DOI: 10.1111/cgf.15093

Eurographics Conference on Visualization (EuroVis) 2024
W. Aigner, D. Archambault, and R. Bujack
(Guest Editors)

COMPUTER GRAPHICS *forum*
*Volume 43 (2024), Number 3*

# AVA: Towards Autonomous Visualization Agents through Visual Perception-Driven Decision-Making

S. Liu[1] H. Miao [1] Z. Li [2] M. Olson [1] V. Pascucci [2] and P-T. Bremer[1,2]

[1]Lawrence Livermore National Laboratory, USA
[2]University of Utah, USA

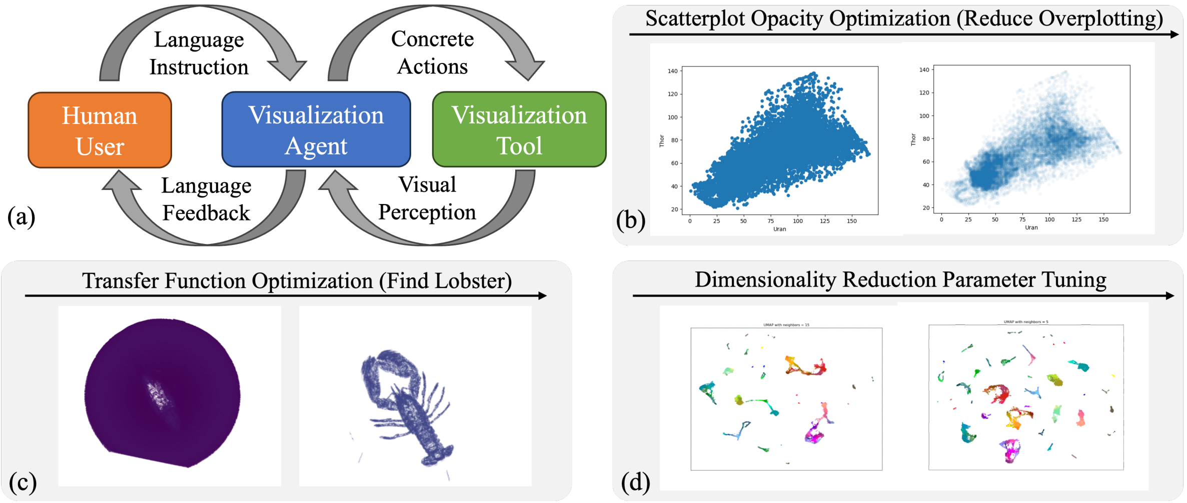

Language Instruction
Concrete Actions
Human User
Visualization Agent
Visualization Tool
Language Feedback
Visual Perception
(a)
Scatterplot Opacity Optimization (Reduce Overplotting)
Thor
Uran
(b)
Transfer Function Optimization (Find Lobster)
(c)
Dimensionality Reduction Parameter Tuning
(d)

**Figure 1:** *Overview of Autonomous Visualization Agents (AVAs): By leveraging the latest development of multi-modal LLM (a) AVAs can not only understand natural language instructions but also control and adjust a visualization system by processing its visual outputs to accomplish user-specified goals.*
We demonstrate the broad applicability of the proposed paradigm in multiple distinct scenarios including scatterplot opacity selection (b), volume rendering (c), and hyperparameter tuning for nonlinear dimensionality reduction (d).

**Abstract**

*With recent advances in multi-modal foundation models, the previously text-only large language models (LLM) have evolved to incorporate visual input, opening up unprecedented opportunities for various applications in visualization. Compared to existing work on LLM-based visualization works that generate and control visualization with textual input and output only, the proposed approach explores the utilization of the visual processing ability of multi-modal LLMs to develop Autonomous Visualization Agents (AVAs) that can evaluate the generated visualization and iterate on the result to accomplish user-defined objectives defined through natural language. We propose the first framework for the design of AVAs and present several usage scenarios intended to demonstrate the general applicability of the proposed paradigm. Our preliminary exploration and proof-of-concept agents suggest that this approach can be widely applicable whenever the choices of appropriate visualization parameters require the interpretation of previous visual output. Our study indicates that AVAs represent a general paradigm for designing intelligent visualization systems that can achieve high-level visualization goals, which pave the way for developing expert-level visualization agents in the future.*

## 1. Introduction

Recently, large language models (LLMs) have been widely adapted to solve a variety of tasks [WXJ*23, WBZ*23, SWW*23, WMF*23]. In the visualization domain, LLMs have been used to produce visualizations [DD19, Dib23] either through visual grammars like *Vegalit* [SMWH17], or directly generating visualization code (e.g., in Matplotlib [Hun07], VTK [SK19]). However, due to the inherently visual nature of these systems, purely language-based models have limited capability to make sense of their output. This significantly hampers or even prevents the analysis of the results and thus severely limits the opportunity for iterative interactions with the given visualization systems. The recent introduction of multimodal LLMs, such as GPT-4V, has the potential to address this fundamental limitation by filling the visual understanding gap, which opens many possibilities for new paradigms of interaction between existing visualization tools and human users.

One particularly interesting and powerful usage is the adoption of an Autonomous Visualization Agent (AVA) that can act as the medium between domain experts and visualization tools to facilitate and enrich user interaction (see Figure 1(a)). Here, the AVA is defined as an entity that can understand high-level instructions (i.e., natural language) and autonomously carry out a sequence of actions in a visualization system. More specifically, given the ability to perceive the visualization output an AVA can adjust and refine the parameters to meet the initial user-specified goal. Such an agent will not only be able to relieve the user from potentially tedious and repetitive tasks but will also haven the potential to accomplish non-trivial visualization goals by iterative refining the existing visualization through visual feedback (following the *visualization -> perception -> action* paradigm, just as a visualization expert would do).

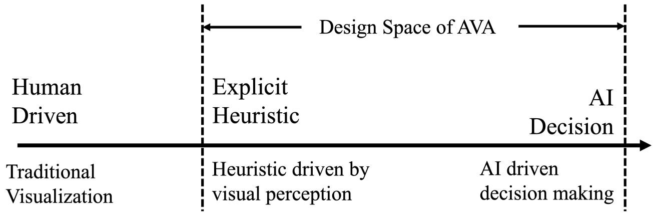

Design Space of AVA
Human Driven
Explicit Heuristic
AI Decision
Traditional Visualization
Heuristic driven by visual perception
AI driven decision making

**Figure 2:** *The design space of AVAs. On one end, we explicitly encode heuristics on how to update the visualization parameter, i.e., how a transfer function should be changed, which is driven by a high-level objective specified through language, i.e., "does this show the structure of interest". Alternatively, we can aim for a fully self-directed system with no explicit guidance on its action beyond the initial instruction (prompt).*

In this work, we aim to take the first step towards making AVAs a reality by exploring their design space and demonstrating their initial capability for solving various visualization tasks and provide an assessment of their feasibility in terms of stability, consistency, and speed. The key power of AVAs derives from their ability to detect visual features associated with natural language instruction. Consequently, they can evaluate complex objectives that cannot be easily expressed algorithmically, i.e., *is there a particular* shaped structure *in the rendering results?* or *does overplotting exist in the given scatterplot?* Despite its power, visual perception capabilities are only part of an agent. Once we obtain the visual understanding, the agent needs to plan its actions to achieve the goal. As illustrated in Figure 2, this presents a range of possible designs. On one end (i.e., more explicit control), we can rely on user-defined heuristics to dictate the response. This is achieved by encoding our prior domain knowledge into decision rules. Alternatively, we can rely on the LLM and its prior knowledge to process the observations and plan the next action in a fully self-directed fashion.

To design an effective AVA, we first need to understand the capability and limitation of visual perception of the state-of-the-art multimodal LLMs (we use GPT4-Vision in all of our studies). We carried out a preliminary exploration of a few perception tasks related to common visualization outputs, including volume rendering, scatterplots, parallel coordinate plots, and graphs. Leveraging what we learned from these simple benchmarks, we avoid areas of visualization where the visual perception of the current models is performing less accurately (e.g., graph, parallel coordinate). To demonstrate the feasibility and broad applicability of the proposed scheme, we intentionally select a distinctive set of applications, ranging from scientific/medical visualization to information visualization and dimensionality reduction. Our key contributions are:

- Introduce AVAs, a new paradigm that leverages the visual perception capability of a machine learning model for autonomous decision-making in visualization. Make the first step toward building visualization agents that can act as virtual visualization experts;
- Provide a preliminary exploration of state-of-the-art multimodal LLM's visual perception ability for interpreting different visualization outputs, including, scatterplots, parallel coordinate plots, graphs, volume rendering outputs, etc.
- Demonstrate AVA's feasibility and adaptability on several distinct visualization applications.

## 2. Related Works

### 2.1. Visualization Generation and Recommendation

Several existing tools explore how to generate visualizations based on user instructions. Data2vis [DD19] utilizes a recurrent network to generate code for visualization. The NL4DV [NSS20] approach turns visualization queries into visualization descriptors within the Vega-lite grammar [SMWH17], and the work by Mitra et al. [MNES22], explores the back and forth interaction with such visualizations using a natural language interface. LLMs have been adopted for a similar role in LIDA [Dib23]. VOICE [JIS*23] employs text-only LLM agents that can turn voice commands into instructions for generating interactive visualization. The KG4Vis work [LWZ*21] adopts knowledge graphs to produce visualization recommendations. Besides just generating the visualization, several methods explore utilizing machine learning (ML) to help explain the rationale behind why a given visualization is recommended. For example, AdaVis [ZWLQ23] adopts an attention-based model for explainable visualization recommendation, whereas the follow-up work leverages LLMs [WZW*23] to achieve a similar goal. Chen et al. [CZW*23] evaluate LLMs for solving visualization coursework by directly feeding them assignment descriptions. Apart from generating code that produces visual output, LLMs are ideal for text description generation. Zong et al. [ZLL*22] utilize LLMs to generate descriptions of visualization for visually impaired users to understand and navigate the visualization. Compared to these previous works, our primary goal is to develop an agent that can interpret visual output and refine visualization iteratively to accomplish specific tasks rather than generate visualization.

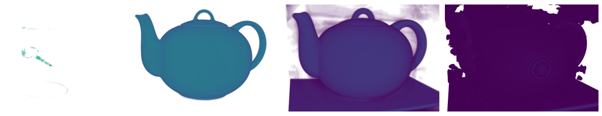
### 2.2. LLM-based Autonomous Agents

With recent advances in LLM, there has been an explosive interest in developing LLM-based autonomous agents. Compared to traditional reinforcement learning agents that often need to develop world understanding from scratch, LLM's in-depth prior knowledge and information processing capability make them more adaptive to complex environments and solving intricate tasks. Voyager [WXJ*23] introduces an LLM-powered embodied agent in Minecraft that can continuously explore the world and achieve milestones in the game world that were not possible with previous reinforcement learning approaches. Due to the vast literature in this space and relevance to the current work, we refer readers to a comprehensive survey on LLM-based agents [WMF*23]. In the following discussion, we will focus on vision task agents-related works. Even though most LLM models are not designed from the ground up for processing visual inputs, many recent works try to incorporate external vision model [SMV23] or develop auxiliary components and fine-tune the model to provide additional vision capability [FZF*23,ZSC*23]. These adaptions often focus on specific tasks and are trained on smaller-scale data, therefore are not designed for more general capabilities. This changes with the recent introduction of the GPT4-V (vision) [Ope23] model by OpenAI, which added visual perception to one of the largest and most capable LLM models. A detailed evaluation of a broad spectrum of visual understanding tasks is discussed in the "The Dawn of LMM" work [YLL*23]. Compared to the more general evaluation task in [YLL*23], we try to explore the GPT4-V model's visual perception capabilities on a specially designed set of visualization tasks, and eventually design an agent that is capable of refining and improving visualization output autonomously.

## 3. Preliminary Exploration of Multimodal-LLM for Static Visualization Perception

Before we can design an effective visualization agent that relies on visual input for decision-making, it is crucial to obtain some basic understanding regarding its capabilities and limitations for perceiving various types of visualization output. It is important to note that this is **not intended to be a rigorous and systematic evaluation of multi-modal LLM ability**, as an in-depth study will require substantial resources and effort that is beyond the scope of this work. We hope this assessment can help illustrate what type of visualization agents we can realistically design and what would be the ideal tasks for such agents.

**Volume Rendering.** We begin with evaluating the LMM's ability to recognize structures of interest within direct volume rendering images. Unlike photo-realistic images, that was explored by the work of Yang et al. [YLL*23], the outputs of volume rendering are subject to additional complexity (i.e., varying transparencies) introduced by the underlying transfer function. To assess the model's capability, we present the model with the task of examining a screenshot and determining whether a specific object or structure of interest is 'recognizable' or 'not recognizable'. We define the assessments for the prompt as follows: **recognizable**: *The structure of interest and its shape can be discerned in the screenshot.* **not recognizable**: *The structure of interest cannot be identified in the image, even if another structure is recognizable.* Moreover, for the iterative refinement discussed in later sections, we include an additional **clear** assessment, as a strict stopping criterion for the optimization, which denotes that the structure of interest is distinctly visible without any other structures occluding it.



**(a)** *90%* **(b)** *40%* **(c)** *20%* **(d)** *5%*

**Figure 3:** *The Boston Teapot dataset volume rendered using the same color map but at varying opacity levels. Structure of interest: the teapot. The response from the LLM model was 3a: 'not recognizable', 3b: 'recognizable', 3c: 'recognizable', and 3d: 'not recognizable'*

We assess the model on two datasets, the Boston Teapot [TIH] , and a downsampled version of the Visible Male [SASW96]. The reason why the Boston Teapot was selected for this experiment is because there is another structure, a lobster, located inside the teapot. As illustrated in Figures 3, we maintained fixed rendering parameters, including viewpoint and colormap, while introducing variations solely in the opacity transfer function. To maintain uniformity across experiments, we employed a fixed-width (1/10th of the value range) triangular function for the opacity transfer function, altering only the peak value in the center of the window. As shown in the Figure, the model consistently provided accurate assessments in all cases. More details on the experiment and other assessments for other datasets can be found in the supplementary material.

**Scatterplot.** Compared to volume rendering images, in which visual recognition is a simple binary task, i.e., object recognition, the assessment of visual structure in scatter plots is more nuanced. Here, we design five basic visualization tasks: *cluster recognition, cluster counting, outlier detection, outlier counting, and correlation detection* to evaluate its performance. The evaluation result is displayed in Table 1. For cluster recognition, our experiments show that the model can easily tell the plot has clusters (100% success rate). However, for the counting task, the success rate of the model is only at 60%. Similarly, we also separate the outlier tasks into recognition and counting. The final result aligns with the cluster recognition task. The model performs well on the outlier recognition task which has a 100% success rate, but has medium performance on the outlier counting task. In the correlation detection task, the model has a 100% success rate.

The experiment results show that the model has a decent ability to understand and analyze the scatter plot. However, in the experiment, most of the visualizations have clear signals to tell whether certain features exist. This raises another question of whether the model is able to identify ambiguous cases. We perform another simple experiment in which a scatter plot has two clusters but with different point spreads. From the result of Fig 4, we can tell that except for the last one which it is hard for humans to tell whether it has two clusters or not, the LLM model is able to identify the rest of the example accurately.

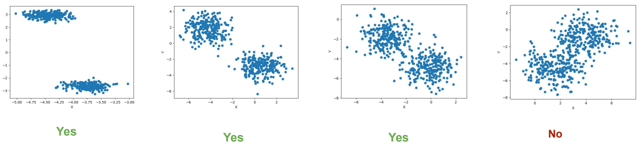


**Figure 4:** *The ability of GPT4-V to identify clusters in the scatter plot with different levels of ambiguity.*

| Tasks | scatter plot(success rate) | parallel coordinates |
|---|---|---|
| cluster | 100% | 100% |
| cluster count | 60% | 20% |
| outlier | 100% | 90% |
| outlier count | 60% | 80% |
| correlation | 100% | 20% |

**Table 1:** *The performance of GPT4-V on a scatter plot and parallel coordinate tasks. GPT4-V can identify outliers and cluster well in both visualizations. However, its ability for object counting is comparatively poor. Meanwhile, the correlation detection in parallel coordinates plots is also limited.*

**Parallel Coordinates Plot.** We examine parallel coordinate plots with the same tasks as the scatter plot. Both experiments have a similar setup on cluster and outlier tasks, except the number of dimensions in each dataset will change from 2D to 5D. The overall results are a bit worse than the model's performance on scatterplot visualization. In cluster and outlier recognition tasks, the model performs well. In the cluster counting task, parallel coordinates perform badly with a 20% success rate but in the outlier counting task, the GPT4-V model performs well. Opposite to the correlation task, the parallel coordinate makes it hard to identify correlation relationships.

| Tasks | node count | find node | connection | neighbor |
|---|---|---|---|---|
| success % | 50% | 100% | 70% | 10% |

**Table 2:** *The performance of GPT4-V on common graph tasks.*

**Graph.** To assess GPT4-V's visual understanding of graphs, we choose the classic graph visualization technique node-link diagram and adjacency matrix. In our experiment, we use the basic graph exploration task [GFC04] to evaluate the performance of the LLM. Instead of performing all tasks, we pick four tasks that are easy to perform without interactions. The overall result is displayed in Table 2. From the evaluation, we can tell that LLM can easily find a node in the graph visualization. However, it is difficult to tell the neighbor of the selected node. The connection tells whether two nodes are connected (directly or indirectly through other nodes), and the final result shows sub-optimal performance. Finally, the node count ability has a 50% success rate which shows that the model again has poor performance on the counting tasks.

Despite the relatively limited exploration, our experiment demonstrates the model's capability to discern structures and objects in volume rendering results. Among the information tasks, the model achieves better performance on scatterplots compared to parallel coordinate plots or graphs. Therefore, to leverage the strength of the system, in our case study (see section 5), we focus on volume rendering and scatterplot-related applications.

## 4. Autonomous Visualization Agent (AVA)

We define AVA as a paradigm for designing AI-driven agents that serve as a medium between a specialized visualization tool and a domain user. The key principle of AVA involves the utilization of machine vision for decision-making. It takes user instruction in natural language and achieves the user-specified goal by operating the visualization tool autonomously based on the visual understanding of visualization outputs. And we refer to the concrete implementation of AVA as AVAs.

### 4.1. Key Components of AVAs

To achieve its design goal, the AVAs need to accurately perceive visual input and make plans on what action to take based on current visualization results and do so by following user natural language instructions. As illustrated in Figure 5, AVAs need to contain at least three key components, namely visual perception, action planning, and memory.

**Visual Perception** the visual perception is at the center of the AVAs' capabilities, and what distinguishes them from existing LLM applications in visualization. There is some similarity between AVA and an embodied agent [ZDS*23] in robotic research, where an agent will take action based on sensory input (e.g., vision) and observe the impact of the action in the environment. Similarly, for AVAs, the sensory input is the visualized image, and the action corresponds to changes in the visualization setting (e.g., update parameters), and the impact of the action is a new visualization output.

**Action Planning** In order to make autonomous decisions and respond to the "sensory" input, the AVAs need an action planning component as the "brain" of the system. Here we have a range of choices for its design. As illustrated in Figure 2, we can either rely more on heuristics to drive the action planning or let the LLM do everything on its own, which corresponds to two distinct approaches to the action planning design.

- Heuristic-Centric: infuse our existing domain knowledge into heuristics for how to update the visualization tool based on assessment from the visual perception component. Their action plan is defined explicitly. In such a scenario, the visual perception and assessment essentially act as a loss function for a predefined optimization procedure.
- LLM-Centric: LLMs have innate knowledge and understanding of various topics, which can be leveraged to guide the exploration of the action space. Such prior knowledge can be combined with the initial prompt feed to the system to guide the action planning process without explicitly defining heuristics.

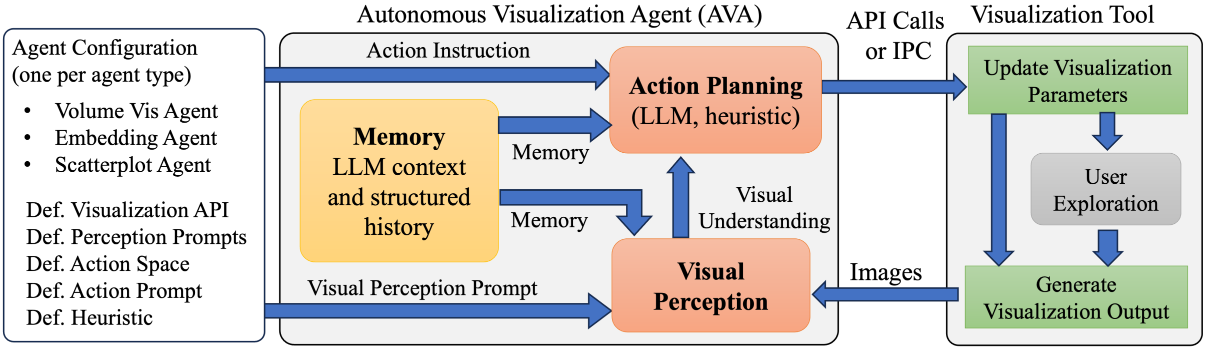


**Figure 5:** *The breakdown of the components of the AVA. The capability of the AVAs hinged on their visual perception, and the visual understanding can then be utilized by the action planning system to modify/steer the visualization tool. In order for AVAs to make informed decisions and the ability to understand context, they also need a memory component that both visual perception and action planning components can easily access.*

One important thing to note is that both approaches will provide autonomous decision-making based on visual perception, so from the user's perspective there may be little difference. The distinction between them comes from whether we want to rely explicitly on rules that are derived from our domain knowledge or we hope to leverage the pretrained LLM's knowledge base for planning and suggestion.

**Memory** Beside the visual perception, and action planning components, the other essential part of AVA that both of these components need is memory of the previous actions or the visualization outputs it observed before. In order to make complex and well-informed decisions, we often need to refer back to or compare with previously examined results or conclusions. The same is true for AVAs.

**Visualization-Perception-Action Loop** Besides the three key components, one essential aspect of AVA, and its key capability, is associated with the autonomous visualization loop, i.e., from visualization to perception and then to action. The process is bootstrapped by the specific high-level task given by the user and starts with a default visualization setup, and then the system:

- Generating visualization output by executing API calls to the visualization tool based on the given parameter.
- Leveraging the visual perception component to comprehend semantics and structure in the current visualization.
- Provide assessments of whether the visualization achieves the user-set goal, and the action planning component makes decisions on what visualization parameters it should use next.

The agent will iterate through these steps until the visualization goal is achieved. This methodological framework forms the foundation of AVAs, enabling us to utilize the visual perception of the multimodal LLM, as well as the planning capability of LLMs or user-defined heuristics to interact with visualization tools effectively. It draws the blueprint for how future autonomous agents can potentially navigate complex visualization tasks with precision and adaptability.

## 4.2. Implementation

So far, we have discussed the conceptual idea of how AVA works. In this section, we provide practical guidance on their implementation. Our implementation utilizes the GPT-4 Vision model [Ope23] for visual perception and action planning (for LLM-centric scenario), harnessing its natural language understanding capabilities alongside a visual perception engine. However, other alternatives such as Gemini [TAB*23] can be also be adopted.

To establish a flexible and reusable foundation for our AVAs, we have created an abstract class that acts as the basis for any specific type of agent. It contains several core functionalities: 1) a unified interface for accessing LLM APIs for visual perception or action planning tasks; 2) a basic blueprint on how an agent should interface with the visualization tool; 3) configuration functionality that helps define the agent, e.g., prompts template; 4) capability to parse and extract visual assessment results, parameter, and function call information from the language response. For specific types of applications, AVAs can be developed as concrete classes that inherit the base one, and in the new class, application-specific logic, e.g., heuristics-centric action planning, can be implemented. Each of the concrete classes will also have an associated JSON configuration file, prompts or part of prompts are organized in a structured fashion. Our AVA implementation is in Python. We designed a straightforward layout for the interface, with the control and conversation history on the left panel and the visualization of interests on the right.

**Agent Initialization** To initiate an AVA's functionality, we establish a context by prompting the Large Language Model (LLM) with the assumed role of the agent. This definition typically encompasses several elements: scenario, visualization task, goal, approach, and constraints. The content of the prompt can be divided into three major parts: 1) the overall template and common functionality (layout the general rule such as output format), 2) the detailed guidance for a specific type of task (e.g., action planning for opacity optimization volume rendering), and 3) the user-specific feature of interests (e.g., object to highlight such as lobster). Crafting effective prompts for 1), and 2) is essential for defining the agent's role and specifying the approach to achieving the visualization task, it is part of the agent design process. However, from the user's point of view, they only need to provide 3) for a given subject of interest.

**Connection Between AVA and Visualization Tool** The AVA can either directly call the API, provided both the agent and the visualization run in the same application/context. However, to maximize the flexibility and support complex external tools (e.g., GPU accelerated direct volume rendering), we also include support for a more generic solution with inter-process communication (IPC) mechanisms to facilitate seamless data exchange between the agent and the visualization tool. In our implementation, we utilize the RPyC (Remote Python Call) [Fil13] to facilitate the IPC. All the output of the LLMs will be in text format. To carry out API/function calls, we need to extract the relevant command and information (the current vision model does not support the default GPT function calling capability, which comes with its own limitation). One way to handle the challenge is through generating structured output (e.g., JSON format, or data array string, however, the LLM output may not always be valid so a format validation is needed to prevent runtime error and improve the stability of the system.

## 5. Case Studies

### 5.1. Opacity Transfer Function for Volume Rendering

In this case study, we focus on the opacity transfer function design process—a crucial task in volume rendering, where structures of interest must be appropriately depicted within the opaque range of the opacity transfer function. We test the agents with a dataset of a head [hea], which is a 3T Time-of-flight Magnetic Resonance Angiography and contains part of the portion around the height of the eyes where the brain arteries are located. It contains skin, soft tissues, the skull, and the vascular structure inside. The most interesting structure inside this dataset is the arterial blood supply of the brain, called the circle of Willis.

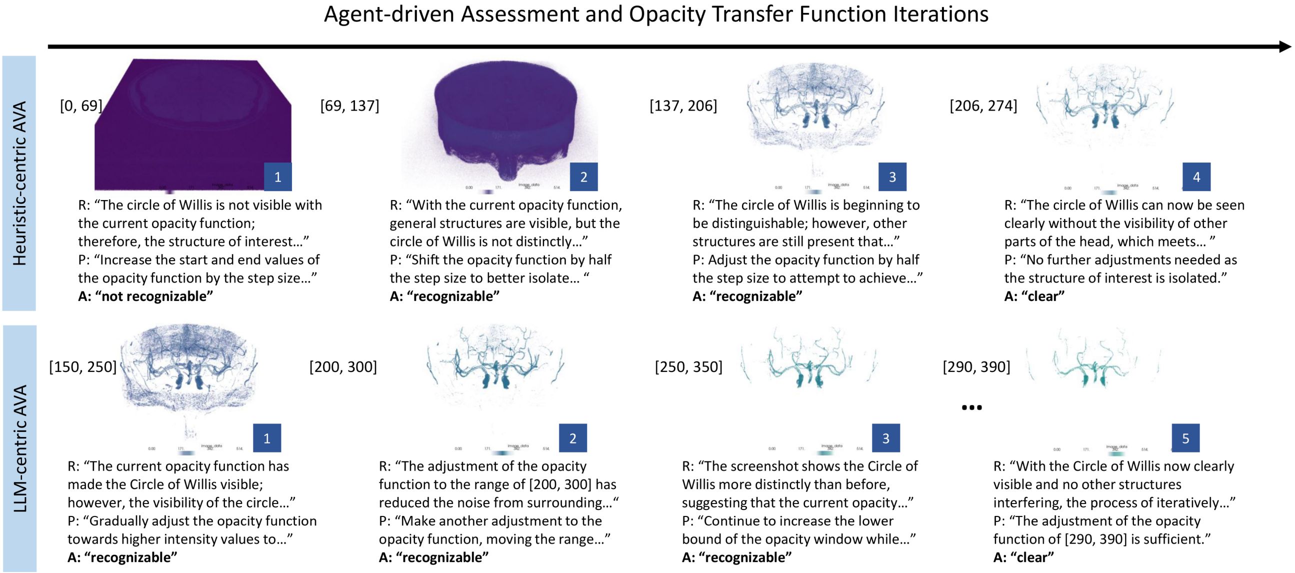
To facilitate a comprehensive discussion of AVA behavior, we implemented two different agents. A heuristic-centric agent receives the action plan as a heuristic defined by the user, while the LLM-centric agent utilizes the model's knowledge about the opacity transfer function design in order to devise a strategy. For both agents, the opacity function remained a triangle function with the peak value positioned between the start and endpoints. The viewpoint and color map were also fixed. Once the AVA recognizes the user-defined structure of interest, it will provide the *recognizable* assessment. We let the agent continue to further refine the opacity function for a *clear* assessment until occluding noise is removed if possible.

**Heuristic-Centric:** In this setup, the agent provides assessments, but the opacity transfer function adjustments are defined by the agent designer. For a proof-of-concept, we utilize a simple linear search-based approach that shifts the window of the opaque range towards higher values, while the function always assumes a triangle shape. For these tests, we selected parameters to separate the value range into 10 bins, where the window width is one bin wide. We shift the window one bin with each iteration. We also added a fine-tune parameter, where we reduce the speed when the structure of interest is "recognizable' but it is not yet 'clear'. In that case, the window shifts only by half of its width. The only information the domain user needs to provide here is the structure of interest (the circle of Willis, a vascular structure in the brain) and the value range.

In general, this action plan can be implemented in two ways, depending on the scenario. This heuristic can be added as code, such as a plugin integrated into the visualization code. To elucidate the agent's behavior in alignment with the heuristic described above, the agent initially employs a triangle function at the far left end of the value range and incrementally moves upward. It continues this process until it can confidently recognize the structure of interest, in this case, the circle of Willis. As the agent recognizes the vascular structure, it takes half a step to make smaller adjustments to the opacity transfer function until it can fully discern the circle of Willis. While we used a linear search strategy in this demonstration, other more sophisticated approaches can also be employed and implemented in a similar fashion.

**LLM-Centric:** This AVA is not limited to a user-defined heuristic for adjusting the opacity function. Instead, it can leverage the prior knowledge of transfer function design inherent in the LLM to facilitate the design process. However, it remains constrained to providing a triangle function as the opacity transfer function. We provided the agent with the acquisition modality and the histogram to provide it with similar information as a human user would have. As depicted in Figure 6, the agent explores various opacity ranges until it successfully generates the function for the circle of Willis. Notably, in this case, the agent operates with a greater degree of autonomy, employs a strategic approach, and reflects on past decisions as explained in its "reasoning" and "plan" as shown in Figure 6. Interestingly, it immediately devised a plan, where it starts with a range higher than the first peak in the histogram, which it correctly assumes is the background.

We further tested the AVA's capabilities on structures that are more challenging to find an appropriate opacity function. Specifically, we utilized the Boston Teapot dataset discussed in Section 4, which contains a lobster inside the teapot. The lobster in this dataset can be only partially visualized due to the low resolution of the data. This presented a more difficult scenario compared to the circle of Willis. Additionally, the lobster is relatively small within the dataset, resulting in its representation by a very low bin in the histogram. Despite these challenges, as demonstrated in Figure 7, the agent successfully determined the correct opacity transfer function within a few steps when tasked with identifying the lobster structure of interest. Following the reasoning in each step reveals its action-planning capabilities. To provide a comparison, we also tested the agent on another dataset, containing a Lobster in Resin (*301x324x56, uint8*, Courtesy of VolVis distribution of SUNY Stony Brook, NY, USA.). In both cases, the model reasoned that the second histogram peak might be the structure of interest, however in the Boston Teapot dataset, it found the teapot instead of the lobster. Remarkably, the LLM-centric agent moved on and tried different value ranges and found the lobster in just a few steps, even though the lobster is harder to recognize due to the low resolution. In comparison, in the Lobster in Resin dataset, the lobster was revealed together with the resin in the first iteration and then the agent fine-tuned the opacity in Step 3 until no resin is visible anymore.



R: "The circle of Willis is not visible with the current opacity function; therefore, the structure of interest…"
P: "Increase the start and end values of the opacity function by the step size…"
**A: "not recognizable"**

R: "With the current opacity function, general structures are visible, but the circle of Willis is not distinctly…"
P: "Shift the opacity function by half the step size to better isolate… "
**A: "recognizable"**

R: "The circle of Willis is beginning to be distinguishable; however, other structures are still present that…"
P: Adjust the opacity function by half the step size to attempt to achieve…"
**A: "recognizable"**

R: "The circle of Willis can now be seen clearly without the visibility of other parts of the head, which meets… "
P: "No further adjustments needed as the structure of interest is isolated."
**A: "clear"**

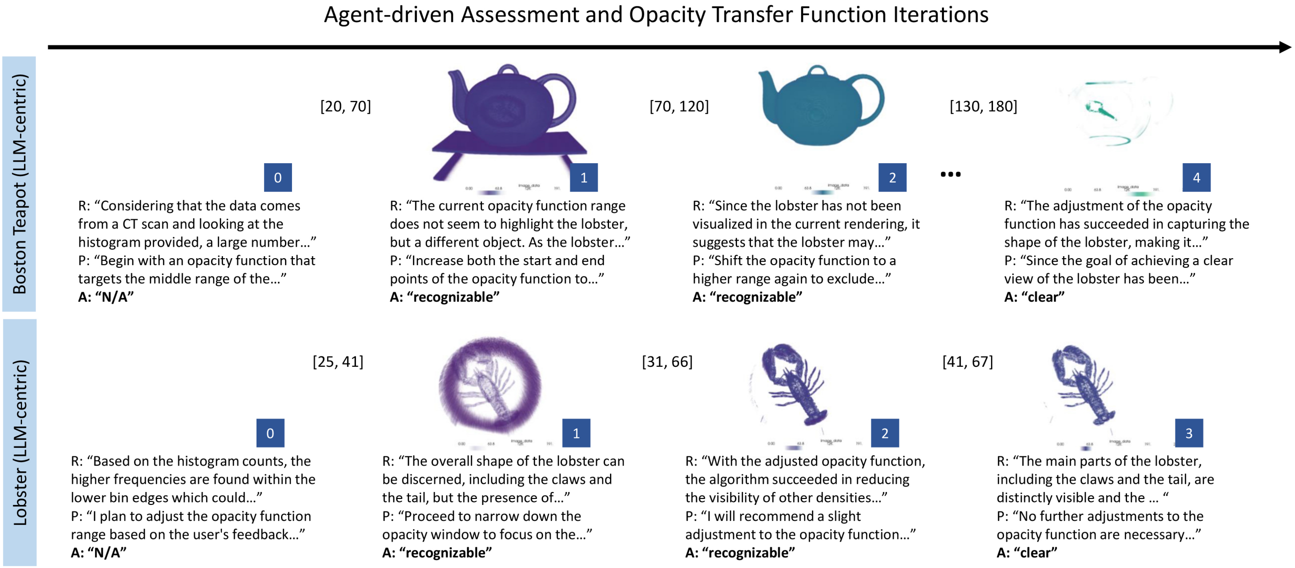
R: "The current opacity function has made the Circle of Willis visible; however, the visibility of the circle…"
P: "Gradually adjust the opacity function towards higher intensity values to…"
**A: "recognizable"**

R: "The adjustment of the opacity function to the range of [200, 300] has reduced the noise from surrounding…"
P: "Make another adjustment to the opacity function, moving the range…"
**A: "recognizable"**

R: "The screenshot shows the Circle of Willis more distinctly than before, suggesting that the current opacity…"
P: "Continue to increase the lower bound of the opacity window while…"
**A: "recognizable"**

R: "With the Circle of Willis now clearly visible and no other structures interfering, the process of iteratively…"
P: "The adjustment of the opacity function of [290, 390] is sufficient."
**A: "clear"**

**Figure 6:** *The results from the heuristic-centric and LLM-centric AVAs. The screenshots are generated from the proposed opacity transfer function. The response includes (R)easoning, (P)lan, and (A)ssessment by the agent, and based on this, the agent suggests a new pair of values to construct the triangle-shaped opacity transfer function. Each agent converged towards an opacity function rendering the structure of interest.*


Agent-driven Assessment and Opacity Transfer Function Iterations
Boston Teapot (LLM-centric)
0
[20, 70]
1
[70, 120]
2
•••
[130, 180]
4
Lobster (LLM-centric)
0
[25, 41]
1
[31, 66]
2
[41, 67]
3

R: "Considering that the data comes from a CT scan and looking at the histogram provided, a large number…"
P: "Begin with an opacity function that targets the middle range of the…"
**A: "N/A"**

R: "The current opacity function range does not seem to highlight the lobster, but a different object. As the lobster…"
P: "Increase both the start and end points of the opacity function to…"
**A: "recognizable"**

R: "Since the lobster has not been visualized in the current rendering, it suggests that the lobster may…"
P: "Shift the opacity function to a higher range again to exclude…"
**A: "recognizable"**

R: "The adjustment of the opacity function has succeeded in capturing the shape of the lobster, making it…"
P: "Since the goal of achieving a clear view of the lobster has been…"
**A: "clear"**

R: "Based on the histogram counts, the higher frequencies are found within the lower bin edges which could…"
P: "I plan to adjust the opacity function range based on the user's feedback…"
**A: "N/A"**

R: "The overall shape of the lobster can be discerned, including the claws and the tail, but the presence of…"
P: "Proceed to narrow down the opacity window to focus on the…"
**A: "recognizable"**

R: "With the adjusted opacity function, the algorithm succeeded in reducing the visibility of other densities…"
P: "I will recommend a slight adjustment to the opacity function…"
**A: "recognizable"**

R: "The main parts of the lobster, including the claws and the tail, are distinctly visible and the … "
P: "No further adjustments to the opacity function are necessary…"
**A: "clear"**

**Figure 7:** *The results from the LLM-centric AVA for the Boston Teapot and the Lobster in Resin datasets. The structure of interest is the lobster [kla] in both cases. The rendered images are generated from the proposed opacity transfer function. The response includes (R)easoning, (P)lan, and (A)ssessment by the agent. In the top row, the agent suggested the first opacity transfer function that revealed the teapot instead of a lobster and it moved on higher value ranges and successfully detected the lobster, even at a low resolution. In the bottom row, AVA found the lobster, which has higher resolution and occupies a larger space in the volume, and immediately and fine-tuned the result until almost no resin was visible in the visualization output.*

This successful demonstration illustrates the agent's robustness in handling challenging scenarios and its ability to swiftly adapt to different datasets and structures of interest. The detailed results of these agents and their responses are provided in the supplementary material.

### 5.2. Scatterplot Opacity Optimization

Apart from rendering output, from our initial assessment (section 3) the GPT-V has better visual perception for scatterplot compared to all other common information visualization encodings (e.g., parallel coordinate, graph). Therefore, we focus the rest of the case studies on the scatterplot type of visual output. In this section, we examine the optimization of opacity value for scatterplot points to mitigate the occlusion effects from overplotting.

The perceptual base opacity optimization has been explored in the visualization domain [MAF15, MPOW17], either through a data-driven modeling perspective based on user preferences [MAF15] or through a visual perception modeling approach by designing a cost function that captures relevant aspects of the human visual response [MPOW17]. Here we do not aim to directly compare with these existing methods, as a meaningful comparison requires an extensive and controlled study. We hope to use this case study to illustrate how a fundamentally different approach to address the opacity optimization challenge can be obtained by a straightforward adoption of the AVA framework. From the existing study on user preference [MAF15], the relationship between the point opacity and assessment of overplotting level follows an inverse logarithmic relationship, i.e., overplotting only gets better when point opacity gets much lower in the [0.0, 1.0] range. This is crucial prior knowledge that should be incorporated into the design of the agent. Therefore, we adopted the heuristic-centric approach outlined in Section 4, where we encode the logarithmic relationship into our search procedure. At the start of the optimization, we set the initial opacity $O = 1.0$. The floor opacity, i.e., lowest allowable opacity $O_f = 0.0$. For each step, we will update the new opacity as $O' = O_f + (O - O_f)/2$, essentially half the opacity value different between the current opacity and the floor opacity. By providing the model with scatterplot images generated with opacity $O'$ and $O$, we then evaluate which opacity is better suited for the given data. If the new opacity is deemed too low, we then set it as the new floor opacity $O_f$. We continue to iterate to narrow down the selection until the opacity different threshold is reached. For the comparison criteria we use the following prompts: "the chosen opacity allows the viewer to obtain a better understanding of the overall distribution of the underlying data while does not have a serious overplotting issue that can obscure the structure of interest".

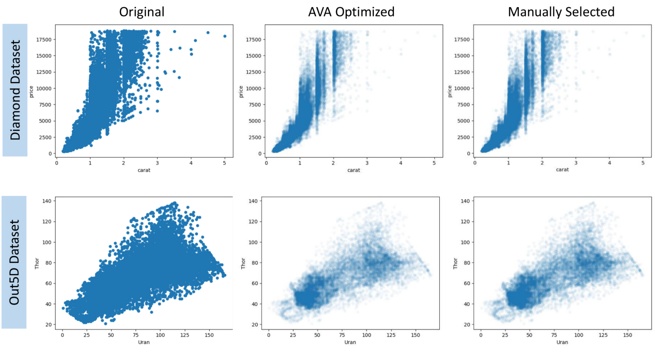


**Figure 8:** *Scatterplot opacity optimization results. The left column shows the original plots with severe overplotting, the middle column shows the agent optimization results, and the right column shows independent manual selection results, there are some minor differences, but the overall results are comparable.*

As shown in Figure 8, each row indicates a different dataset, namely Diamond data [kag], and the Out5D data [xmd]. The first column includes the original scatterplots with the overplotting issue when opacity is 1.0, the middle row is the AVA opacity-optimized scatterplots, and the last row is the human user reference obtained independently from the optimization interface. As we can see, the AVA-generated scatterplot closely matched the user preference.

### 5.3. Dimension Reduction Hyperparameter Tuning

The choice of hyperparameters can greatly impact t-SNE [VdMH08] and UMAP [MHSG18] results. Inappropriate hyperparameters may lead to misleading interpretations of the high-dimensional structure, and they often need to be tuned for a given dataset. Here, we utilize AVA to perform automatic hyperparameter tunning for identifying more suitable hyperparameters, for both single-hyperparameter and multi-hyperparameter cases. Considering the prior knowledge the LLM is likely to have on these common methods, we opt for LLM-centric action planning, where the LLM directly suggests hyperparameters. In Figure 9, we show the single parameter optimization result, where we only optimize the most sensitive parameter for each method, i.e., perplexity for t-SNE, and the neighborhood size for UMAP. We withheld the class label from the agent to use as the ground truth for evaluation. For the t-SNE evaluation criteria (UMAP can be defined in a similar fashion), we use the following prompt: "better embedding can be defined by having good cluster patterns, and a lack of common artifacts due to unsuitable perplexity value." All plots are generated from the data RNA sequence data [TYG*18] with 20 classes. As we can see for the UMAP embedding, the default parameter gives a small number of stringy clusters (a), whereas, in the optimized embedding (b), several classes that were linked together are now separated. For the t-SNE case, there is a less clear advantage for the optimized embedding in terms of cluster separability, however, it does show more compact clusters.

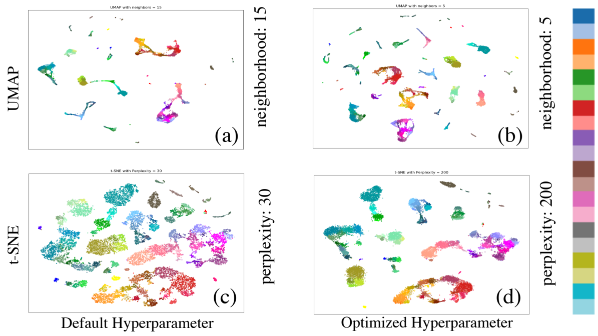


**Figure 9:** *Hyperparameter optimization results. For UMAP, the optimized hyperparameter (b) shows greater class separation compared to the default (a). For t-SNE, (d) shows a similar class separation but with more compact clusters.*

Our experiment with the multi-hyperparameter(up to 5) agent case, however, is largely unsuccessful. As we see the suggested hyperparameters bounce back and forth during the optimization process. This indicates the potential challenge for the agent to explore higher dimensional action space (see more detail in Section 7).

## 6. Feasibility Assessment

One of the major objectives of this work is assessing the feasibility of the proposed paradigm and understanding its robustness and po-

**Table 3:** *Summary of repeated experiments, including, convergence rate (CR), number of steps for convergence (Steps), average spend per step (Step Time), and final output opacity functions lower and high ranges (L,H), the means and standard deviations are shown.*

| Dataset | Structure of interest | CR(*Recognizable*) | CR(*Clear*) | Steps | Step Time (s) | Opacity L,H |
|---|---|---|---|---|---|---|
| Head | Circle of Willis | 100 % | 90 % | 4.2 | 13.1 | $182{\pm}84, 248 \pm 73$ |
| Boston Teapot | Lobster | 60 % | 40 % | 6.3 | 12.61 | $80{\pm}37, 103 \pm 45$ |
| Boston Teapot | Teapot | 100 % | 100 % | 2.2 | 10.96 | $52{\pm}27, 106 \pm 57$ |
| Lobster | Lobster | 100 % | 70 % | 5.2 | 15.7 | $52{\pm}28, 100 \pm 53$ |
| Tooth | Tooth | 100 % | 70 % | 5.4 | 12.3 | $98{\pm}35, 121 \pm 45$ |
| Engine | Two Metal Rings | 100 % | 40 % | 7.4 | 12.3 | $189{\pm}36, 223 \pm 35$ |

tential modes of failure. In this section, we plan to approach the assessment challenges: 1) in terms of low-level mechanism and their consistency and responsiveness, and 2) on the potential usage scenarios of the higher-level scheme from the perspectives of experts in various domains.

### 6.1. Consistency and Interactivity

As the proposed approach utilizes LLMs in a fashion that is quite different from most existing approaches, i.e., involves iteration and refinement, it may not be easy to grasp the basic mechanism of the system with respect to its consistency and interactivity. The LLM API has inherent randomness in its inference process, which means we will likely have slightly different output each time. Therefore, we must investigate the consistency of the overall pipeline in light of the limitations of the existing multi-modal LLMs. As illustrated in Table 3, we performed a systematic evaluation of the agents across multiple datasets (including additional datasets that is not presented in the result section) and conducted a series of ten trials per task as outlined. Here we focus on the volume rendering example as it provides more objective evaluation criteria as described in Section 5, the *recognizable* indicates the agent is able to find the structure of interests, whereas *clear* is defined as a strict condition in which no visual occlusion remains. We terminated the agent if it did not reach clear within 10 iterations. The agents demonstrated a high convergence rate in pinpointing opacity functions that rendered the target structures *recognizable*. In instances of strict *clear* assessment criteria, convergence rates are lower, suggesting that the final triangular opacity function shape might not align perfectly with actual data intensity profiles, leaving residual noise. This was particularly pronounced when in the task of finding the Lobster in the Boston Teapot, where the structure is not fully resolved in the data, making it more difficult to recognize it as a lobster. When achieving ‘recognizable’ outcomes, agents required on average 4-6 iterations, with each taking approximately 12 seconds. For each step, the assessment and planned action will be generated as text output alongside suggested parameters, and visualization will be updated with these new parameters. The variation of the output only indicates the value ranges rather than the visual difference, as we note sometimes the rendered images look very similar even though the opacity function ranges are different. So the visual variation is smaller than what is indicated by the statistics.

### 6.2. Experts Feedback on AVA

Beside the assessment of the feasibility of AVAs on the mechanical level, we also want to understand the potential of the general paradigm and the associated limitations. Given that we did not target a specific application area, we chose to gather informal feedback from experts in key domains. Our feedback collection process involved two senior AI researchers, a professor in medical visualization, and a professor who heads an Institute for Radiological Diagnostics and Intervention. The latter two were selected due to their daily workflow experience with volumetric data and general visualization tasks. For all the feedback sessions, we first demonstrated the use cases of AVAs as described in Section 5, and then conducted unstructured interviews with the experts. While the discussions were mostly open-ended, we did inquire about their overall feedback on how such visualization agents might impact their workflow, and where the potential benefits and limitations of this paradigm are.

**The medical visualization expert said:** *“Their ability to comprehend visual elements and identify structures is indeed impressive, laying a solid foundation for the future development of such agents. With this substantial potential at hand, the utilization of these agents now lies in the hands of visualization researchers, who have the opportunity to harness their capabilities for innovative applications.”* She extended the discussion by suggesting the creation of a generic workflow that can incorporate how visualization experts use the volume rendering tool into the agent prompts to enhance their capabilities.

**The head of the radiology institute said:** *“I’m impressed with the semantic understanding, reasoning capabilities, and high autonomy exhibited by the agents. There is exciting potential to replace trivial visualization tasks that until today require a radiologist.”* He envisioned the use of these agents for double reading in radiology, where two independent radiology reports could be generated to cross-validate diagnoses. However, he remained skeptical about whether the recognition could go beyond simple shapes, for example actually perceiving differences between arteries and veins, as well as extending the visual perception capability to make assessments based on multiple image modalities.

**The senior AI researchers said:** *“1: This can be a very general approach. One additional application I can see this working is for finding more informative views for 3D plots, which I always have trouble with.” “2: The AVA setup can be easily extended to other types of user interfaces beyond just visualization. One thing I am interested to know is how well it handles a larger action space, will the search fail or converge?”* The AI experts believe this is a fundamentally different way to think about data visualization problems and see the connection with embodied agent research. One potential concern they mentioned is whether the action planning can work with a much bigger action space. As a response to their feedback, we extend our dimensionality reduction case studies to include additional experiments with up to 5D space. Overall, the feedback from all the experts underscores the transformative potential of AVA.

## 7. Limitation and Future Work

Despite demonstrating AVA's feasibility through our case studies, it is also essential to discussion the various potential limitations.

**Visual Understanding Capability.** The current model likely works best for natural images and is not optimized for visualization output and particularly detailed and localized structures (as revealed by our preliminary evaluation in Section 3). However, even with its limitation, the first available multi-modal LLM still demonstrates strong recognition capabilities, we believe fine-tuning a model that focuses on visualization output or a more capable future multi-modal model can alleviate some of the current concerns.

**The Persistency of Server-based LLM API.** Since we access the capabilities of GPT-4's API through web-API, we have limited control regarding the actual implementation on the server side. The change in their inference algorithm and the specific version of the model used could have a direct impact on the usability of the tool, i.e., modify the expected and default behavior without our knowledge. Such risk can be mitigated by utilizing local and self-hosted models so that we can control the entire inference pipeline.

**Prompts Engineering and Natural Language.** With the flexibility and usability of natural language, it also brings certain limitations. The precision of AVAs relies on the choice of the prompt. Currently, devising an effective prompt can be a trial-and-error process that lacks a principled paradigm. Moreover, the prompt can also be sensitive with respect to the change in model, i.e., after major model updates, the prompt may need to be adjusted. Additionally, natural language may also have limited expressiveness for describing visualization structures, therefore, the best approach likely needs to combine the advantages of both image and language modalities. This challenge can be partially mitigated through heuristic-centric action planning through explicitly coded search logic, nevertheless, the visual perception still relies on prompts to convey the assessment objectives.

**Large Action Space and Mult-Step Tasks.** The action planning component of AVA is essentially doing an exploration of a potentially high-dimensional action space. The search is guided by the visual assessment, which theoretically can be considered as the loss in a zeroth order optimization scenario [SM22]. Beyond a single action planning step, more complex tasks may require multiple sub-goals to be achieved in a multi-step setup. The generalization to more complex tasks and tools likely requires such capability. We can potentially utilize LLM for decomposing complex tasks into individual steps, however, more research is required for such agents to reach their full potential.

**Future Directions.** Our plan for future work involves a more extensive evaluation of the current models' capabilities in understanding visualization output, expanding on the foundation laid in Section 4. This will be a more general and comprehensive evaluation of multimodal LLM's capabilities on a wide range of visualization perception tasks, which is necessarily a direct evaluation of AVA framework, but on what AVA framework can potentially be applied to. Additionally, we plan to explore different agent setups, including increasing the number of agents and increasing the interactivity of the agent. By implementing multiple independent agents with slightly different definitions, we can offer a means of cross-validation for applications with low error tolerances. So far, we have demonstrated agents employing a closed-loop optimization strategy with intermittent communication with the user. By tuning the level of interactivity, as in a chatbot, we could create an even tighter symbiosis between a human expert and an AI for a joint visualization task. For additional application scenarios. There are many possibilities, as mentioned by one expert we interviewed, adjusting the viewpoint to avoid visual occlusion in 3D visualization or 3D plots can be a great use case. Other use cases include aesthetic optimization [LWG*22]. Lastly, the model's ability for scatterplot understanding can also be utilized to design customized diagnostics metrics for exploratory data analysis.

## 8. Conclusion

The primary objective of this work is to investigate the feasibility of autonomous visualization agents by leveraging the latest development of multi-modal LLM that is capable of visual input. We have demonstrated that we can design robust agents that achieve specific user-defined natural language visualization goals through a visualization-perception-action loop. As multi-modal foundation models continue to advance in capabilities and sophistication, we anticipate a corresponding increase in the capabilities of such agents and generalization for more complex scenarios. In many ways, we are speculating on how the ongoing development of Large Language Models (LLMs) can reshape the landscape of visualization research. With further development, we believe AVAs can eventually serve as "co-experts" alongside the non-expert users to streamline the visualization tool usage. The goal is not to replace the need for visual exploration but rather to enhance the experience and provide alternative avenues to non-experts who may have trouble operating the tool on their own. In conclusion, our research opens exciting possibilities for the future of visualization tool design that aims at the collaboration between humans and AI-driven agents. The fusion of image understanding and language understanding within these multi-modal foundation models holds the potential to fundamentally transform the way we think about visualization and user interaction.

## Acknowledgement

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. The project is supported by LLNL LDRD (23-ERD-029, 23-SI-003). The work is reviewed and released under LLNL-CONF-857838. We thank the experts Prof. Renata Raidou (TU Wien), Prof. Christian Nasel (Medical University Vienna), Dr. Jayaraman Thiagarajan (LLNL), and Dr. Rushil Anirudh (LLNL) for their valuable feedback on the capabilities of our agents.

## References

[CZW*23] Chen Z., Zhang C., Wang Q., Troidl J., Warchol S., Beyer J., Gehlenborg N., Pfister H.: Beyond generating code: Evaluating gpt on a data visualization course. *arXiv preprint arXiv:2306.02914* (2023). 2

[DD19] Dibia V., Demiralp Ç.: Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE computer graphics and applications 39*, 5 (2019), 33–46. 2

[Dib23] Dibia V.: LIDA: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)* (Toronto, Canada, July 2023), Association for Computational Linguistics, pp. 113–126. URL: https://aclanthology.org/2023.acl-demo.11, doi:10.18653/v1/2023.acl-demo.11. 2

[Fil13] Filiba T.: Rpyc (remote python call), 2013. Python library for remote procedure calls. URL: https://rpyc.readthedocs.io/en/latest/. 6

[FZF*23] Feng W., Zhu W., Fu T.-j., Jampani V., Akula A., He X., Basu S., Wang X. E., Wang W. Y.: Layoutgpt: Compositional visual planning and generation with large language models. *arXiv preprint arXiv:2305.15393* (2023). 3

[GFC04] Ghoniem M., Fekete J.-D., Castagliola P.: A comparison of the readability of graphs using node-link and matrix-based representations. In *IEEE Symposium on Information Visualization* (2004), pp. 17–24. doi:10.1109/INFVIS.2004.1. 4

[hea] Head dataset. http://www.celebisoftware.com/Dataset.aspx?catId=3. Accessed: [Your Access Date Here]. 6

[Hun07] Hunter J. D.: Matplotlib: A 2d graphics environment. *Computing in science & engineering 9*, 03 (2007), 90–95. 2

[JIS*23] Jia D., Irger A., Strnad O., Bjorklund J., Ynnerman A., Viola I.: Voice: Visual oracle for interaction, conversation, and explanation. *arXiv preprint arXiv:2304.04083* (2023). 2

[kag] Kaggle diamond dataset. https://www.kaggle.com/datasets/shivam2503/diamonds/data. Accessed: YYYY-MM-DD. 8

[kla] Open scientific visualization datasets. https://klacansky.com/open-scivis-datasets/. Accessed: 3/28/2024. 7

[LWG*22] Loos S., Wolk S. v. d., Graaf N. d., Hekkert P., Wu J.: Towards intentional aesthetics within topology optimization by applying the principle of unity-in-variety. *Structural and Multidisciplinary Optimization 65*, 7 (2022), 185. 10

[LWZ*21] Li H., Wang Y., Zhang S., Song Y., Qu H.: Kg4vis: A knowledge graph-based approach for visualization recommendation. *IEEE Transactions on Visualization and Computer Graphics 28*, 1 (2021), 195–205. 2

[MAF15] Matejka J., Anderson F., Fitzmaurice G.: Dynamic opacity optimization for scatter plots. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), pp. 2707–2710. 8

[MHSG18] McInnes L., Healy J., Saul N., Grossberger L.: Umap: Uniform manifold approximation and projection. *Journal of Open Source Software 3*, 29 (2018), 861. 8

[MNES22] Mitra R., Narechania A., Endert A., Stasko J.: Facilitating conversational interaction in natural language interfaces for visualization. In *2022 IEEE Visualization and Visual Analytics (VIS)* (2022), IEEE, pp. 6–10. 2

[MPOW17] Micallef L., Palmas G., Oulasvirta A., Weinkauf T.: Towards perceptual optimization of the visual design of scatterplots. *IEEE transactions on visualization and computer graphics 23*, 6 (2017), 1588–1599. 8

[NSS20] Narechania A., Srinivasan A., Stasko J.: NL4DV: A Toolkit for generating Analytic Specifications for Data Visualization from Natural Language queries. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* (2020). doi:10.1109/TVCG.2020.3030378. 2

[Ope23] OpenAI: Gpt-4 vision. https://openai.com/research/gpt-4v-system-card, 2023. Accessed: [Insert date of access here]. 3, 5

[SASW96] Spitzer V., Ackerman M. J., Scherzinger A. L., Whitlock D.: The visible human male: a technical report. *Journal of the American Medical Informatics Association 3*, 2 (1996), 118–130. 3

[SK19] Sullivan C., Kaszynski A.: Pyvista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (vtk). *Journal of Open Source Software 4*, 37 (2019), 1450. 2

[SM22] Slavin I., McKenzie D.: Adapting zeroth order algorithms for comparison-based optimization. *arXiv preprint arXiv:2210.05824* (2022). 10

[SMV23] Surís D., Menon S., Vondrick C.: Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128* (2023). 3

[SMWH17] Satyanarayan A., Moritz D., Wongsuphasawat K., Heer J.: Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization & Computer Graphics (Proc. InfoVis)* (2017). URL: http://idl.cs.washington.edu/papers/vega-lite, doi:10.1109/tvcg.2016.2599030. 2

[SWW*23] Song C. H., Wu J., Washington C., Sadler B. M., Chao W.-L., Su Y.: Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 2998–3009. 1

[TAB*23] Team G., Anil R., Borgeaud S., Wu Y., Alayrac J.-B., Yu J., Soricut R., Schalkwyk J., Dai A. M., Hauth A., et al.: Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805* (2023). 5

[TIH] Terarecon Inc MERL B., Hospital W.: Teapot dataset. http://www.gris.uni-tuebingen.de/areas/scivis/volren/datasets/data/BostonTeapot.raw.gz. Accessed: 3/28/2024. 3

[TYG*18] Tasic B., Yao Z., Graybuck L. T., Smith K. A., Nguyen T. N., Bertagnolli D., Goldy J., Garren E., Economo M. N., Viswanathan S., et al.: Shared and distinct transcriptomic cell types across neocortical areas. *Nature 563*, 7729 (2018), 72–78. 8

[VdMH08] Van der Maaten L., Hinton G.: Visualizing data using t-sne. *Journal of machine learning research 9*, 11 (2008). 8

[WBZ*23] Wu Q., Bansal G., Zhang J., Wu Y., Zhang S., Zhu E., Li B., Jiang L., Zhang X., Wang C.: Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155* (2023). 1

[WMF*23] Wang L., Ma C., Feng X., Zhang Z., Yang H., Zhang J., Chen Z., Tang J., Chen X., Lin Y., et al.: A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432* (2023). 1, 3

[WXJ*23] Wang G., Xie Y., Jiang Y., Mandlekar A., Xiao C., Zhu Y., Fan L., Anandkumar A.: Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv: Arxiv-2305.16291* (2023). 1, 3

[WZW*23] Wang L., Zhang S., Wang Y., Lim E.-P., Wang Y.: Llm4vis: Explainable visualization recommendation using chatgpt. *arXiv preprint arXiv:2310.07652* (2023). 2

[xmd] Xmdvtool homepage. http://davis.wpi.edu/xmdv/datasets.html. Accessed: [Access Date]. 8

[YLL*23] YANG Z., LI L., LIN K., WANG J., LIN C.-C., LIU Z., WANG L.: The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421 9* (2023). 3

[ZDS*23] ZHANG H., DU W., SHAN J., ZHOU Q., DU Y., TENENBAUM J. B., SHU T., GAN C.: Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485* (2023). 4

[ZLL*22] ZONG J., LEE C., LUNDGARD A., JANG J., HAJAS D., SATYANARAYAN A.: Rich screen reader experiences for accessible data visualization. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 15–27. 2

[ZSC*23] ZHANG S., SUN P., CHEN S., XIAO M., SHAO W., ZHANG W., CHEN K., LUO P.: Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601* (2023). 3

[ZWLQ23] ZHANG S., WANG Y., LI H., QU H.: : Adaptive and explainable visualization recommendation for tabular data. *IEEE Transactions on Visualization and Computer Graphics* (2023). 2