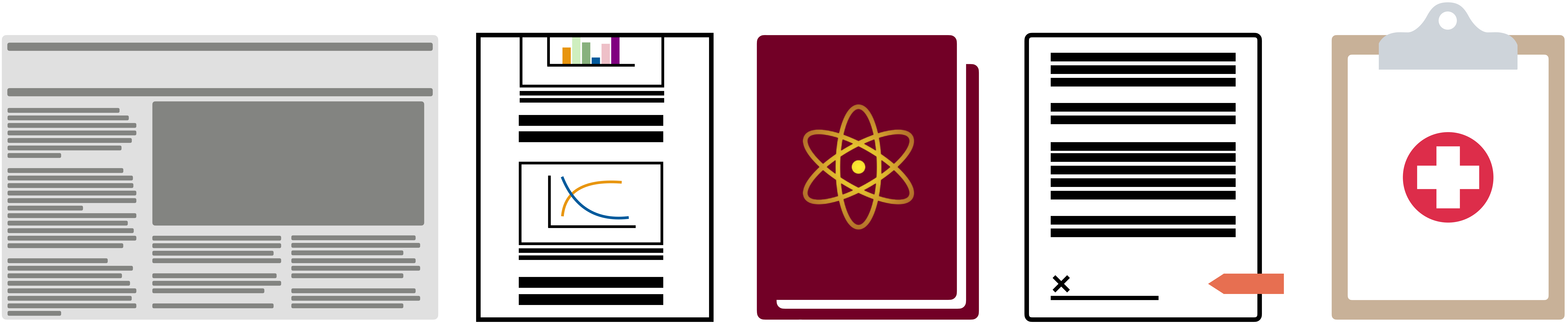
User Interfaces for Fine-grained Integration of Information

Dissertation Defense • Wednesday, November 26, 2025

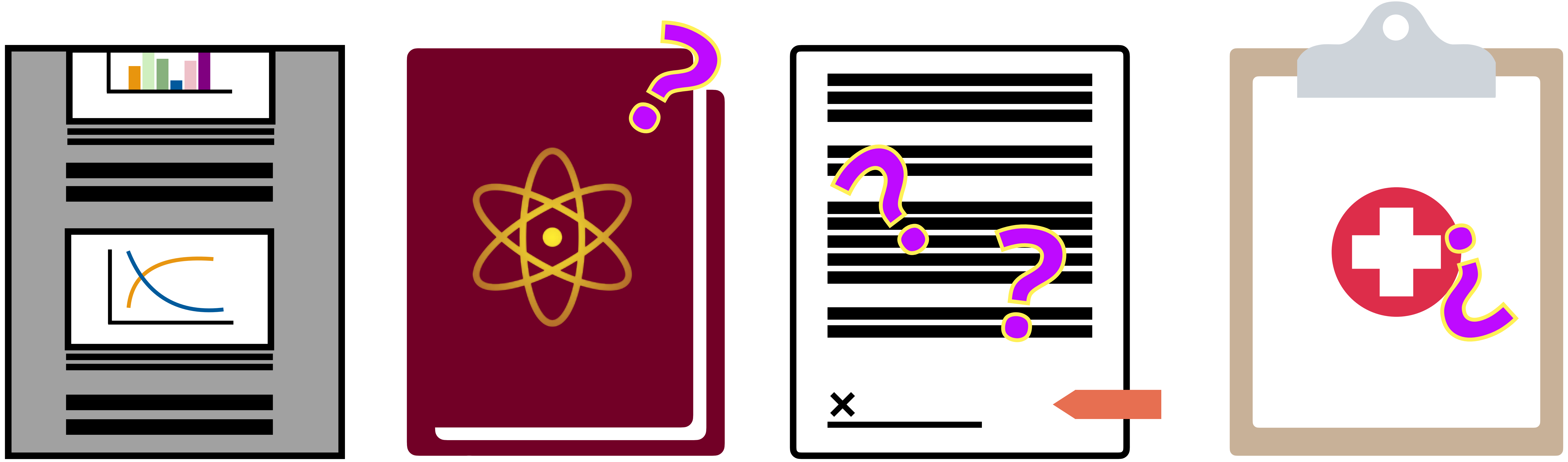
Committee: Anindya De (Chair), Susan Davidson, Insup Lee, Yale Cohen (PSOM)

Alyssa Hwang

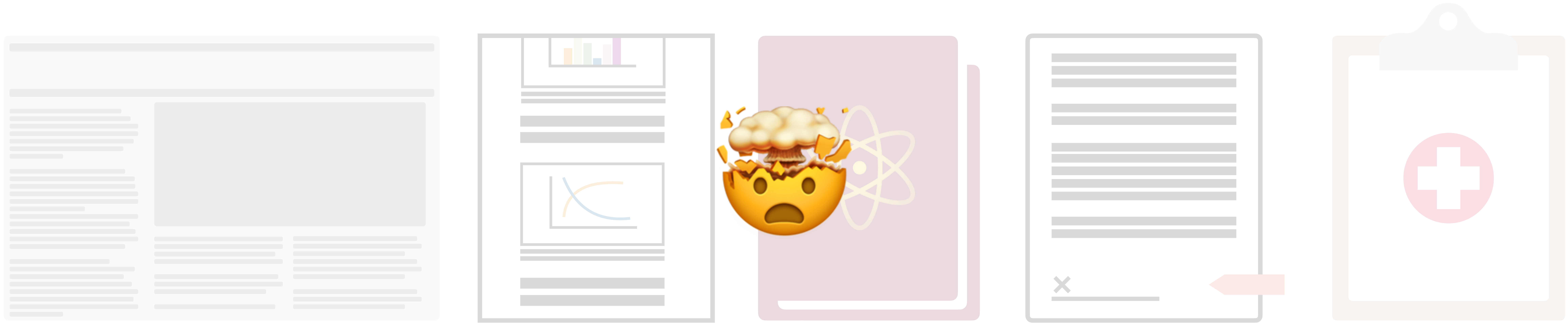
Knowledge is shared through (complex) documents.



Knowledge is shared through (complex) documents.



Knowledge is shared through (complex) documents.



Research questions

1. What **challenges** do users face when synthesizing ideas within complex documents?
2. How can we systematically **represent** connections between ideas to make them easier to find?
3. Does surfacing these connections measurably **improve** comprehension?

Thesis statement: exposing connections between related details in complex documents can improve comprehension without penalizing time or cognitive load.

Presentation structure

1. Needs-finding study
2. Framework design
3. Instantiation
4. Evaluation
5. Findings

Presentation structure

1. Needs-finding study
2. Framework design
3. Instantiation
4. Evaluation
5. Findings

1. Needs-finding study

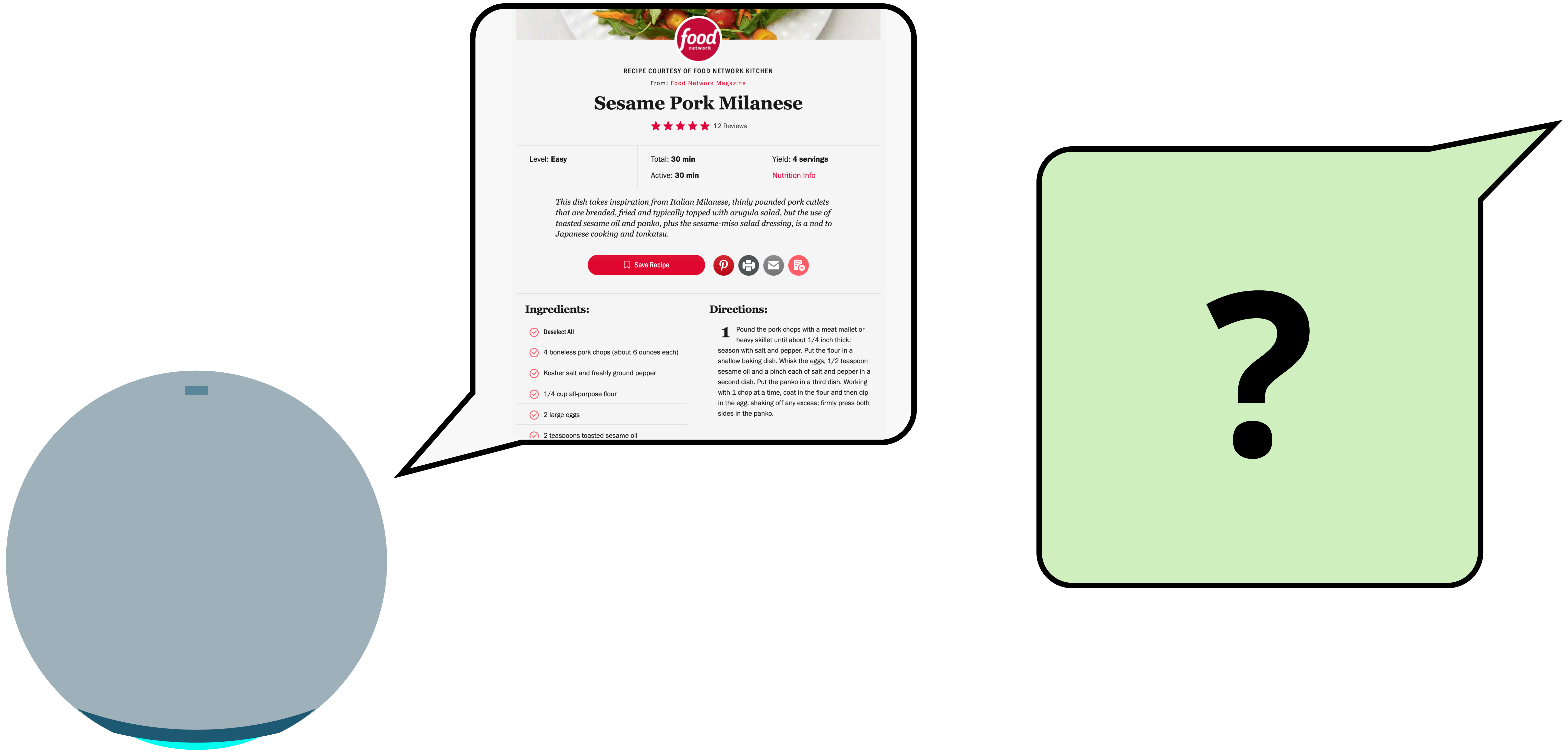
Goal: understand user needs when integrating individual details

Goal: understand user needs when integrating individual details

Method: observe users cooking at home with voice assistant

Goal: understand user needs when integrating individual details

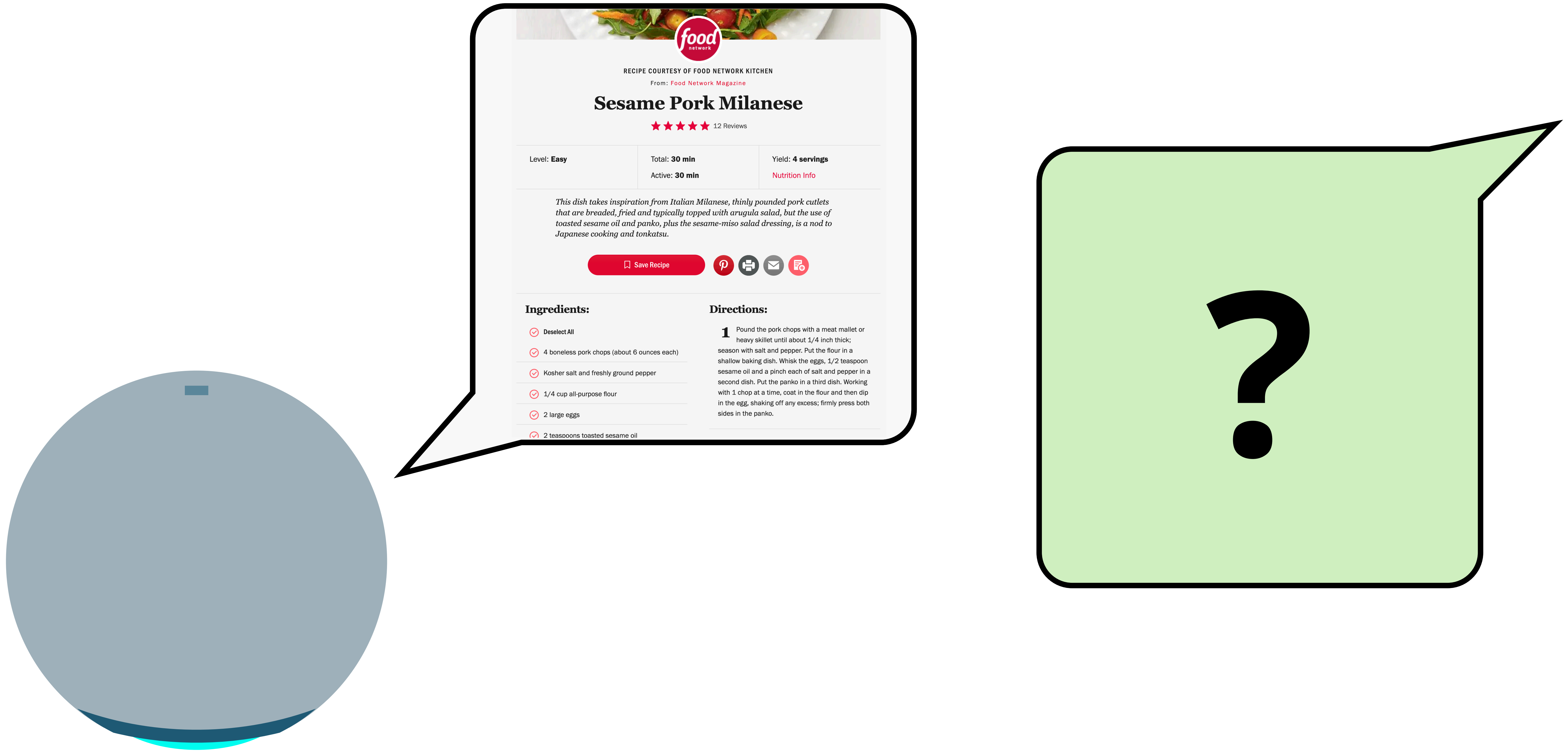
Method: observe users cooking at home with voice assistant



Goal: understand user needs when integrating individual details

Method: observe users cooking at home with voice assistant

Outcome: definition of user challenges and potential augmentations



9 common challenges

1. Missing the big picture
2. Information overload
3. Fragmentation
4. Time insensitivity
5. Missing details
6. Discarded context
7. Failure to listen
8. Uncommunicated affordances
9. Limitations of audio

9 common challenges

1. Missing the big picture
2. Information overload
3. Fragmentation
4. Time insensitivity
5. Missing details
6. Discarded context
7. Failure to listen
8. Uncommunicated affordances
9. Limitations of audio

9 common challenges

1. Missing the big picture
2. Information overload
3. Fragmentation
4. Time insensitivity
5. Missing details
6. Discarded context
7. Failure to listen
8. Uncommunicated affordances
9. Limitations of audio

9 common challenges

1. Missing the big picture
2. Information overload
3. Fragmentation
4. Time insensitivity
5. Missing details
6. Discarded context
7. Failure to listen
8. Uncommunicated affordances
9. Limitations of audio

> Put the avocados in a large bowl and gently toss with the tomatoes, lemon juice, shallots, 2 tablespoons oil, 1/2 teaspoon salt and the reserved herbs. Transfer to a serving bowl.

Herb-Roasted Salmon with Tomato-Avocado Salsa, Food Network

Ingredients:

- Deselect All
- 2 tablespoons olive oil, plus more for the baking sheet and salmon
- 1/3 cup finely chopped fresh dill
- 1/3 cup finely chopped fresh flat-leaf parsley
- 3 tablespoons finely chopped fresh chives
- 3 tablespoons finely chopped fresh basil
- 2 1/4 pounds center-cut salmon fillet, skin and bones removed
- Kosher salt and freshly ground black pepper
- 2 large avocados
- 12 ounces mixed-colored cherry or grape tomatoes, halved or quartered if large
- 2 tablespoons fresh lemon juice
- 1 small shallot, minced

Directions:

1 Preheat the oven to 350 degrees F. Line a large rimmed baking sheet with parchment paper and brush it lightly with oil.

2 Mix together the dill, parsley, chives and basil in a small bowl. Reserve 2 tablespoons of the mixture for the salsa and set aside.

3 Put the salmon on the prepared baking sheet and sprinkle all over with salt and pepper. Drizzle the top lightly with oil, then top evenly with the herb mix. Bake until just cooked through, 20 to 25 minutes.

4 Meanwhile, halve and peel the avocados and cut them into 1/2-inch pieces. Put the avocados in a large bowl and gently toss with the tomatoes, lemon juice, shallots, 2 tablespoons oil, 1/2 teaspoon salt and the reserved herbs. Transfer to a serving bowl.

5 Serve the salmon with the salsa on the side.

1. Missing t
2. Informat
3. Fragmen
4. Time inse
5. Missing c
6. Discarde
7. Failure to
8. Uncomm
9. Limitatio

Herb-Roasted Salmon with Tomato-Avocado Salsa, Food Network

9 Common Challenges

9. Limitations of audio

1

What challenges do users face when synthesizing ideas within complex documents?

Participants needed help acquiring the right information at the right time.

2. Framework design

Goal: develop a method to expose connections between related info

Goal: develop a method to expose connections between related info

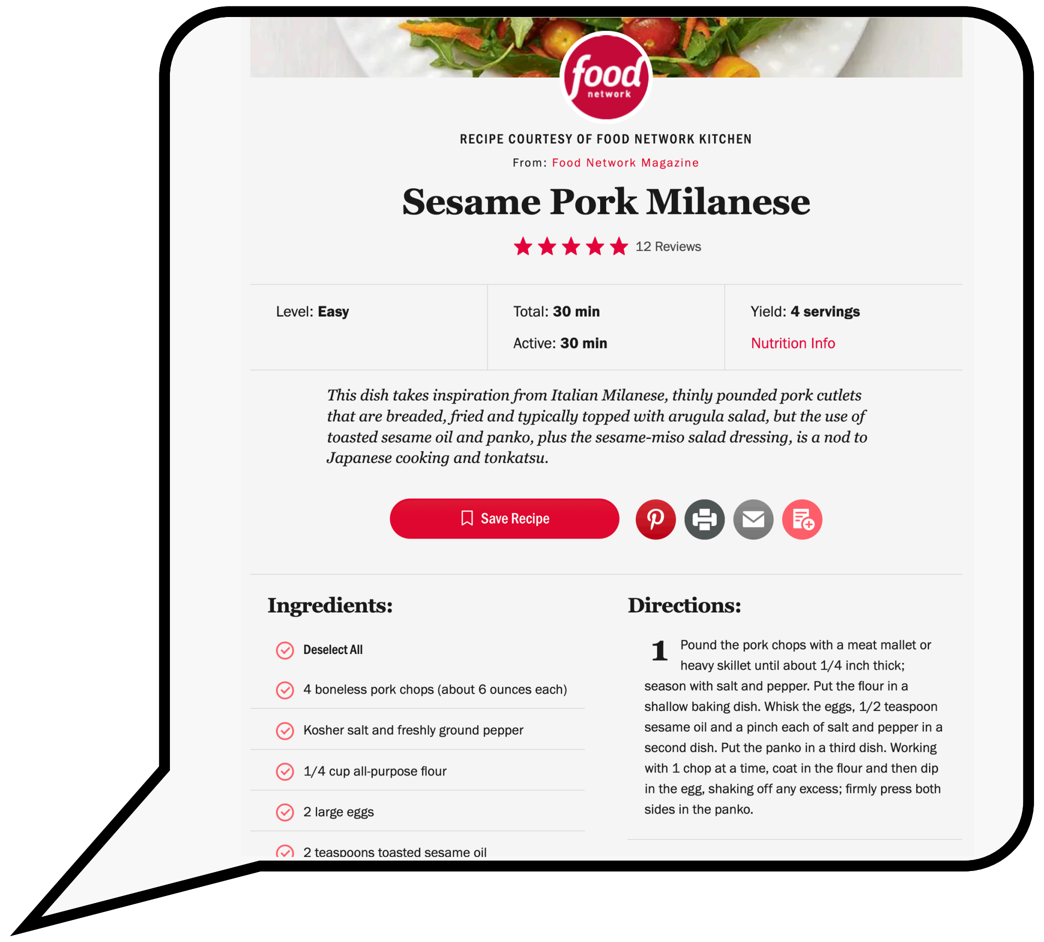
Method: iterative design with feedback from think-aloud user studies

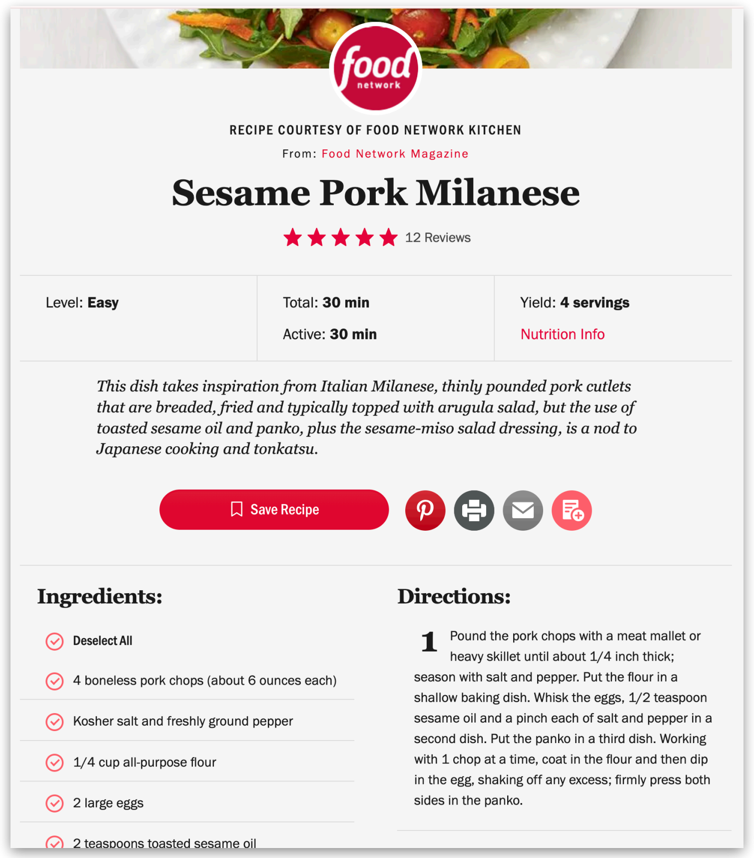
Goal: develop a method to expose connections between related info

Method: iterative design with feedback from think-aloud user studies

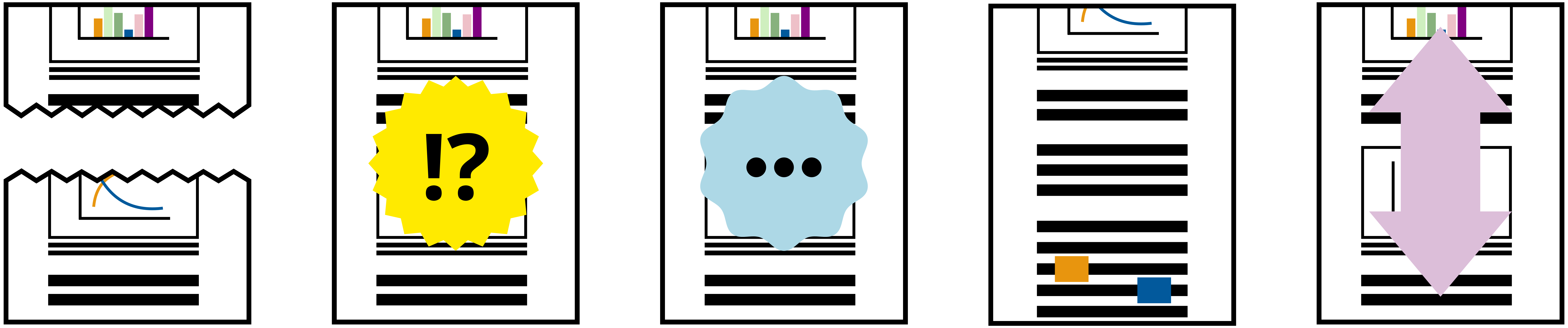
Outcome: framework for fine-grained augmentations

Listening to recipes ➔ reading papers



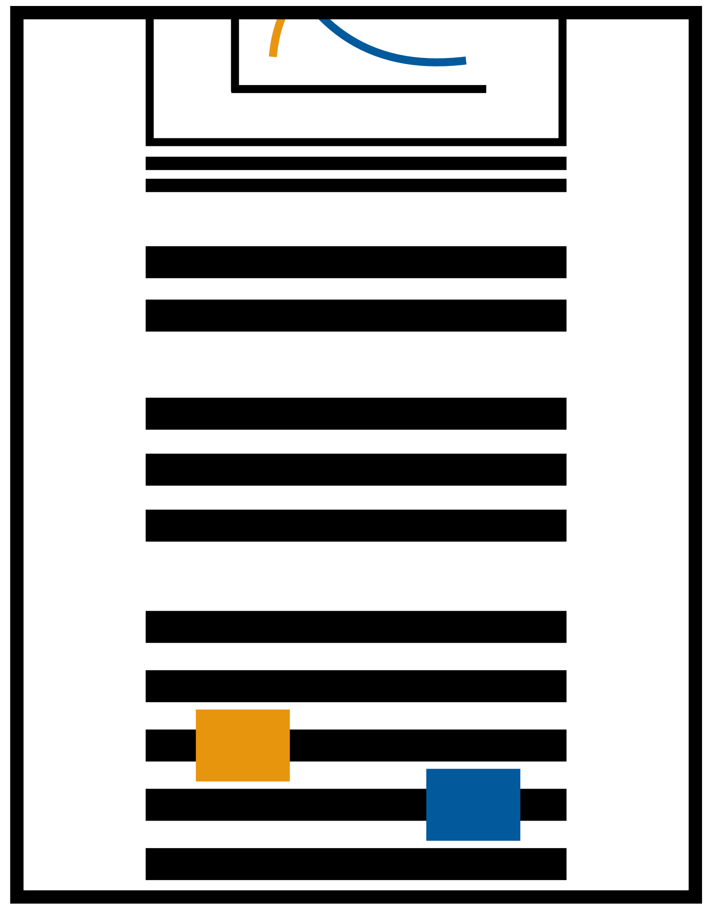


Reading challenges

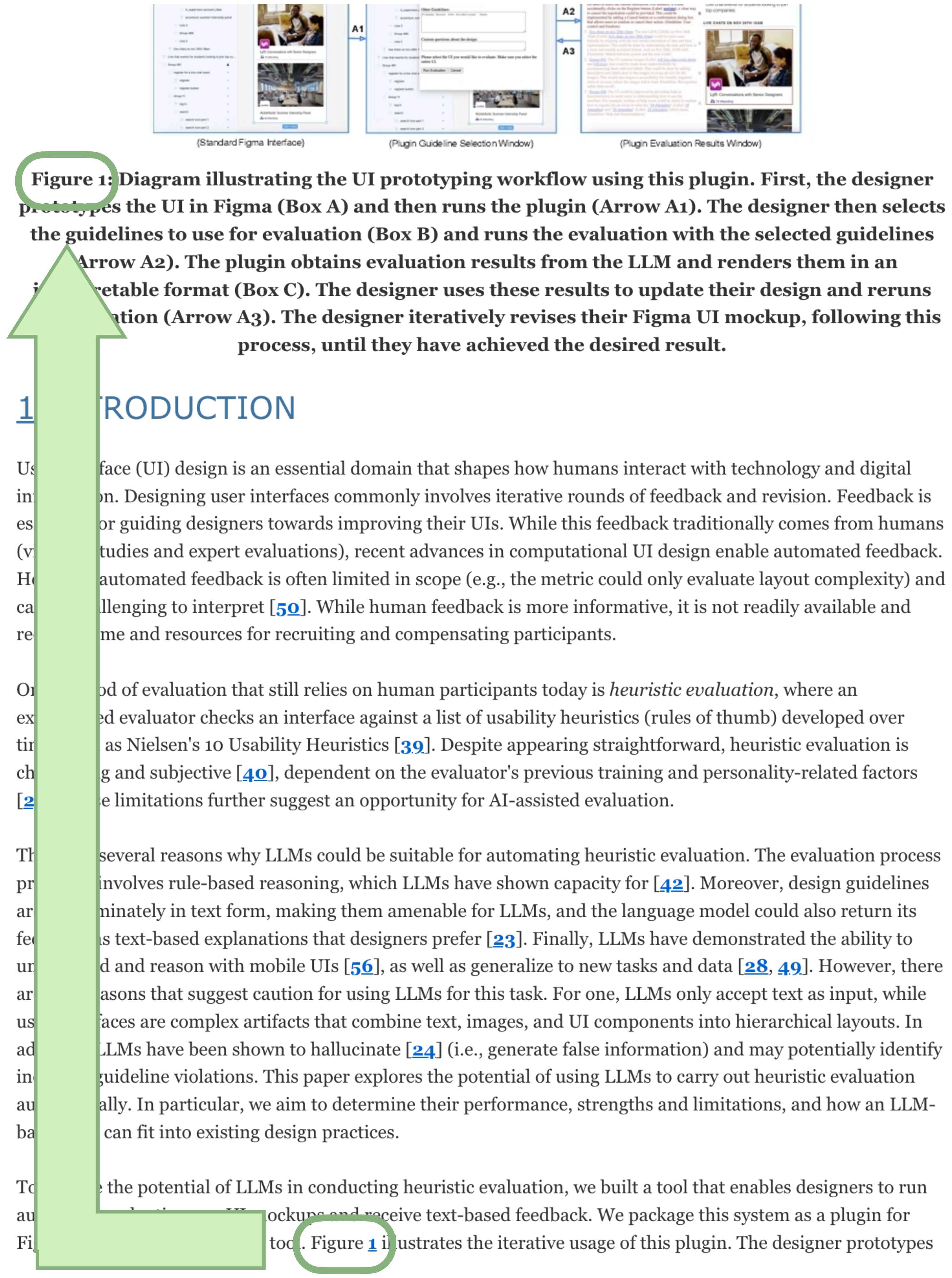


Fragmentation | Complexity | Interpretation | Visibility | Searching

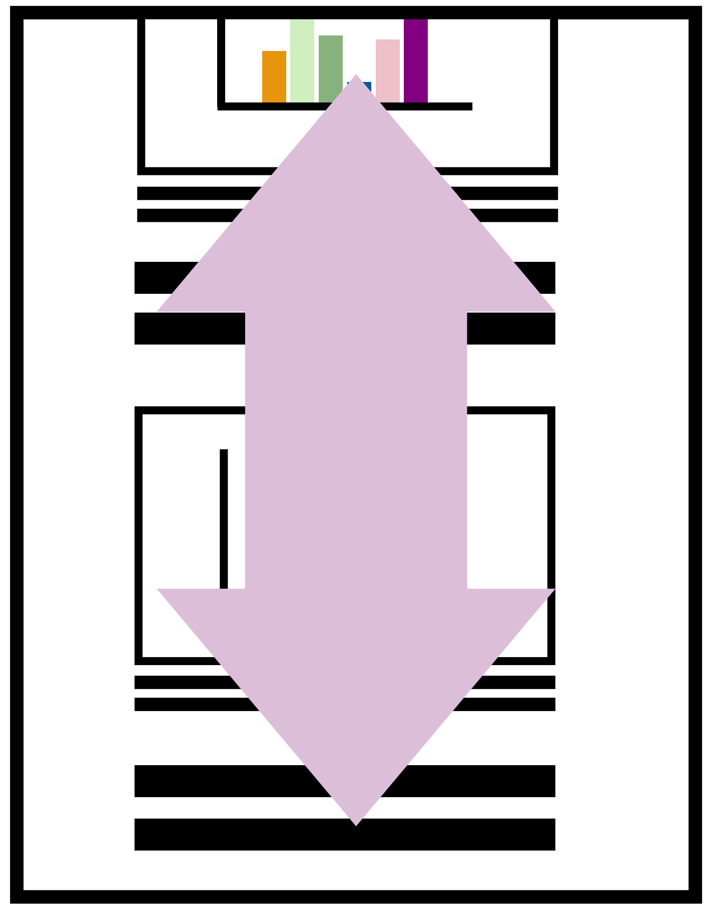
Reading challenges



Visibility



Reading challenges



Searching

4 STUDY METHOD

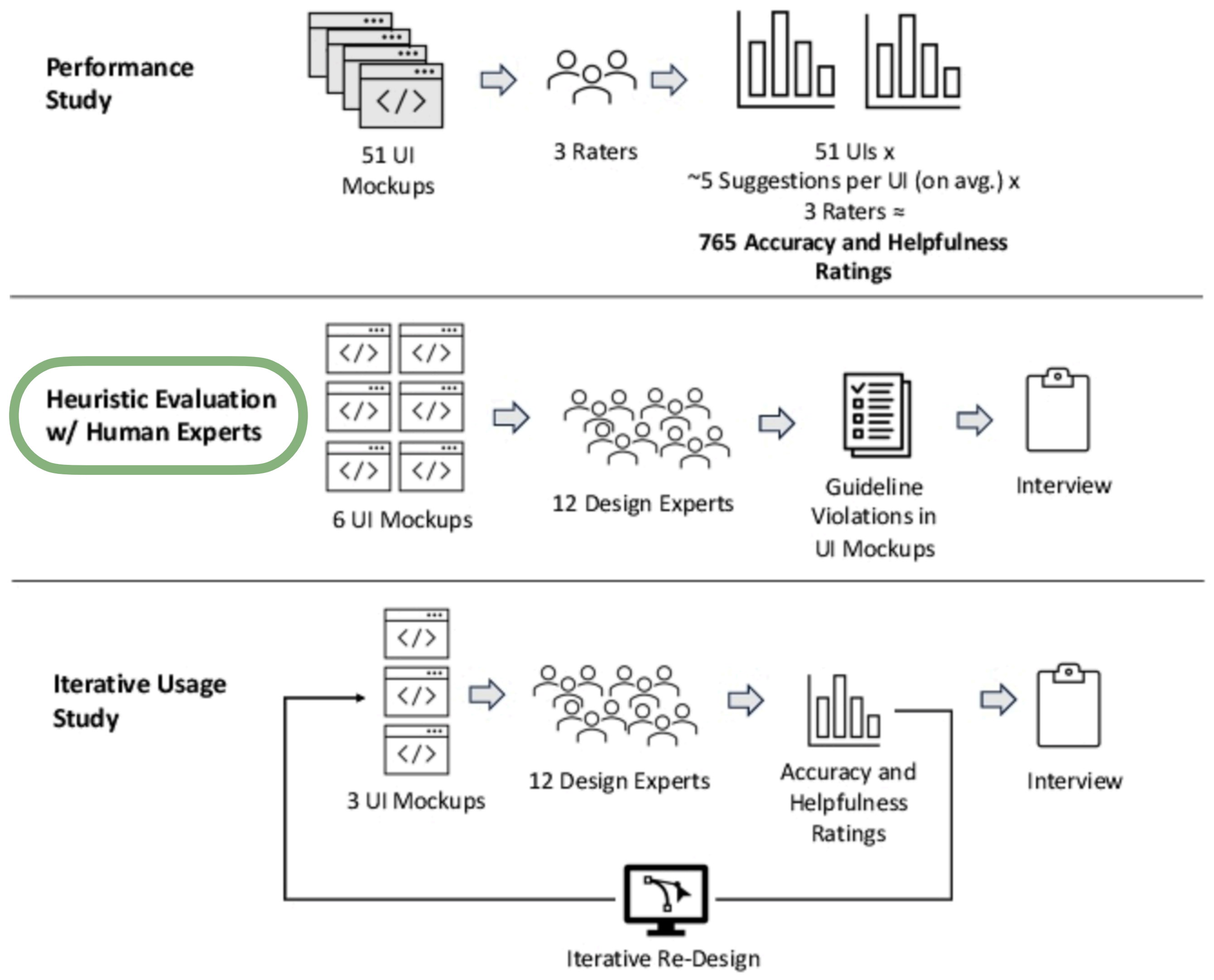
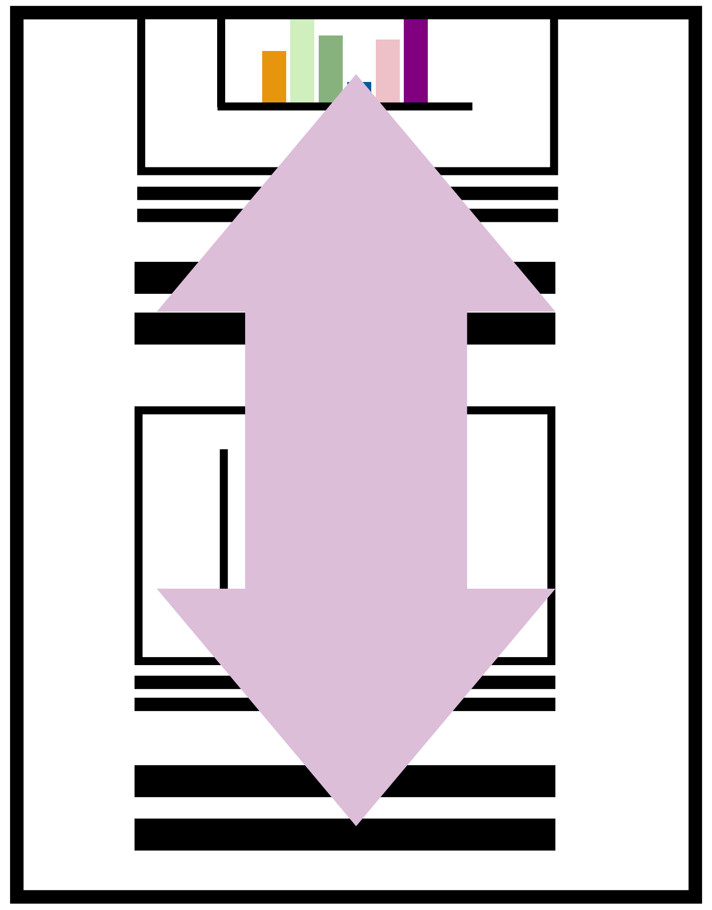


Figure 5: An illustration of the formats of the three studies. The Performance Study consists of 3 raters evaluating the accuracy and helpfulness of GPT-4-generated suggestions for 51 UI mockups. The Heuristic Evaluation Study with Human Experts consists of 12 design experts, who each looked for guideline violations in 6 UIs, and finishes with an interview asking them to compare their violations with those found by the LLM. Finally, the Iterative Usage study comprises of another group of 12 design experts, each working with 3 UI mockups. For each mockup, the expert iteratively revises the design based on the LLM's valid suggestions and rates the LLM's feedback, going through 2-3 rounds of this per UI. The Usage study concludes with an interview about the expert's experience with the tool.

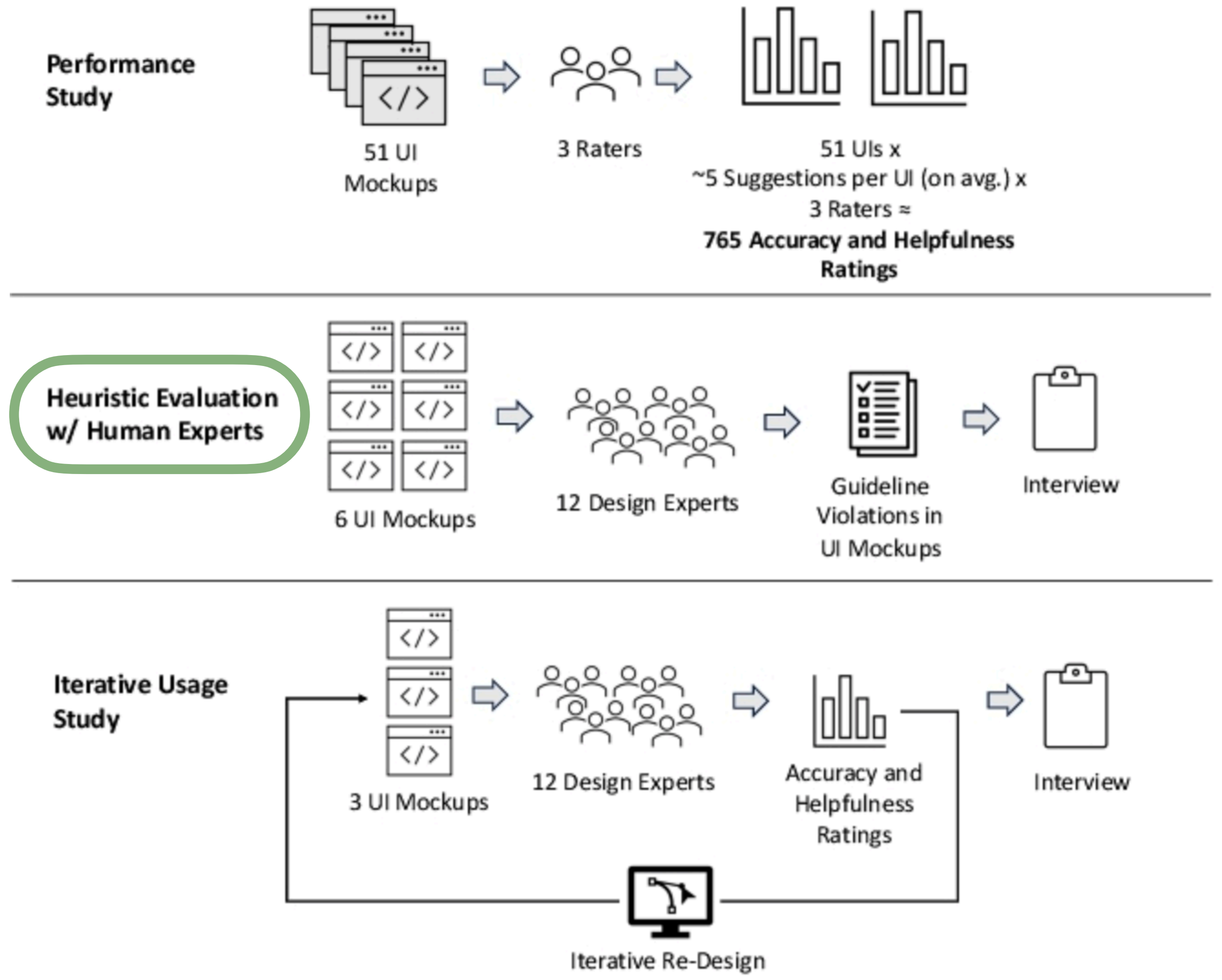
To explore the potential of GPT-4 in automating heuristic evaluation, we carried out three studies (see Figure 5).

Reading challenges



Searching

4 STUDY METHOD



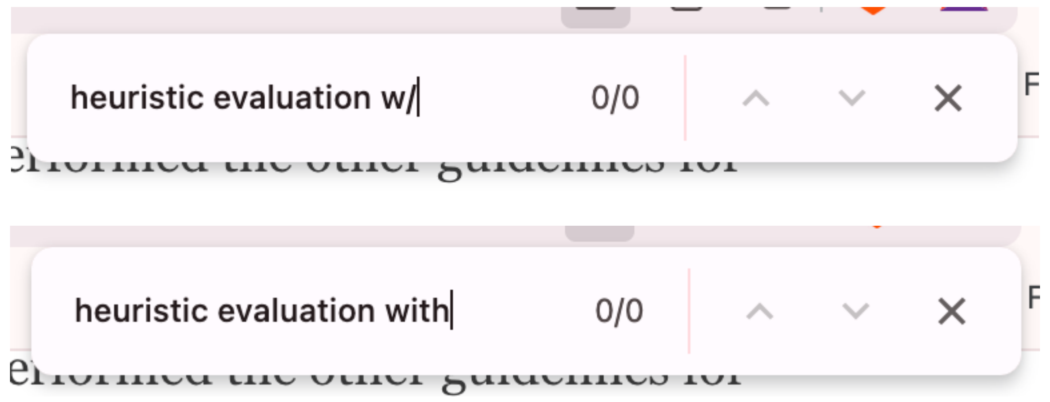
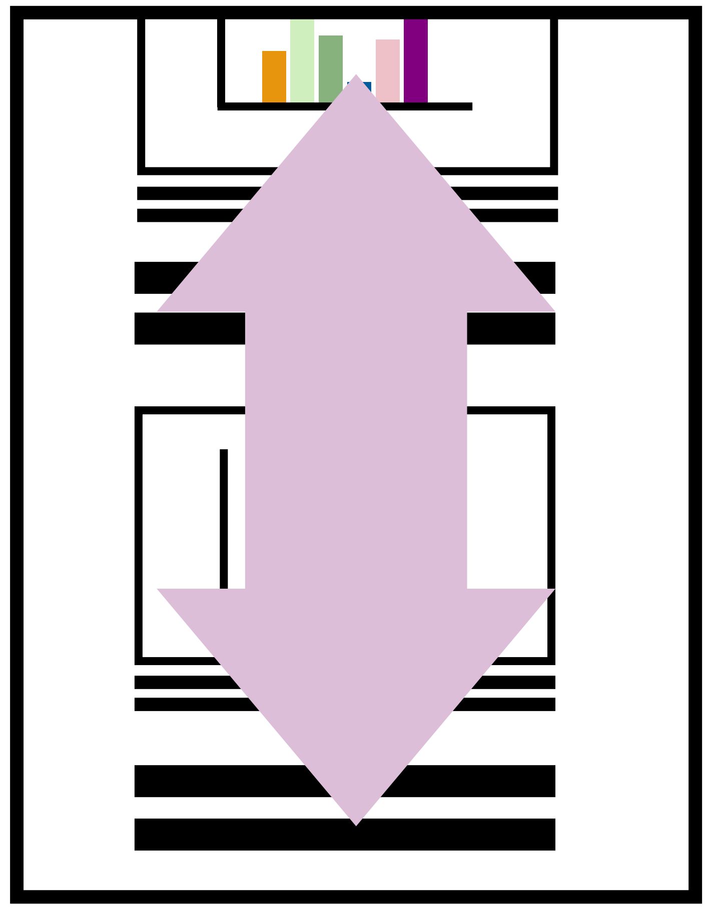


Figure 5: An illustration of the formats of the three studies. The Performance Study consists of 3 raters evaluating the accuracy and helpfulness of GPT-4-generated suggestions for 51 UI mockups. The Heuristic Evaluation Study with Human Experts consists of 12 design experts, who each looked for guideline violations in 6 UIs, and finishes with an interview asking them to compare their violations with those found by the LLM. Finally, the Iterative Usage study comprises of another group of 12 design experts, each working with 3 UI mockups. For each mockup, the expert iteratively revises the design based on the LLM's valid suggestions and rates the LLM's feedback, going through 2-3 rounds of this per UI. The Usage study concludes with an interview about the expert's experience with the tool.

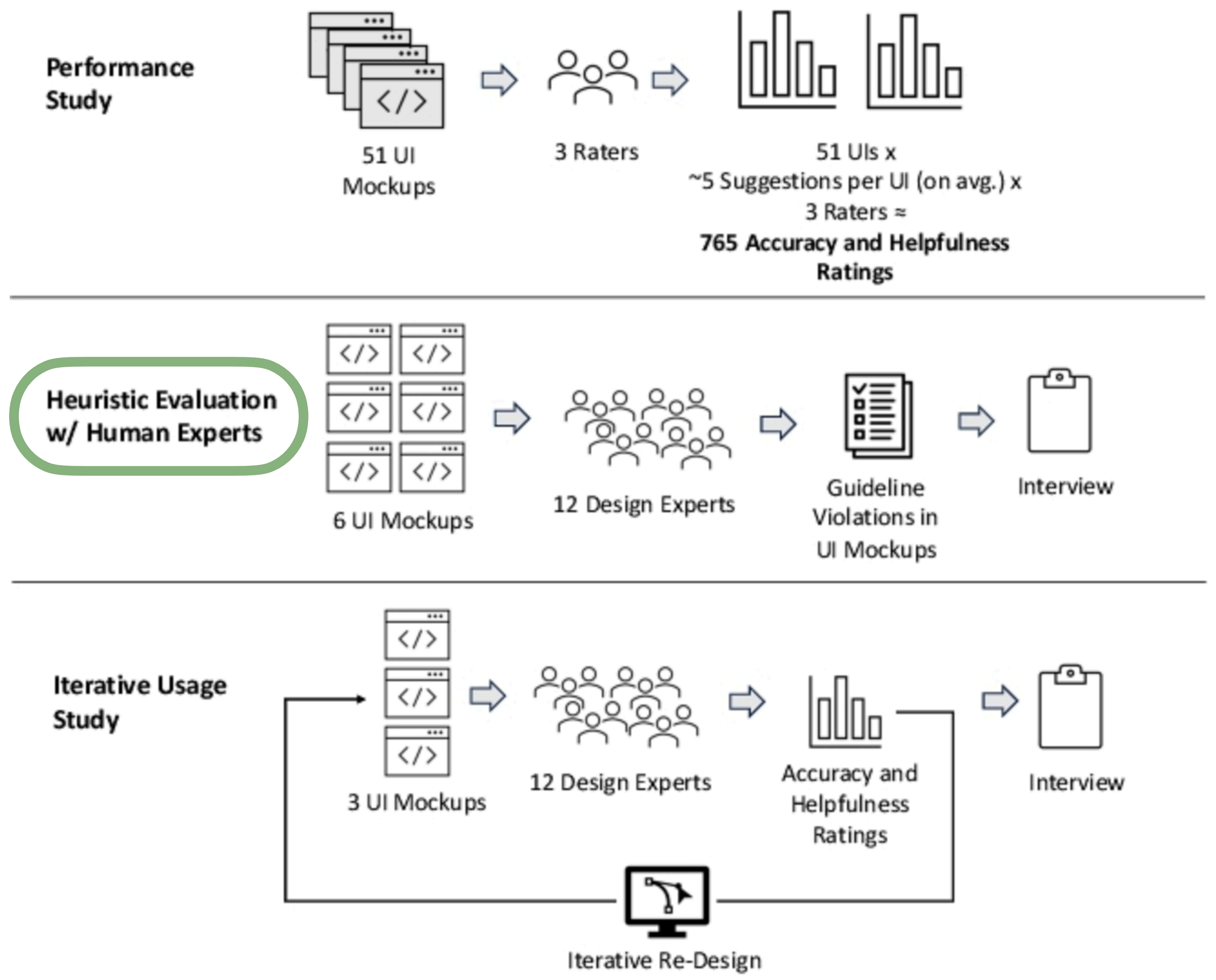
To explore the potential of GPT-4 in automating heuristic evaluation, we carried out three studies (see Figure 5).

Reading challenges



Searching

4 STUDY METHOD



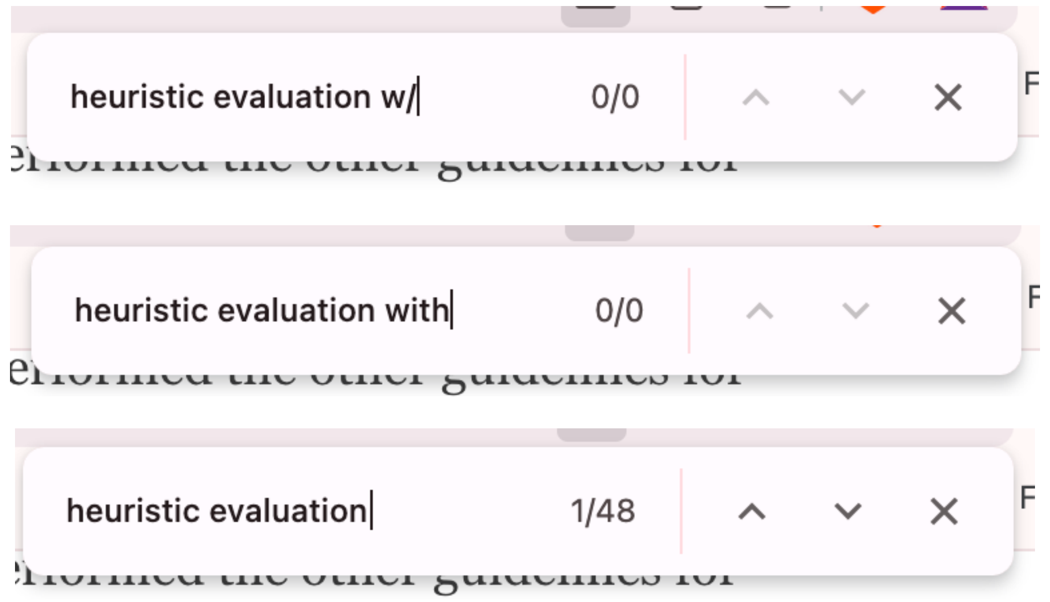
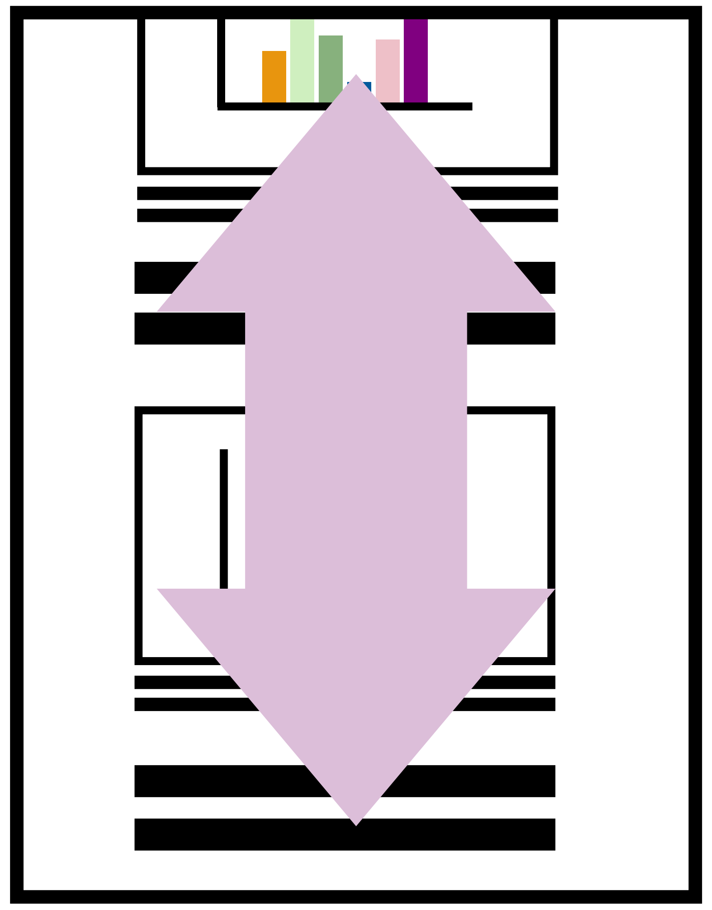


Figure 5: An illustration of the formats of the three studies. The Performance Study consists of 3 raters evaluating the accuracy and helpfulness of GPT-4-generated suggestions for 51 UI mockups. The Heuristic Evaluation Study with Human Experts consists of 12 design experts, who each looked for guideline violations in 6 UIs, and finishes with an interview asking them to compare their violations with those found by the LLM. Finally, the Iterative Usage study comprises of another group of 12 design experts, each working with 3 UI mockups. For each mockup, the expert iteratively revises the design based on the LLM's valid suggestions and rates the LLM's feedback, going through 2-3 rounds of this per UI. The Usage study concludes with an interview about the expert's experience with the tool.

To explore the potential of GPT-4 in automating heuristic evaluation, we carried out three studies (see Figure 5).

Reading challenges



Searching

4 STUDY METHOD

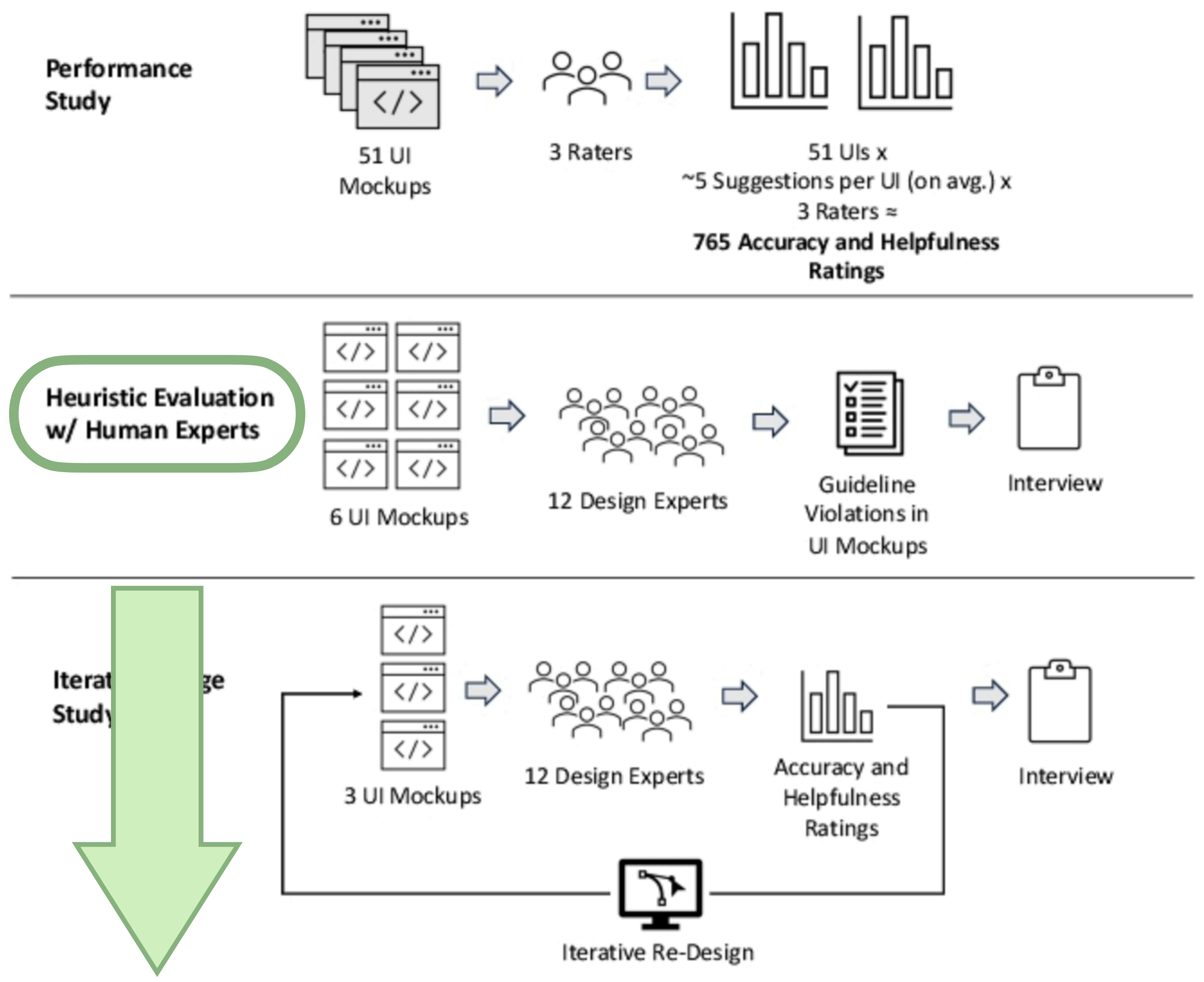
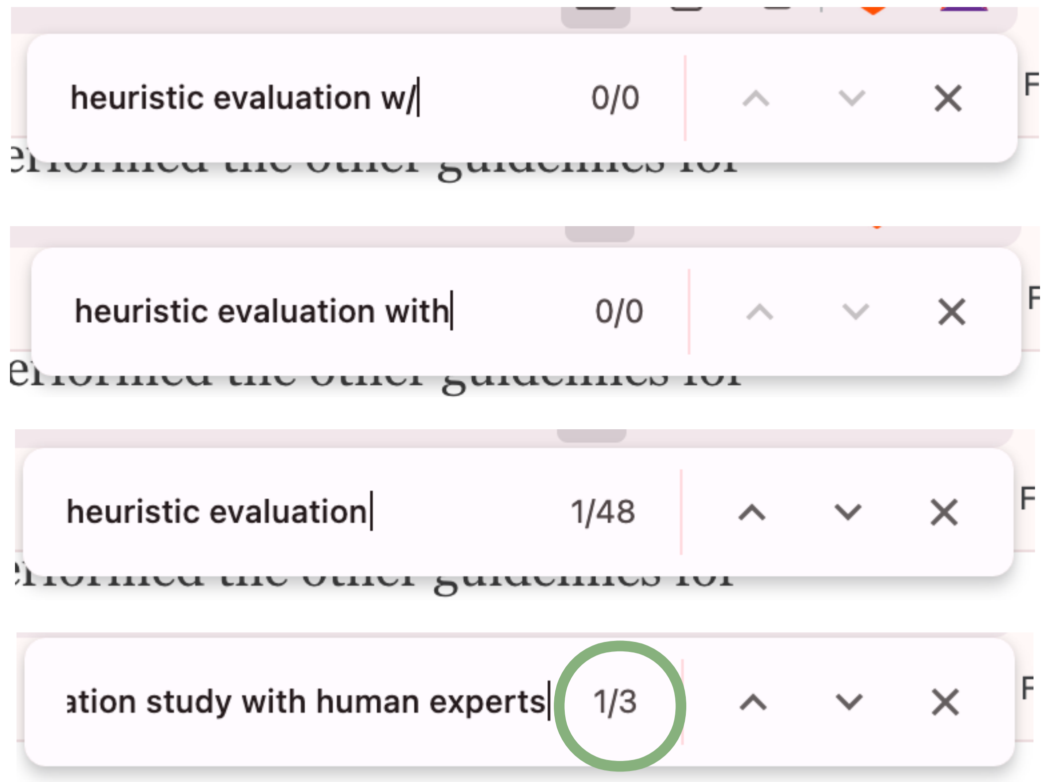
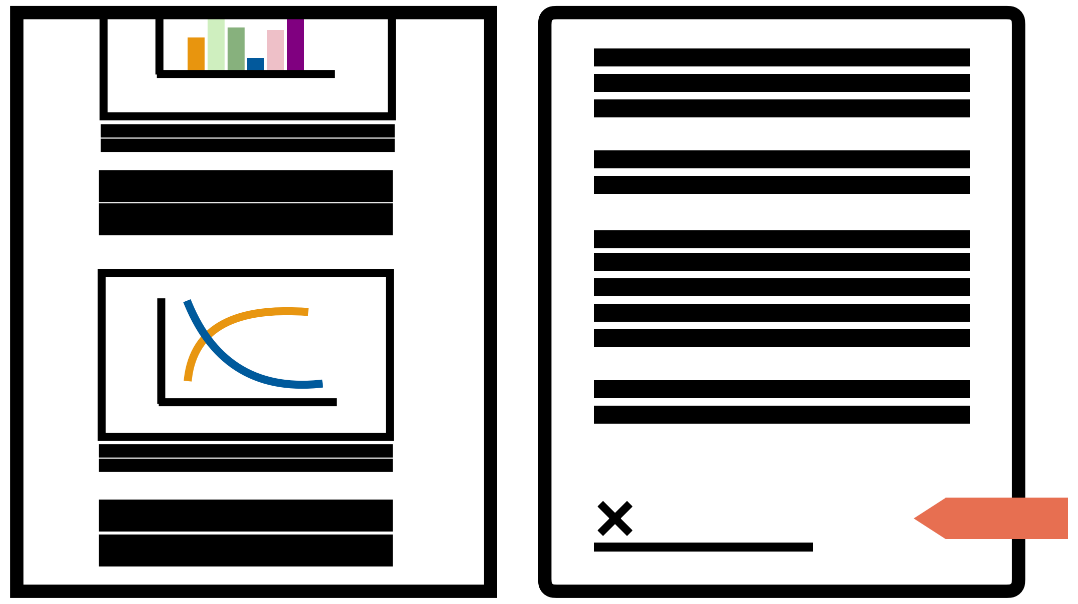


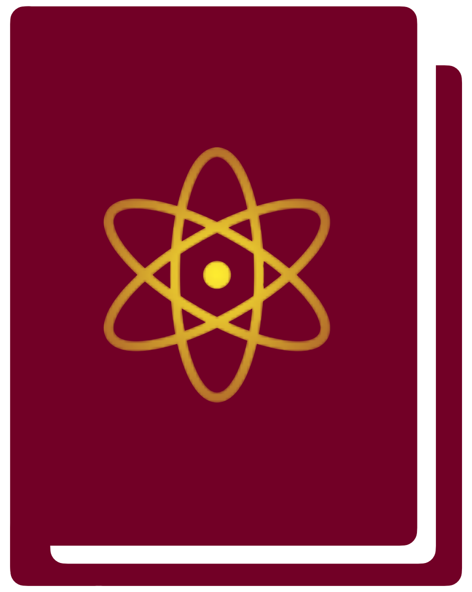
Figure 5: An illustration of the formats of the three studies. The Performance Study consists of 3 raters evaluating the accuracy and helpfulness of GPT-4-generated suggestions for 51 UI mockups. The Heuristic Evaluation Study with Human Experts consists of 12 design experts, who each looked for guideline violations in 6 UIs, and finishes with an interview asking them to compare their violations with those found by the LLM. Finally, the Iterative Usage study comprises of another group of 12 design experts, each working with 3 UI mockups. For each mockup, the expert iteratively revises the design based on the LLM's valid suggestions and rates the LLM's feedback, going through 2-3 rounds of this per UI. The Usage study concludes with an interview about the expert's experience with the tool.

To explore the potential of GPT-4 in automating heuristic evaluation, we carried out three studies (see Figure 5).



Basic units of the framework

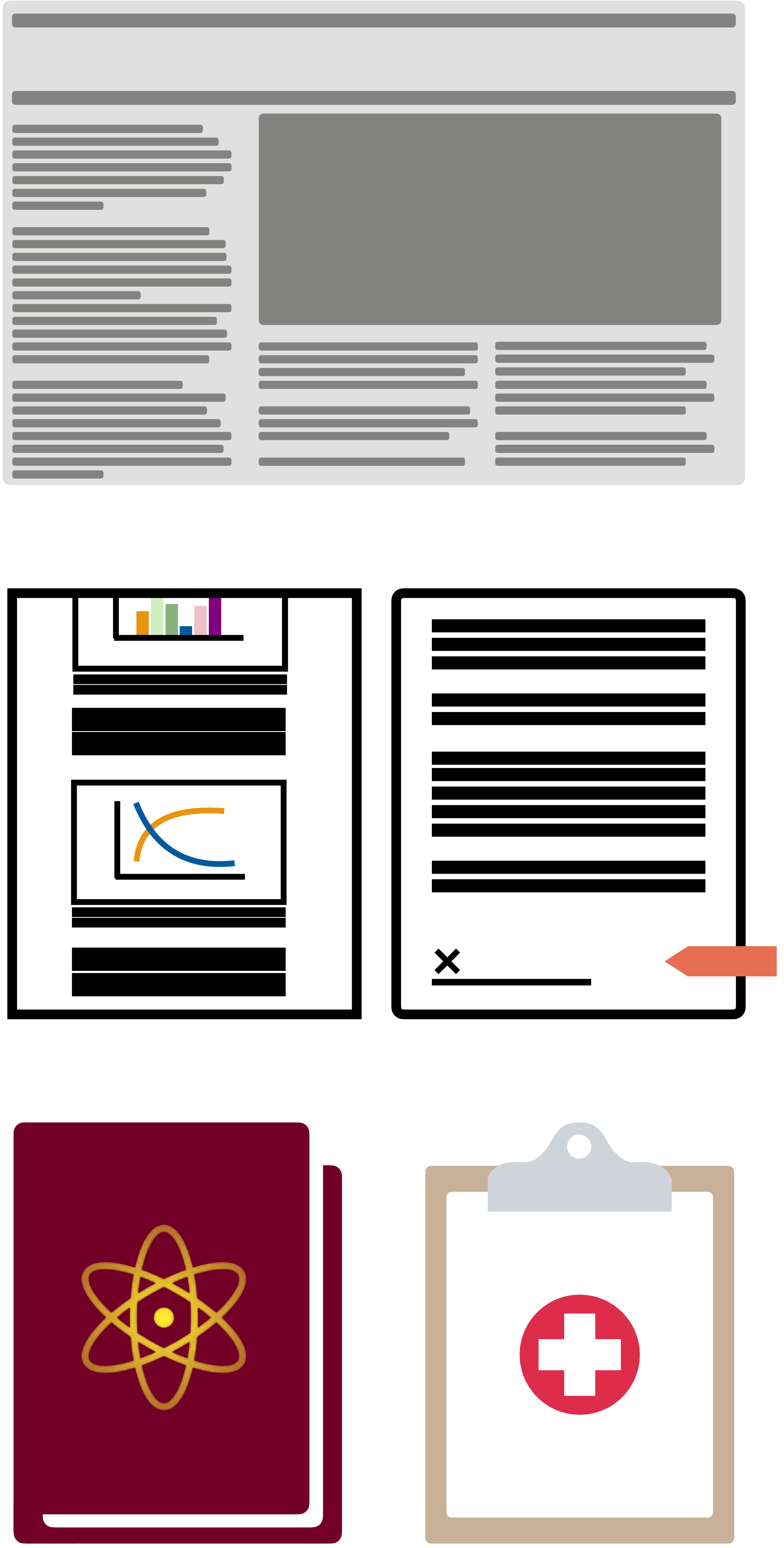




entities: discrete, interpretable, semantically meaningful items within a document

e.g., object in a photo, jargon, data in a chart or table, element in a diagram, claim in a passage

Basic units of the framework



links: relationships between entities

e.g., supporting evidence, definition, contradiction, similar ideas, elaboration

Basic units of the framework

Now that we have an abstract framework, we need a concrete example.

3. Instantiation

Goal: develop a concrete version of the framework

Goal: develop a concrete version of the framework

Method: implement augmentations for a research paper

Goal: develop a concrete version of the framework

Method: implement augmentations for a research paper

Outcome: fully functional augmented reading interface

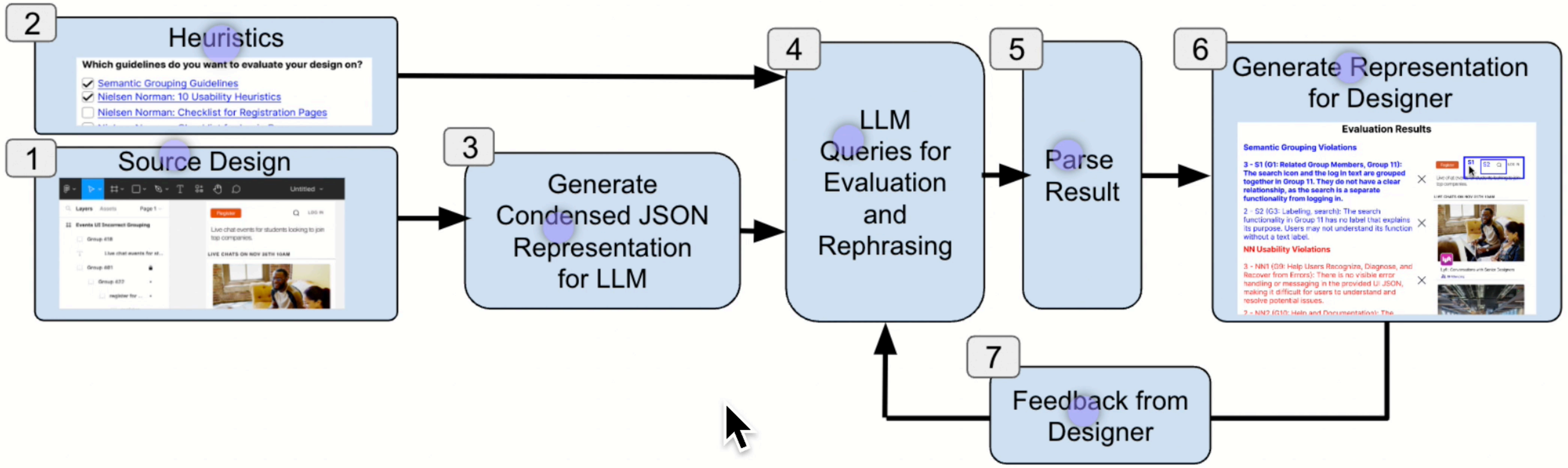
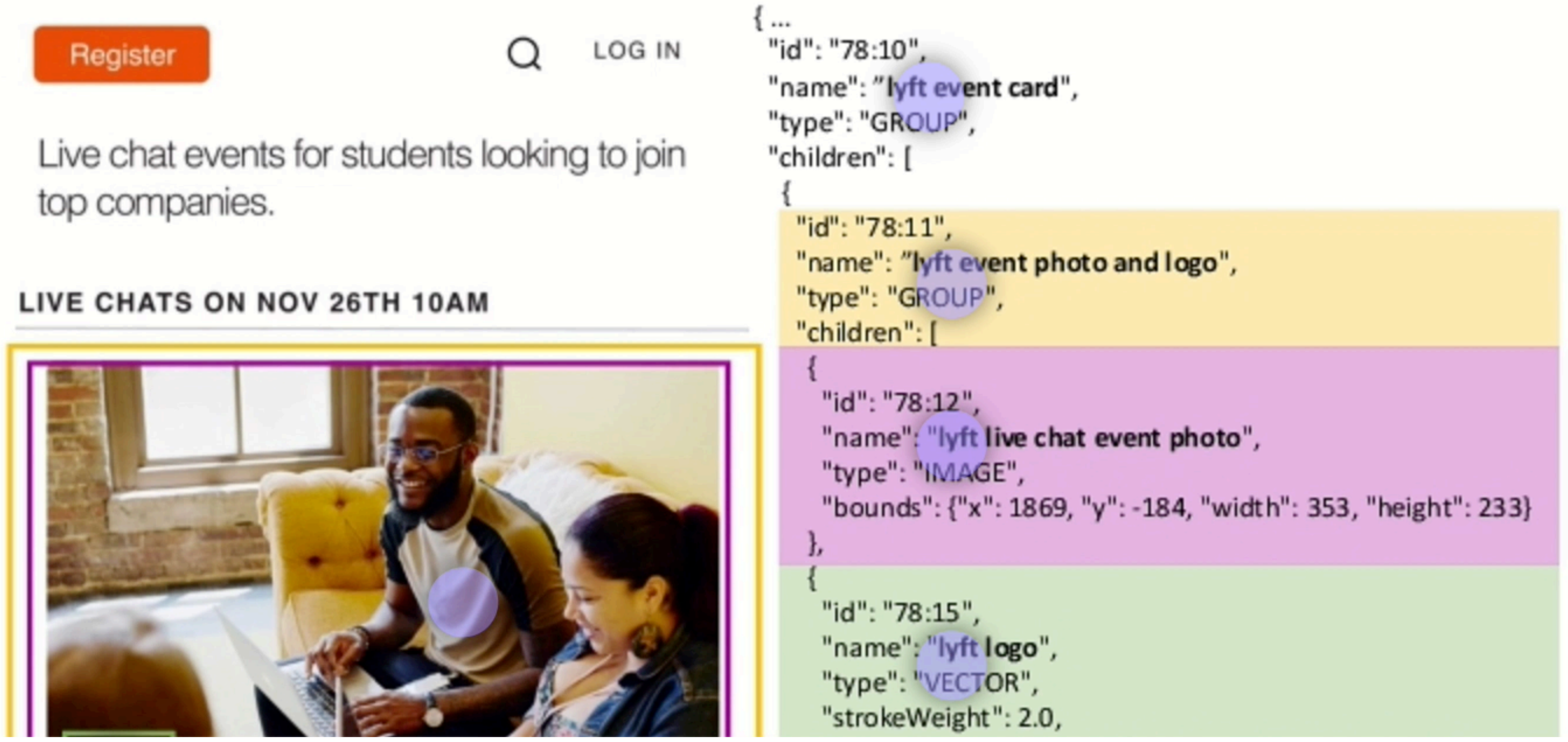
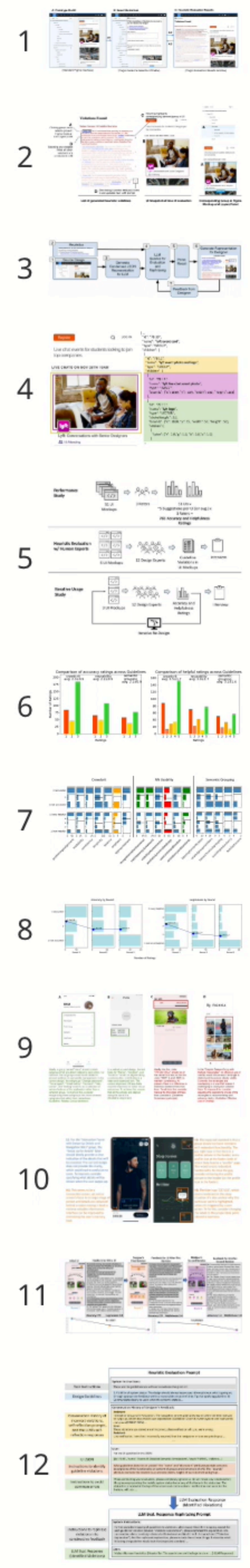


Figure 3: Our LLM-based plugin system architecture. The designer prototypes a UI in Figma (Box 1), and the plugin generates a UI representation to send to an LLM (3). The designer also selects heuristics/guidelines to use for evaluating the prototype (2), and a prompt containing the UI representation (in JSON) and guidelines is created and sent to the LLM (4). After identifying all the guideline violations, another LLM query is made to rephrase the guideline violations into constructive design advice (4). The LLM response is then programmatically parsed (5), and the plugin produces an interpretable representation of the response to display (6). The designer dismisses incorrect suggestions, which are incorporated in the LLM prompt for the next round of evaluation, if there is room in the context window (7). Pop Out Figure Scan





AI pipeline

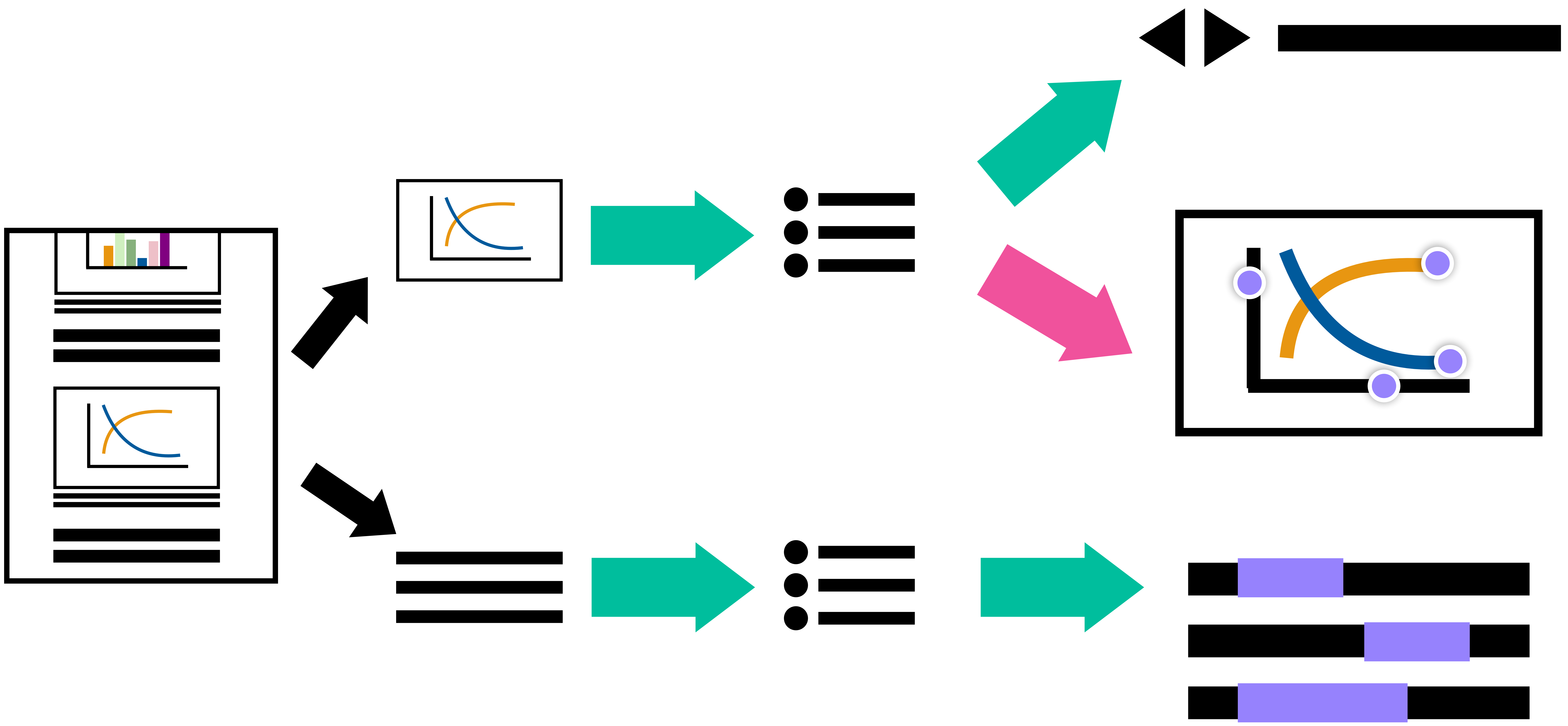
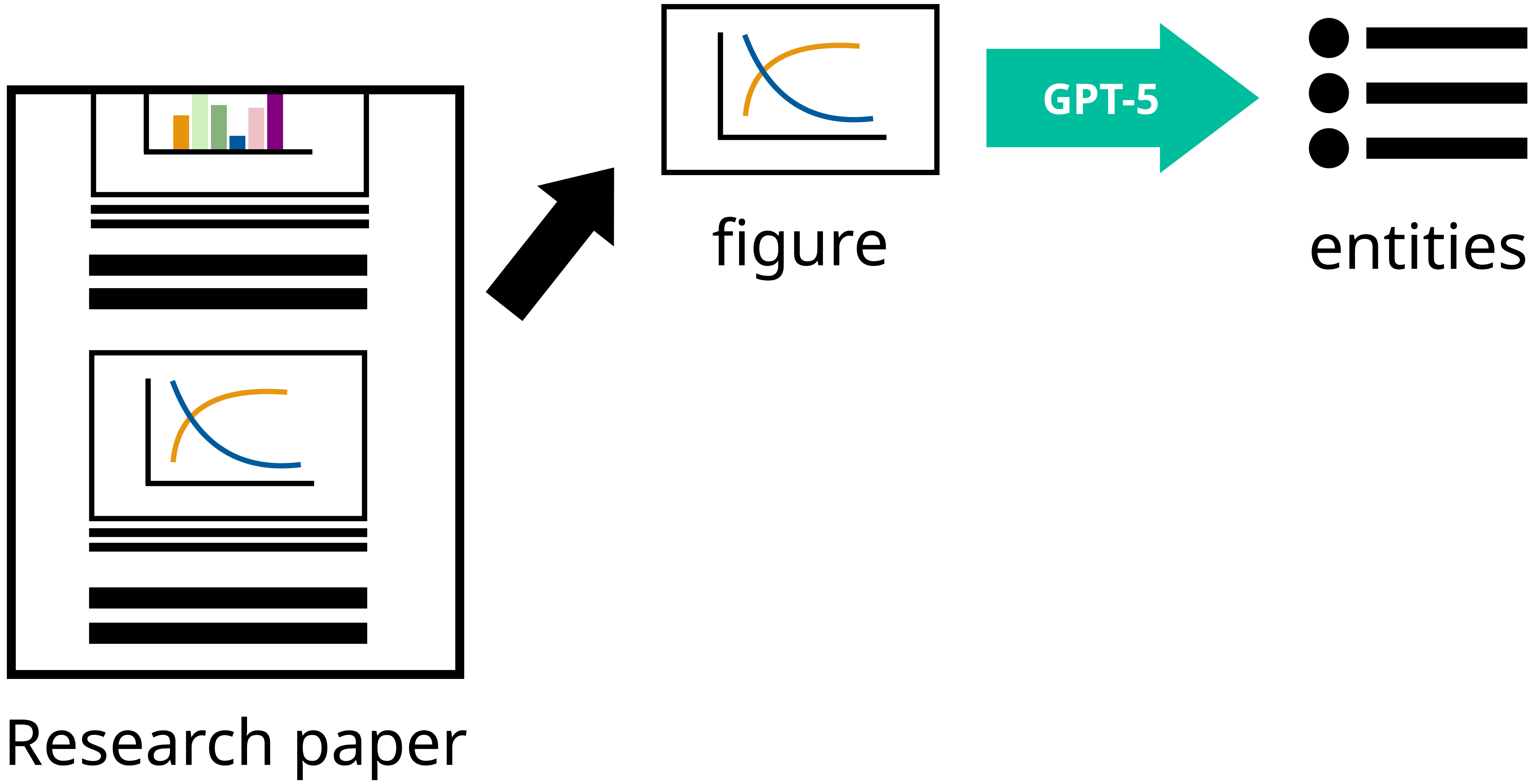
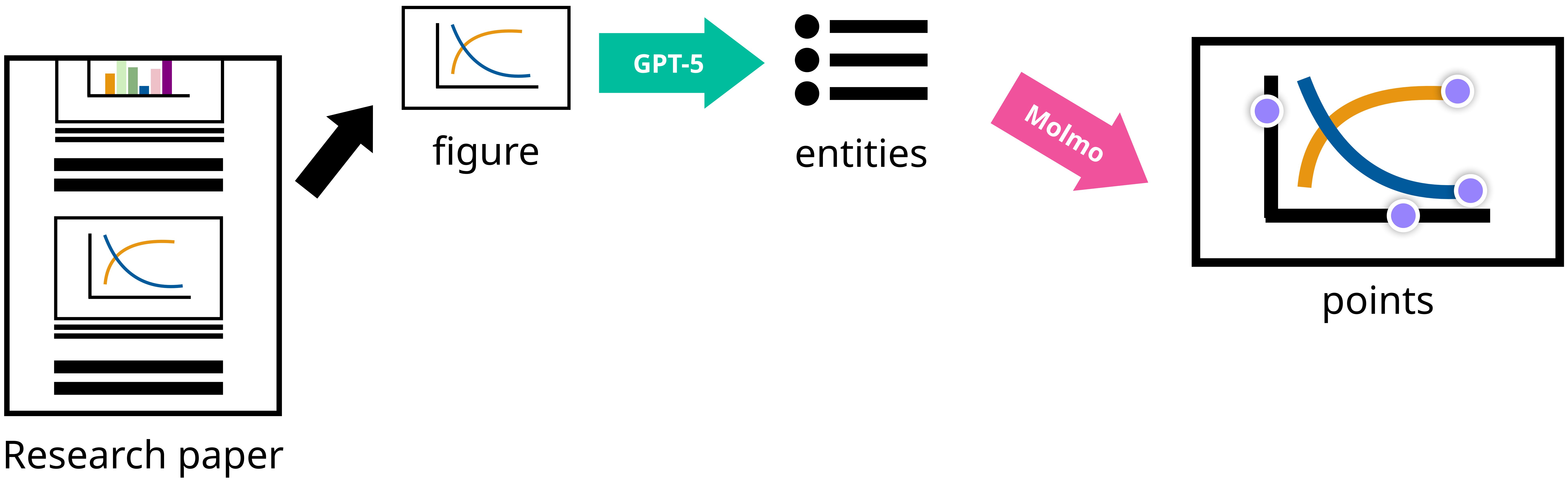


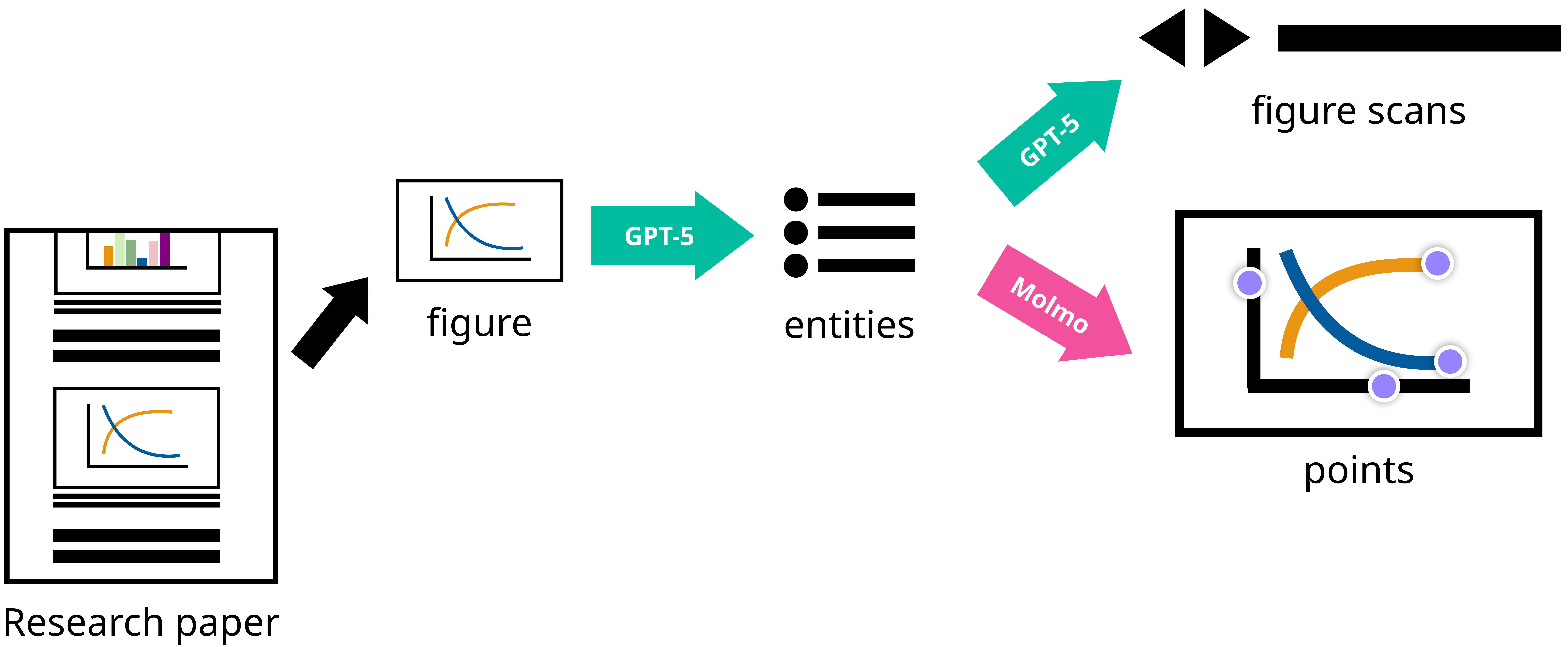
Figure entity extraction with GPT-5



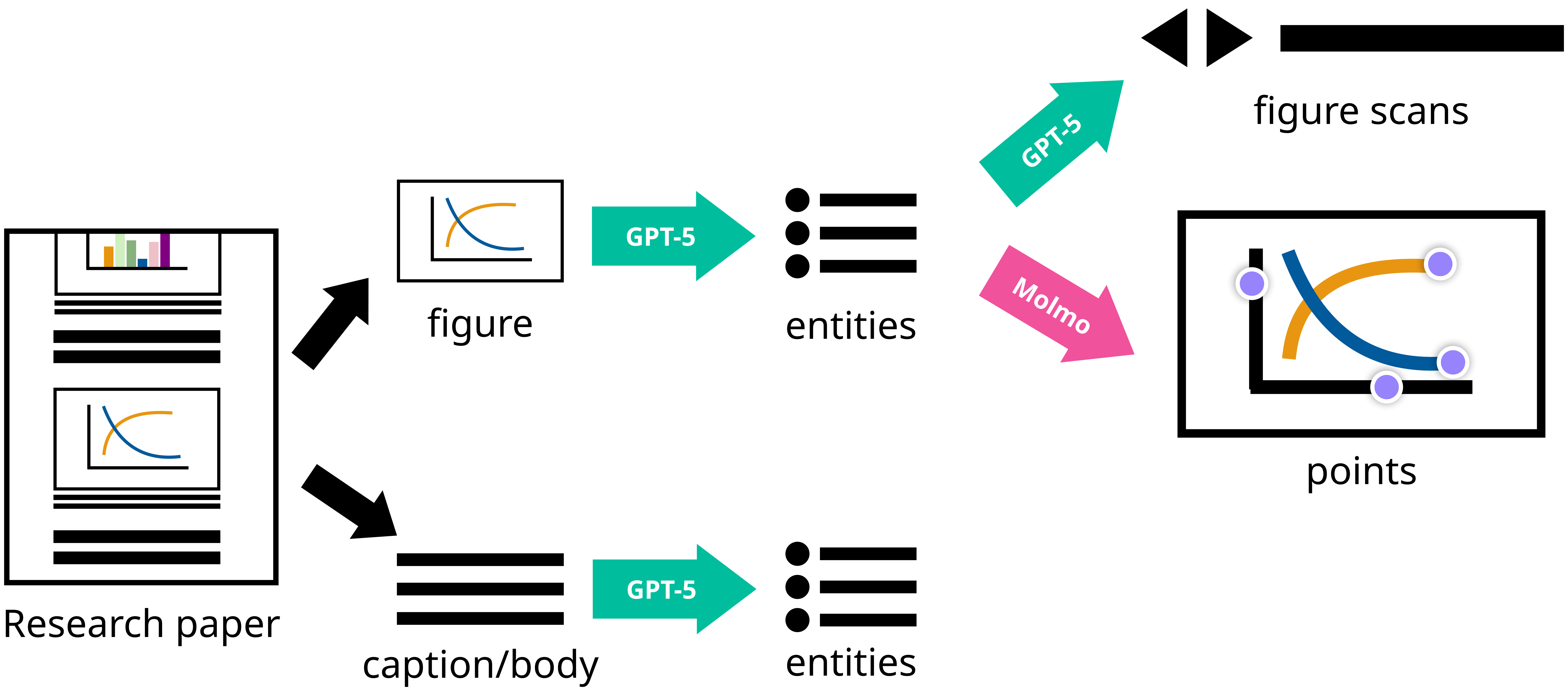
Coordinate identification with Molmo



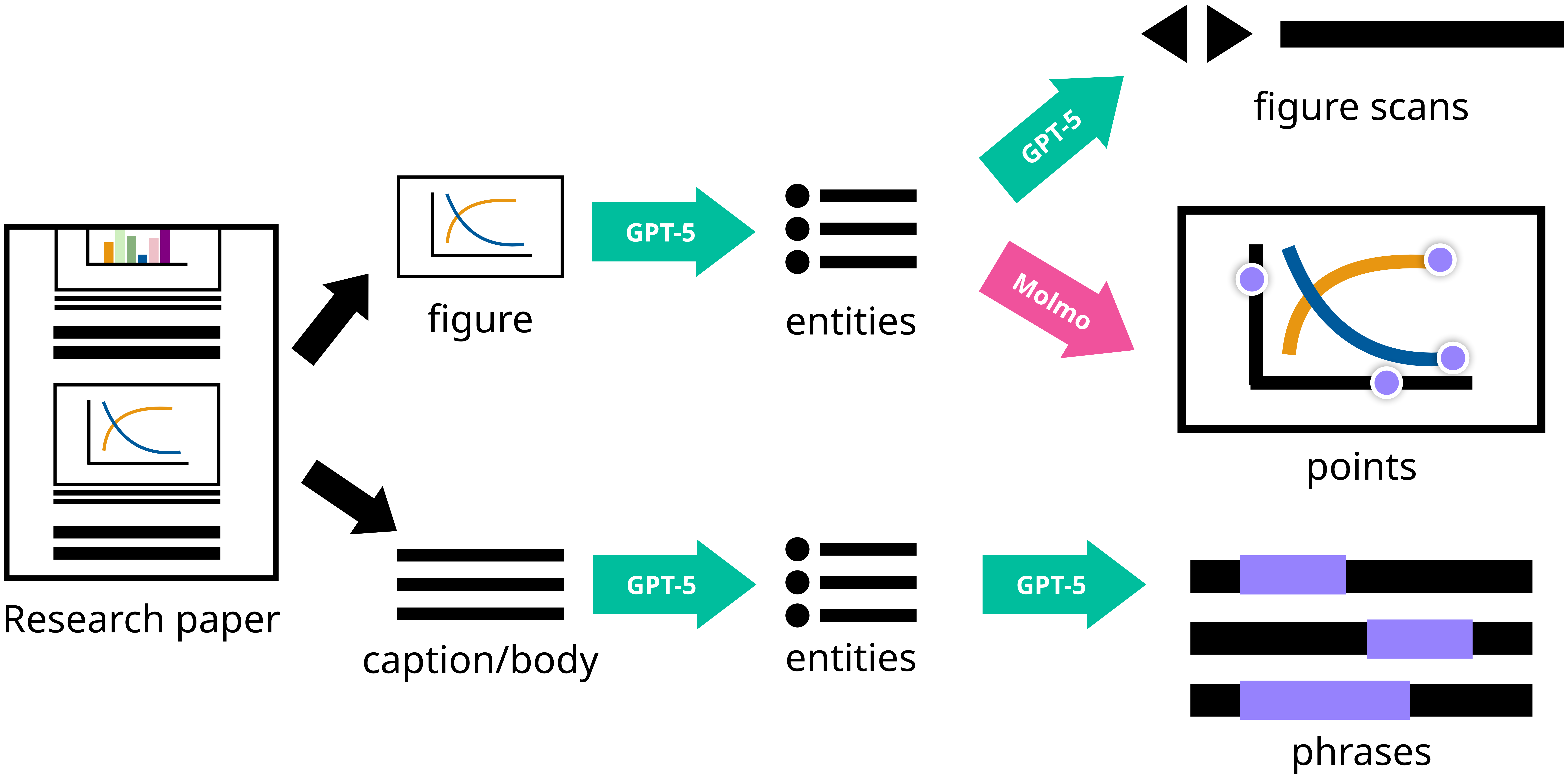
Description generation with GPT-5



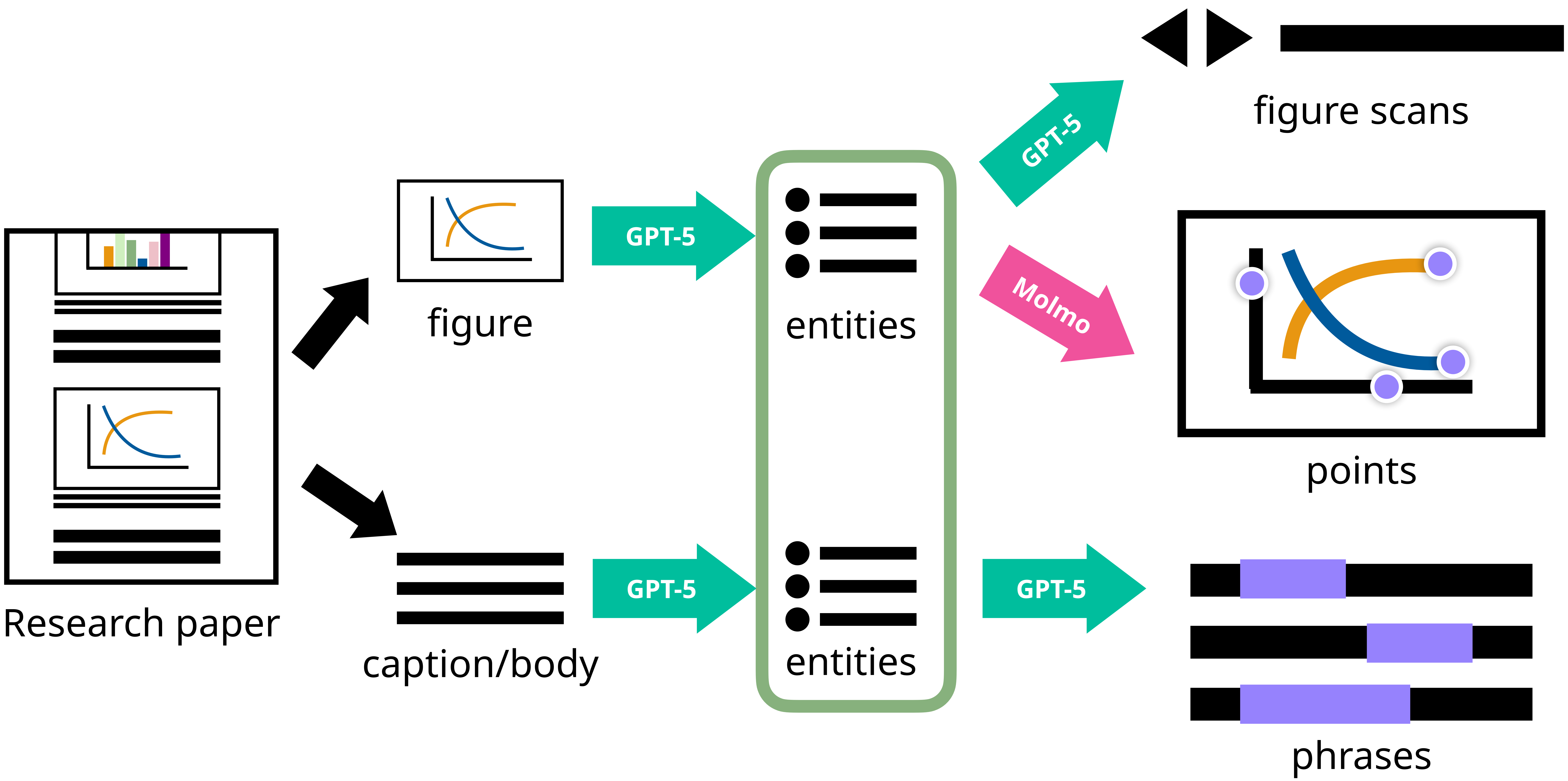
Phrase entity extraction with GPT-5



Phrase entity extraction with GPT-5



Links through **matching entities**



Links through **matching entities**

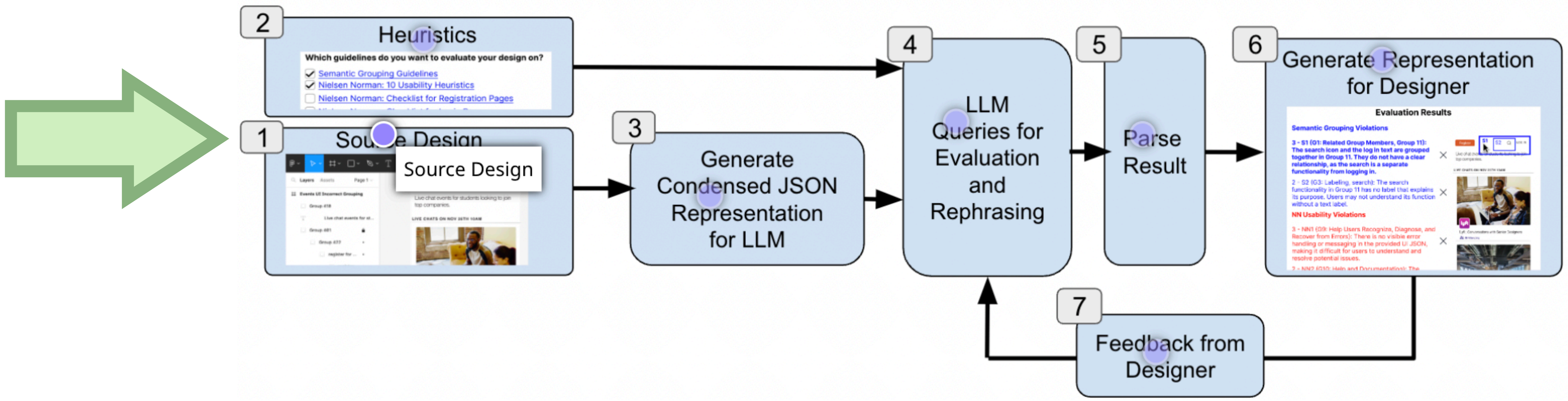
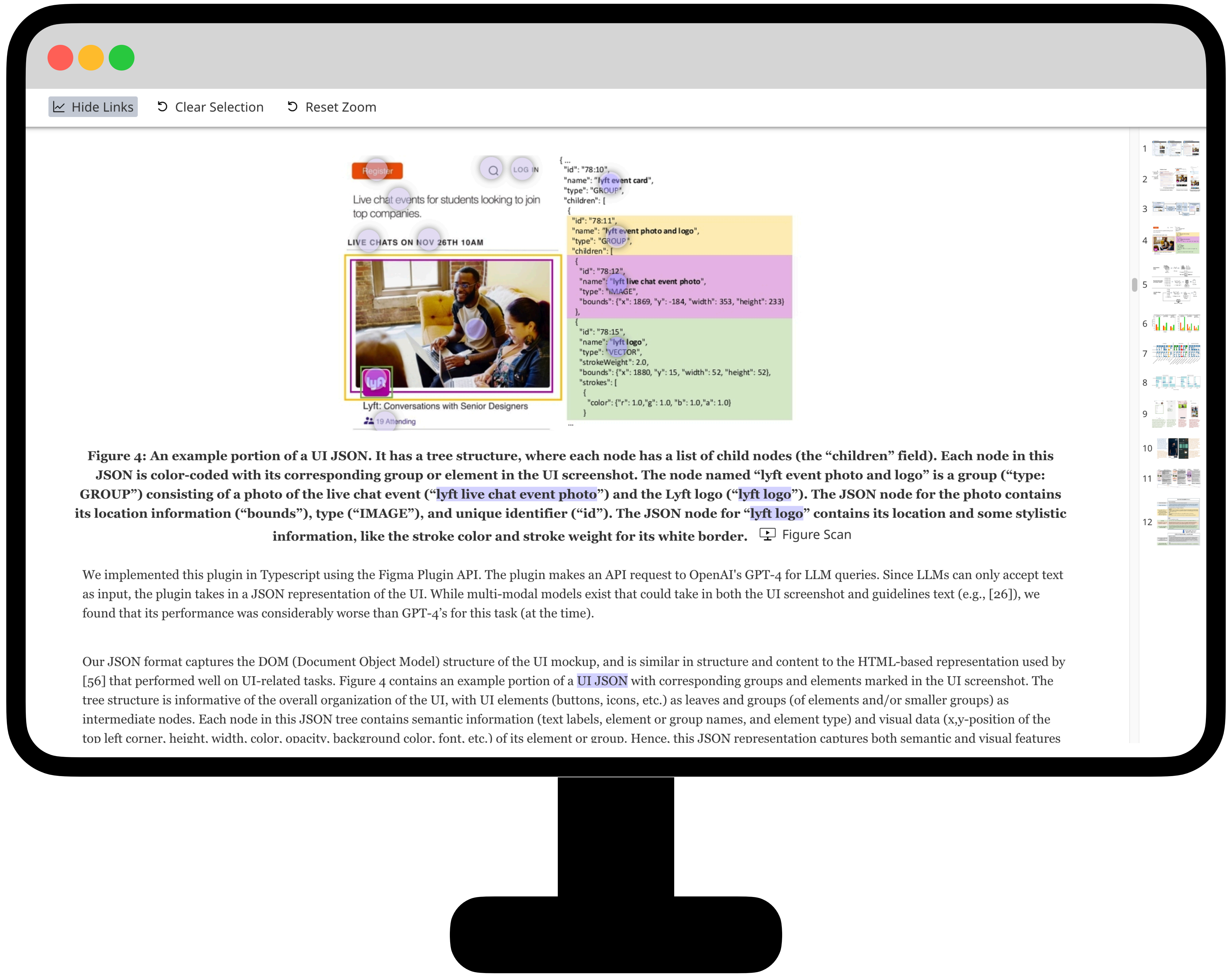


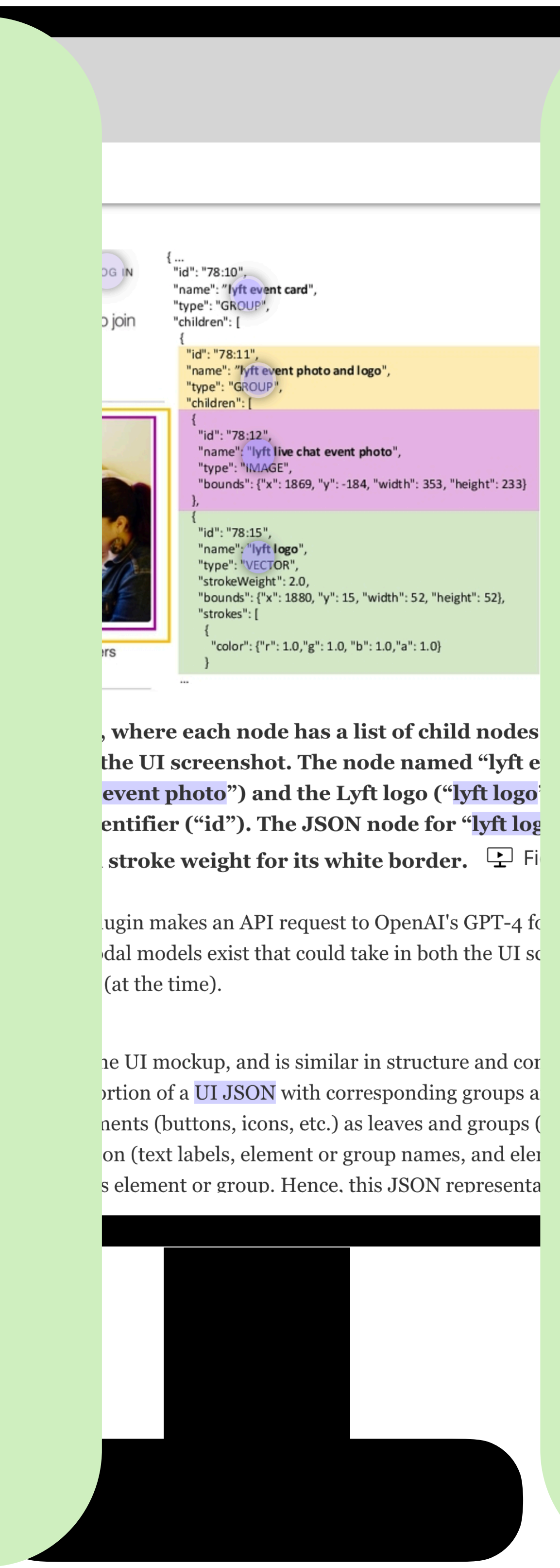
Figure 3: Our LLM-based plugin system architecture. The designer prototypes a UI in Figma (Box 1), a generates a UI representation to send to an LLM (3). The designer also selects heuristics/guidelines to use for valuating the prototype (2), and a prompt containing the UI representation (in JSON) and guidelines is created and sent to the LLM (4). After identifying all the guideline violations, another LLM query is made to rephrase the guideline violations into constructive design advice (4). The LLM response is then programmatically parsed (5), and the plugin produces an interpretable representation of the response to display (6). The designer dismisses incorrect suggestions, which are incorporated in the LLM prompt for the next round of evaluation, if there is room in the context window (7).

phrases


Hide Links
Clear Selection
Reset Zoom
Live chat events for students looking to join top companies.
LIVE CHATS ON NOV 26TH 10AM
Lyft: Conversations with Senior Designers
19 Attending
Figure 4: An example portion of a UI JSON. It has a tree structure, where each node has a list of child nodes (the “children” field). Each node in this JSON is color-coded with its corresponding group or element in the UI screenshot. The node named “lyft event photo and logo” is a group (“type: GROUP”) consisting of a photo of the live chat event (“lyft live chat event photo”) and the Lyft logo (“lyft logo”). The JSON node for the photo contains its location information (“bounds”), type (“IMAGE”), and unique identifier (“id”). The JSON node for “lyft logo” contains its location and some stylistic information, like the stroke color and stroke weight for its white border.
Figure Scan
We implemented this plugin in Typescript using the Figma Plugin API. The plugin makes an API request to OpenAI's GPT-4 for LLM queries. Since LLMs can only accept text as input, the plugin takes in a JSON representation of the UI. While multi-modal models exist that could take in both the UI screenshot and guidelines text (e.g., [26]), we found that its performance was considerably worse than GPT-4’s for this task (at the time).
Our JSON format captures the DOM (Document Object Model) structure of the UI mockup, and is similar in structure and content to the HTML-based representation used by [56] that performed well on UI-related tasks. Figure 4 contains an example portion of a UI JSON with corresponding groups and elements marked in the UI screenshot. The tree structure is informative of the overall organization of the UI, with UI elements (buttons, icons, etc.) as leaves and groups (of elements and/or smaller groups) as intermediate nodes. Each node in this JSON tree contains semantic information (text labels, element or group names, and element type) and visual data (x,y-position of the top left corner, height, width, color, opacity, background color, font, etc.) of its element or group. Hence, this JSON representation captures both semantic and visual features

2

How can we systematically represent connections between ideas to make them easier to find?



Links between entities. For papers: figure points and text highlights.

4. Evaluation

Goal: determine if the framework improves comprehension

Goal: determine if the framework improves comprehension

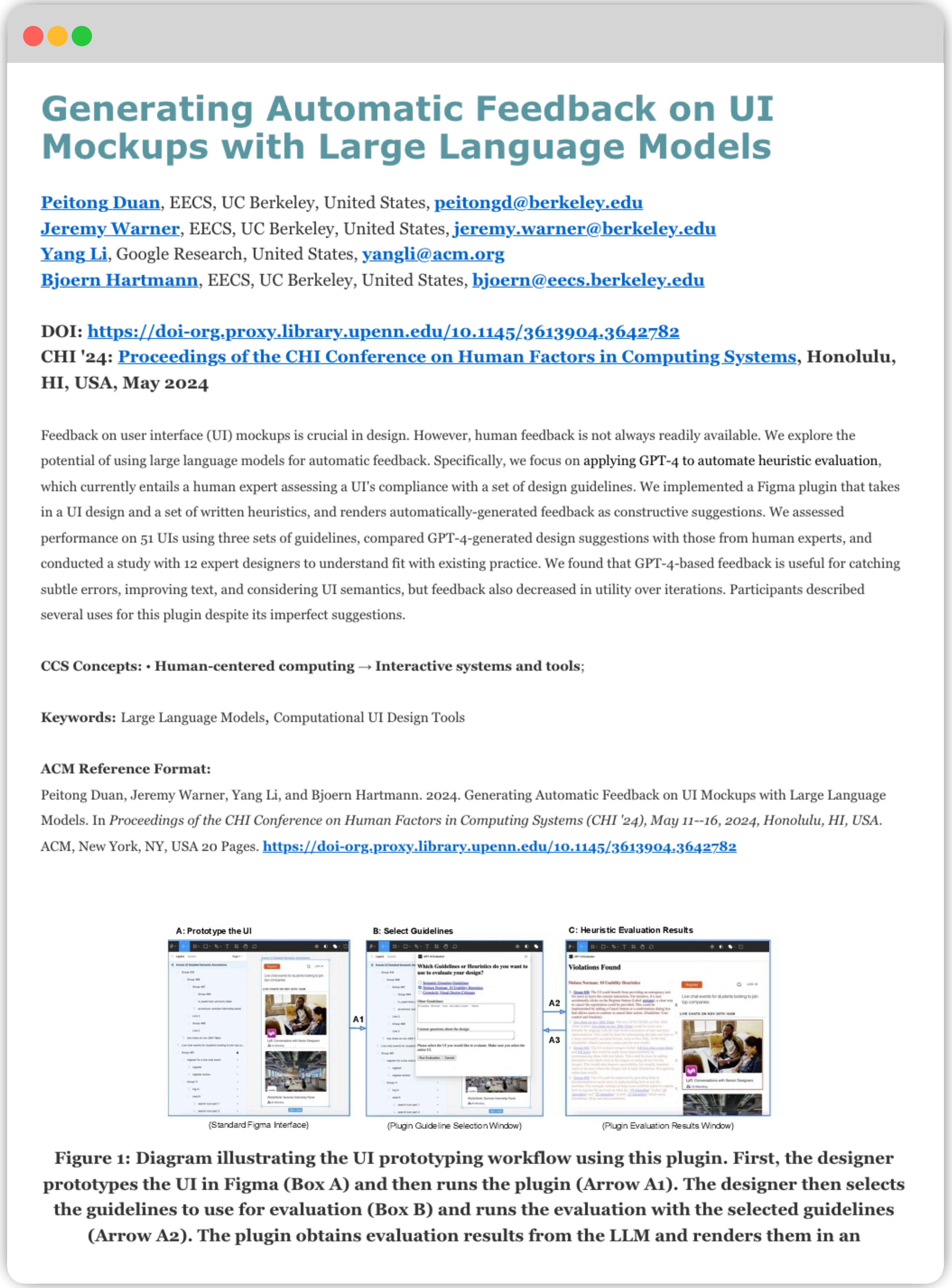
Method: between-subjects study with reading session and quiz

Goal: determine if the framework improves comprehension

Method: between-subjects study with reading session and quiz

Outcome: gains in accuracy without increase in time or cognitive load

Study design



Generating Automatic Feedback on UI Mockups with Large Language Models

Peitong Duan, EECS, UC Berkeley, United States, peitongd@berkeley.edu
Jeremy Warner, EECS, UC Berkeley, United States, jeremy.warner@berkeley.edu
Yang Li, Google Research, United States, yangli@acm.org
Bjoern Hartmann, EECS, UC Berkeley, United States, bjoern@eecs.berkeley.edu

DOI: https://doi-org.proxy.library.upenn.edu/10.1145/3613904.3642782
CHI '24: Proceedings of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, May 2024

Feedback on user interface (UI) mockups is crucial in design. However, human feedback is not always readily available. We explore the potential of using large language models for automatic feedback. Specifically, we focus on applying GPT-4 to automate heuristic evaluation, which currently entails a human expert assessing a UI's compliance with a set of design guidelines. We implemented a Figma plugin that takes in a UI design and a set of written heuristics, and renders automatically-generated feedback as constructive suggestions. We assessed performance on 51 UIs using three sets of guidelines, compared GPT-4-generated design suggestions with those from human experts, and conducted a study with 12 expert designers to understand fit with existing practice. We found that GPT-4-based feedback is useful for catching subtle errors, improving text, and considering UI semantics, but feedback also decreased in utility over iterations. Participants described several uses for this plugin despite its imperfect suggestions.

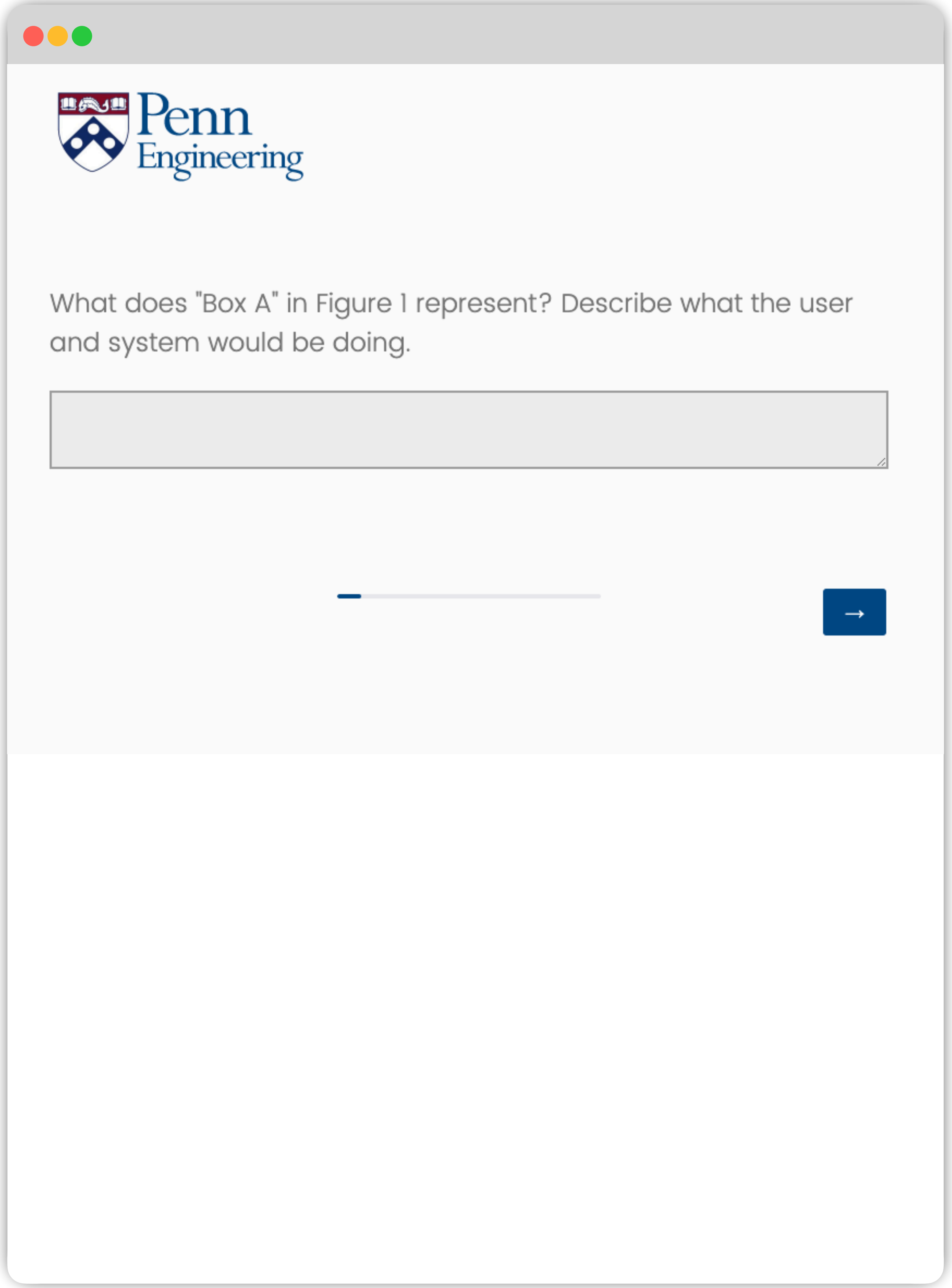
CCS Concepts: • **Human-centered computing** → **Interactive systems and tools;**

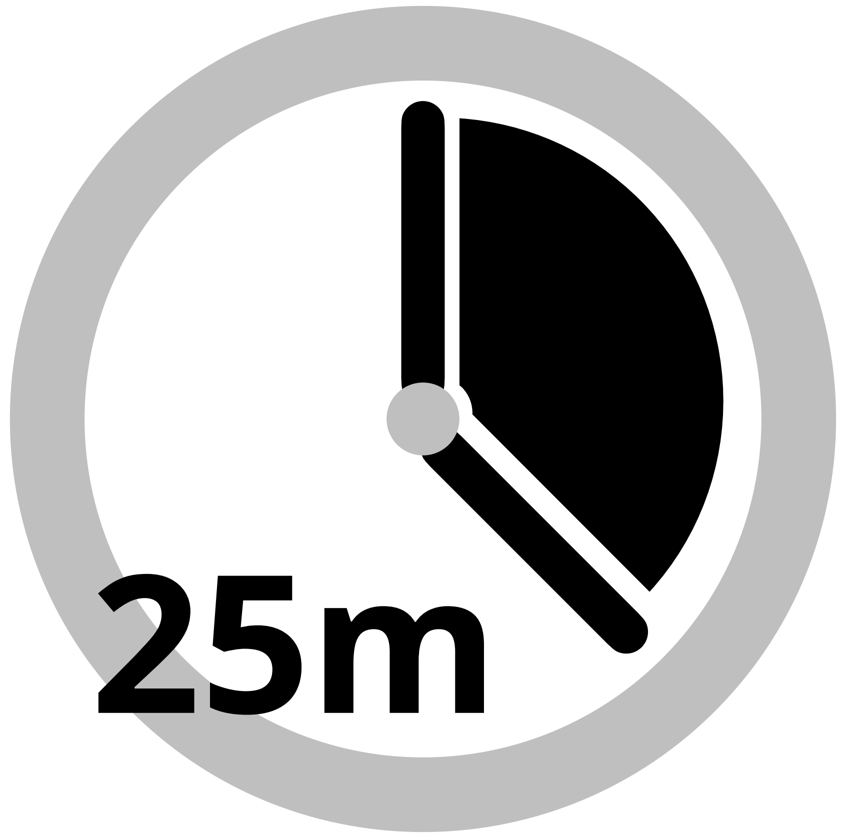
Keywords: Large Language Models, Computational UI Design Tools

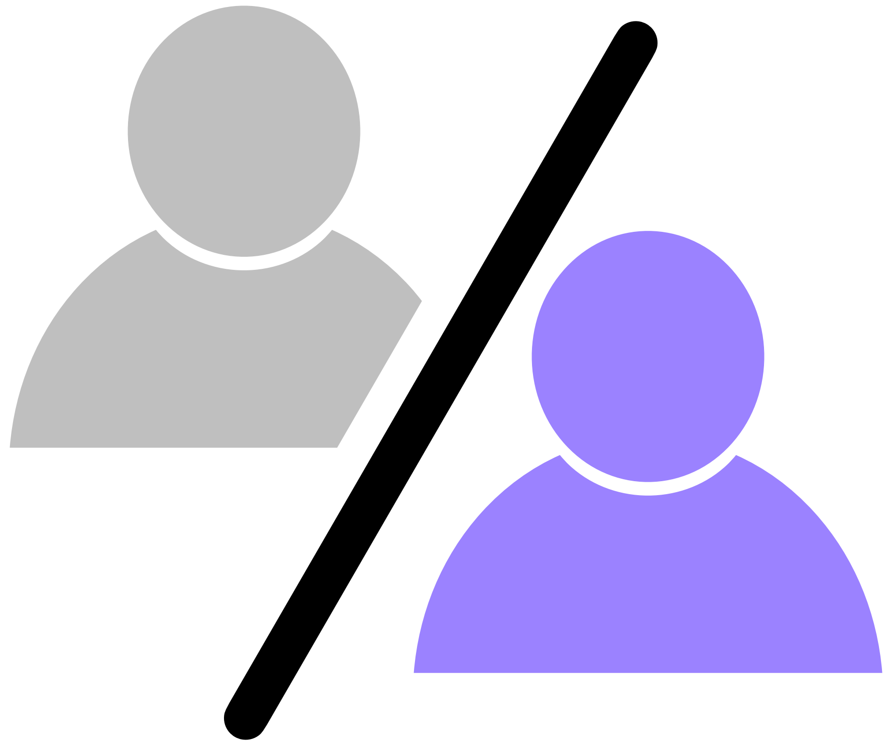
ACM Reference Format:
Peitong Duan, Jeremy Warner, Yang Li, and Bjoern Hartmann. 2024. Generating Automatic Feedback on UI Mockups with Large Language Models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11--16, 2024, Honolulu, HI, USA.* ACM, New York, NY, USA 20 Pages. https://doi-org.proxy.library.upenn.edu/10.1145/3613904.3642782

A: Prototype the UI | B: Select Guidelines | C: Heuristic Evaluation Results
(Standard Figma Interface) | (Plugin Guideline Selection Window) | (Plugin Evaluation Results Window)

Figure 1: Diagram illustrating the UI prototyping workflow using this plugin. First, the designer prototypes the UI in Figma (Box A) and then runs the plugin (Arrow A1). The designer then selects the guidelines to use for evaluation (Box B) and runs the evaluation with the selected guidelines (Arrow A2). The plugin obtains evaluation results from the LLM and renders them in an







between-subjects

Participants

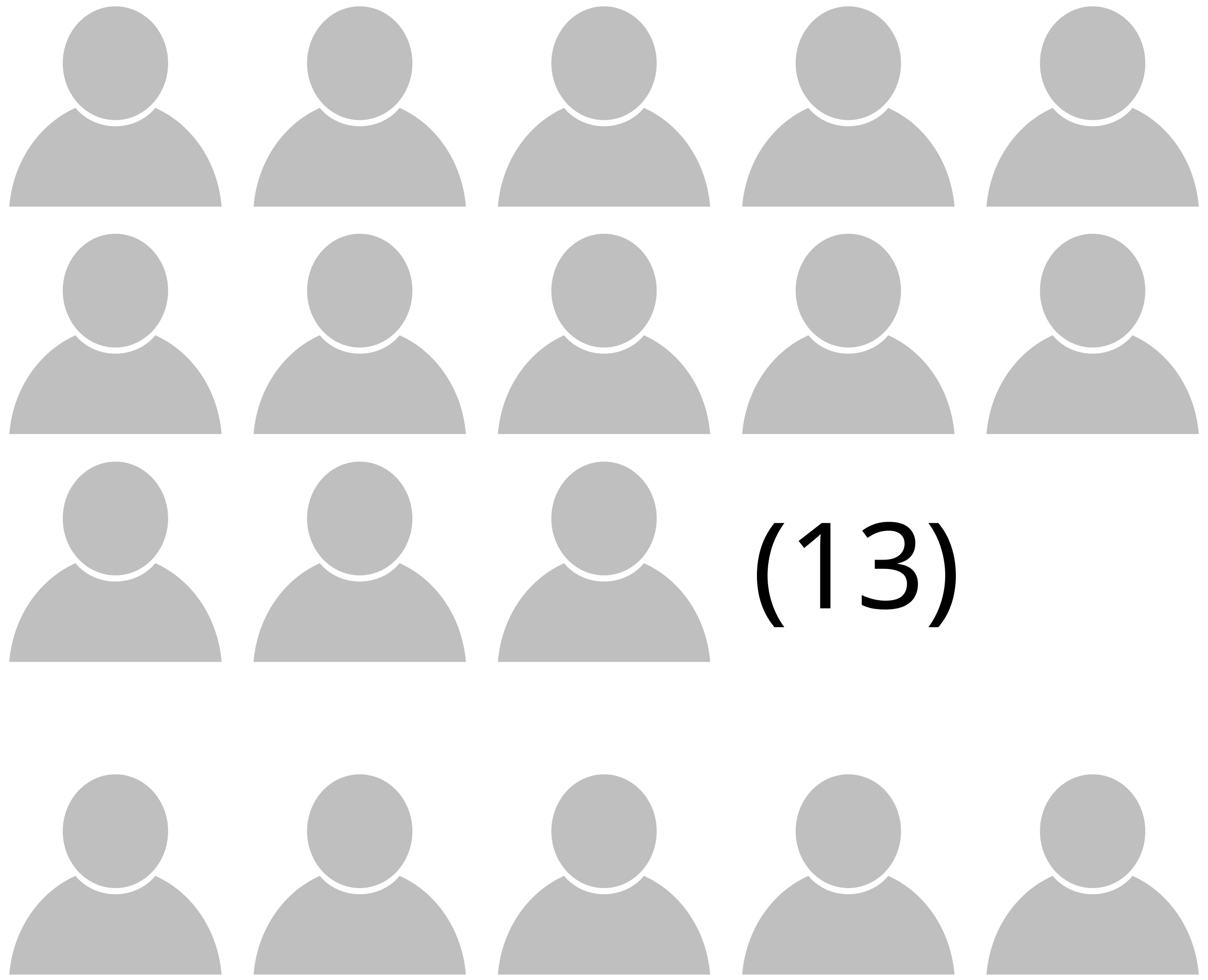
18 official (38 pilot)

Participants

18 official (38 pilot)
Academic year

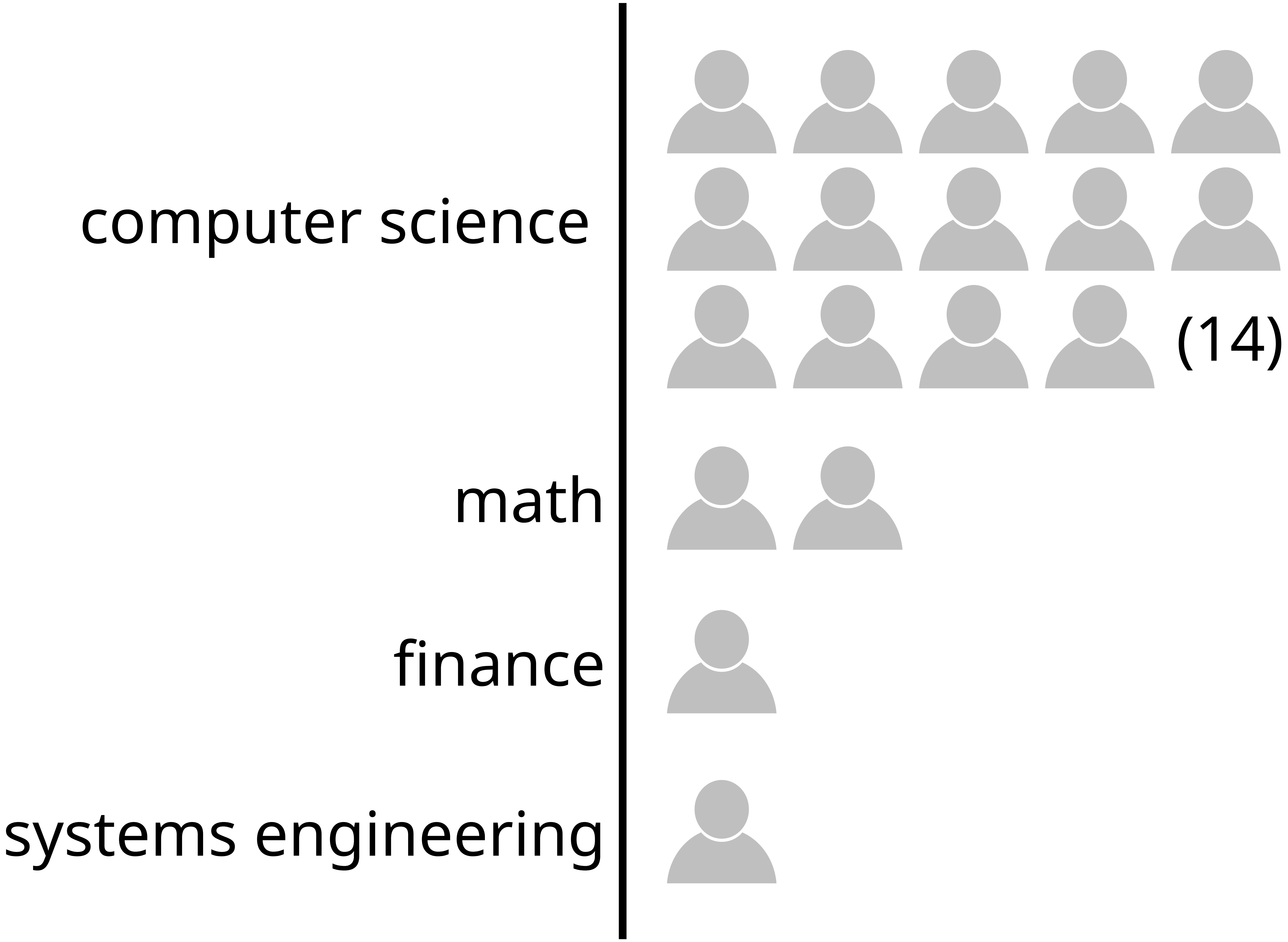
senior undergrad

accelerated master's



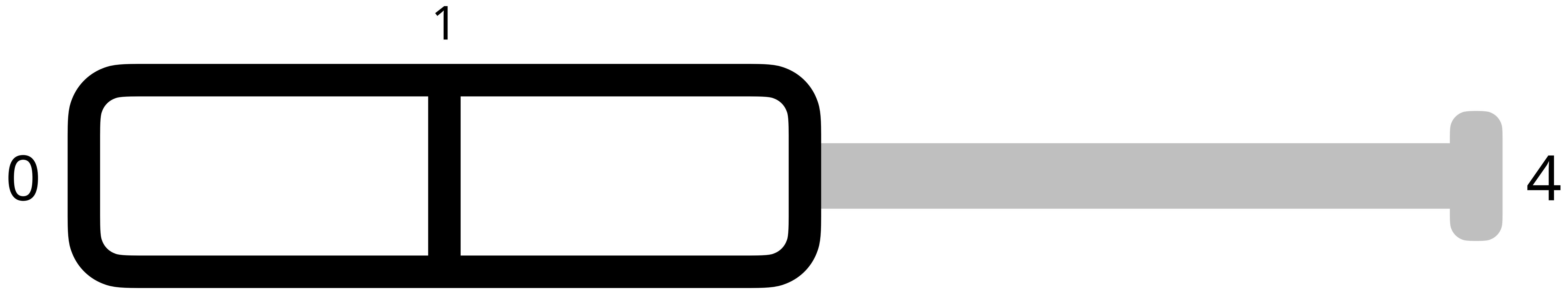
Participants

18 official (38 pilot)
Majors



Participants

18 official (38 pilot)



Years of research experience

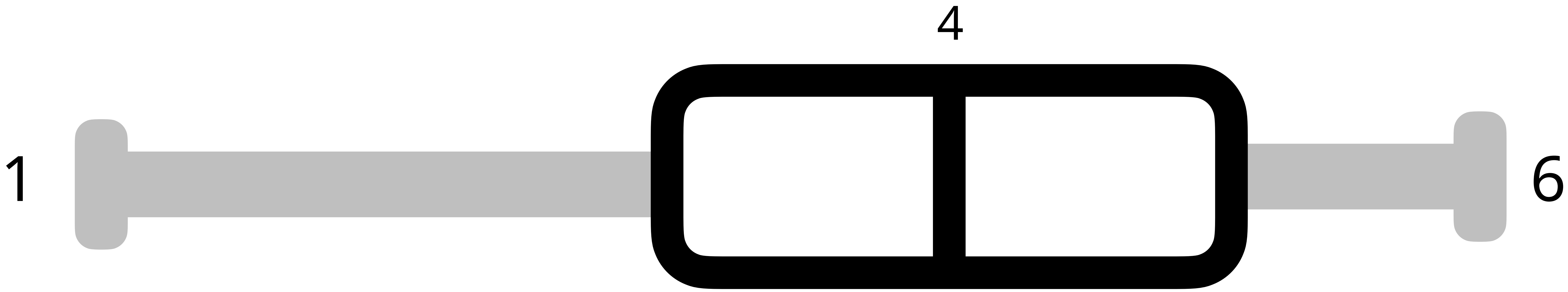
Participants

18 official (38 pilot)

Level of comfort reading research papers

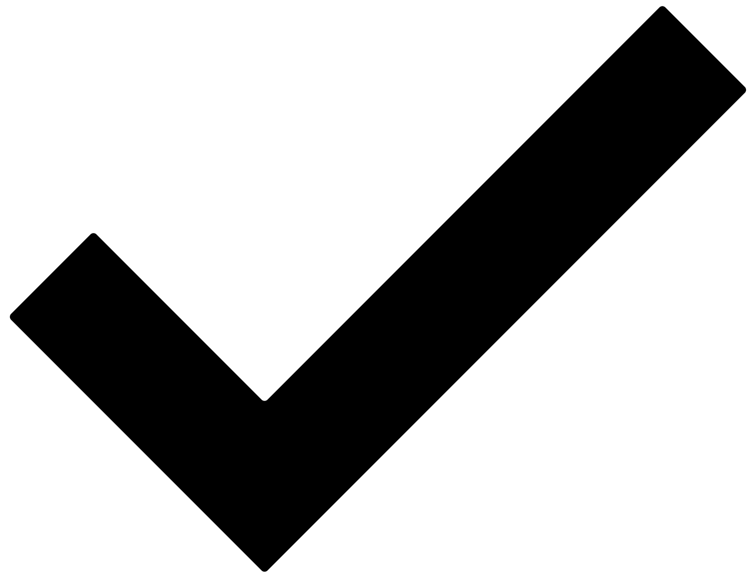
Participants

18 official (38 pilot)

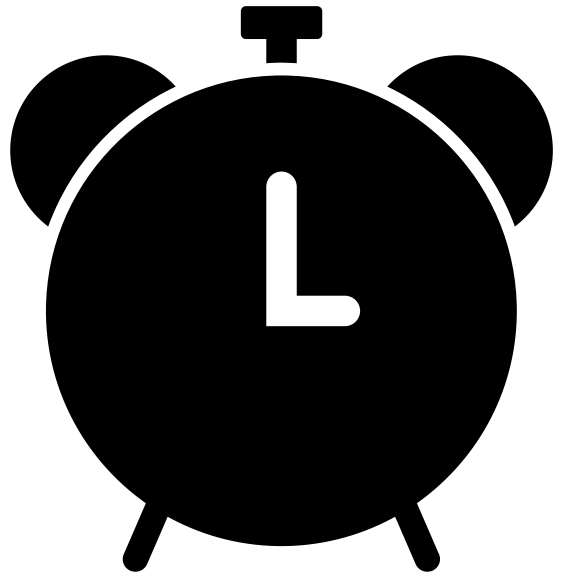


Familiarity with field of stimulus paper

Data collected

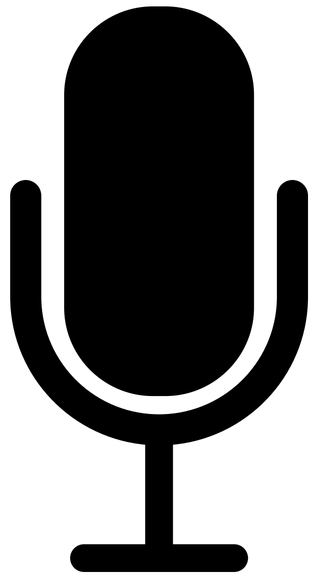


Quiz scores



Time to completion

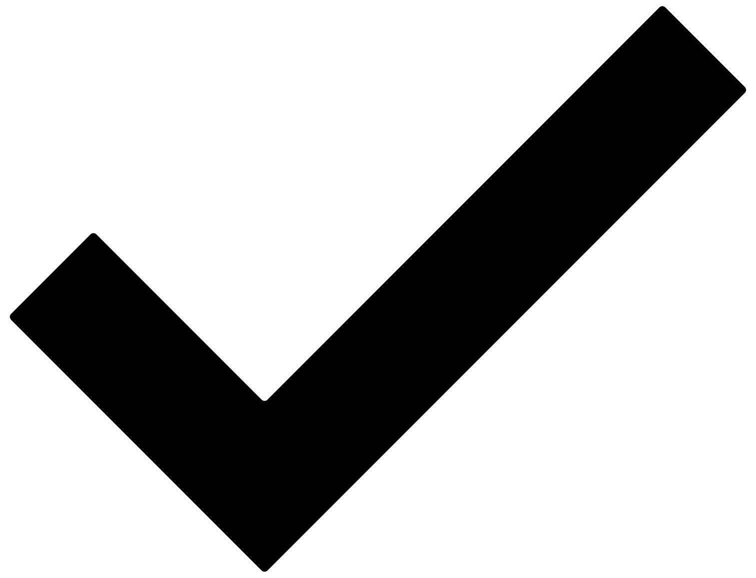
NASA TLX



Interviews

Screen recordings

Data collected



Quiz scores

Time to completion

NASA TLX

Interviews

Screen recordings

Evaluated with a rubric by 2 external annotators (Krippendorff’s α = 0.75)

Data collected

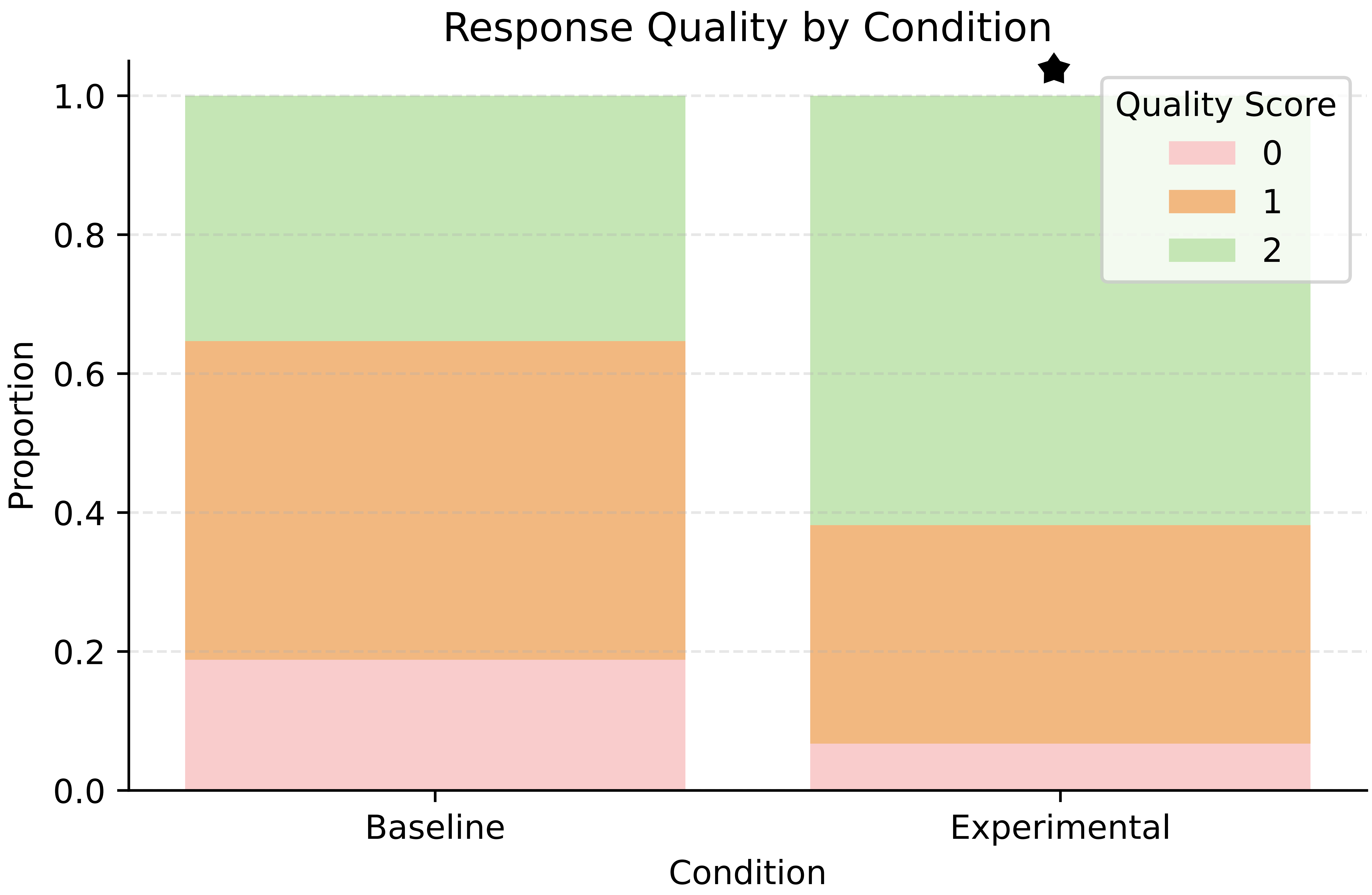
Quiz scor

Screen
recordings

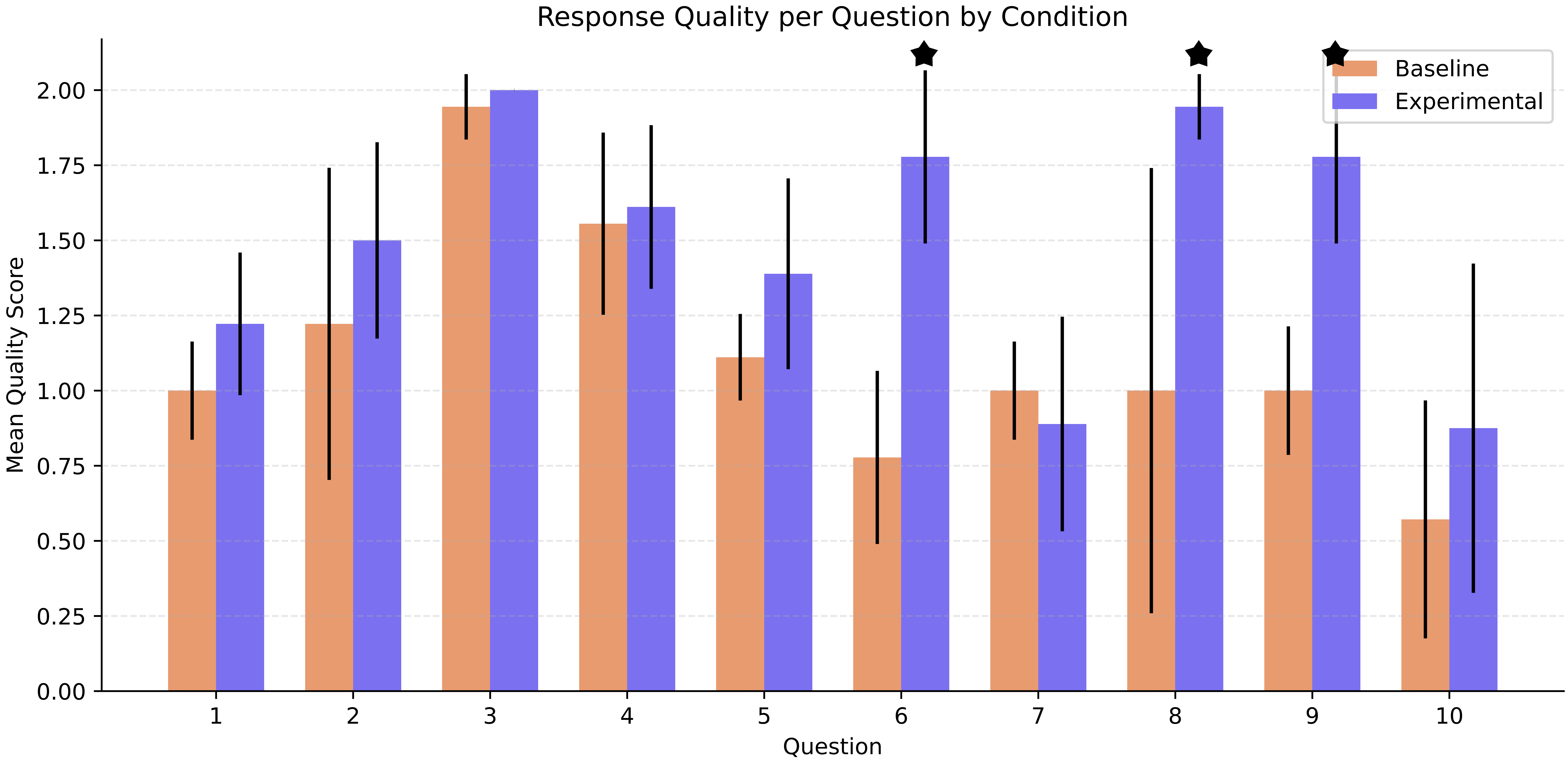
In the final section, we will analyze the data and discuss its implications on our framework.

4. Findings

Significant improvement in response quality



Significant improvement in response quality



CCS Concepts: • **Human-centered computing** → **Interactive systems and tools**;

Keywords: Large Language Models, Computational UI Design Tools

ACM Reference Format:

Peitong Duan, Jeremy Warner, Yang Li, and Bjoern Hartmann. 2024. Generating Automatic Feedback on UI Mockups with Large Language Models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11--16, 2024, Honolulu, HI, USA*. ACM, New York, NY, USA 20 Pages. https://doi-org.proxy.library.upenn.edu/10.1145/3613904.3642782

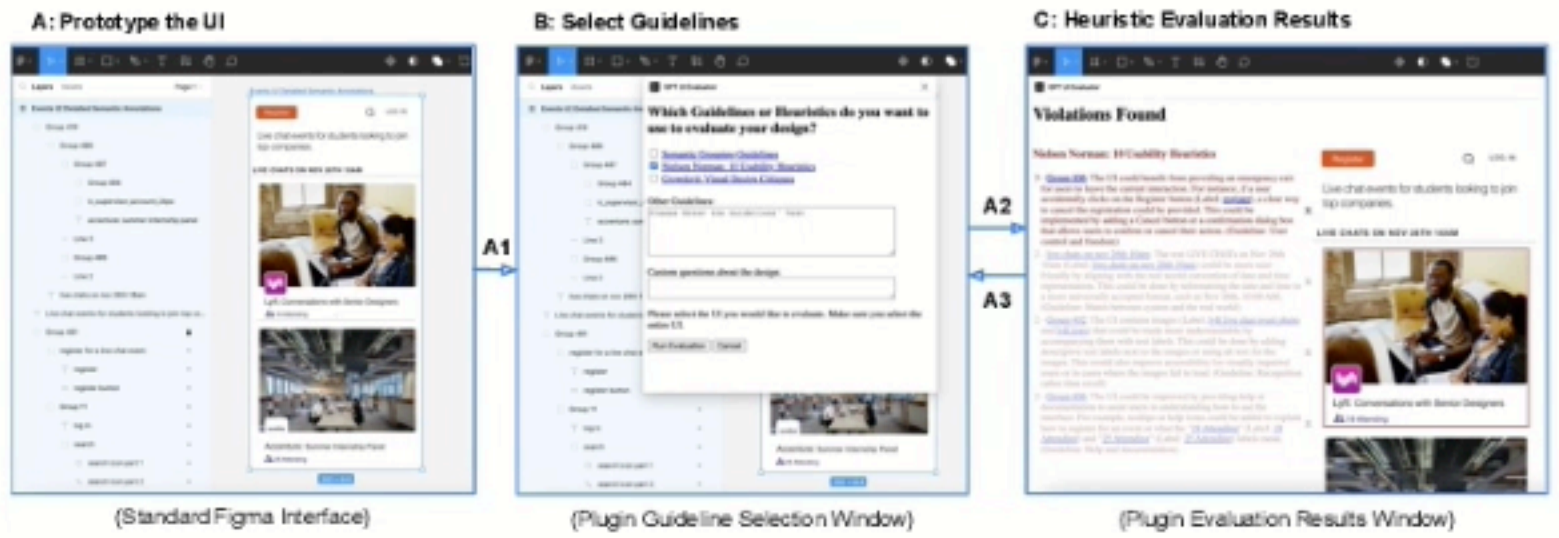


Figure 1: Diagram illustrating the UI prototyping workflow using this plugin. First, the designer prototypes the UI in Figma (Box A) and then runs the plugin (Arrow A1). The designer then selects the guidelines to use for evaluation (Box B) and runs the evaluation with the selected guidelines (Arrow A2). The plugin obtains evaluation results from the LLM and renders them in an interpretable format (Box C). The designer uses these results to update their design and reruns the evaluation (Arrow A3). The designer iteratively revises their Figma UI mockup, following this process, until they have achieved the desired result.

1 INTRODUCTION

User interface (UI) design is an essential domain that shapes how humans interact with technology and digital information. Designing user interfaces commonly involves iterative rounds of feedback and revision. Feedback is essential for guiding designers towards improving their UIs. While this feedback traditionally comes from humans (via user studies and expert evaluations), recent advances in computational UI design enable automated feedback.

Penn Engineering

What does "Box A" in Figure 1 represent? Describe what the user and system would be doing.

→

5.4 Qualitative Results: GPT-4 Strengths and Weakness

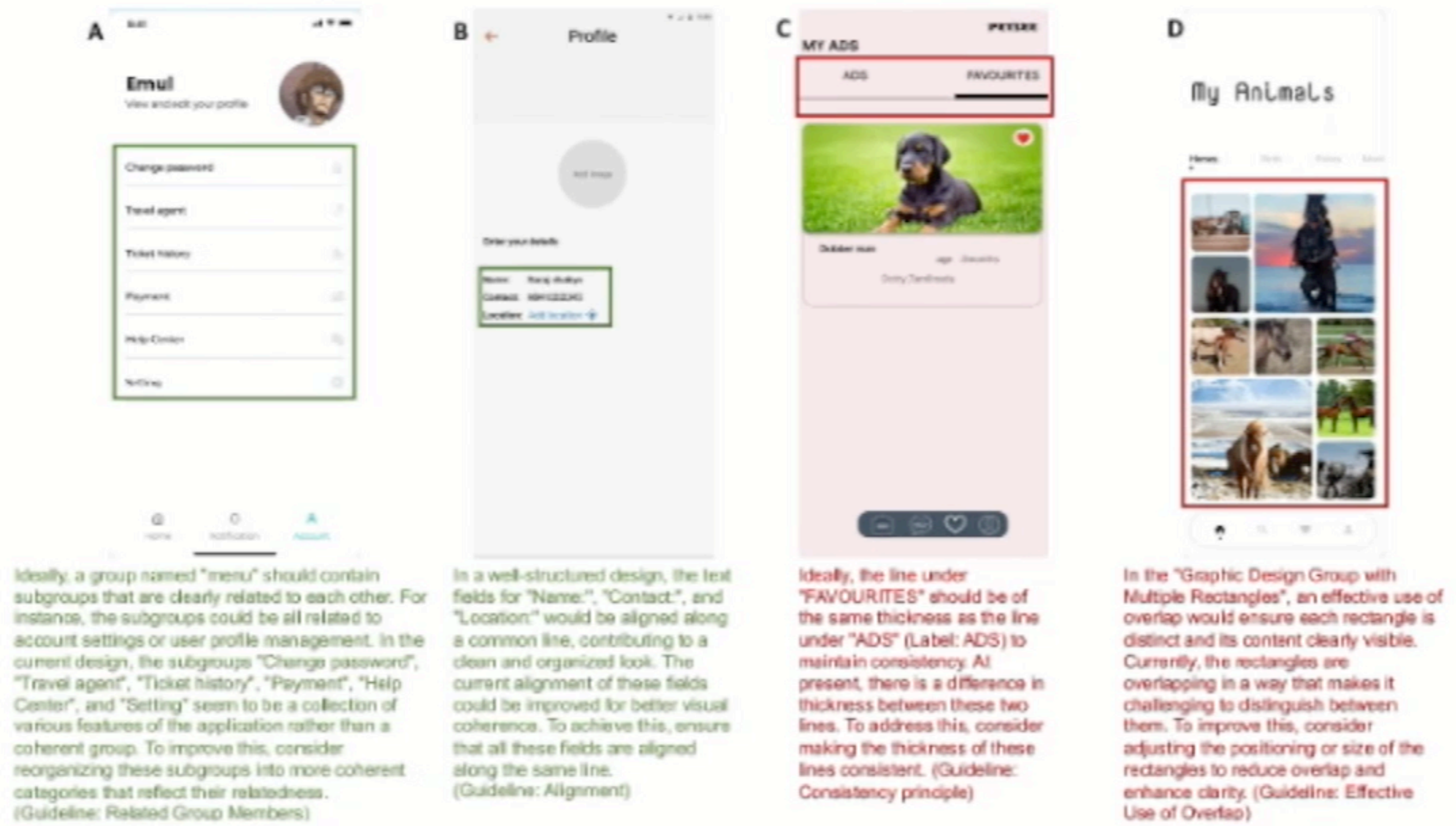


Figure 9: Examples of GPT-4 suggestions that all participants found very helpful or unhelpful, along with their corresponding UIs above (with the relevant group marked). The suggestions for UIs A and B received ratings of 5 for helpfulness and were rated as accurate by all three participants (from the Usage study). The "Contact:" field for UI B is slightly misaligned from the other fields, which GPT-4 caught. UIs C and D were rated 1 for helpfulness by all three participants. For UI C, the LLM stated that the line thickness was uneven under the "ADS" and "FAVORITES" tab, which is technically accurate (and some participants rated it as accurate) but unhelpful as the uneven line thickness is meant to indicate the selected tab.

We analyzed GPT-4's suggestions, corresponding expert ratings, explanations, and interview responses from the Usage study. Through grounded theory coding [16] of the qualitative data and subsequent thematic analysis [4], we identified the following emerging themes on GPT-4's strengths and weaknesses. Figure 9 contains examples of high and low-rated LLM suggestions to illustrate some of these themes.

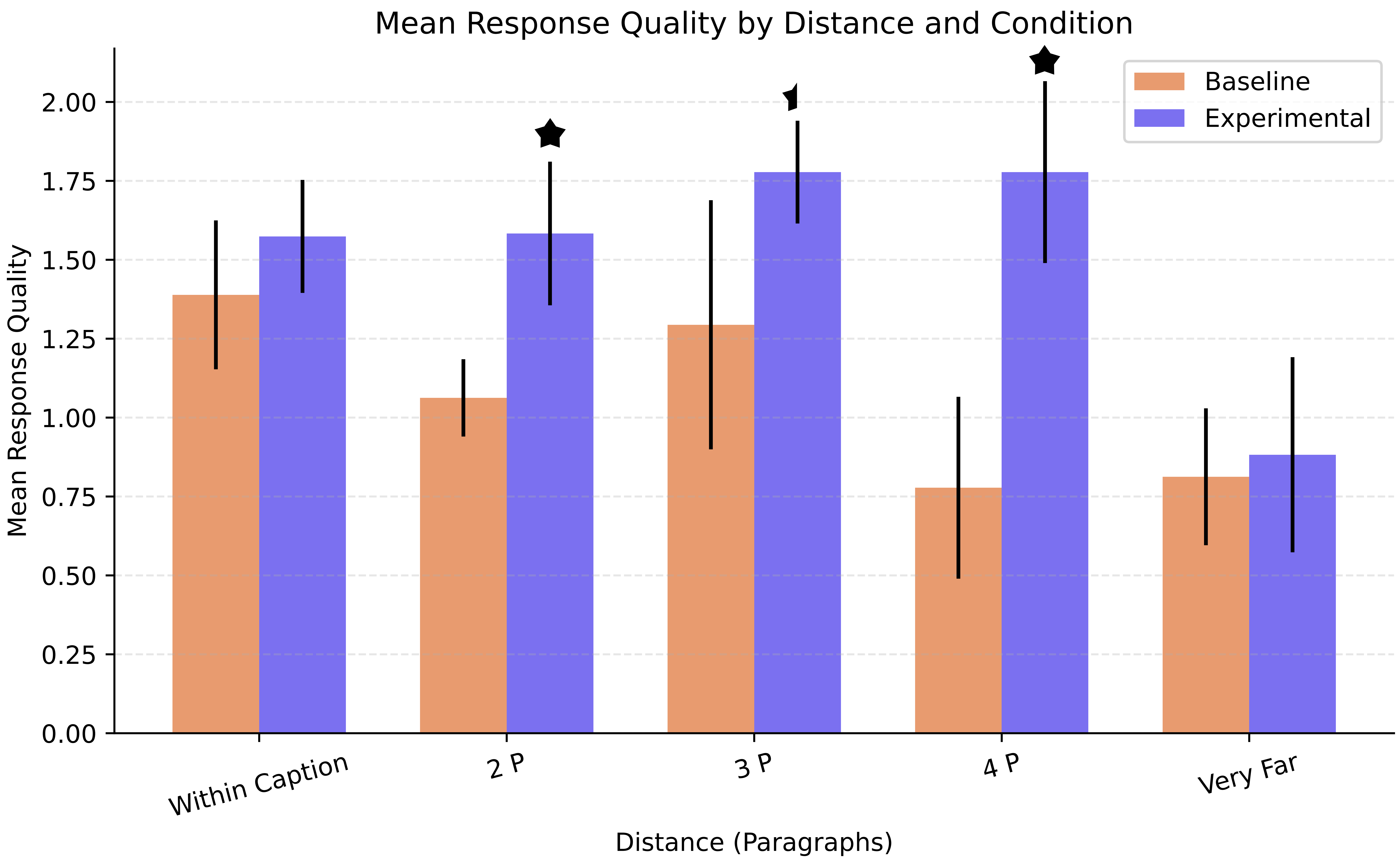
5.4.1 Strength 1: Identification of Subtle Issues (12/12 Participants). All participants found GPT-4's ability to identify subtle, easy-to-miss issues helpful. This includes problems like misalignment, uneven spacing, poor color contrast,

Penn Engineering

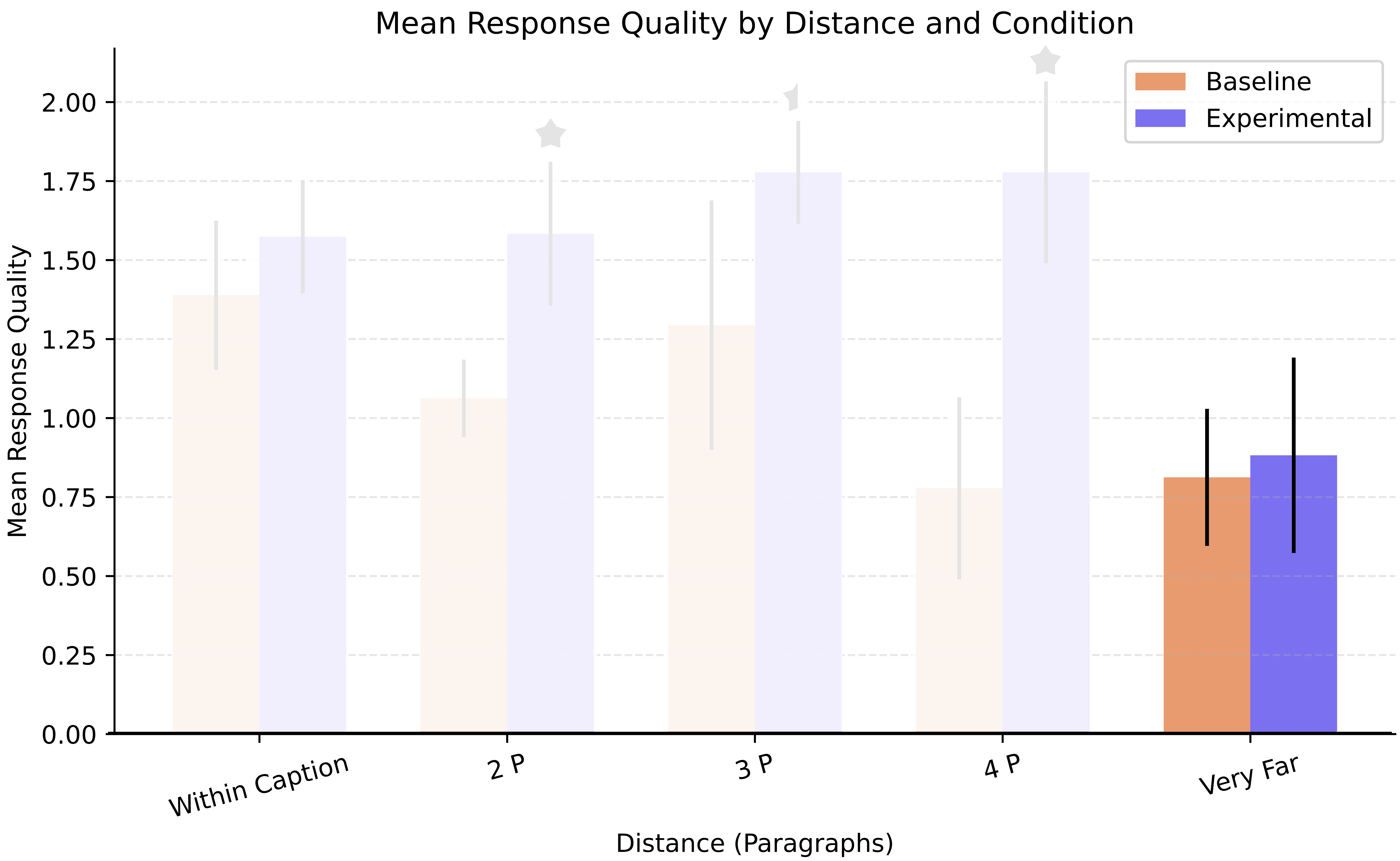
Why was UI D in Figure 9 rated 1 for helpfulness?

→

Significant improvement in response quality



Significant improvement in response quality



3.3 Implementation

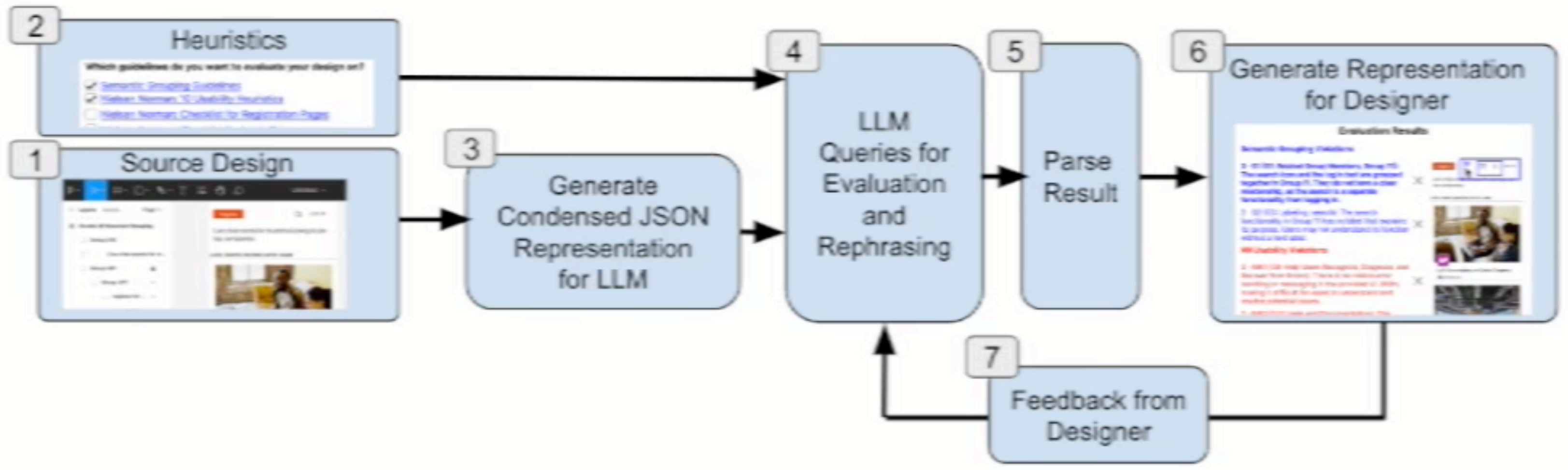


Figure 3: Our LLM-based plugin system architecture. The designer prototypes a UI in Figma (Box 1), and the plugin generates a UI representation to send to an LLM (3). The designer also selects heuristics/guidelines to use for evaluating the prototype (2), and a prompt containing the UI representation (in JSON) and guidelines is created and sent to the LLM (4). After identifying all the guideline violations, another LLM query is made to rephrase the guideline violations into constructive design advice (4). The LLM response is then programmatically parsed (5), and the plugin produces an interpretable representation of the response to display (6). The designer dismisses incorrect suggestions, which are incorporated in the LLM prompt for the next round of evaluation, if there is room in the context window (7).

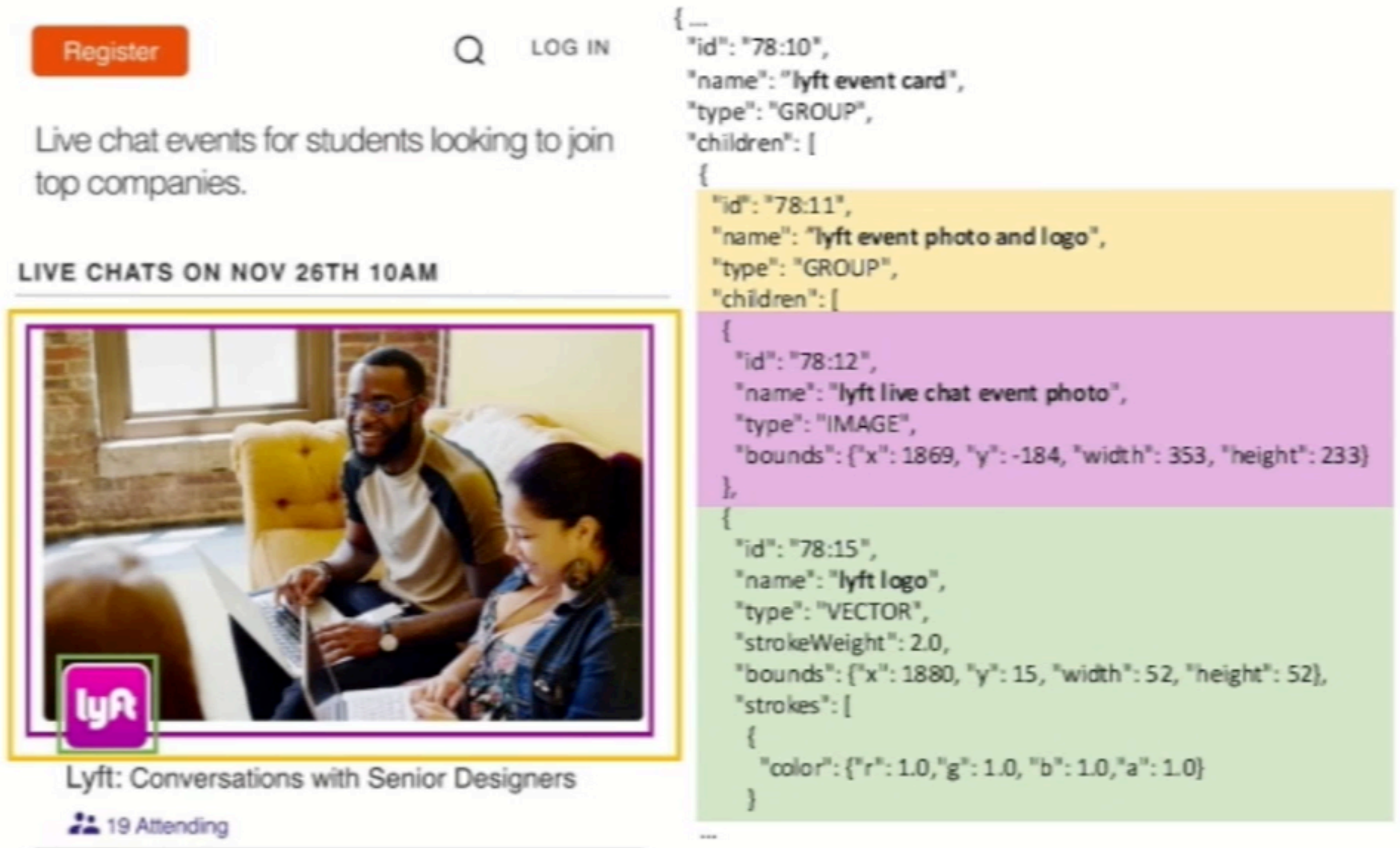


Figure 4: An example portion of a UI JSON. It has a tree structure, where each node has a list of

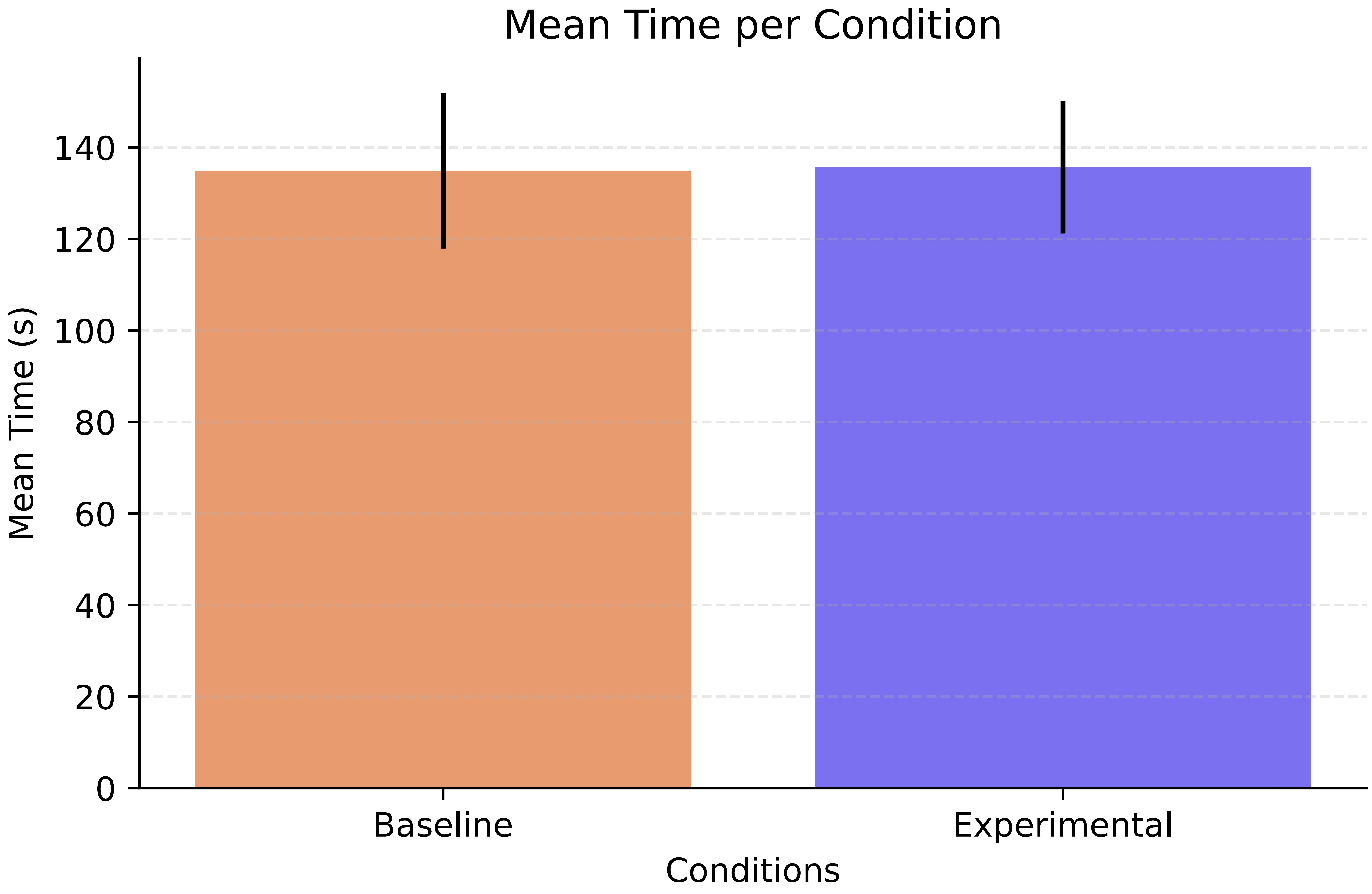
In what format are heuristics injected into the LLM prompt after the designer selects guidelines?

Continue to the next question when you are ready.

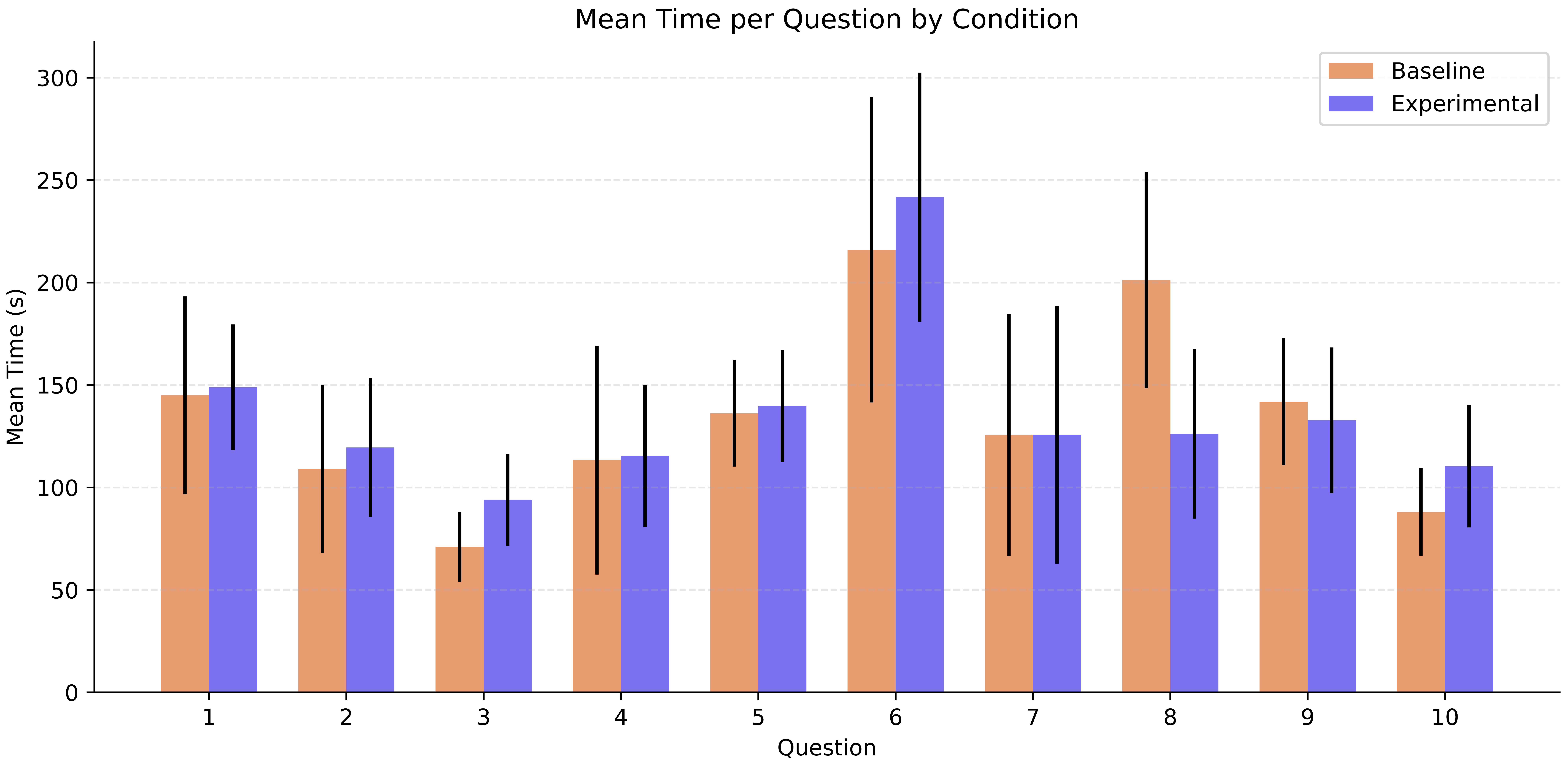
→

Our framework helped more for moderate distances (with opportunity for future work on far distances).

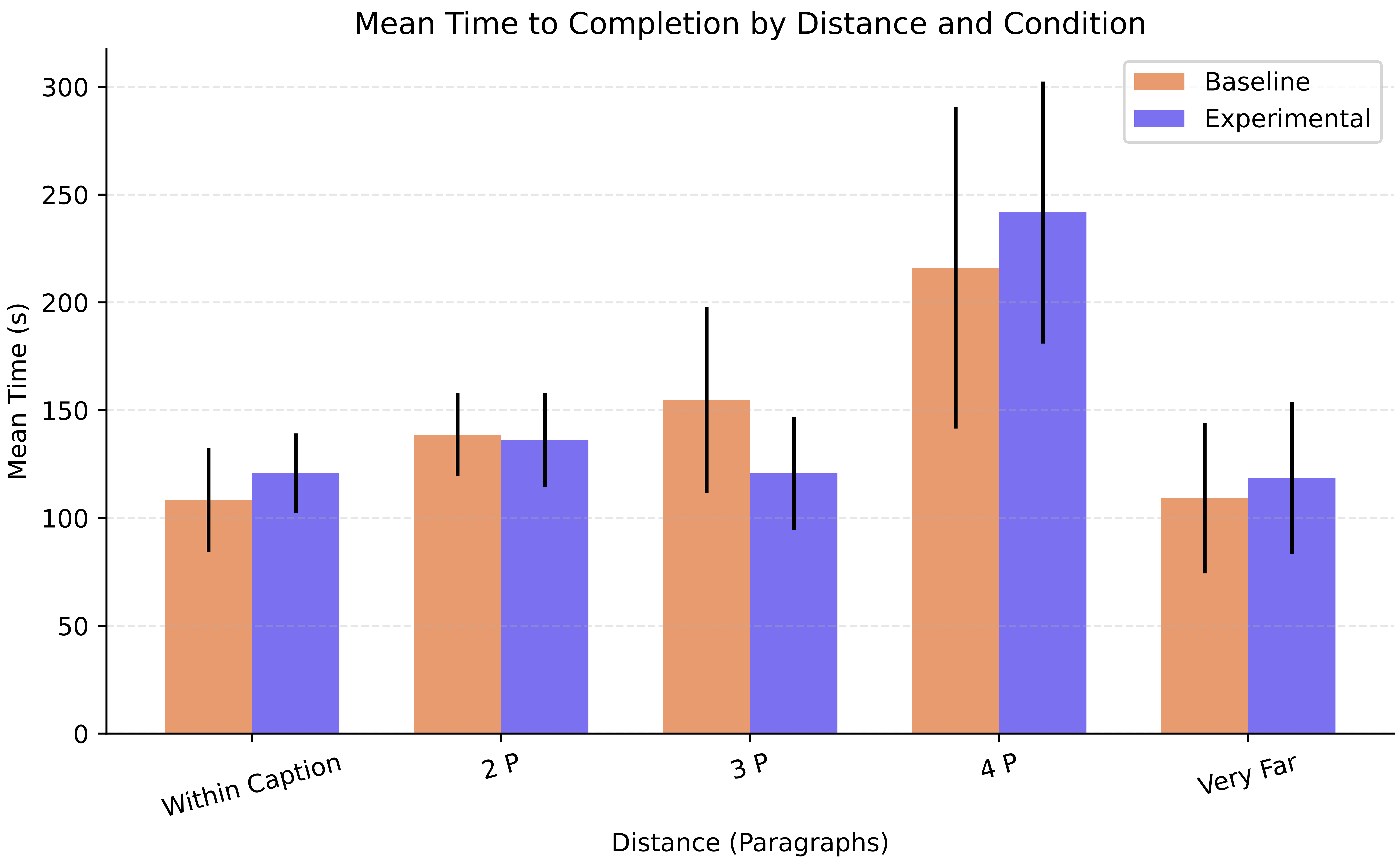
No meaningful difference in duration



No meaningful difference in duration



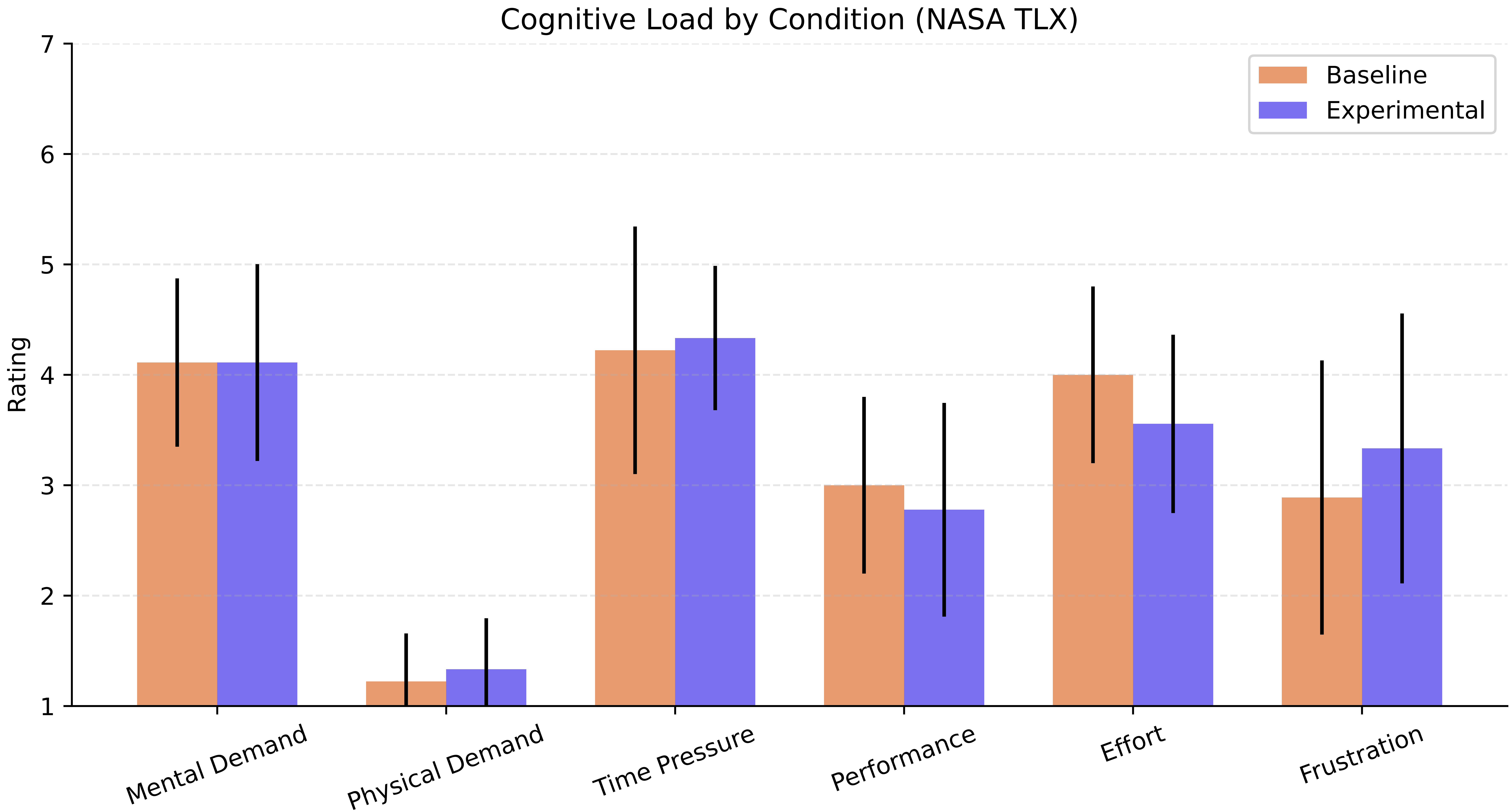
No meaningful difference in duration



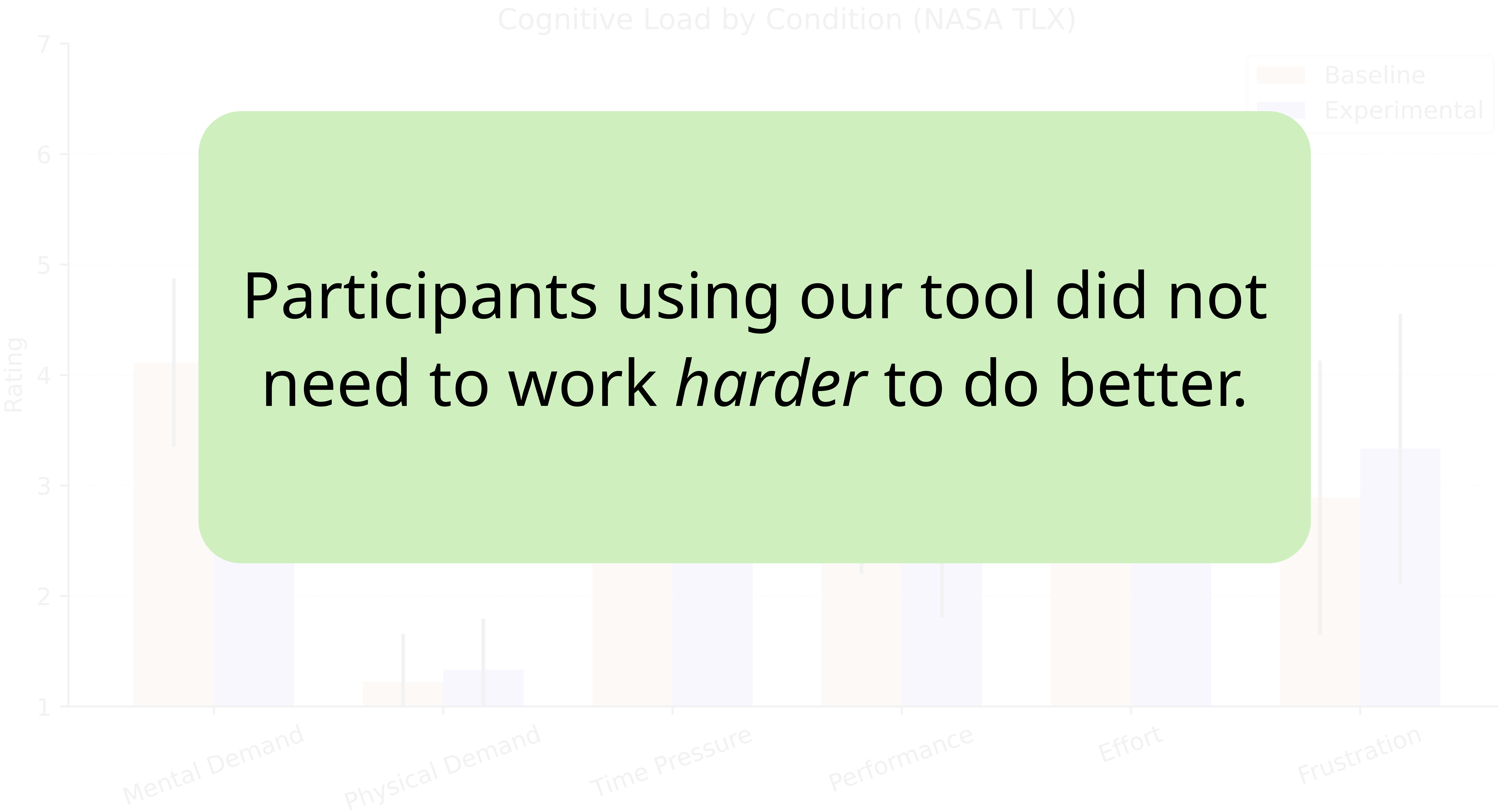
No meaningful difference in duration

Participants using our tool did not need to work *longer* to do better.

No meaningful difference in cognitive load



No meaningful difference in cognitive load



Participants using our tool did not need to work *harder* to do better.

3

Does surfacing these connections measurably improve comprehension?

Our framework improved response quality *without* increasing time to completion or cognitive load.

Reinforced by qualitative findings

- Reduced navigational burden: reference panel and visual index
- Ease of searching: figure points
- Increased engagement through verification
- Simultaneously developing mental map of information layout

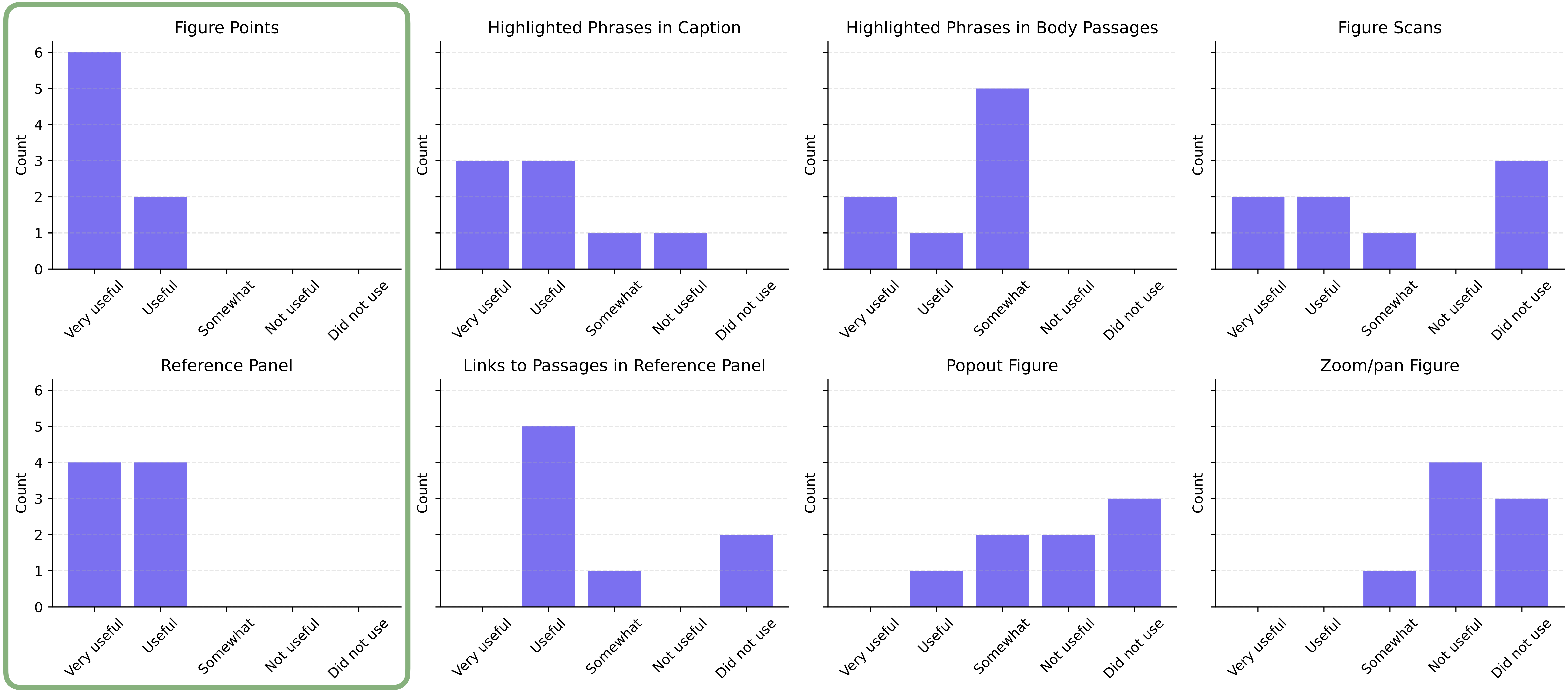
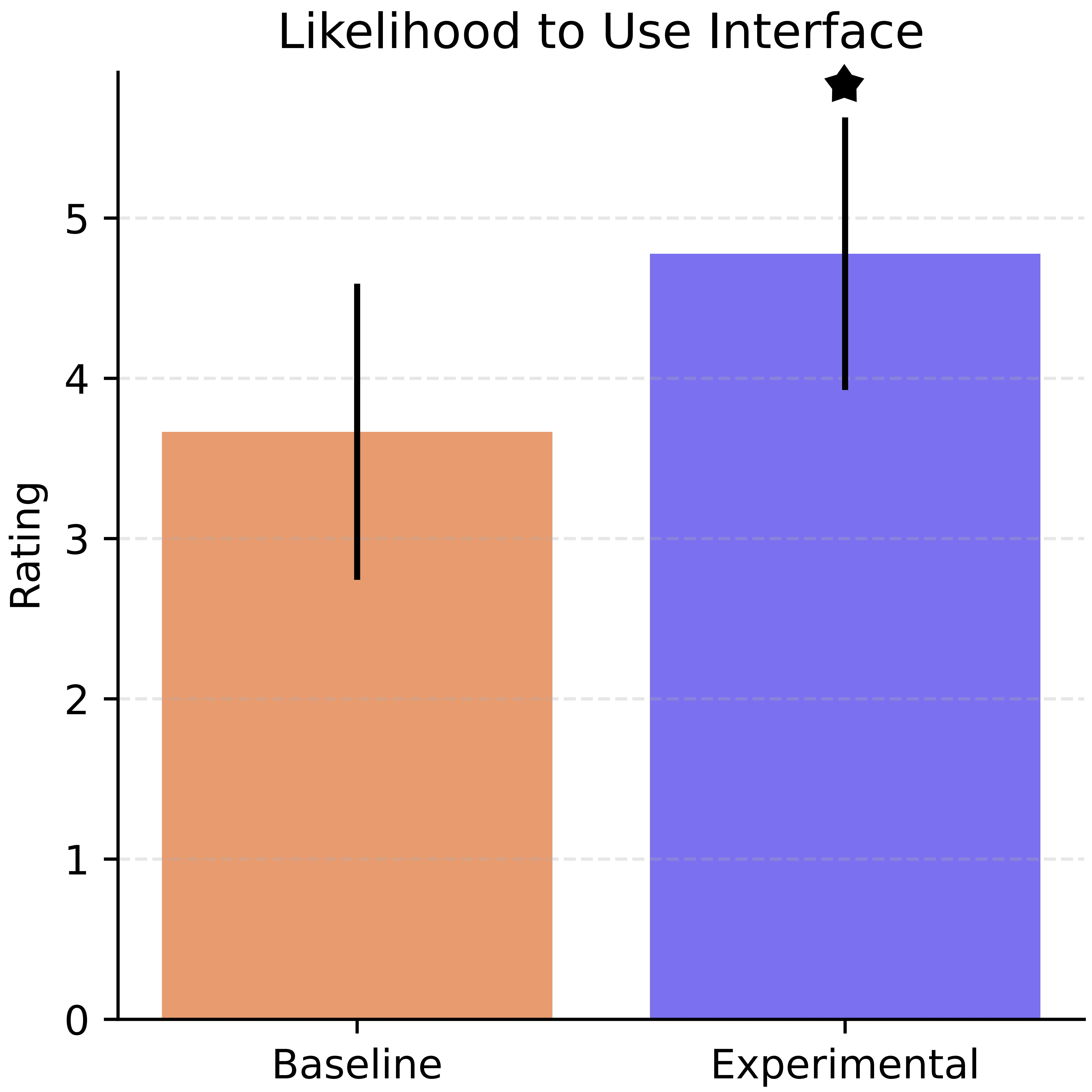
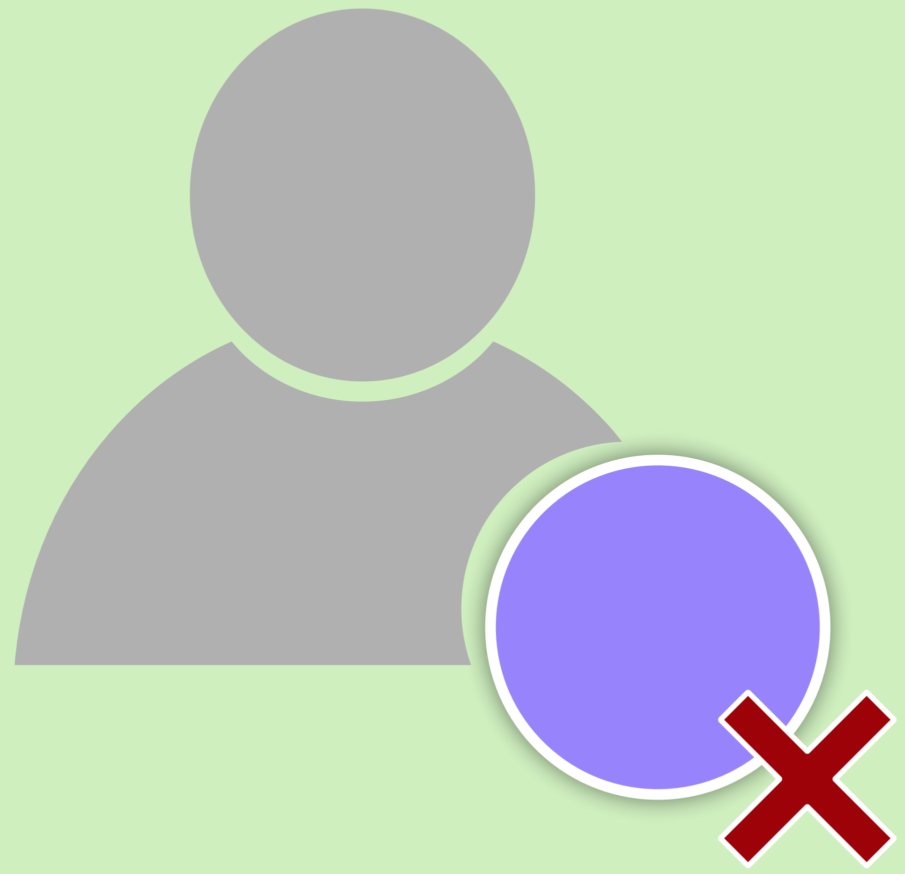


Figure Points
Highlighted Phrases in Caption
Highlighted Phrases in Body Passages
Figure Scans
Reference Panel
Links to Passages in Reference Panel
Popout Figure
Zoom/pan Figure
Count
0
1
2
3
4
5
6
Very useful
Useful
Somewhat
Not useful
Did not use

Significant preference for our framework

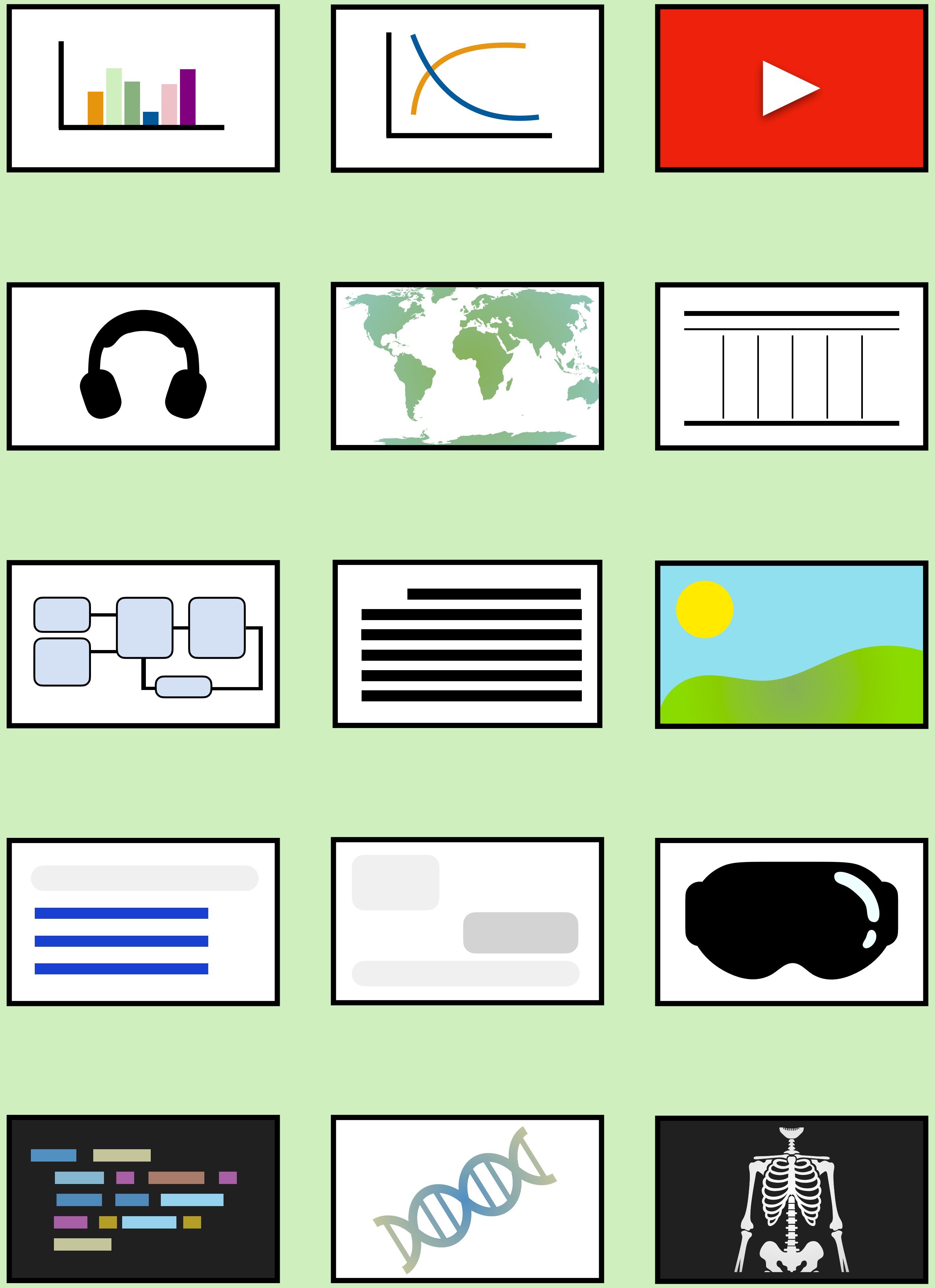


Future work in fine-grained integration of information



Living with and mitigating AI errors

Living with and mitigating AI errors



Designing for different media types

Living with and mitigating AI errors

Designing for different media types

Interacting through different modalities

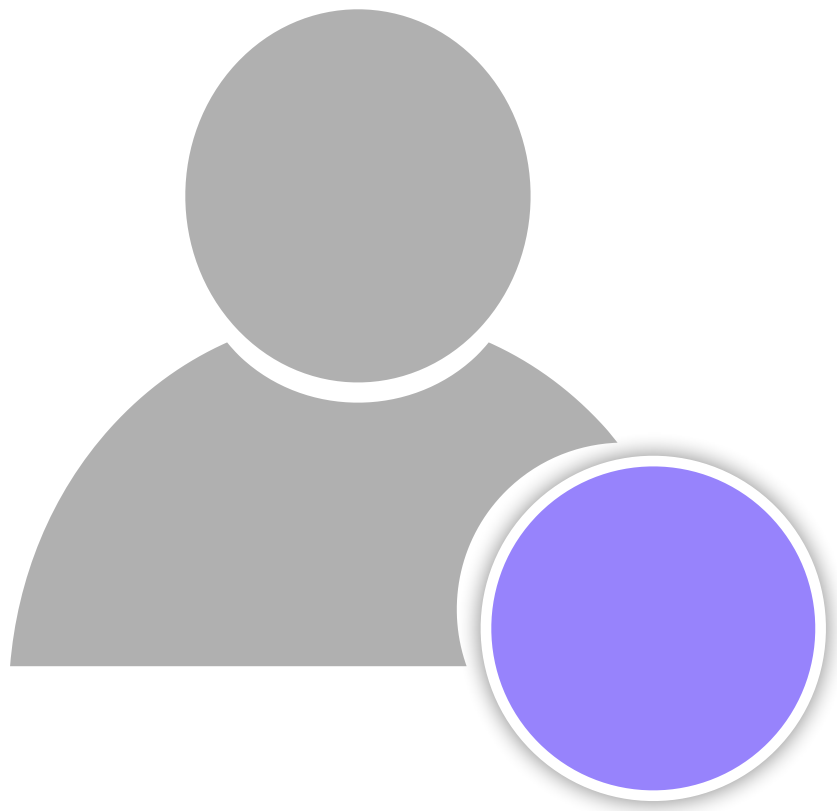
Summary

1. Challenges
2. Representations
3. Improvements

synthesizing scattered details

fine-grained augmentations

quality w/o inc. time or cog. load



Thank you!

Questions?