

# FluidViews: Adaptive Drag-and-Drop Token Filters for Heterogeneous Multi-View Visual Analytics

Bhanu Sunku

Northern Illinois University  
z1974769@students.niu.edu

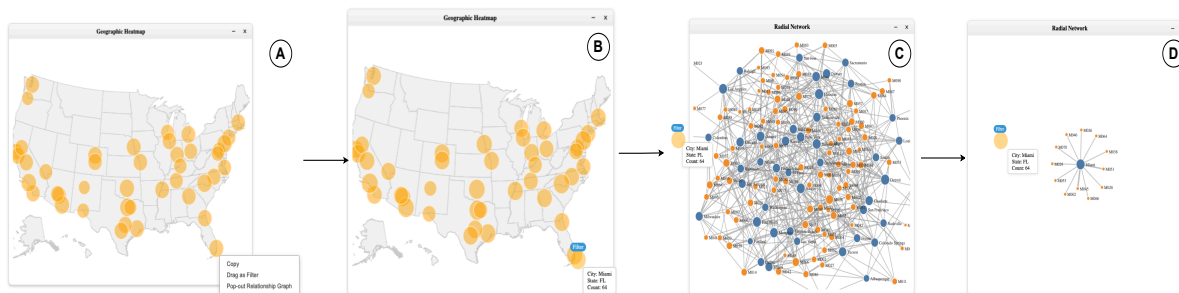


Figure 1: FluidViews’ drag-as-filter interactions. (A) The user initiates a direct-manipulation gesture by right-clicking a geographic data point and selecting “Drag as Filter.” (B) A clone of the selected mark is dragged across the interface. (C) Dropping the token onto the radial network view triggers a context-aware filter, highlighting related entities. (D) A pop-out micro-view reveals first-hop connections for the filtered element, supporting detailed exploration without disrupting the main workspace.

## ABSTRACT

Interactive visual analytics workflows are often disrupted by rigid filter panels and context switches that break analysts’ cognitive flow. We introduce **FluidViews**, a web-based framework that elevates filters to first-class, manipulable objects through two novel direct-manipulation interactions. *Copy-as-Highlight* enables users to duplicate any visual mark into a persistent highlight token for rapid, transient cross-view comparison, while *Drag-as-Filter* allows analysts to pick up a mark and drop it onto another view to apply context-sensitive filters in place no menus, panels, or modal dialogs required. An optional pop-out micro-view provides on-demand, spatially independent subviews for detailed inspection without disrupting the primary workspace. By embedding these lightweight gestures into coordinated multi-view environments, FluidViews preserves analytic momentum, reduces cognitive overhead, and supports fluid, multi-step exploration across heterogeneous datasets. We describe the system’s design and implementation, illustrate its application in exploratory workflows, and discuss how tangible filter objects can transform interactive data exploration.

**Index Terms:** Multi-view Visual Analytics, Direct Manipulation, Drag-and-Drop Interaction, Highlighting Techniques, Detail-on-Demand

## 1 INTRODUCTION

*Coordinated multiple-view visualizations* (MVs) are a central strategy for supporting exploratory data analysis across complex, multi-dimensional datasets. [13, 3, 17]. By juxtaposing different representations, MV systems help users detect relationships, form hypotheses, and reason across diverse facets of data [13]. To support such cross-view analysis, users must often apply dynamic filters, isolating relevant subsets, focusing on anomalies, or comparing

conditions across views. However, the design of filtering interactions in conventional MV systems remains limited.

Conventional approaches often delegate filtering to static side panels, faceted search widgets, or modal dialogs that are separate from the views themselves, fragmenting user attention and analytic flow [23, 7]. Moreover, many systems lack mechanisms for constructing filters directly through interactions within the views. Instead, they often limit cross-view coordination to brushing and linking, where transient, hover-based highlights provide only momentary associations without supporting persistent filtering or multi-condition query building [3, 20]. As a result, users are forced to shift focus between views, leading to disjointed analysis experiences and an increased cognitive burden when composing or recalling complex filter states [19, 9].

Despite the importance of fluid, tightly integrated filtering in coordinated multiple-view systems, existing techniques offer limited support for constructing, combining, and managing filters directly within the visualization space [3, 8]. This gap raises critical questions for multi-view interaction design: How might we make filters tangible and manipulable within and across views to support more direct, expressive analysis workflows? How might we enable users to fluidly compose, persist, and transfer filters across coordinated views without disrupting analytic flow? Addressing these challenges requires rethinking filters not as static menu configurations, but as first-class, interactive objects embedded within the analytic workspace.

To address these challenges, we present *FluidViews*, a web-based visual analytics framework that enables seamless, expressive interaction across coordinated views. Our contributions are twofold: (1) we introduce a novel interaction framework, *Direct Filter Manipulation* (DFM), that treats visual marks as manipulable, first-class objects embedded directly within the analytic workspace, allowing users to fluidly construct, transfer, and persist analytic context without relying on detached control panels; (2) we implement this framework in a proof-of-concept system that supports a suite of complementary techniques, including *drag-as-filter*, which allows users to drop visual marks onto other views to apply filters; *copy-*

*as-highlight*, which propagates selections across views for transient comparison; hover-based multi-facet highlighting with click-to-lock persistence; and pop-out micro-views that enable detailed inspection without disrupting the main analytic flow. Together, these techniques extend DFM principles to support a more fluid, multi-step exploration experience across diverse visualizations.

## 2 BACKGROUND WORK

### 2.1 Coordinated Multiple Views

Coordinated Multiple views have long been used to provide different perspectives on complex data, allowing users to explore relationships via interactive linking and brushing. Early guidelines for using multiple views stressed minimizing context switching, as each additional view incurs cognitive cost in preserving the mental map of the data across views [3]. Foundational systems like Snap-Together[13] Visualization introduced user-defined coordination of independent visualizations, enabling synchronized selection and highlighting of data points across views. Such brushing-and-linking techniques, originally conceptualized for 2D plots [4], remain a cornerstone: when a user selects an “interesting” subset in one view, corresponding elements in other views are emphasized to maintain contextual awareness of that subset throughout the interface.

A persistent weakness of early coordinated-view tools was that inter-view relationships were implicit typically a fleeting brush highlight, so users lost track of selections when views were spatially or conceptually distant. Researchers therefore introduced explicit visual links to preserve context. VisLink drew 3-D arcs between related objects in separate 2-D plots, letting analysts trace connections without leaving either view [5]. Cross-application systems extended the idea to multiple windows, overlaying straight link lines on the desktop to guide attention across programs [27]. Context-preserving visual links later refined this by routing curved paths around clutter, maintaining each view’s local readability while still tying distant representations together [21]. Together, these advances turned once-fragile highlights into durable, navigable bridges across heterogeneous views.

Beyond simple linking lines, later work softens or removes chart boundaries. Weaver’s cross-filtered views let a drag or slider in one chart instantly filter all others, keeping chosen data visible [28]. Systems such as ConnectedCharts embed small connector graphics [26], while MyBrush lets users define their own link sets [10], giving analysts finer control over how views stay synchronized and salient patterns stay in sight.

Recent work focuses on integrating context across views to cut fragmentation. Boundary-blending techniques visibly merge, highlight, or embed elements from one view into another so that noteworthy data remain in sight [24]. Situated analytics applies the same idea in augmented-reality settings, using persistent cues to link physical and virtual views [16]. These advances reduce context loss, yet important subsets can still slip away when filters or layouts change gap FluidViews addresses by turning filters into persistent, manipulable tokens.

### 2.2 Filtering Techniques in Interactive Visualization

Interactive filtering enables users to narrow data by hiding clutter or isolating subsets of interest. Early dynamic query systems like FilmFinder and HomeFinder demonstrated the benefits of real-time visual feedback through direct manipulation, tightly coupling input with output to support fluid exploration [2, 18]. In multi-view environments, filtering often propagates via brushing across coordinated views, a principle generalized by cross-filtering systems to allow iterative, multi-dimensional subset refinement across multiple visualizations [28].

However, aggressive filtering can obscure users’ memory of prior contexts by removing or muting non-selected items. Fo-

cus+context techniques attempt to preserve awareness by softly rendering filtered-out elements instead of erasing them entirely [11]. Still, basic filtering struggles to maintain visibility of previously identified “interesting” chunks across interaction sequences. Progressive selection models offer a solution: systems like GraphTrail [6] record subsets selected through each filtering step as nodes in an analytic trail, enabling backtracking and comparison of intermediate findings. Beyond preserving selections incrementally, contextual bookmarking provides users with explicit means to save and restore analytical states. Contextual snapshots capture selections along with visualization parameters to preserve the original discovery context, allowing restoration even after drastic changes [12]. Analytic provenance systems such as Voyager [29] extend this by automatically logging user interactions, offering undo trails and exploration histories for revisiting earlier analytical paths.

Recent advances have introduced context-aware filtering approaches that dynamically adjust or recommend filters based on prior selections, aiming to preserve notable patterns even as contexts shift. Progressive visual analytics frameworks further reduce the cognitive burden by allowing users to iteratively refine filters while observing partial, real-time results, mitigating the risks of sudden context loss [25, 22]. Despite these innovations, maintaining awareness of salient data subsets across multi-view exploration and dynamic filtering remains challenging, highlighting the need for mechanisms that persist user-identified “interesting chunks” across shifting analytic contexts.

## 3 DESIGNING FLUIDVIEWS

MV exploration requires eliminating the high cognitive load and fragmentation common in traditional coordinated-view systems. To address the limitations identified in prior systems, we introduce a new interaction paradigm: *Direct Filter Manipulation* (DFM). DFM reimagines filters not as static UI controls, but as tangible, manipulable objects embedded within the visualization space. This design shift is guided by two key principles, each motivated by cognitive theory and empirical insights from visualization research.

**Explicit Persistence (P1):** Traditional brushing and linking techniques rely on transient interactions hover highlights or ephemeral linkages that disappear after a moment, disrupting multi-step analysis [3, 4]. Moreover, filters in many systems are buried in detached control panels or require multiple clicks to reveal or reconfigure. In contrast, we treat filters and highlights as *persistent, visible, and manipulable objects*. This reduces working memory load, externalizes analytic state, and enables users to build upon previous interactions without losing context.

**Seamless In-Place Filtering (P2):** Prior work highlights the cognitive cost of interface fragmentation and context switching [15, 7]. Systems that require users to navigate modal dialogs, side panels, or separate dashboards for filtering disrupt analytic momentum and inhibit fluid insight generation. Our second principle promotes *in-situ interaction*, where filtering and selection happen directly through gestures and visual marks, embedded in the same space as data exploration. This supports spatial reasoning, analytic flow, and minimizes interface indirection.

To instantiate the DFM paradigm, we introduce four composable, low-friction interaction techniques designed to support core tasks in exploratory analysis filtering, comparison, inspection, and state retention. These techniques emphasize direct manipulation, fluid transitions, and embedded interaction. They are:

**Drag-as-Filter (T1):** This interaction allows users to pick up any visual mark (e.g., a city on a map or a node in a network) and drop it onto another view to apply a scoped filter(Fig 1). The dropped object is transformed into a persistent filter token, stored in a visible filter tray, and applied to the target visualization. Multiple marks can be dropped to compose complex, conjunctive filters. This supports rapid, context-preserving filtering across heterogeneous views

and reduces dependence on external UI component (P1).

**Copy-as-Highlight (T2):** To support lightweight, non-committal comparison, users can copy a visual mark and drop it into another view to trigger a transient highlight (Fig 2). Unlike drag-as-filter, copy-based highlights do not alter the data shown, but instead emphasize relevant subsets without changing the global filter state. These highlights remain until manually removed, supporting revisitation and short-term memory extension (P1, P2).

**Hover-and-Lock Selection (T3):** Hovering over a visual mark triggers cross-view highlights in all coordinated views. When a user clicks on the hovered mark, the highlight “locks” in place, making the selection persistent. This mechanism bridges ephemeral exploration with stable state construction, enabling users to incrementally build context while keeping the interface responsive and low-effort (P1).

**Pop-out (T4):** For detail-on-demand, any visual element can be right-clicked to spawn a floating micro-view (Fig 3). These views present focused information (e.g., detailed timelines, entity relations) without disrupting the main dashboard. Micro-views can be repositioned, layered, and bookmarked, supporting parallel exploration and progressive sensemaking [18, 14] (P1, P2).

## 4 FLUIDVIEWS TECHNIQUE

To implement the DFM framework, we developed *FluidViews*, a modular, event-driven multi view system that coordinates interactive operations across heterogeneous visualization types. This section describes the architectural components, data propagation mechanisms, and drag-and-drop infrastructure that together implement the four primary interaction techniques (T1–T4). Our goal is to ensure responsive, low-latency coordination filtering across views, while preserving analytic continuity through shared context and persistent interaction states.

### 4.1 Architecture

FluidViews is built around a modular, extensible workspace that supports four coordinated visualization types representing different views of transactional data: (1) a force-directed radial network graph encoding city-to-merchant relationships, (2) a geographic heatmap showing state- and city-level aggregates, (3) a multipartite Sankey diagram visualizing flows from State → City → Occupation → Merchant, and (4) a stacked time-series histogram displaying temporal distributions of transaction volume.

Each view is embedded within a draggable, resizable container, allowing analysts to dynamically arrange their workspace to suit evolving tasks. These containers expose standard interaction hooks and subscribe to a global interaction context, enabling views to remain loosely coupled yet tightly synchronized. Visualization modules are implemented in D3.js and React, with view updates triggered by changes in shared application state.

### 4.2 InteractionContext

A central architectural element in FluidViews is the *InteractionContext*. A global, observable state container built using React’s context system [1]. This shared context serves as the communication bus for all filtering, highlighting, and selection events across the system.

When a user initiates an interaction (e.g., dragging a city node, hovering over a histogram bin), the source visualization generates a structured payload that encodes key metadata about the entity: a unique identifier, type (e.g., city, occupation), origin view, and optionally associated data attributes (e.g., time range, category label). This payload is passed to the *InteractionContext*, which broadcasts it to all subscribed visual components.

Receiving views apply this payload in a view-specific manner. For instance, dragging a city node and dropping it onto the histogram triggers a scoped filter that updates the histogram to show

only transactions related to that city. At the same time, the network and Sankey views highlight nodes and paths corresponding to the same city entity. This unified mechanism ensures consistent semantic interpretation of interaction intent, while allowing each view to maintain autonomy over visual encoding.

The event propagation follows a unidirectional flow: user gesture → payload generation → context update → view reaction. This architecture ensures predictable, side-effect-free updates, promotes testability, and supports future extensibility.

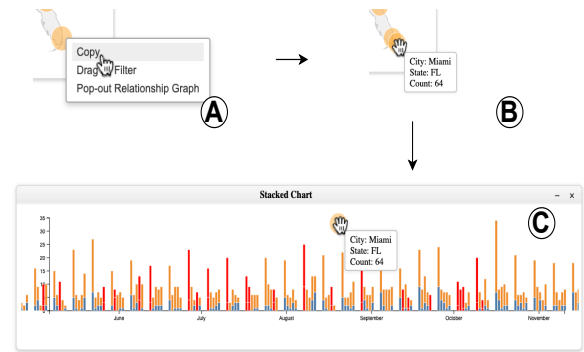


Figure 2: FluidViews’ *Copy-as-Highlight* workflow. (A) The analyst right-clicks a visual mark and chooses *Copy*, creating a draggable duplicate token. (B) The token is carried across the visualization while retaining the mark’s data identity. (C) Dropping the token onto the stacked time-series chart transiently highlights all bars associated with that entity, enabling rapid cross-view comparison without changing existing filters.

### 4.3 Drag-and-Drop

To support (T1) Drag-as-Filter (Fig 1) and (T2) Copy-as-Highlight (Fig 2), FluidViews implements a unified drag-and-drop infrastructure that binds low-level browser events to high-level analytic actions. Visual objects in all views are equipped with event listeners for *dragstart*, *dragover*, and *drop*, with event handlers registered dynamically based on the visual type and data binding.

**Drag Initiation:** When a user right-clicks a visual element and selects either *Drag as Filter* or *Copy*, the system creates a *dragObject* containing the selected mark’s metadata: entity ID, entity type, view of origin, and any contextual parameters. This payload is serialized and attached to the browser-native drag event.

**Drop Handling:** Drop targets in receiving visualizations decode the drag payload and invoke appropriate handlers. For Drag-as-Filter, the payload is interpreted as a filter predicate, which updates the *InteractionContext*’s selection state. For Copy-as-Highlight, the payload triggers a visual highlight operation without modifying the filter state. In both cases, the receiving view updates immediately, providing real-time feedback and maintaining the illusion of continuity.

**Filter Persistence:** Drag-as-Filter interactions instantiate persistent tokens in a filter tray, reflecting active filters in the workspace. These tokens remain visible and modifiable, promoting user awareness and traceability of active constraints. While not a standalone interaction, this design complements the drag interaction by preserving its analytic footprint.

### 4.4 Hover-Highlight and Click-to-Lock

To support (T3) Hover-and-Lock Selection, all visual marks in FluidViews expose hover event bindings. Hovering over a visual element (e.g., a Sankey node or heatmap cell) dispatches a highlight payload to the *InteractionContext*, which then propagates to other subscribed views. The system uses soft highlighting (e.g., opacity

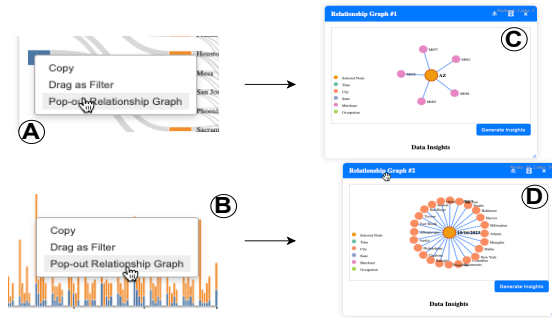


Figure 3: Pop-out micro-views in FluidViews. (A) From the Sankey diagram, the user invokes the Pop-out micro-views to inspect a selected occupation, triggering a focused micro-view (C) that visualizes first-hop entity relationships. (B) Similarly, from the stacked time-series chart, a right-click on a visual mark spawns a pop-out graph (D) showing temporal connections for the selected city. These floating, spatially independent micro-views allow analysts to perform targeted inspections without disrupting the main dashboard context.

changes, stroke overlays) to draw attention to semantically related elements without disrupting layout.

To transition from transient exploration to persistent comparison, users can click on a hovered item to “lock” the highlight. This promotes analytic continuity by allowing users to accumulate focus without needing to re-hover. Locked selections persist until explicitly cleared, supporting multi-step comparisons and enabling advanced inspection workflows.

The hover-lock interaction model is lightweight and non-intrusive, providing a middle ground between traditional brushing and more heavyweight filter construction. It encourages rapid exploration while giving users fine-grained control over attention management.

#### 4.5 Pop-out Micro-Views for Detail-on-Demand

FluidViews also supports (T4) Pop-out Micro-Views to enable focused, context-preserving inspection. Right-clicking on any visual mark opens a context menu with the option to spawn a micro-view (Fig 3). These pop-outs are floating containers that render secondary views such as zoomed-in timelines, entity-specific Sankey flows, or localized heatmaps based on the selected data entity. Micro-views are spatially independent and persist across dashboard interactions, enabling analysts to construct side-by-side comparisons, stage insights, or bookmark specific threads of analysis. Each micro-view is linked to its originating entity and updates dynamically if the underlying data changes.

Users can reposition, resize, and layer these containers to suit complex exploratory workflows. This design extends classic detail-on-demand interactions [18], while supporting modern analytic behaviors such as parallel investigation, multi-entity tracking, and iterative drill-down.

### 5 EVALUATION

To assess the effectiveness and usability of FluidViews, we conducted a qualitative user study focusing on how the system supports multi-view exploration and coordination tasks. Three participants with prior experience in data analysis and visual interfaces participated: a professional data scientist, an HCI researcher, and a graduate student specializing in visualization. Each session lasted approximately 45 minutes and included a structured introduction,

hands-on exploration of a multi-dimensional transactional dataset, and a post-session semi-structured interview.

Participants were presented with three core tasks reflecting realistic exploratory goals: (1) identify regional variations in merchant activity using the geographic heatmap and time-series view, (2) trace occupation-based flows in the Sankey diagram and correlate with merchant types, and (3) investigate time-based anomalies for specific cities using pop-out micro-views. All participants were encouraged to verbalize their thought process, and their interactions and comments were recorded for subsequent analysis.

Participants quickly adapted to the interface, particularly the drag-and-drop filtering and hover-based selection mechanisms. No formal training was needed beyond a two-minute walkthrough. All users noted that persistent filter tokens and copy-as-highlight helped them layer and compare multiple analytic hypotheses. These artifacts served as visual anchors and supported tracking across views. One participant said, “Even after five steps, I didn’t lose track of what I filtered, everything was there in front of me.” Participants frequently used a combination of hover-to-lock and drag-as-filter to probe hypotheses, lock insights, and propagate context. This coordination enabled fast, iterative discovery. For example, one participant locked highlights of two occupations, then filtered by city to compare distributional effects across time. The micro-view functionality was used extensively to isolate and inspect sub-patterns. Two participants created multiple micro-views in parallel, aligning them to compare temporal shifts between entities. This emergent behavior suggests the value of the design in supporting nonlinear exploration paths. Despite overall positive reception, participants identified several challenges. Two noted the absence of an *undo* action after misplaced filters. Managing multiple micro-views became cumbersome without layout snapping or grouping support. One participant suggested a command center or summary view to monitor all active filters and highlights.

### 6 DISCUSSION AND LIMITATIONS

The results of the preliminary study support our core design rationale: that embedding filtering and selection within the views, and making these interactions persistent and manipulable, can improve users’ ability to construct, coordinate, and refine complex queries across multiple dimensions. The participants’ ability to discover and adopt new interaction sequences without training indicates that the DFM approach has strong potential for supporting sensemaking in visual analytics environments.

The study, however, is subject to limitations. With only three participants and a single dataset, the findings are exploratory and not statistically generalizable. All users had at least moderate prior experience with interactive visualizations, which may have positively biased usability perceptions. The evaluation did not measure task performance quantitatively; future work should include comparative timing, accuracy, and cognitive load assessments.

Moving forward, we plan to conduct larger controlled studies comparing FluidViews to traditional coordinated systems that rely on static side panels and filter dialogs. We are also exploring mechanisms for adaptive view layouts, filter grouping, and collaborative provenance trails to enhance traceability and group analysis scenarios. Ultimately, we believe that the Direct Filter Manipulation framework opens a promising direction for building more expressive and context-aware analytic interfaces.

### 7 CONCLUSION

FluidViews introduces a direct-manipulation framework for multi-view visual analytics, enabling users to construct persistent filters, trigger transient highlights, and access on-demand micro-views through intuitive drag-and-drop gestures. By embedding these interactions directly within the visualization space, our approach addresses key limitations of conventional systems by reducing re-

liance on external filter panels, minimizing context switches, and supporting more fluid cross-view exploration. Future work will extend FluidViews to new analytic domains and explore adaptive layouts for multi-token interactions.

## REFERENCES

- [1] React.js. Accessed: 2025-04-19. 3
- [2] C. Ahlberg and B. Shneiderman. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, p. 313–317. Association for Computing Machinery, New York, NY, USA, 1994. doi: 10.1145/191666.191775 2
- [3] M. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. pp. 110–119, 01 2000. 1, 2
- [4] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, May 1987. 2
- [5] C. Collins and S. Carpendale. Vislink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1192–1199, 2007. doi: 10.1109/TVCG.2007.70521 2
- [6] C. Dunne, N. Henry Riche, B. Lee, R. Metoyer, and G. Robertson. Graphtrail: analyzing large multivariate, heterogeneous networks while supporting exploration history. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, p. 1663–1672. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2207676.2208293 2
- [7] M. Hearst. Design recommendations for hierarchical faceted search interfaces. In *ACM SIGIR workshop on faceted search*, pp. 1–5. Seattle, WA, 2006. 1, 2
- [8] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, 2007. doi: 10.1109/TVCG.2007.70539 1
- [9] M.-E. Hernandez, S. M. Falconer, M.-A. D. Storey, S. Carini, and I. Sim. Synchronized tag clouds for exploring semi-structured clinical trial data. In *Conference of the Centre for Advanced Studies on Collaborative Research*, 2008. 1
- [10] P. Koytek, C. Perin, J. Vermeulen, E. André, and S. Carpendale. Mybrush: Brushing and linking with personal agency. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):605–615, 2018. doi: 10.1109/TVCG.2017.2743859 2
- [11] Z. Liu and J. Heer. The effects of interactive latency on exploratory visual analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 20:2122–2131, 12 2014. doi: 10.1109/TVCG.2014.2346452 2
- [12] P. Mindek, M. Gröller, and S. Bruckner. Managing spatial selections with contextual snapshots. *Comput. Graph. Forum*, 33(8):132–144, Dec. 2014. doi: 10.1111/cgf.12406 2
- [13] C. North and B. Shneiderman. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pp. 128–135, 2000. doi: 10.1145/345513.345282 1, 2
- [14] C. Perin. Direct manipulation for information visualization. 2014. 3
- [15] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of International Conference on Intelligence Analysis*, vol. 5, pp. 1–6, 2005. 2
- [16] C. Quijano-Chavez, N. Doerr, B. Lee, D. Schmalstieg, and M. Sedlmair. Brushing and linking for situated analytics. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 597–603, 2024. doi: 10.1109/VRW62533.2024.00116 2
- [17] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proceedings of International Conference on Coordinated and Multiple Views in Exploratory Visualization*, pp. 61–71. IEEE, 2007. doi: 10.1109/CMV.2007.20 1
- [18] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pp. 336–343, 1996. doi: 10.1109/VL.1996.545307 2, 3, 4
- [19] M. J. Sifer. Filter co-ordinations for exploring multi-dimensional data. *J. Vis. Lang. Comput.*, 17:107–125, 2006. 1
- [20] H. Song, B. Kim, B. Lee, and J. Seo. A comparative evaluation on tree visualization methods for hierarchical structures with large fan-outs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 223–232, 2010. 1
- [21] M. Steinberger, M. Waldner, M. Streit, A. Lex, and D. Schmalstieg. Context-preserving visual links. *IEEE Trans. on Visualization & Computer Graphics*, 17(12):2249–2258, 2011. doi: 10.1109/TVCG.2011.183 2
- [22] C. Stolper, A. Perer, and D. Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Transactions on Visualization and Computer Graphics*, 20:1–1, 12 2014. doi: 10.1109/TVCG.2014.2346574 2
- [23] C. Stolte, D. Tang, and P. Hanrahan. Polaris: a system for query, analysis, and visualization of multidimensional databases. *Communications of the ACM*, 51(11):75–84, 2008. 1
- [24] M. Sun, A. R. Shaikh, Y. Ma, D. Koop, and H. Alhoori. Boundary blending: Reconsidering the design of multi-view visualizations, 2023. 2
- [25] A. Ulmer, M. Angelini, J.-D. Fekete, J. Kohlhammer, and T. May. A survey on progressive visualization. *IEEE Transactions on Visualization and Computer Graphics*, 30(9):6447–6467, 2024. doi: 10.1109/TVCG.2023.3346641 2
- [26] C. Viau and M. J. McGuffin. Connectedcharts: Explicit visualization of relationships between data graphics. *Comput. Graph. Forum*, 31(3pt4):1285–1294, June 2012. doi: 10.1111/j.1467-8659.2012.03121.x 2
- [27] M. Waldner, W. Puff, A. Lex, M. Streit, and D. Schmalstieg. Visual links across applications. In *Proceedings of Graphics Interface 2010*, GI '10, p. 129–136. Canadian Information Processing Society, CAN, 2010. 2
- [28] C. Weaver. Cross-filtered views for multidimensional visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):192–204, 2009. doi: 10.1109/TVCG.2009.94 2
- [29] K. Wongsuphasawat, D. Moritz, A. Anand, J. D. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22:649–658, 2016. 2