Benthic: Perceptually Congruent Structures for Accessible Charts and Diagrams

Catherine Mei*
meic1212@mit.edu
MIT CSAIL
Cambridge, MA, USA

Josh Pollock*
jopo@mit.edu
MIT CSAIL
Cambridge, MA, USA

Daniel Hajas
dh256@mit.edu
MIT CSAIL
Cambridge, MA, USA

Jonathan Zong
jzong@mit.edu
MIT CSAIL
Cambridge, MA, USA

Arvind Satyanarayan
arvindsatya@mit.edu
MIT CSAIL
Cambridge, MA, USA

ABSTRACT

Graphical representations — such as charts and diagrams — have a visual structure that communicates the relationship between visual elements. For instance, we might consider two elements to be connected when there is a line or arrow between them, or for there to be a part-to-whole relationship when one element is contained within the other. Yet, existing screen reader solutions rarely surface this structure for blind and low-vision readers. Recent approaches explore hierarchical trees or adjacency graphs, but these structures capture only parts of the visual structure — containment or direct connections, respectively. In response, we present Benthic, a system that supports *perceptually congruent* screen reader structures, which align screen reader navigation with a graphic's visual structure. Benthic models graphical representations as *hypergraphs*: a relaxed tree structure that allows a single *hyperedge* to connect a parent to a set of children nodes. In doing so, Benthic is able to capture both hierarchical and adjacent visual relationships in a manner that is *domain-agnostic* and enables *fluid* (i.e., concise and reversible) traversal. To evaluate Benthic, we conducted a study with 15 blind participants who were asked to explore two kinds of graphical representations that have previously been studied with sighted readers. We find that Benthic's perceptual congruence enabled flexible, goal-driven exploration and supported participants in building a clear understanding of each diagram's structure.

CCS CONCEPTS

• **Human-centered computing → Visualization systems and tools**; **Accessibility systems and tools**.

KEYWORDS

Diagramming, Visualization, Accessibility, Screen Reader, Relations

ACM Reference Format:
Catherine Mei, Josh Pollock, Daniel Hajas, Jonathan Zong, and Arvind Satyanarayan. 2025. Benthic: Perceptually Congruent Structures for Accessible Charts and Diagrams. In *The 27th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '25), October 26–29, 2025, Denver, CO, USA.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3663547.3746342

*Authors contributed equally to this research.

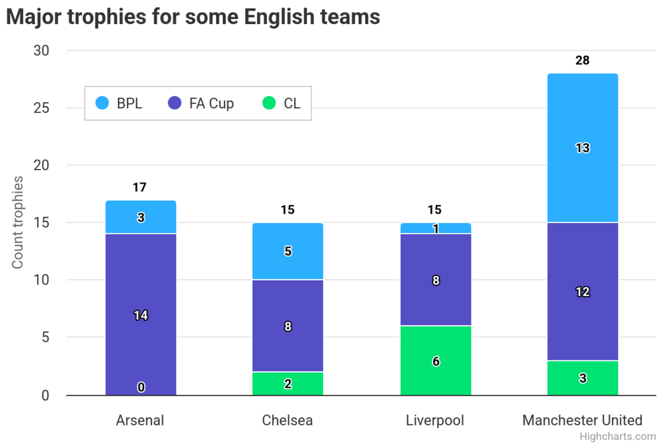
ASSETS '25, October 26–29, 2025, Denver, CO, USA

ACM ISBN 979-8-4007-0676-9/2025/10
https://doi.org/10.1145/3663547.3746342

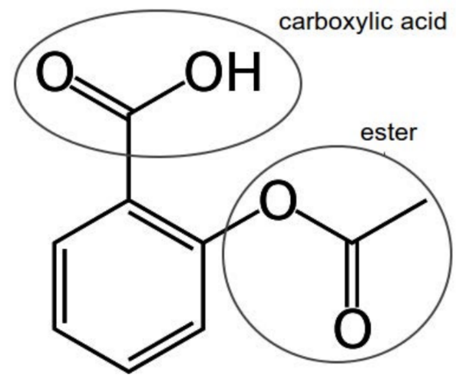
1 INTRODUCTION

Graphical representations (charts, infographics, diagrams, mathematical formulae, etc.) are more than just a collection of shapes, symbols, and text. Rather, they make judicious use of visual structures known as Gestalt grouping principles [33, 35] to convey relationships between visual elements. For instance, consider a stacked bar chart such as the one shown in Figure 1a — in an image filled with visual structure, the *uniform spacing* between the bars indicates four distinct categories of data; the *aligned colors* for the bar segments depict an additional cross-cutting category; and the *proximity* of the textual annotations to the bars and bar segments conveys an association that we read as labeling. Similarly, consider a diagram of an aspirin molecule as shown in Figure 1b — the linear *connections* reflect single- or double-bonds between atoms in the molecule while the circles *enclose* functional groups which are, once again, labeled through spatially *proximal* textual annotations.

While sighted audiences can access this visual structure to read, understand, and interpret a graphical representation, these same structural cues are typically inaccessible to blind and low-vision (BLV) readers. Alt text (short for alternative text) can provide a useful high-level overview of the information in a graphic; but, without access to a graphic's underlying structure, it is more difficult for BLV people to interpret the visualized information for themselves. Recent efforts such as Olli [2], Chart Reader [32], and Data Navigator [9] have begun to explore how to make this structure accessible to screen readers. All three systems primarily target data visualizations, with Olli and Chart Reader using tree structures to represent visualized data while Data Navigator uses node-link graphs. While these approaches are meaningful improvements over unstructured textual descriptions, they capture only part of the structural logic present in graphical representations: tree structures reflect *hierarchical* relationships but make it difficult to express cross-cutting associations such as labels and annotations; on the other hand, graph structures better capture these *adjacent* relationships but can't represent information hierarchies in a consistent way. But, as we saw with our examples in Figure 1, graphical representations make extensive use of both kinds of relationships, and



(a) A stacked bar chart comparing the number of major trophies won by 4 teams across 3 contests competitions. Sourced from Highcharts [16].



(b) A structural diagram of an aspirin molecule with its two functional groups. Sourced from Sorge et al. [29].

Figure 1: Two graphical representations illustrating visual groupings and spatial structure.

being able to traverse them — independently and jointly — is crucial to developing a robust, high fidelity understanding of the information the graphic conveys. For instance, with the chemical molecule, a BLV reader is likely to want to traverse the adjacencies of the atomic bonds while also drilling into and out of the hierarchies of the functional groups.

To address this gap, we introduce Benthic, a screen reader system that provides BLV people affordances for reading and interpreting the structure of graphical representations that are commensurate to those sighted readers regularly use. To provide these affordances, we aim to design *perceptually congruent* screen reader structures that parallel the structures of the graphics they represent (Section 4). For instance, if graphical elements are visually grouped (e.g., explicitly with connecting lines, through aligned colors such as the segments of the stacked bar chart, or through spatial clustering or enclosure as in the chemical molecule) then the screen reader output should preserve that grouping in both content and navigational flow. Perceptually congruent structures have two additional desirable properties. First, they are *domain-agnostic*: many aspects of visual structure (such as proximity, alignment, and containment) apply across a broad range of graphical domains including charts, diagrams, notations, and maps. By targeting these cross-cutting visual groupings, Benthic builds a foundation for supporting many different domains with a shared navigation interface. Second, perceptually congruent structures facilitate *fluid traversal* — akin to how sighted readers can quickly shift their attention to different parts of a graphic [3, 17], Benthic is designed to allow BLV readers to rapidly switch between groups and easily reverse their actions.

Benthic achieves perceptual congruence with two components: a *hypergraph* that encodes the visual structure of a graphical representation (Section 5.1), and a *screen reader interface* that renders the hypergraph as interactive text descriptions (Section 5.2). Hypergraphs generalize both trees and graphs. They consist of nodes connected by *hyperedges*. Unlike a tree, where each node must have a single parent, hypergraph nodes can have any number of parents. Unlike a graph edge, which must connect exactly two nodes, a hyperedge connects a set of nodes. This flexibility allows hyperedges to express both hierarchical and adjacent relationships: in Benthic's hypergraph, each hyperedge connects a parent to a set of children nodes that are considered adjacent.

Benthic's screen reader interface allows BLV users to traverse hypergraphs by moving laterally across adjacent nodes, into child nodes, or up to parent nodes. This interaction model supports navigation through both hierarchical and adjacent relationships, mirroring the structure of the original graphic. We designed this interface over a six-month iterative co-design process with our co-author Daniel Hajas, a blind researcher who has both lived and professional experience with designing assistive technologies for diagrammatic communication. This process helped reveal that a key complexity with designing for Benthic's hypergraph is that nodes may have multiple parents; thus, during upward traversal, the interface needs to present all available parent contexts and let users choose a path. This design involved trade-offs between traversal fluidity and structural clarity, which we discuss in Section 5.2.4.

To evaluate Benthic, we conducted a user study with 15 blind participants who explored structured graphical representations using our system (Section 6). Participants completed tasks with pulley diagrams and bar charts adapted from prior studies on diagrammatic reasoning and graphical perception. In the final condition, we replicated an experiment by Boger & Franconeri which tested whether differences in Gestalt groupings affected sighted readers' abilities to identify surprising data relationships [3]. We found that Benthic enabled flexible, goal-driven exploration and supported participants in building a clear understanding of the diagram's structure, with participants more likely to detect data anomalies when the grouping structure aligned with their interpretive strategies — mirroring the effects observed in sighted users.

2 RELATED WORK

A wide body of research has explored how to make graphical representations accessible to BLV readers. Benthic draws from prior approaches to structured navigation of graphical representations, and applies ideas about spatial relations developed in visual and tactile graphics to its screen reader interface.

2.1 Structured Access to Graphical Representations

A growing body of literature suggests that screen reader users benefit from structured representations of charts and diagrams that offer navigation through multiple descriptions. Studies have shown that BLV readers share information-seeking goals with sighted readers of graphical representations: they typically want to acquire an initial overview followed by details [26], akin to the information-seeking mantra of "overview first, zoom and filter, and details on demand" [27]. Conventional alt text is typically a static summary, without any ability to focus on specific aspects of a graphic or control information granularity. As a result, it does not provide sufficient affordances to support this exploratory reading process. While some libraries have added accessibility features for graphical representations like maps and diagrams [18, 20], they are typically limited to tagging existing graphic elements with ARIA labels. The result is that users are limited to linear navigation between labels in a pre-set order.

In response, recent research systems have developed structured textual interfaces for accessible visualization that support multiple ways for users to navigate [38]. These systems can be grouped into two main categories: tree-based and graph-based. A primary example of a tree-based system is Olli [2], which enables users to read summary information at the root of the tree structure, and traverse deeper into branches to acquire details-on-demand about different parts of a chart. This type of system has become popular for data visualization [13, 30, 32, 39], but is limited to statistical graphics — it is less amenable to maps or diagrams. In contrast, an example of a graph-based system is Data Navigator [9], which enables users to flexibly navigate across visual elements on a chart using connections between nodes and edges. Beyond data visualization, researchers have developed navigational systems targeted to domain-specific diagrams like chemical molecules [28, 29], but there is a lack of more general accessible interfaces for diagramming.

Though these existing approaches improve upon static alt text and simple approaches to tagging graphics, they force creators of accessible graphical representations into a tradeoff. Trees support hierarchy but not adjacency (e.g., shared labels or linked elements), whereas graphs capture adjacency but lack hierarchy (e.g. containment relationships). Benthic introduces a unified *hypergraph* representation that captures both hierarchies and adjacencies, with support for cross-cutting groupings. The resultant navigational structures support information-seeking for users by enabling navigation that corresponds to the visual structure of a graphic, while affording more control over exploration and information granularity. Because Benthic is not limited to domain-specific diagrams, it supports consistent, flexible exploration across a wider variety of graphical representations.

2.2 Spatial and Relational Access to Graphical Representations

Graphical representations such as diagrams, charts, and schematics communicate structure not only through explicit elements, but also through spatial layout and perceptual grouping. Gestalt principles—such as proximity, similarity, and enclosure—enable sighted users to quickly infer relationships among elements [33, 35]. These visual strategies support diagrammatic reasoning by reducing cognitive load and revealing meaningful groupings [17]. Cheng's studies of pulley diagrams [6] further illustrate how spatially organized representations outperform text in supporting inference.

Indeed, Gestalt principles are also essential in design for BLV readers [4, 10]: Gallace and Spence [12] demonstrate that tactile graphics rely on proximity and enclosure for structure recognition. Tactile graphics specialists have long emphasized that diagram comprehension in tactile form requires careful attention to spatial separation, grouping, and simplification [15]. Unlike visual perception, which supports holistic recognition, tactile exploration is sequential—building understanding part-by-part. As a result, tactile graphics often rely on principles such as spatial layering, consistent texture use, and logical segmentation to convey relationships and hierarchy. These insights echo those found in mid-air haptics, where dynamic tactile cues support spatial grouping and recognition [14].

The spatial organization provided by graphics is also valuable to BLV readers when delivered through description. Guidelines for describing interactive scientific graphics to BLV readers recommend using spatial descriptors—such as "left to right," "top to bottom," or "clockwise from the top"—or other references to a known location [38] to help users mentally reconstruct the layout of a chart or diagram [24], consistent with findings on BLV spatial awareness [7, 19, 37]. These spatial anchors provide reference points that reduce cognitive load and support orientation, particularly when users cannot perceive the graphic holistically.

Unfortunately, few screen reader systems explicitly encode or expose such spatial and relational structures in navigable form. Benthic addresses this gap by structurally encoding both adjacency and hierarchy, which could be augmented in future work with spatial metadata to enable similar orientation cues across modalities.

3 USAGE SCENARIO

To demonstrate how Benthic enables BLV users to explore structured graphical representations, we walk through a scenario in which a screen reader user named Marina navigates the stacked bar chart from Figure 1 (sourced from Highcharts)[1]. The chart depicts the number of trophies won by four football teams, Arsenal, Chelsea, Liverpool, and Manchester United, across three contests. Figure 2 depicts Marina's sequence of steps, which we reference throughout this section, and Table 1 provides a complete list of Benthic's keyboard commands.

Navigating between and into nodes. After Marina loads the Benthic screen reader interface for the stacked bar chart, she presses the `h` key, which acts as a "home" key in Benthic. This focuses the screen reader on the root node of the hypergraph, which provides a summary of the graphic, including its title, the chart type, and a

[1] This chart is also used to demonstrate Data Navigator [9]. We provide a more detailed compared Benthic to Data Navigator in Section 5.1.2.

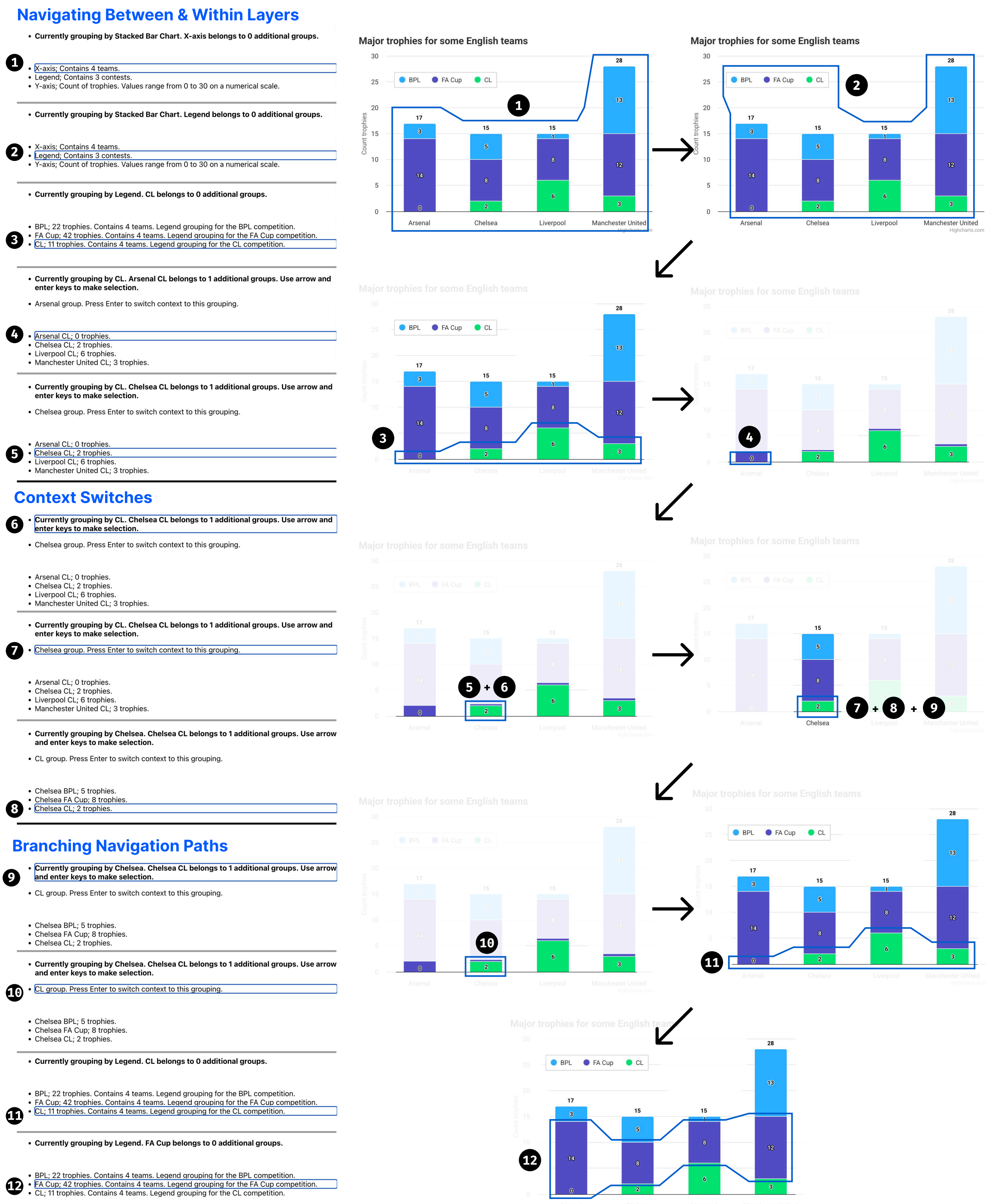


Figure 2: Walkthrough of a usage scenario showing how Marina navigates a stacked bar chart using Benthic. The left-hand side shows 12 interface states as Marina moves between different groupings in the chart and context switches during her exploration. The right-hand side summarizes this sequence using numbered illustrations that correspond to each of the 12 steps. In each illustration, the blue outlined box shows Marina's current location, while the shaded area highlights the context she is currently exploring. Other parts of the chart are visually faded to reduce distraction. Together, these views show how Benthic reveals structure step by step, helping Marina build understanding as she explores the stacked bar chart.

description of the chart's axes. From here, Marina presses `shift` and `down arrow` (↓) to step into the chart. ❶ The screen reader focuses on the `x-axis`, and reads that it contains four teams. ❷ Pressing `right arrow` (→), Marina moves laterally to the `legend` node and hears that the `legend` contains three contests. Another → press brings her to the `y-axis` node.

At this point, Marina wants to explore the data grouped by contest. She presses `left arrow` (←) to return to the `legend` and then `shift + ↓` to descend further into the `legend`. The screen reader is directed to the `BPL` node and provides a description of the node. ❸ Marina presses → to move to the `CL` group and hears a similarly structured description.

❹ While the screen reader is focused on the `CL` node, Marina presses `shift + ↓` to move down a layer and see the breakdown of trophies won by each team at the `CL` contest. The screen reader focus moves to `Arsenal CL`, and Marina hears a description of the node and the number of trophies won by the `Arsenal` team at the `CL` contest. ❺ Again, Marina presses → to explore the nodes in this layer, where she finds the `Chelsea CL` node.

Switching parent contexts. At this point, Marina wants to examine how the `Chelsea` team performed in the 3 contests. Using existing tools like Olli, Marina would have to ascend the layers of the stacked bar chart, cross over to the branch in the structure that represents the `x-axis`, and descend that branch to find the data for the `Chelsea` team. With Benthic, Marina can quickly perform a *context switch* to regroup the data by team instead of contest. ❻ With the screen reader still focused on the `Chelsea CL` node, she presses `shift + ↑` to navigate up to a *parent context layer*. The screen reader reads that the current node, `Chelsea CL`, belongs to two groups: `CL` and `Chelsea`.

❼ Marina presses → to switch the parent context from `CL` to `Chelsea`, then presses `enter` to confirm her selection.

❽ Benthic updates, reorienting the local context around Marina's cursor focus (the `Chelsea CL` node) to group the data by team instead of contest. Thus, she can now press ← and → to navigate to different nodes for `Chelsea`'s performance at the `BPL`, `FA Cup`, and `CL` contests. This ability to fluidly regroup the data allows Marina to quickly perform analytical tasks, such as comparative analysis across various dimensions.

Navigating out of nodes. After learning more about `Chelsea`'s performance across the 3 contests, Marina noticed that Chelsea won the most trophies at the `FA Cup`. She is curious how `Chelsea`'s performance in the `FA Cup` compares to that of the other teams, so she wants to navigate to the `FA Cup` node. Marina knows that she can find the `Chelsea FA Cup` data point and perform a *context switch*. However, she decides to take an alternate navigation path. ❾ While the screen reader is focused on `Chelsea CL`, Marina presses `shift + ↑` to navigate up to a parent context layer. The screen reader announces that the data is currently grouped by `Chelsea`, and that `Chelsea CL` belongs to one additional grouping (`CL`). ❿ Marina presses → to move to the `CL` node, but this time, rather than pressing `enter` to perform a *context switch*, she presses `shift + ↑` again to move out of the `Chelsea CL` node up to the `CL` node. ⓫ The screen reader is now focused on `CL`. Using ← and →, Marina explores the other nodes at this level: `BPL` and `FA Cup`.

⓬ Marina navigates to the `FA Cup` node and presses `shift + ↓` to view the breakdown of trophies won by each team.

This usage scenario illustrates how Benthic gives screen reader users greater control over how they explore graphical representations. Marina's ability to reorient the structure around a data point and switch between different organizational views—such as grouping by contest or by team—shows how Benthic supports navigation that reflects the overlapping visual relationships in the chart. Unlike prior approaches, Benthic lets users decide how information is grouped and traversed, making tasks like comparison and pattern recognition more efficient and intuitive.

4 GOAL: PERCEPTUALLY CONGRUENT SCREEN READER STRUCTURES

In this paper, we say that a screen reader structure is **perceptually congruent** with a graphical representation if it mirrors the visual structure of that graphic (as described in Section 4.1). Benthic's main design goal is to construct perceptually congruent representations that can be navigated by screen readers. This gives BLV users access to affordances similar to those provided by graphical representations.

4.1 Motivation: Graphical Representations as Data Structures for Computation

Graphical representations such as diagrams and data visualizations are more than just visual aids—they serve as data structures that actively support and scaffold cognitive processes. As Simon and Larkin argue in "Why a Diagram is (Sometimes) Worth Ten Thousand Words" [17], diagrams are not merely illustrations but structured representations that can directly aid problem-solving by reducing the cognitive load associated with abstract reasoning. When sighted users engage with diagrams for analytical purposes, they aren't simply processing visual features like shapes, colors, or text. Instead, they are interacting with spatially organized information that concretizes relationships, encodes constraints, and supports perceptual inference [17, 21, 31].

Graphical representation use *Gestalt grouping principles* to structure shapes [33, 34]. Figure 3 shows examples of some of the Gestalt grouping principles that are most salient in charts and diagrams: spatial proximity, connectedness, alignment, and containment. Where shapes represent individual pieces of data or elements, Gestalt grouping principles represent relationships between those elements. For example, a letter in a diagram of a chemical molecule might represent an atom, whereas a line connecting two letters might represent the bond between two atoms. This supports readers in several ways. First, compared to purely textual representations, graphics make structure explicit. For example, a circuit diagram reveals connections between components much more readily than a block of text that verbally references other components using phrases like "component A" or "this component." Such references require a reader to revisit previous parts of the description to make sense of the relationships or else keep many relationships in working memory [17]. Second, Gestalt groupings support perceptual inference by leveraging the human visual system's efficiency with perceiving those groupings [31, 34], transforming complex logical operations into quick perceptual judgments. Finally, diagrams help

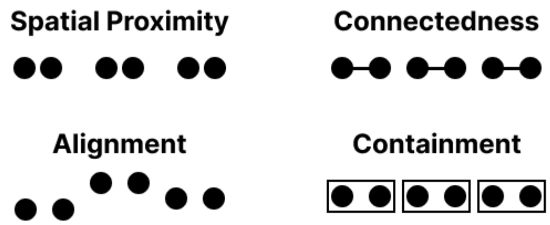


Figure 3: Some important Gestalt grouping principles for graphical representations: spatial proximity, connectedness, alignment, and containment.

users construct accurate mental models by mirroring the structure of the objects and relationships they represent [33]. This is especially critical in domains like physics, mechanics, and geometry, where spatial organization is important for conceptual understanding.

Yet while graphical representations offer powerful computational affordances for sighted users, these benefits remain largely inaccessible to blind and low vision (BLV) users—not because graphical representations are visual, but because existing systems fail to represent and operationalize the underlying structure that makes graphical representations effective in the first place. Current screen reader experiences typically flatten or abstract away the spatial and relational organization that supports inference, pattern recognition, and reasoning. Even approaches like Olli [2, 38] and Data Navigator [9] fall short of capturing the inherent structure embedded in visual representations. As a result, BLV users do not have access to the affordances that have been shown to enhance computational efficiency for sighted users. To bridge this gap, we must design systems that make these structures accessible—not just the visual content itself, but the cognitive scaffolding it provides.

4.2 Design Heuristics

The primary design goal of Benthic is to create screen reader structures that are **perceptually congruent** to the visual structures sighted readers perceive in graphical representations. There are two qualities of perceptually congruent structures that we used to guide the design and evaluation of Benthic: domain-agnostic structure and fluid traversal.

Domain-agnostic structure. Sighted readers understand graphical representations through a combination of low-level perceptual principles — such as the Gestalt groupings shown in Figure 3 — and higher-level, context-specific cues like cultural interpretations of colors and symbols. In designing Benthic, we focused on capturing the former as they apply across diagram types regardless of the subject matter [3, 17, 22, 33]. As these relationships are grounded in how the visual system naturally organizes information, they offer a robust foundation for designing screen reader interfaces that generalize across graphical representations. Domain-agnostic structures can later be layered with context-specific information such as natural language descriptions of symbols or regions of a visualization. Additionally, a domain-agnostic structure provides a consistent interface that helps users build expectations and transferrable skills across graphical representations, letting them focus on the structure at hand.

Fluid traversal. Another guiding heuristic is the need to support a navigation style that mirrors the flexibility of sighted reader's attention. For sighted readers, the eye naturally shifts across a visual layout without restriction — jumping between related groups, zooming out to grasp global structure, or drilling into detail [3, 17]. This type of navigation is not linear, and allows readers to form and revise hypotheses as they explore. An important aspect of fluid traversal is ease of movement — both in being able to move in different directions and in how much effort each movement requires. Traversal should be *reversible*: users should be able to explore multiple paths in the structure while revisiting prior locations and undoing navigation actions as needed. But fluidity also includes the number of steps or actions required to move between parts of a diagram. For sighted users, shifting visual attention requires little conscious effort; for screen reader users, however, each movement requires deliberate keyboard actions. The more key presses or structural detours required, the less fluid the experience becomes. Hotkeys and shortcuts can help reduce this friction, but ultimately, the underlying structure must support direct, efficient traversal. When it doesn't, traversal becomes *viscous* — slow, effortful, and prone to disorientation.

5 THE BENTHIC SYSTEM

Benthic consists of two core components: (1) the Benthic hypergraph (Section 5.1), an intermediate representation that is perceptually congruent with the structure of charts and diagrams; and, (2) a screen reader interface for navigating that structure (Section 5.2).

5.1 The Benthic Hypergraph

To design an intermediate representation that mirrors how graphical content is structured, we drew on both theoretical and practical foundations: we were inspired by Larkin & Simon, who view diagrams as data structures that scaffold reasoning through spatial and structural organization [17], as well as Bluefish [23], a diagramming library that uses a relational scenegraph to encode visual semantics. Though these structures were developed with sighted readers in mind, we found they bear striking resemblance to the structures in screen reader tools for data visualizations like Olli [2, 38] and Data Navigator [9], yet address limitations of both of them.

Benthic uses *hypergraphs*, which generalize both Olli's tree structure and Data Navigator's node-link graphs. Rather than increasing expressive power, Benthic's hypergraph imposes constraints that ensure the resulting representation is perceptually congruent and consistent in how it encodes relationships. First, hypergraphs allow elements to participate in multiple groupings without duplication, preserving the visual affordances of overlapping structures (e.g., the stacked bar chart from Figure 1 that groups elements both by team and by competition). Second, hypergraphs enforce a structural invariant: children have a shared parent if and only if they are neighbors. This invariant guarantees that traversal paths are reversible and group membership is unambiguous — unlike in graph-based representations where such relationships must be explicitly specified. Third, hypergraphs avoid the complexity of tagging edges with semantic roles or introducing intermediary nodes, making the structure easier to reason about for both developers and users.

We illustrate these benefits using the stacked bar chart in Figure 1a, which contains two perceptual structures: (1) grouping by team, encoded through spatial proximity within each stack, and (2) grouping by competition, encoded through color alignment across stacks. To motivate the need for hypergraphs, we begin by showing how trees and graphs fall short in capturing the perceptual structure of this chart.

5.1.1 Tree Representation (Figure 4a). Consider a tree-structured representation of this chart, similar to the one produced by Olli, in Figure 4a. We use program function call syntax to express the hierarchical relationships of the Olli tree structure. For example,

```
bpl(
  "bpl1-legend",
  "bpl2-legend",
  "bpl3-legend",
  "bpl4-legend")
```

denotes that `bpl` is a parent with four children.

The tree structure encodes two separate subtrees: one for the legend (grouping bars by competition) and one for the x-axis (grouping by team). Because nodes in a tree structure can only have one parent, the same visual element must be duplicated in both subtrees — for example, Arsenal's BPL bar appears as both `bpl1-legend` and `bpl1-x`. This duplication breaks *perceptual congruence*: even though these nodes represent the same visual shape, they are treated as distinct elements. To switch from the `bpl` grouping to the `arsenal` grouping, a user must traverse up to the `title` node and down to `arsenal`. This makes navigation *viscous* and unintuitive.

Olli's structure is also *domain-specfic* — it is specialized for chart representations and does not generalize well to other diagram types. For example, chemical molecules are full of overlapping relationships that do not map cleanly to a tree structure.

5.1.2 Graph-based Representation (Figure 4b). Data Navigator represents the same chart using a graph structure with two kinds of edges, `parents` and `neighbors`, that represent hierarchical and adjacent relationships, respectively[2]. We represent these relations using function call syntax where the only two functions are `parent` and `neighbor`. In a tree structure, a parent node always induces neighbor relationships among its children, which Data Navigator makes explicit. For example, Data Navigator would represent the Olli node from Section 5.1.1 like this:

```
parent("bpl", "bpl1-legend")
parent("bpl", "bpl2-legend")
parent("bpl", "bpl3-legend")
parent("bpl", "bpl4-legend")
neighbor(
  "bpl1-legend",
  "bpl2-legend",
  "bpl3-legend",
  "bpl4-legend")
```

[2]We have streamlined the Data Navigator representation for ease of comparison. In the actual implementation, all edges have exactly two children and the distinction between "parent" and "neighbor" edges is implicit in the "navigationRule" field of its data structure.

By making these relations explicit, Data Navigator allows nodes to participate in multiple hierarchical and adjacent relationships. As a result, a node such as `bpl1` is reachable via a `neighbor` edge from both `fa1` and `bpl2` (Figure 4b). Thus for adjacent relationships, Data Navigator is both more *congruent* and more *fluid* than the tree structure.

But this flexibility comes at a cost. Splitting `parent` and `neighbor` edges introduces the potential for structural inconsistencies. Data Navigator does not guarantee that nodes which are adjacent also share a parent, like Olli does. For example, while the Olli node translates to five Data Navigator relations, the structure presented in the original Data Navigator prototype provides only two:

```
parent("bpl", "bpl1")
neighbor("bpl1", "bpl2", "bpl3", "bpl4")
```

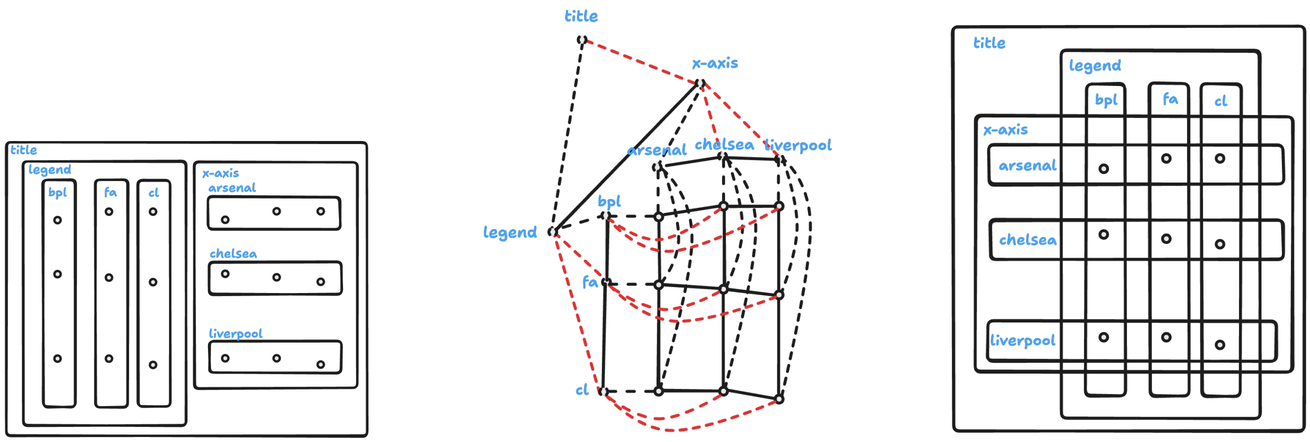
As a result, if a Data Navigator user traverses from `bpl` down to `bpl1` and then across to `bpl2`, they cannot return directly to `bpl`, because this parent relationship is missing. This breaks both *perceptual congruence* and *fluidity* (specifically *reversibility*) for hierarchies. Data Navigator is missing several of these relations, which we represent with red edges in Figure 4b.

5.1.3 Hypergraph Representation (Figure 4c). Whereas Olli uses trees and Data Navigator uses node-link graphs, Larkin & Simon and Bluefish encode graphics as *hypergraphs*. Hypergraphs generalize both trees and graphs, maintaining the close connection between `parent` and `neighbor` relationships from trees, while allowing adjacent connections like graphs. This duality makes hypergraphs well-suited for representing diagrams that involve overlapping groupings and layered visual structure.

Intuitively, a hypergraph can be understood as a flexible hierarchical structure in which nodes are allowed to participate in multiple groupings simultaneously. Like in a tree, a hypergraph supports nested structure through parent-child relationships. However, unlike in a tree, the same node can appear under multiple parents. This addresses Olli's limitation when representing overlapping groups like the contests and teams.

Hypergraphs also offer more structure and predictability than graphs. Graphs allow arbitrary connections between nodes, but they do not enforce a consistent relationship between adjacency and hierarchy. In contrast, Benthic's hypergraph imposes a simple but powerful invariant: nodes are neighbors if and only if they share a parent. This is possible because, unlike graphs where edges connect only two vertices, a hypergraph's edges (called *hyperedges*) can connect any set of vertices. Thus Benthic interprets the term `arsenal("bpl", "fa1", "cl1")` as a hyperedge, `arsenal`, that connects the nodes `"bpl"`, `"fa1"`, and `"cl1"`. This structure directly encodes both hierarchy (the parent is `arsenal`) and adjacency (its children are neighbors) in a single unit. As a result, Benthic can maintain a tight correspondence between `parent` and `neighbor` relationships by construction. Compared to Data Navigator, Benthic enforces a consistent encoding of relationships, resulting in clearer and more predictable structures.

Figure 5 provides a formal description of the Benthic hypergraph in TypeScript. The hypergraph is represented as a collection of hyperedges, each uniquely identified by an `Id`. Each hyperedge




```
title(
  legend(
    bpl(
      "bpl1-legend",
      "bpl2-legend",
      "bpl3-legend",
      "bpl4-legend"),
    fa(
      "fa1-legend",
      "fa2-legend",
      "fa3-legend",
      "fa4-legend"),
    cl(
      "cl1-legend",
      "cl2-legend",
      "cl3-legend",
      "cl4-legend")),
    x-axis(
      arsenal("bpl1-x",
        "fa1-x", "cl1-x"),
      chelsea("bpl2-x",
        "fa2-x", "cl2-x"),
      liverpool("bpl3-x",
        "fa3-x", "cl3-x"),
      manchester("bpl4-x",
        "fa4-x", "cl4-x")))
```

(a) Olli tree structure. The data points must be duplicated, since Olli nodes cannot overlap.

```
parent("title", "legend")
neighbor("legend", "x-axis")

parent("x-axis", "arsenal")
neighbor("arsenal",
         "chelsea",
         "liverpool",
         "manchester")

parent("arsenal", "bpl1")
parent("arsenal", "fa1")
parent("arsenal", "cl1")
neighbor("bpl1", "fa1", "cl1")

...

parent("manchester", "bpl4")
parent("manchester", "fa4")
parent("manchester", "cl4")
neighbor("bpl4", "fa4", "cl4")

parent("legend", "bpl")
neighbor("bpl", "fa", "cl")

parent("bpl", "bpl1")
neighbor("bpl1", "bpl2", "bpl3", "bpl4")

...

parent("cl", "cl1")
neighbor("cl1", "cl2", "cl3", "cl4")
```

(b) Data Navigator graph structure. Dashed lines denote parent-child edges. Red lines denote missing parent-child edges.

```
title(
  legend(
    bpl("bpl1", "bpl2", "bpl3", "bpl4"),
    fa("fa1", "fa2", "fa3", "fa4"),
    cl("cl1", "cl2", "cl3", "cl4")),
  x-axis(
    arsenal("bpl1", "fa1", "cl1"),
    chelsea("bpl2", "fa2", "cl2"),
    liverpool("bpl3", "fa3", "cl3"),
    manchester("bpl4", "fa4", "cl4")))
```

(c) Benthic hypergraph structure. Benthic's structure generalizes both Olli's and Data Navigator's, using hierarchy to express relationships like in Olli, but allowing those relationships to overlap like in Data Navigator.

Figure 4: A comparison of the Olli, Data Navigator, and Benthic representations of the same stacked bar chart from Figure 1a. (Note, the diagrams have been truncated to show only the first three teams to reduce complexity.) The tree doesn't capture adjacent relationships, and the graph captures parent-child relationships in an ad-hoc way. Meanwhile the hypergraph captures both hierarchical and adjacent relationships.

also includes a `displayName` and a `description`, which provide localized alt text that is read aloud to users during navigation.

Each hyperedge encodes a grouping by listing its `children` — the IDs of the nodes it connects. In addition, each hyperedge tracks its `parents` — the hyperedges in which it is nested. This `parents` array is redundant in theory (as it can be derived from all `children` arrays), but is stored explicitly for faster lookup during navigation. When users move to a hyperedge, both the `displayName` and `description` are read aloud; when switching between parents

```
type Id = string;

type Hypergraph = {
  [id: Id]: Hyperedge;
};
```

```
type Hyperedge = {
  id: Id;
  displayName: string;
  description?: string;
  children: Id[];
  parents: Id[];
};
```

Figure 5: A TypeScript formalism of the Benthic hypergraph. The hypergraph is a collection of hyperedges tagged by `Id`. Like with a tree structure, a leaf node in the hypergraph is a special case of `Hyperedge` where the child array is empty. Each hyperedge contains a display name, an optional longer description, an array of its children, and an array of its parents. Both the display name and longer description are used when visiting a node, but only the display name is presented when switching the parent context.

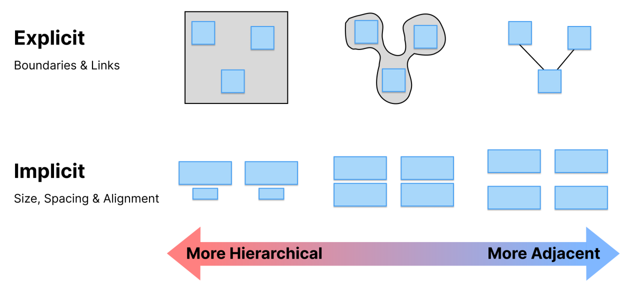


Figure 6: This figure illustrates a continuum from hierarchy to adjacency. The horizontal axis ranges from "More Hierarchical" to "More Adjacent," while the vertical axis contrasts explicit versus implicit grouping cues. This continuum suggested to us that the underlying structure of graphical representations ought to have a unified or blended concept of hierarchical and adjacent relationships — namely, hyperedges in a hypergraph.

during context changes, only the `displayName` is announced to maintain brevity.

This model generalizes both trees and graphs. A tree is a special case of a hypergraph in which each node appears in only one `children` array — that is, every node has a unique parent. A graph is another special case where each hyperedge either represents a leaf node (with no children) or an edge that connects exactly two such leaf nodes. This unification allows Benthic to seamlessly represent both hierarchical and adjacent relationships in a single, coherent structure — preserving perceptual congruence without requiring duplicated nodes or ad hoc edge logic.

5.1.4 Continuum between hierarchy and adjacency. Existing approaches often distinguish rigidly between hierarchical relationships, such as those in trees, and adjacent relationships, such as those in graphs. Hierarchies often represent nested information where smaller parts form bigger ones. On the other hand, we typically use adjacent structures like node-link graphs to represent "flat" relationships like friendship networks.

By coupling `parent` (hierarchical) and `neighbor` (adjacent) relationships, Benthic hyperedges blur this distinction. While this approach may seem surprising, it actual mimics ambiguity between hierarchy and adjacency found in graphical representations. Consider Figure 6. We often represent hierarchies with enclosures or tight clustering (e.g., stacking) and adjacencies with links or loose proximity (e.g., annotations). However, the boundaries between hierarchical and adjacent relationships are nebulous. We can continuously deform an enclosure relationship into a connecting relationship. We can continuously deform tight spatial proximity that affords hierarchy into looser relationships.

Due to *perceptual congruence*, we therefore expect that a domain-agnostic structure ought to exhibit this same ambiguity between hierarchical and adjacent relationships, unlike trees and node-link graphs. A hypergraph is just such a structure. In fact, hyperedges in hypergraphs are often visualized as ambiguous enclosures that resemble connecting lines [8], as in Figure 6.

5.2 Benthic's Screen Reader Interface

The Benthic interface enables screen reader users to navigate complex diagrams represented using hypergraphs. At any given time, a screen reader user will be focused on a node in the hypergraph within a given parent context that defines the node's siblings. However, as hypergraphs can contain overlapping nodes (unlike trees) and "edges" with more than two elements (unlike graphs), they can pose a challenge for *fluid* screen reader traversal. As a result, we co-designed a new screen reader navigation interface with our co-author Hajas, a blind researcher with experience in accessible representations. Over the course of roughly six months, we developed multiple prototype interfaces. We met regularly on Zoom and communicated via email to evaluate the strengths and limitations of each prototype. In this section, we describe the resultant interface and summarize Benthic's key bindings in Table 1.

5.2.1 Focusing on the current hypergraph node. The `h` key acts like a "home" key that focuses the screen reader on the selected hypergraph node. This key binding helps users quickly recover their place if the interface behaves unexpectedly or the screen reader loses focus.

5.2.2 Moving between children of a hypergraph node. To support navigation between neighboring elements, Benthic presents nodes at the same level of the hypergraph as a one-dimensional, bidirectional list. For example, in step 3 of Figure 2, nodes `BPL`, `FA Cup`, and `CL` are displayed as neighboring elements in the same layer. Users can navigate through this list using the arrow keys: `←` or `↑` to move backward, and `→` or `↓` to move forward.

5.2.3 Moving up and down the parent-child hierarchy. Users press `shift + ↓` to descend into the currently focused child node, making it the new parent context. They press `shift + ↑` to ascend from the focused child into the parent context. However, unlike in a tree structure, where a child only has one parent, in a hypergraph, a child can have arbitrarily many parents. For example, in Figure 2, `Arsenal CL` is a child of both `CL` and `Arsenal`. If the currently

Table 1: Key bindings for navigating Benthic's screen reader interface

Key(s) Pressed	Description
`h`	The h key acts as a home key that can help users refocus screen readers on the current node.
`←` / `↑`	Users can press the left arrow or up arrow to navigate to the previous neighboring (sibling) node during navigation within a hyperedge or previous menu option during parent context switches.
`→` / `↓`	Users can press the right arrow or down arrow to navigate to the next neighboring (sibling) node during navigation within a hyperedge or next menu option during parent context switches.
`shift + ↑`	Users can press shift in conjunction with up arrow to navigate up a layer in the hierarchy (towards a node's parent).
`shift + ↓`	Users can press shift in conjunction with down arrow to navigate down a layer in the hierarchy (towards a node's children).
`enter`	Users can press enter for selection. This includes using the enter key to select a new parent context during context switches.

focused child has only one parent, pressing `shift + ↑` takes the user directly to that parent node, updating the interface with information from the new layer. But when a node has multiple parents, pressing `shift + ↑` brings the user to a parent context layer. This state, shown in Figure 2 ❻, is presented at the top of the Benthic screen reader interface. It includes a description of the focused child, the current parent context, and any additional parent contexts the child may have. A prompt then instructs users to use the left and right arrow keys to explore alternate parent contexts. Pressing `enter` selects one of these as the new parent context, while pressing `shift + ↑` again continues traversal upward along the selected path. This design ensures that users are aware of multiple possible paths in the hierarchy and can make informed choices when navigating upward through the hypergraph.

5.2.4 Design Rationale and Alternatives. We iterated on several prototypes with co-author Hajas before arriving at our final design. Here, we describe three alternatives, each reflecting different trade-offs between traversal fluidity and user orientation.

Flat traversal. To minimize explicit context switching, we tried augmenting the list of the currently focused child's neighbors to include "cousin" nodes from sibling parent contexts. This structure was inspired by Data Navigator's traversal structure for the stacked bar chart. For example, in the stacked bar chart from Figure 1a, the `Arsenal CL` and `Chelsea BPL` nodes would both be in the list of adjacent nodes for `Liverpool FA Cup` even though these 3 data points do not share any common parent contexts. While this fluid navigation made lateral movement easy, parent context switches became hidden. As co-designer Hajas moved between adjacent nodes, the parent context of the focused node changed without any cues or verbal indications. As a result, when he tried to ascend the hierarchy, the parent grouping he intended to follow was sometimes inaccessible.

Default Parent Contexts with Context Cycling. This model used traversal history to determine the upward path: pressing `shift + ↑` always returned the user along the path they had originally descended. A separate hotkey (`p`) let users cycle through alternative parent contexts, updating both the list of neighboring elements and the default path for future ascent. This approach reduced friction by minimizing keypresses and avoiding separate menus or prompts for context switches. However, it offered no cues when multiple parent contexts were available. In testing with Hajas, we found that it was easy to remain unaware of alternate upward paths unless a user already knew to press `p` at specific nodes.

Default Parent Contexts with Context Switching Menu. This prototype worked similar to the previous one. However, instead of pressing `p` to cycle between parent contexts, the user presses `g` to jump to a dedicated menu listing alternate parent contexts. This menu offered more explicit guidance than the `p` key in the previous model, clearly signaling what other groupings were available and how to change the current node's context. While this approach made context switches more discoverable and deliberate, it interrupted the traversal flow by pulling users out into a separate interaction.

Across these prototypes, we identified a central trade-off between fluidity and wayfinding. More fluid interfaces require fewer key presses, fewer user decisions, and fewer interruptions with separate menus. Meanwhile, designs that were less fluid provided stronger signposting: they surfaced cues about the structure, made parent contexts explicit, and helped users understand where they were and what actions were possible. However, these same designs slowed navigation and introduced more friction. In contrast, more fluid designs allowed for faster, uninterrupted movement but at the cost of structural transparency — users often missed alternative parent contexts. Our final design aims to balance these forces: integrating parent context selection directly into the traversal flow, making parent context selection explicit and actionable when they occur.

6 EVALUATION

In evaluating Benthic, our goal was not to measure task accuracy. Rather, we designed an *exploratory study* in order to understand how participants used Benthic to make sense of, draw inferences from, and reason about relationships within graphical representations.

6.1 Study Design & Procedure

We conducted 90-minute Zoom studies with 15 blind participants. Participants were asked to explore and reason about two kinds of graphical representations that have been previously studied with sighted readers: a set of pulley diagrams [6, 17], and a set of bar charts drawn from a prior study by Boger & Franconeri [3]. In selecting these diagram types, we aimed to examine whether

Benthic affords BLV readers reasoning patterns that are equivalent to those sighted readers use.

6.1.1 Participants. We recruited 15 blind individuals through public calls shared on Twitter and relevant mailing lists. Each participant received $60 for their participation in a 90-minute Zoom session. To protect participant privacy, we report only aggregate demographic information and acknowledge that socially constructed data such as race and ethnicity should be collected and represented with care. 80% (n=12) of participants self-identified as totally blind and 20% (n=3) self-identified as totally blind with some light perception or low vision. 33% (n=5) have been blind since birth, 53% (n=8) have not been blind since birth, and 14% (n=2) did not answer. Participants were split into 33% (n=5) NVDA users, 60% (n=9) JAWS users, and 7% (n=1) Voiceover users, aligning with screen reader statistics [36]. Demographically, 27% of our participants use she/her pronouns (n=4) and the rest used he/him pronouns (n=11). Participants self-reported their ethnicities (Asian, Black/African, Hispanic/Latinx, and Caucasian/white), covered a diverse range of ages (20–50+), and had a variety of educational backgrounds (high school, undergraduate, and graduate). 10 participants self-reported as slightly to moderately familiar with data visualization concepts and methods, 1 as not at all familiar, 3 as expertly familiar, and 1 as no answer. Participants reported a high variety of frequency interacting with data or visualizations, from 1-2 times a year to 3 or more times/week, with most reporting 1–2 times/month.

6.1.2 Study Procedure. Co-authors Mei, Pollock, and Zong conducted the studies following Frøkjær and Hornbæk's Cooperative Usability Testing (CUT) method [11]: Mei served as the guide, introducing the Benthic prototype and navigation structures and speaking with participants, while Pollock and Zong acted as loggers, documenting usability issues and relevant participant comments.

Each interview lasted approximately 90 minutes and began with a semi-structured interview about the participant's background, including their current strategies for interacting with diagrams and data visualizations, the challenges they face, and what they find helpful in existing approaches. Participants were then introduced to Benthic through a line chart depicting average summer temperatures in Chicago and Seattle during June, July, and August (adapting an example from the gallery of Vega-Lite [25], a popular visualization tool). This introduction was designed to familiarize participants with Benthic's interface and navigation commands, and the guide walked participants through the available interactions and answered any questions to ensure they felt comfortable using the interface. Following this tutorial, participants used Benthic to independently explore two kinds of graphical representation — pulley diagrams and bar charts — described in the next subsection.

6.1.3 Data Collection & Analysis Methods. We followed a mixed methods approach, collecting both quantitative and qualitative data. For the quantitative data, we developed two Likert-scale surveys to assess the impact of Benthic's design on participants' experiences navigating the two diagrams. These surveys measured aspects such as ease of navigation, cognitive load, and support for context switching. Responses were collected on a five-point scale, with higher values indicating greater ease or agreement. For the qualitative analysis, we followed a grounded theory approach [5] to identify recurring themes in how participants described their interactions with Benthic. Co-authors Mei and Pollock open-coded study transcripts and logger notes, iteratively organizing the codes into broader categories.

6.2 Structural Representations & Study Tasks

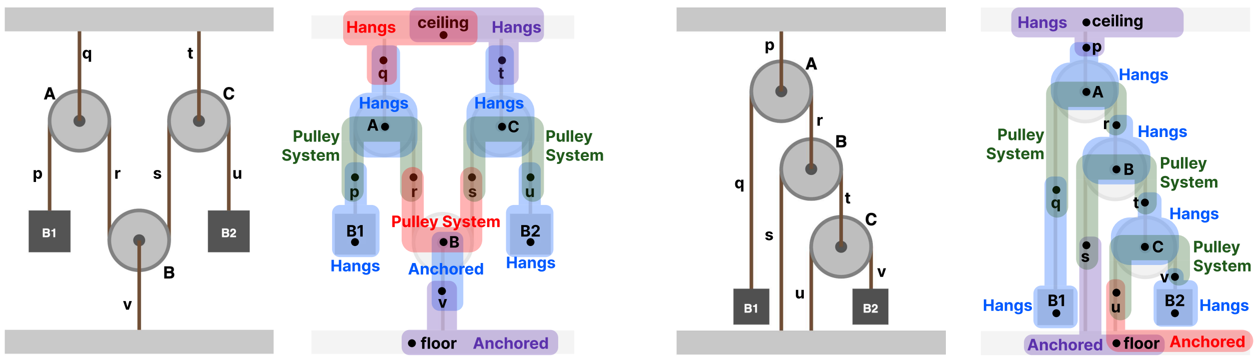
Our evaluation was designed primarily to assess perceptual congruence. As such, we selected examples from two distinct domains of graphical representation: scientific diagrams and information visualization. These representations have existing theories about their perceptual structures, allowing us to interpret whether users' understanding of the Benthic screen reader structure aligned with established expectations for sighted readers.

6.2.1 Pulley Diagrams. The first set of Benthic structures depicted a system of pulleys (such as those found in physics pedagogy). These diagrams have been cited by Larkin & Simon as examples where graphical representations, by externalizing spatial and relational structures, support more efficient reasoning than text [17]. Peter Cheng later empirically studied this idea by comparing how participants solved pulley problems using diagrammatic, tabular, or sentential (i.e, textual) representations [6]. To do so, Cheng used three pulley diagrams of varying levels of complexity: "simple," "medium," and "complex." These require applying two, three, and four physics rules, respectively.

To understand how blind participants conceptualize and traverse diagrams using Benthic, our Benthic structures adapt Cheng's "simple" and "medium" pulley diagrams, reproduced in Figure 7. In the "simple" diagram, a reader must apply two rules to determine the weight of box B2 given the weight of box B1: the *Rope Support Rule* (a rope's value is equal to the weight it supports) and the *Equal Tension Rule* (ropes passing over or under the same pulley have equal values). The "medium" diagram adds a third rule: the *Combine Rule*, which specifies that a rope supporting an entire pulley system takes on the combined value of the ropes within that system.

In our study, we anticipated that participants would have varying levels of experience with physics concepts, so we gave them a conceptual overview of how pulley systems work before beginning the pulley tasks. This introduction included explanations of how the rope typically runs across the pulley and how it supports objects on either side, and was designed to ensure all participants had a shared foundation for interpreting the diagram. All participants started with the "simple" structure, and 10 participants progressed to exploring the "medium" one. Across both diagrams, participants were asked to determine the weight of the second box (B2) given the weight of a known box (B1). As participants operated Benthic, we devoted particular attention to how they traversed the diagram, interpreted the components in relation to one another, and applied the physics principles we provided. After completing the pulley task, participants were asked to describe their understanding of the diagram's structure — what components were connected, how they related to one another, and how they envisioned the diagram's overall layout.

6.2.2 Bar Chart: Replicating Boger & Franconeri. With the final set of Benthic structures, we sought to replicate a graphical perception study conducted by Boger & Franconeri [3] to determine whether



(a) Cheng's "simple" pulley system and Benthic hypergraph, with a single rope connecting B1 and B2.

(b) Cheng's "medium" pulley system and Benthic hypergraph, where the rope of the previous pulley supports the next pulley.

Figure 7: The diagrams and Benthic hypergraphs used for the pulley diagram section of the study.

the impact they identified certain graphical structures having on sighted readers carries over for our blind participants operating Benthic. In the original study, sighted participants viewed two bar charts containing identical data about the heights of two children, Charlie and River, at ages 8, 10, and 12. The first bar chart grouped the data by child while the second chart grouped the data by age — that is, the first chart depicts two groups of three bars, while the second chart shows three groups of two bars. Crucially, however, the study embedded an implausible trend in the data: River's height decreased from age 10 to 12. In the original study, participants were much more likely to miss this trend when shown the second bar chart (i.e., when the data was grouped by age rather than child), demonstrating how structural design choices can influence the patterns sighted readers perceive.

In our study, we recreated these two bar charts as Benthic navigation structures, and our goal was to see whether making one structure appear more salient would influence interpretation for screen reader users in the same way visual grouping influences sighted users. The original bar charts and our hypergraphs for each are shown in Figure 8. To promote a sense of salience, we intentionally limited access to the alternate grouping at the top level of the hypergraph. For example, in the age-grouped structure, participants began their navigation at a top-level `Age` node, followed by options for specific ages (e.g., age 8, age 10), which then led to each child's data at that age. The alternative grouping — by child — was not accessible from the top level and was only reachable from the individual data points. Similarly, in the child-grouped structure, participants initially navigated by child, with age-based grouping hidden in deeper levels of the Benthic structure. This asymmetry was enforced by removing a top-level edge in the hypergraph, so participants could only access the alternate grouping after reaching the data points.

Participants were randomly assigned to interact with one or the other chart, and were first asked to freely explore the data and describe any patterns they found interesting. If they did not mention the implausible trend, we followed up with increasingly specific prompts adapted from the original study like "Did you notice anything that didn't make sense in the plot?" and "What happens to River's height between 10 and 12?" A final task with the structure prompted participants to engage in context switching by asking "What happens to Charlie's height between 8 and 10?" and "Who is taller at age 8?" Participants described how they pictured the chart, including how they imagined the bars were arranged and what they thought appeared on each axis. We placed this navigation structure at the end of our study to maximize the chances that participants were comfortable operating Benthic and minimize the likelihood that navigational difficulties would interfere with participants' exploration and analysis.

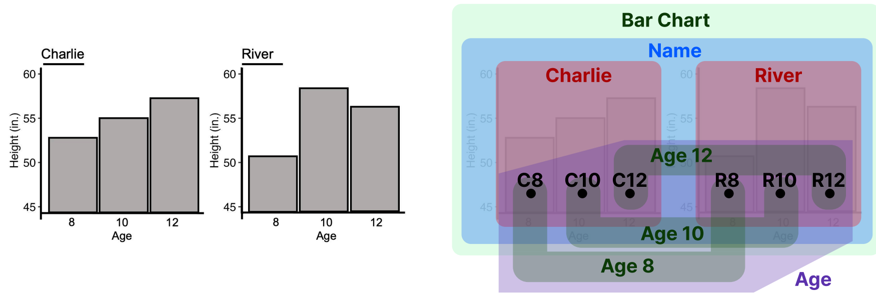
6.3 Quantitative Results

We report participants' responses to our Likert-scale survey questions in Table 2. Median scores suggest that participants found the Benthic interface enjoyable and easy to use. They reported that context switching and navigating branching paths were generally easy, with slightly more intuitive navigation in the bar chart structures. Participants also felt the tool supported the formation of mental models and required minimal cognitive effort to perform computations — again, with the bar chart rated as somewhat easier to interact with. Finally, participants generally agreed that spatial navigation would be moderately helpful for navigating the system.

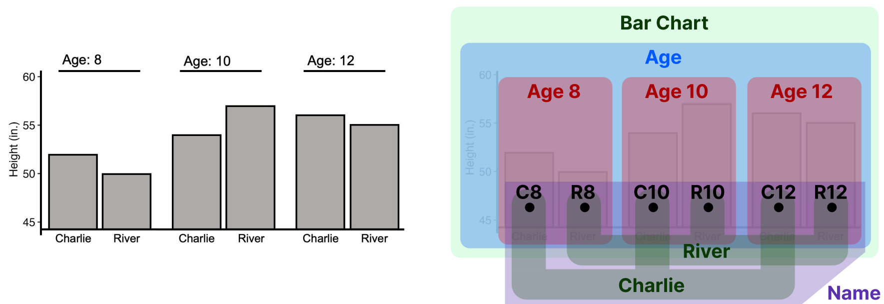
6.4 Qualitative Results

Statistics provide only a partial picture of participants' experiences [1]. Thus, to contextualize their scores, we identified the following themes through qualitative analysis.

6.4.1 Benthic enables flexible, goal-driven exploration. In contrast to existing approaches which impose a predefined summary (in the case of alt text) or predefined groupings (in the case of systems like Olli), Benthic enabled participants to freely reorganize the data according to their own interpretive needs. Participants were able to use context switches to shift and regroup elements and navigate relationships that were interesting to them. For instance, while exploring the bar chart structures, P13 preferred to look at the data grouped by `child` instead of grouped by `age`, emphasizing that *"That's how I was thinking about it."* Grouping by `age` did not align



(a) Boger & Franconeri bar chart and Benthic hypergraph, with data grouped by child, then by age. The intention of this chart is to emphasize the surprising trend in the data.



(b) Boger & Franconeri bar chart and Benthic hypergraph, with data grouped by age, then by child. The intention of this chart is to obscure the surprising trend in the data.

Figure 8: The charts and Benthic hypergraphs used for the bar chart section of the study.

Table 2: Rating scores pulley diagram and bar chart on a five-point Likert scale where 1 = Very Difficult (Not helpful at all) and 5 = Very Easy (Extremely helpful). Median scores are shown in bold, averages in brackets [], standard deviations in parentheses ().

Pulley Diagram	Score	Bar Chart	Score
How easy was it to learn to use the screen reader tool for the pulley diagram?	**4** [3.47] (1.41)	How easy was it to learn to use the screen reader tool for the bar chart?	**5** [4.47] (0.83)
How easy was it to navigate the different paths that move up a layer in the pulley diagram?	**4** [3.80] (1.15)	How easy was it to navigate the different paths that move up a layer in the bar chart?	**5** [4.01] (1.28)
How easy was it to perform a context switch using the tool?	**4** [3.40] (1.24)	How easy was it to perform a context switch using the tool?	**4** [4.21] (0.89)
Once you understood the tool, how easy was it to build a clear mental model of the pulley diagram?	**4** [3.50] (1.40)	Once you understood the tool, how easy was it to form a clear mental model of the bar chart?	**5** [4.73] (0.46)
How easy was it to keep track of all the information needed to calculate the values of the ropes and boxes?	**4** [3.47] (1.25)	How easy was it to keep track of all the information needed to understand the trends in the data?	**5** [4.53] (0.83)
How helpful would spatial navigation be for the pulley diagram?	**3** [3.13] (1.19)	How helpful would spatial navigation be for the bar chart?	**3** [3.13] (1.36)
		How easy was it to find that River's height decreases between ages 10 and 12?	**5** [4.4] (0.83)

with how he wanted to approach the task: *"If someone were to say, tell me the difference between River and Charlie, [...] I wouldn't even look at it from the age of each person. That doesn't even make sense to me. I would want to group it by the person."* But even if a reader has a preferred grouping, the ability to switch between groupings was essential to gaining a complete understanding. For instance, in the task containing the implausible trend in height (§ 6.2.2), P4 noticed that River was *"lopsided"*. He found that *"when [River's] height [was reduced] to 55 [inches], then I began to have doubts. I had to compare. Was I thinking correctly, or was it something wrong?"* Using the Benthic interface, P4 performed a context switch to group the data by `child` instead of by `age` to confirm the anomaly he had discovered.

Participants also highlighted that these context switches enabled them to filter information in order to focus on the elements they cared most about. For example, while exploring the introductory line chart, P7 described that the ability to change how the data is grouped made it very easy to make comparisons: *"I like that you can go, here's Seattle [in] July. So if you said 'which one is hotter in X month?' then I would have gone to X month and told you."* P6 echoed a similar sentiment, emphasizing the efficiency of filtering irrelevant data: *"if you've got 12 cities and I'm interested in 2, why should I have to look at all 12? And it looks like this could allow me to [filter]."*

6.4.2 Benthic helps build a conceptual understanding of hierarchy and adjacency. Participants appeared to quickly learn how to navigate the Benthic interface, and described the interactions in ways that mirrored the underlying structure of the diagrams. Commands like `shift + ↓` were often described as *"digging deeper"* or *"opening up"*, while the `←` and `→` arrow keys were associated with *"moving across"* — language that reflects hierarchical and adjacent relationships between elements in the diagrams and visualizations participants explored. For example, as P1 explained, *"Once I get to the legend and I want to go to the axis, I do the shift down arrow to open up that axis. Then I use left or right arrows to navigate within that axis."* Similarly, P11 imagined their position in the structure as a *"cursor on the diagram"* and navigational commands would move the cursor laterally or zoom in and out of a region of the diagram. These metaphors suggest that participants were building an understanding of the diagram's structure as they explored it.

Several participants also used Benthic's structural metaphor to articulate broader differences in how each graphical representation was organized. For instance, P3 contrasted the hierarchical nature of the line chart with the more adjacent layout of the pulley diagram: *"There's a difference between the [line chart and the pulley diagram] because in the first diagram, there's hierarchy: city and month. Grouping is different. [The pulley diagram] is not exactly hierarchy. There's a box, you can associate with the rope, and you can associate with the pulley."* Similarly, P14 described the pulley diagram as fundamentally flat, explaining *"In this case, it doesn't make sense to context switch, because everything is a 2-dimensional layout. Context switching to me is like looking at the data from a different angle. Whereas here, that's not necessarily what you're doing, because they're all hanging. So they're all in the plane."* While participants' prior familiarity with these diagram types likely had an effect on these insights, Benthic's hypergraph and interface nevertheless helped participants make such structural distinctions explicit — offering a vocabulary and interaction model that encouraged participants to reflect on how information was organized.

Curiously, while Benthic's hypergraph structure supports both hierarchy and adjacency, participants often found the navigation experience to overly emphasize hierarchy. P14's metaphor captured this frustration: *"It's like if you were showing me a skeleton and you said: okay, find the finger. But I couldn't find the finger by looking at the hand. I had to follow it all the way down from the shoulder to the arm, to the wrist...now find the next finger, and I can't just look at the hand and go: it's beside the first one. I have to go back up to the wrist."* Benthic's current navigation model prevents users from jumping laterally across groups, instead requiring them to first "move up" to context switch and access adjacent elements in other groups. While participants acknowledged the usefulness of the current approach, they suggested an approach that would toggle between hierarchical and lateral movement, as both are necessary to understand a graphical representation. As P14 explained, *"But you always need to move between context and focus. And here you're stuck in focus. [...] When you're looking at something like this, you're always moving between context and focus. There's the pulley, and there's a box... and zooming back out to see the linkages."*

6.4.3 Benthic supports local understanding of diagram structure. When a participant focused on an element in a diagram, they demonstrated a strong understanding of the local relationships that element participated in. Participants often gave detailed descriptions of how ropes, pulleys, and boxes were connected at specific points in their traversal. For instance, P1 explained, *"Rope p is actually the rope that's holding the box. And then on the left side of pulley A is rope p. And then on the right side is rope r."* Similarly, P3 remarked, *"Concept wise, I can explain... between pulleys, connections between ropes, and the boxes hanging on which rope."* However, when asked to describe the overall layout of the diagram, many struggled to explain how components were situated in relation to each other. P6 noted, *"I don't know if we [found out] where Pulley B was connected to other than the ropes. Is it on the floor? Is it on the overhead?"* P14 similarly expressed frustration: *"I'm stuck in focus mode, but I have no idea how these parts relate to each other."* P7 suggested that additional reference points might help with understanding the broader structure: *"Perhaps what could be helpful here would be for you to say floor, then rope S, then Pulley C, rope U... Just tell me when there's the floor, and when is there the ceiling."* When key anchor points like the floor or ceiling weren't organically part of a participant's traversal, they struggled to situate the components they encountered — underscoring the need for navigation that better supports awareness of surrounding structure.

6.4.4 Diagrams suggest spatial expectations that are not captured by structural navigation. Prior work in screen reader interfaces for visualization draws a distinction between *structural navigation*, which refers to movement locally along an accessible structure, and *spatial navigation*, which refers to movement organized by spatial directions in the coordinate space of the graphic [38]. Though our prototypes focused on structural navigation, we found that spatial navigation is crucial in contexts where users expect to move along physical paths. For instance, when participants encountered the pulley system, they anticipated navigation to reflect spatial movement along ropes and pulleys. P3 suggested how they envisioned navigating the pulley diagram: *"Let's say [you use the] up and down arrow keys with a rope. If there's a box hanging down, and then you follow the rope, then [the box is] at the end. When you pass the rope left and right, there's a pulley in the middle."* Similarly, P9 also expected spatial movement, saying, *"I thought if I selected the rope and went up, it would take me to whatever that rope is part of"* where in our structural navigation model, they were instead iterating through a list of connected items. These expectations show how users conceptualize the diagram as a physical space; and, without clear spatial cues, participants struggled to make sense of the layout. Participants also described the mental effort required to align key presses with actual movement when their expectations differed (P10).

Finally, to streamline navigation, Benthic integrated an intermediate parent context layer that appears when nodes have multiple parent paths — allowing participants to choose their route upward through the diagram. As we discuss in Section 5.2.4, this design aims to balance fluid traversal with users' awareness of other groupings. While the parent context layer makes discovery of different grouping relations more explicit, some participants found it confusing, likely because it broke expectations about traversal reversibility (P3, P10).

Some participants also found it difficult to anticipate when this intermediate branching layer would appear (P14, P15). P15 shared, *"It usually takes a second to remember if I need to press enter or if it's enough to switch with the arrow keys."* These responses suggest that while the parent contexts supported necessary structure in the diagram, its interaction pattern occasionally clashed with participants' expectations of consistent, bidirectional traversal. Building on P15's observation, we wonder if the interface could show both a node's children and its parents when the user presses `enter`, and have `esc` or `delete` always undo `enter`. This would ensure parent contexts are always accessible while maintaining reversibility.

6.4.5 Results of the Replication Study. As described in 6.2.2, one of our tasks was designed to replicate a graphical perception study testing whether readers could notice an anomalous trend in height data [3]. The goal of this task was to determine whether Benthic's structure would have a similar influence on readers as the graphical structures in the study.

In contrast to the original study, which found strong effects based on bar chart structure, our participants could perform context switches to easily change the grouping of the data. 12 of the 15 participants identified the implausible trend in River's height, with just one person assigned to the obfuscated structure missing it. This is in stark contrast to the 46% of participants in Boger & Franconeri who missed the same trend [3]. We found this to be revealing of participants' interpretive processes. We had initially expected that the "most salient" grouping in each structure — either `age` or `child` — would influence how participants conceptualized the bar chart. That is, we hypothesized that participants would describe the chart in ways that aligned with the version they were assigned. However, the impact of this salient structure was minimal as most participants performed a context switch almost immediately after descending the hypergraph, regardless of which navigation structure they were working with. Indeed, some participants recognized that the bar chart could be interpreted in multiple ways, and called out that the navigation structure did not clearly specify which one it was meant to primarily represent. For example, P8 remarked, *"I suspect that probably the x-axis is age and that the y-axis is the height of the kids,"* but went on to say, *"Of course that's difficult because I don't know if there are actually 2 bar charts or if there's 1."* Similarly, P9 described: *"If you were grouping them by age, you would have the two bars next to each other for the same age...If you're going to group them by child, then I guess you would have two sets of bars. But then you would need a third dimension — you would need another graph."*

Participants also appeared to have a strong grasp of Benthic's hypergraph structure, as six participants noticed the deliberate asymmetry we introduced to enforce a "salient" grouping. Specifically, they pointed out that they expected to be able to access both age and child groups from the top level of the hypergraph, and were surprised when that was not the case. For example, P7 suggested that both grouping options should be immediately available: *"Here in this second phase you have age. What I would suggest here is, have age, and then you have the ages to choose from, and then I would have names, too."* P12 noted that *"age is like this separate group that does not belong to the main graph. You can only really get to it when you get to the data points themselves."* Similarly, P15 observed, *"This time we only have age, not age and children."* These insights show that participants quickly grasped the hypergraph structure and developed expectations for its navigation.

7 DISCUSSION AND FUTURE WORK

In this paper, we presented Benthic, a perceptually congruent intermediate representation and screen reader interface for graphical representations. We have also shown how Benthic's hypergraph structure is domain-agnostic and allows for fluid screen reader traversal. The Benthic system provides several promising directions for future research and tool development.

7.1 Barriers to Adoption

While Benthic offers a promising framework for perceptually congruent screen reader navigation, broader adoption of the system will require addressing several practical challenges. First, Benthic introduces a new interaction model that differs from familiar tree or graph-based paradigms in most screen readers. Although participants in our study were able to learn and operate the interface with guidance, supporting independent onboarding will be essential. Future work could explore inline tutorials, adaptive scaffolding, or interface cues that help users gradually build fluency.

Second, adoption also depends on making it easier to generate Benthic-compatible structures. In this study, we focused on diagrams with established perceptual structures, but supporting a wider range of diagrams will require scalable methods for generating hypergraphs. One promising direction is to integrate with diagramming libraries like Bluefish [23], which define diagrams in terms of explicit visual and semantic relationships, enabling broader use across domains.

7.2 Towards a Spatial Benthic

Although Benthic's hypergraph structure allowed many participants to understand graphical representations, our evaluation revealed obstacles to achieving stronger perceptual congruence. In tasks involving the pulley diagram, several participants successfully navigated through both hierarchies and adjacency but still expressed uncertainty about the overall layout of the diagram. Their comments and expectations revealed a spatial interpretation of the diagram: they imagined ropes stretching vertically, pulleys suspended from above, and boxes positioned below. However, Benthic's navigation model, grounded in structural relationships, does not explicitly represent spatial layout, orientation, or directionality. This finding suggests that spatial reasoning requires more than structural relationships alone. Perceptual groupings like "above," "next to," or "hanging from" seem not to be reducible to just hierarchy and adjacency and may require special support in future tools.

Another promising direction for future work is to better characterize the kinds of diagrams Benthic supports well — particularly those with discrete components and interpretable groupings — versus those that rely on continuous spatial relationships. Benthic's hypergraph model works effectively for diagrams like bar charts or pulley systems, where content is organized into distinct, separable elements. But it is less clear how Benthic could represent diagrams such as maps or heatmaps, which emphasize spatial continuity, gradients, and orientation. Because Benthic does not currently support spatial navigation, these formats present new challenges for designing perceptually congruent screen reader experiences. Extending Benthic's design goals to address such representations remains an open and compelling area for future research.

7.3 Congruent Structures Across Modalities

The key conceptual insights in Benthic were the result of integrating formal structures from research on graphical representations and research on screen reader traversal. We believe that a promising way to understand how spatial information could be integrated with hypergraphs is to study how spatial information works in other modalities. Sonification, for example, relies not just on the discrete relationships we explored in this paper, but also on continuous ones presented with tones that vary continuously in frequency or amplitude. Screen magnification requires an even closer congruence between graphical representations and alternative representations than in Benthic, because those representations must be presented simultaneously. Magnification could potentially take advantage of a Benthic-like structure to aid in faster navigation or to present users with timely information that is offscreen but related to elements that are currently onscreen. Tactile graphics present interesting issues of context, similar to those in magnification. Whereas a sighted user has a very large field of view and a user of a screen reader can only see one data point at a time, magnification and tactile views present small context windows that allow users to perceive a small collection of items at a time. We suspect that Benthic's structure may help us look for ways to augment these context windows.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under awards #1942659 and #2341748, as well as an Alfred P. Sloan Fellowship.

REFERENCES

[1] Richard P. Bagozzi. 2007. The Legacy of the Technology Acceptance Model and a Proposal for a Paradigm Shift. *Journal of the Association for Information Systems* 8, 4 (April 2007). https://doi.org/10.17705/1jais.00122
[2] Matthew Blanco, Jonathan Zong, and Arvind Satyanarayan. 2022. Olli: An extensible visualization library for screen reader accessibility. *IEEE VIS Posters* 6 (2022).
[3] Tal Boger and Steven Franconeri. 2024. Reading a graph is like reading a paragraph. *Journal of Experimental Psychology: General* 153, 7 (2024), 1699.
[4] Dempsey Chang, Keith V. Nesbitt, and Kevin Wilkins. 2007. The gestalt principles of similarity and proximity apply to both the haptic and visual grouping of elements. In *Proceedings of the eight Australasian conference on User interface - Volume 64 (AUIC '07)*. Australian Computer Society, Inc., AUS, 79–86.
[5] Kathy Charmaz. 2006. *Constructing grounded theory.* Sage Publications, London ; Thousand Oaks, Calif.
[6] Peter C. H. Cheng. 2002. Do diagrams equal deep understanding? The role of diagrams in physics problem solving. In *Proceedings of the 24th Annual Conference of the Cognitive Science Society.* 180–185.
[7] Pramod Chundury, Biswaksen Patnaik, Yasmin Reyazuddin, Christine Tang, Jonathan Lazar, and Niklas Elmqvist. 2022. Towards Understanding Sensory Substitution for Accessible Visualization: An Interview Study. *IEEE Transactions on Visualization and Computer Graphics* 28, 1 (Jan. 2022), 1084–1094. https://doi.org/10.1109/TVCG.2021.3114829
[8] Christopher Collins, Gerald Penn, and Sheelagh Carpendale. 2009. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE transactions on visualization and computer graphics* 15, 6 (2009), 1009–1016.
[9] Frank Elavsky, Lucas Nadolskis, and Dominik Moritz. 2023. Data navigator: an accessibility-centered data navigation toolkit. *IEEE transactions on visualization and computer graphics* 30, 1 (2023), 803–813.
[10] Christian Frings and Charles Spence. 2013. Gestalt grouping effects on tactile information processing: when touching hands override spatial proximity. *Attention, Perception, & Psychophysics* 75, 3 (April 2013), 468–480. https://doi.org/10.3758/s13414-012-0417-6
[11] Erik Frøkjær and Kasper Hornbæk. 2005. Cooperative usability testing: complementing usability tests with user-supported interpretation sessions. In *CHI'05 extended abstracts on Human factors in computing systems.* 1383–1386.
[12] Alberto Gallace and Charles Spence. 2011. To what extent do gestalt grouping principles influence tactile perception? *Psychological Bulletin* 137, 4 (2011), 538–561. https://doi.org/10.1037/a0022335
[13] Joshua Gorniak, Yoon Kim, Donglai Wei, and Nam Wook Kim. 2024. VizAbility: Enhancing Chart Accessibility with LLM-based Conversational Interaction. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24)*. Association for Computing Machinery, New York, NY, USA, 1–19. https://doi.org/10.1145/3654777.3676414
[14] Daniel Hajas, Dario Pittera, Alessio Nasce, Orestis Georgiou, and Marianna Obrist. 2020. Mid-Air Haptic Rendering of 2D Geometric Shapes With a Dynamic Tactile Pointer. *IEEE Transactions on Haptics* 13, 4 (2020), 806–817. https://doi.org/10.1109/TOH.2020.2993151
[15] Sally Hart. 2010. Planning and Designing Quality Tactile Graphics. http://brailleauthority.org/tg/ American Printing House for the Blind, Access Academy webinar.
[16] Highsoft. 2018. Stacked column demo. https://www.highcharts.com/demo/column-stacked. Accessed April 2025.
[17] Jill H Larkin and Herbert A Simon. 1987. Why a diagram is (sometimes) worth ten thousand words. *Cognitive science* 11, 1 (1987), 65–100.
[18] Leaflet.js contributors. 2023. Accessibility Examples — Leaflet. https://leafletjs.com/examples/accessibility/. Accessed April 2025.
[19] Jingyi Li, Son Kim, Joshua A. Miele, Maneesh Agrawala, and Sean Follmer. 2019. Editing Spatial Layouts through Tactile Templates for People with Visual Impairments. In *ACM Conference on Human Factors in Computing Systems (CHI)*. Glasgow, Scotland Uk, 1–11. https://doi.org/10.1145/3290605.3300436
[20] Mermaid.js contributors. 2023. Accessibility Configuration — Mermaid. https://mermaid.js.org/config/accessibility.html. Accessed April 2025.
[21] Don Norman. 2014. *Things that make us smart: Defending human attributes in the age of the machine.* Diversion Books.
[22] Steven Pinker. 1990. A theory of graph comprehension. *Artificial intelligence and the future of testing* 73 (1990), 126.
[23] Josh Pollock, Catherine Mei, Grace Huang, Elliot Evans, Daniel Jackson, and Arvind Satyanarayan. 2024. Bluefish: Composing Diagrams with Declarative Relations. In *Proceedings of the 37th Annual ACM Symposium on User Interface*

Software and Technology. 1–21.

[24] Wolfram Research. 2013. *Interactive Scientific Graphics: Recommended Practices for Verbal Description.* Technical Report. DIAGRAM Center. https://diagramcenter.org/interactive-scientific-graphics.html

[25] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2016. Vega-lite: A grammar of interactive graphics. *IEEE transactions on visualization and computer graphics* 23, 1 (2016), 341–350.

[26] Ather Sharif, Sanjana Shivani Chintalapati, Jacob O. Wobbrock, and Katharina Reinecke. 2021. Understanding Screen-Reader Users' Experiences with Online Data Visualizations. In *ACM Conference on Computers and Accessibility (SIGACCESS) (ASSETS '21)*. New York, NY, USA, 1–16. https://doi.org/10.1145/3441852.3471202

[27] Ben Shneiderman. 2003. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *The Craft of Information Visualization*, BENJAMIN B. Bederson and BEN Shneiderman (Eds.). Morgan Kaufmann, San Francisco, 364–371. https://doi.org/10.1016/B978-155860915-0/50046-9

[28] Volker Sorge, Thomas Auzinger, and Pablo Gervás. 2016. Polyfilling accessible chemistry diagrams. In *International Conference on Computers Helping People with Special Needs*. Springer, 472–479. https://doi.org/10.1007/978-3-319-41264-1_6

[29] Volker Sorge, Myung Lee, and Simon Wilkinson. 2015. End-to-End Solution for Accessible Chemical Diagrams. In *Proceedings of the 12th Web for All Conference* (Florence, Italy) *(W4A '15)*. ACM, 1–4. https://doi.org/10.1145/2745555.2746667

[30] Arjun Srinivasan, Tim Harshbarger, Darrell Hilliker, and Jennifer Mankoff. 2023. Azimuth: Designing Accessible Dashboards for Screen Reader Users. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility* (New York, NY, USA) *(ASSETS '23)*. Association for Computing Machinery, New York, NY, USA, Article 49, 16 pages. https://doi.org/10.1145/3597638.3608405

[31] Masaki Suwa and Barbara Tversky. 2002. External representations contribute to the dynamic construction of ideas. In *Diagrammatic Representation and Inference: Second International Conference, Diagrams 2002 Callaway Gardens, GA, USA, April 18–20, 2002 Proceedings 2*. Springer, 341–343.

[32] John R Thompson, Jesse J Martinez, Alper Sarikaya, Edward Cutrell, and Bongshin Lee. 2023. Chart reader: Accessible visualization experiences designed with screen reader users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18.

[33] Barbara Tversky. 2001. Spatial schemas in descriptions. In *Spatial Cognition: An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, Daniel R. Montello (Ed.). Springer, 79–112. https://doi.org/10.1007/3-540-45460-8_5

[34] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh, and R. von der Heydt. 2012. A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization. *Psychol Bull* 138, 6 (Nov 2012), 1172–1217.

[35] Johan Wagemans, Jacob Feldman, Sergei Gepshtein, Ruth Kimchi, James R. Pomerantz, Peter A. van der Helm, and Cees van Leeuwen. 2012. A century of Gestalt psychology in visual perception: II. Conceptual and theoretical foundations. *Psychological Bulletin* 138, 6 (2012), 1218–1252. https://doi.org/10.1037/a0029334

[36] WebAIM. 2021. Screen Reader User Survey #9 Results. https://webaim.org/projects/screenreadersurvey9/. Accessed: 2025-04-15.

[37] Joseph W. Wiedel and Paul A. Groves. 1969. *Tactual Mapping: Design, Reproduction, Reading and Interpretation.* Technical Report D-2557-S 1969. Department of Health, Education, and Welfare, Washington, D.C. 140 pages. https://archive.org/details/tactualmappingde00jose_0

[38] Jonathan Zong, Crystal Lee, Alan Lundgard, JiWoong Jang, Daniel Hajas, and Arvind Satyanarayan. 2022. Rich Screen Reader Experiences for Accessible Data Visualization. *Computer Graphics Forum (Proc. EuroVis)* (2022). http://vis.csail.mit.edu/pubs/rich-screen-reader-vis-experiences

[39] Jonathan Zong, Isabella Pedraza Pineros, Mengzhu (Katie) Chen, Daniel Hajas, and Arvind Satyanarayan. 2024. Umwelt: Accessible Structured Editing of Multi-Modal Data Representations. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM. https://doi.org/10.1145/3613904.3641996